# TRAFFIC GENERATOR FOR FIREWALL TESTING

**A Thesis Submitted to**
**The Graduate School of Engineering and Sciences of**
**İzmir Institute of Technology**
**In Partial Fulfillment of the Requirements for the Degree of**

**MASTER OF SCIENCE**

**in Computer Software**

**by**
**Özgür KAYA**

**July 2009**
**İZMİR**

We approve the thesis of **Özgür KAYA**

_____

**Assist. Prof. Dr. Tuğkan TUĞLULAR**
Supervisor

_____

**Assist. Prof. Dr. Tolga AYAV**
Committee Member

_____

**Prof. Dr. Şaban EREN**
Committee Member

**6 July 2009**

_____

**Prof. Dr. Sıtkı AYTAÇ**
Head of the Computer Engineering
Departmant

_____

**Prof. Dr. Hasan BÖKE**
Dean of the Graduate School of
Engineering and Science

# ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my advisor, Assist. Prof. Dr. Tuğkan TUĞLULAR, for his guidance.

I would also like to acknowledge my institution, Yaşar University, for providing me research facilities and the test bed during my graduate study.

Special thanks go to my Head of the Computer Engineering Department in Yasar University, Korhan KARABULUT, for his understanding and guidance. He provided me salutary study environment and many helpful suggestions.

I am thankful to Mr. Graeme Hanssen for his inspirations about critical thinking.

Last but not least, I wish to express my gratitude to my family for their encouragement support and patience during the course of my studies

# ABSTRACT


## TRAFFIC GENERATOR FOR FIREWALL TESTING

Firewalls lead at the front line of a computer network to restrict unauthorized access. The desired security level is determined by a policy and implemented by a firewall which not only has to be effective but also stable and reliable service is expected. In order to verify the level of security of the system, testing is required.

The objective of this thesis is to test a firewall with software testing techniques taking into consideration the nominated policy and the firewall. Iptables software was examined and tested by two different algorithms that were modified according to software testing techniques, and the results were observed. Packets sent through the Firewall Under Test (FUT) are compared to packets passed through the FUT and test results were observed. The security performance of the modified algorithms proved to be successful.

# ÖZET

## GÜVENLİK DUVARI SINAMA AMAÇLI AĞ TRAFİK ÜRETECİ

Güvenlik duvarları, bir bilgisayar ağı üzerinden izinsiz erişim sağlamayı amaçlayan ataklara karşı oluşturulan savunma mekanizmasının ön sıralarında yer almaktadır. Hem güvenilir ve istikrarlı, hem de performanslı bir hizmet saglaması önemlidir. Bu doğrultuda, arzu edilen güvenlik seviyesinin sağlanabilmesi, oluşturulan güvenlik politikası ile mümkün olmaktadır. Güvenlik duvarının, beklenen güvenliği ne ölçüde karşıladığını doğrulamak için ise, sistem test edilmelidir.

Bu çalışmada, güvenlik duvarlarının izledikleri politika göz önüne alınarak yazılım testi yöntemleri ile sınanması hedef alınmıştır. Güvenlik duvarı uygulaması için seçilen Iptables yazılımı, geliştirililen algoritmalar ile 2 farklı yazılım testi uygulanarak sınanmış ve test sonuçları gözlenmiştir. Test edilmekte olan güvenlik duvarına gönderilen test paketleri, güvenlik duvarının arkasına geçen paketler ile karşılaştırılarak test sonuçlarına karar verilmiştir.

# TABLE OF CONTENTS

# LIST OF FIGURES

**Figure** **Page**

# LIST OF TABLES

# CHAPTER 1

# INTRODUCTION

A firewall controls network traffic to and from a computer, based on a security policy. Although there was an omission from most of the literature of firewall studies, that is the systematic testing of firewalls, recent studies on firewalls began to fill this gap.

Frantzen et al. proposed that given the large number of firewall vulnerabilities that have surfaced in recent years, it is important to develop a comprehensive framework for understanding both what firewalls actually do when they receive incoming traffic and what can go wrong when they process this traffic. They studied on a Framework for understanding vulnerabilities in firewalls using a dataflow model of firewall internals [1].

Dataflow based adequacy criteria [2, 3] provides a test set to satisfy certain define-use (DU) associations that exist in the code. On the tests, achieving DU coverage level generally requires larger test sets and it is concluded frequently that higher detection rates achieved by the coverage based test, but with lower confidence the DU coverage sets had better fault detection also achieving 100% coverage. This is not necessarily a good indication that the test is adequate since the differences in fault detections are not statistically significant according to [4].

The studies based on data flow testing [5, 6, 7, 8, 9, 10] has been restricted to testing data dependencies that exist within a procedure which requires information about the flow of data including calls and returns, across procedure boundaries. Intra-procedural data flow tests focus on source code by building and searching program's def-use graph and determines the dependencies or definition use pairs. Although existed inter-procedural data flow algorithms cannot provide information about locations of definitions needed for inter-procedural data flow testing, but consist of determining the def-use information and guiding selection and execution of test cases that meet requirements [11].

Zaugg worked on the test packet driven approach of firewall testing including two phases: the identification of appropriate test cases that examine the behavior of the firewall

and the practical performance of these tests where the study focuses on the second phase [12].There are three general approaches to firewall testing: Penetration testing, testing of the firewall implementation and testing of the firewall rules [12]. However, most of the academic work concentrates on the firewall rules and assume the implementation is error-free [13]. This study focuses on the implementation testing of the firewall with software testing techniques and aims to test Iptables Firewall.

This thesis introduces a Packet Generation Framework for Firewall Testing. The thesis is composed of five chapters. Chapter 2 gives some background information about software testing techniques, types of firewalls and recent works about firewall testing. Chapter 3 explains the details of conformance testing of firewall and equivalence partitioning testing of firewalls on which IPTABLES firewall is used in Linux platform. Moreover, database design and firewall configuration also explained in this section. Chapter 4 describes experiments which are performed using Packet Generation Framework. Finally, Chapter 5 gives the conclusion of this thesis work.

# CHAPTER 2

# BACKGROUND

In this chapter, software techniques with comparisons and types of firewalls are explained. In addition, the usages of software testing methods for firewall testing are reviewed.

## 2.1. Software Testing

Software testing is an activity which aims to reveal possible defects in computer software [14]. Moderately software or a system have an implementation with specifications, according to testers perspective, it requires different approaches based on amount of knowledge. By the complete knowledge of software or system with data flows, source codes a structural test can be applied which is called white box testing. Otherwise the system can be identified as "unknown" according to its specifications, therefore a functional test would be more appropriate approach namely its black box testing. As referred in [15:15], Myers summarizes the important aspects of the testing principles in 10 steps, namely:

1. "A necessary part of a test case is a definition of the expected output or result.
2. A programmer should avoid attempting to test his or her own program.
3. A programming organization should not test its own programs.
4. Thoroughly inspect the results of each test.
5. Test cases must be written for input conditions that are invalid and unexpected, as well as for those that are valid and expected.
6. Examining a program to see if it does not do what it is supposed to do is only half the battle; the other half is seeing whether the program does what it is not supposed to do.
7. Avoid throwaway test cases unless the program is truly a throwaway program.
8. Do not plan a testing effort under the tacit assumption that no errors will be found.

9. The probability of the existence of more errors in a section of a program is proportional to the number of errors already found in that section.
10. Testing is an extremely creative and intellectually challenging task."

As offered in these steps, a well defined test case requires a purpose definition with an appropriate input to serve that purpose and an expected output. Also a successful test case needs to be unique, compare to other test cases in the way of uncovering defects.

In this thesis, a black box approach is considered and a comparison to white box approach is made for stating the differences better. In black box approaches, equivalence partitioning is one of the techniques applied in proposed approach and it is compared to random testing to state difference with a reason to be selected. The other approach is conformance testing, performed in proposed approach. Also a penetration testing can be explained as a complementary to conformance testing.

### 2.1.1. Black Box Testing

Black box testing focuses on behaviors of the software based on functional requirements. Consequently, Input/Output conditions are based on no knowledge of internal logic about the software's structure. The test cases are derived from specification of the requirements [16:161-166].The studies about automated test case generation by FSM can be referred as [17].

The number of driven test cases should contain all the possible function specifications to achieve reasonable testing and the test cases should reveal the presence or absence of the possible types of errors as referred in [18]

The observation of test results reveal the difference between how the software acts and how its intent to behave, under given input. Each specification of the software is compared with expected behavior. All functional requirements of the system are considered to be fully exercised by black box testing [19].

Generally black box testing aims to find errors that can be categorized as [20:459-468]:

- “Incorrect or missing functions
- Interface errors
- Errors in data structures or external data base access
- Behavior or performance errors
- Initialization and termination errors”

Black box testing mostly focuses on functionality of software. Compare to white box testing, it is applied in further stages of testing where a control flow and structural design is not considered. Only the information domain is regarded as an input to apply test cases.

### 2.1.2. White Box Testing

White box testing is also called Structural Testing or Logic Deriving Testing [15:11]. As the name implies, information about the structural design of the program and source code are required to derive test cases as a source. The determination of suitable test cases- derived from the structure of the software- is considered with the help of control flow, event sequence graphs or some other models.

Since a model such as data control flow is required to derive test cases, every possible path in that graph needs to be examined at least once to complete the test of software. But as stated in [15:9-14], there are some disadvantages which make it infeasible. First one is the possibility of huge unique path exists in graphs; of that the cause high cost not only to determine uniqueness but also to execute each of them at least once, which is called exhaustive input test. Second one is, however each path should ever executed at least once, there is a path required to be executed. In case of missing paths, software runs incorrect. And there is also a possibility of absence in this missing patch that would not be recognized or examined paths would not uncover the error.

### 2.1.3. Black Box vs. White Box Testing

As the name offers Black box testing [21], also called functional testing, derives test cases from the specification of the SUT, not the implementation where white box testing considers within. Furthermore in white box testing every statement or branch of the SUT is required to be executed at least once with some techniques. For the detailed information and comparisons of those white box techniques such as statement testing, branch testing, data-flow testing [18] is referred. Since black box testing focuses on functions of the system, it is applied at further stages comparison to white box testing, where it is applied at early stages of life cycle [20:460].

"Functional testing can be applied at any level of granularity where some form of specification is available, from overall system testing to individual units, although the level of granularity and the type of software influence the choice of the specification styles and notations, and consequently the functional testing techniques that can be used" [16:161-166]. All levels of granularity can be functionally tested and still some forms of specifications will be available, e.g. overall system testing and even as far as individual unit testing will be possible. Specification styles and notations can influence the level of granularity and the type of software that is being used.

However, there is a particular level of granularity which cannot be run over because structural and fault based testing techniques are invariably tied to specific program structures. Fine-grain program structures (statements, classes , etc.) are usually tested by common structural tests, and these tests can only be used at modules which have small levels, or from another point of view and small collection of modules (small subsystems, component, or libraries) [16]. Also in [22], by analyzing structure of the specifications, its shown how to apply white box testing techniques into black box testing under the condition of requirements are implemented in formal model.

### 2.1.4. Equivalence Partitioning

It is mentioned previous chapters how derive a test case from a model and how to decide correct test case to perform. There is another approach called, Equivalence Partitioning; that testing aims to limit infinitely many test cases, which causes the same problem we have mentioned in white box testing namely exhaustive input test, into a small subset of test cases. But there are 2 important properties of each test case partitioned in a subset defined by Pressman in [13] such as:

- "It reduces, by more than a count of one, the number of other test cases that must be developed to achieve some predefined goal of "reasonable" testing"
- "It covers a large set of other possible test cases. That is, it tells us something about the presence or absence of errors over and above this specific set of input values"

The first property aims to minimize the possible input domain by defining a unique and most representative test case to uncover maximum number of possible input. The second property tells that if a test case that chosen from one partition detects an error, the other test case chosen from same partition acts exactly like first test case. That behavior must give satisfaction to all the test cases defined in same partition.

### 2.1.5. Random Testing

Each system requires an input to test its specifications for functional testing, namely black box testing and there should be a domain to select inputs. If any input from the domain is randomly selected, it's called random testing. With that approach, there come some questions such as, how many selections should be done? [23:66-67].

Of course, it's not always possible to know which test cases are better that can identify the defects. Since different test cases can more likely reveal defects then running the same test case, reasonably we can conclude that more different test cases are more valuable than similar selected test cases [16:161-166].

### 2.1.6. Random vs. Equivalence Partitioning Testing

Considering advantages of random testing, “Accidental bias may be avoided by choosing test cases from a random distribution. Random sampling is often an inexpensive way to produce a large number of test cases. If we assume absolutely no knowledge on which to place a higher value on one test case than another, then random sampling maximizes value by maximizing the number of test cases that can be created (without bias) for a given budget. Even if we do attain some knowledge suggesting that some cases are more valuable than others, the efficiency of random sampling may in some cases outweigh its inability to use any knowledge we may have” [16:161-166]. But it is assumed that random selection has less chance of select correct set of inputs to reveal defects in [18] [24] [25]. For further discussions [26] and [27] can be considered.

Partitioning testing, which separates the domain by grouping an infinite set into finite sets, increases the cost. But the size of domain is important factor by the cost of creating a partition to compare with random selection. But some knowledge about the system is required for that estimation [23:67-72].

We can assume in the case of, each sub domain [28] of domain is uniform, and then the input from the partitioning class reveals the same defects from any input from same partition. So the partitions must be disjoint (the case of joint, defined in [29]) to satisfy uniformity. Also the experience takes an important role to select more likely failure-prone test cases [16]. That comparison between random and partitioning testing is also considered by an analytical approach in [30] and by experimental results in [25].

### 2.1.7. Conformance Testing

It is also known as compliance testing or type testing [31]. The purpose of conformance testing is to determine the product, protocol, computer program or a system work or perform as intended. An implementation conforms to the specification based on formal standard. There are standards to conform the term “intended”. The conformance testing ensures that, each individual requirement of the specified standard works properly

where those standards are defined by independent institutes such as IEEE [32]. Each protocol implementation needs different formal definitions to state its specific aspects conformance to standards [33, 34, 35, 36, and 37].

In the test platform where conformance testing is applied to, System Under Test (SUT) factor takes an important role to define boundaries of test. The tests should be run by a system which can observe the SUT's attitude. Also that dedicated system must be isolated from external effects to avoid miscalculated results [38].

As seen in Figure 2.1. Conformance testing has phases namely, implementation, compilation, testing, logging and analysis. In compilation stage by defining the test purposes, test cases are generated. In testing stage, test suite is set up and the created test system uses test suite to apply testing. Before testing stage, the Product (IUT) needs to be implemented according to base standards. The test is performed by the test system to product. As a last stage the test report is generated by logs of SUT.

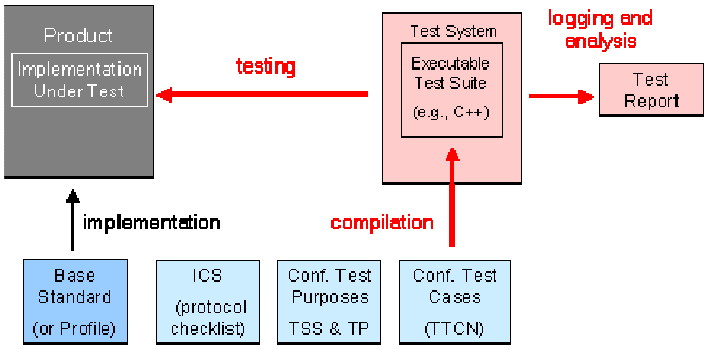



Figure 2. 1. Conformance Testing [67]

## 2.1.8. Penetration Testing

Penetration testing also known as Black Hat Hacker and the purpose is to identify the vulnerabilities which can be caused by system configuration or component flaws. According to black box approach, security analyze of the system is covered without any prior infrastructure knowledge which is a "blind" test or attack. On the other hand, white box approach requires system infrastructure and can reveals leaks, so it is considered as "inside job" test or attack [39].

From the perspective of functional testing, penetration testing considers with negative cases that cause system not as intended (produce fault), where functional testing considers only positives to test system. A positive test checks system behaviors with legal inputs and reveals the input-output nonconformance. But neither negative test cases reveals no error, it does not prove system has no faults because of very little assurance [40].

On the development life cycle of the application, penetration testing must be integrated in early phases to avoid implementation errors or systematic faults. Otherwise it uncovers defects or bugs too late which cost expensive. The integration phase at the system level, component testing can be used to check inter-component communication and global error handling, unit testing is used to divide security [40] into discrete parts. But root-cause architecture of vulnerabilities must be prevail [40].

Result interpolation such as bugs and vulnerabilities, is one of the major issues of penetration testing. Since the tests can identify small risks in early life cycle, the errors caused by mitigation of subsets can effect system at later life cycles which  The success of the tests depend on standards and metrics where tester skill, knowledge and experience lead with “software risk analysis” [41]. By use of tools provides efficient analysis which based on metrics [40].

## 2.2. Firewalls

Firewalls are softwares or hardwares which provide permission control, obstructing and filtering to unauthorized reaching come from on network or on internet. It can be separated in to three groups of software in Figure 2.2, hardware in Figure 2.3 or both of them to each other [42].Broadly speaking, a computer firewall, prevents unauthorized access to or from a private network. Firewalls are tools that can be used to enhance the security of computers connected to a network, such as LAN or the Internet. They are an integral part of a comprehensive security framework for your network.

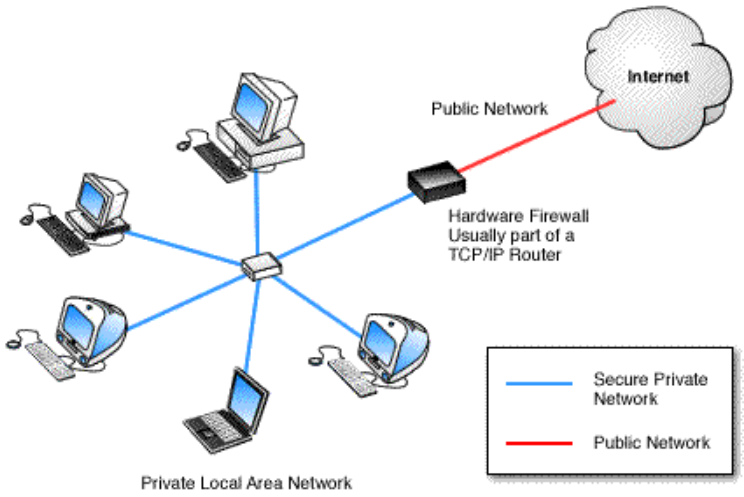



Figure 2. 2. Hardware Firewalls [68]

Firewall is one of the core element network and internet securities. However, managing firewall rules, especially for enterprise networks, has become complex and error-prone [43]. Firewall filtering rules have to be carefully written and organized in order to correctly implement the security policy [44]. In addition, inserting or modifying a filtering rule requires thorough analysis of the relationship between this rule and other rules in order to determine the proper order of this rule and commit the updates.

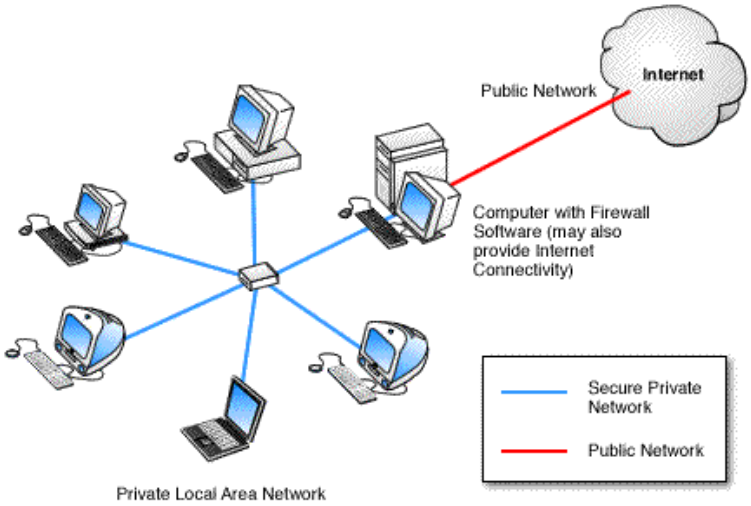



Figure 2. 3. Software Firewalls [68]

In theory, a firewall is supposed to isolate your computer from the Internet, absolutely using a "wall of code" that inspects each individual "packet" of data as it arrives at either side of the firewall to determine whether it should be allowed to pass or be blocked. Firewalls have the ability to further enhance security by enabling regular control over what types of system functions and processes have access to networking resources. These firewalls can use various types of signatures and host conditions to allow or deny traffic.

DMZ network serves as appendix for computer systems and sources that need to be accessible either externally or internally, but that could not be placed on internal protected networks [45] .DMZ networks are typically implemented as network switches that sit between two firewalls or between a firewall and a boundary router. Given the special nature of DMZ networks, they typically serve as attachment points for systems that require or foster external connectivity [45].

A DMZ is a set of machines under common administrative control, with a common security policy and security level. There are many good reasons to erect internal firewalls [43].

### 2.2.1. Types of Firewalls

Firewalls could be classifying three main categories. It could be arranging generally, packet filtering, circuit gateways, and application gateways [42] [43] [46]. Some sources mentions about fourth filtering of statefull inspection firewalls [45]. Traditionally, firewalls are placed between an organization and the outside world [43]. But a large organization may need internal firewalls as well to isolate DMZ (also known as administrative domains) [44]. DMZ which known as De Militarized Zone network, is created out of a network connecting two firewalls when two or more firewalls exist in the networks connecting the firewalls [45].

First category of firewalls, as seen in Figure 2.4, is application - layer gateway Contrast to packet filter firewall design, in this type of firewall, prefer to use general specified mechanism to allow many different kids of traffic to flow. The code can be used

for each desired applications. This type of firewalls search some protocols which work in application layer in OSI models [43].

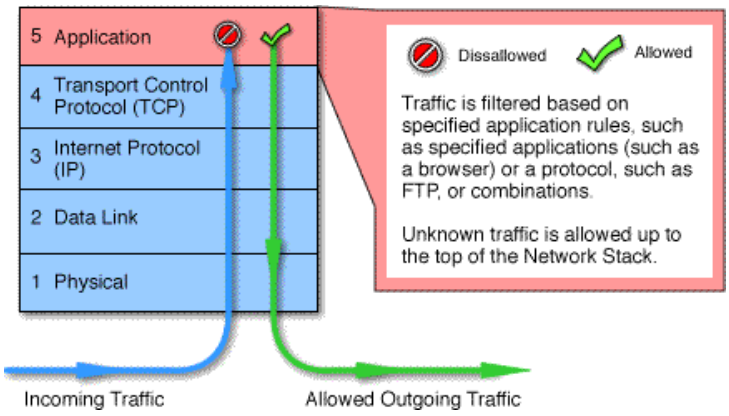



Figure 2. 4. Application Layer Firewall [69]

The second category of firewalls, as seen in Figure 2.5, is circuit - level gateway. Circuit gateways relay TCP connections. The caller connects to a TCP port on the gateway, which connects to some destination on the other side of the gateway. During the call the gateway's relay programs copy the bytes back and forth: the gateway acts as a wire. In some cases a circuit connection is made automatically [43]. For example, we have a host outside our gateway that needs to use an internal printer. We have told that host to connect to the print service on the gateway. Our gateway is configured to relay that particular connection to the printer port on an internal machine.

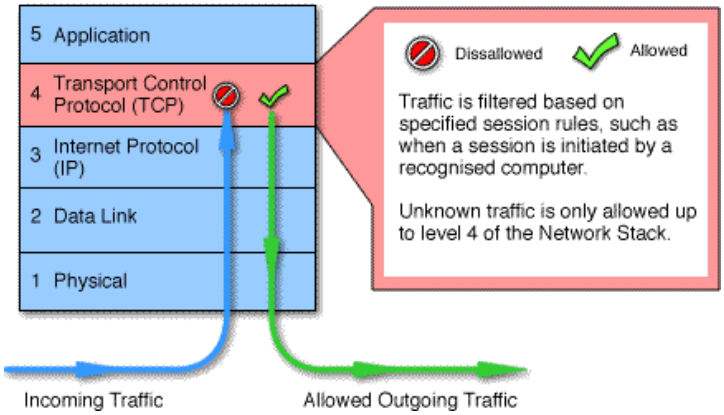



Figure 2. 5. Circuit Level Firewall [69]

The third category of firewalls, as seen in Figure 2.6, is packet - filtering gateways. This type firewalls control IP protocol, IP address and port number include own rules [47]. From there, the firewall would start at the top of the rule set and work down through the rules. In the table a sample of a packet filter firewall rule set for an imaginary network is shown. In packet filtering firewalls, the rule set would be much larger and detailed [45]. It is more troublesome than other firewalls [43]. A permission or denied rule is following some actions taken of accept, deny or discard.

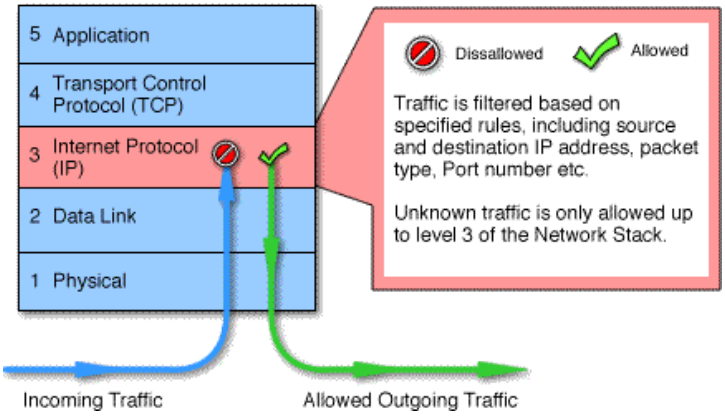



Figure 2. 6.Packet Filter Firewall [69]

### 2.2.2. Policy

So far it has been mentioned firewalls are an authority to control and monitor the network traffic between a local area network and internet. The restriction of unauthoritezied accesses is crucial. Therefore configuration of firewall is the most important step to restrict unauthorized access from internet. Different security levels require for different types of services. Referred to [45], building a firewall design requires 4 components, namely policy, authentication, packet filtering, application gateway.

The most effective way to provide security for firewalls is to design a correct and well defined policy to control network traffic pass through firewall. A firewall provides security mostly depends on the policy. As referred in [48], there are 5 stages to create a firewall policy such as zone of trust, administrator change, logging, stability and documentation.

Each zone stands behind firewall requires different trust levels, depends on the service they provide. Since firewall controls the traffic between different zones, a policy should be configured considering different traffic flow for each zone. Therefore zones must be identified correctly with the level of trust. As stated in Zone of Trust, firewall policy defines different rules which provide different security levels for different services. By the change of those services, policy rules need to be updated. Addition or modification of some rules, affects the policy. In case of firewall administrator change, such a scenario may arise that the new administrator can hesitate to modify or remove a rule when the policy requires an update. Consequently inserting a new rule would work but also override the some rules hierarchically. Far too often firewall may contain so many garbage rules which can cause an anomaly. So, well defined documentation of addicted or removed rules with date prevents that scenario. Also penetration testing and IDS can check the rules to verify agreed policy works as intended.

Firewall policy meets the security requirements as much as the rule set serves as it is intended. Because the behaviors of the firewall actions are defined in the policy by implemented rules, each rule almost contains some fields have information about source address of packet, destination address of packet, type of traffic namely protocol and the ports of communicated services [45].

Beside of controlling the input and output network traffic by accepting or denying packets, firewall can also log the traffic pass through. That should also be considered as primary purpose which affects security indirectly but efficiently. But a log is as useful as it is reviewed. A firewall should be monitored to check whether it is up and works as intended. Therefore it is ensured that firewall provides maximum security as it stands. Even a firewall, with well designed and well defined rules, is only as good as it is stable. Otherwise an intruder may find a way to avoid. Each rule in the firewall stands for the policy that provides security. Therefore every rule needs to be documented with required information about why that rule stands for and which rules it is cooperated with.

## 2.3. Tool Support

The implementation process devoloped in Windows XP SP3 platform. JDK 1.6.0u13 version of Java software with a JPCAP version 0.7 library is used for packet generation, sending and capturing. The generated traffic sent to the firewall, where IPTABLES version 1.4.1 is used in linux platform (ubuntu version 9.04) is used. According to firewall policy, accepted packets are passed through firewall and captured by a sniffer. The generated and captured packets are stored in MYSQL version 5.1 community server edition database.

That chapter is composed of 3 sub-chapters. Chapter 2.3.1 gives basic information about IPTABLES and configuration, chapter 2.3.2 explains the implementation details of traffic generation tool with JPCAP library and chapter 2.3.3 describes the database model.

### 2.3.1. IPTABLES

Iptables is the userspace command line program used to configure the Linux 2.4.x, 2.6.x and upper versions IPv4 packet filtering ruleset [63, 64]. It is targeted towards system administrators. Also, iptables is a software which could used for inspect Netfilter system into userspace have been in the core [64]. This system will provide properties of packet filtering firewalls, application level firewalls and statefull firewalls by means of Extensions which named widening package[64]. For example, it could made possible to use like IPS and IDS properties with Snort softwares [64]. There are three different tables in the iptables [64]. The mangle table, the nat table and the filter table [64]. Each table has its own built-in chains like the filter table has input, output and forward chains [64]. Each iptables rule, must specify the table and the chain within the table that it should be applied to [64].

The first table of iptables is mangle table. This table is used mainly for mangling packets.Among other things, the contents of different packets and that of their headers could changed. Examples of this would be to change the TTL is used to modify the Time To Live field in the IP header, TOS is used to set the Type Of Service or MARK [63]. The following marks are only effective in the mangle table.

The nat table is used mainly for Network Address Translation. " NAT " packets get their IP addresses altered, according to our rules. Packets in a stream only traverse this table once . Assume that the first packet of a stream is allowed. The rest of the packets in the same stream are automatically " NAT " or Masqueraded etc, and will be subject to the same actions as the first packet . These will, not go through this table again, but will nevertheless be treated like the first packet in the stream. This is the main reason why could not to do any filtering in this table, which discussion at greater length further on. The PREROUTING chain is used to alter packets as soon as they get in to the firewall. The OUTPUT chain is used for altering locally generated packets before they get to the routing decision. Finally, the POSTROUTING chain which is used to alter packets just as they are about to leave the firewall [64].

The filter table should be used exclusively for filtering packets. For example, it could DROP, LOG, ACCEPT or REJECT packets without problems, as it could in the other tables. There are three chains built in to this table. The first one is named FORWARD and is used on all nonlocally generated packets that are not destined for our local host. INPUT is used on all packets that are destined for our local host (the firewall) and OUTPUT is finally used for all locally generated packets [65].

Figure 2.7 will clarify to FORWARD chain process. If a packet have gotten into the first routing decision that is not targeted for the local machine itself and it will be routed through the FORWARD chain. If the packet is destined for an IP address that the local machine is listening to, the packet could sent through the INPUT chain and to the local machine [64].

First step in Figure 2.7, a packet comes in on the interface to mangle PREROUTING. This chain is normally used for mangling packets, changing TOS and so on. This is also where the non - locally generated connection tracking takes place. In second step, nat PREROUTING chain is used for DNAT mainly. First step in Figure 2.7, a packet comes in on the interface to mangle PREROUTING.

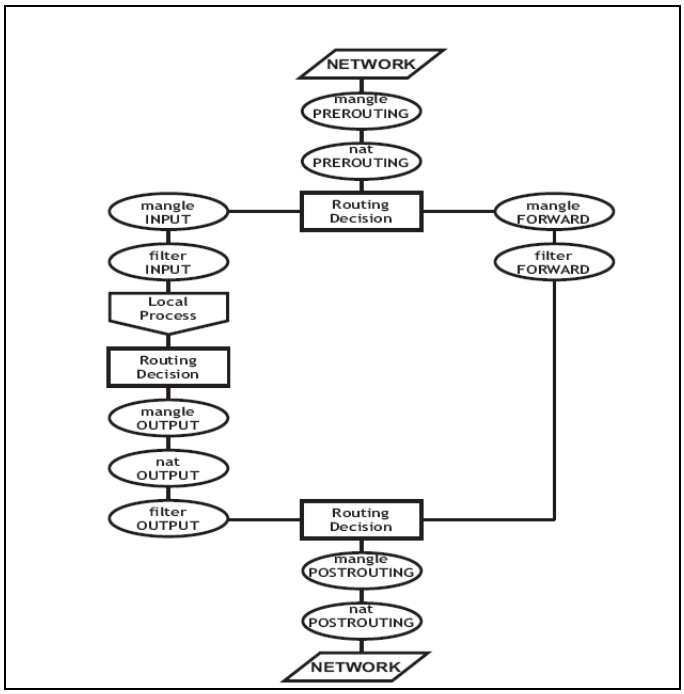



Figure 2. 7. Forward Chain of Filter Table

This can be used for very specific needs, where wanted to mangle the packets after the initial routing decision, but before the last routing decision made just before the packet is sent out. In forth step, the packet gets routed onto the FORWARD chain of the filter. Only forwarded packets go through here where doing all the filtering. In fifth step which mangle POSTROUTING is used for specific types of packet mangling that we wish to take place after all kinds of routing decisions have been done, but still on this machine. In sixth step of forward packets, nat POSTROUTING should first and foremost be used for SNAT. Avoid doing filtering here, since certain packets might pass this chain without ever hitting it. This is also where Masquerading is done. And the last, packets go out on the outgoing interface.

### 2.3.2. JPCAP Library

Packet Generation Software tool is implemented in Java by using JPCAP library which enables to generate, send and capture TCP packets. Packet generation is composed 3 of 4 phases which is based on a TCP/IP model layers as seen in Figure 2.8, namely:

1) Transport Layer
2) Internet Layer
3) Link Layer

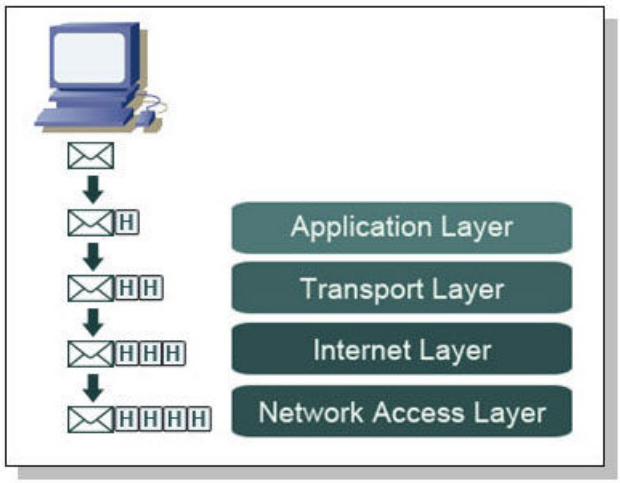
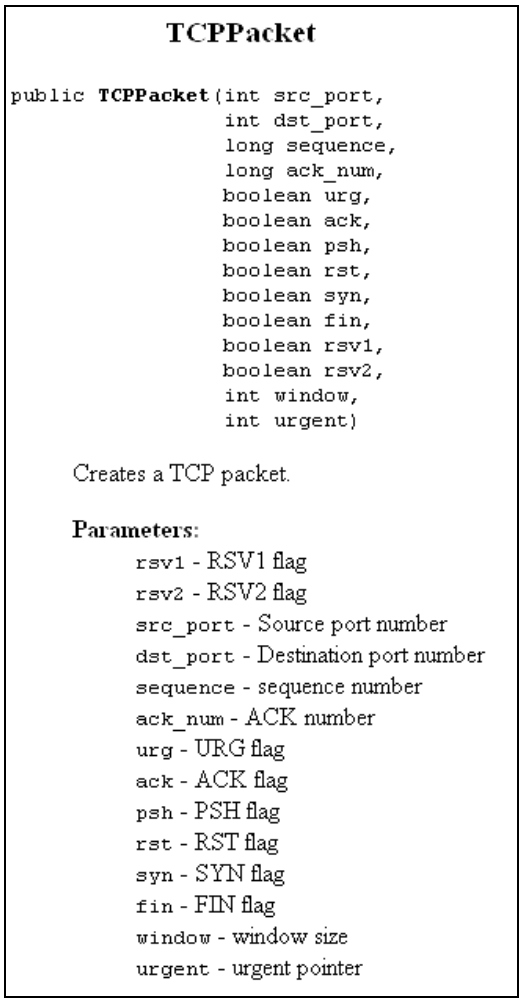



Figure 2. 8. TCP/IP Protocol Layers [70]

In Transport Layer, packet is constructed in TCP protocol and all flag , port information are specified. TCP packet generation requires datagram fields such as:Ack flag, Ack number, Destination port number, Fin flag, Tcp option, Psh flag, Rst flag, RSV1 flag, RSV2 flag, Sequence number, Source port number, SYN flag, URG flag, Urgent Pointer, Window size as seen in the Table 2.1. TCP packet constructor is explained in the Figure 2.9 with the fields and the required parameters.

Table 2.1. TCP Layer Fields

| Field Name | Explanation |
|---|---|
| Acknowledgment flag | Boolean variable stands for Acknowledgement number. When ack flag is set to true, Ack number field defines the next expected byte that the receiver is expecting and restricted up to 32 bits. |
| Destination port | Identifies the receiver's port number. |
| Source port | Identifies the sender's port number. |
| FIN flag | 1 bit Boolean field that tells to receiver, sender will not send no more data. |
| TCP option | optional multiple of 32 bits field. |
| PSH flag | 1 bit Boolean Push function field. |
| RST flag | 1 bit Boolean field to reset connection. |
| RSV1 and RSV2 | Reserved fields. |
| SYN flag | 1 bit Boolean field, stands for synchronizing sequence number. |
| Sequence number | 32 bit field, depends on SYN flag. If it's set true, it replaces the sequence number with actual first data bytes plus 1. If its set false then it replaces with the first data byte. |
| URG flag | 1 bit Boolean field and identifies the urgent pointer is set. |
| Urgent pointer | 16 Bits field, synchronizes the last urgent data byte from the sequence number. |
| Sequence number | 32 bit field, depends on SYN flag. If it's set true, it replaces the sequence number with actual first data bytes plus 1. If its set false then it replaces with the first data byte. |
| Window size | 16 Bits field and identifies the receiver's data size of able to receive. |



**TCPPacket**

```
public TCPPacket(int src_port,
                 int dst_port,
                 long sequence,
                 long ack_num,
                 boolean urg,
                 boolean ack,
                 boolean psh,
                 boolean rst,
                 boolean syn,
                 boolean fin,
                 boolean rsv1,
                 boolean rsv2,
                 int window,
                 int urgent)
```

Creates a TCP packet.

**Parameters:**

- `rsv1` - RSV1 flag
- `rsv2` - RSV2 flag
- `src_port` - Source port number
- `dst_port` - Destination port number
- `sequence` - sequence number
- `ack_num` - ACK number
- `urg` - URG flag
- `ack` - ACK flag
- `psh` - PSH flag
- `rst` - RST flag
- `syn` - SYN flag
- `fin` - FIN flag
- `window` - window size
- `urgent` - urgent pointer

Figure 2. 9. TCP Packet Constructor [71]

After that arguments set in the TCP generate function, it is encapsulated by an IP frame. ” In most protocol specifications related to the TCP/IP protocol suite, the term fragmentation rather than segmentation is used. The meaning is the same” [66] That encapsulation requires IPv4 parameter datagram fields such as: Priority, IP Delay flag bit , IP Through flag bit, IP Reliability flag bit, Type of Service, Fragmentation reservation flag, Don't fragment flag, More fragment flag, Fragment offset, Identification, TTL, Protocol, Source IP address, Destination IP address as seen in the Table 2.2. In internet layer, data adressing and packet fragmentation operations are completed. IP version 4 packet datagram fields with the required parameters are shown in the Figure 2.10.

Table 2. 2. IP Layer Fields

| Field Name | Explanation |
|---|---|
| Version | 4 bits field, specifies the packet is IP version 4 or IP version 6. |
| Priority | Priority of a packet. |
| IP Delay flag | Minimizes the Delay, 0 is normal delay, 1 is low delay. |
| IP Through flag | Maximizes the thorughput, 0 is normal, 1 is high. |
| IP Reliability flag | Set the reliability, 0 is normal, 1 is high. |
| Type of Service | Refers to Quality of Service and identifies the priority of packet. |
| Fragment offset | 13 Bits field and specifies the fragment that packet is attached. |
| Fragmentation reservation flag | 3 bits field and identifies the packet is allowed to be fragmented or not (Don't Fragment) or followed by more fragments (More Fragment). |
| Don't fragment flag | Used for packet recovery. |
| More fragment flag | Used for packet recovery. |
| Fragment offset | Used for packet recovery. |
| Identification | 16 Bits field that helps to recover packet from small pieces only enough to pass through, and uses the Fragmentation field with Don't fragmentation or More fragmentation fields. |
| TTL | 8 bits field, specifies the number of hops the packet pass through before it is discarded. |
| Protocol | 8 bits field identifies the protocol of packet. But when packet is encapsulated by higher level protocol such as TCP, that field is ignored. |
| Source IP address | Identifies the senders IP address in 32 bits field. |
| Destination IP address | Identifies the receiver's IP address in 32 bits field. |

**setIPv4Parameter**

```
public void setIPv4Parameter(int priority,
                             boolean d_flag,
                             boolean t_flag,
                             boolean r_flag,
                             int rsv_tos,
                             boolean rsv_frag,
                             boolean dont_frag,
                             boolean more_frag,
                             int offset,
                             int ident,
                             int ttl,
                             int protocol,
                             java.net.InetAddress src,
                             java.net.InetAddress dst)
```

Sets the IPv4 parameters

**Parameters:**

`d_flag` - IP flag bit: [D]elay
`t_flag` - IP flag bit: [T]hrough
`r_flag` - IP flag bit: [R]eliability
`rsv_tos` - Type of Service (TOS)
`priority` - Priority
`rsv_frag` - Fragmentation Reservation flag
`dont_frag` - Don't fragment flag
`more_frag` - More fragment flag
`offset` - Offset
`ident` - Identifier
`ttl` - Time To Live
`protocol` - Protocol
This value is ignored when this packets inherits a higher layer protocol(e.g. TCPPacket)
`src` - Source IP address
`dst` - Destination IP address

Figure 2. 10. IP Packet Parameters [71]

After IP layer, the packet is encapsulated in Link Layer by the pysical medium information and prosedure is specified data transmission. Ethernet header includes fields such as: destination mac address, source mac address, frame type as shown in the table 2.3.

Table 2. 3. Link Layer Header

| **Fielad Name** | **Explanation** |
|---|---|
| Destination mac address | 6 Bytes field of receiver's pysical interface address. |
| Source mac address | 6 Bytes field of sender's pysical interface address. |
| Frame type | Ethernet Frame Type. |

After packets are generated, JPCAP library enables to send them via sender methot which requires initializing network interface device and a packet as an input parameter. Sent packets can be captured by an capture methot which requires input parameters namely, network interface device, maximum number of bytes can be captured, promiscuous mode for destination mac address control and time out period to restrict incoming packets.

# CHAPTER 3

# PROPOSED APPROACH

Designed proposed approach aims to test specifications of the software, which is a firewall policy in our case. There are some steps for testing in our approach such as defining a firewall policy in a formal language, deriving firewall rules in terms of tuples, defining an algorithm to automatize test case generation, generating abstract test cases from tuples by that algorithm, generating packets from abstract test cases, applying a formal model to test firewall.

## 3.1. Formal Methods

The increasing complexity of systems requires well defined specifications in a systematic way which consists of mathematical models to observe system's behavior. Formal Description Technique (FDT) provides an implementation of a system with an implemented language with a syntax and semantics with formal descriptions. That makes it possible to question and analyze by using mathematical basics [49:4-8]. As an example of that standardization with mathematical models based on finite state machines are Estelle [61] and SDL [62] may be pointed.

Briefly, formal protocol development of a system requires firstly capturing phase where requirements of the system is collected, then specifications are defined and implemented from that requirements using FDT. By the combination of specifications and their implementations, correctness of the system can be experimented with tests derived from FSM and verified with a formal model [49:10-23].

## 3.2. Firewall Policy Formalization

Firewalls may have a different policy syntaxes adapted for their own algorithms. Therefore comparison or testing policies (if it works as intended) come up with problems of understanding different policies or comparing them by a common specification. That situation requires a formal definition for firewall policies to meet them in a common point. So, even different syntaxes can be interpreted in terms of their formal definitions.

As seen in the Figure 3.1. there is a network layout, contains internet, intranet and a firewall. Firewall has 2 interfaces for an input and output. Considering intranet (subnet) as a packet sender, eth 1 interface is an input and eth 0 interface is an output but considering packets come from internet, it is vice versa..

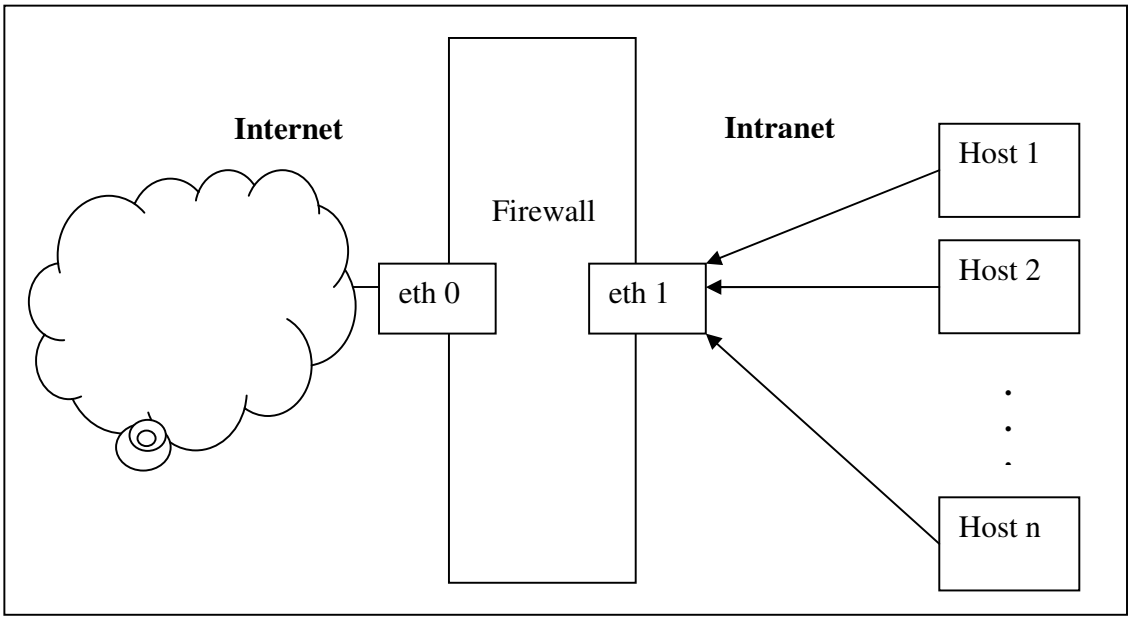



Figure 3. 1. Network Layout

Assume that, there is an internet access from intranet to internet for all subnet hosts from 1 to n. In our formal approach we state a subnet as a set that all the hosts are identical therefore the policy behaves same for each member of the subnet. The internet access from intranet to internet can be shown in Table 3.1.

In given Table 3.1 source is a subnet where hosts are participated in intranet. They request for a permission to access internet so the destination is an internet for the subnet

hosts in the intranet as a source and the restriction is allowed for a web access service. So the formal policy figured in Table 3.1 shows that each request from intranet to internet for a web access services is accepted by firewall as stated in the policy.

Table 3. 1. Formal Policy

| Source | Destination | Restriction | Permission |
|---|---|---|---|
| Intranet | Internet | Web Access | Accept |

But in case of some other requests apart from web access, there must be a rule defined in the policy, in out case default policy is deny. Table 3.2. shows that any other requests from intranet to internet is denied by the firewall policy.

Table 3. 2. Formal Default Policy

| Source | Destination | Restriction | Permission |
|---|---|---|---|
| Intranet | Internet | Web Access | Accept |
| Intranet | Internet | Any | Deny |

## 3.3. Formal Framework

Depends on configuration of test architecture, an abstract test case can be generated. That will state the how the system will be tested which is Implementation Under Test (IUT) as seen in Figure 3.2 for the comformance testing. The relationship between IUT and testing environment must be stated based on test purpose derived from requirements, then that step followed by development of an abstract test suite which is implementation independent. The next step is abstract test case generation for each test purpose [49:27-45]. As last step to apply testing on a real system, that test cases transformed into executable form. To interpret the results more accurately and consistently, meaningful test selections for execution can be chosen [50, 51, 52].

In [53], Test sequence generation technique approach can be used together with conformance testing where 4 (T-, U-, D- and W-) methods are used to generate test sequences for FSM, such as UIO, DS or Wset and helped compared to each other with pros and cons.

But sometimes for each input there isn't a completely specified output stated. That condition differs from state oriented (deterministic) models. Therefore the test generation from that input/output can be considered nondeterministic FSM [54, 55, 56, 57, 58]. And that should be handling as a problem which caused by an output sequences of the specification FSM, does not corresponds to input sequences, However "conforming implementation of FSM must produce all of them" [59].

## 3.4. Deriving Test Cases from FSM

Test case generation is a directly effective factor on determining performance and cost measurements. The utility model is used for test case generation from specifications (formal or informal) and this model aims for the improved performance, reduced cost in test design [16:246-250].Specifications may be indicated straightly as some form of finite state machine. As an example, "embedded control systems are frequently specified with State charts, communication protocols are commonly described with SDL diagrams, and menu driven applications are sometimes modeled with simple diagrams representing states and transitions" stated in [16]. Sometimes the finite state essence of systems is left implicit in informal specifications.

Occasionally, test cases are enclosed in infinite states of control or interactive systems. But still FSM model can be used, if these infinite states are simple. Considering a communication example that issues an network traffic such as incoming and outgoing packets are transmitted from different ports. But under the condition of services' content is disregarded , the protocol system still be considered as an FSM. Most common strategy for deriving test cases from FSM is to checking each state transmission which is specification of precondition and post condition pair [16].

## 3.5. Test Selection

As mentioned in previous Chapter 3.3, at test method's level of abstraction, a specific test purpose is used which is derived from the abstract test case phase. Although abstract test method is specific to environment requirements, there are large amount of test cases (or infinite) can be generated for FSM by generation algorithms.

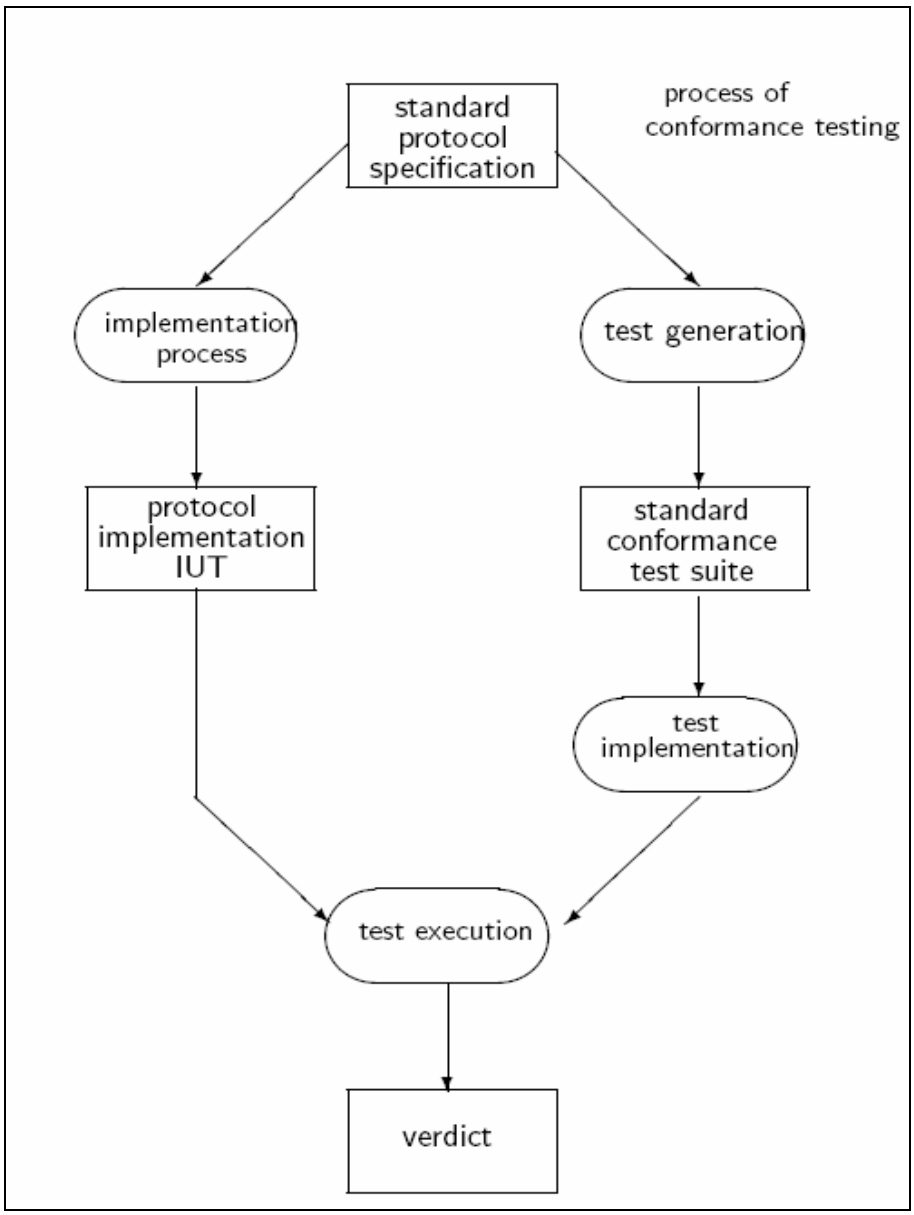



Figure 3. 2. Conformance Testing [49]

Proportionally that increases the cost of generation or makes it impossible to execute all test cases, if number of generated test cases are infinite. That should lead by a reduction in the size of test cases by an selection of meaningful portions. Apart from random selection, there can be used a heuristic from software testing [49:149-150] such as equivalence partitioning and boundary value analysis [18].

## 3.6. Rule Tuples

By the formal policy definition, we must set a standard syntax in terms of tuples. So the firewall policy can be given as an input to execute and generate abstract test cases by the test case generation algorithm.

An example network layer is stated in the Figure 3.3 with the rule stated in Table 3.3. "*" notation and any show all the possible values can be applied in that field. The intranet contains a subnet which is 193.140.248.* by the connection port any. Destination for the internet as an ip is considered as *.*.*.* but the port number is 80 for the web service.

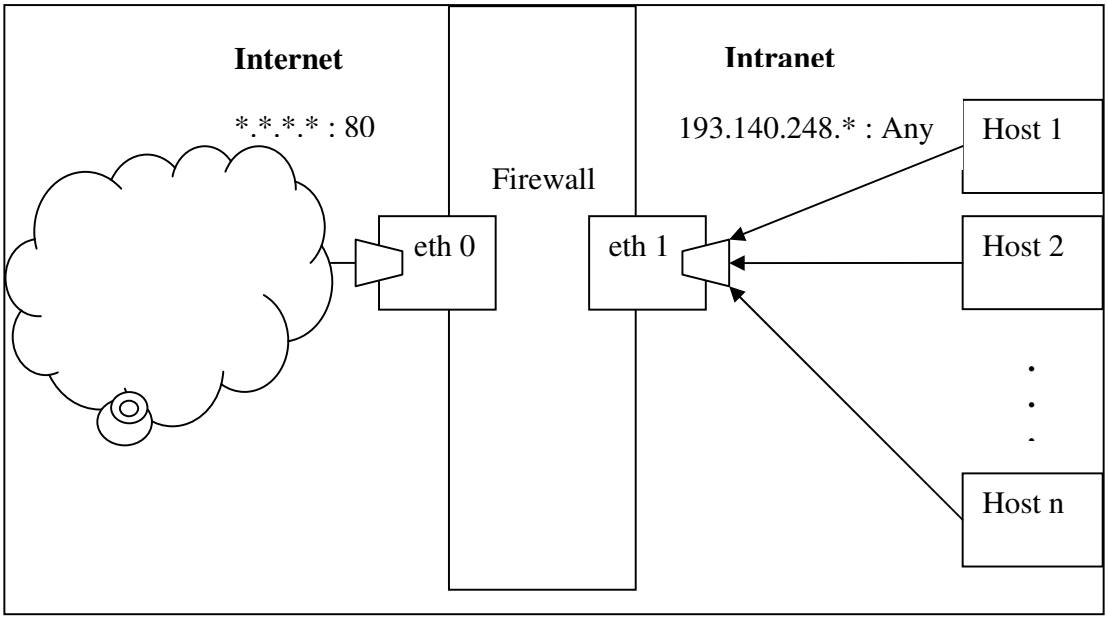



Figure 3. 3. Network Layer 2

Table 3. 3. Firewall Rule (Default Policy: Deny)

| Order | Protocol | Source IP | Source Port | Destination IP | Destination Port | Action |
|---|---|---|---|---|---|---|
| 1 | 6 | 193.140.248.* | Any | *.*.*.* | 80 | Accept |

As rewritten the policy in tuple form, it is:

- <1, 6, 193.140.248.*, Any, *.*.*.*, 25, Accept>.

The rule covers the packets generated from the tuple as:

- <6, 193.140.248.*, Any, *.*.*.*, 25>

with the fields; protocol, source ip, source port, destination ip, destination port.

IP address for source or destination of packets is considered in 3 cases such as host (193.140.248.1), subnet (193.140.248.*), and "*.*.*.*". In case of subnet ip address is given, the algorithm evaluates the subnet range for top, bottom and center values.

As an example;

193.140.248.1 is generated for bottom value

193.140.248.127 is generated for center value

193.140.248.254 is generated for top value

In case of "*.*.*.*" is given, the algorithm uses 6 ip addresses to replace it by the approach of all ip addresses are identical but each selected ip aims to uncover the mostly predicted ip address range. That ip addresses are;

- 10.0.01
- 172.16.0.1
- 192.168.0.1
- 127.0.0.1
- 88.241.34.41
- 215.15.168.23

The port number is stated as an integer or "any". In case of integer value, it is used as it is given. But in "any" case, the port addresses are "0, 65535, 23, 80".

## 3.7. Network Traffic Generator

The graphical notation of testing platform is shown in Figure 3.4 with use case diagram. The detailed describtion of use case :

**Use-case :** Firewall Testing

**Primary actor :** Tester

**Secondary actors :** Firewall, Sniffer and Evaluator

**Goal in context :** Testing a firewall by generating network traffic via software testing techniques

**Preconditions :** 1) The test case generation methot needs to be selected

2) Firewall policy needs to be defined and set to DENY as default policy

3) Sniffer needs to be ready to listen traffic

**Trigger :** The tester decides to test a firewall

**Scenario :** 1) Tester: selects a testing technique

2) Tester: test is performed

3) Firewall: Filters incoming packets and route them

4) Sniffer: Captures packets behind firewall and store them

5) Evaluator: Compares the packets of Tester and Sniffer

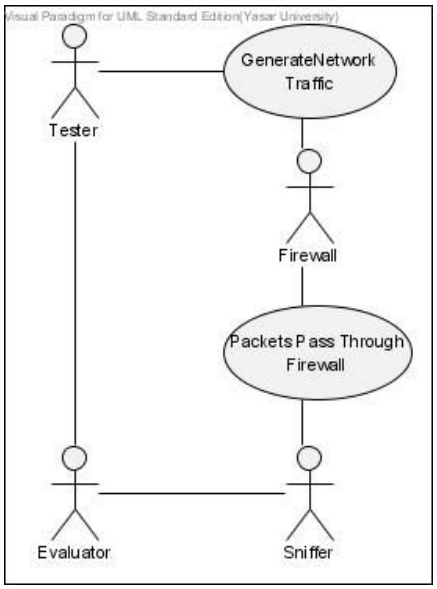



Figure 3.4. Use case diagram of FUT

The implemented Network Traffic Generator tool GUI is shown in Figure 3.5. Functions of the tool are enumerated in circles namely;

1) Perform Test : It confers 2 testing techniques to perform such as, conformance testing and equivalence partitioning testing.
2) OK Button : When pressed, it runs whole selected testing technique and includes phases such as reading a firewall rule, generating test cases, generating packets and finalizes test with sending packets.
3) Firewall Rules : That text area shows firewall rules.
4) List Rule Button : When pressed, firewall rules are read from file and listed in text field.
5) Test Case Generation : It confers 2 testing techniques to generate test cases
6) Show Equivalence DB : When pressed, that button establishes a database connection and sorts all stored packet information generated by equivalence partitioning testing technique.
7) Database Table : That table shows the queried data by Conformance DB and Equivalence DB buttons.
8) Console : That field notifies user about the information flow during software runs.

The Activity Diagram of implemented software is shown in Figure 3.6. When software run, testing technique needs to be selected. Due to selected testing technique, firewall rule policy is read from a file and convert that rule into a rule tuple. An appropiate algorithm, designed for conformance testing or equivalence partitioning testing, is applied for deriving test cases. Then packets are generated from each test case. Generated packets can be sent directly or written to text file or stored in database. Those 3 functions also can be combined by an order.

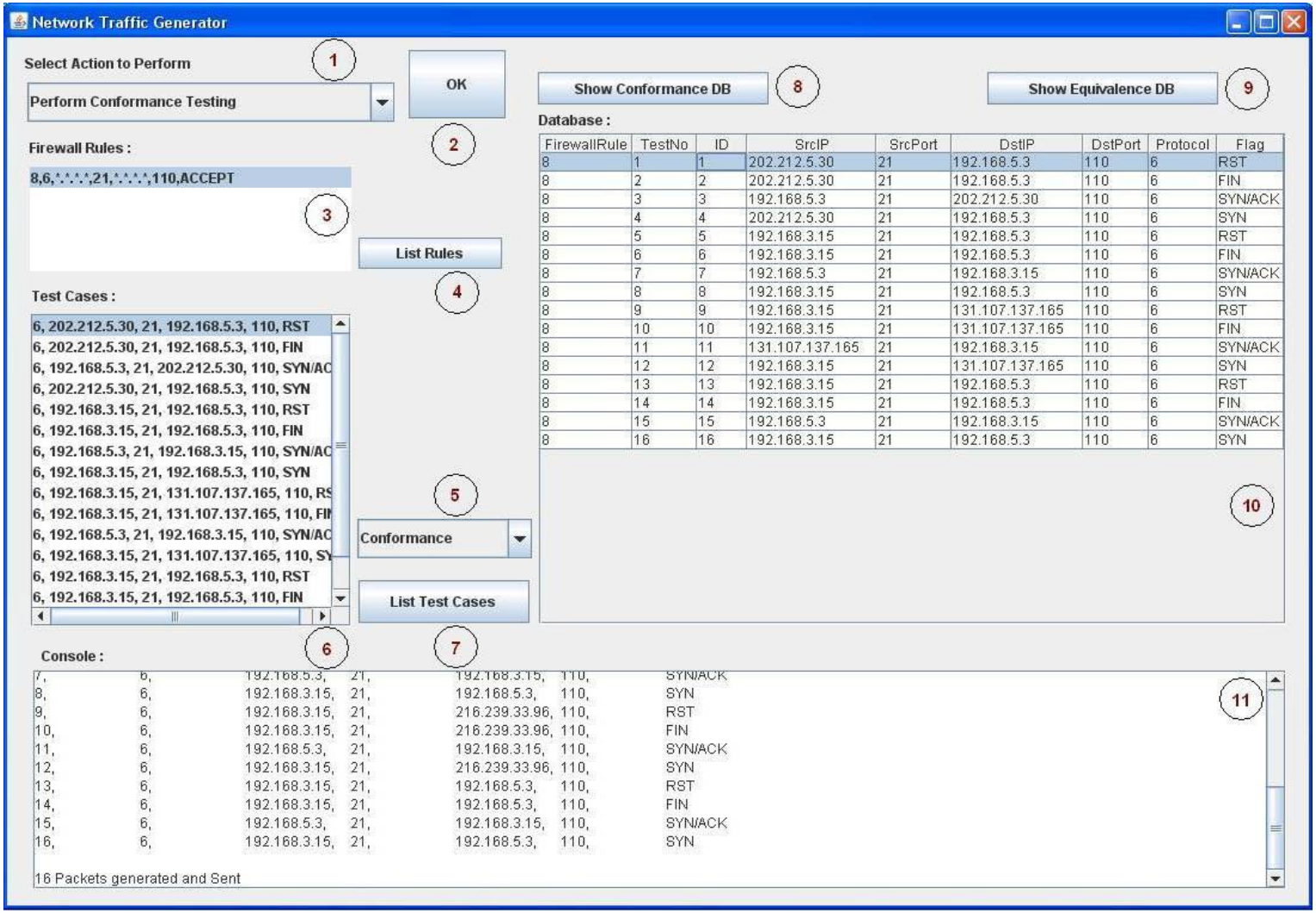



Figure 3.5. Network Traffic Generator Software GUI

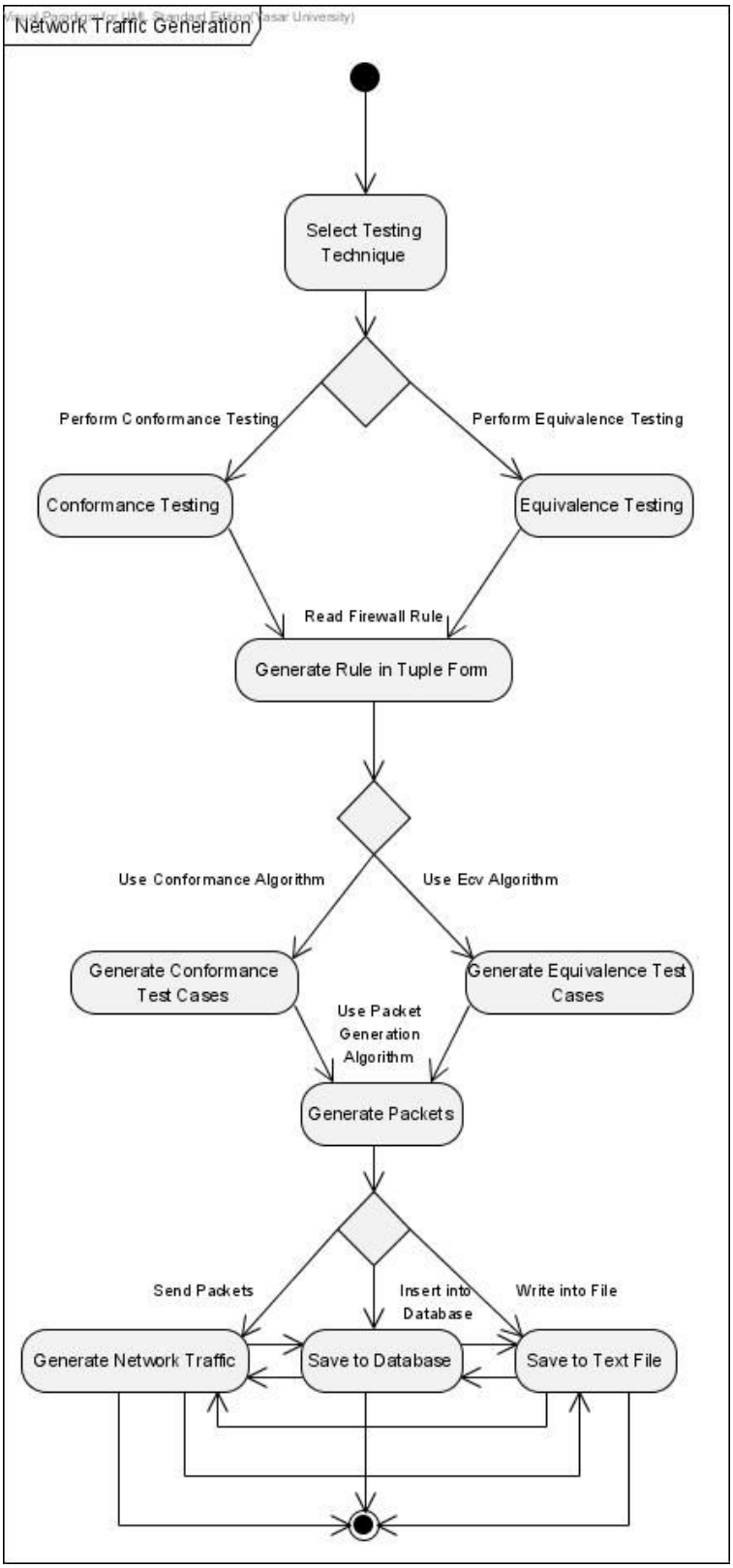



Figure 3.6. Activity Diagram

## 3.8. Database Design

MYSQL version 5.1 community server edition database is used for 3 purposes namely, storing generated packets, captured packets and comparing generated and captured packets as an evaluator. For database connection via java platform, mysql-connector-java version 5.1.7 driver is used. Generated packets are stored in a database table, created with sql for equivalence testing as seen in Figure 3.7, and followed by the database model in figure 3.8, for conformance testing the sql is shown in Figure 3.9 and followed by the database model in Figure 3.10.

```
CREATE TABLE `EquivalenceTesting` (
  `FirewallRule`        int(11) NOT NULL,
  `TestNo`              int(5) NOT NULL,
  `ID`                  int(11) NOT NULL,
  `SrcIP`               varchar(20) NOT NULL,
  `SrcPort`             int(5) NOT NULL,
  `DstIP`               varchar(20) NOT NULL,
  `DstPort`             int(5) NOT NULL,
  `Protocol`            int(2) NOT NULL );
```

Figure 3.7. Create Equivalence Testing Table

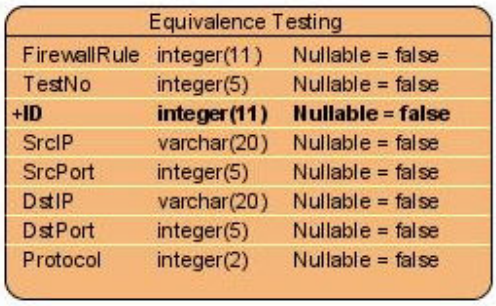



Figure 3. 8. Equivalence Testing Database Model

```
CREATE TABLE `ConformanceTesting` (
  `FirewallRule`    int(11) NOT NULL,
  `TestNo`          int(5) NOT NULL,
  `ID`              int(11) NOT NULL,
  `SrcIP`           varchar(20) NOT NULL,
  `SrcPort`         int(5) NOT NULL,
  `DstIP`           varchar(20) NOT NULL,
  `DstPort`         int(5) NOT NULL,
  `Protocol`        int(2) NOT NULL,
  `Flag`            varchar(2) NOT NULL );
```

Figure 3. 9. Create Conformance Testing Table

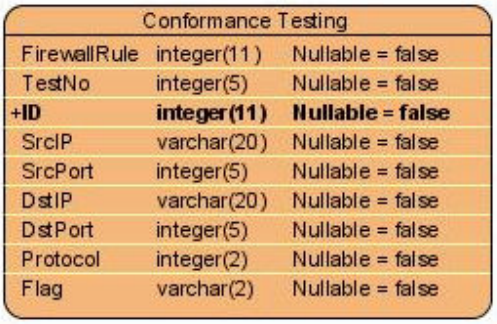



Figure 3. 10. Conformance Testing Database Model

The packets which pass through firewall, are captured with sniffer tool and stored in a table with only their hash values are considered .

## 3.9. Automatized Test Case Generation Algorithm

### 3.9.1. Conformance Testing

As the formal policy defined in tuple form, that study aims to generate test cases automated by the test case generation algorithm. The presented algorithm is modified from [60] as shown in 3.1, applied for TCP protocol by using Reset, FIN, SYN/ACK and SYN flags.

```
for each <FirewallRule>
        for <source_IP>
                Generate IP
        for <source_port>
                Generate Port
        for <destination_ip>
                Generate IP
        for <destination_port>
                Generate Port
Return TestCases

for each <TestCase>
        TestCase.setFlag(RST, true)
        TestCase.setFlag(FIN, true)
        switch(source_IP and destination_IP) and TestCase.setFlag(SYN,ACK, true)
        TestCase.setFlag(SYN, true)

for <IP>
        if ip is a host
                IP is a host IP
        if ip is a subnet
                IP is a random IP in subnet
        if ip is "*.*.*.*"
                IP is a random IP from set of most used IPs = {
                202.212.5.30,           // yahoo
                216.239.33.96,          // google
                131.107.137.165,        // msn }
for <Port>
        if port is an integer
                Port is port
        if port is "any"
                Generate random port in range [1-1024]
```

(3.1)

### 3.9.2. Equivalence Partitioning Testing

Our Equivalence Class Partitioning approach aims to uncover faults in FUT by exercising each independent path, where each path represents a unique test case. As referred in [15], input domain of FUT is divided into finite number of equivalence classes and ECP consists of 2 steps namely equivalence class identification and test case generation.

In [60] Equivalence classes are reclaimed as valid and invalid. Valid equivalence classes are denominated by ECv and represented by valid inputs. Invalid equivalence classes are denominated by ECinv and the domain contains of invalid inputs. The algorithm referred in [60] is modified and presented in 3.2.

```
for <ip>
        if ip is a host
                ECv is a host ip
                ECinv(0) is [1.0.0.1 … ip-1]
                ECinv(1) is [ip+1 … 255.255.255.254]
        if ip is a subnet
                call subnet_range for
                        ECv is subnet
                        ECinv(0) is lower subnet-1
                        ECinv(1) is upper subnet+1
        if ip is "*.*.*.*"
                ECv (0) is 10.0.01
                ECv (1) is 172.16.0.1
                ECv (2) is 192.168.0.1
                ECv (3) is 127.0.0.1
                ECv (4) is 88.241.34.41
                ECv (5) is 215.15.168.23
        for <subnet_range>
                ECv (0) is center value of subnet_range
                ECv (1) is bottom value of subnet_range
                ECv (2) is top value of subnet_range
for <port>
        if port is an integer
                ECv is port
                ECinv(0) is (port-1)
                ECinv(1) is (port+1)
        if port is "any"
                ECv (0) is 0
                ECv (1) is 65535
                ECv (2) is 23
                ECv (3) is 80
```

(3.2)

## 3.10. Abstract Test Cases

### 3.10.1. Scenario 1: Subnet IP Range for Source IP

The abstract test cases auto-generated by the given algorithm as the form of tuple which is source IP is a subnet; <6, 193.140.248.*, Any, *.*.*.*, 25>.

Those test cases -generated by Algorithm 1- are presented below in Table 3.4 at abstract form without flags. As stated in the algorithm those test cases aim to uncover errors in firewall caused by IP addresses and ports with the TCP protocol.

That abstract generated test cases, from no.1 to 3 are for the source IP, from no. 4 to 7 aims the source port, from no. 8 to 13 are for the destination IP and the no. 14 rule stands for the destination port.

Table 3. 4. Conformance Test Cases Scenario-1

| Test No | Protocol | Source IP | Source Port | Destination IP | Destination Port |
|---|---|---|---|---|---|
| 1 | 6 | 193.140.248.1 | Any | *.*.*.* | 25 |
| 2 | 6 | 193.140.248.127 | Any | *.*.*.* | 25 |
| 3 | 6 | 193.140.248.254 | Any | *.*.*.* | 25 |
| 4 | 6 | 193.140.248.* | 0 | *.*.*.* | 25 |
| 5 | 6 | 193.140.248.* | 65535 | *.*.*.* | 25 |
| 6 | 6 | 193.140.248.* | 23 | *.*.*.* | 25 |
| 7 | 6 | 193.140.248.* | 80 | *.*.*.* | 25 |
| 8 | 6 | 193.140.248.* | Any | 10.0.0.1 | 25 |
| 9 | 6 | 193.140.248.* | Any | 172.16.0.1 | 25 |
| 10 | 6 | 193.140.248.* | Any | 192.168.0.1 | 25 |
| 11 | 6 | 193.140.248.* | Any | 127.0.0.1 | 25 |
| 12 | 6 | 193.140.248.* | Any | 88.241.34.41 | 25 |
| 13 | 6 | 193.140.248.* | Any | 215.15.168.23 | 25 |
| 14 | 6 | 193.140.248.* | Any | *.*.*.* | 25 |

In the Table 3.5, the abstract test cases are shown, generated by Alg2. The test cases from no.1 to no. 9 are for source IP, no.10 to no. 13 are for source port, no.14 to no. 19 are for destination IP and no.20 to no.22 are for destination port.

Table 3. 5. Equivalence Partitioning Test Cases Scenario-1

| **Test No** | **Protocol** | **Source IP** | **Source Port** | **Destination IP** | **Destination Port** |
|---|---|---|---|---|---|
| 1 | 6 | 193.140.248.1 | Any | *.*.*.* | 25 |
| 2 | 6 | 193.140.248.127 | Any | *.*.*.* | 25 |
| 3 | 6 | 193.140.248.254 | Any | *.*.*.* | 25 |
| 4 | 6 | 193.140.247.1 | Any | *.*.*.* | 25 |
| 5 | 6 | 193.140.247.127 | Any | *.*.*.* | 25 |
| 6 | 6 | 193.140.247.254 | Any | *.*.*.* | 25 |
| 7 | 6 | 193.140.249.1 | Any | *.*.*.* | 25 |
| 8 | 6 | 193.140.249.127 | Any | *.*.*.* | 25 |
| 9 | 6 | 193.140.249.254 | Any | *.*.*.* | 25 |
| 10 | 6 | 193.140.248.* | 0 | *.*.*.* | 25 |
| 11 | 6 | 193.140.248.* | 65535 | *.*.*.* | 25 |
| 12 | 6 | 193.140.248.* | 23 | *.*.*.* | 25 |
| 13 | 6 | 193.140.248.* | 80 | *.*.*.* | 25 |
| 14 | 6 | 193.140.248.* | Any | 10.0.0.1 | 25 |
| 15 | 6 | 193.140.248.* | Any | 172.16.0.1 | 25 |
| 16 | 6 | 193.140.248.* | Any | 192.168.0.1 | 25 |
| 17 | 6 | 193.140.248.* | Any | 127.0.0.1 | 25 |
| 18 | 6 | 193.140.248.* | Any | 88.241.34.41 | 25 |
| 19 | 6 | 193.140.248.* | Any | 215.15.168.23 | 25 |
| 20 | 6 | 193.140.248.* | Any | *.*.*.* | 26 |
| 21 | 6 | 193.140.248.* | Any | *.*.*.* | 24 |
| 22 | 6 | 193.140.248.* | Any | *.*.*.* | 25 |

### 3.10.2. Scenario 2: Source IP is a Host

In this scenario, the abstract test cases auto-generated by the given algorithm as the form of tuple where the source ip represents a host;

- <6, 193.140.248.15, Any, *.*.*.*, 25>

The following test cases in Table 3.6 represent the case of source of host IP and the test case no.1 is for the source IP, from no.2 to no.5 are for the source port, from no.6 to no.11 are for the destination IP and no.12 is for the destination port. And the Table 3.7 represents the Equivalence Partitioning test cases.

Table 3. 6. Conformance Test Cases Scenario-2

| Test No | Protocol | Source IP | Source Port | Destination IP | Destination Port |
|---|---|---|---|---|---|
| 1 | 6 | 193.140.248.15 | Any | *.*.*.* | 25 |
| 2 | 6 | 193.140.248.15 | 0 | *.*.*.* | 25 |
| 3 | 6 | 193.140.248.15 | 65535 | *.*.*.* | 25 |
| 4 | 6 | 193.140.248.15 | 23 | *.*.*.* | 25 |
| 5 | 6 | 193.140.248.15 | 80 | *.*.*.* | 25 |
| 6 | 6 | 193.140.248.15 | Any | 10.0.0.1 | 25 |
| 7 | 6 | 193.140.248.15 | Any | 172.16.0.1 | 25 |
| 8 | 6 | 193.140.248.15 | Any | 192.168.0.1 | 25 |
| 9 | 6 | 193.140.248.15 | Any | 127.0.0.1 | 25 |
| 10 | 6 | 193.140.248.15 | Any | 88.241.34.41 | 25 |
| 11 | 6 | 193.140.248.15 | Any | 215.15.168.23 | 25 |
| 12 | 6 | 193.140.248.15 | Any | *.*.*.* | 25 |

Table 3. 7. Equivalence Partitioning Test Cases Scenario-2

| Test No | Protocol | Source IP | Source Port | Destination IP | Destination Port |
|---|---|---|---|---|---|
| 1 | 6 | 193.140.248.15 | Any | *.*.*.* | 25 |
| 2 | 6 | 1.0.0.1 | Any | *.*.*.* | 25 |
| 3 | 6 | 193.140.248.14 | Any | *.*.*.* | 25 |
| 4 | 6 | 97.70.124.7 | Any | *.*.*.* | 25 |
| 5 | 6 | 193.140.248.16 | Any | *.*.*.* | 25 |
| 6 | 6 | 255.255.255.254 | Any | *.*.*.* | 25 |
| 7 | 6 | 224.197.251.135 | Any | *.*.*.* | 25 |
| 8 | 6 | 193.140.248.15 | 0 | *.*.*.* | 25 |
| 9 | 6 | 193.140.248.15 | 65535 | *.*.*.* | 25 |
| 10 | 6 | 193.140.248.15 | 23 | *.*.*.* | 25 |
| 11 | 6 | 193.140.248.15 | 80 | *.*.*.* | 25 |
| 12 | 6 | 193.140.248.15 | Any | 10.0.0.1 | 25 |
| 13 | 6 | 193.140.248.15 | Any | 172.16.0.1 | 25 |
| 14 | 6 | 193.140.248.15 | Any | 192.168.0.1 | 25 |
| 15 | 6 | 193.140.248.15 | Any | 127.0.0.1 | 25 |
| 16 | 6 | 193.140.248.15 | Any | 88.241.34.41 | 25 |
| 17 | 6 | 193.140.248.15 | Any | 215.15.168.23 | 25 |
| 18 | 6 | 193.140.248.15 | Any | *.*.*.* | 26 |
| 19 | 6 | 193.140.248.15 | Any | *.*.*.* | 24 |
| 20 | 6 | 193.140.248.15 | Any | *.*.*.* | 25 |

### 3.10.3. Scenario 3: Source IP is Any

In scenario-3, the abstract test cases auto-generated by the given algorithm as the form of tuple in case of source IP is any;

- <6, Any, Any, *.*.*.*, 25>

The test cases listed below in Table 3.8. represents the case when source IP is denoted as any in the firewall rule. The generated test cases, from no1. to no.6 are for source IP, from no.7 to no.10 are for the source port, from no.11 to no.16 are for the destination IP, and the no.17 stand for the destination port to uncover errors.

Table 3. 8. Conformance Test Cases Scenario-3

| Test No | Protocol | Source IP | Source Port | Destination IP | Destination Port |
|---|---|---|---|---|---|
| 1 | 6 | 10.0.0.1 | Any | *.*.*.* | 25 |
| 2 | 6 | 172.16.0.1 | Any | *.*.*.* | 25 |
| 3 | 6 | 192.168.0.1 | Any | *.*.*.* | 25 |
| 4 | 6 | 127.0.0.1 | Any | *.*.*.* | 25 |
| 5 | 6 | 88.241.34.41 | Any | *.*.*.* | 25 |
| 6 | 6 | 215.15.168.23 | Any | *.*.*.* | 25 |
| 7 | 6 | *.*.*.* | 0 | *.*.*.* | 25 |
| 8 | 6 | *.*.*.* | 65535 | *.*.*.* | 25 |
| 9 | 6 | *.*.*.* | 23 | *.*.*.* | 25 |
| 10 | 6 | *.*.*.* | 80 | *.*.*.* | 25 |
| 11 | 6 | *.*.*.* | Any | 10.0.0.1 | 25 |
| 12 | 6 | *.*.*.* | Any | 172.16.0.1 | 25 |
| 13 | 6 | *.*.*.* | Any | 192.168.0.1 | 25 |
| 14 | 6 | *.*.*.* | Any | 127.0.0.1 | 25 |
| 15 | 6 | *.*.*.* | Any | 88.241.34.41 | 25 |
| 16 | 6 | *.*.*.* | Any | 215.15.168.23 | 25 |
| 17 | 6 | *.*.*.* | Any | *.*.*.* | 25 |

The generated test cases in Table 3.9, from no1. to no.6 are for source IP, from no.7 to no.10 are for the source port, from no.11 to no.16 are for the destination IP, and no.17 to no.19 stand for the destination port to uncover errors.

Table 3. 9. Equivalence Partitioning Test Cases Scenario-3

| **Test No** | **Protocol** | **Source IP** | **Source Port** | **Destination IP** | **Destination Port** |
|---|---|---|---|---|---|
| 1 | 6 | 10.0.0.1 | Any | *.*.*.* | 25 |
| 2 | 6 | 172.16.0.1 | Any | *.*.*.* | 25 |
| 3 | 6 | 192.168.0.1 | Any | *.*.*.* | 25 |
| 4 | 6 | 127.0.0.1 | Any | *.*.*.* | 25 |
| 5 | 6 | 88.241.34.41 | Any | *.*.*.* | 25 |
| 6 | 6 | 215.15.168.23 | Any | *.*.*.* | 25 |
| 7 | 6 | *.*.*.* | 0 | *.*.*.* | 25 |
| 8 | 6 | *.*.*.* | 65535 | *.*.*.* | 25 |
| 9 | 6 | *.*.*.* | 23 | *.*.*.* | 25 |
| 10 | 6 | *.*.*.* | 80 | *.*.*.* | 25 |
| 11 | 6 | *.*.*.* | Any | 10.0.0.1 | 25 |
| 12 | 6 | *.*.*.* | Any | 172.16.0.1 | 25 |
| 13 | 6 | *.*.*.* | Any | 192.168.0.1 | 25 |
| 14 | 6 | *.*.*.* | Any | 127.0.0.1 | 25 |
| 15 | 6 | *.*.*.* | Any | 88.241.34.41 | 25 |
| 16 | 6 | *.*.*.* | Any | 215.15.168.23 | 25 |
| 17 | 6 | *.*.*.* | Any | *.*.*.* | 26 |
| 18 | 6 | *.*.*.* | Any | *.*.*.* | 24 |
| 19 | 6 | *.*.*.* | Any | *.*.*.* | 25 |

# CHAPTER 4

# EXPERIMENTS AND EVALUATION

The implemented network traffic generation tool uses 2 different algorithms namely Conformance Testing Algorithm and Equivalence Partitioning Algorithm. These 2 algorithms are applied to same network topology as shown in Figure 4.1. Equivalence Partitioning Algorithm aims to reveal defacts in IPTABLES by using equivalence classes and expects all the packets exist in same partition act same. Other algorithm uses an approach which generats packets by changing flags to observe a stateless firewall acts same for each packet with different flag values.

Establishing a test bed, shown in Figure 4.1, begins with configuring IP address of network interfaces. Firewall's interface configuration is set up in linux by the commands "ifconfig eth0 192.168.3.1 netmask 255.255.255.0" and "ifconfig eth1 192.168.5.1 netmask 255.255.255.0".

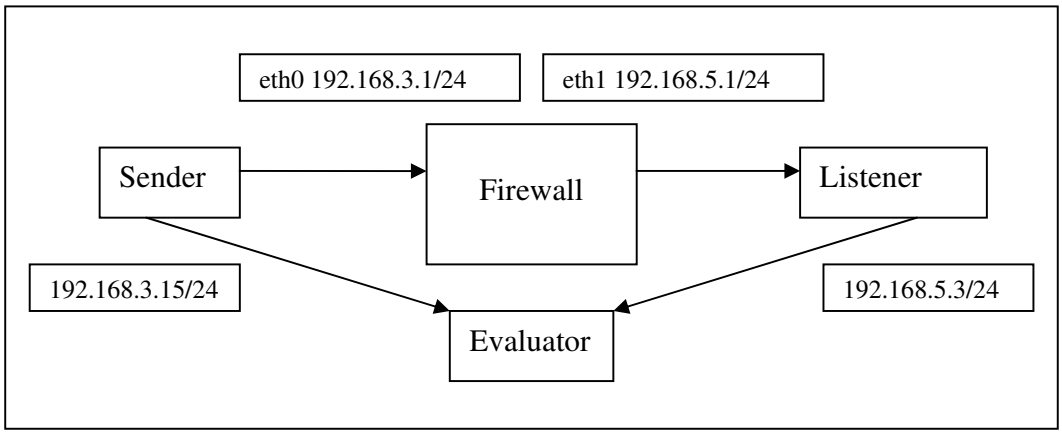



Figure 4. 1. Test Bed

Windows IP configuration is done in "Network Connections" window via TCP/IP addresses tab as seen in the Figure 4.2 for Sender PC and Listener PC .

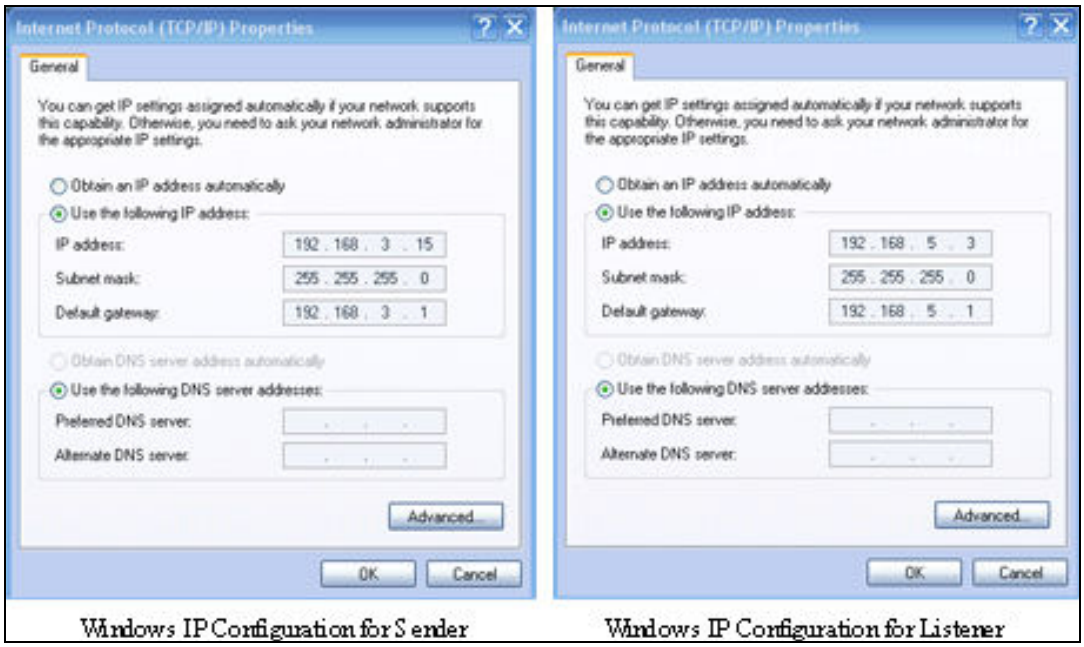



Figure 4. 2. Windows IP Configuration

The next step is to configure kernel's IP routing table in linux for routing packets come from eth1 to eth0 interfaces. The new routes are added to kernel by the commands, "add route –net 192.168.3.0 netmask 255.255.255.0 gw 192.168.3.1" and "add route –net 192.168.5.0 netmask 255.255.255.0 gw 192.168.5.1".

After configuration of all interfaces are completed, the firewall, IPTABLES, needs to be configured by defined the policy. That step requires different configuration due to the different policies for each experiment and will be explained detailed in each experiment.

In order to apply testing approaches, the firewall policy will be experimented based on source IP, source port, destination IP, destination port fields. For the packets aim to reveal defacts based on source or destination IP, in the policy host and subnet addresses need to be presented in source or destination IP field. The other defacts, may caused by port values, are considered as "any" or a port number, and need to be presented in policy as source or destination port fields.

## 4.1. Experimental Setup

In order to distinguish results and uncover all brances of the nominated policy, test cases are generated from each rule. The fields of the rules are taking into consideration and the conditions of IP can be Any, Subnet or Host, while port can be Any or Decimal as seen in Table 4.1.

Table 4. 1. Rule Fields

| Experiment No | Source IP | Source Port | Destination IP | Destination Port |
|---|---|---|---|---|
| 1 | Any | Any | Any | Any |
| 2 | Any | Any | Any | Port |
| 3 | Any | Any | Subnet | Any |
| 4 | Any | Any | Subnet | Port |
| 5 | Any | Any | Host | Any |
| 6 | Any | Any | Host | Port |
| 7 | Any | Port | Any | Any |
| 8 | Any | Port | Any | Port |
| 9 | Any | Port | Subnet | Any |
| 10 | Any | Port | Subnet | Port |
| 11 | Any | Port | Host | Any |
| 12 | Any | Port | Host | Port |
| 13 | Subnet | Any | Any | Any |
| 14 | Subnet | Any | Any | Port |
| 15 | Subnet | Any | Subnet | Any |
| 16 | Subnet | Any | Subnet | Port |
| 17 | Subnet | Any | Host | Any |
| 18 | Subnet | Any | Host | Port |
| 19 | Subnet | Port | Any | Any |
| 20 | Subnet | Port | Any | Port |
| 21 | Subnet | Port | Subnet | Any |
| 22 | Subnet | Port | Subnet | Port |
| 23 | Subnet | Port | Host | Any |
| 24 | Subnet | Port | Host | Port |
| 25 | Host | Any | Any | Any |
| 26 | Host | Any | Any | Port |
| 27 | Host | Any | Subnet | Any |
| 28 | Host | Any | Subnet | Port |
| 29 | Host | Any | Host | Any |
| 30 | Host | Any | Host | Port |
| 31 | Host | Port | Any | Any |
| 32 | Host | Port | Any | Port |
| 33 | Host | Port | Subnet | Any |
| 34 | Host | Port | Subnet | Port |
| 35 | Host | Port | Host | Any |
| 36 | Host | Port | Host | Port |

The sets $S_{ip}$, $S_{port}$, $D_{ip}$, $D_{port}$ represent Source IP, Source port, Destination IP and Destination port where $S_{ip}$={"*.*.*.*", "192.168.3.*", "192.168.3.15"}, $S_{port}$={"Any", "21"}, $D_{ip}$={"*.*.*.*", "192.168.5.*", "192.168.5.3"}and $D_{port}$={"Any", "110"}. By the cartesian product of these sets, there are 36 experiments planned for equivalence paritioning testing and conformance testing as seen in Table 4.2. The firewall rule tuple is set <"*.*.*.*","Any","*.*.*.*","Any"> and each field is represented by the defiend set elements until all the unknowns are revealed and tuple takes the form of <"192.168.3.15","21","192.168.5.3","110">. The firewall default policy is set to drop for each experiment with the command, "iptables –P FORWARD DROP".

Table 4. 2. Firewall Rules

| Experiment No | Source IP | Source Port | Destination IP | Destination Port | Protocol | Action |
|---|---|---|---|---|---|---|
| 1 | *.*.*.* | Any | *.*.*.* | Any | TCP | ACCEPT |
| 2 | *.*.*.* | Any | *.*.*.* | 110 | TCP | ACCEPT |
| 3 | *.*.*.* | Any | 192.168.5.* | Any | TCP | ACCEPT |
| 4 | *.*.*.* | Any | 192.168.5.* | 110 | TCP | ACCEPT |
| 5 | *.*.*.* | Any | 192.168.5.3 | Any | TCP | ACCEPT |
| 6 | *.*.*.* | Any | 192.168.5.3 | 110 | TCP | ACCEPT |
| 7 | *.*.*.* | 21 | *.*.*.* | Any | TCP | ACCEPT |
| 8 | *.*.*.* | 21 | *.*.*.* | 110 | TCP | ACCEPT |
| 9 | *.*.*.* | 21 | 192.168.5.* | Any | TCP | ACCEPT |
| 10 | *.*.*.* | 21 | 192.168.5.* | 110 | TCP | ACCEPT |
| 11 | *.*.*.* | 21 | 192.168.5.3 | Any | TCP | ACCEPT |
| 12 | *.*.*.* | 21 | 192.168.5.3 | 110 | TCP | ACCEPT |
| 13 | 192.168.3.* | Any | *.*.*.* | Any | TCP | ACCEPT |
| 14 | 192.168.3.* | Any | *.*.*.* | 110 | TCP | ACCEPT |
| 15 | 192.168.3.* | Any | 192.168.5.* | Any | TCP | ACCEPT |
| 16 | 192.168.3.* | Any | 192.168.5.* | 110 | TCP | ACCEPT |
| 17 | 192.168.3.* | Any | 192.168.5.3 | Any | TCP | ACCEPT |
| 18 | 192.168.3.* | Any | 192.168.5.3 | 110 | TCP | ACCEPT |
| 19 | 192.168.3.* | 21 | *.*.*.* | Any | TCP | ACCEPT |
| 20 | 192.168.3.* | 21 | *.*.*.* | 110 | TCP | ACCEPT |
| 21 | 192.168.3.* | 21 | 192.168.5.* | Any | TCP | ACCEPT |
| 22 | 192.168.3.* | 21 | 192.168.5.* | 110 | TCP | ACCEPT |
| 23 | 192.168.3.* | 21 | 192.168.5.3 | Any | TCP | ACCEPT |
| 24 | 192.168.3.* | 21 | 192.168.5.3 | 110 | TCP | ACCEPT |
| 25 | 192.168.3.15 | Any | *.*.*.* | Any | TCP | ACCEPT |
| 26 | 192.168.3.15 | Any | *.*.*.* | 110 | TCP | ACCEPT |
| 27 | 192.168.3.15 | Any | 192.168.5.* | Any | TCP | ACCEPT |
| 28 | 192.168.3.15 | Any | 192.168.5.* | 110 | TCP | ACCEPT |
| 29 | 192.168.3.15 | Any | 192.168.5.3 | Any | TCP | ACCEPT |
| 30 | 192.168.3.15 | Any | 192.168.5.3 | 110 | TCP | ACCEPT |
| 31 | 192.168.3.15 | 21 | *.*.*.* | Any | TCP | ACCEPT |
| 32 | 192.168.3.15 | 21 | *.*.*.* | 110 | TCP | ACCEPT |
| 33 | 192.168.3.15 | 21 | 192.168.5.* | Any | TCP | ACCEPT |
| 34 | 192.168.3.15 | 21 | 192.168.5.* | 110 | TCP | ACCEPT |
| 35 | 192.168.3.15 | 21 | 192.168.5.3 | Any | TCP | ACCEPT |
| 36 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | TCP | ACCEPT |

## 4.2. Experiment and Evaluation

The number of generated test cases are represented by a histogram graph in Figure 4.3 for 36 firewall rules, considering both algorithms, namely conformance testing and equivalence partitioning testing. As seen in the Figure 4.3, conformance testing algorithm generated 16 test cases for each rules. But the range of test cases, that are generated by equivalence partitioning algorithm, varies between 18 and 26, and the average number of test cases generated for each rule is approximately 22. Even the least number of generated test cases by equivalence partitioning algorithm for a firewall rule is 18 while conformance testing generates 16 test cases. Therefore, conformance testing algorithm is more appropriate while considering number of generated test cases. These generated test cases are experimented during FUT process and the experiment results take place in the Appendix A.

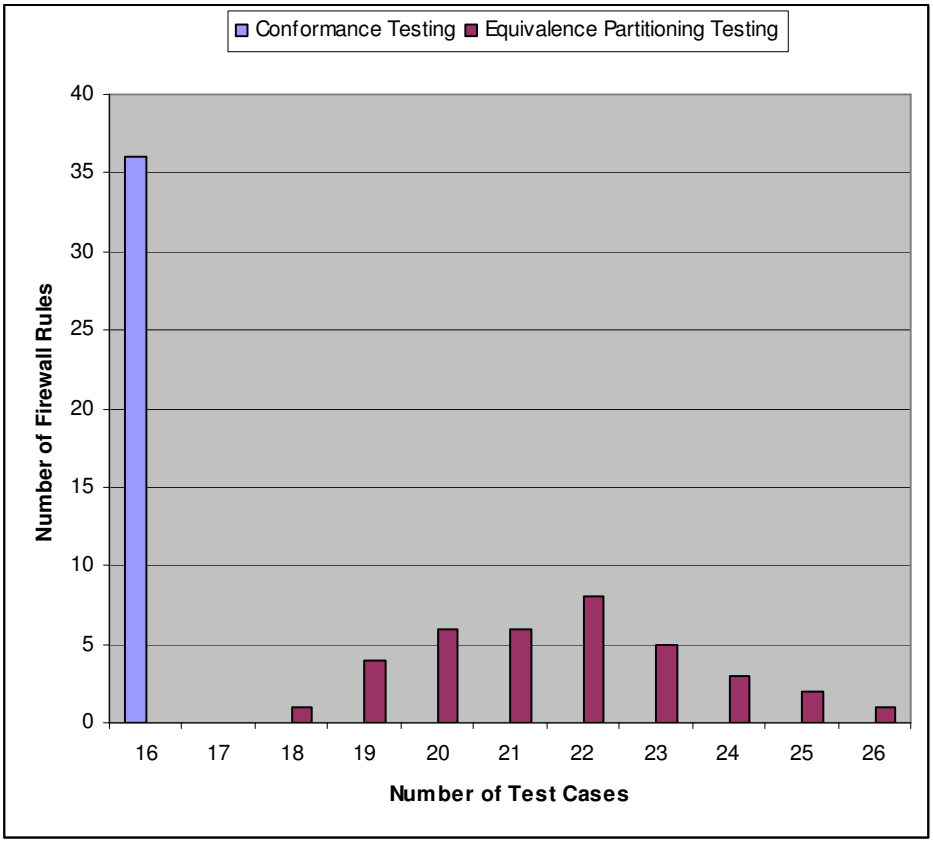



Figure 4. 3. Number of Test Case

# CHAPTER 5

# CONCLUSION

The main objective of this thesis is to develop an approach that provides an algorithm to cooperate with software testing methods for firewall testing. Therefore chosen software testing methods oriented to our proposed FUT approach. Designed approach is performed via implemented network traffic generator tool which enables to generate test cases from policy rules, than each test case is represented with a packet to test FUT. The most important feature of this thesis is that test cases are auto-generated. Implemented two different algorithms, which are conformance testing and equivalence partitioning, aim to uncover defects caused by firewall implementation or the nominated policy. These algorithms are used in developed packet generation framework. During framework design, library support is provided to assist packet generation. The test suite is based on the sent / filtered packets paradigm, in which test cases are created, sent through to firewall that has a implemented policy with certain iptables rules and than the comparison of generated and filtered packets are observed to see if the packets are filtered correctly.

Each test case of equivalence partitioning testing technique is represented by a packet. According the circumstances of test case is valid which is appropriate for the nominated policy, it is expected to packet is forwarded by firewall and the invalid packets are dropped. Conformance testing aims to generate all valid packets with unordered flag sequence considering to tcp handshake protocol and it is expected to packets are forwarded by firewall. During this study, all the test cases are implemented as it's intended by the security policy. It is observed while all invalid packets are dropped by the firewall , the valid packets are forwarded.

Clearly, concept of firewall policy rules can be extended by containing other protocols such as icmp, arp and udp. The appropriate algorithms can be developed with considered protocols. Also correctness test suites can be implemented for generating packets from both direction of firewall and the sent / expect pairs can be compared.

# REFERENCES


[1] M. Frantzen et al. "A Framework for Understanding Vulnerabilities in Firewalls Using a Dataflow Model of Firewall Internals," In *Proceedings of Computers Security*,2001, pp. 263-270.

[2] S. Rapps, E.J. Weyuker, "Data Flow Analysis Techniques for Test Data Selection", In *Roc. Sixth Int. Conf. Software Engineering*, Tokyo, 1982.

[3] S. Rapps, E.J Weyuker, "Selecting Software Test Data Using Data Flow Information," In *IEEE Trans. Softw. Eng*., SE-11, April 1985.

[4] M. Hutchins et al., "Experiments on the Effectiveness of Dataflow- and Controlflow-Based Test Adequacy Criteria", In *Siemens Corporate Research, Inc*.

[5] L. A. Clarke et al., "A comparison of data flow path selection criteria," In *Proceedings of 8th International Conference on Software Engineering*, August 1985, pp. 244-251, London, UK.

[6] P. G. Frankl and E. J. Weyuker, "An applicable family of data flow testing criteria," In *IEEE Transactions on Software Engineering*, vol. 14, no. 10, pp. 1483-1498, October 1988.

[7] B. Korel and J. Laski, "A tool for data flow oriented program testing," In *ACM Software Proceedings*, pp. 35-37, December 1985.

[8] B. Korel and J. Laski, "A data flow oriented program testing strategy," In *IEEE Transactions on Software Engineering*, vol. SE-9, no. 3, pp. 347- 354, May 1983.

[9] S. C. Ntafos, "An evaluation of required element testing strategies," In *Proceedings of 7th International Conference on Sofhvare Engineering*, March 1984, pp. 250-256, Orlando, Florida.

[10] S. Rapps, E. J. Weyuker, "Selecting software test data using data flow information," In *IEEE Transactionsof Software Engineering*, vol. SE- 11, no. 4, April 1985, pp. 367-375.

[11] M. J. Harrold, M. L. Soffa, "Interprocedural Data Flow Testing," Department of Computer Science University of Pittsburgh.

[12] G. Zaugg. "Firewall Testing." M.A. Thesis, Swiss Federal Institute of Technology Zurich, 2005.

[13] R. S. Pressman, *Software Engineering: A Practitioner's Approach*, European 3th ed., UK, 1994.

[14] A. Heitzel and C. William, *The Complete Guide to Software Testing*, 2nd ed., QED Information Sciences, 1988.

[15] G. J. Myers, *The Art of Software Testing*, 2nd Ed, 2004.

[16] M. Pezzand and M. Young, *Software Testing and Analysis: Process, Principles and Techniques*, John Wiley & Sons Inc, 2008.

[17] K. Meinke, "Automated Black-Box Testing of Functional Correctness using Function Approximation", In *ACM SIGSOFT Software Engineering Notes*, 2004.

[18] G. J. Myers, "Art of Software Testing", John Wiley & Sons, Inc., New York 1979.

[19] W. E. Perry, *Effective Methods for Software Testing*, 3rd Ed., 2006.

[20] R. S. Pressman, *Software Engineering A Practitioner's Approach*, 5th Ed., 2001.

[21] B. Beizer, "Black-Box Testing: Techniques for Functional Testing of Software and Systems," John Wiley & Sons, Inc., 1995.

[22] B. K. Aichernig, "Systematic Black-Box Testing of Computer-Based Systems through Formal Abstraction Techniques", Ph. D. Thesis 2001.

[23] I. Burnstein, "Practical Software Testing: A Process Oriented Approach," ISBN 0-387-95131-8, 2003.

[24] J. Duran and S. Ntafos, "An Evaluation of Random Testing," In *IEEE Trans. on Software Engineering*, Vol. SE-10, No. 4, July 1984, pp. 438.

[25] R. Hamlet and R. Taylor, "Partition Testing does not Inspire Confidence," In *IEEE Trans. on Software Engineering*, Vol. 16, No. 12, Dec. 1990, pp. 1402-1411.

[26] T. Chen and Y. Yu, "On the expected number of failures detected by sub domain testing and random testing," In *IEEE Trans. Software Engineering*, Vol. 22, 1996, pp. 109–119.

[27] W. Gutjahr, "Partition testing vs. random testing: the influence of uncertainty," In *IEEE Trans. Software Engineering*, Vol. 25, No. 5, Sept./Oct. 1999, pp. 661–674.

[28] P.G. Frankl and E.J. Weyuker, "A Formal Analysis of the Fault-Detecting Ability of Testing Methods," In *IEEE Trans. Software Eng*., vol. 19, pp. 202–213, 1993.

[29] D. Richardson and L. Clarke, "A partition analysis method to increase program reliability", in *Proc. 5th Int. Conf Software Engineering*, San Diego, CA, 1981, pp. 244-253.

[30] E.J. Weyuker and B. Jeng, "Analyzing Partition Testing Strategies," In *IEEE Trans. Software Eng.*, vol. 17, pp. 703–711, 1991.

[31] "Conformance Testing." Internet: http://en.wikipedia.org/wiki/Conformance_testing, May, 2009.

[32] "Conformance Testing Standards." Internet: http://searchsoftwarequality.techtarget.com/sDefinition/0,,sid92_gci1243147,00.html, May, 2009.

[33] R. J. Linn and W. H. McCoy, "Producing tests for implementations of OS1 protocols," in *Protocol Specifications, Testing and Verification*. III. H. Rudin and C. H. West. Eds. Amsterdam. The Netherlands: North-Holland. 1983. pp. 505-520.

[34] D. P. Sidhu, "Protocol verification via executable logic specifications," In *Protocol Specifications, Testing and Verification*. III. H. Rudin and C. H. West. Eds. Amsterdam. The Netherlands: North-Holland. 1983. pp. 237-248.

[35] D. P. Sidhu and C. S. Crall, "Executable logic specifications for protocol service interfaces," In *IEEE Trans. Software Eng*.. vol. 14. Jan. 1988.

[36] D. Rayner, "Standardizing conformance testing for OSI. in Protocol Specifications," In *Proc of Testing and Verification*, V. M. Diaz. Ed. Amsterdam. The Netherlands: North-Holland. 1986.

[37] D. P. Sidhu. "Protocol verification using prolog". ISU Preprint. 1985.

[38] "Conformance Testing." Internet: http://portal.etsi.org/mbs/Testing/conformance/conformance.htm#TP, May, 2009.

[39] "Penetration Testing." Internet: http://en.wikipedia.org/wiki/Penetration_testing , April, 2009.

[40] G. McGraw, "Software Security," *In Proc. of IEEE Security & Privacy*, vol. 2, no.2, 2004, pp. 80–83.

[41] D. Verndon and G. McGraw, "Software Risk Analysis," In *Proc. of IEEE Security & Privacy,* vol. 2, no. 5, 2004, pp. 81–85.

[42] "Firewalls." Internet: http://tr.wikipedia.org/wiki/Firewall,May, 2009.

[43] S. M. Bellovin and R. W. Cheswick, "Network Firewalls," In *IEEE Communications Magazine*, p: 50-57, 1994.

[44] E. S. Al-Shaer and H. Hamed, "Modeling and Management of Firewall Policies," In *IEEE Transactions On Network and Service Management*, 2004.

[45] J. Wack, K. Cutler, J. Pole, "Guidelines on Firewalls and Firewall Policy", 2004.

[46] “Comodo Firewall Program User Guide”, Internet: http://forums.comodo.com/computer_firewalls-b67.0, April, 2009.
[47] D. B. Chapman,” Network (in)security through IP packet filtering,” In *Proceedings of the Third Usenix UNlX Security Symposium*, pp. 63-76, 1992.

[48] “Firewall Policy.” Internet: http://www.net-security.org/article.php?id=440&p=1, April, 2009.

[49] J. Tretmans, *A Formal Approach to Conformance Testing*, ISBN 90–9005643–2, 2001.

[50] R. M. Hierons and H. Ural, “Concerning the Ordering of Adaptive Test Sequences,” In *23rd IFIP International Conference on Formal Techniques for Networked and Distributed Systems (FORTE 2003),* 2003, LNCS volume 2767, pp. 289-302.

[51] I. Hwang et al., “Test Selection for a Nondeterministic FSM,” In *Proc of Computer Communications*, 2001, Vol. 24/12, 7, pp.1213-1223.

[52] F. Zhang and T. Cheung, “Optimal Transfer Trees and Distinguishing Trees for Testing Observable Nondeterministic Finite-State Machines,” In *IEEE Transactions on Software Engineering*, 2003, Vol. 29, No. 1, pp. 1-14.

[53] D. P. Sidhu and T. Leung, “Formal Methods for Protocol Testing: A Detailed Study,” In *IEEE Transactions On Software Engineering*, Vol. 15, No.4, 1989.

[54] I. Kufareva, N. Yevtushenko, and A. Petrenko, “Design of Tests for Nondeterministic Machines with Respect to Reduction,” *Automatic Control and Computer Sciences*, Allerton Press Inc., USA, No. 3, 1998.

[55] G. L. Luo, G. v. Bochmann, and A. Petrenko, “Test Selection Based on Communicating Nondeterministic Finite-State Machines Using a Generalized Wp-method.,” In *IEEE Transactions on Software Engineering,* 1994, 20(2), pp. 149–161.

[56] A. Petrenko et al., “Nondeterministic State Machines in Protocol Conformance Testing,” In *Proceedings of the IFIP Sixth International Workshop on Protocol Test Systems*, France, 1993, pp. 363-378.

[57] A. Petrenko, N. Yevtushenko, and G. v. Bochmann, “Testing Deterministic Implementations from their Nondeterministic Specifications,” In *Proceedings of the IFIP Ninth International Workshop on Testing of Communicating Systems*, 1996, pp. 125-140.

[58] N. Yevtushenko, A. Lebedev, and A. Petrenko, “On Checking Experiments with Nondeterministic Automata,” In *Automatic Control and Computer Sciences*, 1991, 6, pp. 81–85.

[59] A. Petrenko, and Nina Yevtushenko, "Conformance Tests as Checking Experiments for Partial Nondeterministic FSM," In *Springer Proceedings FATES,* pp. 118-133, 2005.

[60] T. Tuglular, "Test Case Generation for Firewall Implementation Testing using Software Testing Techniques," In *SIN 2007- Internation Conference on Security of Information and Networks*, 2007.

[61] Recommendation X.500, *-The Directory - Overview of Concepts, Models, and Services*, 1989.

[62] CCITT Z.100, *Specification and Description Language SDL*, 1988.

[63] "IPTABLES in Netfilter." Internet: http://www.netfilter.org, May, 2009

[64] Oskar Andreasson, IPTABLES Tutorial v1.2.0.

[65] Danial Hoffman and Durga Prabhakar, "Testing IPTABLES," M.S. Thesis, Departmant of Computer Science, University of Victoria.

[66] William Stallings, *Data and Computer Communications*,5th Edition, 2003.

[67] "Conformance Testing." Internet: http://portal.etsi.org/mbs/Testing/conformance/conformance.htm, April, 2009.

[68] "Firewall Types." Internet: http://www.vicomsoft.com/knowledge/reference/firewalls1.html#2, May, 2009.

[69] "Firewall Types." Internet: http://www.c-sharpcorner.com/UploadFile/pmalik/firewall12212008134435PM/firewall.aspx, May, 2009.

[70] "TCP/IP Model." Internet: http://learn-networking.com/tcp-ip/how-encapsulation-works-within-the-tcpip-model, May, 2009.

[71] "JPCAP Library." Internet: http://netresearch.ics.uci.edu/kfujii/jpcap/doc/javadoc/index.html, May, 2009.

# APPENDIX A

# EXPERIMENT RESULTS

## A.1. Experiment I

The first experiment is performed for the firewall policy that allows all traffic from tcp protocol. The firewall configuration by the policy is done with the command, “iptables –A FORWARD –p tcp –j ACCEPT”.

Table A.1. Equivalence Partitioning-I

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 10.0.0.1 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 1 | 2 | 2 | 172.16.0.1 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 1 | 3 | 3 | 192.168.0.1 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 1 | 4 | 4 | 127.0.0.1 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 1 | 5 | 5 | 88.241.34.41 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 1 | 6 | 6 | 215.15.168.23 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 1 | 7 | 7 | 192.168.3.15 | 0 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 1 | 8 | 8 | 192.168.3.15 | 65535 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 1 | 9 | 9 | 192.168.3.15 | 23 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 1 | 10 | 10 | 192.168.3.15 | 80 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 1 | 11 | 11 | 192.168.3.15 | 21 | 10.0.0.1 | 110 | TCP | ALLOW | PASSED |
| 1 | 12 | 12 | 192.168.3.15 | 21 | 172.16.0.1 | 110 | TCP | ALLOW | PASSED |
| 1 | 13 | 13 | 192.168.3.15 | 21 | 192.168.0.1 | 110 | TCP | ALLOW | PASSED |
| 1 | 14 | 14 | 192.168.3.15 | 21 | 127.0.0.1 | 110 | TCP | ALLOW | PASSED |
| 1 | 15 | 15 | 192.168.3.15 | 21 | 88.241.34.41 | 110 | TCP | ALLOW | PASSED |
| 1 | 16 | 16 | 192.168.3.15 | 21 | 215.15.168.23 | 110 | TCP | ALLOW | PASSED |
| 1 | 17 | 17 | 192.168.3.15 | 21 | 192.168.5.3 | 0 | TCP | ALLOW | PASSED |
| 1 | 18 | 18 | 192.168.3.15 | 21 | 192.168.5.3 | 65535 | TCP | ALLOW | PASSED |
| 1 | 19 | 19 | 192.168.3.15 | 21 | 192.168.5.3 | 23 | TCP | ALLOW | PASSED |
| 1 | 20 | 20 | 192.168.3.15 | 21 | 192.168.5.3 | 80 | TCP | ALLOW | PASSED |

Table A.2. Conformance Testing - I

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Flag | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 131.107.137.165 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 1 | 2 | 2 | 131.107.137.165 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 1 | 3 | 3 | 192.168.5.3 | 21 | 131.107.137.165 | 110 | SYN/ACK | TCP | ALLOW | PASSED |
| 1 | 4 | 4 | 131.107.137.165 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 1 | 5 | 5 | 192.168.3.15 | 26 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 1 | 6 | 6 | 192.168.3.15 | 26 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 1 | 7 | 7 | 192.168.5.3 | 26 | 192.168.3.15 | 110 | SYN/ACK | TCP | ALLOW | PASSED |
| 1 | 8 | 8 | 192.168.3.15 | 26 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 1 | 9 | 9 | 192.168.3.15 | 21 | 202.212.5.30 | 110 | RST | TCP | ALLOW | PASSED |
| 1 | 10 | 10 | 192.168.3.15 | 21 | 202.212.5.30 | 110 | FIN | TCP | ALLOW | PASSED |
| 1 | 11 | 11 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | ALLOW | PASSED |
| 1 | 12 | 12 | 192.168.3.15 | 21 | 202.212.5.30 | 110 | SYN | TCP | ALLOW | PASSED |
| 1 | 13 | 13 | 192.168.3.15 | 21 | 192.168.5.3 | 26 | RST | TCP | ALLOW | PASSED |
| 1 | 14 | 14 | 192.168.3.15 | 21 | 192.168.5.3 | 26 | FIN | TCP | ALLOW | PASSED |
| 1 | 15 | 15 | 192.168.5.3 | 21 | 192.168.3.15 | 26 | SYN/ACK | TCP | ALLOW | PASSED |
| 1 | 16 | 16 | 192.168.3.15 | 21 | 192.168.5.3 | 26 | SYN | TCP | ALLOW | PASSED |

## A.2. Experiment II

The second experiment is run for the policy that allows all traffic targets the destination port 110 under the condition of protocol is tcp. The firewall policy is configured as "iptables –A FORWARD –p tcp –dport 110 –j ACCEPT".

Table A.3. Equivalence Partitioning-II

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|
| 2 | 1 | 1 | 10.0.0.1 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 2 | 2 | 2 | 172.16.0.1 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 2 | 3 | 3 | 192.168.0.1 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 2 | 4 | 4 | 127.0.0.1 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 2 | 5 | 5 | 88.241.34.41 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 2 | 6 | 6 | 215.15.168.23 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 2 | 7 | 7 | 192.168.3.15 | 0 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 2 | 8 | 8 | 192.168.3.15 | 65535 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 2 | 9 | 9 | 192.168.3.15 | 23 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 2 | 10 | 10 | 192.168.3.15 | 80 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 2 | 11 | 11 | 192.168.3.15 | 21 | 10.0.0.1 | 110 | TCP | ALLOW | PASSED |
| 2 | 12 | 12 | 192.168.3.15 | 21 | 172.16.0.1 | 110 | TCP | ALLOW | PASSED |
| 2 | 13 | 13 | 192.168.3.15 | 21 | 192.168.0.1 | 110 | TCP | ALLOW | PASSED |
| 2 | 14 | 14 | 192.168.3.15 | 21 | 127.0.0.1 | 110 | TCP | ALLOW | PASSED |
| 2 | 15 | 15 | 192.168.3.15 | 21 | 88.241.34.41 | 110 | TCP | ALLOW | PASSED |
| 2 | 16 | 16 | 192.168.3.15 | 21 | 215.15.168.23 | 110 | TCP | ALLOW | PASSED |
| 2 | 17 | 17 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 2 | 18 | 18 | 192.168.3.15 | 21 | 192.168.5.3 | 109 | TCP | DENY | PASSED |
| 2 | 19 | 19 | 192.168.3.15 | 21 | 192.168.5.3 | 111 | TCP | DENY | PASSED |

Table A.4. Conformance Testing - II

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Flag | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 1 | 1 | 131.107.137.165 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 2 | 2 | 2 | 131.107.137.165 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 2 | 3 | 3 | 192.168.5.3 | 21 | 131.107.137.165 | 110 | SYN/ACK | TCP | ALLOW | PASSED |
| 2 | 4 | 4 | 131.107.137.165 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 2 | 5 | 5 | 192.168.3.15 | 28 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 2 | 6 | 6 | 192.168.3.15 | 28 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 2 | 7 | 7 | 192.168.5.3 | 28 | 192.168.3.15 | 110 | SYN/ACK | TCP | ALLOW | PASSED |
| 2 | 8 | 8 | 192.168.3.15 | 28 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 2 | 9 | 9 | 192.168.3.15 | 21 | 131.107.137.165 | 110 | RST | TCP | ALLOW | PASSED |
| 2 | 10 | 10 | 192.168.3.15 | 21 | 131.107.137.165 | 110 | FIN | TCP | ALLOW | PASSED |
| 2 | 11 | 11 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | ALLOW | PASSED |
| 2 | 12 | 12 | 192.168.3.15 | 21 | 131.107.137.165 | 110 | SYN | TCP | ALLOW | PASSED |
| 2 | 13 | 13 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 2 | 14 | 14 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 2 | 15 | 15 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | ALLOW | PASSED |
| 2 | 16 | 16 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |

## A.3. Experiment III

The third experiment is performed for the case, firewall is configured as the destination ip is a subnet of 192.168.3.* with the command of "iptables –A FORWARD –p tcp –d 192.168.3.0/24 –j ACCEPT".

Table A.5. Equivalence Partitioning-III

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|
| 3 | 1 | 1 | 10.0.0.1 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 3 | 2 | 2 | 172.16.0.1 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 3 | 3 | 3 | 192.168.0.1 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 3 | 4 | 4 | 127.0.0.1 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 3 | 5 | 5 | 88.241.34.41 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 3 | 6 | 6 | 215.15.168.23 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 3 | 7 | 7 | 192.168.3.15 | 0 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 3 | 8 | 8 | 192.168.3.15 | 65535 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 3 | 9 | 9 | 192.168.3.15 | 23 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 3 | 10 | 10 | 192.168.3.15 | 80 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 3 | 11 | 11 | 192.168.3.15 | 21 | 192.168.5.1 | 110 | TCP | ALLOW | PASSED |
| 3 | 12 | 12 | 192.168.3.15 | 21 | 192.168.5.127 | 110 | TCP | ALLOW | PASSED |
| 3 | 13 | 13 | 192.168.3.15 | 21 | 192.168.5.254 | 110 | TCP | ALLOW | PASSED |
| 3 | 14 | 14 | 192.168.3.15 | 21 | 192.168.4.1 | 110 | TCP | DENY | PASSED |
| 3 | 15 | 15 | 192.168.3.15 | 21 | 192.168.4.127 | 110 | TCP | DENY | PASSED |
| 3 | 16 | 16 | 192.168.3.15 | 21 | 192.168.4.254 | 110 | TCP | DENY | PASSED |
| 3 | 17 | 17 | 192.168.3.15 | 21 | 192.168.6.1 | 110 | TCP | DENY | PASSED |
| 3 | 18 | 18 | 192.168.3.15 | 21 | 192.168.6.127 | 110 | TCP | DENY | PASSED |
| 3 | 19 | 19 | 192.168.3.15 | 21 | 192.168.6.254 | 110 | TCP | DENY | PASSED |
| 3 | 20 | 20 | 192.168.3.15 | 21 | 192.168.5.3 | 0 | TCP | ALLOW | PASSED |
| 3 | 21 | 21 | 192.168.3.15 | 21 | 192.168.5.3 | 65535 | TCP | ALLOW | PASSED |
| 3 | 22 | 22 | 192.168.3.15 | 21 | 192.168.5.3 | 23 | TCP | ALLOW | PASSED |
| 3 | 23 | 23 | 192.168.3.15 | 21 | 192.168.5.3 | 80 | TCP | ALLOW | PASSED |

Table A.6. Conformance Testing - III

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Flag | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|---|
| 3 | 1 | 1 | 202.212.5.30 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 3 | 2 | 2 | 202.212.5.30 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 3 | 3 | 3 | 192.168.5.3 | 21 | 202.212.5.30 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 3 | 4 | 4 | 202.212.5.30 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 3 | 5 | 5 | 192.168.3.15 | 468 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 3 | 6 | 6 | 192.168.3.15 | 468 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 3 | 7 | 7 | 192.168.5.3 | 468 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 3 | 8 | 8 | 192.168.3.15 | 468 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 3 | 9 | 9 | 192.168.3.15 | 21 | 192.168.5.362 | 110 | RST | TCP | ALLOW | PASSED |
| 3 | 10 | 10 | 192.168.3.15 | 21 | 192.168.5.362 | 110 | FIN | TCP | ALLOW | PASSED |
| 3 | 11 | 11 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 3 | 12 | 12 | 192.168.3.15 | 21 | 192.168.5.362 | 110 | SYN | TCP | ALLOW | PASSED |
| 3 | 13 | 13 | 192.168.3.15 | 21 | 192.168.5.3 | 468 | RST | TCP | ALLOW | PASSED |
| 3 | 14 | 14 | 192.168.3.15 | 21 | 192.168.5.3 | 468 | FIN | TCP | ALLOW | PASSED |
| 3 | 15 | 15 | 192.168.5.3 | 21 | 192.168.3.15 | 468 | SYN/ACK | TCP | DENY | PASSED |
| 3 | 16 | 16 | 192.168.3.15 | 21 | 192.168.5.3 | 468 | SYN | TCP | ALLOW | PASSED |

## A.4. Experiment IV

The fourth experiment runs for the case of the traffic pass through firewall, targets a subnet with the port of 110. The firewall policy is edited as “iptables –A FORWARD –p tcp –s–d 192.168.3.0/24 –dport 110 –j ACCEPT”.

Table A.7. Equivalence Partitioning-IV

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|
| 4 | 1 | 1 | 10.0.0.1 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 4 | 2 | 2 | 172.16.0.1 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 4 | 3 | 3 | 192.168.0.1 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 4 | 4 | 4 | 127.0.0.1 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 4 | 5 | 5 | 88.241.34.41 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 4 | 6 | 6 | 215.15.168.23 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 4 | 7 | 7 | 192.168.3.15 | 0 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 4 | 8 | 8 | 192.168.3.15 | 65535 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 4 | 9 | 9 | 192.168.3.15 | 23 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 4 | 10 | 10 | 192.168.3.15 | 80 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 4 | 11 | 11 | 192.168.3.15 | 21 | 192.168.5.1 | 110 | TCP | ALLOW | PASSED |
| 4 | 12 | 12 | 192.168.3.15 | 21 | 192.168.5.127 | 110 | TCP | ALLOW | PASSED |
| 4 | 13 | 13 | 192.168.3.15 | 21 | 192.168.5.354 | 110 | TCP | ALLOW | PASSED |
| 4 | 14 | 14 | 192.168.3.15 | 21 | 192.168.4.1 | 110 | TCP | DENY | PASSED |
| 4 | 15 | 15 | 192.168.3.15 | 21 | 192.168.4.127 | 110 | TCP | DENY | PASSED |
| 4 | 16 | 16 | 192.168.3.15 | 21 | 192.168.4.254 | 110 | TCP | DENY | PASSED |
| 4 | 17 | 17 | 192.168.3.15 | 21 | 192.168.6.1 | 110 | TCP | DENY | PASSED |
| 4 | 18 | 18 | 192.168.3.15 | 21 | 192.168.6.127 | 110 | TCP | DENY | PASSED |
| 4 | 19 | 19 | 192.168.3.15 | 21 | 192.168.6.254 | 110 | TCP | DENY | PASSED |
| 4 | 20 | 20 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 4 | 21 | 21 | 192.168.3.15 | 21 | 192.168.5.3 | 109 | TCP | DENY | PASSED |
| 4 | 22 | 22 | 192.168.3.15 | 21 | 192.168.5.3 | 111 | TCP | DENY | PASSED |

Table A.8. Conformance Testing - IV

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Flag | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|---|
| 4 | 1 | 1 | 131.107.137.165 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 4 | 2 | 2 | 131.107.137.165 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 4 | 3 | 3 | 192.168.5.3 | 21 | 131.107.137.165 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 4 | 4 | 4 | 131.107.137.165 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 4 | 5 | 5 | 192.168.3.15 | 674 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 4 | 6 | 6 | 192.168.3.15 | 674 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 4 | 7 | 7 | 192.168.5.3 | 674 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 4 | 8 | 8 | 192.168.3.15 | 674 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 4 | 9 | 9 | 192.168.3.15 | 21 | 192.168.2.90 | 110 | RST | TCP | ALLOW | PASSED |
| 4 | 10 | 10 | 192.168.3.15 | 21 | 192.168.2.90 | 110 | FIN | TCP | ALLOW | PASSED |
| 4 | 11 | 11 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 4 | 12 | 12 | 192.168.3.15 | 21 | 192.168.2.90 | 110 | SYN | TCP | ALLOW | PASSED |
| 4 | 13 | 13 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 4 | 14 | 14 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 4 | 15 | 15 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 4 | 16 | 16 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |

## A.5. Experiment V

The fifth experiment is performed for the case that traffic is allowed for the destination ip is a host. The firewall policy is configured as "iptables –A FORWARD –p tcp –d 192.168.5.3 –j ACCEPT".

Table A.9. Equivalence Partitioning - V

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|
| 5 | 1 | 1 | 10.0.0.1 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 5 | 2 | 2 | 172.16.0.1 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 5 | 3 | 3 | 192.168.0.1 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 5 | 4 | 4 | 127.0.0.1 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 5 | 5 | 5 | 88.241.34.41 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 5 | 6 | 6 | 215.15.168.23 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 5 | 7 | 7 | 192.168.3.15 | 0 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 5 | 8 | 8 | 192.168.3.15 | 65535 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 5 | 9 | 9 | 192.168.3.15 | 23 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 5 | 10 | 10 | 192.168.3.15 | 80 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 5 | 11 | 11 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 5 | 12 | 12 | 192.168.3.15 | 21 | 1.0.0.1 | 110 | TCP | DENY | PASSED |
| 5 | 13 | 13 | 192.168.3.15 | 21 | 192.168.5.2 | 110 | TCP | DENY | PASSED |
| 5 | 14 | 14 | 192.168.3.15 | 21 | 96.84.0.1 | 110 | TCP | DENY | PASSED |
| 5 | 15 | 15 | 192.168.3.15 | 21 | 192.168.5.4 | 110 | TCP | DENY | PASSED |
| 5 | 16 | 16 | 192.168.3.15 | 21 | 255.255.255.254 | 110 | TCP | DENY | PASSED |
| 5 | 17 | 17 | 192.168.3.15 | 21 | 223.211.128.128 | 110 | TCP | DENY | PASSED |
| 5 | 18 | 18 | 192.168.3.15 | 21 | 192.168.5.3 | 0 | TCP | ALLOW | PASSED |
| 5 | 19 | 19 | 192.168.3.15 | 21 | 192.168.5.3 | 65535 | TCP | ALLOW | PASSED |
| 5 | 20 | 20 | 192.168.3.15 | 21 | 192.168.5.3 | 23 | TCP | ALLOW | PASSED |
| 5 | 21 | 21 | 192.168.3.15 | 21 | 192.168.5.3 | 80 | TCP | ALLOW | PASSED |

Table A.10. Conformance Testing - V

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Flag | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|---|
| 5 | 1 | 1 | 216.239.33.96 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 5 | 2 | 2 | 216.239.33.96 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 5 | 3 | 3 | 192.168.5.3 | 21 | 216.239.33.96 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 5 | 4 | 4 | 216.239.33.96 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 5 | 5 | 5 | 192.168.3.15 | 594 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 5 | 6 | 6 | 192.168.3.15 | 594 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 5 | 7 | 7 | 192.168.5.3 | 594 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 5 | 8 | 8 | 192.168.3.15 | 594 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 5 | 9 | 9 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 5 | 10 | 10 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 5 | 11 | 11 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 5 | 12 | 12 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 5 | 13 | 13 | 192.168.3.15 | 21 | 192.168.5.3 | 594 | RST | TCP | ALLOW | PASSED |
| 5 | 14 | 14 | 192.168.3.15 | 21 | 192.168.5.3 | 594 | FIN | TCP | ALLOW | PASSED |
| 5 | 15 | 15 | 192.168.5.3 | 21 | 192.168.3.15 | 594 | SYN/ACK | TCP | DENY | PASSED |
| 5 | 16 | 16 | 192.168.3.15 | 21 | 192.168.5.3 | 594 | SYN | TCP | ALLOW | PASSED |

## A.6. Experiment VI

The sixth experiment is performed for the case that traffic is allowed for the destination ip is a host with the port of 110. The firewall policy is configured as "iptables –A FORWARD –p tcp –d 192.168.5.3 –dport 110 –j ACCEPT".

Table A.11. Equivalence Partitioning - VI

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|
| 6 | 1 | 1 | 10.0.0.1 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 6 | 2 | 2 | 172.16.0.1 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 6 | 3 | 3 | 192.168.0.1 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 6 | 4 | 4 | 127.0.0.1 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 6 | 5 | 5 | 88.241.34.41 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 6 | 6 | 6 | 215.15.168.23 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 6 | 7 | 7 | 192.168.3.15 | 0 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 6 | 8 | 8 | 192.168.3.15 | 65535 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 6 | 9 | 9 | 192.168.3.15 | 23 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 6 | 10 | 10 | 192.168.3.15 | 80 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 6 | 11 | 11 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 6 | 12 | 12 | 192.168.3.15 | 21 | 1.0.0.1 | 110 | TCP | DENY | PASSED |
| 6 | 13 | 13 | 192.168.3.15 | 21 | 192.168.5.2 | 110 | TCP | DENY | PASSED |
| 6 | 14 | 14 | 192.168.3.15 | 21 | 96.84.0.1 | 110 | TCP | DENY | PASSED |
| 6 | 15 | 15 | 192.168.3.15 | 21 | 192.168.5.4 | 110 | TCP | DENY | PASSED |
| 6 | 16 | 16 | 192.168.3.15 | 21 | 255.255.255.254 | 110 | TCP | DENY | PASSED |
| 6 | 17 | 17 | 192.168.3.15 | 21 | 223.211.128.128 | 110 | TCP | DENY | PASSED |
| 6 | 18 | 18 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 6 | 19 | 19 | 192.168.3.15 | 21 | 192.168.5.3 | 111 | TCP | DENY | PASSED |
| 6 | 20 | 20 | 192.168.3.15 | 21 | 192.168.5.3 | 109 | TCP | DENY | PASSED |

Table A.12. Conformance Testing - VI

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Flag | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|---|
| 6 | 1 | 1 | 131.107.137.165 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 6 | 2 | 2 | 131.107.137.165 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 6 | 3 | 3 | 192.168.5.3 | 21 | 131.107.137.165 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 6 | 4 | 4 | 131.107.137.165 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 6 | 5 | 5 | 192.168.3.15 | 441 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 6 | 6 | 6 | 192.168.3.15 | 441 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 6 | 7 | 7 | 192.168.5.3 | 441 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 6 | 8 | 8 | 192.168.3.15 | 441 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 6 | 9 | 9 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 6 | 10 | 10 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 6 | 11 | 11 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 6 | 12 | 12 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 6 | 13 | 13 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 6 | 14 | 14 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 6 | 15 | 15 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 6 | 16 | 16 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |

## A.7. Experiment VII

The seventh experiment is performed for the case, policy allows all traffic comes from port 21 with the command of "iptables –A FORWARD –p tcp –sport 21 –j ACCEPT".

Table A.13. Equivalence Partitioning-VII

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|
| 7 | 1 | 1 | 10.0.0.1 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 7 | 2 | 2 | 172.16.0.1 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 7 | 3 | 3 | 192.168.0.1 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 7 | 4 | 4 | 127.0.0.1 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 7 | 5 | 5 | 88.241.34.41 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 7 | 6 | 6 | 215.15.168.23 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 7 | 7 | 7 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 7 | 8 | 8 | 192.168.3.15 | 20 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 7 | 9 | 9 | 192.168.3.15 | 22 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 7 | 10 | 10 | 192.168.3.15 | 21 | 10.0.0.1 | 110 | TCP | ALLOW | PASSED |
| 7 | 11 | 11 | 192.168.3.15 | 21 | 172.16.0.1 | 110 | TCP | ALLOW | PASSED |
| 7 | 12 | 12 | 192.168.3.15 | 21 | 192.168.0.1 | 110 | TCP | ALLOW | PASSED |
| 7 | 13 | 13 | 192.168.3.15 | 21 | 127.0.0.1 | 110 | TCP | ALLOW | PASSED |
| 7 | 14 | 14 | 192.168.3.15 | 21 | 88.241.34.41 | 110 | TCP | ALLOW | PASSED |
| 7 | 15 | 15 | 192.168.3.15 | 21 | 215.15.168.23 | 110 | TCP | ALLOW | PASSED |
| 7 | 16 | 16 | 192.168.3.15 | 21 | 192.168.5.3 | 0 | TCP | ALLOW | PASSED |
| 7 | 17 | 17 | 192.168.3.15 | 21 | 192.168.5.3 | 65535 | TCP | ALLOW | PASSED |
| 7 | 18 | 18 | 192.168.3.15 | 21 | 192.168.5.3 | 23 | TCP | ALLOW | PASSED |
| 7 | 19 | 19 | 192.168.3.15 | 21 | 192.168.5.3 | 80 | TCP | ALLOW | PASSED |

Table A.14. Conformance Testing - VII

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Flag | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|---|
| 7 | 1 | 1 | 202.212.5.30 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 7 | 2 | 2 | 202.212.5.30 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 7 | 3 | 3 | 192.168.5.3 | 21 | 202.212.5.30 | 110 | SYN/ACK | TCP | ALLOW | PASSED |
| 7 | 4 | 4 | 202.212.5.30 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 7 | 5 | 5 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 7 | 6 | 6 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 7 | 7 | 7 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | ALLOW | PASSED |
| 7 | 8 | 8 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 7 | 9 | 9 | 192.168.3.15 | 21 | 131.107.137.165 | 110 | RST | TCP | ALLOW | PASSED |
| 7 | 10 | 10 | 192.168.3.15 | 21 | 131.107.137.165 | 110 | FIN | TCP | ALLOW | PASSED |
| 7 | 11 | 11 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | ALLOW | PASSED |
| 7 | 12 | 12 | 192.168.3.15 | 21 | 131.107.137.165 | 110 | SYN | TCP | ALLOW | PASSED |
| 7 | 13 | 13 | 192.168.3.15 | 21 | 192.168.5.3 | 789 | RST | TCP | ALLOW | PASSED |
| 7 | 14 | 14 | 192.168.3.15 | 21 | 192.168.5.3 | 789 | FIN | TCP | ALLOW | PASSED |
| 7 | 15 | 15 | 192.168.5.3 | 21 | 192.168.3.15 | 789 | SYN/ACK | TCP | ALLOW | PASSED |
| 7 | 16 | 16 | 192.168.3.15 | 21 | 192.168.5.3 | 789 | SYN | TCP | ALLOW | PASSED |

## A.8. Experiment VIII

The eighths experiment is performed for the case, policy allows all traffic comes from port 21 and goes to 110 with the command of "iptables –A FORWARD –p tcp –sport 21 –dport 110 –j ACCEPT".

Table A.15. Equivalence Partitioning - VIII

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|
| 8 | 1 | 1 | 10.0.0.1 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 8 | 2 | 2 | 172.16.0.1 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 8 | 3 | 3 | 192.168.0.1 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 8 | 4 | 4 | 127.0.0.1 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 8 | 5 | 5 | 88.241.34.41 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 8 | 6 | 6 | 215.15.168.23 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 8 | 7 | 7 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 8 | 8 | 8 | 192.168.3.15 | 20 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 8 | 9 | 9 | 192.168.3.15 | 22 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 8 | 10 | 10 | 192.168.3.15 | 21 | 10.0.0.1 | 110 | TCP | ALLOW | PASSED |
| 8 | 11 | 11 | 192.168.3.15 | 21 | 172.16.0.1 | 110 | TCP | ALLOW | PASSED |
| 8 | 12 | 12 | 192.168.3.15 | 21 | 192.168.0.1 | 110 | TCP | ALLOW | PASSED |
| 8 | 13 | 13 | 192.168.3.15 | 21 | 127.0.0.1 | 110 | TCP | ALLOW | PASSED |
| 8 | 14 | 14 | 192.168.3.15 | 21 | 88.241.34.41 | 110 | TCP | ALLOW | PASSED |
| 8 | 15 | 15 | 192.168.3.15 | 21 | 215.15.168.23 | 110 | TCP | ALLOW | PASSED |
| 8 | 16 | 16 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 8 | 17 | 17 | 192.168.3.15 | 21 | 192.168.5.3 | 109 | TCP | DENY | PASSED |
| 8 | 18 | 18 | 192.168.3.15 | 21 | 192.168.5.3 | 111 | TCP | DENY | PASSED |

Table A.16. Conformance Testing - VIII

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Flag | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|---|
| 8 | 1 | 1 | 216.239.33.96 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 8 | 2 | 2 | 216.239.33.96 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 8 | 3 | 3 | 192.168.5.3 | 21 | 216.239.33.96 | 110 | SYN/ACK | TCP | ALLOW | PASSED |
| 8 | 4 | 4 | 216.239.33.96 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 8 | 5 | 5 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 8 | 6 | 6 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 8 | 7 | 7 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | ALLOW | PASSED |
| 8 | 8 | 8 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 8 | 9 | 9 | 192.168.3.15 | 21 | 131.107.137.165 | 110 | RST | TCP | ALLOW | PASSED |
| 8 | 10 | 10 | 192.168.3.15 | 21 | 131.107.137.165 | 110 | FIN | TCP | ALLOW | PASSED |
| 8 | 11 | 11 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | ALLOW | PASSED |
| 8 | 12 | 12 | 192.168.3.15 | 21 | 131.107.137.165 | 110 | SYN | TCP | ALLOW | PASSED |
| 8 | 13 | 13 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 8 | 14 | 14 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 8 | 15 | 15 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | ALLOW | PASSED |
| 8 | 16 | 16 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |

## A.9. Experiment IX

The ninth experiment is performed for the case, policy allows all traffic comes from port 21 and goes to a subnet with the command of "iptables –A FORWARD –p tcp –sport 21 –dip 192.168.5.0/24 –j ACCEPT".

Table A.17. Equivalence Partitioning - IX

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|
| 9 | 1 | 1 | 10.0.0.1 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 9 | 2 | 2 | 172.16.0.1 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 9 | 3 | 3 | 192.168.0.1 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 9 | 4 | 4 | 127.0.0.1 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 9 | 5 | 5 | 88.241.34.41 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 9 | 6 | 6 | 215.15.168.23 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 9 | 7 | 7 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 9 | 8 | 8 | 192.168.3.15 | 20 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 9 | 9 | 9 | 192.168.3.15 | 22 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 9 | 10 | 10 | 192.168.3.15 | 21 | 192.168.5.1 | 110 | TCP | ALLOW | PASSED |
| 9 | 11 | 11 | 192.168.3.15 | 21 | 192.168.5.127 | 110 | TCP | ALLOW | PASSED |
| 9 | 12 | 12 | 192.168.3.15 | 21 | 192.168.5.354 | 110 | TCP | ALLOW | PASSED |
| 9 | 13 | 13 | 192.168.3.15 | 21 | 192.168.4.1 | 110 | TCP | DENY | PASSED |
| 9 | 14 | 14 | 192.168.3.15 | 21 | 192.168.4.127 | 110 | TCP | DENY | PASSED |
| 9 | 15 | 15 | 192.168.3.15 | 21 | 192.168.4.254 | 110 | TCP | DENY | PASSED |
| 9 | 16 | 16 | 192.168.3.15 | 21 | 192.168.6.1 | 110 | TCP | DENY | PASSED |
| 9 | 17 | 17 | 192.168.3.15 | 21 | 192.168.6.127 | 110 | TCP | DENY | PASSED |
| 9 | 18 | 18 | 192.168.3.15 | 21 | 192.168.6.254 | 110 | TCP | DENY | PASSED |
| 9 | 19 | 19 | 192.168.3.15 | 21 | 192.168.5.3 | 0 | TCP | ALLOW | PASSED |
| 9 | 20 | 20 | 192.168.3.15 | 21 | 192.168.5.3 | 65535 | TCP | ALLOW | PASSED |
| 9 | 21 | 21 | 192.168.3.15 | 21 | 192.168.5.3 | 23 | TCP | ALLOW | PASSED |
| 9 | 22 | 22 | 192.168.3.15 | 21 | 192.168.5.3 | 80 | TCP | ALLOW | PASSED |

Table A.18. Conformance Testing - XI

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Flag | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|---|
| 9 | 1 | 1 | 202.212.5.30 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 9 | 2 | 2 | 202.212.5.30 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 9 | 3 | 3 | 192.168.5.3 | 21 | 202.212.5.30 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 9 | 4 | 4 | 202.212.5.30 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 9 | 5 | 5 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 9 | 6 | 6 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 9 | 7 | 7 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 9 | 8 | 8 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 9 | 9 | 9 | 192.168.3.15 | 21 | 192.168.5.314 | 110 | RST | TCP | ALLOW | PASSED |
| 9 | 10 | 10 | 192.168.3.15 | 21 | 192.168.5.314 | 110 | FIN | TCP | ALLOW | PASSED |
| 9 | 11 | 11 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 9 | 12 | 12 | 192.168.3.15 | 21 | 192.168.5.314 | 110 | SYN | TCP | ALLOW | PASSED |
| 9 | 13 | 13 | 192.168.3.15 | 21 | 192.168.5.3 | 592 | RST | TCP | ALLOW | PASSED |
| 9 | 14 | 14 | 192.168.3.15 | 21 | 192.168.5.3 | 592 | FIN | TCP | ALLOW | PASSED |
| 9 | 15 | 15 | 192.168.5.3 | 21 | 192.168.3.15 | 592 | SYN/ACK | TCP | DENY | PASSED |
| 9 | 16 | 16 | 192.168.3.15 | 21 | 192.168.5.3 | 592 | SYN | TCP | ALLOW | PASSED |

## A.10. Experiment X

The 10th experiment is performed for the case, policy allows all traffic comes from port 21 and goes to a subnet via port 110 with the command of "iptables –A FORWARD –p tcp –sport 21 –dip 192.168.5.0/24 –dport 110 –j ACCEPT".

Table A.19. Equivalence Partitioning - X

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|
| 10 | 1 | 1 | 10.0.0.1 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 10 | 2 | 2 | 172.16.0.1 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 10 | 3 | 3 | 192.168.0.1 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 10 | 4 | 4 | 127.0.0.1 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 10 | 5 | 5 | 88.241.34.41 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 10 | 6 | 6 | 215.15.168.23 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 10 | 7 | 7 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 10 | 8 | 8 | 192.168.3.15 | 20 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 10 | 9 | 9 | 192.168.3.15 | 22 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 10 | 10 | 10 | 192.168.3.15 | 21 | 192.168.5.1 | 110 | TCP | ALLOW | PASSED |
| 10 | 11 | 11 | 192.168.3.15 | 21 | 192.168.5.127 | 110 | TCP | ALLOW | PASSED |
| 10 | 12 | 12 | 192.168.3.15 | 21 | 192.168.5.354 | 110 | TCP | ALLOW | PASSED |
| 10 | 13 | 13 | 192.168.3.15 | 21 | 192.168.4.1 | 110 | TCP | DENY | PASSED |
| 10 | 14 | 14 | 192.168.3.15 | 21 | 192.168.4.127 | 110 | TCP | DENY | PASSED |
| 10 | 15 | 15 | 192.168.3.15 | 21 | 192.168.4.254 | 110 | TCP | DENY | PASSED |
| 10 | 16 | 16 | 192.168.3.15 | 21 | 192.168.6.1 | 110 | TCP | DENY | PASSED |
| 10 | 17 | 17 | 192.168.3.15 | 21 | 192.168.6.127 | 110 | TCP | DENY | PASSED |
| 10 | 18 | 18 | 192.168.3.15 | 21 | 192.168.6.254 | 110 | TCP | DENY | PASSED |
| 10 | 19 | 19 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 10 | 20 | 20 | 192.168.3.15 | 21 | 192.168.5.3 | 109 | TCP | DENY | PASSED |
| 10 | 21 | 21 | 192.168.3.15 | 21 | 192.168.5.3 | 111 | TCP | DENY | PASSED |

Table A.20. Conformance Testing - X

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Flag | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|---|
| 10 | 1 | 1 | 202.212.5.30 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 10 | 2 | 2 | 202.212.5.30 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 10 | 3 | 3 | 192.168.5.3 | 21 | 202.212.5.30 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 10 | 4 | 4 | 202.212.5.30 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 10 | 5 | 5 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 10 | 6 | 6 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 10 | 7 | 7 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 10 | 8 | 8 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 10 | 9 | 9 | 192.168.3.15 | 21 | 192.168.2.215 | 110 | RST | TCP | ALLOW | PASSED |
| 10 | 10 | 10 | 192.168.3.15 | 21 | 192.168.2.215 | 110 | FIN | TCP | ALLOW | PASSED |
| 10 | 11 | 11 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 10 | 12 | 12 | 192.168.3.15 | 21 | 192.168.2.215 | 110 | SYN | TCP | ALLOW | PASSED |
| 10 | 13 | 13 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 10 | 14 | 14 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 10 | 15 | 15 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 10 | 16 | 16 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |

## A.11. Experiment XI

The 11th experiment is performed for the case, policy allows all traffic comes from port 21 and goes to a host with the command of "iptables –A FORWARD –p tcp –sport 21 –dip 192.168.5.3 –j ACCEPT".

Table A.21. Equivalence Partitioning - XI

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|
| 11 | 1 | 1 | 10.0.0.1 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 11 | 2 | 2 | 172.16.0.1 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 11 | 3 | 3 | 192.168.0.1 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 11 | 4 | 4 | 127.0.0.1 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 11 | 5 | 5 | 88.241.34.41 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 11 | 6 | 6 | 215.15.168.23 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 11 | 7 | 7 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 11 | 8 | 8 | 192.168.3.15 | 20 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 11 | 9 | 9 | 192.168.3.15 | 22 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 11 | 10 | 10 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 11 | 11 | 11 | 192.168.3.15 | 21 | 1.0.0.1 | 110 | TCP | DENY | PASSED |
| 11 | 12 | 12 | 192.168.3.15 | 21 | 192.168.5.2 | 110 | TCP | DENY | PASSED |
| 11 | 13 | 13 | 192.168.3.15 | 21 | 96.84.0.1 | 110 | TCP | DENY | PASSED |
| 11 | 14 | 14 | 192.168.3.15 | 21 | 192.168.5.4 | 110 | TCP | DENY | PASSED |
| 11 | 15 | 15 | 192.168.3.15 | 21 | 255.255.255.254 | 110 | TCP | DENY | PASSED |
| 11 | 16 | 16 | 192.168.3.15 | 21 | 223.211.128.128 | 110 | TCP | DENY | PASSED |
| 11 | 17 | 17 | 192.168.3.15 | 21 | 192.168.5.3 | 0 | TCP | ALLOW | PASSED |
| 11 | 18 | 18 | 192.168.3.15 | 21 | 192.168.5.3 | 65535 | TCP | ALLOW | PASSED |
| 11 | 19 | 19 | 192.168.3.15 | 21 | 192.168.5.3 | 23 | TCP | ALLOW | PASSED |
| 11 | 20 | 20 | 192.168.3.15 | 21 | 192.168.5.3 | 80 | TCP | ALLOW | PASSED |

Table A.22. Conformance Testing - XI

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Flag | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|---|
| 11 | 1 | 1 | 216.239.33.96 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 11 | 2 | 2 | 216.239.33.96 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 11 | 3 | 3 | 192.168.5.3 | 21 | 216.239.33.96 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 11 | 4 | 4 | 216.239.33.96 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 11 | 5 | 5 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 11 | 6 | 6 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 11 | 7 | 7 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 11 | 8 | 8 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 11 | 9 | 9 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 11 | 10 | 10 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 11 | 11 | 11 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 11 | 12 | 12 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 11 | 13 | 13 | 192.168.3.15 | 21 | 192.168.5.3 | 950 | RST | TCP | ALLOW | PASSED |
| 11 | 14 | 14 | 192.168.3.15 | 21 | 192.168.5.3 | 950 | FIN | TCP | ALLOW | PASSED |
| 11 | 15 | 15 | 192.168.5.3 | 21 | 192.168.3.15 | 950 | SYN/ACK | TCP | DENY | PASSED |
| 11 | 16 | 16 | 192.168.3.15 | 21 | 192.168.5.3 | 950 | SYN | TCP | ALLOW | PASSED |

## A.12. Experiment XII

The 12th experiment is performed for the case, policy allows all traffic comes from port 21 and goes to a host via port 110 with the command of "iptables –A FORWARD –p tcp –sport 21 –dip 192.168.5.3 –dport 110 –j ACCEPT".

Table A.23. Equivalence Partitioning - XII

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|
| 12 | 1 | 1 | 10.0.0.1 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 12 | 2 | 2 | 172.16.0.1 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 12 | 3 | 3 | 192.168.0.1 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 12 | 4 | 4 | 127.0.0.1 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 12 | 5 | 5 | 88.241.34.41 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 12 | 6 | 6 | 215.15.168.23 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 12 | 7 | 7 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 12 | 8 | 8 | 192.168.3.15 | 20 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 12 | 9 | 9 | 192.168.3.15 | 22 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 12 | 10 | 10 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 12 | 11 | 11 | 192.168.3.15 | 21 | 1.0.0.1 | 110 | TCP | DENY | PASSED |
| 12 | 12 | 12 | 192.168.3.15 | 21 | 192.168.5.2 | 110 | TCP | DENY | PASSED |
| 12 | 13 | 13 | 192.168.3.15 | 21 | 96.84.0.1 | 110 | TCP | DENY | PASSED |
| 12 | 14 | 14 | 192.168.3.15 | 21 | 192.168.5.4 | 110 | TCP | DENY | PASSED |
| 12 | 15 | 15 | 192.168.3.15 | 21 | 255.255.255.254 | 110 | TCP | DENY | PASSED |
| 12 | 16 | 16 | 192.168.3.15 | 21 | 223.211.128.128 | 110 | TCP | DENY | PASSED |
| 12 | 17 | 17 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 12 | 18 | 18 | 192.168.3.15 | 21 | 192.168.5.3 | 109 | TCP | DENY | PASSED |
| 12 | 19 | 19 | 192.168.3.15 | 21 | 192.168.5.3 | 111 | TCP | DENY | PASSED |

Table A.24. Conformance Testing - XII

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Flag | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|---|
| 12 | 1 | 1 | 202.212.5.30 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 12 | 2 | 2 | 202.212.5.30 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 12 | 3 | 3 | 192.168.5.3 | 21 | 202.212.5.30 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 12 | 4 | 4 | 202.212.5.30 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 12 | 5 | 5 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 12 | 6 | 6 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 12 | 7 | 7 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 12 | 8 | 8 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 12 | 9 | 9 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 12 | 10 | 10 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 12 | 11 | 11 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 12 | 12 | 12 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 12 | 13 | 13 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 12 | 14 | 14 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 12 | 15 | 15 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 12 | 16 | 16 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |

## A.13. Experiment XIII

The 13th experiment is performed for the case, policy allows traffic comes from a subnet with the command of "iptables –A FORWARD –p tcp –sip 192.168.3.0/24 –j ACCEPT".

Table A.25. Equivalence Partitioning - XIII

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|
| 13 | 1 | 1 | 192.168.3.1 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 13 | 2 | 2 | 192.168.3.127 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 13 | 3 | 3 | 192.168.3.254 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 13 | 4 | 4 | 192.168.2.1 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 13 | 5 | 5 | 192.168.2.127 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 13 | 6 | 6 | 192.168.2.254 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 13 | 7 | 7 | 192.168.4.1 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 13 | 8 | 8 | 192.168.4.127 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 13 | 9 | 9 | 192.168.4.254 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 13 | 10 | 10 | 192.168.3.15 | 0 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 13 | 11 | 11 | 192.168.3.15 | 65535 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 13 | 12 | 12 | 192.168.3.15 | 23 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 13 | 13 | 13 | 192.168.3.15 | 80 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 13 | 14 | 14 | 192.168.3.15 | 21 | 10.0.0.1 | 110 | TCP | ALLOW | PASSED |
| 13 | 15 | 15 | 192.168.3.15 | 21 | 172.16.0.1 | 110 | TCP | ALLOW | PASSED |
| 13 | 16 | 16 | 192.168.3.15 | 21 | 192.168.0.1 | 110 | TCP | ALLOW | PASSED |
| 13 | 17 | 17 | 192.168.3.15 | 21 | 127.0.0.1 | 110 | TCP | ALLOW | PASSED |
| 13 | 18 | 18 | 192.168.3.15 | 21 | 88.241.34.41 | 110 | TCP | ALLOW | PASSED |
| 13 | 19 | 19 | 192.168.3.15 | 21 | 215.15.168.23 | 110 | TCP | ALLOW | PASSED |
| 13 | 20 | 20 | 192.168.3.15 | 21 | 192.168.5.3 | 0 | TCP | ALLOW | PASSED |
| 13 | 21 | 21 | 192.168.3.15 | 21 | 192.168.5.3 | 65535 | TCP | ALLOW | PASSED |
| 13 | 22 | 22 | 192.168.3.15 | 21 | 192.168.5.3 | 23 | TCP | ALLOW | PASSED |
| 13 | 23 | 23 | 192.168.3.15 | 21 | 192.168.5.3 | 80 | TCP | ALLOW | PASSED |

Table A.26. Conformance Testing - XIII

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Flag | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|---|
| 13 | 1 | 1 | 192.168.3.75 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 13 | 2 | 2 | 192.168.3.75 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 13 | 3 | 3 | 192.168.5.3 | 21 | 192.168.3.75 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 13 | 4 | 4 | 192.168.3.75 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 13 | 5 | 5 | 192.168.3.15 | 644 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 13 | 6 | 6 | 192.168.3.15 | 644 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 13 | 7 | 7 | 192.168.5.3 | 644 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 13 | 8 | 8 | 192.168.3.15 | 644 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 13 | 9 | 9 | 192.168.3.15 | 21 | 202.212.5.30 | 110 | RST | TCP | ALLOW | PASSED |
| 13 | 10 | 10 | 192.168.3.15 | 21 | 202.212.5.30 | 110 | FIN | TCP | ALLOW | PASSED |
| 13 | 11 | 11 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 13 | 12 | 12 | 192.168.3.15 | 21 | 202.212.5.30 | 110 | SYN | TCP | ALLOW | PASSED |
| 13 | 13 | 13 | 192.168.3.15 | 21 | 192.168.5.3 | 644 | RST | TCP | ALLOW | PASSED |
| 13 | 14 | 14 | 192.168.3.15 | 21 | 192.168.5.3 | 644 | FIN | TCP | ALLOW | PASSED |
| 13 | 15 | 15 | 192.168.5.3 | 21 | 192.168.3.15 | 644 | SYN/ACK | TCP | DENY | PASSED |
| 13 | 16 | 16 | 192.168.3.15 | 21 | 192.168.5.3 | 644 | SYN | TCP | ALLOW | PASSED |

## A.14. Experiment XIV

The 14th experiment is performed for the case, policy allows traffic comes from a subnet and goes to port 110 with the command of "iptables –A FORWARD –p tcp –sip 192.168.3.0/24 –dport 110 –j ACCEPT".

Table A.27. Equivalence Partitioning - XIV

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|
| 14 | 1 | 1 | 192.168.3.1 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 14 | 2 | 2 | 192.168.3.127 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 14 | 3 | 3 | 192.168.3.254 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 14 | 4 | 4 | 192.168.2.1 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 14 | 5 | 5 | 192.168.2.127 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 14 | 6 | 6 | 192.168.2.254 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 14 | 7 | 7 | 192.168.4.1 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 14 | 8 | 8 | 192.168.4.127 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 14 | 9 | 9 | 192.168.4.254 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 14 | 10 | 10 | 192.168.3.15 | 0 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 14 | 11 | 11 | 192.168.3.15 | 65535 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 14 | 12 | 12 | 192.168.3.15 | 23 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 14 | 13 | 13 | 192.168.3.15 | 80 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 14 | 14 | 14 | 192.168.3.15 | 21 | 10.0.0.1 | 110 | TCP | ALLOW | PASSED |
| 14 | 15 | 15 | 192.168.3.15 | 21 | 172.16.0.1 | 110 | TCP | ALLOW | PASSED |
| 14 | 16 | 16 | 192.168.3.15 | 21 | 192.168.0.1 | 110 | TCP | ALLOW | PASSED |
| 14 | 17 | 17 | 192.168.3.15 | 21 | 127.0.0.1 | 110 | TCP | ALLOW | PASSED |
| 14 | 18 | 18 | 192.168.3.15 | 21 | 88.241.34.41 | 110 | TCP | ALLOW | PASSED |
| 14 | 19 | 19 | 192.168.3.15 | 21 | 215.15.168.23 | 110 | TCP | ALLOW | PASSED |
| 14 | 20 | 20 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 14 | 21 | 21 | 192.168.3.15 | 21 | 192.168.5.3 | 111 | TCP | DENY | PASSED |
| 14 | 22 | 22 | 192.168.3.15 | 21 | 192.168.5.3 | 109 | TCP | DENY | PASSED |

Table A.28. Conformance Testing - XIV

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Flag | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|---|
| 14 | 1 | 1 | 192.168.3.32 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 14 | 2 | 2 | 192.168.3.32 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 14 | 3 | 3 | 192.168.5.3 | 21 | 192.168.3.32 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 14 | 4 | 4 | 192.168.3.32 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 14 | 5 | 5 | 192.168.3.15 | 95 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 14 | 6 | 6 | 192.168.3.15 | 95 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 14 | 7 | 7 | 192.168.5.3 | 95 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 14 | 8 | 8 | 192.168.3.15 | 95 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 14 | 9 | 9 | 192.168.3.15 | 21 | 131.107.137.165 | 110 | RST | TCP | ALLOW | PASSED |
| 14 | 10 | 10 | 192.168.3.15 | 21 | 131.107.137.165 | 110 | FIN | TCP | ALLOW | PASSED |
| 14 | 11 | 11 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 14 | 12 | 12 | 192.168.3.15 | 21 | 131.107.137.165 | 110 | SYN | TCP | ALLOW | PASSED |
| 14 | 13 | 13 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 14 | 14 | 14 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 14 | 15 | 15 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 14 | 16 | 16 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |

## A.15. Experiment XV

The 15th experiment is performed for the case, policy allows traffic comes from a subnet and goes to a host with the command of "iptables –A FORWARD –p tcp –sip 192.168.3.0/24 –dip 192.168.5.0/24 –j ACCEPT".

Table A.29. Equivalence Partitioning - XV

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|
| 15 | 1 | 1 | 192.168.3.1 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 15 | 2 | 2 | 192.168.3.127 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 15 | 3 | 3 | 192.168.3.254 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 15 | 4 | 4 | 192.168.2.1 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 15 | 5 | 5 | 192.168.2.127 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 15 | 6 | 6 | 192.168.2.254 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 15 | 7 | 7 | 192.168.4.1 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 15 | 8 | 8 | 192.168.4.127 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 15 | 9 | 9 | 192.168.4.254 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 15 | 10 | 10 | 192.168.3.15 | 0 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 15 | 11 | 11 | 192.168.3.15 | 65535 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 15 | 12 | 12 | 192.168.3.15 | 23 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 15 | 13 | 13 | 192.168.3.15 | 80 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 15 | 14 | 14 | 192.168.3.15 | 21 | 192.168.5.1 | 110 | TCP | ALLOW | PASSED |
| 15 | 15 | 15 | 192.168.3.15 | 21 | 192.168.5.127 | 110 | TCP | ALLOW | PASSED |
| 15 | 16 | 16 | 192.168.3.15 | 21 | 192.168.5.354 | 110 | TCP | ALLOW | PASSED |
| 15 | 17 | 17 | 192.168.3.15 | 21 | 192.168.4.1 | 110 | TCP | DENY | PASSED |
| 15 | 18 | 18 | 192.168.3.15 | 21 | 192.168.4.127 | 110 | TCP | DENY | PASSED |
| 15 | 19 | 19 | 192.168.3.15 | 21 | 192.168.4.254 | 110 | TCP | DENY | PASSED |
| 15 | 20 | 20 | 192.168.3.15 | 21 | 192.168.6.1 | 110 | TCP | DENY | PASSED |
| 15 | 21 | 21 | 192.168.3.15 | 21 | 192.168.6.127 | 110 | TCP | DENY | PASSED |
| 15 | 22 | 22 | 192.168.3.15 | 21 | 192.168.6.254 | 110 | TCP | DENY | PASSED |
| 15 | 23 | 23 | 192.168.3.15 | 21 | 192.168.5.3 | 0 | TCP | ALLOW | PASSED |
| 15 | 24 | 24 | 192.168.3.15 | 21 | 192.168.5.3 | 65535 | TCP | ALLOW | PASSED |
| 15 | 25 | 25 | 192.168.3.15 | 21 | 192.168.5.3 | 23 | TCP | ALLOW | PASSED |
| 15 | 26 | 26 | 192.168.3.15 | 21 | 192.168.5.3 | 80 | TCP | ALLOW | PASSED |

Table A.30. Conformance Testing - XV

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Flag | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|---|
| 15 | 1 | 1 | 192.168.3.20 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 15 | 2 | 2 | 192.168.3.20 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 15 | 3 | 3 | 192.168.5.3 | 21 | 192.168.3.20 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 15 | 4 | 4 | 192.168.3.20 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 15 | 5 | 5 | 192.168.3.15 | 418 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 15 | 6 | 6 | 192.168.3.15 | 418 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 15 | 7 | 7 | 192.168.5.3 | 418 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 15 | 8 | 8 | 192.168.3.15 | 418 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 15 | 9 | 9 | 192.168.3.15 | 21 | 192.168.2.20 | 110 | RST | TCP | ALLOW | PASSED |
| 15 | 10 | 10 | 192.168.3.15 | 21 | 192.168.2.20 | 110 | FIN | TCP | ALLOW | PASSED |
| 15 | 11 | 11 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 15 | 12 | 12 | 192.168.3.15 | 21 | 192.168.2.20 | 110 | SYN | TCP | ALLOW | PASSED |
| 15 | 13 | 13 | 192.168.3.15 | 21 | 192.168.5.3 | 418 | RST | TCP | ALLOW | PASSED |
| 15 | 14 | 14 | 192.168.3.15 | 21 | 192.168.5.3 | 418 | FIN | TCP | ALLOW | PASSED |
| 15 | 15 | 15 | 192.168.5.3 | 21 | 192.168.3.15 | 418 | SYN/ACK | TCP | DENY | PASSED |
| 15 | 16 | 16 | 192.168.3.15 | 21 | 192.168.5.3 | 418 | SYN | TCP | ALLOW | PASSED |

## A.16. Experiment XVI

The 16th experiment is performed for the case, policy allows traffic comes from a subnet and goes to a subnet via port 110 with the command of "iptables –A FORWARD –p tcp –sip 192.168.3.0/24 –dip 192.168.5.0/24 –dport 110 –j ACCEPT".

Table A.31. Equivalence Partitioning - XVI

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|
| 16 | 1 | 1 | 192.168.3.1 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 16 | 2 | 2 | 192.168.3.127 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 16 | 3 | 3 | 192.168.3.254 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 16 | 4 | 4 | 192.168.2.1 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 16 | 5 | 5 | 192.168.2.127 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 16 | 6 | 6 | 192.168.2.254 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 16 | 7 | 7 | 192.168.4.1 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 16 | 8 | 8 | 192.168.4.127 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 16 | 9 | 9 | 192.168.4.254 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 16 | 10 | 10 | 192.168.3.15 | 0 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 16 | 11 | 11 | 192.168.3.15 | 65535 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 16 | 12 | 12 | 192.168.3.15 | 23 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 16 | 13 | 13 | 192.168.3.15 | 80 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 16 | 14 | 14 | 192.168.3.15 | 21 | 192.168.5.1 | 110 | TCP | ALLOW | PASSED |
| 16 | 15 | 15 | 192.168.3.15 | 21 | 192.168.5.127 | 110 | TCP | ALLOW | PASSED |
| 16 | 16 | 16 | 192.168.3.15 | 21 | 192.168.5.354 | 110 | TCP | ALLOW | PASSED |
| 16 | 17 | 17 | 192.168.3.15 | 21 | 192.168.4.1 | 110 | TCP | DENY | PASSED |
| 16 | 18 | 18 | 192.168.3.15 | 21 | 192.168.4.127 | 110 | TCP | DENY | PASSED |
| 16 | 19 | 19 | 192.168.3.15 | 21 | 192.168.4.254 | 110 | TCP | DENY | PASSED |
| 16 | 20 | 20 | 192.168.3.15 | 21 | 192.168.6.1 | 110 | TCP | DENY | PASSED |
| 16 | 21 | 21 | 192.168.3.15 | 21 | 192.168.6.127 | 110 | TCP | DENY | PASSED |
| 16 | 22 | 22 | 192.168.3.15 | 21 | 192.168.6.254 | 110 | TCP | DENY | PASSED |
| 16 | 23 | 23 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 16 | 24 | 24 | 192.168.3.15 | 21 | 192.168.5.3 | 111 | TCP | DENY | PASSED |
| 16 | 25 | 25 | 192.168.3.15 | 21 | 192.168.5.3 | 109 | TCP | DENY | PASSED |

Table A.32. Conformance Testing - XVI

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Flag | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|---|
| 16 | 1 | 1 | 192.168.3.174 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 16 | 2 | 2 | 192.168.3.174 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 16 | 3 | 3 | 192.168.5.3 | 21 | 192.168.3.174 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 16 | 4 | 4 | 192.168.3.174 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 16 | 5 | 5 | 192.168.3.15 | 240 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 16 | 6 | 6 | 192.168.3.15 | 240 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 16 | 7 | 7 | 192.168.5.3 | 240 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 16 | 8 | 8 | 192.168.3.15 | 240 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 16 | 9 | 9 | 192.168.3.15 | 21 | 192.168.5.374 | 110 | RST | TCP | ALLOW | PASSED |
| 16 | 10 | 10 | 192.168.3.15 | 21 | 192.168.5.374 | 110 | FIN | TCP | ALLOW | PASSED |
| 16 | 11 | 11 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 16 | 12 | 12 | 192.168.3.15 | 21 | 192.168.5.374 | 110 | SYN | TCP | ALLOW | PASSED |
| 16 | 13 | 13 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 16 | 14 | 14 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 16 | 15 | 15 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 16 | 16 | 16 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |

## A.17. Experiment XVII

The 17th experiment is performed for the case, policy allows traffic comes from a subnet and goes to a host with the command of "iptables –A FORWARD –p tcp –sip 192.168.3.0/24 –dip 192.168.5.3 –j ACCEPT".

Table A.33. Equivalence Partitioning - XVII

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|
| 17 | 1 | 1 | 192.168.3.1 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 17 | 2 | 2 | 192.168.3.127 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 17 | 3 | 3 | 192.168.3.254 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 17 | 4 | 4 | 192.168.2.1 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 17 | 5 | 5 | 192.168.2.127 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 17 | 6 | 6 | 192.168.2.254 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 17 | 7 | 7 | 192.168.4.1 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 17 | 8 | 8 | 192.168.4.127 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 17 | 9 | 9 | 192.168.4.254 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 17 | 10 | 10 | 192.168.3.15 | 0 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 17 | 11 | 11 | 192.168.3.15 | 65535 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 17 | 12 | 12 | 192.168.3.15 | 23 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 17 | 13 | 13 | 192.168.3.15 | 80 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 17 | 14 | 14 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 17 | 15 | 15 | 192.168.3.15 | 21 | 1.0.0.1 | 110 | TCP | DENY | PASSED |
| 17 | 16 | 16 | 192.168.3.15 | 21 | 192.168.5.2 | 110 | TCP | DENY | PASSED |
| 17 | 17 | 17 | 192.168.3.15 | 21 | 96.84.0.1 | 110 | TCP | DENY | PASSED |
| 17 | 18 | 18 | 192.168.3.15 | 21 | 192.168.5.4 | 110 | TCP | DENY | PASSED |
| 17 | 19 | 19 | 192.168.3.15 | 21 | 255.255.255.254 | 110 | TCP | DENY | PASSED |
| 17 | 20 | 20 | 192.168.3.15 | 21 | 223.211.128.128 | 110 | TCP | DENY | PASSED |
| 17 | 21 | 21 | 192.168.3.15 | 21 | 192.168.5.3 | 0 | TCP | ALLOW | PASSED |
| 17 | 22 | 22 | 192.168.3.15 | 21 | 192.168.5.3 | 65535 | TCP | ALLOW | PASSED |
| 17 | 23 | 23 | 192.168.3.15 | 21 | 192.168.5.3 | 23 | TCP | ALLOW | PASSED |
| 17 | 24 | 24 | 192.168.3.15 | 21 | 192.168.5.3 | 80 | TCP | ALLOW | PASSED |

Table A.34. Conformance Testing - XVII

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Flag | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|---|
| 17 | 1 | 1 | 192.168.3.28 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 17 | 2 | 2 | 192.168.3.28 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 17 | 3 | 3 | 192.168.5.3 | 21 | 192.168.3.28 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 17 | 4 | 4 | 192.168.3.28 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 17 | 5 | 5 | 192.168.3.15 | 563 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 17 | 6 | 6 | 192.168.3.15 | 563 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 17 | 7 | 7 | 192.168.5.3 | 563 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 17 | 8 | 8 | 192.168.3.15 | 563 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 17 | 9 | 9 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 17 | 10 | 10 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 17 | 11 | 11 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 17 | 12 | 12 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 17 | 13 | 13 | 192.168.3.15 | 21 | 192.168.5.3 | 563 | RST | TCP | ALLOW | PASSED |
| 17 | 14 | 14 | 192.168.3.15 | 21 | 192.168.5.3 | 563 | FIN | TCP | ALLOW | PASSED |
| 17 | 15 | 15 | 192.168.5.3 | 21 | 192.168.3.15 | 563 | SYN/ACK | TCP | DENY | PASSED |
| 17 | 16 | 16 | 192.168.3.15 | 21 | 192.168.5.3 | 563 | SYN | TCP | ALLOW | PASSED |

## A.18. Experiment XVIII

The 18th experiment is performed for the case, policy allows traffic comes from a subnet and goes to a host via port 110 with the command of "iptables –A FORWARD –p tcp –sip 192.168.3.0/24 –dip 192.168.5.3 –dport 110 –j ACCEPT".

Table A.35. Equivalence Partitioning - XVIII

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|
| 18 | 1 | 1 | 192.168.3.1 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 18 | 2 | 2 | 192.168.3.127 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 18 | 3 | 3 | 192.168.3.254 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 18 | 4 | 4 | 192.168.2.1 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 18 | 5 | 5 | 192.168.2.127 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 18 | 6 | 6 | 192.168.2.254 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 18 | 7 | 7 | 192.168.4.1 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 18 | 8 | 8 | 192.168.4.127 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 18 | 9 | 9 | 192.168.4.254 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 18 | 10 | 10 | 192.168.3.15 | 0 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 18 | 11 | 11 | 192.168.3.15 | 65535 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 18 | 12 | 12 | 192.168.3.15 | 23 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 18 | 13 | 13 | 192.168.3.15 | 80 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 18 | 14 | 14 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 18 | 15 | 15 | 192.168.3.15 | 21 | 1.0.0.1 | 110 | TCP | DENY | PASSED |
| 18 | 16 | 16 | 192.168.3.15 | 21 | 192.168.5.2 | 110 | TCP | DENY | PASSED |
| 18 | 17 | 17 | 192.168.3.15 | 21 | 96.84.0.1 | 110 | TCP | DENY | PASSED |
| 18 | 18 | 18 | 192.168.3.15 | 21 | 192.168.5.4 | 110 | TCP | DENY | PASSED |
| 18 | 19 | 19 | 192.168.3.15 | 21 | 255.255.255.254 | 110 | TCP | DENY | PASSED |
| 18 | 20 | 20 | 192.168.3.15 | 21 | 223.211.128.128 | 110 | TCP | DENY | PASSED |
| 18 | 21 | 21 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 18 | 22 | 22 | 192.168.3.15 | 21 | 192.168.5.3 | 111 | TCP | DENY | PASSED |
| 18 | 22 | 22 | 192.168.3.16 | 21 | 192.168.5.3 | 109 | TCP | DENY | PASSED |

Table A.36. Conformance Testing - XVIII

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Flag | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|---|
| 18 | 1 | 1 | 192.168.3.125 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 18 | 2 | 2 | 192.168.3.125 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 18 | 3 | 3 | 192.168.5.3 | 21 | 192.168.3.125 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 18 | 4 | 4 | 192.168.3.125 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 18 | 5 | 5 | 192.168.3.15 | 675 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 18 | 6 | 6 | 192.168.3.15 | 675 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 18 | 7 | 7 | 192.168.5.3 | 675 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 18 | 8 | 8 | 192.168.3.15 | 675 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 18 | 9 | 9 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 18 | 10 | 10 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 18 | 11 | 11 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 18 | 12 | 12 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 18 | 13 | 13 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 18 | 14 | 14 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 18 | 15 | 15 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 18 | 16 | 16 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |

## A.19. Experiment XIX

The 19th experiment is performed for the case, policy allows traffic comes from a subnet with the port 21 with the command of "iptables –A FORWARD –p tcp –sip 192.168.3.0/24 –sport 21 –j ACCEPT".

Table A.37. Equivalence Partitioning - XIX

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|
| 19 | 1 | 1 | 192.168.3.1 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 19 | 2 | 2 | 192.168.3.127 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 19 | 3 | 3 | 192.168.3.254 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 19 | 4 | 4 | 192.168.2.1 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 19 | 5 | 5 | 192.168.2.127 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 19 | 6 | 6 | 192.168.2.254 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 19 | 7 | 7 | 192.168.4.1 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 19 | 8 | 8 | 192.168.4.127 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 19 | 9 | 9 | 192.168.4.254 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 19 | 10 | 10 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 19 | 11 | 11 | 192.168.3.15 | 20 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 19 | 12 | 12 | 192.168.3.15 | 22 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 19 | 13 | 13 | 192.168.3.15 | 21 | 10.0.0.1 | 110 | TCP | ALLOW | PASSED |
| 19 | 14 | 14 | 192.168.3.15 | 21 | 172.16.0.1 | 110 | TCP | ALLOW | PASSED |
| 19 | 15 | 15 | 192.168.3.15 | 21 | 192.168.0.1 | 110 | TCP | ALLOW | PASSED |
| 19 | 16 | 16 | 192.168.3.15 | 21 | 127.0.0.1 | 110 | TCP | ALLOW | PASSED |
| 19 | 17 | 17 | 192.168.3.15 | 21 | 88.241.34.41 | 110 | TCP | ALLOW | PASSED |
| 19 | 18 | 18 | 192.168.3.15 | 21 | 215.15.168.23 | 110 | TCP | ALLOW | PASSED |
| 19 | 19 | 19 | 192.168.3.15 | 21 | 192.168.5.3 | 0 | TCP | ALLOW | PASSED |
| 19 | 20 | 20 | 192.168.3.15 | 21 | 192.168.5.3 | 65535 | TCP | ALLOW | PASSED |
| 19 | 21 | 21 | 192.168.3.15 | 21 | 192.168.5.3 | 23 | TCP | ALLOW | PASSED |
| 19 | 22 | 22 | 192.168.3.15 | 21 | 192.168.5.3 | 80 | TCP | ALLOW | PASSED |

Table A.38. Conformance Testing - XIX

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Flag | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|---|
| 19 | 1 | 1 | 192.168.3.118 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 19 | 2 | 2 | 192.168.3.118 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 19 | 3 | 3 | 192.168.5.3 | 21 | 192.168.3.118 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 19 | 4 | 4 | 192.168.3.118 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 19 | 5 | 5 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 19 | 6 | 6 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 19 | 7 | 7 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 19 | 8 | 8 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 19 | 9 | 9 | 192.168.3.15 | 21 | 216.239.33.96 | 110 | RST | TCP | ALLOW | PASSED |
| 19 | 10 | 10 | 192.168.3.15 | 21 | 216.239.33.96 | 110 | FIN | TCP | ALLOW | PASSED |
| 19 | 11 | 11 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 19 | 12 | 12 | 192.168.3.15 | 21 | 216.239.33.96 | 110 | SYN | TCP | ALLOW | PASSED |
| 19 | 13 | 13 | 192.168.3.15 | 21 | 192.168.5.3 | 433 | RST | TCP | ALLOW | PASSED |
| 19 | 14 | 14 | 192.168.3.15 | 21 | 192.168.5.3 | 433 | FIN | TCP | ALLOW | PASSED |
| 19 | 15 | 15 | 192.168.5.3 | 21 | 192.168.3.15 | 433 | SYN/ACK | TCP | DENY | PASSED |
| 19 | 16 | 16 | 192.168.3.15 | 21 | 192.168.5.3 | 433 | SYN | TCP | ALLOW | PASSED |

## A.20. Experiment XX

The 20th experiment is performed for the case, policy allows traffic comes from a subnet with the port 21 and goes to port 110 with the command of "iptables –A FORWARD –p tcp –sip 192.168.3.0/24 –sport 21 –dport 110 –j ACCEPT".

Table A.39. Equivalence Partitioning - XX

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|
| 20 | 1 | 1 | 192.168.3.1 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 20 | 2 | 2 | 192.168.3.127 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 20 | 3 | 3 | 192.168.3.254 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 20 | 4 | 4 | 192.168.2.1 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 20 | 5 | 5 | 192.168.2.127 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 20 | 6 | 6 | 192.168.2.254 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 20 | 7 | 7 | 192.168.4.1 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 20 | 8 | 8 | 192.168.4.127 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 20 | 9 | 9 | 192.168.4.254 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 20 | 10 | 10 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 20 | 11 | 11 | 192.168.3.15 | 20 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 20 | 12 | 12 | 192.168.3.15 | 22 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 20 | 13 | 13 | 192.168.3.15 | 21 | 10.0.0.1 | 110 | TCP | ALLOW | PASSED |
| 20 | 14 | 14 | 192.168.3.15 | 21 | 172.16.0.1 | 110 | TCP | ALLOW | PASSED |
| 20 | 15 | 15 | 192.168.3.15 | 21 | 192.168.0.1 | 110 | TCP | ALLOW | PASSED |
| 20 | 16 | 16 | 192.168.3.15 | 21 | 127.0.0.1 | 110 | TCP | ALLOW | PASSED |
| 20 | 17 | 17 | 192.168.3.15 | 21 | 88.241.34.41 | 110 | TCP | ALLOW | PASSED |
| 20 | 18 | 18 | 192.168.3.15 | 21 | 215.15.168.23 | 110 | TCP | ALLOW | PASSED |
| 20 | 19 | 19 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 20 | 20 | 20 | 192.168.3.15 | 21 | 192.168.5.3 | 111 | TCP | DENY | PASSED |
| 20 | 21 | 21 | 192.168.3.15 | 21 | 192.168.5.3 | 109 | TCP | DENY | PASSED |

Table A.40. Conformance Testing - XX

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Flag | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|---|
| 20 | 1 | 1 | 192.168.3.148 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 20 | 2 | 2 | 192.168.3.148 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 20 | 3 | 3 | 192.168.5.3 | 21 | 192.168.3.148 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 20 | 4 | 4 | 192.168.3.148 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 20 | 5 | 5 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 20 | 6 | 6 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 20 | 7 | 7 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 20 | 8 | 8 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 20 | 9 | 9 | 192.168.3.15 | 21 | 216.239.33.96 | 110 | RST | TCP | ALLOW | PASSED |
| 20 | 10 | 10 | 192.168.3.15 | 21 | 216.239.33.96 | 110 | FIN | TCP | ALLOW | PASSED |
| 20 | 11 | 11 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 20 | 12 | 12 | 192.168.3.15 | 21 | 216.239.33.96 | 110 | SYN | TCP | ALLOW | PASSED |
| 20 | 13 | 13 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 20 | 14 | 14 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 20 | 15 | 15 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 20 | 16 | 16 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |

## A.21. Experiment XXI

The 21th experiment is performed for the case, policy allows traffic comes from a subnet with the port 21 and goes to a subnet with the command of "iptables –A FORWARD –p tcp –sip 192.168.3.0/24 –sport 21 –dip 192.168.5.0/24 –j ACCEPT".

Table A.41. Equivalence Partitioning - XXI

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|
| 21 | 1 | 1 | 192.168.3.1 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 21 | 2 | 2 | 192.168.3.127 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 21 | 3 | 3 | 192.168.3.254 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 21 | 4 | 4 | 192.168.2.1 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 21 | 5 | 5 | 192.168.2.127 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 21 | 6 | 6 | 192.168.2.254 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 21 | 7 | 7 | 192.168.4.1 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 21 | 8 | 8 | 192.168.4.127 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 21 | 9 | 9 | 192.168.4.254 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 21 | 10 | 10 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 21 | 11 | 11 | 192.168.3.15 | 20 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 21 | 12 | 12 | 192.168.3.15 | 22 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 21 | 13 | 13 | 192.168.3.15 | 21 | 192.168.5.1 | 110 | TCP | ALLOW | PASSED |
| 21 | 14 | 14 | 192.168.3.15 | 21 | 192.168.5.127 | 110 | TCP | ALLOW | PASSED |
| 21 | 15 | 15 | 192.168.3.15 | 21 | 192.168.5.354 | 110 | TCP | ALLOW | PASSED |
| 21 | 16 | 16 | 192.168.3.15 | 21 | 192.168.4.1 | 110 | TCP | DENY | PASSED |
| 21 | 17 | 17 | 192.168.3.15 | 21 | 192.168.4.127 | 110 | TCP | DENY | PASSED |
| 21 | 18 | 18 | 192.168.3.15 | 21 | 192.168.4.254 | 110 | TCP | DENY | PASSED |
| 21 | 19 | 19 | 192.168.3.15 | 21 | 192.168.6.1 | 110 | TCP | DENY | PASSED |
| 21 | 20 | 20 | 192.168.3.15 | 21 | 192.168.6.127 | 110 | TCP | DENY | PASSED |
| 21 | 21 | 21 | 192.168.3.15 | 21 | 192.168.6.254 | 110 | TCP | DENY | PASSED |
| 21 | 22 | 22 | 192.168.3.15 | 21 | 192.168.5.3 | 0 | TCP | ALLOW | PASSED |
| 21 | 23 | 23 | 192.168.3.15 | 21 | 192.168.5.3 | 65535 | TCP | ALLOW | PASSED |
| 21 | 24 | 24 | 192.168.3.15 | 21 | 192.168.5.3 | 23 | TCP | ALLOW | PASSED |
| 21 | 25 | 25 | 192.168.3.15 | 21 | 192.168.5.3 | 80 | TCP | ALLOW | PASSED |

Table A.42. Conformance Testing - XXI

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Flag | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|---|
| 21 | 1 | 1 | 192.168.3.120 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 21 | 2 | 2 | 192.168.3.120 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 21 | 3 | 3 | 192.168.5.3 | 21 | 192.168.3.120 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 21 | 4 | 4 | 192.168.3.120 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 21 | 5 | 5 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 21 | 6 | 6 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 21 | 7 | 7 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 21 | 8 | 8 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 21 | 9 | 9 | 192.168.3.15 | 21 | 192.168.5.320 | 110 | RST | TCP | ALLOW | PASSED |
| 21 | 10 | 10 | 192.168.3.15 | 21 | 192.168.5.320 | 110 | FIN | TCP | ALLOW | PASSED |
| 21 | 11 | 11 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 21 | 12 | 12 | 192.168.3.15 | 21 | 192.168.5.320 | 110 | SYN | TCP | ALLOW | PASSED |
| 21 | 13 | 13 | 192.168.3.15 | 21 | 192.168.5.3 | 713 | RST | TCP | ALLOW | PASSED |
| 21 | 14 | 14 | 192.168.3.15 | 21 | 192.168.5.3 | 713 | FIN | TCP | ALLOW | PASSED |
| 21 | 15 | 15 | 192.168.5.3 | 21 | 192.168.3.15 | 713 | SYN/ACK | TCP | DENY | PASSED |
| 21 | 16 | 16 | 192.168.3.15 | 21 | 192.168.5.3 | 713 | SYN | TCP | ALLOW | PASSED |

## A.22. Experiment XXII

The 22th experiment is performed for the case, policy allows traffic comes from a subnet with the port 21 and goes to a subnet via port 110 with the command of "iptables –A FORWARD –p tcp –sip 192.168.3.0/24 –sport 21 –dip 192.168.5.0/24 –dport 110 –j ACCEPT".

Table A.43. Equivalence Partitioning - XXII

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|
| 22 | 1 | 1 | 192.168.3.1 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 22 | 2 | 2 | 192.168.3.127 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 22 | 3 | 3 | 192.168.3.254 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 22 | 4 | 4 | 192.168.2.1 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 22 | 5 | 5 | 192.168.2.127 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 22 | 6 | 6 | 192.168.2.254 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 22 | 7 | 7 | 192.168.4.1 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 22 | 8 | 8 | 192.168.4.127 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 22 | 9 | 9 | 192.168.4.254 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 22 | 10 | 10 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 22 | 11 | 11 | 192.168.3.15 | 20 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 22 | 12 | 12 | 192.168.3.15 | 22 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 22 | 13 | 13 | 192.168.3.15 | 21 | 192.168.5.1 | 110 | TCP | ALLOW | PASSED |
| 22 | 14 | 14 | 192.168.3.15 | 21 | 192.168.5.127 | 110 | TCP | ALLOW | PASSED |
| 22 | 15 | 15 | 192.168.3.15 | 21 | 192.168.5.354 | 110 | TCP | ALLOW | PASSED |
| 22 | 16 | 16 | 192.168.3.15 | 21 | 192.168.4.1 | 110 | TCP | DENY | PASSED |
| 22 | 17 | 17 | 192.168.3.15 | 21 | 192.168.4.127 | 110 | TCP | DENY | PASSED |
| 22 | 18 | 18 | 192.168.3.15 | 21 | 192.168.4.254 | 110 | TCP | DENY | PASSED |
| 22 | 19 | 19 | 192.168.3.15 | 21 | 192.168.6.1 | 110 | TCP | DENY | PASSED |
| 22 | 20 | 20 | 192.168.3.15 | 21 | 192.168.6.127 | 110 | TCP | DENY | PASSED |
| 22 | 21 | 21 | 192.168.3.15 | 21 | 192.168.6.254 | 110 | TCP | DENY | PASSED |
| 22 | 22 | 22 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 22 | 23 | 23 | 192.168.3.15 | 21 | 192.168.5.3 | 111 | TCP | DENY | PASSED |
| 22 | 24 | 24 | 192.168.3.15 | 21 | 192.168.5.3 | 109 | TCP | DENY | PASSED |

Table A.44. Conformance Testing - XXII

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Flag | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|---|
| 22 | 1 | 1 | 192.168.3.215 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 22 | 2 | 2 | 192.168.3.215 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 22 | 3 | 3 | 192.168.5.3 | 21 | 192.168.3.215 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 22 | 4 | 4 | 192.168.3.215 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 22 | 5 | 5 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 22 | 6 | 6 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 22 | 7 | 7 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 22 | 8 | 8 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 22 | 9 | 9 | 192.168.3.15 | 21 | 192.168.2.215 | 110 | RST | TCP | ALLOW | PASSED |
| 22 | 10 | 10 | 192.168.3.15 | 21 | 192.168.2.215 | 110 | FIN | TCP | ALLOW | PASSED |
| 22 | 11 | 11 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 22 | 12 | 12 | 192.168.3.15 | 21 | 192.168.2.215 | 110 | SYN | TCP | ALLOW | PASSED |
| 22 | 13 | 13 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 22 | 14 | 14 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 22 | 15 | 15 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 22 | 16 | 16 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |

## A.23. Experiment XXIII

The 23th experiment is performed for the case, policy allows traffic comes from a subnet with the port 21 and goes to a host with the command of "iptables –A FORWARD –p tcp –sip 192.168.3.0/24 –sport 21 –dip 192.168.5.3 –j ACCEPT".

Table A.45. Equivalence Partitioning - XXIII

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|
| 23 | 1 | 1 | 192.168.3.1 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 23 | 2 | 2 | 192.168.3.127 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 23 | 3 | 3 | 192.168.3.254 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 23 | 4 | 4 | 192.168.2.1 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 23 | 5 | 5 | 192.168.2.127 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 23 | 6 | 6 | 192.168.2.254 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 23 | 7 | 7 | 192.168.4.1 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 23 | 8 | 8 | 192.168.4.127 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 23 | 9 | 9 | 192.168.4.254 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 23 | 10 | 10 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 23 | 11 | 11 | 192.168.3.15 | 20 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 23 | 12 | 12 | 192.168.3.15 | 22 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 23 | 13 | 13 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 23 | 14 | 14 | 192.168.3.15 | 21 | 1.0.0.1 | 110 | TCP | DENY | PASSED |
| 23 | 15 | 15 | 192.168.3.15 | 21 | 192.168.5.2 | 110 | TCP | DENY | PASSED |
| 23 | 16 | 16 | 192.168.3.15 | 21 | 96.84.0.1 | 110 | TCP | DENY | PASSED |
| 23 | 17 | 17 | 192.168.3.15 | 21 | 192.168.5.4 | 110 | TCP | DENY | PASSED |
| 23 | 18 | 18 | 192.168.3.15 | 21 | 255.255.255.254 | 110 | TCP | DENY | PASSED |
| 23 | 19 | 19 | 192.168.3.15 | 21 | 223.211.128.128 | 110 | TCP | DENY | PASSED |
| 23 | 20 | 20 | 192.168.3.15 | 21 | 192.168.5.3 | 0 | TCP | ALLOW | PASSED |
| 23 | 21 | 21 | 192.168.3.15 | 21 | 192.168.5.3 | 65535 | TCP | ALLOW | PASSED |
| 23 | 22 | 22 | 192.168.3.15 | 21 | 192.168.5.3 | 23 | TCP | ALLOW | PASSED |
| 23 | 23 | 23 | 192.168.3.15 | 21 | 192.168.5.3 | 80 | TCP | ALLOW | PASSED |

Table A.46. Conformance Testing - XXIII

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Flag | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|---|
| 23 | 1 | 1 | 192.168.3.81 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 23 | 2 | 2 | 192.168.3.81 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 23 | 3 | 3 | 192.168.5.3 | 21 | 192.168.3.81 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 23 | 4 | 4 | 192.168.3.81 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 23 | 5 | 5 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 23 | 6 | 6 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 23 | 7 | 7 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 23 | 8 | 8 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 23 | 9 | 9 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 23 | 10 | 10 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 23 | 11 | 11 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 23 | 12 | 12 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 23 | 13 | 13 | 192.168.3.15 | 21 | 192.168.5.3 | 742 | RST | TCP | ALLOW | PASSED |
| 23 | 14 | 14 | 192.168.3.15 | 21 | 192.168.5.3 | 742 | FIN | TCP | ALLOW | PASSED |
| 23 | 15 | 15 | 192.168.5.3 | 21 | 192.168.3.15 | 742 | SYN/ACK | TCP | DENY | PASSED |
| 23 | 16 | 16 | 192.168.3.15 | 21 | 192.168.5.3 | 742 | SYN | TCP | ALLOW | PASSED |

## A.24. Experiment XXIV

The 24th experiment is performed for the case, policy allows traffic comes from a subnet with the port 21 and goes to a host via port 110, with the command of "iptables –A FORWARD –p tcp –sip 192.168.3.0/24 –sport 21 –dip 192.168.5.3 –dport 110 –j ACCEPT".

Table A.47. Equivalence Partitioning - XXIV

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|
| 24 | 1 | 1 | 192.168.3.1 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 24 | 2 | 2 | 192.168.3.127 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 24 | 3 | 3 | 192.168.3.254 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 24 | 4 | 4 | 192.168.2.1 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 24 | 5 | 5 | 192.168.2.127 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 24 | 6 | 6 | 192.168.2.254 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 24 | 7 | 7 | 192.168.4.1 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 24 | 8 | 8 | 192.168.4.127 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 24 | 9 | 9 | 192.168.4.254 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 24 | 10 | 10 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 24 | 11 | 11 | 192.168.3.15 | 20 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 24 | 12 | 12 | 192.168.3.15 | 22 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 24 | 13 | 13 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 24 | 14 | 14 | 192.168.3.15 | 21 | 1.0.0.1 | 110 | TCP | DENY | PASSED |
| 24 | 15 | 15 | 192.168.3.15 | 21 | 192.168.5.2 | 110 | TCP | DENY | PASSED |
| 24 | 16 | 16 | 192.168.3.15 | 21 | 96.84.0.1 | 110 | TCP | DENY | PASSED |
| 24 | 17 | 17 | 192.168.3.15 | 21 | 192.168.5.4 | 110 | TCP | DENY | PASSED |
| 24 | 18 | 18 | 192.168.3.15 | 21 | 255.255.255.254 | 110 | TCP | DENY | PASSED |
| 24 | 19 | 19 | 192.168.3.15 | 21 | 223.211.128.128 | 110 | TCP | DENY | PASSED |
| 24 | 20 | 20 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 24 | 21 | 21 | 192.168.3.15 | 21 | 192.168.5.3 | 111 | TCP | DENY | PASSED |
| 24 | 22 | 22 | 192.168.3.15 | 21 | 192.168.5.3 | 109 | TCP | DENY | PASSED |

Table A.48. Conformance Testing - XXIV

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Flag | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|---|
| 24 | 1 | 1 | 192.168.3.59 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 24 | 2 | 2 | 192.168.3.59 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 24 | 3 | 3 | 192.168.5.3 | 21 | 192.168.3.59 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 24 | 4 | 4 | 192.168.3.59 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 24 | 5 | 5 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 24 | 6 | 6 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 24 | 7 | 7 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 24 | 8 | 8 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 24 | 9 | 9 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 24 | 10 | 10 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 24 | 11 | 11 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 24 | 12 | 12 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 24 | 13 | 13 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 24 | 14 | 14 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 24 | 15 | 15 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 24 | 16 | 16 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |

## A.25. Experiment XXV

The 25th experiment is performed for the case, policy allows traffic comes from a host, with the command of "iptables –A FORWARD –p tcp –sip 192.168.3.15 –j ACCEPT".

Table A.49. Equivalence Partitioning - XXV

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|
| 25 | 1 | 1 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 25 | 2 | 2 | 1.0.0.1 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 25 | 3 | 3 | 192.168.3.14 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 25 | 4 | 4 | 96.84.1.7 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 25 | 5 | 5 | 192.168.3.16 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 25 | 6 | 6 | 255.255.255.254 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 25 | 7 | 7 | 223.211.129.135 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 25 | 8 | 8 | 192.168.3.15 | 0 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 25 | 9 | 9 | 192.168.3.15 | 65535 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 25 | 10 | 10 | 192.168.3.15 | 23 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 25 | 11 | 11 | 192.168.3.15 | 80 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 25 | 12 | 12 | 192.168.3.15 | 21 | 10.0.0.1 | 110 | TCP | ALLOW | PASSED |
| 25 | 13 | 13 | 192.168.3.15 | 21 | 172.16.0.1 | 110 | TCP | ALLOW | PASSED |
| 25 | 14 | 14 | 192.168.3.15 | 21 | 192.168.0.1 | 110 | TCP | ALLOW | PASSED |
| 25 | 15 | 15 | 192.168.3.15 | 21 | 127.0.0.1 | 110 | TCP | ALLOW | PASSED |
| 25 | 16 | 16 | 192.168.3.15 | 21 | 88.241.34.41 | 110 | TCP | ALLOW | PASSED |
| 25 | 17 | 17 | 192.168.3.15 | 21 | 215.15.168.23 | 110 | TCP | ALLOW | PASSED |
| 25 | 18 | 18 | 192.168.3.15 | 21 | 192.168.5.3 | 0 | TCP | ALLOW | PASSED |
| 25 | 19 | 19 | 192.168.3.15 | 21 | 192.168.5.3 | 65535 | TCP | ALLOW | PASSED |
| 25 | 20 | 20 | 192.168.3.15 | 21 | 192.168.5.3 | 23 | TCP | ALLOW | PASSED |
| 25 | 21 | 21 | 192.168.3.15 | 21 | 192.168.5.3 | 80 | TCP | ALLOW | PASSED |

Table A.50. Conformance Testing - XXV

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Flag | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|---|
| 25 | 1 | 1 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 25 | 2 | 2 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 25 | 3 | 3 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 25 | 4 | 4 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 25 | 5 | 5 | 192.168.3.15 | 427 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 25 | 6 | 6 | 192.168.3.15 | 427 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 25 | 7 | 7 | 192.168.5.3 | 427 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 25 | 8 | 8 | 192.168.3.15 | 427 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 25 | 9 | 9 | 192.168.3.15 | 21 | 202.212.5.30 | 110 | RST | TCP | ALLOW | PASSED |
| 25 | 10 | 10 | 192.168.3.15 | 21 | 202.212.5.30 | 110 | FIN | TCP | ALLOW | PASSED |
| 25 | 11 | 11 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 25 | 12 | 12 | 192.168.3.15 | 21 | 202.212.5.30 | 110 | SYN | TCP | ALLOW | PASSED |
| 25 | 13 | 13 | 192.168.3.15 | 21 | 192.168.5.3 | 427 | RST | TCP | ALLOW | PASSED |
| 25 | 14 | 14 | 192.168.3.15 | 21 | 192.168.5.3 | 427 | FIN | TCP | ALLOW | PASSED |
| 25 | 15 | 15 | 192.168.5.3 | 21 | 192.168.3.15 | 427 | SYN/ACK | TCP | DENY | PASSED |
| 25 | 16 | 16 | 192.168.3.15 | 21 | 192.168.5.3 | 427 | SYN | TCP | ALLOW | PASSED |

## A.26. Experiment XXVI

The 26th experiment is performed for the case, policy allows traffic comes from a host and goes to via port 110, with the command of "iptables –A FORWARD –p tcp –sip 192.168.3.15 –dport 110 –j ACCEPT".

Table A.51. Equivalence Partitioning - XXVI

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|
| 26 | 1 | 1 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 26 | 2 | 2 | 1.0.0.1 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 26 | 3 | 3 | 192.168.3.14 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 26 | 4 | 4 | 96.84.1.7 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 26 | 5 | 5 | 192.168.3.16 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 26 | 6 | 6 | 255.255.255.254 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 26 | 7 | 7 | 223.211.129.135 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 26 | 8 | 8 | 192.168.3.15 | 0 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 26 | 9 | 9 | 192.168.3.15 | 65535 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 26 | 10 | 10 | 192.168.3.15 | 23 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 26 | 11 | 11 | 192.168.3.15 | 80 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 26 | 12 | 12 | 192.168.3.15 | 21 | 10.0.0.1 | 110 | TCP | ALLOW | PASSED |
| 26 | 13 | 13 | 192.168.3.15 | 21 | 172.16.0.1 | 110 | TCP | ALLOW | PASSED |
| 26 | 14 | 14 | 192.168.3.15 | 21 | 192.168.0.1 | 110 | TCP | ALLOW | PASSED |
| 26 | 15 | 15 | 192.168.3.15 | 21 | 127.0.0.1 | 110 | TCP | ALLOW | PASSED |
| 26 | 16 | 16 | 192.168.3.15 | 21 | 88.241.34.41 | 110 | TCP | ALLOW | PASSED |
| 26 | 17 | 17 | 192.168.3.15 | 21 | 215.15.168.23 | 110 | TCP | ALLOW | PASSED |
| 26 | 18 | 18 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 26 | 19 | 19 | 192.168.3.15 | 21 | 192.168.5.3 | 111 | TCP | DENY | PASSED |
| 26 | 20 | 20 | 192.168.3.15 | 21 | 192.168.5.3 | 109 | TCP | DENY | PASSED |

Table A.52. Conformance Testing - XXVI

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Flag | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|---|
| 26 | 1 | 1 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 26 | 2 | 2 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 26 | 3 | 3 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 26 | 4 | 4 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 26 | 5 | 5 | 192.168.3.15 | 518 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 26 | 6 | 6 | 192.168.3.15 | 518 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 26 | 7 | 7 | 192.168.5.3 | 518 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 26 | 8 | 8 | 192.168.3.15 | 518 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 26 | 9 | 9 | 192.168.3.15 | 21 | 131.107.137.165 | 110 | RST | TCP | ALLOW | PASSED |
| 26 | 10 | 10 | 192.168.3.15 | 21 | 131.107.137.165 | 110 | FIN | TCP | ALLOW | PASSED |
| 26 | 11 | 11 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 26 | 12 | 12 | 192.168.3.15 | 21 | 131.107.137.165 | 110 | SYN | TCP | ALLOW | PASSED |
| 26 | 13 | 13 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 26 | 14 | 14 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 26 | 15 | 15 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 26 | 16 | 16 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |

## A.27. Experiment XXVII

The 27th experiment is performed for the case, policy allows traffic comes from a host and goes to a subnet, with the command of "iptables –A FORWARD –p tcp –sip 192.168.3.15 –dip 192.168.5.0/24 –j ACCEPT".

Table A.53. Equivalence Partitioning - XXVII

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|
| 27 | 1 | 1 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 27 | 2 | 2 | 1.0.0.1 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 27 | 3 | 3 | 192.168.3.14 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 27 | 4 | 4 | 96.84.1.7 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 27 | 5 | 5 | 192.168.3.16 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 27 | 6 | 6 | 255.255.255.254 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 27 | 7 | 7 | 223.211.129.135 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 27 | 8 | 8 | 192.168.3.15 | 0 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 27 | 9 | 9 | 192.168.3.15 | 65535 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 27 | 10 | 10 | 192.168.3.15 | 23 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 27 | 11 | 11 | 192.168.3.15 | 80 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 27 | 12 | 12 | 192.168.3.15 | 21 | 192.168.5.1 | 110 | TCP | ALLOW | PASSED |
| 27 | 13 | 13 | 192.168.3.15 | 21 | 192.168.5.127 | 110 | TCP | ALLOW | PASSED |
| 27 | 14 | 14 | 192.168.3.15 | 21 | 192.168.5.354 | 110 | TCP | ALLOW | PASSED |
| 27 | 15 | 15 | 192.168.3.15 | 21 | 192.168.4.1 | 110 | TCP | DENY | PASSED |
| 27 | 16 | 16 | 192.168.3.15 | 21 | 192.168.4.127 | 110 | TCP | DENY | PASSED |
| 27 | 17 | 17 | 192.168.3.15 | 21 | 192.168.4.254 | 110 | TCP | DENY | PASSED |
| 27 | 18 | 18 | 192.168.3.15 | 21 | 192.168.6.1 | 110 | TCP | DENY | PASSED |
| 27 | 19 | 19 | 192.168.3.15 | 21 | 192.168.6.127 | 110 | TCP | DENY | PASSED |
| 27 | 20 | 20 | 192.168.3.15 | 21 | 192.168.6.254 | 110 | TCP | DENY | PASSED |
| 27 | 21 | 21 | 192.168.3.15 | 21 | 192.168.5.3 | 0 | TCP | ALLOW | PASSED |
| 27 | 22 | 22 | 192.168.3.15 | 21 | 192.168.5.3 | 65535 | TCP | ALLOW | PASSED |
| 27 | 23 | 23 | 192.168.3.15 | 21 | 192.168.5.3 | 23 | TCP | ALLOW | PASSED |
| 27 | 24 | 24 | 192.168.3.15 | 21 | 192.168.5.3 | 80 | TCP | ALLOW | PASSED |

Table A.54. Conformance Testing - XXVII

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Flag | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|---|
| 27 | 1 | 1 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 27 | 2 | 2 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 27 | 3 | 3 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 27 | 4 | 4 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 27 | 5 | 5 | 192.168.3.15 | 438 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 27 | 6 | 6 | 192.168.3.15 | 438 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 27 | 7 | 7 | 192.168.5.3 | 438 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 27 | 8 | 8 | 192.168.3.15 | 438 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 27 | 9 | 9 | 192.168.3.15 | 21 | 192.168.5.365 | 110 | RST | TCP | ALLOW | PASSED |
| 27 | 10 | 10 | 192.168.3.15 | 21 | 192.168.5.365 | 110 | FIN | TCP | ALLOW | PASSED |
| 27 | 11 | 11 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 27 | 12 | 12 | 192.168.3.15 | 21 | 192.168.5.365 | 110 | SYN | TCP | ALLOW | PASSED |
| 27 | 13 | 13 | 192.168.3.15 | 21 | 192.168.5.3 | 438 | RST | TCP | ALLOW | PASSED |
| 27 | 14 | 14 | 192.168.3.15 | 21 | 192.168.5.3 | 438 | FIN | TCP | ALLOW | PASSED |
| 27 | 15 | 15 | 192.168.5.3 | 21 | 192.168.3.15 | 438 | SYN/ACK | TCP | DENY | PASSED |
| 27 | 16 | 16 | 192.168.3.15 | 21 | 192.168.5.3 | 438 | SYN | TCP | ALLOW | PASSED |

## A.28. Experiment XXVIII

The 28th experiment is performed for the case, policy allows traffic comes from a host and goes to a subnet via port 110, with the command of "iptables –A FORWARD –p tcp –sip 192.168.3.15 –dip 192.168.5.0/24 –dport 110 –j ACCEPT".

Table A.55. Equivalence Partitioning - XXVIII

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|
| 28 | 1 | 1 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 28 | 2 | 2 | 1.0.0.1 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 28 | 3 | 3 | 192.168.3.14 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 28 | 4 | 4 | 96.84.1.7 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 28 | 5 | 5 | 192.168.3.16 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 28 | 6 | 6 | 255.255.255.254 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 28 | 7 | 7 | 223.211.129.135 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 28 | 8 | 8 | 192.168.3.15 | 0 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 28 | 9 | 9 | 192.168.3.15 | 65535 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 28 | 10 | 10 | 192.168.3.15 | 23 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 28 | 11 | 11 | 192.168.3.15 | 80 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 28 | 12 | 12 | 192.168.3.15 | 21 | 192.168.5.1 | 110 | TCP | ALLOW | PASSED |
| 28 | 13 | 13 | 192.168.3.15 | 21 | 192.168.5.127 | 110 | TCP | ALLOW | PASSED |
| 28 | 14 | 14 | 192.168.3.15 | 21 | 192.168.5.354 | 110 | TCP | ALLOW | PASSED |
| 28 | 15 | 15 | 192.168.3.15 | 21 | 192.168.4.1 | 110 | TCP | DENY | PASSED |
| 28 | 16 | 16 | 192.168.3.15 | 21 | 192.168.4.127 | 110 | TCP | DENY | PASSED |
| 28 | 17 | 17 | 192.168.3.15 | 21 | 192.168.4.254 | 110 | TCP | DENY | PASSED |
| 28 | 18 | 18 | 192.168.3.15 | 21 | 192.168.6.1 | 110 | TCP | DENY | PASSED |
| 28 | 19 | 19 | 192.168.3.15 | 21 | 192.168.6.127 | 110 | TCP | DENY | PASSED |
| 28 | 20 | 20 | 192.168.3.15 | 21 | 192.168.6.254 | 110 | TCP | DENY | PASSED |
| 28 | 21 | 21 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 28 | 22 | 22 | 192.168.3.15 | 21 | 192.168.5.3 | 111 | TCP | DENY | PASSED |
| 28 | 23 | 23 | 192.168.3.15 | 21 | 192.168.5.3 | 109 | TCP | DENY | PASSED |

Table A.56. Conformance Testing - XXVIII

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Flag | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|---|
| 28 | 1 | 1 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 28 | 2 | 2 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 28 | 3 | 3 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 28 | 4 | 4 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 28 | 5 | 5 | 192.168.3.15 | 490 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 28 | 6 | 6 | 192.168.3.15 | 490 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 28 | 7 | 7 | 192.168.5.3 | 490 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 28 | 8 | 8 | 192.168.3.15 | 490 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 28 | 9 | 9 | 192.168.3.15 | 21 | 192.168.5.308 | 110 | RST | TCP | ALLOW | PASSED |
| 28 | 10 | 10 | 192.168.3.15 | 21 | 192.168.5.308 | 110 | FIN | TCP | ALLOW | PASSED |
| 28 | 11 | 11 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 28 | 12 | 12 | 192.168.3.15 | 21 | 192.168.5.308 | 110 | SYN | TCP | ALLOW | PASSED |
| 28 | 13 | 13 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 28 | 14 | 14 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 28 | 15 | 15 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 28 | 16 | 16 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |

## A.29. Experiment XXIX

The 29th experiment is performed for the case, policy allows traffic comes from a host and goes to a host, with the command of "iptables –A FORWARD –p tcp –sip 192.168.3.15 –dip 192.168.5.3 –j ACCEPT".

Table A.57. Equivalence Partitioning - XXIX

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|
| 29 | 1 | 1 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 29 | 2 | 2 | 1.0.0.1 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 29 | 3 | 3 | 192.168.3.14 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 29 | 4 | 4 | 96.84.1.7 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 29 | 5 | 5 | 192.168.3.16 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 29 | 6 | 6 | 255.255.255.254 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 29 | 7 | 7 | 223.211.129.135 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 29 | 8 | 8 | 192.168.3.15 | 0 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 29 | 9 | 9 | 192.168.3.15 | 65535 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 29 | 10 | 10 | 192.168.3.15 | 23 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 29 | 11 | 11 | 192.168.3.15 | 80 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 29 | 12 | 12 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 29 | 13 | 13 | 192.168.3.15 | 21 | 1.0.0.1 | 110 | TCP | DENY | PASSED |
| 29 | 14 | 14 | 192.168.3.15 | 21 | 192.168.5.2 | 110 | TCP | DENY | PASSED |
| 29 | 15 | 15 | 192.168.3.15 | 21 | 96.84.0.1 | 110 | TCP | DENY | PASSED |
| 29 | 16 | 16 | 192.168.3.15 | 21 | 192.168.5.4 | 110 | TCP | DENY | PASSED |
| 29 | 17 | 17 | 192.168.3.15 | 21 | 255.255.255.254 | 110 | TCP | DENY | PASSED |
| 29 | 18 | 18 | 192.168.3.15 | 21 | 223.211.128.128 | 110 | TCP | DENY | PASSED |
| 29 | 19 | 19 | 192.168.3.15 | 21 | 192.168.5.3 | 0 | TCP | ALLOW | PASSED |
| 29 | 20 | 20 | 192.168.3.15 | 21 | 192.168.5.3 | 65535 | TCP | ALLOW | PASSED |
| 29 | 21 | 21 | 192.168.3.15 | 21 | 192.168.5.3 | 23 | TCP | ALLOW | PASSED |
| 29 | 22 | 22 | 192.168.3.15 | 21 | 192.168.5.3 | 80 | TCP | ALLOW | PASSED |

Table A.58. Conformance Testing - XXIX

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Flag | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|---|
| 29 | 1 | 1 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 29 | 2 | 2 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 29 | 3 | 3 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 29 | 4 | 4 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 29 | 5 | 5 | 192.168.3.15 | 885 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 29 | 6 | 6 | 192.168.3.15 | 885 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 29 | 7 | 7 | 192.168.5.3 | 885 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 29 | 8 | 8 | 192.168.3.15 | 885 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 29 | 9 | 9 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 29 | 10 | 10 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 29 | 11 | 11 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 29 | 12 | 12 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 29 | 13 | 13 | 192.168.3.15 | 21 | 192.168.5.3 | 885 | RST | TCP | ALLOW | PASSED |
| 29 | 14 | 14 | 192.168.3.15 | 21 | 192.168.5.3 | 885 | FIN | TCP | ALLOW | PASSED |
| 29 | 15 | 15 | 192.168.5.3 | 21 | 192.168.3.15 | 885 | SYN/ACK | TCP | DENY | PASSED |
| 29 | 16 | 16 | 192.168.3.15 | 21 | 192.168.5.3 | 885 | SYN | TCP | ALLOW | PASSED |

## A.30. Experiment XXX

The 30th experiment is performed for the case, policy allows traffic comes from a host and goes to a host via port 110, with the command of "iptables –A FORWARD –p tcp –sip 192.168.3.15 –dip 192.168.5.3 –dport 110 –j ACCEPT".

Table A.59. Equivalence Partitioning - XXX

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|
| 30 | 1 | 1 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 30 | 2 | 2 | 1.0.0.1 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 30 | 3 | 3 | 192.168.3.14 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 30 | 4 | 4 | 96.84.1.7 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 30 | 5 | 5 | 192.168.3.16 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 30 | 6 | 6 | 255.255.255.254 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 30 | 7 | 7 | 223.211.129.135 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 30 | 8 | 8 | 192.168.3.15 | 0 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 30 | 9 | 9 | 192.168.3.15 | 65535 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 30 | 10 | 10 | 192.168.3.15 | 23 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 30 | 11 | 11 | 192.168.3.15 | 80 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 30 | 12 | 12 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 30 | 13 | 13 | 192.168.3.15 | 21 | 1.0.0.1 | 110 | TCP | DENY | PASSED |
| 30 | 14 | 14 | 192.168.3.15 | 21 | 192.168.5.2 | 110 | TCP | DENY | PASSED |
| 30 | 15 | 15 | 192.168.3.15 | 21 | 96.84.0.1 | 110 | TCP | DENY | PASSED |
| 30 | 16 | 16 | 192.168.3.15 | 21 | 192.168.5.4 | 110 | TCP | DENY | PASSED |
| 30 | 17 | 17 | 192.168.3.15 | 21 | 255.255.255.254 | 110 | TCP | DENY | PASSED |
| 30 | 18 | 18 | 192.168.3.15 | 21 | 223.211.128.128 | 110 | TCP | DENY | PASSED |
| 30 | 19 | 19 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 30 | 20 | 20 | 192.168.3.15 | 21 | 192.168.5.3 | 111 | TCP | DENY | PASSED |
| 30 | 21 | 21 | 192.168.3.15 | 21 | 192.168.5.3 | 109 | TCP | DENY | PASSED |

Table A.60. Conformance Testing - XXX

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Flag | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|---|
| 30 | 1 | 1 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 30 | 2 | 2 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 30 | 3 | 3 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 30 | 4 | 4 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 30 | 5 | 5 | 192.168.3.15 | 154 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 30 | 6 | 6 | 192.168.3.15 | 154 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 30 | 7 | 7 | 192.168.5.3 | 154 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 30 | 8 | 8 | 192.168.3.15 | 154 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 30 | 9 | 9 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 30 | 10 | 10 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 30 | 11 | 11 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 30 | 12 | 12 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 30 | 13 | 13 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 30 | 14 | 14 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 30 | 15 | 15 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 30 | 16 | 16 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |

## A.31. Experiment XXXI

The 31th experiment is performed for the case, policy allows traffic comes from a host via port 21, with the command of "iptables –A FORWARD –p tcp –sip 192.168.3.15 –sport 21 –j ACCEPT".

Table A.61. Equivalence Partitioning - XXXI

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|
| 31 | 1 | 1 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 31 | 2 | 2 | 1.0.0.1 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 31 | 3 | 3 | 192.168.3.14 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 31 | 4 | 4 | 96.84.1.7 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 31 | 5 | 5 | 192.168.3.16 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 31 | 6 | 6 | 255.255.255.254 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 31 | 7 | 7 | 223.211.129.135 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 31 | 8 | 8 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 31 | 9 | 9 | 192.168.3.15 | 20 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 31 | 10 | 10 | 192.168.3.15 | 22 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 31 | 11 | 11 | 192.168.3.15 | 21 | 10.0.0.1 | 110 | TCP | ALLOW | PASSED |
| 31 | 12 | 12 | 192.168.3.15 | 21 | 172.16.0.1 | 110 | TCP | ALLOW | PASSED |
| 31 | 13 | 13 | 192.168.3.15 | 21 | 192.168.0.1 | 110 | TCP | ALLOW | PASSED |
| 31 | 14 | 14 | 192.168.3.15 | 21 | 127.0.0.1 | 110 | TCP | ALLOW | PASSED |
| 31 | 15 | 15 | 192.168.3.15 | 21 | 88.241.34.41 | 110 | TCP | ALLOW | PASSED |
| 31 | 16 | 16 | 192.168.3.15 | 21 | 215.15.168.23 | 110 | TCP | ALLOW | PASSED |
| 31 | 17 | 17 | 192.168.3.15 | 21 | 192.168.5.3 | 0 | TCP | ALLOW | PASSED |
| 31 | 18 | 18 | 192.168.3.15 | 21 | 192.168.5.3 | 65535 | TCP | ALLOW | PASSED |
| 31 | 19 | 19 | 192.168.3.15 | 21 | 192.168.5.3 | 23 | TCP | ALLOW | PASSED |
| 31 | 20 | 20 | 192.168.3.15 | 21 | 192.168.5.3 | 80 | TCP | ALLOW | PASSED |

Table A.62. Conformance Testing - XXX

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Flag | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|---|
| 31 | 1 | 1 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 31 | 2 | 2 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 31 | 3 | 3 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 31 | 4 | 4 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 31 | 5 | 5 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 31 | 6 | 6 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 31 | 7 | 7 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 31 | 8 | 8 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 31 | 9 | 9 | 192.168.3.15 | 21 | 131.107.137.165 | 110 | RST | TCP | ALLOW | PASSED |
| 31 | 10 | 10 | 192.168.3.15 | 21 | 131.107.137.165 | 110 | FIN | TCP | ALLOW | PASSED |
| 31 | 11 | 11 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 31 | 12 | 12 | 192.168.3.15 | 21 | 131.107.137.165 | 110 | SYN | TCP | ALLOW | PASSED |
| 31 | 13 | 13 | 192.168.3.15 | 21 | 192.168.5.3 | 328 | RST | TCP | ALLOW | PASSED |
| 31 | 14 | 14 | 192.168.3.15 | 21 | 192.168.5.3 | 328 | FIN | TCP | ALLOW | PASSED |
| 31 | 15 | 15 | 192.168.5.3 | 21 | 192.168.3.15 | 328 | SYN/ACK | TCP | DENY | PASSED |
| 31 | 16 | 16 | 192.168.3.15 | 21 | 192.168.5.3 | 328 | SYN | TCP | ALLOW | PASSED |

## A.32. Experiment XXXII

The 32th experiment is performed for the case, policy allows traffic comes from a host via port 21 and goes via port 110, with the command of "iptables –A FORWARD –p tcp –sip 192.168.3.15 –sport 21 –dport 110 –j ACCEPT".

Table A.63. Equivalence Partitioning - XXXII

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|
| 32 | 1 | 1 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 32 | 2 | 2 | 1.0.0.1 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 32 | 3 | 3 | 192.168.3.14 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 32 | 4 | 4 | 96.84.1.7 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 32 | 5 | 5 | 192.168.3.16 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 32 | 6 | 6 | 255.255.255.254 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 32 | 7 | 7 | 223.211.129.135 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 32 | 8 | 8 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 32 | 9 | 9 | 192.168.3.15 | 20 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 32 | 10 | 10 | 192.168.3.15 | 22 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 32 | 11 | 11 | 192.168.3.15 | 21 | 10.0.0.1 | 110 | TCP | ALLOW | PASSED |
| 32 | 12 | 12 | 192.168.3.15 | 21 | 172.16.0.1 | 110 | TCP | ALLOW | PASSED |
| 32 | 13 | 13 | 192.168.3.15 | 21 | 192.168.0.1 | 110 | TCP | ALLOW | PASSED |
| 32 | 14 | 14 | 192.168.3.15 | 21 | 127.0.0.1 | 110 | TCP | ALLOW | PASSED |
| 32 | 15 | 15 | 192.168.3.15 | 21 | 88.241.34.41 | 110 | TCP | ALLOW | PASSED |
| 32 | 16 | 16 | 192.168.3.15 | 21 | 215.15.168.23 | 110 | TCP | ALLOW | PASSED |
| 32 | 17 | 17 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 32 | 18 | 18 | 192.168.3.15 | 21 | 192.168.5.3 | 111 | TCP | DENY | PASSED |
| 32 | 19 | 19 | 192.168.3.15 | 21 | 192.168.5.3 | 109 | TCP | DENY | PASSED |

Table A.64. Conformance Testing - XXXII

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Flag | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|---|
| 32 | 1 | 1 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 32 | 2 | 2 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 32 | 3 | 3 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 32 | 4 | 4 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 32 | 5 | 5 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 32 | 6 | 6 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 32 | 7 | 7 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 32 | 8 | 8 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 32 | 9 | 9 | 192.168.3.15 | 21 | 216.239.33.96 | 110 | RST | TCP | ALLOW | PASSED |
| 32 | 10 | 10 | 192.168.3.15 | 21 | 216.239.33.96 | 110 | FIN | TCP | ALLOW | PASSED |
| 32 | 11 | 11 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 32 | 12 | 12 | 192.168.3.15 | 21 | 216.239.33.96 | 110 | SYN | TCP | ALLOW | PASSED |
| 32 | 13 | 13 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 32 | 14 | 14 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 32 | 15 | 15 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 32 | 16 | 16 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |

## A.33. Experiment XXXIII

The 33th experiment is performed for the case, policy allows traffic comes from a host via port 21, and goes to a subnet with the command of "iptables –A FORWARD –p tcp –sip 192.168.3.15 –sport 21 –dip 192.168.5.0/24 –j ACCEPT".

Table A.65. Equivalence Partitioning - XXXIII

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|
| 33 | 1 | 1 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 33 | 2 | 2 | 1.0.0.1 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 33 | 3 | 3 | 192.168.3.14 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 33 | 4 | 4 | 96.84.1.7 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 33 | 5 | 5 | 192.168.3.16 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 33 | 6 | 6 | 255.255.255.254 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 33 | 7 | 7 | 223.211.129.135 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 33 | 8 | 8 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 33 | 9 | 9 | 192.168.3.15 | 20 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 33 | 10 | 10 | 192.168.3.15 | 22 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 33 | 11 | 11 | 192.168.3.15 | 21 | 192.168.5.1 | 110 | TCP | ALLOW | PASSED |
| 33 | 12 | 12 | 192.168.3.15 | 21 | 192.168.5.127 | 110 | TCP | ALLOW | PASSED |
| 33 | 13 | 13 | 192.168.3.15 | 21 | 192.168.5.354 | 110 | TCP | ALLOW | PASSED |
| 33 | 14 | 14 | 192.168.3.15 | 21 | 192.168.4.1 | 110 | TCP | DENY | PASSED |
| 33 | 15 | 15 | 192.168.3.15 | 21 | 192.168.4.127 | 110 | TCP | DENY | PASSED |
| 33 | 16 | 16 | 192.168.3.15 | 21 | 192.168.4.254 | 110 | TCP | DENY | PASSED |
| 33 | 17 | 17 | 192.168.3.15 | 21 | 192.168.6.1 | 110 | TCP | DENY | PASSED |
| 33 | 18 | 18 | 192.168.3.15 | 21 | 192.168.6.127 | 110 | TCP | DENY | PASSED |
| 33 | 19 | 19 | 192.168.3.15 | 21 | 192.168.6.254 | 110 | TCP | DENY | PASSED |
| 33 | 20 | 20 | 192.168.3.15 | 21 | 192.168.5.3 | 0 | TCP | ALLOW | PASSED |
| 33 | 21 | 21 | 192.168.3.15 | 21 | 192.168.5.3 | 65535 | TCP | ALLOW | PASSED |
| 33 | 22 | 22 | 192.168.3.15 | 21 | 192.168.5.3 | 23 | TCP | ALLOW | PASSED |
| 33 | 23 | 23 | 192.168.3.15 | 21 | 192.168.5.3 | 80 | TCP | ALLOW | PASSED |

Table A.66. Conformance Testing - XXXIII

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Flag | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|---|
| 33 | 1 | 1 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 33 | 2 | 2 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 33 | 3 | 3 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 33 | 4 | 4 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 33 | 5 | 5 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 33 | 6 | 6 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 33 | 7 | 7 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 33 | 8 | 8 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 33 | 9 | 9 | 192.168.3.15 | 21 | 192.168.2.96 | 110 | RST | TCP | ALLOW | PASSED |
| 33 | 10 | 10 | 192.168.3.15 | 21 | 192.168.2.96 | 110 | FIN | TCP | ALLOW | PASSED |
| 33 | 11 | 11 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 33 | 12 | 12 | 192.168.3.15 | 21 | 192.168.2.96 | 110 | SYN | TCP | ALLOW | PASSED |
| 33 | 13 | 13 | 192.168.3.15 | 21 | 192.168.5.3 | 314 | RST | TCP | ALLOW | PASSED |
| 33 | 14 | 14 | 192.168.3.15 | 21 | 192.168.5.3 | 314 | FIN | TCP | ALLOW | PASSED |
| 33 | 15 | 15 | 192.168.5.3 | 21 | 192.168.3.15 | 314 | SYN/ACK | TCP | DENY | PASSED |
| 33 | 16 | 16 | 192.168.3.15 | 21 | 192.168.5.3 | 314 | SYN | TCP | ALLOW | PASSED |

## A.34. Experiment XXXIV

The 34th experiment is performed for the case, policy allows traffic comes from a host via port 21, and goes to a subnet via port 110, with the command of "iptables –A FORWARD –p tcp –sip 192.168.3.15 –sport 21 –dip 192.168.5.0/24 –dport 110 –j ACCEPT".

Table A.67. Equivalence Partitioning - XXXIV

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|
| 34 | 1 | 1 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 34 | 2 | 2 | 1.0.0.1 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 34 | 3 | 3 | 192.168.3.14 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 34 | 4 | 4 | 96.84.1.7 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 34 | 5 | 5 | 192.168.3.16 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 34 | 6 | 6 | 255.255.255.254 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 34 | 7 | 7 | 223.211.129.135 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 34 | 8 | 8 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 34 | 9 | 9 | 192.168.3.15 | 20 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 34 | 10 | 10 | 192.168.3.15 | 22 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 34 | 11 | 11 | 192.168.3.15 | 21 | 192.168.5.1 | 110 | TCP | ALLOW | PASSED |
| 34 | 12 | 12 | 192.168.3.15 | 21 | 192.168.5.127 | 110 | TCP | ALLOW | PASSED |
| 34 | 13 | 13 | 192.168.3.15 | 21 | 192.168.5.354 | 110 | TCP | ALLOW | PASSED |
| 34 | 14 | 14 | 192.168.3.15 | 21 | 192.168.4.1 | 110 | TCP | DENY | PASSED |
| 34 | 15 | 15 | 192.168.3.15 | 21 | 192.168.4.127 | 110 | TCP | DENY | PASSED |
| 34 | 16 | 16 | 192.168.3.15 | 21 | 192.168.4.254 | 110 | TCP | DENY | PASSED |
| 34 | 17 | 17 | 192.168.3.15 | 21 | 192.168.6.1 | 110 | TCP | DENY | PASSED |
| 34 | 18 | 18 | 192.168.3.15 | 21 | 192.168.6.127 | 110 | TCP | DENY | PASSED |
| 34 | 19 | 19 | 192.168.3.15 | 21 | 192.168.6.254 | 110 | TCP | DENY | PASSED |
| 34 | 20 | 20 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 34 | 21 | 21 | 192.168.3.15 | 21 | 192.168.5.3 | 111 | TCP | DENY | PASSED |
| 34 | 22 | 22 | 192.168.3.15 | 21 | 192.168.5.3 | 109 | TCP | DENY | PASSED |

Table A.68. Conformance Testing - XXXIV

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Flag | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|---|
| 34 | 1 | 1 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 34 | 2 | 2 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 34 | 3 | 3 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 34 | 4 | 4 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 34 | 5 | 5 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 34 | 6 | 6 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 34 | 7 | 7 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 34 | 8 | 8 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 34 | 9 | 9 | 192.168.3.15 | 21 | 192.168.5.359 | 110 | RST | TCP | ALLOW | PASSED |
| 34 | 10 | 10 | 192.168.3.15 | 21 | 192.168.5.359 | 110 | FIN | TCP | ALLOW | PASSED |
| 34 | 11 | 11 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 34 | 12 | 12 | 192.168.3.15 | 21 | 192.168.5.359 | 110 | SYN | TCP | ALLOW | PASSED |
| 34 | 13 | 13 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 34 | 14 | 14 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 34 | 15 | 15 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 34 | 16 | 16 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |

## A.35. Experiment XXXV

The 35th experiment is performed for the case, policy allows traffic comes from a host via port 21, and goes to a host, with the command of "iptables –A FORWARD –p tcp –sip 192.168.3.15 –sport 21 –dip 192.168.5.3 –j ACCEPT".

Table A.69. Equivalence Partitioning - XXXV

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|
| 35 | 1 | 1 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 35 | 2 | 2 | 1.0.0.1 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 35 | 3 | 3 | 192.168.3.14 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 35 | 4 | 4 | 96.84.1.7 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 35 | 5 | 5 | 192.168.3.16 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 35 | 6 | 6 | 255.255.255.254 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 35 | 7 | 7 | 223.211.129.135 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 35 | 8 | 8 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 35 | 9 | 9 | 192.168.3.15 | 20 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 35 | 10 | 10 | 192.168.3.15 | 22 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 35 | 11 | 11 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 35 | 12 | 12 | 192.168.3.15 | 21 | 1.0.0.1 | 110 | TCP | DENY | PASSED |
| 35 | 13 | 13 | 192.168.3.15 | 21 | 192.168.5.2 | 110 | TCP | DENY | PASSED |
| 35 | 14 | 14 | 192.168.3.15 | 21 | 96.84.0.1 | 110 | TCP | DENY | PASSED |
| 35 | 15 | 15 | 192.168.3.15 | 21 | 192.168.5.4 | 110 | TCP | DENY | PASSED |
| 35 | 16 | 16 | 192.168.3.15 | 21 | 255.255.255.254 | 110 | TCP | DENY | PASSED |
| 35 | 17 | 17 | 192.168.3.15 | 21 | 223.211.128.128 | 110 | TCP | DENY | PASSED |
| 35 | 18 | 18 | 192.168.3.15 | 21 | 192.168.5.3 | 0 | TCP | ALLOW | PASSED |
| 35 | 19 | 19 | 192.168.3.15 | 21 | 192.168.5.3 | 65535 | TCP | ALLOW | PASSED |
| 35 | 20 | 20 | 192.168.3.15 | 21 | 192.168.5.3 | 23 | TCP | ALLOW | PASSED |
| 35 | 21 | 21 | 192.168.3.15 | 21 | 192.168.5.3 | 80 | TCP | ALLOW | PASSED |

Table A.70. Conformance Testing - XXXV

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Flag | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|---|
| 35 | 1 | 1 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 35 | 2 | 2 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 35 | 3 | 3 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 35 | 4 | 4 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 35 | 5 | 5 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 35 | 6 | 6 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 35 | 7 | 7 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 35 | 8 | 8 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 35 | 9 | 9 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 35 | 10 | 10 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 35 | 11 | 11 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 35 | 12 | 12 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 35 | 13 | 13 | 192.168.3.15 | 21 | 192.168.5.3 | 311 | RST | TCP | ALLOW | PASSED |
| 35 | 14 | 14 | 192.168.3.15 | 21 | 192.168.5.3 | 311 | FIN | TCP | ALLOW | PASSED |
| 35 | 15 | 15 | 192.168.5.3 | 21 | 192.168.3.15 | 311 | SYN/ACK | TCP | DENY | PASSED |
| 35 | 16 | 16 | 192.168.3.15 | 21 | 192.168.5.3 | 311 | SYN | TCP | ALLOW | PASSED |

## A.36. Experiment XXXVI

The 36th experiment is performed for the case, policy allows traffic comes from a host via port 21, and goes to a host via port 110, with the command of "iptables –A FORWARD –p tcp –sip 192.168.3.15 –sport 21 –dip 192.168.5.3 –dport 110 –j ACCEPT".

Table A.71. Equivalence Partitioning - XXXVI

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|
| 36 | 1 | 1 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 36 | 2 | 2 | 1.0.0.1 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 36 | 3 | 3 | 192.168.3.14 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 36 | 4 | 4 | 96.84.1.7 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 36 | 5 | 5 | 192.168.3.16 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 36 | 6 | 6 | 255.255.255.254 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 36 | 7 | 7 | 223.211.129.135 | 21 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 36 | 8 | 8 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 36 | 9 | 9 | 192.168.3.15 | 20 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 36 | 10 | 10 | 192.168.3.15 | 22 | 192.168.5.3 | 110 | TCP | DENY | PASSED |
| 36 | 11 | 11 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 36 | 12 | 12 | 192.168.3.15 | 21 | 1.0.0.1 | 110 | TCP | DENY | PASSED |
| 36 | 13 | 13 | 192.168.3.15 | 21 | 192.168.5.2 | 110 | TCP | DENY | PASSED |
| 36 | 14 | 14 | 192.168.3.15 | 21 | 96.84.0.1 | 110 | TCP | DENY | PASSED |
| 36 | 15 | 15 | 192.168.3.15 | 21 | 192.168.5.4 | 110 | TCP | DENY | PASSED |
| 36 | 16 | 16 | 192.168.3.15 | 21 | 255.255.255.254 | 110 | TCP | DENY | PASSED |
| 36 | 17 | 17 | 192.168.3.15 | 21 | 223.211.128.128 | 110 | TCP | DENY | PASSED |
| 36 | 18 | 18 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | TCP | ALLOW | PASSED |
| 36 | 19 | 19 | 192.168.3.15 | 21 | 192.168.5.3 | 111 | TCP | DENY | PASSED |
| 36 | 20 | 20 | 192.168.3.15 | 21 | 192.168.5.3 | 109 | TCP | DENY | PASSED |

Table A.72. Conformance Testing - XXXVI

| Firewall Rule No | Test Case No | ID | Source IP | Source Port | Destination IP | Destination Port | Flag | Protocol | Action on Packet | Test Results |
|---|---|---|---|---|---|---|---|---|---|---|
| 36 | 1 | 1 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 36 | 2 | 2 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 36 | 3 | 3 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 36 | 4 | 4 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 36 | 5 | 5 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 36 | 6 | 6 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 36 | 7 | 7 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 36 | 8 | 8 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 36 | 9 | 9 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 36 | 10 | 10 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 36 | 11 | 11 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 36 | 12 | 12 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |
| 36 | 13 | 13 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | RST | TCP | ALLOW | PASSED |
| 36 | 14 | 14 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | FIN | TCP | ALLOW | PASSED |
| 36 | 15 | 15 | 192.168.5.3 | 21 | 192.168.3.15 | 110 | SYN/ACK | TCP | DENY | PASSED |
| 36 | 16 | 16 | 192.168.3.15 | 21 | 192.168.5.3 | 110 | SYN | TCP | ALLOW | PASSED |