Abstracting network policies

Abstracting Network Policies

by

Ahmed N. Abubakar

A Doctoral Thesis

Submitted in partial fulfilment
of the requirements for the award of

Doctor of Philosophy
of
Loughborough University

2nd April 2018

Abstract

Almost every human activity in recent years relies either directly or indirectly on the smooth and efficient operation of the Internet. The Internet is an interconnection of multiple autonomous networks that work based on agreed upon policies between various institutions across the world. The network policies guiding an institution's computer infrastructure both internally (such as firewall relationships) and externally (such as routing relationships) are developed by a diverse group of lawyers, accountants, network administrators, managers amongst others. Network policies developed by this group of individuals are usually done on a white-board in a graph-like format. It is however the responsibility of network administrators to translate and configure the various network policies that have been agreed upon. The configuration of these network policies are generally done on physical devices such as routers, domain name servers, firewalls and other middle boxes. The manual configuration process of such network policies is known to be tedious, time consuming and prone to human error which can lead to various network anomalies in the configuration commands. In recent years, many research projects and corporate organisations have to some level abstracted the network management process with emphasis on network devices (such as Cisco VIRL) or individual network policies (such as Propane).

The research conducted in this thesis will be introducing a range of network management process abstraction models in a more holistic manner. The abstraction models proposed in this research are expressed in a graph-like format and ensures the network management process is much easier and faster to deploy for network administrators compared to the manual process. The research work carried out in this thesis builds on current research in the field of network automation. There are two levels of abstractions that are discussed in this thesis - network policy abstractions and network layout abstractions. The abstract specifications that have been implemented in this research work include: inter domain routing (BGP) relationships, firewall relationships and cyber security competitions. A system called Network Policy Abstraction System (NePAS) has been developed in order to evaluate the abstraction models proposed in this research. NePAS is composed of four phases: policy intention; network layout; anomaly resolution and compilation. Experiments are conducted in order to showcase our abstraction models' flexibility, scalability and their various strengths and weaknesses.

Acknowledgements

I am forever indebted to my academic supervisor, Dr Iain Phillips, you have been a tremendous mentor for me. I would like to thank you for your guidance, enthusiasm, unrelenting support, encouraging my research and for allowing me to grow as a research scientist. Your advice on both research as well as on discussions on life in general have been priceless. My name may be alone on the front cover of this thesis but I could not have completed this thesis without you.

I would also like to thank Professor Olaf Maennel without whom I probably would not be able to write this thesis. Our discussions have helped me develop my research and writing skills before he left Loughborough University. Many thanks to my colleagues and academics at Loughborough University, Awos Ali, Mehdi Mousavi, Dr Vasili Michopoulos and Dr Lin Guan for providing a friendly and productive discussions over the years. As for all the other members of Loughborough University particularly the networks group whom I may have forgotten to mention, I would like to say thank you.

I would like to thank my life partner, Sa'adatu Jamila Mamman Daura, for her unremitting encouragement. Put simply, I have never met anyone who believes in me more. Thank you for making me more than I am.

I especially thank my mom, dad, siblings and in-laws. My hard-working parents have sacrificed their lives for our family and provided unconditional love and care. I love them so much, and I would not have made it this far without them. I also thank my friends (too many to list here but you know who you are!) for providing support and friendship that I needed.

Lastly, I thank the Petroleum Technology Development Fund (PTDF) which provided full funding for this work.

Contents

List of Figures

List of Tables

Listings

Chapter 1

Introduction

The corporate setting of various institutions, both public and private, rely on information systems for their operational activities. These information systems are usually dispersed across large geographical regions and connected via the Internet. The Internet is an interconnection of multiple autonomous networks connected using various network policies. These networks that make up the Internet are implemented using various network devices (such as firewalls, routers, servers, etc.) and governed by an array of network policies. These network policies guide how the various users of such networks interconnect, access resources or secure critical data amongst other functionalities. Computer network policies such as routing relationships, firewall rules and many other network policies are typically proposed and designed based on a network-wide policy developed by a group of network administrators, lawyers, accountants, managers, etc. At the onset, this group of individuals first design a whiteboard high-level policy to have an understanding of how the proposed network will operate once deployed. The design of these proposed high-level policies are usually done on the white board in a graph-like format. This approach is used to aide understanding of proposed settings for routing, security and many other network policy requirements between the various devices and users of a proposed network. Once the network-wide high-level policies have been agreed upon, the various network policy intentions have to be translated into low level platform-specific configurations. These policies are usually statically configured in low-level configuration languages or triggered post deployment due to changes in the network intention such as the need to change firewall rule relations due to prevailing circumstance. For example, an institution might want to deny its employees from accessing a particular website after monitoring its network traffic and observing a decline in employee productivity over a given period. These sets of platform specific configurations are required to be implemented on network devices such as routers, firewalls, switches and other middle-boxes. A number of issues arise during the manual configuration of low level vendor specific network devices that are used to implement various network policies needed for the connectivity, security and other networking

requirements. These issues are discussed in the following section.

1.1 Problem Statement

Researchers and/or network administrators design and deploy experimental networks in a virtual environment so as to run tests either as part of measurement project or ensure the suitability of such policies before been deployed on physical devices. The task of manually configuring such experimental networks takes hours or days in some instances even though its not the core objective of such individuals. This set of individuals most of the times spend more time designing and configuring such networks than on their core objective. The following section describes the problems of configuration of networks and also the research gaps inherent in the proposed projects looking to abstract the network management process.

One of the issues that arise from the manual configuration process of network policies on devices is the time consuming nature of such an approach. This issue arises due to the fact that a network administrator needs to configure many network devices in order components is the time consuming nature of such an approach. . This process is known to be long, tedious and time consuming due to the number of command syntax needed to be configured on various devices. This issue can be exacerbated when the network administrator is dealing with a very large network that employs multiple network policies as part of the network's operational requirement.

The second known issue that arises due to the manual configuration process has to do with the complacency and possibility of human error in the process. This issue arises due to the fact that configuring network devices requires the network administrator to implement virtually the same set of command syntax except for a few variable changes. This repeated process especially when dealing with multiple similar network devices can lead to complacency, human error and in the long run a deployed network with numerous anomalies that deviates from the intended network policy requirement.

The third issue that arises due to the manual configuration process occurs when a network has different vendor products network devices. The presence of multiple vendor network devices makes it imperative for network administrators to learn multiple low level configuration languages. For example when a network to be configured has network devices from both Cisco and Juniper, the network administrator has to learn both low level languages. Learning multiple low level languages can be time consuming and very confusing.

The fourth issue that arises due to the manual configuration process of networks has to do with the lack of a real-time automated debugging and anomaly resolution in the process. This can lead to a deployed network with numerous anomalies. These inconsistencies and/or anomalies can end up rendering users of the network partially

or completely disconnected from network resources. Debugging such problems in a deployed infrastructure is usually very difficult and may end up consuming as much time as the configuration in the first place.

The fifth issue has to do with the research gap left by recent network management process abstraction projects. In recent years, many projects such as [4], [32] have to some level abstracted the network management process typically on the network device level or individual network policies (such as Propane). There is still a research gap for abstracting a higher level of network management process concepts so as to help network administrators deploy networks from a combination of high level policies and network topologies. Some of the current network management process abstraction projects only support one network policy. This limitation makes it difficult for network administrators designing and deploying complex networks with multiple network policies to utilise. The network administrators that use such tools have to manually configure physical devices deployed with policies not supported by such projects. This introduces the same manual configuration issues described above.

1.2 Outline of Areas of Research

The research carried out in this thesis can be categorised into three broad areas: network policies; network anomaly resolution (or configuration consistency) and network abstraction.

1.2.1 Network Policies

This section of the research deals with the various policies that guide network connectivity and access control amongst many others. A rudimentary understanding of what policies entail and how they are specified is necessary to appreciate the research conducted in this thesis. The following section defines and looks at the categories of policies in the sense of usage, triggers and how they are applied.

A policy rule can be defined as a set of conditions that need to be met in order for an action to be taken. A policy document is a collection of policy rules that controls certain aspects of an activity. A network policy document hence can be defined as a collection of rules to be implemented by an administrator to manage network devices. According to Stone et al[89], policies can be grouped into three areas: (i) how the policy is used (ii) how the policy is triggered (iii) at which level the policy is applied. The following section discusses these three policy areas.

1. A usage policy describes which services will be used to maintain the current state of the network or to transition to a new state. Services which may be available

in the network include differentiated service classes, routing and connectivity, virtual private networks, encryption capability, and so on. A usage policy also describes how those services will be used. For example, the ability to differentiate the handling of separate flows of traffic based on the service class in which they reside or to which virtual channel they belong describes how a service is used.

2. Policies can be triggered in two ways, either statically or dynamically. Static policies apply a fixed set of actions in a pre-determined way according to a set of pre-defined parameters that determine how the policy is used. Examples of static policies are:

   - Transit traffic is not permitted during normal working hours
   - For security reasons certain network addresses are denied access to network components.

   Dynamic policies are enforced when needed, and are based on changing conditions of the network such as congestion, packet loss, or the loss of a network router. To support the dynamic and sometimes unexpected nature of the network, actions can be triggered when an event causes a policy condition to be met. Examples of dynamic polices are:

   - When the network gets congested, streaming video traffic is disallowed
   - When a particular service class of user is utilising the network, lower best-effort traffic to only 25 percent of link capacity.

   The research conducted in this thesis will be using both static and dynamic policies in such a manner that abstracts the network management process. Static policies are first implemented for firewall access control configurations while BGP policies are known to be dynamically configured after the routers within a proposed network converges based on agreed upon intentions.

3. Lastly, the level of the policy is applied as a category. These policies are differentiated by their granularity, such as the application level, user level, class level, or service level. For example, a mission-critical application may be given priority over all other network traffic, or all users in the silver class (differentiated services) have priority over the bronze class, but must succumb to the gold class.

There are three policies that will be the focus of the research in this thesis and will be discussed in-depth in the following chapters. These policies include - inter-domain routing protocol or Border Gateway Protocol (BGP), firewall rule relations and cyber security competitions. These policies have been chosen for the research work conducted in this thesis because they cover a diverse level of networking concepts. BGP deals with

network layer manipulation, firewall relationships deal with both transport and network layer while domain name systems are applied on the application layer.

1.2.2 Network Anomaly Resolution

Network anomalies can be defined as the circumstance when the operation of a network deviates from its intended policy requirement. Network anomalies can arise due to numerous reasons including amongst others - configuration mistakes or errors; malfunctioning network devices; network intrusions that disrupt delivery of services, etc. We posit that network anomalies can be categorised into two groups - configuration anomalies and post deployment anomalies.

Configuration anomalies typically occur during low level configuration of network devices based on the policy requirements developed by an organisation and to be configured by network administrators. Manually configuring network policies on network devices is known to be susceptible to anomalies (or conflicts). An example of such an anomalies is when a network administrator mistakenly configures two identical firewall rules in a firewall device. This is the category of anomalies that are relevant to the research work done in this thesis and hence will be discussed in-depth in each network policy abstraction chapter.

Post deployment anomalies generally occur after the network has been configured and is up and running. An example of this type of anomaly is when hackers malicious inject bogus BGP prefixes into a network. The anomalies in this category are outside the context of this research and will not be discussed in this thesis.

1.2.3 Network Abstractions

Abstraction is the process of deciding the details of an activity that need to be highlighted and those that need to be hidden or ignored. The rationale behind abstraction is to make complex ideas much easier or simplistic to users by getting rid of reoccurring concepts. Network abstraction can therefore be defined as the process of hiding of network management details from the administrator. Current abstraction research projects have successfully abstracted details such as IP address allocation, names of network devices and basic configuration of network components from administrators. An in-depth discussion of the two network abstraction techniques including their advantages and disadvantages will be carried in the following chapter of this thesis.

1.3 Aims and Objectives

The research proposed in this thesis aims to simplify the network management process by providing a scalable and flexible method of expressing high level network policies by abstracting (or reducing) the non-essential details. The proposed research hopes to abstract the network management process on a higher level (policy) than current projects. The following outlines the aims and objectives of the research conducted in this thesis.

- The main aim or objective of the research conducted in this thesis is to simplify the network management process for network administrators. A set of network policy level abstractions independent of network layouts are proposed in a format (graph) that network administrators are familiar with. The level of network abstraction that we hope to achieve here will be higher than any other network management process that has been proposed using template design approach. The policy intentions that have been implemented to showcase policy level abstractions have to do with inter domain routing relationships, firewall relationships and the relationships between the various teams participating in a cyber security competition. The abstraction technique can be easily used for specifying various other policies that govern networks.

- The second objective of the research work conducted in this thesis is de-coupling the policy intentions from the network layout. This is important because network administrators can use the policy intentions or network layouts for different projects. De-coupling the policy intentions from network layouts make it easier to isolate and correct mistakes by network administrators.

- The third objective of the research work conducted is to integrate a network anomaly resolution system that ensures the networks proposed by administrators behave as intended. This is very important because resolving network anomalies post deployment can be a very tasking job for administrators.

- The fourth objective of the research work conducted has to do with developing a network management process abstraction system called (NePAS) that will be used to read various network abstractions in any proposed experiment. The system developed is composed of four phases: policy intention phase which is used for specifying network policies independent of network layouts; network layout phase which is used for expressing the topology of network devices that will be used during experiments; anomaly resolution phase which will be used to ensure low level configurations are conflict free; and the compilation phase which is where the actual low level configuration commands are generated for deployment.

- The last objective of the research work conducted in this thesis is to present a series of experiments and show how network policy anomalies in abstraction models will be resolved by the system developed. Network policy anomalies will be purposefully injected in the experiments presented so as to show how they can be resolved before low level deployment. The experiments presented will also be used to show the strengths and weaknesses of our abstraction models.

1.4 Contribution of Thesis

The research conducted in this thesis builds on current projects that provide abstract network layout specification for network administrators. The contributions hope to use template design approach of network abstractions and implemented using graph theory. Graph theory has been adopted because organisations propose and design their networks using such methods. This will make the job of network administrators tasked with configuring networks much easier. The contributions of this thesis can be summarised as follows:

- The first contribution of the research proposed in this thesis is providing a set of abstract specifications that will enable network administrators configure their networks from high level BGP business relationships. The abstractions proposed enables the generation of anomaly-free low level configuration for routers using community filters from a graph of high level policy intentions of business relationships. The abstractions are designed independent of network topology effectively raising the network management process of configuring BGP policies from network devices (or routers) to business relationship specification. The BGP abstractions enable network administrators to specify business relationships such as provider, customer, peer and sibling between autonomous systems.

- The second contribution of the research proposed in this thesis is providing a scalable and highly flexible set of firewall policy intentions. The abstractions that were developed can be specified (or expressed) on the policy intention level or network layout level. The firewall policy abstract specification enables the generation of anomaly-free low level rules from a graph of high level policies independent of the network layout. The network layout firewall intentions will give network administrators flexibility and granularity by allowing actual network devices to have isolated rules from other devices within their high level realm when deploying firewall enabled networks.

- The third contribution of the research proposed in this thesis is providing organisers of cyber security competitions a set of abstract competition intentions from high

level graph specifications. The research conducted presents a set of abstractions that allow organisers to specify the various participating teams and the relationship between them. The policy intention of team relationships expressed on a graph can be used with a slice (or cookie dough) of network topology to replicated for all the participating teams so as to generate low level competition infrastructure.

1.5 Thesis Outline

This thesis is organised as follows: Chapter 2 looks at the background rationale of the research work presented in this thesis. The first two sections of the chapter detail the various network abstraction and virtualisation techniques. The next three sections discusses some policy abstraction projects that have been implemented in the past for Border Gateway Protocol, Firewall Rule Relations and Cyber Security Competitions respectively.

Chapter 3 will present a philosophical overview of template design approach to the network abstraction design process. The chapter will begin with a brief overview of why abstracting the network management process is important. The chapter is then split into three sections looking at network management process abstraction on two levels - policy level and network topology level.

Chapter 4 will present our approach for abstracting network policies using template design. The chapter will present a detailed assessment of our system's four phases which are: (i) network policy intention abstraction, (ii) network layout abstraction, (iii) anomaly resolution and (iv) compilation. This chapter will explain the design consideration for each phase and how they co-depend.

Chapter 5 will present Border Gateway Protocol policy abstraction technique implemented for our proposed system. The chapter begins with a detailed overview of BGP, its path attributes and the various business relationships supported by the protocol. The subsequent sections of the chapter is used to discuss how the various phases (with simple example) of our system have been used to abstract BGP policies with an example every step of the way.

Chapter 6 will present firewall rule relation abstraction technique implemented for our proposed system. The chapter begins with a detailed explanation of firewalls, three classifications of firewalls and how firewalls work. The subsequent sections of the chapter is used to discuss how the various phases (with simple example) of our system have been to abstract firewall policies.

Chapter 7 will present cyber security competition abstraction technique implemented for our proposed system. The chapter begins with an in-depth assessment of the step-by-step guide of organising cyber security competitions. This entails different concepts from team formations and objectives to infrastructure components used in typical cyber

security competitions. The subsequent sections of the chapter is used to discuss how the various phases (with simple example) of our system have been used to abstract cyber security competition infrastructures.

Chapter 8 will present a critical assessment of our proposed system. The chapter is composed of two sections - experimentations and evaluation. The first section is used for in-depth complex experiments of the network policies abstracted in this thesis. The second section is used to evaluate our proposed system based on its performance, usability, flexibility and scalability.

Chapter 9 will detail conclusions drawn from our work and directions of any future work.

Chapter 2

Background and Related Work

2.1 Network Abstraction Design

Network abstraction is the process of hiding the details or reducing the complexity of a network management process so that the administrator can focus on the high level relationships of the network. By abstracting or hiding details from the network administrator when designing and deploying high-level policies, some of the network anomalies that deal with the configuration process listed in Chapter 1.2.3 can be greatly minimised. Current network abstraction systems allow administrators to concentrate on individual network devices at the topology level through to generation of low level configuration commands without complex network policy intentions implemented on them. Network configuration especially when dealing with large scale networks is known to lead to large outages when small but critical mistakes are made. Critical mistakes of large scale configuration is caused by the long and repetitive nature of such a process. Manual configuration on large scale networks is known to be prone to human errors and consume time. These factors amongst others have made researchers to look at ways of abstracting the network management process in recent years. Some of the benefits of abstracting the network configuration and management include among others:

- Automating the network management process will reduce misconfiguration due to human errors to the minimum and ensure end-to-end connectivity of network devices [92].
- Automating the network management process will make it easier for operators to deploy networks containing devices from different vendors without learning multiple low level languages[77].
- Automating the network management process will drastically reduce the cost and time it takes to configure networks [92].

Even though abstracting the network management process has many benefits, some researchers however think otherwise for the following reasons amongst others:

- Automating the network management process can lead to an increased cost of IT operation when the overhead of creating and executing the automation software is high when the task being automated does not need to be repeated often.
- By using automation, some valuable information necessary for decision making can be hidden to the network administrator.
- Lee et al [66] argue that automating simple and intuitive tasks can hurt network operations in the long run. This is because the network administrator will lose context of what is going on in the network.

Some things that designers of automated configuration tools need to be aware of when creating such software include the following among others:

- Lee et al [66] argue that automated configuration tools should provide peepholes to the low level details of a network even when the normal requirements by the operators is to hide them.
- Designers of automated configuration tools should choose the *right* level of abstraction while hiding the unnecessary details [26].
- Designers of automated configuration tools should ensure that their tools support the semantics of various vendor devices and not be bound to a particular implementation [40].
- Designers of automated configuration tools should ensure their tools meld together different services with varying degrees of interactions and dependencies.

There are two approaches to designing network abstraction tools - template design approach and language design approach. These two techniques will be discussed in the following section.

2.1.1 Template Design

The template design approach for designing automated configuration tools uses a bottom-up methodology[62]. This approach uses a combination of template snippets and contents from a network information data base(NIDB) so as to generate low level configuration commands. The template snippets which are for various network management processes such as BGP policies contain variables that will be substituted with contents from the NIDB so as to generate configuration files for specific vendor devices. This approach

is known to be simple, flexible and extensible [62]. Bellovin and Bush [18] created an automated configuration tool using template snippets for various network configuration processes such as BGP peering, packet over SONET(POS) interfaces etc. The tool is used to generate configuration files by substituting some of the variables in the template snippets with data retrieved from a database.

The template snippets support different models and vendor devices including routers and switches. Some of the advantages of the template design approach include amongst others: (i) there is increased consistency as data items only need to be entered once, (ii) new configurations can be pushed out to nodes very quickly, (iii) it reduces costs of having to correct errors arising due to manual configurations.

Many other developers have used template design approach to create automated configuration tools in the past. Some major disadvantages of using this approach include among others:

- Efficiently allocating network resources such as IP address blocks and BGP community attributes can become very complex when dealing with large scale networks [77].

- Another major disadvantage of this approach is that it is fundamentally device centric in nature. This is due to the template snippets required for low level configuration commands that will be generated and configured on network devices are for specific vendors.

2.1.2 Language Design

The Languages design approach for designing automated configuration tools use a top-down methodology [62]. Designers taking this approach to design an automated configuration tool need to write an assembler that can be used by network operators to specify high-level network policies that will be compiled into network configuration files. Examples of how this approach has been used to create automated configuration tools include:

- Chen at al [26] created an automated configuration tool using this approach where they used Mosaic, a variant of Datalog to program automated network operations. Their tool provides network operators a flexible way of programming network-wide management and high-level task schedule using recursive queries. Some advantages of their automated configuration tool include the following: (i) it is straightforward to write queries that configure routers based on network conditions (ii) the tool can roll back to previous consistent state when a failure or policy violation occurs (iii) provides network administrators an easy way of specifying high-level policies

- Voellmy and Hudak [97] have also used language design approach to create their automated configuration tool. The tool uses a compiler created using Haskell language to translate Nettle programs used to generate routing configuration files for XORP routers. This allows network administrators to specify network-wide BGP policies and vendor-specific router details in a uniform language. The tool only supports two routing protocols, BGP and static routes. Nettle only supports XORP routers but the designers intend to support all major router platforms. Some advantages of this automated configuration tool include the following: (i) provides a flexible and safe way to merge multiple routing policies (ii) a single policy specification can be given for a heterogenous collection of routers (iii) it allows operators to specify one cohesive policy for the entire local network routing configuration

Many other developers have used language design approach to create automated configuration tools in the past. Some major disadvantages of using this approach include among others:

- The language design approach leads to another language that network administrators need to learn just to use the tool
- When dealing with a compiler that does not support all aspects of the network in use, it might lead to the network not behaving as intended.

2.2 BGP Policy Abstraction

There has been extensive research regarding various aspects of network routing especially on inter domain routing. This research covers a wide range of fields including but not limited to security, traffic engineering, inferring AS relationships and anomaly resolution. However there has not been such extensive research on inter domain routing policy abstraction especially on the business relationship level. The following section summarises some projects that have sought to abstract BGP policy intentions on the business level in the past.

Vanbever et al propose in [93] a hierarchical structured model that uses chains and filters to specify various levels of BGP policy abstraction. The proposed model uses two chains of high-level routing filters to express BGP routing policies. The tool implemented was developed in Java and composed of three components: (i) a data repository containing BGP session descriptions, import and export policy chains; (ii) a predicate matching engine which relies on regular expressions to represent various attributes and BGP relationships; (iii) a generation engines that iterates over all BGP sessions and applies import and export chains to generate low-level import and export

filters for the sessions. One of the major drawbacks of the abstraction developed is its lack of a very expressive level of BGP business relationship abstraction. Another major problem with this project is the lack of information with regards to which BGP business relationships are supported if any by this project.

Gottlieb et al proposed in [51] a tool that uses provisioning databases to build configurations of large ISP eBGP sessions. This project only configures customer sessions using a set of technical questionnaires used by the provider to identify unique identifiers on behalf of the customer. The customer-specific information provided in the technical questionnaires are then used by a set of provisioning rules to generate a sequence of Cisco IOS configuration commands for adding the new connection to the network. A database schema is then used for storing and accessing customer-specific data that populates a template for capturing the syntax of the router commands. One of the major drawbacks of this project is its lack of support for other BGP routing business relationships such as transit, peering and siblings.

Autonetkit [4] is a tool that takes high-level network representations and compiles it into a choice of emulated and simulated environments. It uses a graphical editor such as yED to visualise the intended network layout and policy representation within the graph. The network topology graph created is read using NetworkX graph library for python with nodes and edges representing routers and physical connections respectively. Attributes such as AS numbers, link speed or weight for edges can be easily represented just as additional custom attributes such router type, local preference value amongst others are supported. One of the major limitations of this project is that the level abstraction provided is at the device level or network layout abstraction. Abstract representations of BGP business relationships at a level higher than the network layout level is not supported.

Cisco VIRL [32] is a vendor-specific proprietary simulation platform developed by Cisco. The project uses VM Maestro which is a powerful cross-platform GUI environment that allows a user to drag and drop various network devices in order to represent complex topologies. The versatile tool provided in the project allows a network administrator to either manually configure various network devices or leverage Autonetkit to create bootstrap configurations using official Cisco images for components in the workspace. The project allows network administrators to perform different network measurement activities which includes amongst others: (i) ability to see path taken by network using tools such as traceroutes; (ii) ability to set the latency, packet loss and jitter on links so as to model complex network environments. As the automation engine of Cisco VIRL depends on Autonetkit, they share the same limitations. Cisco VIRL just like Autonetkit neither supports a higher level of abstraction on the business relationship level nor a network-wide firewall enabled network such as the one proposed in this project. However, we view Cisco VIRL/Autonetkit and our work as complementary

research projects, with each focusing on a different level of abstraction.

2.3 Firewall Policy Abstraction

There has been extensive research about firewalls over the years. The research encompasses configuration management, conflict resolution and visual analysis. The literature on firewall policy abstraction that is scalable, modular, conflict free and easy to use is however very limited. The following section discusses some of the work carried out in field of firewall policy abstraction.

Firewall Builder [2] is a GUI based application that can be used to configure firewalls on various platforms such as Cisco (ASA, PIX, router access lists); Linux iptables; HP ProCurve ACL firewalls amongst others. Firewall Builder can be used to configure firewalls using the following five steps: create firewall; define objects; configure policy; compile rules and deploy configuration. This tool does not support automatic generation of IP addresses and hence can be prone to error and tedious consuming. The lack of automatic generation of IP addresses makes Firewall Builder time consuming and prone to error especially when dealing with large enterprise networks. The tool also does not have support for any other network policies such as inter domain routing and hence when dealing with a network that is composed of both firewall and routing policies, it will not be suitable.

The Cisco Adaptive Security Device Manager (Cisco ASDM) [31] is a web-based tool used for configuring, monitoring and troubleshooting Cisco firewall appliances. This tool uses a setup wizard that guides network administrators on how to configure and manage firewall devices. It also provides a built-in packet trace and packet debugging tools for troubleshooting the device. This tool is restricted to configuring only Cisco firewalls and does not support any other platform. It can only configure firewall devices on a per-device basis and not from a network-wide policy abstraction basis. Similar to Firewall Builder, it does not support other network policies such as the ones for inter domain routing and hence is rather limited when dealing with enterprise networks with various network policies.

Firmato [17] is a tool used in generating low-level firewall ACLs from high-level policies. The high-level policies are modelled using Entity-Relationship Diagrams that will be used to generate the low-level firewall rules. This tool is highly complex and similar to many existing low-level languages and does not support many vendor hardware devices. This makes it highly unsuitable for network administrators because it is not easy to use. Firmato also does not support NAT which is a very important network policy in the world today. Firmato also cannot express negative ("drop" action) rules. This can make expressing exceptions highly complex as a lot of rules will be needed in order to express negative rules. Similar to the above tools, Firmato does not support any other

network policy abstraction and hence is highly limited when it comes to configuring a network with various policies.

Mignis [9] is a semantic based tool used for firewall configurations. Mignis uses a top-down approach that allows the network administrator to specify firewall rules which are then translated into Netfliter firewall configuration commands. The tool takes an abstract firewall rule configuration file which it then uses to generate a series of low-level iptables rules. Mignis supports NAT and the notion of grouping multiple host devices into a security policy group that have similar firewall configurations. One of the major limitations of the Mignis tool is its lack of support for various platforms as it currently only generates iptables rules. Mignis does not support a separated high-level policy and the network policy which makes automatic allocation of IP addresses to the various devices easier to generate. This means the network administrator has to specify the IP addresses of the various devices and hence makes the entire process more time consuming. Mignis also does not support any other network policy abstraction which makes it highly limited and unsuitable for configuring large scale enterprise networks.

2.4 Cyber Security Competitions Abstraction

There is a wide range of organisations from academic, government and private institutions that host various kinds of cyber security competitions (CDX) on a yearly basis. There has also been a lot of research and publications on organising CDX events and experiences learnt during such events. However, there are limited proposals on abstracting these events in such a way that will enable organisers to deploy such events with ease and flexibility. The following are the major projects proposed in the past that hope to abstract CDX events so as to make it easy for organisers to design, configure, deploy and manage.

The SecLab Group of University of California in Santa Barbara developed a framework called iCTF [6] that can be used to create customisable interactive security competitions. The iCTF architecture is split into two - the game and admin networks. The game network hosts all the virtual machines running the vulnerable software of various participating teams. The virtual machines are VirtualBox appliances which can either be hosted by the various participating teams and connected using a VPN concentrator VM generated by the iCTF. They use VirtualBox in order to be able to leverage its internal networking feature. The admin network is composed of a central database, scorebot and web server. The primary function of the central database is enforcing game rules and keeping track of the competition's state.The scorebot's function is to monitor the status of services running in the game network. The web server is used to host the scoreboard, chat messenger, hints and challenges.

The Tele-Lab [103] project provides an e-learning system for practical security training. The Tele-Lab server developed for the project consists of two major parts:

(i) web-based tutoring system; and (ii) a training environment composed of virtual machines. The tutoring system consists of three kinds of content: information chapters; introduction to security and hacking tool; and practical exercises. The information chapters and introduction to security and hacking tool content of the tutoring system is used to introduce the user to the theoretical aspects of several networking concepts such as "wireless networks", "reconnaissance" and "malware" amongst others. The practical exercises section are carried out on the training environment (using virtual machines) which can be accessed by users via remote desktop access.

The iTEE project [41] is an open-source platform that enables creation of interactive cyber security competitions. The iTEE platform developed is composed of three layers: virtualisation layer; virtual lab layer; and virtual learning space layer. (i) The virtualisation layer is used to host the virtual machines created and all the upper layers of the platform depend on this layer. The hypervisor used for this project is based on Oracle VirtualBox Headless run on an Ubuntu Server. (ii) The virtual lab layer is used to provide remote access, control functions for the virtual machines and executing the competitions. The virtual machines are provisioned automatically using a template-based abstraction approach. This leverages on puppet configuration manager, shell and Python scripts to customise the virtual machines and networks. (iii) The virtual learning space layer is used to provide the competition scenario, objectives, automated scoring and automated attacks during the competition.

2.5 Closing Remarks

Template design approach will be employed in abstracting the complex network policies in this thesis. This method was chosen because we believe it will be easier for network administrators to express and deploy experiments in a seamless manner similar to white-board style design currently employed. The network policies chosen in this thesis are complex and designing a low level abstraction language will force network administrators to learn another language which defeats one of the major objectives of this project which is to simplify the network management process.

The projects described as part of the related work in the above section all fall short of the level of abstraction we have achieved in this thesis. Some of the projects such as Autonetkit and Cisco VIRL only allow network administrators to express networks on a network layout level without complex policies such as inter domain routing relationships and firewall relationships configured on the deployed devices. This limitation means network administrators will have to manually configure such network devices with complex network policy configurations which can be timing consuming, tedious, complacency and likely to introduce anomalies in such networks. Also, to the best of our knowledge there is no proposed inter domain routing abstraction project that uses template design

approach that gives network administrator the functionality of expressing network with complex policies such as transit, peer and sibling relationships. This limitation makes it difficult for network administrators to utilise such projects because they would have to manually configure network routers with the other routing relationships that are not supported by the tools being used. This also introduces complacency, time wasting and possibilities of having anomalies in deployed experiments by the network administrators. Lastly, all the network policy abstraction projects currently proposed support only one network policy such as firewall builder for firewalls or propane for inter domain routing. This limitation we posit defeats the purpose of network management process simplification because most networks having two or more network policies working seamlessly together. This limitation forces network administrators to deploy partially configured networks with such tools and having to manually configure the unsupported network policies which can is counterintuitive for network administrators. The focus of the research conducted in this thesis is to develop a set of high level network policy abstractions beyond the network layout level that network administrators can use to design and deploy experiments that have a combination of both firewall and inter domain routing policies. This will be an improvement to current projects and will enable network administrators to design and deploy anomaly free complex networks that have such network policies with ease.

This chapter of the thesis has introduced the two known techniques for network abstractions: language and template design approaches. The chapter also highlighted and discussed some of the major related research projects conducted on the three network policies focused in this thesis: BGP, firewall relationships and cyber security competitions. The next chapter of this thesis will be exploring the philosophical background of template design abstraction technique which has been adopted for the research conducted in this thesis.

Chapter 3

Philosophy of Network Abstractions

3.1 Introduction

A solution to computer network complexity especially when trying to simplify the management process problem is abstraction. Abstraction can make a seemingly complex network less so by removing or reducing details such as network devices, network services, few protocols and anything that can be removed without tempering with the behaviour of the network once deployed. Network abstraction models should be able to deploy high level policy intentions that matches down into low level vendor specific configuration commands easily. The low level configuration commands generated from the abstract representation of the network specified should be conflict free and behave as intended by the network administrators. Abstraction can also greatly simplify and minimise errors in the network management process when updating deployed enterprise networks. For example, a user can change or update a single firewall intention that reverberates across a large set of network devices. This chapter is broken into four major subsection as follows:

- Section 3.2 of this chapter is used to examine the various ways network information can be obtained from administrators and why graph theory was preferred over other data structures in this research.
- Section 3.3 of this chapter is used to state the deployment environment of proposed experimental network models.
- Section 3.4 of this chapter showcases how computer network policies can be abstracted by removing details such as the network devices in the proposed layout to be deployed. The abstractions proposed in this section have no bearing the network layout and/or devices that such policies will be implemented on during deployment. Inter-domain (BGP) routing policies, firewall intention policies and

cyber security infrastructure relationships will be introduced in this section, with more in depth discussions presented in Chapter 5 , 6 and 7 respectively.

- Section 3.5 of this chapter showcases how the network layout of a proposed experiment can be abstracted either as re-usable blocks (or collection of devices) or slices (cookie dough) of network layouts as the case maybe. It should be noted that the abstractions presented in this section remove details such as the vendor product being used, IP addresses and others from the network administrator.

3.2 Graph Theory

This chapter of the research conducted hopes to give a philosophical background of how abstractions based on template design approach be used to simplify and/or improve the network management process for network administrators. The following section will be used in showcasing how our proposed abstraction methodologies can be implemented in order to achieve the aim of simplifying the network management process.

There are multiple ways in which the network information of a proposed experiment can be gotten from a network administrator, these include amongst many others data structures, sets, mathematical algebraic expressions and graph theory. All these and many others can be used to get details of a network experiment from an administrator. Graph theory is the study of graphs which are used as a way to formally represent networks or inter-connected devices. The inner-workings of graph theory is beyond the scope of work conducted in this research and hence will not discussed in-depth. The graph in Figure 3.1 below shows a basic graph with two nodes labelled A and B. The nodes (also called vertices) are connected using an edge (also called link) in the diagram below.

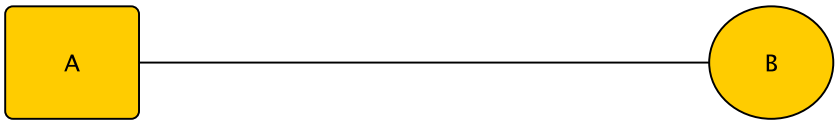

Figure 3.1: A Basic Graph

We use the nodes of a graph similar to the example in Figure3.1 above to represent either a collection of network devices (realms) or an individual network device. The edges of the graph in this research is used to represent the relationship between two realms or as a link between two network devices. Graph theory have been adopted in this research for expressing both high level network policy intentions and the intended network layout of any given experiment. An overview of how graph theory was used to

abstract the network management process will be conducted in this chapter and a more detailed discussion of it will be carried out in subsequent chapters.

The abstractions proposed in this research will be adopted in graph-like formats (using graph theory) similar to white board representations in order to simplify and make the network management process easier to showcase and understand. The proposed network abstractions presented in this research will be based on template design approach discussed in Chapter 2.1.1. Graphs will be used to abstract network policy intention and layout information of experiments which will be parsed to template snippets in order to generate low level configuration commands. There are two types of graphs used for the proposed abstractions in this research, these are: policy intention and network layout graphs. The first graph type is used to express high level network policy intention abstractions that are independent of the proposed network layout of the experiment. The second graph type is used to express an abstraction of the network layout that is independent of the network device vendor such as Juniper or Cisco. The nodes and edges of the graphs have custom properties that are used in populating the NIDB of the proposed network. A more detailed and comprehensive discussion of the graph and properties in them will be carried out in the next Chapter 4. This chapter will be looking at network abstraction from two angles: network policy intention level and network layout level. The policy intention level of network abstraction is the highest level where the relationships between group of devices with identical intentions are specified. The network layout level of network abstraction is on a lower level than the policy intention and deals with how network layouts of proposed experiments can be simplified and expressed.

3.3 Deployment

The progression of the research work conducted in this thesis is to propose high level network policy abstractions that will implemented in a virtualised environment and then move on to real work or physical hardware deployments. There is a number of issues that manifest with the deployment of networks in the real world such as BGP wedgies (discussed in the next chapter) that are beyond the scope of work conducted. Also, BGP policies are implemented based on multiple service provider agreements that are opaque and not readily available to individuals outside the organisation. This makes configuring BGP policies in the real world much more complex for the various network administrators managing those networks. All the work carried out in this thesis is therefore envisaged to be deployed in a controlled virtualised environment.

3.4 Policy Intention Level Abstraction

This is the first part of our philosophical research contribution on network abstraction. This section will be focusing on the highest level of the network abstraction process, network policy intention abstraction. This level of network abstraction sought to view the network management process from a policy specification angle. As stated in chapter 1 of this research, network policies are a collection of rules used to govern the connectivity, routing and/or access requirements of the devices within a proposed network layout. At this level of abstraction, the network layout or infrastructure devices are not in contention or specified. The network policy of devices with identical behaviour are grouped together expressed in this section. Two network policies dealing with network routing (BGP) and access control (firewall) will be introduced in this section. It should be noted that a more comprehensive discussion of the network abstraction process for network inter-domain routing and access control will be detailed in Chapter 5 and 6 of this research respectively.

3.4.1 BGP Abstractions

This section will be giving a brief overview at our proposed high level inter domain routing intention abstractions. A comprehensive review of BGP and what it entails is be discussed in Chapter 5.1. However, a brief introduction regarding BGP will be given in the following section.

Border gateway protocol, BGP, is an inter domain routing protocol that is used in selecting the best route to any given destination; advertising reachability of prefixes; and accepting routes from neighbouring internet service providers (ISP) or autonomous systems. An autonomous system is a network or group of networks whose prefixes and routing policies are under a common administration. The following are some of the BGP business relationships that can be implemented between adjoining ISPs.

- **Peer relationship** - This relationship occurs when two or more similar sized autonomous systems establish a mutually reciprocal agreement where their network traffic pass through each others' networks without charging any fee.
- **Provider relationship** - This relationship occurs when an AS agrees to route the network traffic of another AS based on an agreed upon fee. The AS charging the fee is said to be the provider.
- **Customer relationship** - This relationship occurs when an AS solicits the help of another AS to route its network traffic after paying some agreed on fee. The AS paying the money is said to be the customer of the other AS.

- **Sibling relationship** - This relationship usually occurs when an organisation owns multiple autonomous systems. This relationship typically occurs when an Internet Service Provider (ISP) buys another ISP via acquisitions or mergers. In a sibling relationship, an AS can export its routes, customer routes as well as provider and peer routes.

- **Hybrid relationship** - This BGP business relationships occur when two ASes agree to have a combination of more than business relationship either based on geographical location or IP version.

These relationships are implemented on network routers using BGP path attributes such as the community attribute. The community attribute provides a way of grouping destination that have similar routing decisions applied on them.

As stated earlier, the abstraction proposed in this section hopes to concentrate on the high level BGP business relationship between autonomous systems. Network abstractions are only good if the proposed devices in the experiments are configured and behave as intended by the high level policy specification. A graph is used to express the high level BGP business relationship between autonomous systems. The nodes in the BGP policy intention graph are used to specify the autonomous systems involved in the proposed network. The edges between the nodes (or autonomous systems) are used to indicate the BGP business relationship using the edge label which can be peer, provider, customer or any of the other relationships mentioned above.

The example in figure 3.2 below shows a high level abstraction of a hypothetical relationship between five autonomous systems and their various business relationships.

The example in figure 3.3 below shows the network layout diagram of the hypothetical experiment from the high level BGP routing relationship discussed above. Routers rOrange1 and rOrange2 belong to Orange autonomous system; rTata belongs to Tata autonomous system; rBT belongs to BT autonomous system; rVodafone belongs to Vodafone autonomous system and rVirgin1, rVirgin2 and rVirgin3 belong to Virgin autonomous system.

From the diagram above, all links between routers in the same autonomous system will have internal BGP sessions configured. Links between routers in different autonomous system will however have external BGP sessions configured according to the business relationship from Figure 3.2 above. In order to implement the business relationship between routers in different autonomous systems, we configure community attributes of BGP to achieve low level implementation. The community attribute will be used to tag prefixes belonging to customer, peer and provider routes. The appropriate prefixes depending on the business relationship between the routers are then exported according to the high level policy intentions.

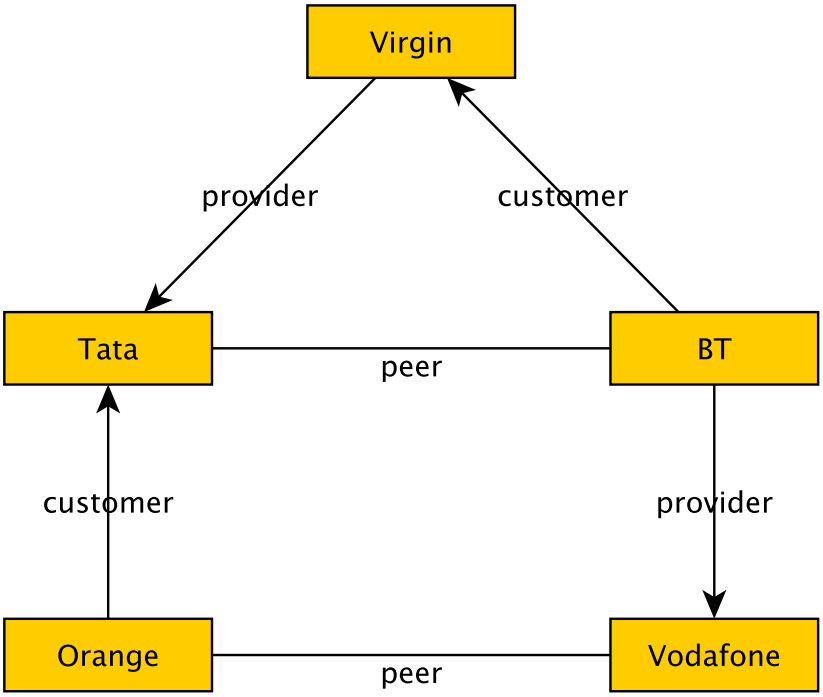

Figure 3.2: Abstraction of a hypothetical Internet

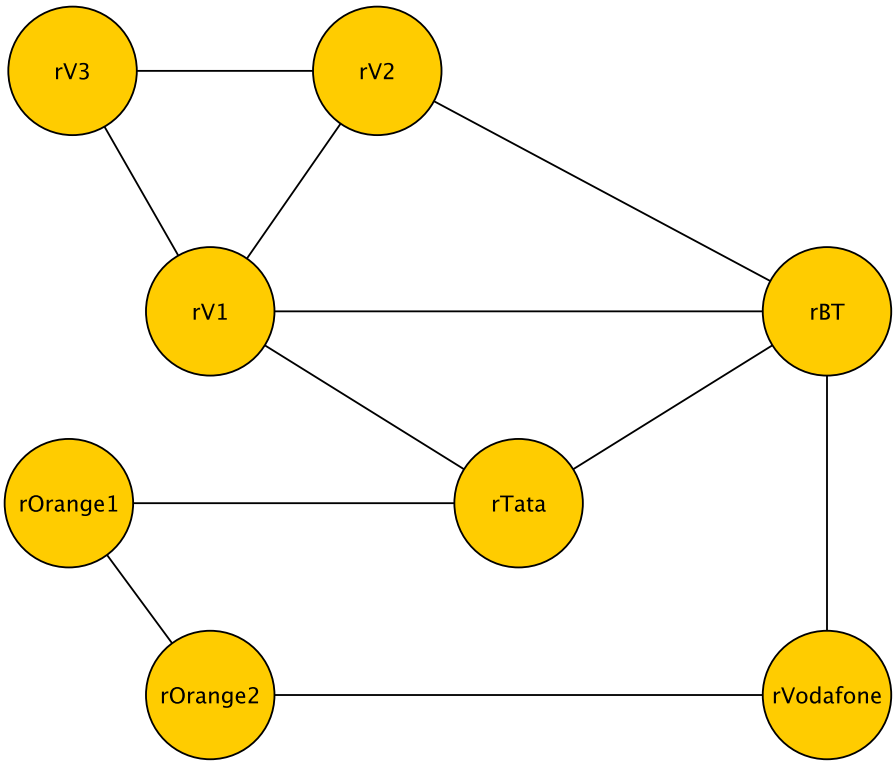

Figure 3.3: Hypothetical Internet Layout

3.4.2 Firewall Abstractions

This section will be giving a brief overview of our proposed high level firewall (or access control) abstractions. A comprehensive review of firewalls, how they work and where they are placed in networks will be discussed in Chapter 6. However, a brief introduction regarding firewalls will be given in the following section.

A firewall is a computer hardware or software that is used to protect individual computers and/or corporate networks from hostile attacks. A firewall achieves this by filtering (or blocking) the passage of undesirable data traffic from crossing into or out of a network. Firewalls block or allow network traffic based on a set of rules established by the organisation in its security policy and configured by the network administrator. The rules define a specific pattern based on a tuple that the firewall detects and an action to be taken. Therefore a firewall rule has the following tuple options: <action> <protocol> <source IP> <source port> <destination IP> <destination port> Firewalls try to match network traffic patterns from the first rule to the last as contained in the rule base in a sequential order. Firewalls always enforce the first rule that matches and hence rule ordering in the rule base is very important. The most important and utilised rules should be at the top of the rule base so as to ensure the firewall works efficiently. Firewall rules designed and configured by network administrators may be prone to anomalies that can render parts of the network either unaccessible or open to malicious attacks. There are two categories in which firewall anomalies can be grouped into, these are: intra firewall anomalies and inter firewall anomalies. Intra-firewall anomalies include: shadowing, irrelevance, correlation, generalisation and redundancy. Inter-firewall anomalies include: redundancy, spuriousness, shadowing and correlation.

As stated earlier, the abstraction proposed in this section hopes to concentrate on the high level access control policies using firewalls between network device or group of network devices (or realms) with identical policy intentions. Similar to the abstractions proposed for inter domain routing, graphs will be used to express the high level access control policies between two realms.

The nodes in the firewall policy intention graphs are used to specify realms (a device of group of network devices with identical policies) which transforms on the network layout level as a network firewall enabled device or an IP address in the firewall tuple information during low level firewall commands. For example, universities can choose to group network devices in their networks according to students, staff, campus, laboratory and such. Likewise an organisation can choose to group their network according to staff, visitors, management and departments amongst many others.

The edges in our firewall policy intention abstraction graphs are used to specify the remaining firewall tuple information between the two realms such as protocol; source and destination ports; and action to be taken on the matched network traffic pattern.

Labels of edges are used in specifying whether the action to be taken on the network traffic is either "permit" or "deny". It should be noted that any other phrases used during proposed experiments for abstractions will not work. There are five edge custom properties that have been developed for specifying the other firewall tuple information needed for firewall rules with default values in case a network administrator omits any of them. These edge properties are service (service), protocol (protocol), source port (sPort), destination port (dPort) and an external source/destination IP address or network (dest).

- The **protocol** edge property has been developed to give network administrators an avenue of specifying the protocol which can be either TCP, UDP or any other type of protocol value accepted by the firewall vendor product to be used during the experiment. This edge property has a default value of "IP" if a network administrator does not specify any value here.

- The **dPort** edge property is used to specify the destination port value for a given firewall tuple as a way of giving network administrators additional granularity during firewall policy abstraction specification. It should be noted that this value when specified along with the service property option discussed above, the pre-configured port number of a service will be overridden with the dPort value. For example, if a network administrator enters www (for web service pre-configured with TCP on ports 80 and 8080) and specifies a **dPort** value of 52435, ports 80 and 8080 of the pre-configured service value will be overridden with the new **dPort** value of 52435. This edge property has a default value of "NULL" if a user does not specify any value here.

- The **sPort** edge property is used to specify the source port value for a given firewall tuple as a way of giving network administrators additional granularity during firewall policy intention specification. This edge property has a default value of "NULL" if a network adminstrator does not specify any value here.

- The **service** edge property is used as an abstraction of the protocol and destination port number of a firewall tuple information. This gives the network administrator a way of specifying various network services such as dns (for domain name service), www (for web service using TCP on ports 80 and 8080), torrent (for torrent service using UDP on ports such as). This edge property has a default value of "NULL" if a network administrator does not specify any value here.

- The **dest** edge property is used to specify a network device or group of network devices not within the experimental network layout to be deployed. This edge property can either be a source or destination IP address of a network device or

group of network devices. This edge property has a default value of "NULL" if a user does not specify any value here. Additional information of how an external source or destination address is specified will be covered in Chapter 6.

The example in Figure 3.4 below shows a high level abstraction of a firewall policies with three realms (group of network devices that have similar policies) and the various access control relationships between them. There are fourteen firewall university network experiment.

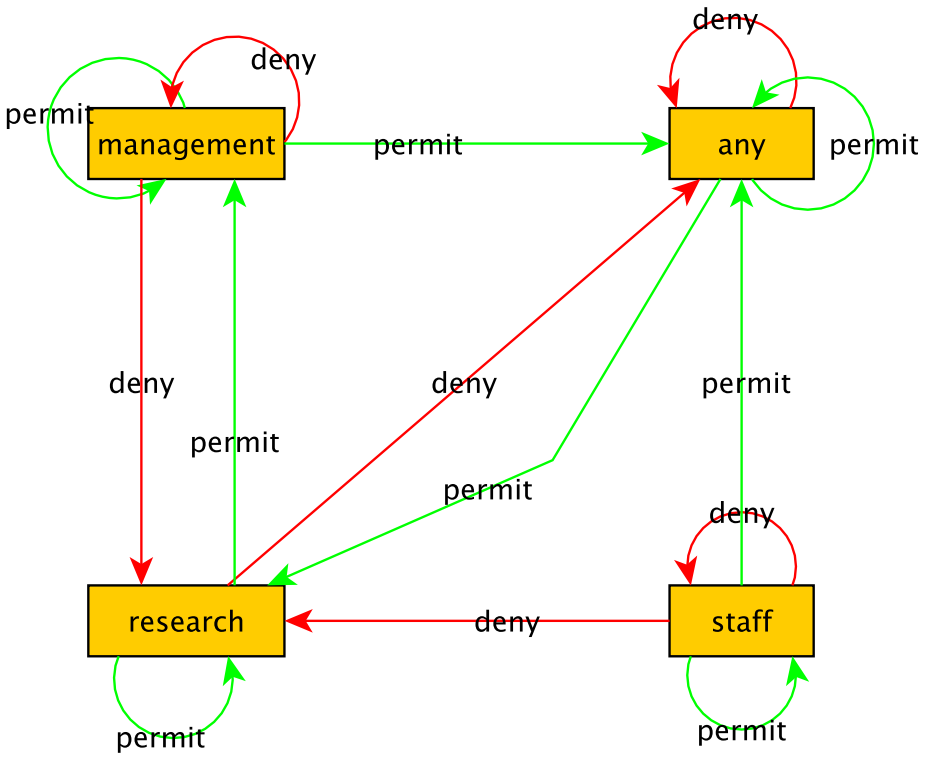

Figure 3.4: Abstraction of a hypothetical university network

The full detail of the edge properties used for each firewall rule abstraction is outlined in Table below.

Table 3.1: Proposed University Firewall Policy Intention Details

Source Realm	Destination Realm	Action	Protocol	Destination Port	Destination Address	Source Port
any	any	permit	ICMP	NULL	NULL	NULL
any	any	deny	TCP	80, 8080	d www.gorillavid.in, d www.twitter.com	NULL
management	any	permit	TCP	40728, 3689	NULL	NULL
management	management	deny	ICMP	NULL	NULL	NULL
management	management	permit	IP	NULL	s 1.2.3.4	NULL
management	research	deny	IP	NULL	NULL	NULL
research	management	permit	TCP	54321	NULL	54321
campus	any	deny	TCP	telnet, ftp	NULL	NULL
campus	campus	permit	IP	80, 8080	d www.gorillavid.in, d www.twitter.com	NULL
any	campus	permit	TCP	ssh	NULL	NULL

3.5 Network Layout Level Abstraction

The second level of the network management process abstractions showcases how network layouts can be abstracted. It should be noted that the abstractions presented in this section hide details such as IP addresses and proprietary network devices from the network administrator. The abstractions proposed in this section allow network administrators to specify network layouts using graphical editors for proposed experiments.

The nodes in the network layout graph are used to represent various network devices such as firewalls, servers, routers and others during any proposed experiment. Network devices within the graph can be assigned into various BGP or firewall intention policy realms within any proposed experiment. The policy intention selected at this phase is used in configuring policies that network devices will be using in the proposed experiment. Node options have been developed to specify which firewall or BGP realm a network device belongs to in a proposed experiment. This node option takes a string value of the policy intention realm's label of the proposed experiment. The value entered must be an exact match of the characters used as the policy intention realm's label.

The edges between the devices in this graph are used to represent the link connecting

two network devices. The only set of edge options implemented for our research abstractions on the network layout level has to do with manually specifying IP addresses so as to support the varying degrees of flexibility for network administrators. Edge options called *sIP* and *dIP* are used to specify custom addresses for each link within the proposed network. It should be noted that by default IP addresses are automatically generated for the entire network devices in an experiment.

This section outlines how to abstract a network topology in two ways - as a chunk of networks and/or slices of networks.

3.5.1 Chunks of Network Topologies

This section outlines how to abstract a collection of network devices that can be used in multiple instances of an experiment or different experiments by a network administrators. This abstraction has been designed in order to further reduce the time spent by a network administrator when developing an experiment that has multiple instances of the same block or group of network devices during the design process of a proposed experiment. For example, when designing an enterprise firewall network for a university, there might be a need for multiple instances of a laboratory network. A number of departments can all have a laboratory network with the same number of computers; services running on the computers; wireless routers and firewalls as the case maybe. Abstracting this collection of network devices can amongst others: reduce the time needed to design this block of devices whenever the network administrator encounters such a collection of network devices; reduce the likelihood of mistakes during the design process and ensure the high level policies of this block is the same throughout the proposed network. This motivated us to provide a way of abstracting such blocks for network administrators in an easy and modular way.

This set of abstractions can be used by a network administrator by creating a graph of the proposed block (group of network devices) in exactly the same way as the network layout phase of the system that has been developed for validating our research abstractions. This graph is as described in the Phase II of our system detailed in the previous chapter. Nodes labeled **ext** are used to indicate where the collection of network devices will be connected to a proposed experiment. It should be noted that there can be more than one connection points for network blocks in any experiment. The following sections show how this can be achieved.

The collection of network devices, or block, can be placed within the network layout graph during the design process. A sample university network will be used to showcase our proposed abstraction technique in this section. The high level firewall policy intention described in Figure 3.4 above will be used in the example detailed in this section. The network layout graph of the proposed hypothetical university network is depicted in

Figure 3.5 below.

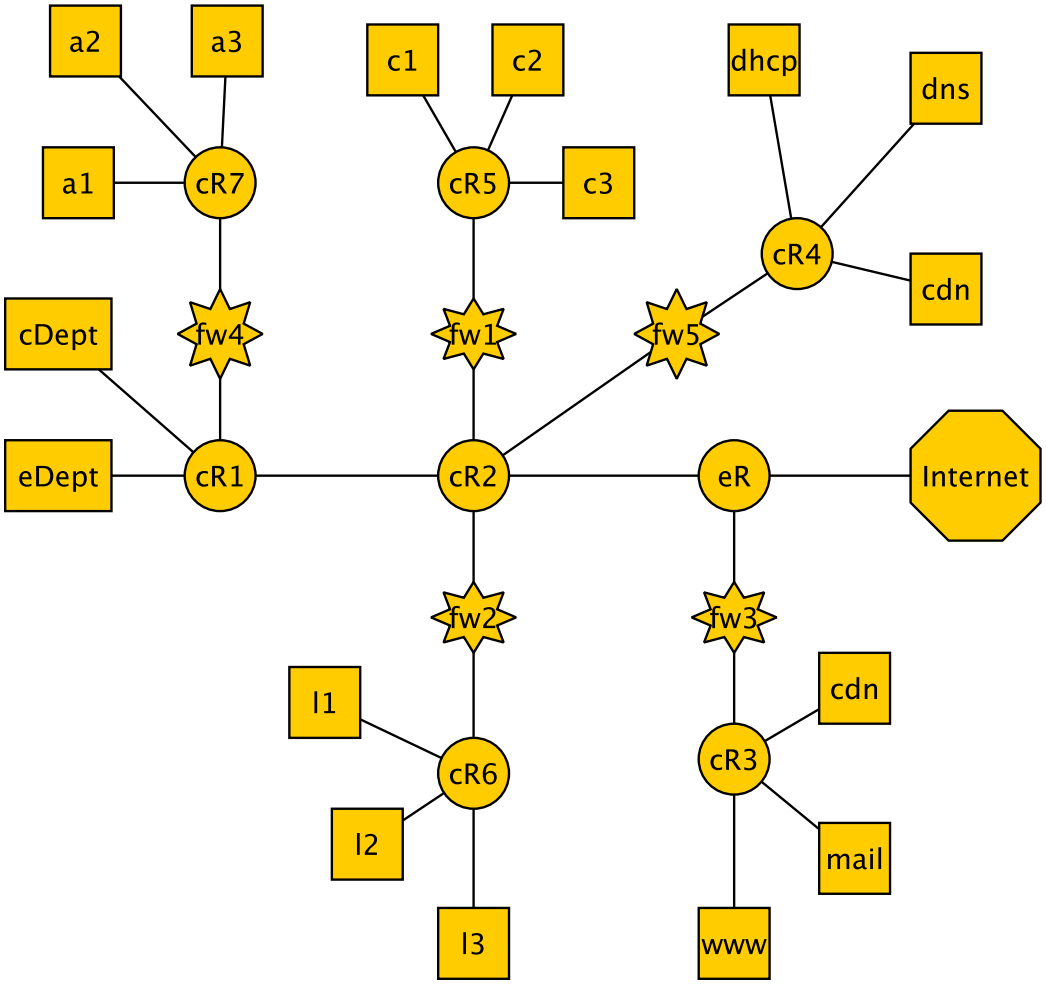

Figure 3.5: Proposed Hypothetical University Network Layout Diagram

The departments of the proposed university have identical network layouts hence the nodes eDept and cDept have block custom property value of department as they abstractions of such infrastructures. These nodes are abstraction of a collection of network devices used in representing entire departmental networks of the proposed university. Figure 3.6 below shows the network layout of the collection devices abstracted for the departmental networks.

As indicated in Figure 3.6 below, all departmental networks will have a set of servers that will be used by staff members of hypothetical university in our example designated as S1, S2 and S3. The network will also have a set of servers for research members of the department designated as R1, R2 and R3 in the same diagram. There is also another abstract network layout representation called Lab for all departmental networks. It should be noted that all the departments in our hypothetical university example have these exact network devices and links as represented in figure 3.6 below.

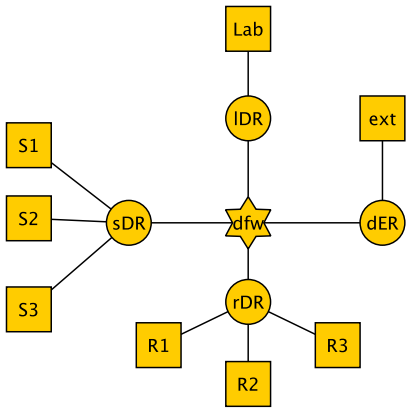

Figure 3.6: Department Block Network Layout Diagram

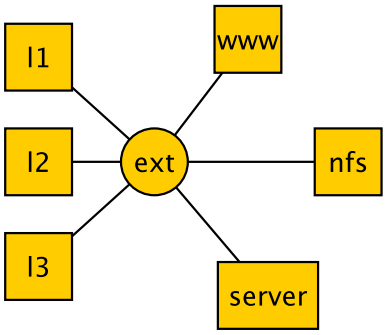

Figure 3.7: Department Block Network Layout Diagram

The laboratory network depicted in Figure 3.6 is another abstract representation of a collection of network devices specified as node, Lab in the diagram. The laboratory network of our proposed hypothetical university is also representation of lecture halls where students are taught within departments. The network is composed of student laptops/tablets represented as l1, l2 and l3 in figure 3.7 above. The network also has a server which is a machine connected to a projector typically used by the instructor for teaching. The devices www and nfs are used to represent a web server where students can access course materials directly and a storage server for submitting projects respectively. It is expected that all departmental networks in the university have a similar setup for their laboratories.

The diagram in figure 3.8 shows the network of how our hypothetical university will look like during low level deployment. As can be seen, both blocks, cDept and eDept, have been added into the final network layout. Likewise the laboratory networks of block Lab nested within the departmental networks has also been added into the final network topology.

Additional departmental networks can be easily integrated into the existing infrastructure by adding more blocks in figure 3.5 to represent new departments commissioned by the hypothetical university. The network administrators of the university can easily update agreed upon firewall policy intentions or network infrastructures using abstractions on blocks of departments or laboratories. For example, if the management of the university wants to change the network layout of laboratory networks to include additional network devices or change the firewall policy intention realm of some devices within the infrastructure, they can do so easily without disrupting the university network. The single change will also be replicated across all departmental networks.

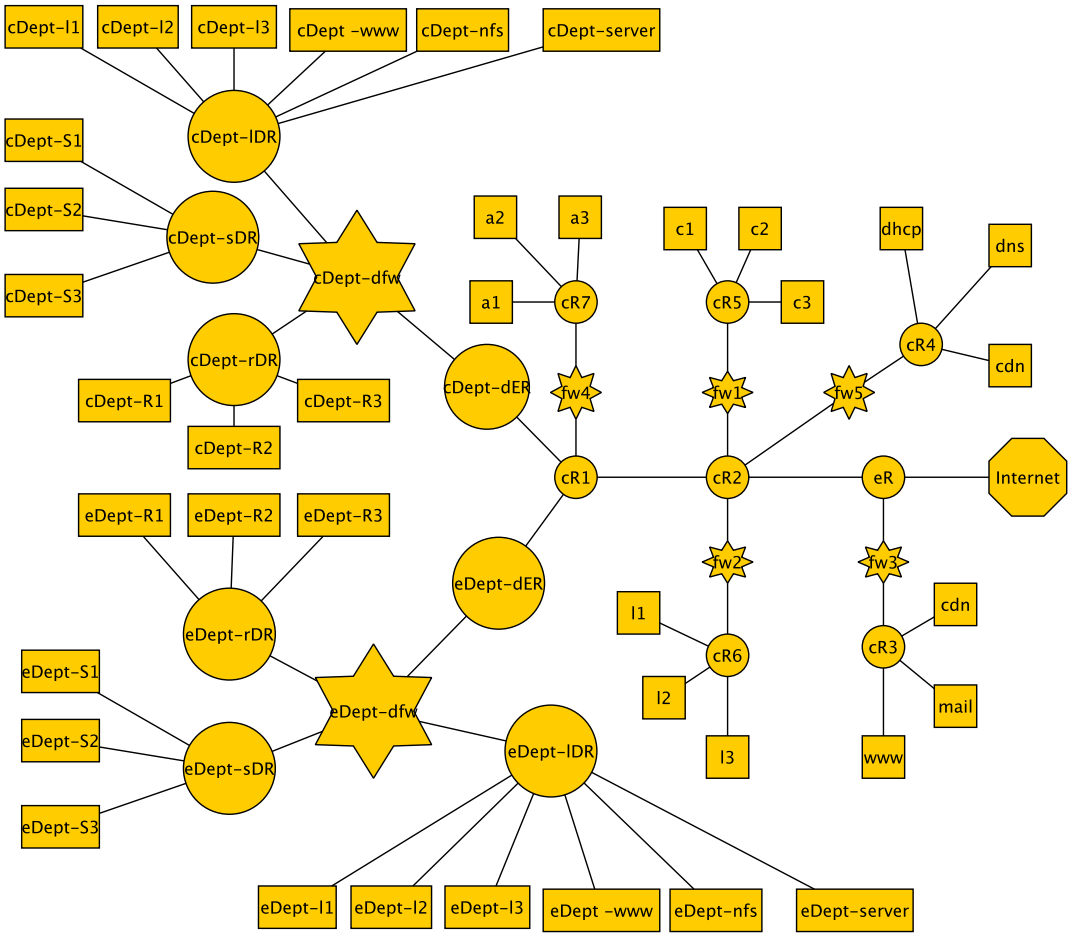

Figure 3.8: Final Rendered Hypothetical University Network Layout Diagram

3.5.2 Slices of Network Layouts

The second part of our abstractions in the network layout abstraction level looks at designing network layout slices that will be replicated as the high level network policy intention requires. This enables the replication of identical network layouts except for the allocation of IP addresses for the network devices in the proposed experiment. A good example of where slices of network layouts are used has to do with cyber security competitions abstractions. This is because all participating teams in such competitions need to have identical layouts so as to ensure a level playing field for all participants. A more comprehensive assessment of this network layout abstraction will be carried out in chapter 7.

3.6 Closing Remarks

This chapter of the thesis introduced how network concepts can be abstracted so as to make the network management process easier. The chapter introduced template

design approaches to abstracting firewall policies; inter domain routing policies and network layout abstraction. This chapter was written so as to show how network processes can be abstracted for the user's benefit. Many other network concepts such as tunnelling, datacenter and other services such as DNS, DHCP can be easily abstracted using techniques detailed in this section of the thesis. The layout diagram in Figure 3.9 below details how many other network concepts can be abstracted for network administrators. For example, policies such as all network traffic between the branch networks in the diagram and the HQNet will be encrypted using IPSEC or any other tunnelling protocols can be abstracted on a policy intention level using a between the realms and having a encryption property option.

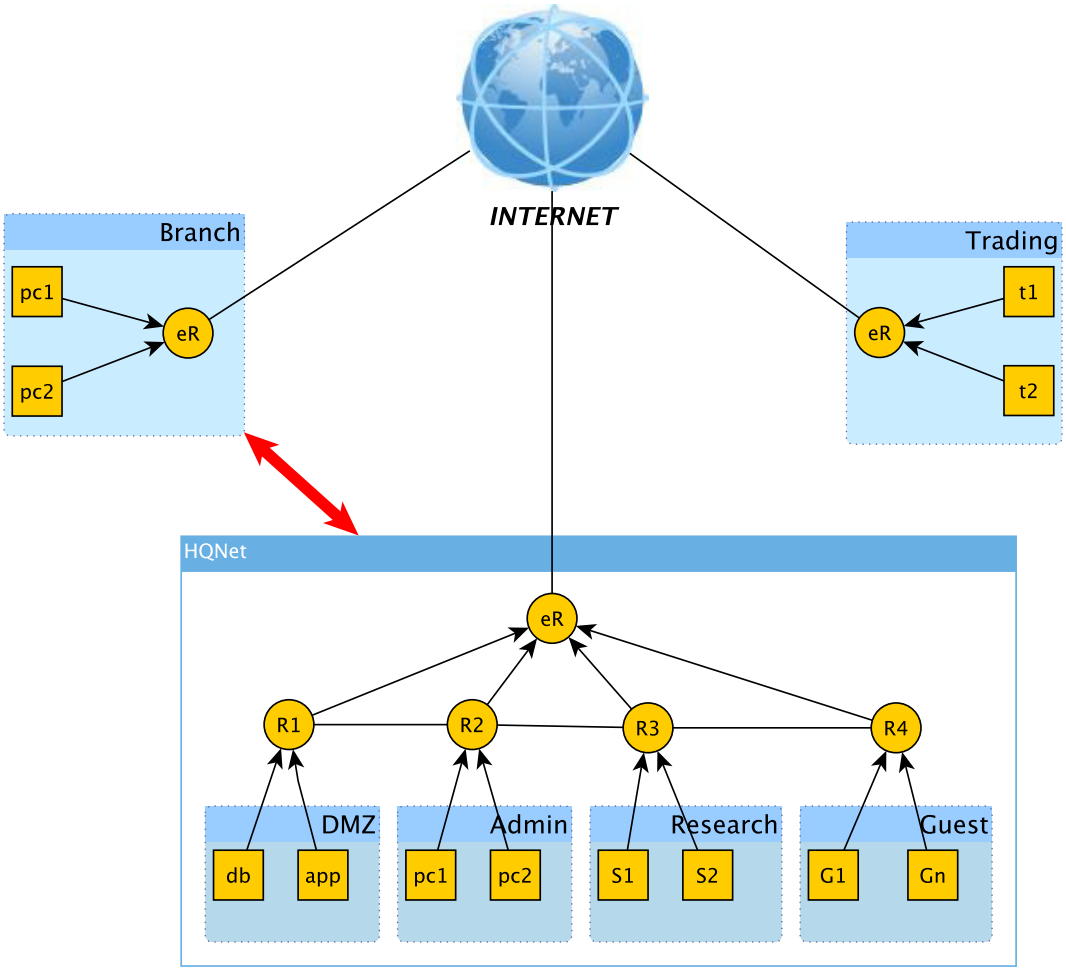

Figure 3.9: Illustrational Topology Showcasing Network Concepts

The next chapter of the thesis will be used to introduce and give an overview of the system we developed to evaluate the scalability, flexibility and limitations of the network policy abstraction techniques proposed.

Chapter 4

NePAS Framework Design

This section discusses the design consideration of a system that will be used to evaluate the abstraction techniques proposed for various network policies called Network Policy Abstraction System (NePAS).

NePAS is a framework that is used in compiling and deploying conflict free vendor-specific configuration from abstract high level network policy intentions and associated network topology graphs. The graphs that are generated for the high-level policy intentions are designed to be independent of the network layout devices. Likewise the graphs generated for the representation of the proposed network layout are independent of vendor-platform that will be used for compiling low-level configuration commands. Both the policy intention and network layout graphs are produced using graphical editors such as yED. The graphs to be generated use nodes and edges to express both the intended policy consideration and network layouts. The NePAS framework was developed using template design approach discussed in chapter 2.1.1. NePAS uses a combination of network policy template snippets and NIDB gotten from the various in order to generate Cisco VIRL configuration files that can be used to deploy the proposed network in Cisco VIRL. The NePAS framework is designed around five phases as follows:

- *Phase I - Policy Intention*: the first phase of NePAS is where the network policy intention is described independent of the network topology layout. Inter-domain routing policy intention, firewall access policy intention, cyber security exercise intentions are expressed in an abstract, simple, and easy manner using graphs in this case.
- *Phase II - Network Layout*: the second phase of NePAS is where the actual layout of the proposed network is represented in a vendor-independent manner using graphs. The various network policies designed in Phase I can then be assigned to various network devices in this phase of NePAS.
- *Phase III - Anomaly Resolution*: the third phase of NePAS is where the network layout devices with their respective high level policy intentions are passed through

various network anomaly resolution algorithms before low level configuration commands are compiled. This is to ensure the final deployments are conflict free and behave as intended by network administrators.

- *Phase IV - Compilation*: the fourth phase of NePAS is where a Cisco VIRL XML file with a set of conflict free vendor-specific device-centric configuration commands for all the devices of the proposed network are generated.

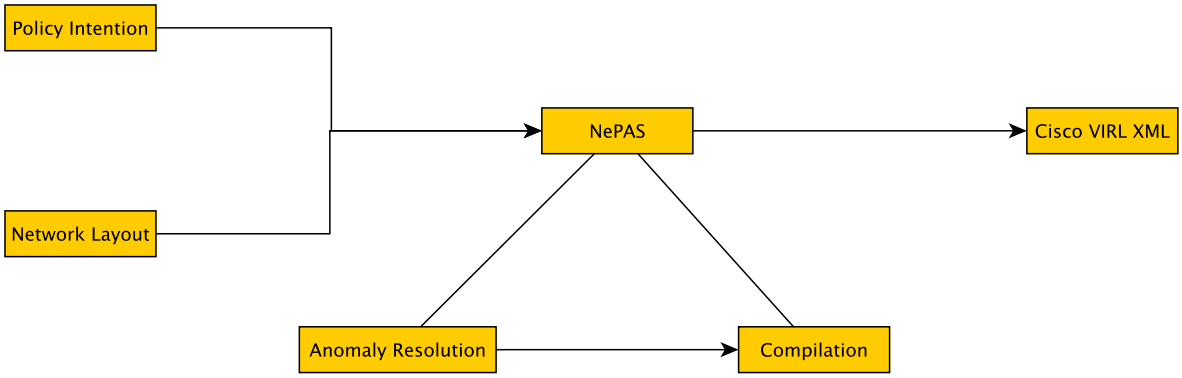

Figure 4.1: NePAS Framework

Figure 4.1 above illustrates the sequencing of phases of the system proposed in this thesis. A combination of high level network policy intention and a vendor-independent network layout are processed using template design approach so as to automatically generate Cisco VIRL configuration files that can then be deployed in VM Maestro workspaces. The following section provides and in-depth discussion of the design of the various phases of NePAS.

4.1 Policy Intention Abstraction Phase

This is the first phase of NePAS where the network policy intention of the proposed network is expressed. At this level of abstraction, network devices such as routers, firewalls, switches, and many other middle-boxes that are used in enterprise networks are in contention. NePAS is able to abstract that level of detail so as to give a network administrator the freedom of representing various network policy intentions such as firewall relationships, BGP routing policies or cyber security exercise approaches using graphical editors without thinking about how the actual layout of the network will look like or the type of devices that will be used for the proposed experiment. The following are the two major aspects used in specifying any type of network intention policy implemented by NePAS in this phase.

4.1.1 Realms

The nodes of a policy intention graph are used by NePAS to represent realms which can either be a single network device (such as a servers, firewalls, routers, etc.) or a group of devices that form a network (such Autonomous Systems, CDX teams or devices with identical access control policies) as the case maybe. It should be noted that a device or group of devices within a realm must have identical network policy intentions. Realms need to be labeled as their labels will be useful in the network layout phase of NePAS to be discussed in the following section of the thesis. The only reserved NePAS label for realms in a policy intention graph is the keyword 'any'. This label, *any*, has been reserved for expressing firewall relationships in NePAS and can only be used in that context. The use of the keyword 'any' for a purpose other than firewall relationships will be ignored by NePAS. Detailed explanation of the usage of the keyword *any* will be covered in Chapter 6.2. There are additional details needed to fully represent realms in a network policy intentions graph of a proposed network such details are referred to as realm options. *Realm Options* - are policy details that will be used on the nodes (or realms) of a given network policy intention graph to further express additional details of the proposed network experiment. For example a realm in a BGP network policy intention graph can have custom optional values that will allow a user to specify the autonomous system number. These custom attributes can either have default values while in other cases it is imperative to declare a specific value. This is to enable NePAS to abstract a lot of network policy specification details while providing a very flexible setup so as to express various levels of network abstractions.

4.1.2 Relationships

The edges that connect realms are used by NePAS to represent the network policy relationship between the two associated realms. The labels on the edges between realms state the exact network policy relationship between the two realms. For example in firewall intentions, the edges are either accept or deny to reflect the action that will be taken by the firewall rule. Edges at this level of abstraction do not have the typical default values that are popular with network graph abstraction which activates a default value if details have been omitted on edges in the graph. This is because NePAS uses the label on edges between realms to know what type of network policy relationship it is dealing with during the compilation process. Hence if any edge label is omitted intentionally or unintentionally, an error message will be generated automatically. There are additional details needed to fully represent realm relationships in the network policy intentions graph of a proposed network. The additional details are called edge or relationship options. *Edge Options* - are policy details that are used on the edges (or relationships) of a given graph to further express additional details of the network policy intention. Some

of these details such as the destination port of a firewall rule which represents the port numbers or service names can be abstracted and expressed in a very easy manner for the network administrator in the form of custom properties. For example, service with value *www* in a given firewall relationship can be used to represent web server running on port 80. These custom attributes can either have default values while in other cases it is imperative to declare a specific value. This is to enable NePAS to abstract network policy specification details while providing a very flexible setup so as to express various network policies.

4.2 Network Layout Abstraction Phase

This is the second phase of NePAS where the actual network infrastructure devices of the proposed network are expressed in a vendor independent manner. The network layout is expressed in a graphical format in a way similar to the policy intention detailed in the previous section. The abstraction at this level is similar to current automated configuration systems that use graphical editors for their input such as Autonetkit amongst many others. It should be noted that the abstractions proposed in this phase allow network administrators to express re-usable blocks and network slices as stated in the previous chapter. The level of abstraction in this phase of NePAS is lower than the policy intention abstraction as routers, switches and other network devices are featured here. The network policies defined in the previous section will be assigned to the various network devices as the proposed network dictates in this phase of NePAS. It should be noted that low level configurations such as IP addressing can still be abstracted at this level depending on the requirements of the proposed network. In this phase of NePAS, nodes within a graph are used to represent various network devices such as routers, firewalls, servers, switches and many others. The edges of a network layout graph are used to specify the logical structure (or link) of the connected network devices.

4.2.1 Network Devices

The nodes of a network layout graph in this phase are used to represent network devices such as routers, firewalls and servers amongst others. The label of a node in the network layout graph is used to denote the name of the network device during the low level configuration process. The nodes used to express various network devices in this phase are designed to have node options for additional properties. Node options are essential for expressing the network device and the policy group a node belongs to in the network layout abstraction phase of NePAS. There are three node options that have been provided for devices during the expression of the network layout in NePAS. These node options are as follows: (i) the type of network device the node represents within the graph (ii)

the firewall realm the network device belongs to from the policy intention graph supplied (iii) the BGP realm a router belongs to in the policy intention graph supplied.

The first node option used as a custom property for network devices in the network layout graph is called ***dtype***. This node option is used in specifying the type of network device NePAS will be dealing with. The optional a device type can take include server, router, kali and many others. The device type selected influences the disk image to be used during the low level configuration process as some servers have applications off until activated at this stage by selecting the appropriate node option. A very good example here are servers selected with various vulnerabilities during a cyber security competition. The option selected at this point also influences the type of network interface type to be used for low level configuration of the various device types in the network. The interface used by servers is different from the ones used by network devices such as routers or firewalls.

The second node option used as a custom property for network devices is called ***fwpol***. This option is designed to indicate the firewall policy intention realm a network device belongs to in the policy intention graph. The value selected in this phase is used in configuring the firewall policies of the various network devices in the proposed network. It should be noted that devices such as switches, firewalls and routers are not expected to have firewall policies. Hence NePAS has been designed to ignore firewall policies assigned to such devices during low level configuration command compilation. A detailed description of how this node option is to be implemented is carried out in chapter 6 of this thesis.

The third and last node option used as a custom property designed in this phase for network routers is called ***bgppol***. This option has been designed to indicate the BGP policy intention realm the network router belongs to in the policy intention graph. This option determines the autonomous system a router belongs and the type of BGP relationship it has with other routers in the proposed network as expressed in the policy intention graph. It should be noted that only routers are designed to have this option and if a server or switch is assigned this value, NePAS will ignore it during processing. A detailed description of how this node option is to be implemented is carried out in chapter 5 of this thesis.

4.2.2 Links

The edges of a network layout graph are used to specify the logical structure between the two connected network devices. The edges of a network layout graph do not make use of its label and hence will be ignored if specified during the low level configuration process. The edges between two nodes in a network layout graph has been designed with two custom edge options. These set of options have to do with specification of

low level IP addressing for the network devices connected by the edge. This is so as to give network administrators greater abstraction flexibility when deploying a proposed network. The two edge options designed in this phase can either be manually assigned or generated and allocated to connected devices are called ***sIP*** and ***dIP***. The two options are designed by default to take 'NULL' values and hence a pair of IP addresses will be generated automatically.

4.3 Network Anomaly Resolution Phase

This is the third phase of NePAS where the policy intention and network layout graphs are combined and passed through various anomaly resolution algorithms. This phase is designed to resolve anomalies that arise after the low level configuration process. During the merger of the policy intention and network layout graphs, a number of anomalies can come up that will impact the final deployment of the proposed networks. The various network policy anomaly resolution algorithms designed in this phase ensure final networks deployed do not suffer issues of network connectivity, performance and security lapses amongst others. Network policy anomalies that are handled are those that occur during the generation of network configuration commands. Anomalies that occur after the proposed network has been deployed due to network changes are out the scope of our research and are not handled.

4.4 Compilation Phase

The fourth and last phase of NePAS is designed to generate a set of vendor specific policy configuration commands for all devices of the proposed network. NePAS is designed to use template design approach to generate the low-level vendor specific configuration commands that will be used for configuring the devices of the proposed network with their respective network policies supported by NePAS. Using the template design approach of automated configuration, a number of snippets have been developed for NePAS to generate the appropriate low-level configuration commands of the network policies for the various devices within the proposed network. The configuration snippets will be using the output from the anomaly resolution section such as IP addresses and also handles the ordering of filters so as to generate consistent and anomaly-free low-level configuration commands for the proposed network. Finally, NePAS will generate an XML file that can be easily exported into Cisco VIRL for the deployment of the proposed experiment. The file generated is designed to be used for deployment of the proposed network in Cisco VIRL with configuration snippets for the high level network policy intentions.

During the compilation process, NePAS generates a number of log messages that is used to show what is going on behind the scene and reporting the progress. These log messages can be broadly categorised into three groups: information messages; warning messages and error messages.

4.5 Closing Remarks

This chapter of the thesis gives a detailed overview of the system developed to evaluate the limitation of the network policy abstractions that are proposed for our research project. The system developed is composed of four phases: policy intention; network layout; anomaly resolution and compilation. The next chapter will be used to introduce the concept behind inter domain routing protocol, Border Gateway Protocol (BGP). The chapter will move on to discuss the network policy abstraction that we have developed for BGP business relationships.

Chapter 5

BGP Abstraction

5.1 Background

Internet Service Providers (ISPs) interconnect with each other using routing protocols in order to provide global connectivity to their respective customers based on a set of routing policies. A routing policy defines how routing information is exchanged between ISPs. Routing protocols are used to find the best path between two nodes either within an ISP network or between multiple ISP networks. There are two categories of routing protocols used by ISPs to provide connectivity to their customers, these are as: internal (or intra domain) routing protocols and external (or inter domain) routing. The research done in this project focuses on external routing protocol Border Gateway Protocol (BGP) and hence internal routing protocols will not be discussed. BGP is the standard protocol used for inter domain routing. BGP allows an ISP to choose a routing policy that will be used in (i) selecting the best route to any given destination, (ii) advertising reachability of prefixes and (iii) accepting routes from neighbouring ISPs.

An autonomous system (AS) is a network or group of networks whose prefixes and routing policies are under a common administration. Autonomous systems help reduce huge routing information by reducing the number of entries of networks. Autonomous systems interconnect at dedicated points of presence(POPs) or public Internet Exchange Points (IXPs)[82]. An IXP is a collocation crafted with networking equipment where participating ASes can connect to each other.

An autonomous system is identified using a unique 16-bit number called an autonomous system number. In recent years projects such as [98] are proposing a 32-bit autonomous system numbers The details of this project is beyond the scppe of the research conducted in this thesis and hence will not be discussed. Autonomous system numbers can be categorised into private and public autonomous system numbers. Private autonomous system numbers range between 64,512-65,534 and are used by organisations to implement BGP within their network. Public autonomous system numbers on the other

hand range between 1-64511 and are used to connect multiple autonomous systems together. Autonomous system numbers are not needed for networks that have only one block of IP addresses (single prefix) or a single connection to their ISP[59]. Figure 5.1 below shows a hypothetical network of autonomous systems with their corresponding AS numbers.

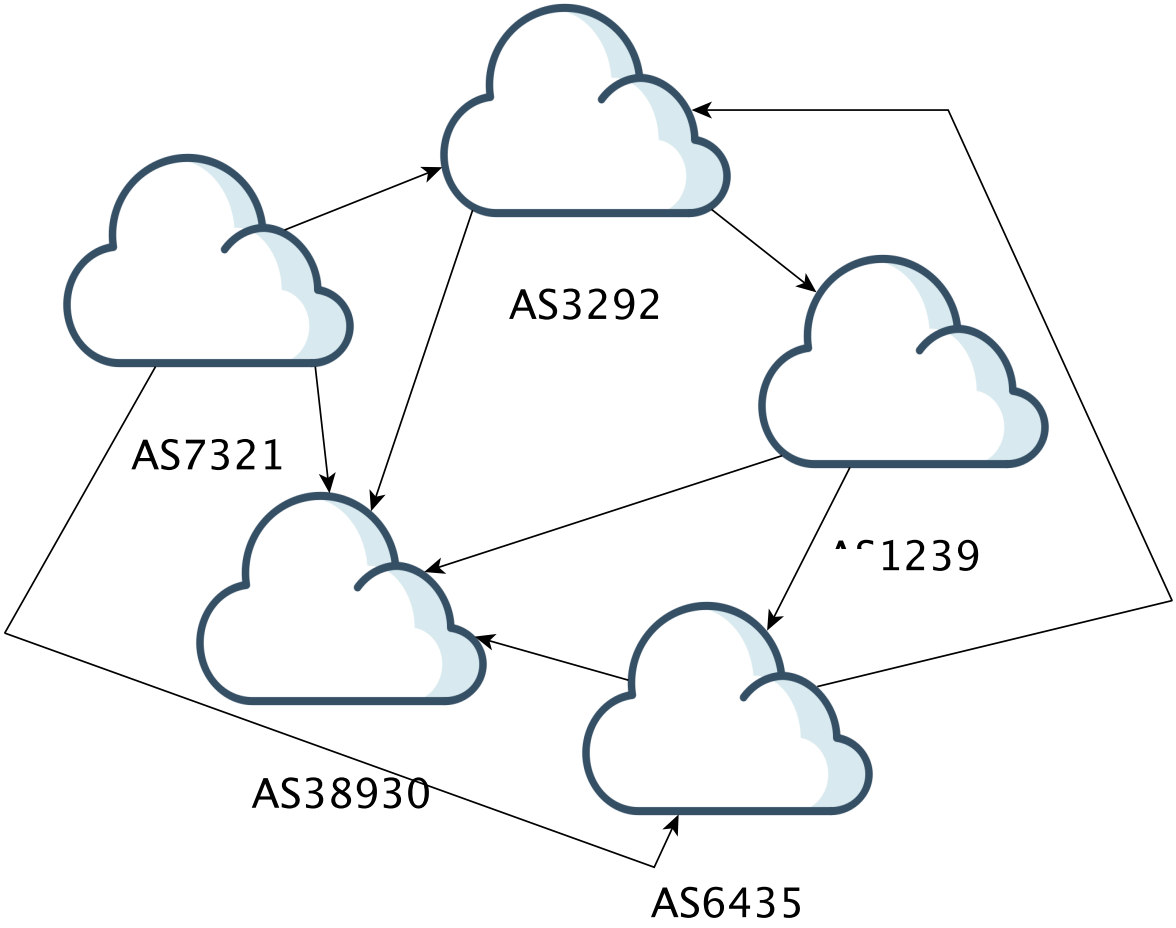

Figure 5.1: Typical BGP Network

BGP as stated earlier is a protocol used by autonomous systems to decide how to get traffic from a user within an AS network to another user in a different AS network. BGP is a path vector (and also distance vector) protocol featuring a set of complex metrics used in exchanging network reachability information within an AS network and between multiple autonomous systems. BGP being a path-vector protocol, allows an AS to add its AS number to the beginning of the AS path before advertising the route to the next AS[22]. In order to reduce the size of the BGP routing tables and improve scalability, BGP can supernet an address block. BGP is concerned with exchanging routing information between autonomous systems in a scalable manner.

BGP is an application layer protocol that works over TCP on port 179 and is used to allow autonomous systems to implement their respective routing policies for selecting and propagating IP prefixes. Network router peers need to have BGP configured and form a TCP connection in order to begin exchange of routing information. Once a TCP connection between the two BGP router peers is active, a router can then send or receive list of routes from configured neighbouring peers. The router will scrutinise the routes to be sent or received using a route selection algorithm specified by the AS routing policy. BGP route updates are stored in a Routing Information Base (RIB) with only one route per destination stored. BGP routers do not send periodic routing updates because BGP is an incremental protocol and hence only changes to routing table information are exchanged. The route configuration process with BGP have two types of messages upon initialisation, these are - update messages[63] and keep alive messages[34].

BGP used within an autonomous system is referred to as Internal BGP or iBGP. BGP routers use internal BGP sessions to communicate within an AS. This is essentially used to distribute routes learned from external autonomous systems among border routers[33]. These sessions are not IGP-like and cannot be used in isolation for routing between nodes within the autonomous system. Internal BGP sessions provide a way in which routers inside an autonomous system can use BGP to exchange information about external routes. These sessions are routed via whatever IGP is being used within the autonomous system. The first router to introduce a route into internal BGP sets the next hop attribute to its loop-back address and all other internal BGP routers within the AS preserve the setting. IGPs do not scale as well as internal BGP and cannot implement the rich attributes present in BGP and hence IGP cannot be used in place of internal BGP[42]. The scalability of internal BGP is limited due to the requirement that internal BGP routers be connected via a complete mesh[64]. The technique used in improving the quadratic scaling of internal BGP sessions include: Route reflectors[20]; and Confederations[39]. The above techniques are out of the scope of this research and hence will not be discussed.

BGP used between autonomous systems is referred to as External BGP or eBGP. External BGP sessions are used between BGP speaking routers in different autonomous systems. This is the standard mode in which BGP is used. The external BGP sessions operate over a one-hop IP path. An external BGP router must disseminate routes of external prefixes to all the other routers in the autonomous system[16]. This external BGP dissemination must meet two important goals: (i) loop-free forwarding: routes picked by all routers should be free of deflections and forwarding loops. (ii) complete visibility: BGP should allow each autonomous system to be treated as a single monolithic entity. The external BGP speaking routers within an autonomous system must exchange external route information so that they have a complete view of all external routes.

Listing 5.1 below shows a sample Cisco BGP configuration based on the business

relationships depicted in Figure 5.1 above.

Listing 5.1: Basic Cisco BGP Configuration for AS6435

```
! (this is a comment)
router bgp 6435
neighbor 10.0.0.6 remote-as 6435
neighbor 10.0.0.6 description TATA-iBGP-Router
neighbor 20.0.0.1 remote-as 1239
neighbor 20.0.0.1 description eBGP-to-Sprint
neighbor 30.0.0.1 remote-as 38930
neighbor 30.0.0.1 description eBGP-to-FiberRing
neighbor 40.0.0.1 remote-as 3292
neighbor 40.0.0.1 description eBGP-to-RDC
neighbor 50.0.0.1 remote-as 7321
neighbor 50.0.0.1 description eBGP-to-TATA2.0
!
```

5.1.1 BGP Path Attributes

BGP path attributes are the characteristics an advertised BGP route uses to influence the autonomous system's routing policies. BGP attributes can be set in the following three possible locations: locally; neighbour and neither. BGP path attributes are categorised into either (i) well-known or (ii) optional attributes[23].

Well-known attributes: the attributes under this category must be recognised during every update message to peers. The well-known attributes can be further broken down into two: *Mandatory (also called well-known mandatory)* are attributes that are always included and carried in both iBGP and eBGP update messages to peers. *Discretionary (also called well-known discretionary)* are attributes that may or may not be included in BGP update messages. The network administrator decides whether or not to use these attributes on update messages to its peers.

Optional attributes: the attributes under this category are not required or expected to be supported by all BGP implementations. The optional attributes can be further broken down into two: *Transitive (also called optional transitive)* are attributes that are recognised by some BGP speaking routers but not all. However if the attributes transitive flag is set, the attribute should be accepted and advertised to its other BGP peers. *Non-transitive (also called optional non-transitive)* are attributes that can be ignored if their transitive flag is not set. A BGP speaking router that receives an update from its neighbouring peer with this attribute can ignore the attribute if its transitive flag is not set and the router will not advertise the attribute to its peers.

Attribute Name	Category
Next_Hop	Well-known mandatory
Local_Preference	Well-known discretionary
Origin	Well-known mandatory
AS_Path	Well-known mandatory
Multi_Exit_Disc (MED)	Optional non-transitive
Community	Optional transitive

Table 5.1: BGP Path Attributes and Categorisation

Table 5.1 above shows the BGP path attributes discussed in this research and their various categorisation. The following section discusses some of the major BGP path attributes supported by BGP-enabled routers in a network.

- Next Hop[83]
  The next hop address specifies the IP address that will be used to reach the destination being advertised. The following set of rules are used to govern various BGP implementation scenarios on using the next hop address. (i) A BGP speaking router advertising a route in an update message uses its address as the next hop IP address if the peer it is sending the update message is in a different AS. (ii) A BGP speaking router will use its address when advertising a route that is within the AS in an update message to a router within the same AS. (iii) A BGP speaking router that is advertising a destination network in another AS in an update message to a BGP speaking router within its AS will use the IP address of the external BGP speaking router that sent the advertisement to the AS.
- Origin[11]
  The origin attribute is a well-known mandatory attribute generated by a BGP speaking router that originates reachability information. This attribute is used to indicate how BGP learned about a particular route. There are three possible values the origin attribute can have, these are: 0 refers to an update originating from IGP; 1 refers to an update originating from EGP; and 2 for INCOMPLETE, when a route originates from another routing protocol instead of BGP.
- Weight[87]
  This is a Cisco-defined attribute that is applied locally within a router. This attribute is not advertised to neighbouring router peers. The value of a weight ranges between 0 and 65,535 and used when a router learns about more than one route to a destination. The route with the highest weight value is preferred over others. All routes learnt from a BGP peer have values of 0 and all routes generated by a local router have values of 32,768 by default. Figure 5.2 below shows a sample network that employs a weight attribute on the router in AS1239. In the

figure, the traffic destined to AS3292 will be routed via AS6435 because the BGP session has a higher weight value compared to the path via AS38930.

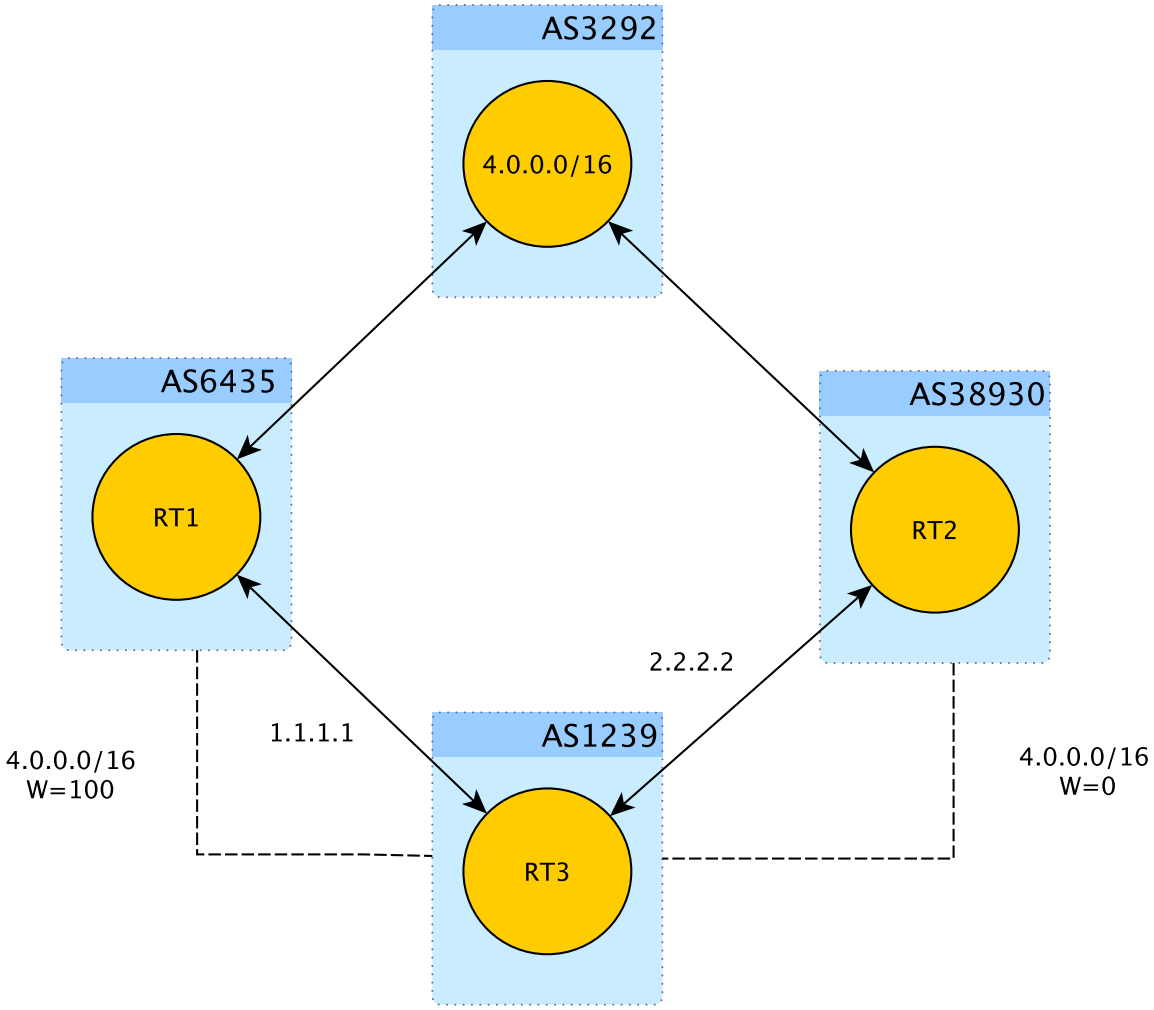

Figure 5.2: Example of a BGP Network Employing Weight Attribute

Listing 5.2 below shows an excerpt of how to configure Cisco Weight attribute using the neighbor command. This is one of three methods used in configuring Cisco weight attribute.

```
RT3#
router bgp 1239
neighbor 1.1.1.1 remote-as 6435
neighbor 1.1.1.1 weight 100
!--- The route to 4.0.0.0/16 via AS6435 has a 100 weight.
neighbor 2.2.2.2 remote-as 38930
neighbor 2.2.2.2 weight 0
!--- The route to 4.0.0.0/16 via AS38930 has a 0 weight.
```

Listing 5.2: Using neighbor Command to Configure Cisco Weight

- Local Preference[52] The local preference attribute is included in all update messages a BGP speaking router sends to other internal peers. Local preference is the internal cost of a destination used to ensure AS-wide consistency. It is used to prefer a particular exit point from the local AS. It is propagated throughout the local AS and used to select the exit point of a specific route. This attribute will be ignored if it is included in an update message to external peers except in BGP confederations. The route with the highest local preference value is preferred over all others and is used to ensure revenue generating routes are preferred over expensive routes. Figure 5.3 below shows a sample network that employs a local preference attribute. In the figure, the traffic from AS38930 will exit via AS1239 because that route has a higher local preference value compared to the path via AS6435.

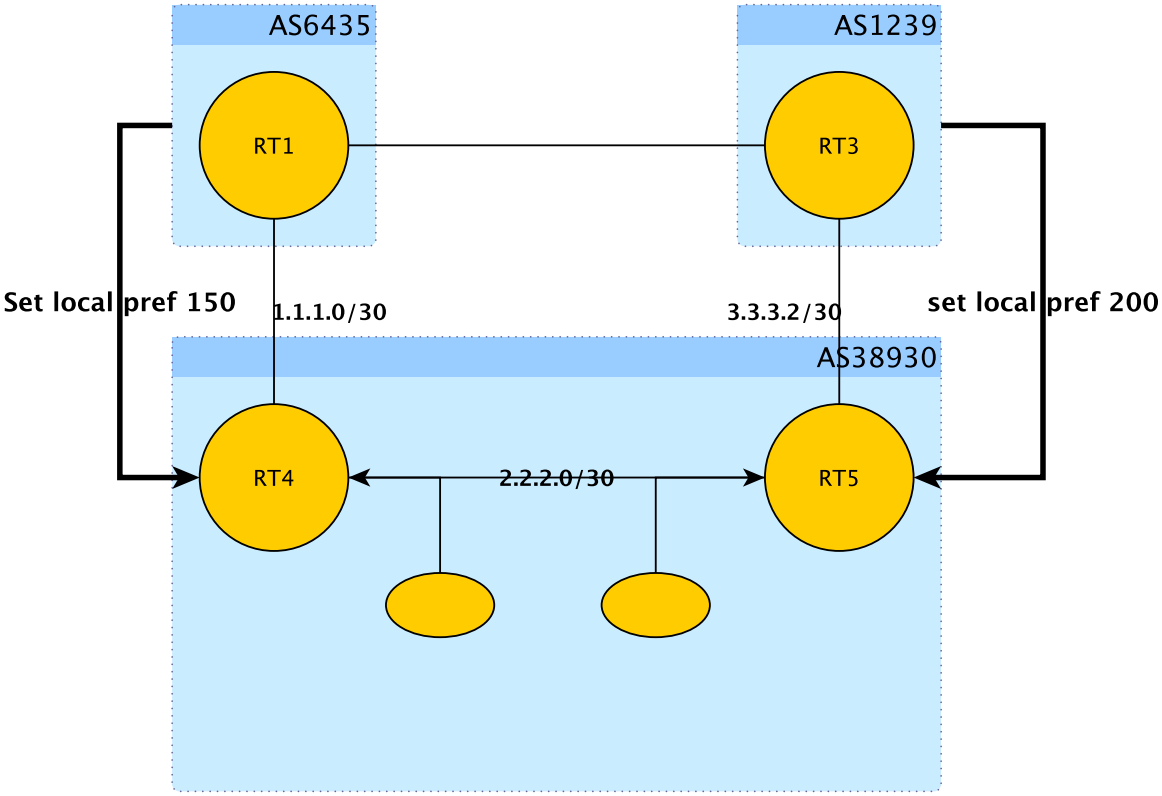

Figure 5.3: Example of a BGP Network Using Local Preference

Listing 5.3 below shows an excerpt of how to configure local preference attribute on a Cisco router.

Listing 5.3: Excerpt Cisco Local Preference Configuration

```
RT4#
router bgp 38930
neighbor 1.1.1.1 remote-as 6435
```

```
neighbor 2.2.2.1 remote-as 38930
bgp default local-preference 150

RT5#
router bgp 38930
neighbor 3.3.3.3 remote-as 1239
neighbor 2.2.2.2 remote-as 38930
bgp default local-preference 200
```

- AS Path[60]
  The AS-Path attribute is used to identify the autonomous systems through which the reachability information in the update message have passed. Every BGP speaking router when advertising a route to its neighbour will prepend its AS number to an ordered list of AS numbers called the AS-Path attribute. The subsequent BGP speaking routers that receive this update will also prepend their own AS numbers to the list. The shorter AS-path lengths are preferred over longer ones. AS-path is a mechanism through which BGP uses to achieve the following two purposes: (i) routing loop avoidance; (ii) help pick suitable path amongst multiple choices. Figure 5.4 below shows a sample network that employs an AS Path attribute.

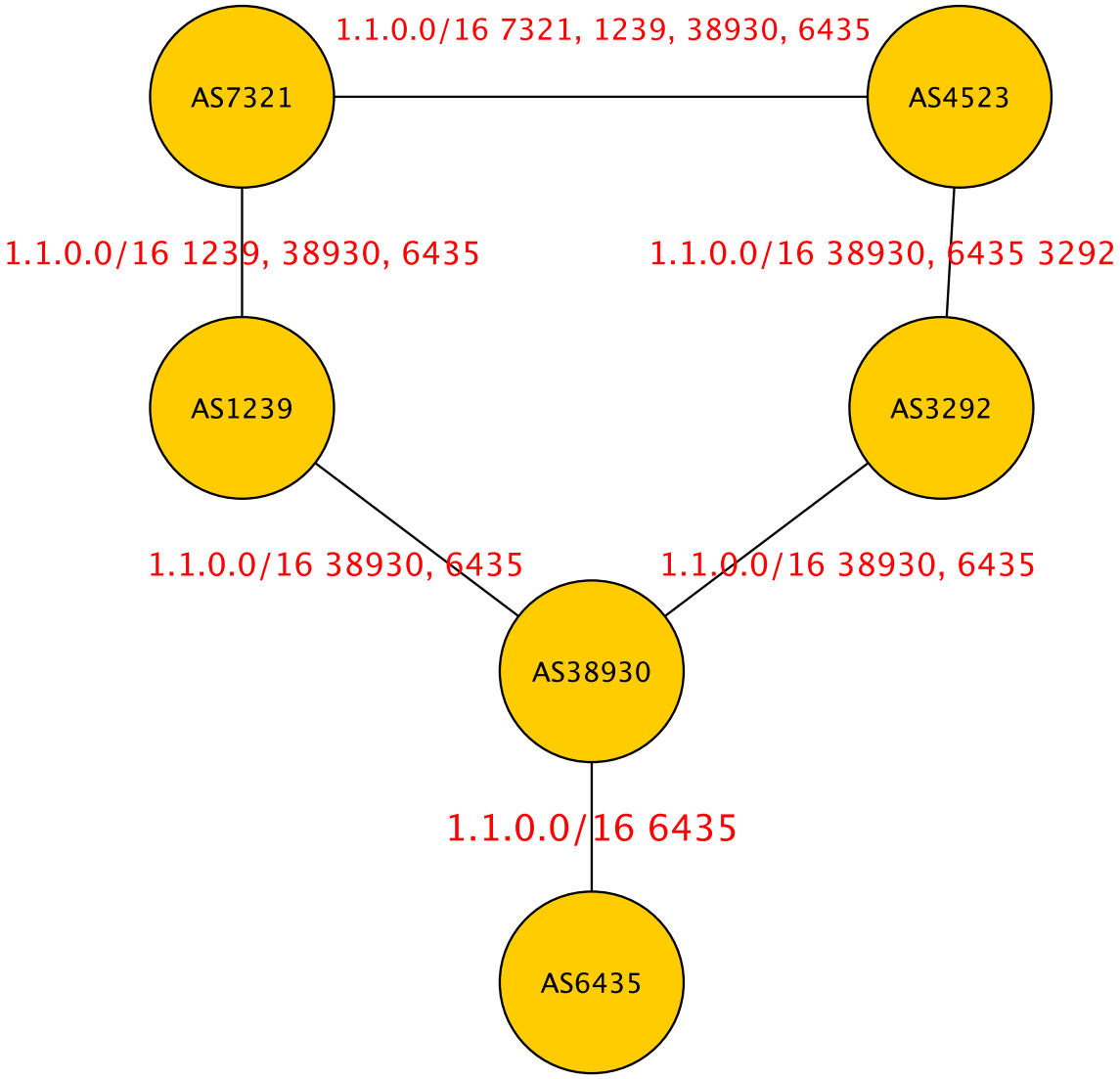

Figure 5.4: Example of a BGP Network Showing AS Path Attribute

- Multi-Exit Discriminator[53]
  The MED attribute is used by an AS to suggest the preferred entry or exit point into an autonomous system when it has multiple options to its eBGP peers. The MED attribute allows an AS with multiple entry or exit points to influence an entry or exit point that is preferred by a neighbouring AS. The value of the MED attribute is a four octet unsigned number called metric. The exit or entry point with the lowest metric is preferred over others. This attribute is propagated over internal BGP to other BGP speaking routers within the AS when it is received over external BGP. The MED attribute however cannot be propagated by the receiving AS because it is used to influence traffic between the two connected autonomous systems. Figure 5.5 below shows a sample network that employs a Multi-Exit Discriminator attribute.

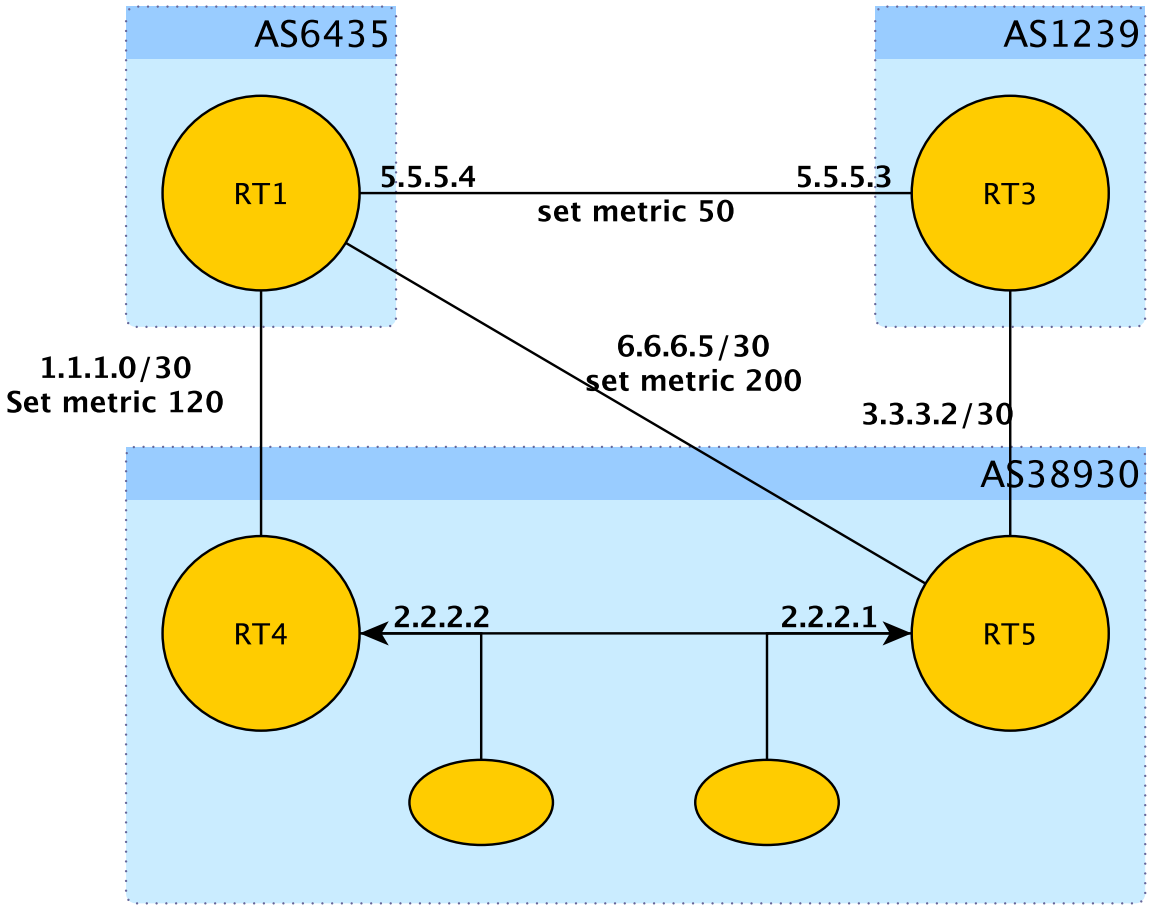

Figure 5.5: Example of a BGP Network Using MED Attribute

Listing 5.4: Excerpt Cisco MED Configuration

```
RT1#
router bgp 6435
neighbor 1.1.1.2 remote-as 38930
neighbor 6.6.6.6 remote-as 38930
neighbor 5.5.5.3 remote-as 1239
bgp bestpath as-path ignore
....
RT4#
router bgp 38930
neighbor 1.1.1.1 remote-as 6435
neighbor 1.1.1.1 route-map setmetricout out
neighbor 2.2.2.1 remote-as 300

route-map setmetricout permit 10
set metric 120
....
RT5#
router bgp 38930
neighbor 6.6.6.7 remote-as 6435
neighbor 6.6.6.7 route-map setmetricout out
neighbor 2.2.2.2 remote-as 38930
```

```
route-map setmetricout permit 10
set metric 200
....
RT3#
router bgp 1239
neighbor 5.5.5.4 remote-as 6435
neighbor 5.5.5.4 route-map setmetricout out

route-map setmetricout permit 10
set metric 50
```

- Community[37]
  The community attribute provides a way of grouping destinations that usually have similar routing decisions applied on them. This attribute tries to simplify policy enforcement by grouping BGP peers with common properties to share a set of routing policy. Community attribute is a set of four octet value represented using the hex format always. According to Chandra and Traina [25], the first two octets are the AS-number and the last two octets are an administratively defined identifier. An AS can use the community attribute to set the local preference and MED attribute discussed above instead of setting these attributes individually. Route maps are generally a good way of implementing the community attribute. There are four major well-known community values, these are as follows:
  (i) No-Export: routes received with this community value cannot be advertised to external BGP peers. The only time routes received with this community value are exported to external BGP peers is when using confederations. It should be noted that these routes cannot be advertised outside the confederations.
  (ii) No-Advertise: routes received with this community value cannot be advertised at all. Routes received here cannot be advertised to both internal and external BGP peers.
  (iii) Internet: this is the default community value and routes that belong to this community value by default can be advertised freely.
  (iv) Local_AS: routes received with this community value cannot be advertised to external BGP and peers in other autonomous systems within a confederation.

BGP routers use a three-step process to implement the routing policy that has been developed for the autonomous system. The three step process are as follows[23]:
(1) **Import Policy** - determines a BGP speaking router's decision on whether or not to filter and/or manipulate the attribute of a route received from its BGP speaking router peer. BGP import policies can be used for moving a portion of traffic within a network from one link to another (traffic engineering) based on prefixes and attributes in the route advertisement. An AS can use import policies to transform incoming route updates which include: deny an update; permit an update; assign local preference to indicate the favourability of a path.

(2) **Decision Process** - a BGP speaking router applies the decision process to select the best route for each prefix. The BGP decision process is a set of steps that the router uses to select a route from a set of routes it learns from its neighbours. The ordering of attributes allows the network administrator to influence various stages of the decision process and engineering within and outside their network. The ordering of attributes in the decision process also simplifies the routing policy of an organisation and makes it easier to predict configuration changes. BGP has a 12-step decision process algorithm used for selecting the route to a particular prefix. Table 5.2 below enumerates the various BGP decision making steps.

Step	**Attribute**
1	The update message is dropped if the next hop of the path specified is inaccessible
2	The route with the highest administrative weight is preferred
3	The route with the highest LOCAL_PREF is preferred if the weights are equal
4	The route that was generated locally is preferred over external routes if the LOCAL_PREF are the same
5	The route that has the shortest AS-PATH is preferred if there are no locally generated routes
6	The route with the lowest origin type is preferred if all the routes have the same AS-PATH length. The order of the origin is: IGP < EGP < Incomplete
7	The route with the lowest MED attribute is preferred if the origin codes are the same
8	EGP routes are preferred over IGP routes if the MEDs are equal
9	The route through the closest IGP neighbour is preferred if the path is equal at this point
10	The oldest route or the route that was learnt first is preferred if the path is still the same
11	The route with the lowest BGP Router ID is preferred if the path is still the same at this point
12	The route on the lowest interface IP address is preferred if the Router ID is the same

Table 5.2: BGP Decision Making Steps

(3) **Export Policy** - an export policy determines whether or not a BGP speaking router will advertise a prefix to neighbouring autonomous systems. Export policies use the following methods to make this decision: permit or deny route; assign MED value;

add community value; prepend its AS number one or more times to the AS path. The business relationship between the autonomous systems largely determines the export policies of an AS. Export policies are very important because no AS wants to transit network traffic that it is not somehow making money on. An AS should hence advertise routes with care because packets flow back using the best route advertisement to it.

There are three ways in which BGP can be used to control import or export policies according to Caesar and Rexford [23], these are filtering, preference and tagging. (i) Filtering eliminates certain routes from consideration and also controls who they will be exported to,
(ii) Preference influences which BGP route will be chosen for each destination prefix,
(iii) Tagging allows a network administrator to associate additional state with a route used to coordinate decisions made by a group of routers in an AS using community attribute.

5.1.2 BGP Business Relationships

The seminal model of routing policies between autonomous systems was developed by Gao and Rexford[46]. The Gao-Rexford (GR) model preference states that autonomous systems use local preference attribute to prefer customers routes over provider or peer routes and peer routes are preferred over provider routers. Figure 5.6 below shows a hypothetical example of a high level BGP relationship using the GR model5.1.

The appeal for the GR model lies not only in its simplicity but also in the stability of BGP if the model is followed adequately[45]. BGP is guaranteed to converge to a stable routing outcome if all autonomous systems strictly adhere to the GR model. The GR model proposes three BGP business relationships: transit, peering and sibling relationships. Over the years however, a lot of research projects such as [69] have expanded on the GR model and inferred additional BGP business relationships. The BGP business relationships discussed in the following section is a combination of the GR model and many other research projects developed over the years. The BGP business relationships discussed for this research thesis can be broadly categorised into five: transit; peering; sibling; hybrid and backup relationships.

5.1.2.1 Transit Relationship

This BGP business relationship is also referred to as provider-to-customer or customer-to-provider relationship[74]. This relationship occurs when a large autonomous system (the provider) agrees to carry the network traffic of a smaller autonomous system (the customer). The provider typically meters the traffic on each link and charge a transit fee for services rendered. The transit fee is typically based on reservation made upfront for the amount of traffic measured in megabits per second (mbps). Customers enter into

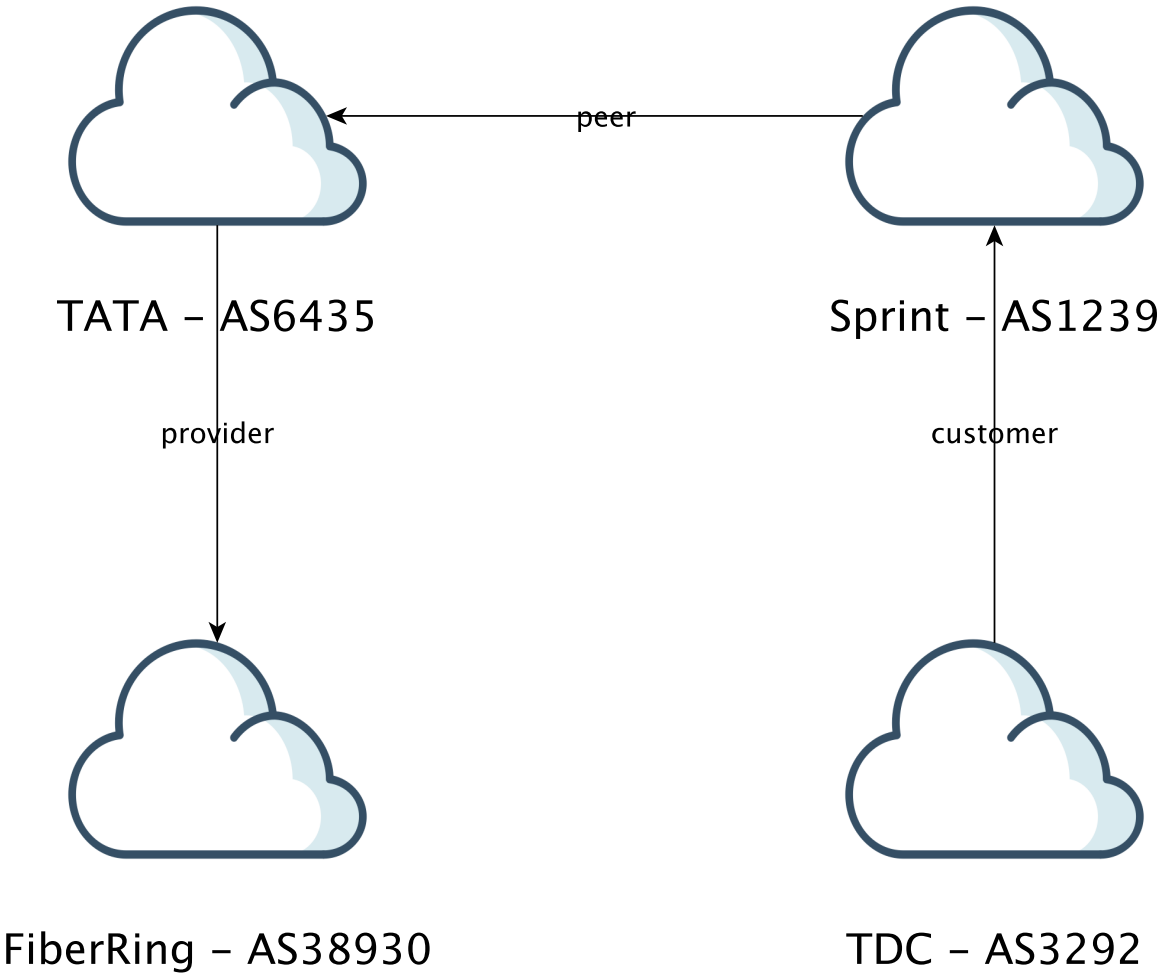

Figure 5.6: Example of a BGP Network With Various Business Relationships

a transit agreement to be globally reachable. The situation where a provider gives a customer access to all destinations in its routing table is called **full transit**. In a **partial transit** relationship, the provider can provide access in one of three ways: (i) within a limited geographical scope. In this scenario, a multi-homed customer may use a community value to instruct a national provider to serve traffic destined in the same country while an international provider serves international traffic. (ii) the provider only provides access for a set of prefixes to the customer. (iii) the provider only provides access to its peers and customers but does not provide access to its provider prefixes.

It should however be noted that providers make more from a customer if the amount of traffic sent on their behalf increases. This is one of the major bases for a large number of complex routing policies. The provider should advertise routes to its customers to as many other connected autonomous systems as possible in order to make more money. An ISP should prefer advertisements made from its customer over all other choices (over peers or providers).
Exporting to a Customer: an AS can export its routes and customer routes as well as provide or peer routes. This is so the provider can make more money.
Exporting to a Provider: an AS can export its routes and customer routes but usually does not export its provider or peer routes. This is because exporting peer and provider routes will make the provider transit more traffic and hence charge the customer more transit fee.

5.1.2.2 Peering Relationship

This BGP business relationship is also referred to as peer-to-peer relationship[81]. This relationship occurs when two or more similar sized autonomous systems establish a mutually reciprocal agreement where their network traffic pass through each others' networks without charging any fee. In a peering relationship, an AS can export its routes and customer routes but does not export routes learned from other peers and providers so as not to transit the traffic and pay more money. Peering happens at a location that is most convenient for both autonomous systems. There are two locations where peering happens, these are public peering and private peering. (i) Public peering at an Internet Exchange Point (IXP) - this location of peering allows anyone to become a member and in some cases, a fee might be charged. (ii) Private peering via a direct connection - this usually involves a service level agreement (SLA) and most of the time comes with better hardware to handle network traffic. Private peering provides a convenient way to establish custom solutions such as trunking two links together.

Peering agreements are usually confidential and require both parties to sign a non-disclosure document. In a peering relationship, an AS should export only selected routes from its routing table to other peer autonomous systems. An AS in a peering relationship

can export its routes and customer routes but usually does not export its provider or peer routes. There are times when two peering links which together function as one virtual peering link, called indirect peering relationship. An indirect peering relationship occurs when two autonomous systems are peering with the same route server at an IXP such that they gain access to each other's network as if they have a peering link. There are three major reasons for peering relationships: (i) Peering between tier-1 ISPs ensure that they have explicit default-free routes to all Internet destinations, (ii) Peering relationships is a way for ISPs to save money of not having to buy more transit services. Also as they do not have to pay any fee for peering and hence save money, (iii) Peering relationships have the advantage that a more direct path would lead to better end-to-end performance for their customers.

5.1.2.3 Sibling Relationship

This relationship usually occurs when an organisation owns multiple autonomous systems and have export policies involving each ASs' peers, providers and customers[15]. In a sibling relationship, an AS can export its routes, customer routes as well as provider and peer routes.

5.1.2.4 Hybrid Relationship

These BGP business relationships occur when two ASes agree to have a combination of two business relationships, usually a peering and transit[49]. Hybrid relationships usually occur between tier-1 and tier-2 autonomous systems that are well connected. There are two major categories of implementing hybrid relationships: IP version and Location dependent.

IP version Dependent: the routing policies and path for IPv4 traffic differ significantly from IPv6 traffic in this business relationship. The autonomous systems negotiate separate relationships for the different IP version prefixes.

Location Dependent: in these BGP business relationships, two autonomous systems collocate at more than one private network access point (NAP) or Internet Exchange Point (IXP). The BGP business relationship at each collocated point is different from the other.

Some hybrid relationships are dependent on both IP version and location. This occurs when two autonomous systems at an IXP have a relationship for an IP version while at a NAP they have another IP version relationship. This is usually achieved by tagging the same link with different sets of community values in different BGP Update messages. It should however be noted that setting dual meanings for a community value is not a good practice.

5.1.2.5 Backup Relationship

This relationship occurs when two autonomous systems set up a link between their networks and is only used when the primary link becomes unavailable due to failure[50]. Backup links are invisible and do not carry any traffic until there is a failure on the primary link. This is usually a good practice especially when an AS has limited connectivity to the rest of the Internet. Routes involving backup link should have a lower local-preference over other routes. BGP advertisement can use community values of NO-EXPORT and NO-ADVERTISE to instruct providers not to advertise the routes to customers or anyone. Backup links can also be implemented by prepending or appending the AS-path attribute such that the link is artificially longer.

5.1.3 BGP Anomalies

The existence of contrary routing policies that are at odds with the network administrator's intention is called a routing anomaly. Routing anomalies occur regularly in networks and last between a few seconds to several months before they are noticed and corrected. Routing anomalies occur due to mistakes during either the configuration process or convergence of the network post deployment. This section discusses the routing anomalies that are caused by router misconfigurations. Routing anomalies that occur post deployment due to network convergence issues or network update problems are not within the scope of this research and will not be discussed. The following are some of the major BGP routing anomalies that can manifest due to router misconfigurations:

1. Origin Misconfiguration[71]: this anomaly occurs when an autonomous system unintentionally injects a prefix into the global BGP table. This anomaly manifests due to either of the following reasons - (i) announcing part of another autonomous system's address space; (ii) advertising prefixes that are meant to stay within the autonomous system based on the network policy intention; (iii) injection of one or more specific prefixes into the global BGP table due to the failure of summarising an address space. Origin misconfiguration anomaly can be caused due to any of the following reasons: initialisation bugs; old configuration; redistribution; use of community attribute in some instances; forgotten filters; incorrect summary; hijacks, e.t.c.
2. Multiple Origin AS (MOAS): this anomaly occurs when a prefix appears to be originated from two or more autonomous systems. MOAS anomalies can sometimes be intentionally implemented for load balancing or to minimise the routing distance for connections to and from different locations. According to Zhao et al [107] some of the reasons for the manifestation of these anomalies include: exchange points; some forms of multi-homing; and faults amongst others

are the reasons behind this anomaly.

3. Invalid Paths or Non-Existent Connectivity[61]: this anomaly occurs when some false connectivity to an external autonomous system is advertised maliciously by an autonomous system so as to make some or all traffic intended for a particular destination transit through the malicious autonomous system.
4. Private AS Number Announcement[72]: the private autonomous system number segments are usually used to divide a single AS into multiple smaller autonomous systems as stated in Chapter 5.1 above. This anomaly occur when an autonomous system advertises a set of prefixes with private AS numbers to the outside world or external autonomous systems.
5. Export Misconfiguration[71]: this anomaly occurs when a prefix is intentionally advertised to a BGP peer in violation of the network's export policy intention. The causes of this anomaly include erroneous prefix based configuration and bad access control list (or ACL) or route maps. These anomalies manifests in cases whereby: (i) routes learned from a provider are advertised to other providers or peers; (ii) routes learned from peers are advertised to providers.
6. BGP Wedgies[101]: this anomaly occurs when the BGP configuration of a proposed network leads to different end states depending on which order routes are advertised after the network has converged. Usually when BGP is configured, it is expected that the proposed network will correctly converge as the policy intention dictates after all the sessions come up. However under BGP wedgies circumstances this does not happen. Making the network converge at a state that is not as intended by the proposed routing policy.

5.2 BGP Policy Intention Abstraction

This section outlines how BGP routing policy abstraction for our research was implemented. The nodes in this policy intention abstraction are used to represent a group of routers that belong to a particular service providers or that have identical inter-domain routing policies. The labels on the nodes will be used in the network layout to determine what service provides or inter-domain routing realms a network router belongs to during the experiment. The edges in this policy intention abstraction are used to indicate the relationship between the two interconnected nodes (realms or service providers). The labels such as provider, sibling, peer and others on the edges states the exact inter-domain routing policies or BGP business relationship between the two interconnected realms connected by the edge in any given experiment.

The nodes or realms in these policy intention abstractions are used to represent autonomous systems within a proposed experiment. The only node option that has

been used in expressing BGP policy abstractions has to do with its autonomous system number. This is to be applied using a node option called *asn* or NePAS will automatically generate and assign a random public AS number between the range specified in Chapter 5.1 above except for numbers used by other realms within the policy intention. This node option hence does not have a default value because an AS number will be generated automatically by NePAS if none is specified. The edge labels in BGP policy abstractions are used to express the routing relationship between the two realms or autonomous systems. The current relationships supported by NePAS are as follows: peer, hybrid, sibling and transit. It should be noted that NePAS policy intention implementation for transit relationships is split into customer and provider relationships. The edges here have two custom options used in expressing hybrid IP dependent relationships called *v4rel* and *v6rel*. By default, these edge options have 'NULL' values except when otherwise specified. These two edge options can take any of the following options: peer, customer, provider or sibling relationships.

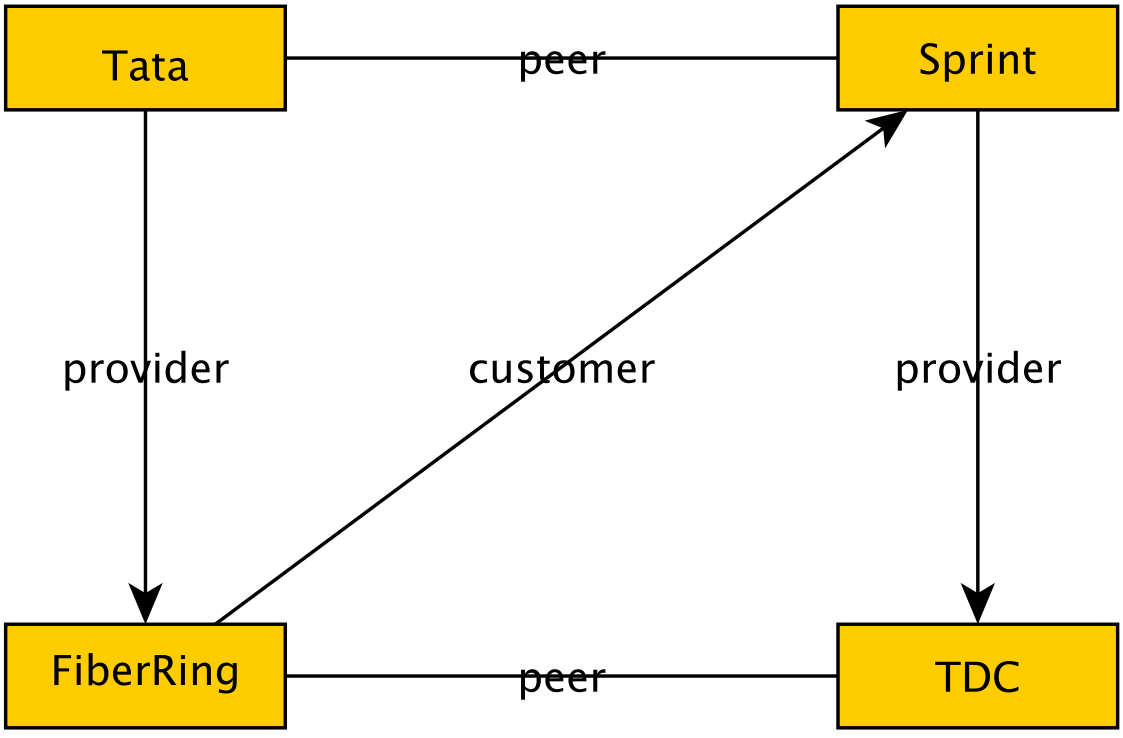

Figure 5.7: Proposed Multi-Tier ISP Policy Intention

Figure 5.7 above depicts a multi-tier ISP proposed network with four ASes. The ASes with their respective AS numbers are as follows: Sprint (AS1239); TDC (AS3292); TATA (AS6435); FiberRing (AS38930). Table 5.3 below shows the various business relationship between the four ASes in the proposed network.

Table 5.3: Business Relationship of Proposed Multi-Tier ISP Network

	TATA	Sprint	TDC	FiberRing
TATA	-	Peer	-	Provider
Sprint	Peer	-	Provider	-
TDC	-	Customer	-	Peer
FiberRing	Customer	-	Peer	-

5.3 BGP Network Layout Abstraction

This section outlines how the network layout abstraction for inter-domain routing (BGP) enabled networks in NePAS was implemented. The ellipse nodes in this network layout abstraction are used to represent routers; rectangular nodes are used to represent user computing devices; 6-pointed star are used to represent firewalls; and trapeziod nodes are used to represent backend network servers. It should however be emphasised that these devices need to specified using the custom property, ***dtype***, explained Chapter 4.2.1 before our system will accept and know what type of network device it will be dealing with during execution. Pointed edges between ellipse nodes are used to represent either internal or external BGP connections in this abstracion. It should be noted that the edges in the BGP network layout abstraction does not use any labels and any labels added will be ignored by our system during the execution.

There are two node options and one edge option that have been provided for the abstraction of a BGP network layout graph. The first set of policy intention custom property is a node option called *bgppol*. This option is used to state the inter-domain routing policy intention realm (AS) a network router belongs to in the policy intention graph. This node option requires a string value of the AS's policy intention label and is used to show the autonomous system a network router belongs to in the proposed experiment. The value entered must be an exact match of characters used as the AS's label from the network policy intention graph. It should be noted that only routers can have this value assigned and if any other network component such as a switch, firewall or server is assigned, NePAS will ignore it and write an informational log message during low level configuration. A network router can only belong to one AS group and if multiple given, NePAS will use the first value and ignore the rest. The default value of *bgppol* is 'NULL' when the network administrator does not provide any value. The routers that have not being assigned a *bgppol* value and other network devices such as firewalls and servers will be grouped into the AS and given an *asn* of the router that has been assigned a *bgppol* value closest to them.

Figure 5.8 below shows a proposed "Internet" experiment with multiple ISP networks. The routers in the core of the network are labeled with the same name used as their

realms/ASs in the policy intention graph introduced in Figure 5.7. All the routers have an eBGP connection to their neighbours based on the business relationships stated in Table 5.3 above. It should be noted that the routers in the network layout are not required to have the same label as the realm AS they represent but having the *bgppol* option value of the AS realm they belong to is what makes them have the business relationship of that realm/AS in relation to its neighbor.

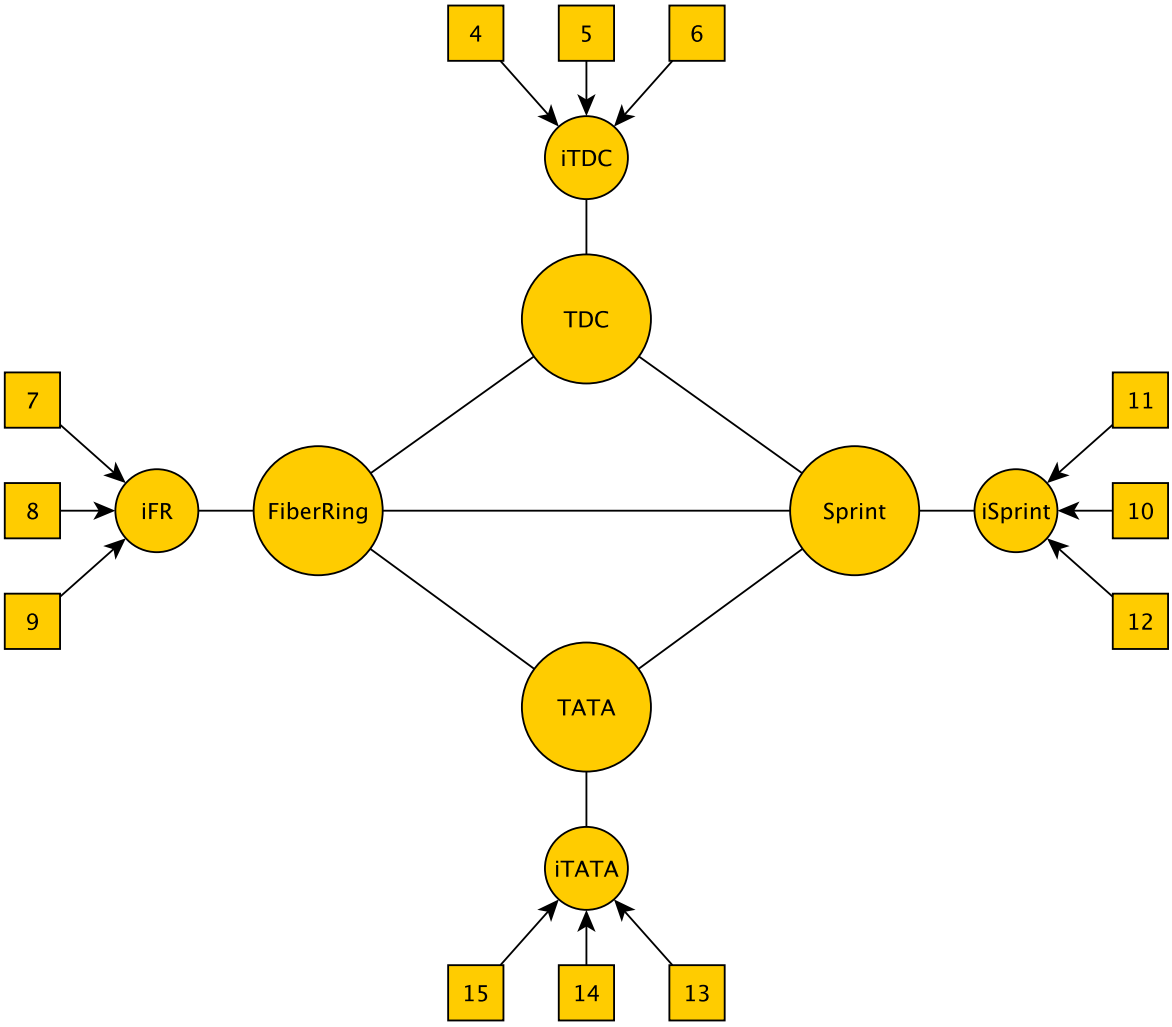

Figure 5.8: Proposed Multi-Tier ISP Network Topology

Table 5.4: Network Layout Options

AS	Networks	Backup	Devices
TATA		FALSE	iT, 15,14,13
Sprint		FALSE	iSprint, 12, 10, 11
TDC		FALSE	iTDC, 4, 5, 6
FiberRing		FALSE	iFR, 9, 8, 7

The second node option implemented for NePAS is called *network* and is to be used strictly by routers within a proposed experiment. It should be noted that if this option is applied on any network device that is not a router, NePAS will disregard it when generating low-level configurations. This option can have a network and mask or a list of networks belonging to the router. This node option has a default value of 'NULL' except when otherwise stated.

The only edge option that has been implemented for NePAS in BGP network layout abstraction is called *backup*. This edge option is used to implement the backup relationship as discussed in Section 4.1.2.5 above. This edge option takes a boolean value of either TRUE or FALSE with the latter being the default.

5.4 BGP Anomaly Resolution

This section discusses the series of techniques implemented so as to ensure the final low level BGP configurations generated for the proposed network are consistent and anomaly free. The techniques contained in this section resolve all the anomalies discussed in Section 5.1.3 above. These techniques resolve various anomalies that can disrupt the BGP routing relationships of the proposed network. The techniques implemented in this section ensures that final network deployments conform to the policy intention of the proposed network. The following are some of the techniques employed by NePAS in order to ensure BGP routing anomalies that can manifest are resolved prior to the deployment of such networks:

1. Origin Misconfiguration: the automated configuration process of NePAS ensures all autonomous systems only advertise prefixes within their domain. NePAS associates all IP addresses of servers or interface addresses of network devices to the router closest to them as prefixes the router will advertise to its neighbours according to the dictates of the network routing policy intention graph expressed. The fact that NePAS allocates unique IP addresses for servers and interface addresses of network devices eliminates the likelihood of origin misconfiguration. Also, NePAS generates code that conforms to the intended network routing policy intention and hence cannot advertise prefixes to neighbours in clear contrast to the expressed network policy intention.
2. Multiple Origin AS (MOAS): there are only two ways in which two autonomous systems can claim to to be originating a prefix in NePAS, these are as follows: (i) by manually specifying the same IP address using the *sIP* or *dIP* options on the interfaces of network devices, or (ii) by specifying the same prefix to multiple routes in different realms using the *network* option in the network layout graph as part of prefix associated to the autonomous systems. NePAS has been fitted with an

algorithm that will ensure all manually specified IP addresses and prefixes supplied by the network administrator are unique before the configuration process begins. By ensuring IP addresses and prefixes are unique to a specific autonomous system, NePAS will be able to generate BGP configurations that are MOAS anomaly free for any proposed network experiment. Whenever NePAS detects the same IP address or prefix in two different autonomous systems, it will generate an error message stating MOAS anomaly will occur within the proposed network experiment and terminate the configuration process.

3. Non-Existent Connectivity: one of the ways for this to occur is by supplying bogus prefixes using the *network* option in the network layout graph of the proposed network. For now, NePAS is not equipped to avoid this sort of anomaly in any deployed network. This is due to the fact that default routes will be configured for all prefixes in the network option during the low level configuration process.
4. Private AS Announcement: as explained in Section 4.1.3, this anomaly occurs when a router with a private AS number advertises a series of prefixes to the outside world (or networks with public AS numbers). NePAS does not support private AS numbers currently and when one is manually provided, it is automatically changed to a public one and a warning log message is generated. This algorithm which automatically replaces a private AS number with a public one ensures that this anomaly does not occur during low level configuration.
5. Export Misconfiguration: the automated configuration process of NePAS ensures that the proposed network policy intention is generated based on the routing policy graph supplied. NePAS will generate the low level configuration commands based on the combination of both the network routing policy intention and network layout graphs supplied.

5.5 BGP Compilation

This section outlines the process NePAS takes to configure low level BGP policy commands on network routers across a proposed network. The configuration process is done over five stages discussed in the following section.

The first stage of the configuration process is used to automatically generate an AS number for BGP realms that have not being assigned one in the policy intention graph. NePAS will go through the nodes of the policy intention graph expressed by the network administrator and checks the *asn* node option whether the value is NULL and a public AS number that has not been specified by any other realm in the policy intention graph is generated. Therefore, if NePAS randomly generates an AS number that is already in use by another realm, another new AS number will be generated. The new AS numbers must

be unique within the proposed network. NePAS checks and automatically generates new AS numbers for any realm in the policy intention that has a private AS number specified as well. This is to ensure that no private AS announcement is done when the proposed network is finally deployed.

The second stage of the configuration process deals with grouping or assigning AS numbers to routers (with no *bgppol* value assigned), servers and firewalls in a proposed network. Routers with no assigned *bgppol* value, firewalls and servers will take the AS numbers of the closest router that have been assigned a *bgppol* value. All devices that have the same AS numbers will have similar BGP routing policies.

The third stage of the configuration process deals with generating low level commands for default routes within an AS. The addresses of devices that have the same AS numbers as assigned in the second stage described above will be appended to the *network* node option. It should be noted that NePAS automatically allocates IP addresses with a /24 mask for each link in the proposed network layout. NePAS moves on to check all the prefixes in the *network* option of all the routers in the proposed network to make sure no two routers have the same prefix(es). This is to ensure no multiple origin AS (MOAS) anomaly occurs when the network is deployed. NePAS then generates low level commands for configuring the default routes using all the addresses within the *network* node option. By doing this, NePAS ensure no non-existent connectivity anomaly occurs when a proposed network is finally deployed.

The fourth stage of the configuration process deals with generating low level commands for back up links within the proposed network. NePAS goes through the edges of the network layout so as to check which backup edge options have been specified as 'TRUE' during the compilation process. The edges that have the value of backup edge option of 'TRUE' are considered to be backup links and NePAS generates backup link configuration on those edges. NePAS uses local preference attribute in order to configure backup links.

It should be noted that all the links in NePAS have the default local preference value of 100 during the compilation process.

The fifth stage of the configuration process deals with generating low level commands for routing policies to be configured on routers that have been assigned *bgppol* value. This stage of the configuration process is split into two to handle: import and export policies.

5.5.1 Import Policies

This stage of the configuration process is used to generate low level commands of import policies between all BGP sessions in a proposed network. Import policies have been employed at this stage in order to label ASes advertised by neighbouring routers

according to the business relationship between them. This way appropriate export policies can be configured that will not violate the high-level policy intention expressed. A warning message will be generated during the configuration process by NePAS. NePAS at this stage goes through the network layout graph supplied for a proposed network and checks the *bgppol* option has been assigned for the two routers that will be configured with an eBGP session. NePAS will not generate low level configuration commands for eBGP sessions between any two routers that do not have a business relationship in the policy intention graph expressed. Any two routers that have been assigned a *bgppol* node option and the values do not represent any realm or AS in the policy intention graph will not have BGP policies configured on them as well. On the other hand, if the two routers in the network layout graph have the *bgppol* value specified, NePAS will use a combination of route-maps, community attributes and local preference to label the routes received from eBGP neighbours. As stated in the Community part of Chapter 5.1.1 above, the community attributes will use the AS number of the router in the first two octets and an administratively defined identifier for the last two octets. NePAS will label community attributes based on this for the various routes in any experiment depending on the BGP business relationship between the routers according to the following:

1. Provider Routes - the AS number will be the *asn* value of the *bgppol* for the router that is being handled. The last two octets for all peer routes will be 222. These routes will have a local preference value of 50 during the configuration process.

2. Peer Routes - the AS number will be the *asn* value of the *bgppol* for the router that is being handled. The last two octets for all peer routes will be 111. These routes will have a local preference value of 100 during the configuration process.

3. Customer Routes - the AS number will be the *asn* value of the *bgppol* for the router that is being handled. The last two octets for all peer routes will be 333. These routes will have a local preference value of 150 during the configuration process.

4. Sibling Routes - the AS number will be the *asn* value of the *bgppol* for the router that is being handled. The last two octets for all peer routes will be 444. These routes will have a local preference value of 200 during the configuration process.

5.5.2 Export Policies

This stage of the configuration process is used to generate low level commands of export policies between all BGP sessions in any proposed network. The export policies that will be generated at this stage depend on the BGP business relationship between the various autonomous systems as specified in the policy intention graph. NePAS after generating

BGP import policies as described in the previous section will also generate export policy community filters for all BGP sessions. As stated in Chapter 5.2 of this thesis, there are four BGP business relationships supported by NePAS: peer, sibling, transit (which is split into provider and customer) and hybrid.

The following section gives a detailed explanation of the various community filters used in generating BGP export policies by NePAS.

- Provider to Customer

  In this relationship, the routers have a *bgppol* option that belongs to realms connected with an edge labeled *provider* in the policy intention specification. The router belonging to the source policy realm will have provider export policies generated for it. As the router is the provider of the other connected router, it will be configured with a community filter that permits advertisement of all routes: peer, provider, customer and sibling.

  The router with destination policy realm however will have customer export policies generated for it. As the router is a customer of the connected router, it will be configured with a community filter that permits advertisement only its customer routes if any while denying advertisement of routes it has learned through peers and other providers.

- Customer to Provider

  In this relationship, the routers have a *bgppol* option that belongs to realms connected with an edge labeled *customer* in the policy intention specification. The router belonging to the source policy realm will have customer export policies generated for it. As the router is the customer of the other connected router, it will be configured with a community filter that permits advertisement of only its customer routes if any while denying advertisement of routes it has learned through peers and other providers.

  The router with destination policy realm however will have provider export policies generated for it. As the router is a provider of the connected router, it will be configured with a community filter that permits advertisement of all routes: peer, provider, customer and sibling.

- Peer

  In this relationship, the routers have a *bgppol* option that belongs to realms connected with an edge labeled *peer* in the policy intention specification. Both routers in this case will have peer export policies generated for them. As both routers have a peer BGP business relationship, they will be configured with a community filter that permits advertisement of only routes learned from their

respective customers while denying advertisement routes learned from other peer relationships and providers.

- Sibling

  In this relationship, the routers have a *bgppol* option that belongs to realms connected with an edge labeled *sibling* in the policy intention specification. Both routers in this case will have sibling export policies generated for them. As both routers have a sibling BGP business relationship, they will be configured with a community filter that permits advertisement of all routes: peer, provider, customer and sibling to each other.

- Hybrid

  A warning message is currently generated whenever a hybrid BGP business relationship is specified due to lack of IPv6 support by NePAS.

5.6 Closing Remarks

This chapter of the thesis gives a thorough review of inter domain routing protocol (BGP), BGP attributes, how BGP is used to influence routing policies and the various types of BGP business relationships. The chapter proceeds to showcase how we implemented our BGP business relationship abstractions which include: provider, customer, peer, sibling, hybrid and backup links. The chapter then goes through a phase-by-phase assessment of how NePAS can be used to evaluate our proposed inter domain routing abstractions. An example is used to give a step-by-step assessment of how NePAS handles BGP business relationship abstractions. The next chapter will be used to discuss firewall policies and our proposed firewall policy intention abstractions.

Chapter 6

Firewall Abstractions

6.1 Background

A firewall is a system of computer hardware or software that is used to protect both individual computers and corporate networks from hostile attacks. A firewall achieves this by filtering (or blocking) the passage of undesirable data traffic from crossing into or out of a network. Firewalls are usually placed between the edge of a network where traffic pass through. A badly configured or a firewall that is inadequate can seriously damage the network it is meant to protect. If a firewall fails, all the traffic that is meant to go through it to the other side of a network will be interrupted.

Firewalls are not a single security solution for an enterprise network. A layered security approach should be used when designing the network security policy of an organisation. A network security policy is a document prepared by an organisation outlining the rules regarding data access; passwords; encryption and network segmentation (zones). The zones detailed in a network security policy document outlines how the network devices will be divided into and their respective access requirements. The security policy of an organisation is used to have a comprehensive document that will be used to guide the security of the organisation when attackers try to penetrate the network amongst many other possible events. Firewalls are configured based on the security policy of the organisation and its effectiveness is only as good as the defined policy. Therefore having an effective and comprehensive security policy can never be overemphasised especially in an enterprise network. A firewall will need the following in order to achieve its objectives adequately: (1) It needs to be placed at a point in the network where traffic must pass through, (2) It must be able to differentiate between allowed and denied network traffic, (3) It must work very fast so that it does not increase the latency of the network traffic.

Firewalls generally have the following properties: (1) outline actions that will be taken in response to circumstances, (2) constantly evolving and changing to meet new

security needs, (3) dictates both acceptable and unacceptable usage parameters.

Some applications of network firewalls include amongst others: (1) It is used to filter network traffic, (2) It is used as a network address translator, (3) It is used to prevent denial of service attacks, (4) It can be used as an anti IP address spoofing device.

Some limitations of firewalls include amongst others[21]: (1) they are ineffective against users with authorised access or internal attacks using malware and other such devices, (2) they cannot defend a network from attacks using software bugs, malicious codes or trojan horses etc, (3) They cannot stop internal users from accessing websites with malicious code, (4) They are ineffective against nonethical security risks such as social engineering.

6.1.1 How Firewalls Work

Firewalls block or allow network traffic based on a set of rules established by the organisation in its security policy and configured by the network administrator. The rules define a specific pattern based on a tuple that the firewall detects and an action to be taken. A firewall tuple comprises of a source and destination IP addresses; source and destination ports; protocol and action to be taken when the network traffic matches the tuple options. Therefore a firewall rule has the following tuple options:
<action> <protocol> <source IP> <source port> <destination IP> <destination port>
Action - this firewall tuple option is used to indicate the decision to be taken when the network traffic match the firewall rule pattern. The decision that can be taken is to either *permit* or *deny* the network traffic into or out of the network.
Protocol - this firewall tuple option is used to indicate the protocol that will be used for the firewall rule relation. The possible options here include amongst others: TCP, UDP, ICMP, IP, GRE, AH, etc.
IP Addressing - this firewall tuple option is used to indicate the addressing (both source and destination) option of the firewall rule relation. There are three different ways to specify this tuple option, these are as follows:
Option 1 - this option specifies all computing devices both within the proposed network layout and outside the network. This option is represented in the firewall rule relation as "any".
Option 2 - this option is used to specify the source or destination IP address representing a single computing device that is either within the proposed network or outside. This option is represented by an IP address being preceded by "host" in the firewall rule relation for Cisco devices.
Option 3 - this option specifies a whole subnetwork that employs similar firewall rule relations for all the computing devices within the subnetwork. This is represented using an IP network and network mask in the firewall rule relation.

Ports - this firewall tuple option is used to indicate the potential port number (either source or destination) that will used for the communication in the firewall rule relation. There are three different ways to specify this tuple option, these are as follows:
Option 1 - this option is specified when a single value is used to represent the port (source or destination) number in the firewall rule relation. This is usually represented by either the name of the service (e.g www, dns, ftp-data etc.) or port number (e.g 80, 21, 8080 etc.) preceded by 'eq' in the firewall rule relation.
Option 2 - this option can be used when a group of port numbers have the same firewall rule relation in the security policy. This is usually represented by the starting and ending numbers preceded by 'range' (e.g 1250-1300) in the firewall rule relation.
Option 3 - some firewall rule relation do not specify the port number of the two devices that will be communicating and hence none is given. This implies the two devices if permitted can use any port number to communicate with each other. On the other hand, if its a deny rule, the two devices will not be able to communicate with each other using any port number.

It should be noted that some firewalls can support other types of actions beyond permitting or denying network traffic such as sending a log message, applying a proxy, and passing the matched packets into a VPN tunnel[88]. This functionality is beyond the scope of this research and will not be discussed further. Firewall rules are stored in the firewall rule base and are applied when network traffic is about to enter or exit the interface of the firewall. Firewalls try to match network traffic patterns from the first rule to the last as contained in the rule base in a sequential order[73]. Firewalls always enforce the first rule that matches and hence rule ordering in the rule base is very important. Also the more rules in a rule base, the longer it will take to determine what action to take with regards to the data packet. The most important and utilised rules should be at the top of the rule base so as to ensure the firewall works efficiently[54]. The rule base should also be reviewed periodically so as to remove rules that are no longer needed in the firewall. It is advised that additional firewalls be added to a network instead of adding more firewall rules into a single device. This is to reduce the number of rules in a firewall and also enhance shorter rule base traversal for firewall efficiency.

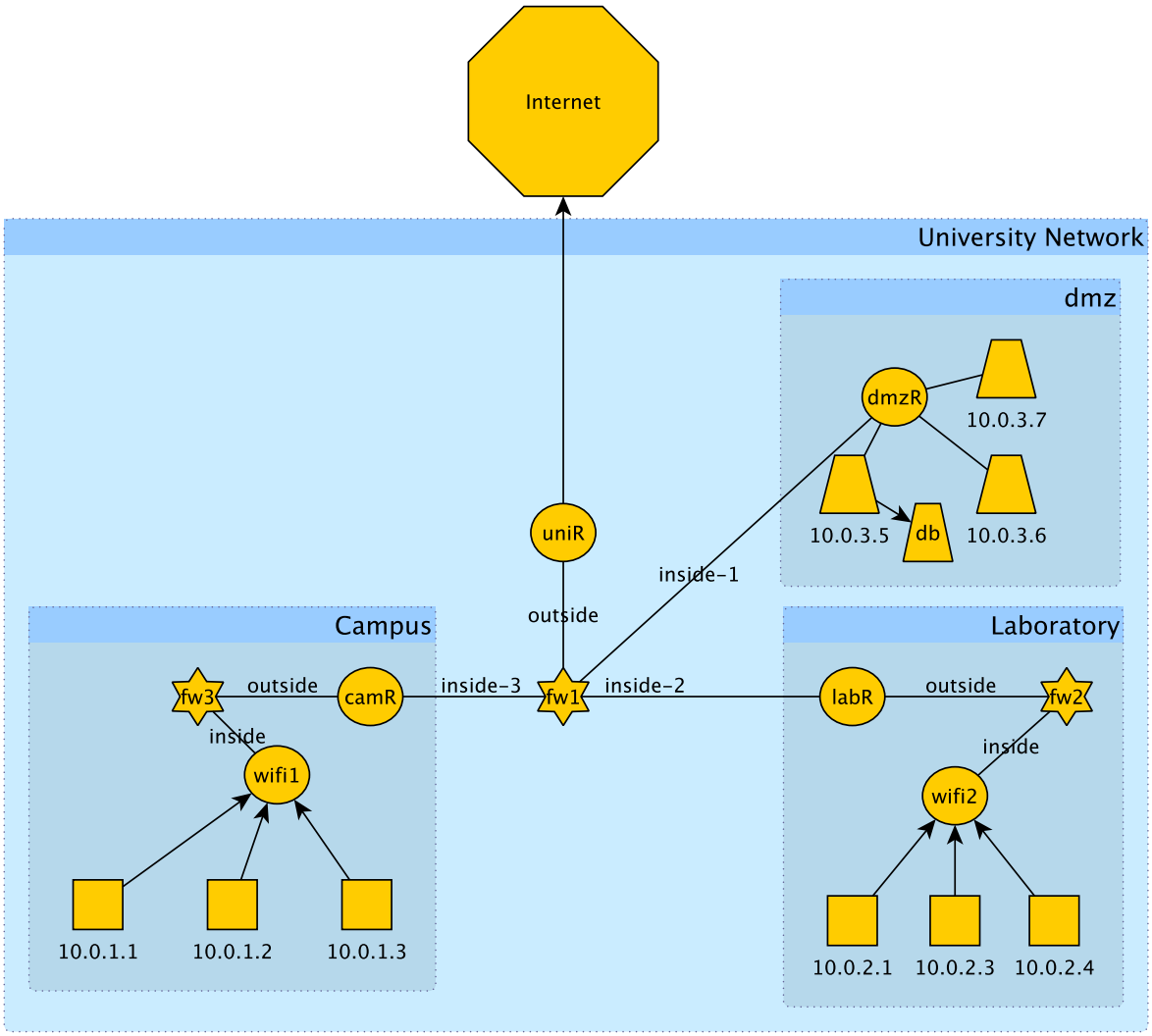

Figure 6.1: Example of a Distributed Firewall Hypothetical University Network

Networks use different security zones to protect important resources and defend against hackers. These are the key elements that define security zones.
(i) External network also called outside, Internet or unprotected networks.
(ii) Internal network also called inside or protected networks.
(iii) Demilitarised Zone (DMZ) - is an interface between the internal and external networks where critical enterprise servers are located. This zone enables the external network hosts to have access to servers that have been isolated in the DMZ while denying them access into the internal network. The ability to confine Internet traffic is the benefit of using a DMZ. The servers within this zone should not be allowed to "initiate" connections to other parts of the network. Firewall rule bases should contain rules for DMZ traffic.
(iv) Perimeter - is the border between the internal and external networks. Firewalls on the perimeter of a network are responsible for incoming and outgoing traffic. Some con-

nections that are usually allowed (though not sacrosanct) by the firewall in the perimeter include amongst others: (1) VPN connections from external to internal networks, (2) outgoing connections from the internal to external networks, (3) connections from both internal and external networks to the DMZ.

In Figure 6.1 above, the firewall at the core of the network, fw1, is at the perimeter of the hypothetical university network and has the Internet as its outside or external network while the remaining parts of the network is its inside or internal network. However for firewalls fw2 and fw3, the network core is regarded as their external and unprotected networks while the Laboratory and Campus networks are regarded as the protected or internal networks respectively.

Outgoing packets from the internal network to the Internet of an organisation should be allowed to pass without scrutiny except otherwise stated in the security policy. However incoming packets are usually filtered based on the predefined set of rules in the rule base according to the organisation's security policy. Firewalls also filter packets with spoofed internal IP addresses at its external interface. Therefore when configuring anti-spoofing, administrators should be very careful when specifying the interfaces that go to the Internet and an internal network. There are times when specific requests from the Internet to the internal network must be allowed and controlled through the firewall. Example of such connections include VPN connections and legitimate requests to the DMZ. Allowing a direct connection from the Internet to an internal network host through a firewall is perilous but common practice. Well implemented firewalls should also be able to filter traffic from internal to external networks so as to ensure no attacks are launched from within the enterprise network. There are certain times that an enterprise network security policy prohibits access from the internal network to the Internet that should also be configured on the firewall.

As NePAS is designed to generate low-level firewall configuration commands on Cisco Adaptive Security Appliance (ASA)[58] devices, the following section discusses how firewall rules are configured and implemented on such devices. It should be noted that by default, Cisco ASA devices allow traffic to flow from a higher security interface to a lower security interface. The only exception to this rule is ICMP protocol based traffic which requires a firewall rule to be configured before such traffic traverses the firewall device. Apart from that, any other traffic that needs to traverse the device must have an access list configured before it is allowed. Firewall rules are configured in Cisco IOS using the *access-lists* command syntax and are deployed on to the interface of the device using the *access-group* command syntax.

- Access Lists:

  The *access-list* command is used to create firewall rules that are designed to control traffic going either in or out of the network through a Cisco ASA device.

The command is used to create firewall rules or access control lists (ACLs) that are used to enforce the organisation's security policy. Once the ACL have been configured by the *access-list* command, it has to be applied on a firewall's interface for it to be active. Cisco ASA devices support numerous ACL types including amongst others[1]: Standard ACLs, Extended ACLs, Named ACLs, Dynamic ACLs, Turbo ACLs, Reflexive ACLs, Receive ACLs, Transit ACLs, Authentication Proxy. However for the purpose of the research conducted in this research thesis only three (standard, extended and named ACLs) will be discussed in detail.

Standard ACLs: these ACLs are used to create filters based on source addresses of traffic passing through the firewall devices. These ACLs are used for normal server based filtering because they only use source addresses to control traffic entering or exiting the network. This ACL type does not support the use of additional firewall tuple options such as port numbers; protocols and destination addresses. These ACLs are numbered between 1-99 an 1300-1999 respectively. This ACL is insufficient for the level of abstraction needed for this research due to its inflexibility in supporting additional firewall tuple options.

Extended ACLs: these ACLs are used to create filters as dictated by the security policy document based on the entire firewall tuple options detailed above. The ACL filter created here is used to control traffic based on a combination of the firewall tuple options as determined by the security policy for all traffic passing through the firewall device. This ACL type are numbered between 100-199 and 2000-2699 respectively.

Named ACLs: these ACLs are extended versions of standard and extended ACLs discussed above. The difference between the two ACLs discussed above and their named ACL equivalent is the usage of a unique name for the named ACLs instead of unique numbers being used. It should be noted that named ACLs are exactly the same as standard ACLs apart from the identification of the ACLs. NePAS will be using named extended ACLs when generating low-level firewall rule relations.

- Access Groups:

  The *access-group* command is used in applying the configured firewall rules (or ACLs) on the appropriate interface of the firewall device. It should be noted that when ACLs are configured but not applied on any interface of the firewall, it will have no effect. The access-group command is automatically removed from the firewall device whenever the ACL entry associated with it is removed from the rule base. This is because access-group commands cannot be configured in an empty firewall rule base. There are two ways in which the access-group command can be used to apply ACLs on an interface, these are: interface rules and global

rules.

Interface ACLs: these are ACLs that are applied on a specific interface of the firewall device so as to match traffic that is going either in or out of the network. There are two types of interface ACLs: incoming and outgoing.
Incoming Interface ACLs - these interface ACLs allow/deny network traffic that have been matched to them which are destined to pass through the firewall and go into the network. These ACLs usually have source addresses of devices outside the network trying to communicate with device(s) inside the network.
Outgoing Interface ACLs - these interface ACLs allow/deny network traffic that have been matched and are destined to go out of the network. These ACLs usually have the source address of devices inside the network to communicate with devices outside the network.

Table 6.1 below provides a step-by-step guide of configuring named extended ACLs in Cisco ASA devices.

Table 6.1: Step-by-Step Cisco ASA Firewall Rule Configuration

1	Enter privileged exec mode	asa>**enable**
2	Enter global config mode	asa#**configure terminal**
3	Create and configure named extended ACL entry *NB: Repeat as needed*	asa(config)#**access-list** <acl-name> **extended** <permit | deny> <protocol> <source IP> <source port> <destination IP> <destination port>
4	Apply ACL to the appropriate interface. *NB: interface name is matched with configured nameif value*	asa(config)#**access-group** <acl-name> <in | out> **interface** <interface-name> asa(config)#**access-group** <acl-name> **global**

Global ACLs: these are ACLs that do not need to be configured on individual interfaces of the firewall device unlike the interface ACLs. This is because these ACLs are automatically applied on all configured interfaces of the firewall device. It should be noted that global ACLs are only considered if no interface ACL match the traffic that is going either in or out of the network through the firewall device.

Listing 6.1 below show a sample excerpt of some Cisco ASA firewall rules syntax for firewall fw1 in Figure 6.1 above.

Listing 6.1: Excerpt of a Cisco ASA Firewall Rule Configuration File

```
1| access-list dmz-out extended deny   ip any any
```

```
2| access-list dmz-in  extended permit tcp any any
3| access-list dmz-in  extended permit udp any 10.0.3.0 255.255.255.0
4| access-list dmz-in  extended permit udp any 10.0.3.0 255.255.255.0
5| access-list cam-out extended permit ip 10.0.1.0 255.255.255.0 any
6| access-list cam-out extended permit tcp  host 10.0.1.1 eq 80  host
    10.0.2.4  eq 8080
7| access-list cam-out extended permit udp 10.0.1.0 255.255.255.0
   10.0.0.0 255.255.0.0  eq 80
8| access-list cam-out extended deny   udp 10.0.1.0 255.255.255.0
   10.0.3.0 255.255.255.0  eq 53
9| access-list cam-out extended permit ip 10.0.1.0 255.255.255.0 host
    107.23.58.1
10| access-list cam-out extended permit  ip 10.0.1.0 255.255.255.0
   host 86.67.42.5
11| access-list cam-out extended deny    tcp 10.0.1.0 255.255.255.0
   host 1.2.3.4
12| access-list cam-out extended deny    ip host 10.0.1.1 host
   4.5.6.7
13| access-list cam-out extended permit  tcp 10.0.1.0 255.255.255.0
   host 10.0.3.5 eq 21
14| access-list cam-out extended permit  tcp 10.0.1.0 255.255.255.0
   any eq 21
15| access-list cam-in  extended permit  tcp any 10.0.1.0
   255.255.255.0 eq ssh
16| access-list lab-out    extended permit  udp 10.0.2.0
   255.255.255.0 any eq 40728
17| access-list lab-out    extended permit  udp 10.0.2.0
   255.255.255.0 any eq 3689
18| access-list lab-out    extended deny    udp  host 10.0.2.4   eq
   3000   host 10.0.3.5  eq 21
19| access-list lab-out    extended permit  tcp    10.0.2.0
   255.255.255.0   10.0.0.0 255.255.0.0 range 0-256
20| access-list lab-out    extended permit  tcp    10.0.0.0
   255.255.255.0   10.0.3.0 255.255.255.0 range 128-512
21| access-list lab-out    extended permit  tcp    host 10.0.2.3
   10.0.3.0 255.255.255.0  eq 80
22| access-list lab-out    extended permit  tcp    10.0.2.0
   255.255.255.0   10.0.0.0 255.255.0.0 eq 80
23| access-list lab-out    extended deny    tcp   host 10.0.2.3 any
   eq 80
24| access-list lab-out    extended permit  tcp   any any
25| access-list lab-out    extended permit  tcp   10.0.2.0
   255.255.255.0 any eq 80
26| access-list lab-out    extended deny    tcp   host 10.0.2.4 any
   eq 80
27| access-list lab-in     extended permit  ip    host 1.2.3.4 eq
   54321 10.0.2.0 255.255.255.0
28| access-list lab-cam extended permit  tcp   host 10.0.1.1 eq 23456
    10.0.2.3 eq 65432
29| access-list lab-cam extended permit  udp   10.0.1.0 255.255.255.0
    10.0.2.0 255.255.255.0
30| access-list lab-cam extended deny    udp   host 10.0.1.1 10.0.2.0
    255.255.255.0
31| access-list out extended deny    tcp   any 107.23.58.1
32| access-list out extended deny    tcp   any 86.67.42.5
33| access-list out extended permit  udp   10.0.5.0 255.255.255.0 any
34| access-list out extended permit  icmp  any any
access-group dmz-out out interface outside
```

```
access-group dmz-in in interface inside-1
access-group cam-out out interface outside
access-group cam-in in interface inside-3
access-group lab-out out interface outside
access-group lab-in in interface inside-2
access-group lab-cam global
access-group out global
```

6.1.2 Firewall Anomalies

Firewall rules designed and configured by network administrators may be prone to anomalies that can render parts of the network either unaccessible or open to malicious attacks. Inter-rule relation or dependency is very important for determining anomalies in a security policy. The following are however different ways to define rule relations in a firewall security policy[55][13]:

- **Disjoint Rules** - there are two variation of these types of firewall rules:
  (1) *completely disjoint rules* is when every firewall tuple option in a given rule Rx is neither a subset nor a superset and not equal to the corresponding tuple option in another rule Ry. The following two firewall rules in Listing 6.1 are said to be completely disjoint:
  Rule 6| extended permit tcp host 10.0.1.1 eq 80 host 10.0.2.4 eq 8080
  Rule 18| extended deny udp host 10.0.2.4 eq 3000 host 10.0.3.5 eq 21
  (2) *partially disjoint rules* between rules Rx and Ry occurs when at least one firewall tuple option in Rx is a subset or superset or equal to the corresponding tuple option in rule Ry. The following two firewall rules in Listing 6.1 are said to be partially disjoint:
  Rule 7| extended permit udp 10.0.1.0 255.255.255.0 10.0.0.0 255.255.0.0 eq 80
  Rule 8| extended deny udp 10.0.1.0 255.255.255.0 10.0.3.0 255.255.255.0 eq 53

- **Correlated Rules** - two rules Rx and Ry are said to be correlated when some firewall tuple options in Rx are a subset or partially intersects with the corresponding firewall tuple option in Ry and the rest of the tuple options in Rx are supersets of the corresponding tuple options in Ry. The following two firewall rules in Listing 6.1 are said to be correlated rules:
  Rule 19| permit tcp 10.0.2.0 255.255.255.0 10.0.0.0 255.255.0.0 range 0-256
  Rule 20| permit tcp 10.0.0.0 255.255.0.0 10.0.3.0 255.255.255.0 range 128-512

- **Exactly Matching Rules** - this occurs between any two given rules Rx and Ry when every given firewall tuple option in Rx is equal to the corresponding firewall tuple option in Ry. The following two firewall rules in Listing 6.1 are said to be exactly matching rules:

Rule 3| extended permit udp any 10.0.3.0 255.255.255.0
Rule 4| extended permit udp any 10.0.3.0 255.255.255.0

- **Inclusively Matching Rules** - this occurs between two rules Rx and Ry when the two rules do not exactly match and every firewall tuple option in Rx is a subset or equal to the corresponding firewall tuple option in Ry. The following two firewall rules in Listing 6.1 are said to be inclusively matching. Rule 21 is said to be a subset match of the relation, while Rule 22 is the superset match:
  Rule 21| extended permit tcp host 10.0.2.3 10.0.3.0 255.255.255.0 eq 80
  Rule 22| extended permit tcp 10.0.2.0 255.255.255.0 10.0.0.0 255.255.0.0 eq 80

The following section discusses the various well-known firewall anomalies in detail. There are two categories in which firewall anomalies can be grouped into, these are: intra firewall anomalies and inter firewall anomalies.

1. **Intra-Firewall Anomalies**[7] [47][14]

   The anomalies that fall within this section are those that affect a single firewall device. Intra-firewall anomalies include the following:

   - *Shadowing[95]*:- this anomaly occurs when two rules Rx and Ry have different firewall rule actions and rule Rx with higher priority order matches all the packets that match a rule Ry. This anomaly causes rule Ry to never be activated. This anomaly is considered an error in the firewall because it can cause a permitted traffic to be denied or a denied traffic to be permitted. The following two firewall rules in Listing 6.1 show an intra firewall shadow anomaly.
     Rule 29| extended permit udp 10.0.1.0 255.255.255.0 10.0.2.0 255.255.255.0
     Rule 30| extended deny udp host 10.0.1.1 10.0.2.0 255.255.255.0
     To fix this anomaly, the filtering list should be re-ordered with the superset (or general) rule coming after the subset (or specific) rule. In this situation, Rule 30 should be moved above Rule 29 for the network to behave as intended.
   - *Irrelevance*:- this anomaly occurs when a firewall rule cannot match any network traffic that might flow through the firewall. This anomaly adds unnecessary overhead to the firewall because the size of the rule base is increased and other network traffic have to traverse this extended rule base if it is not removed from the firewall. The following firewall rule in Listing 6.1 show an intra firewall irrelevance anomaly.
     Rule 33| extended permit udp 10.0.5.0 255.255.255.0 any
   - *Correlation*:- this anomaly occurs when two firewall rules Rx and Ry specify different filtering actions and some packets that match Rx also match Ry

and vice versa. This implies an action that is not explicitly handled by the policy. The following two firewall rules in Listing 6.1 show an intra firewall correlation anomaly.
Rule 23| extended deny tcp host 10.0.2.3 any eq 80
Rule 24| extended permit tcp any any
To fix this anomaly, the proper rule order that complies with the security policy requirements should be used.

- *Generalisation*:- a firewall rule Rx is a generalisation of Ry if they have different filtering actions and the fields in Rx (which has higher priority order than Ry) can match all the fields in rule Ry. This anomaly is considered a warning when the specific firewall rule makes an exception of the general firewall rule. The following two firewall rules in Listing 6.1 show an intra firewall generalisation anomaly.
Rule 25| extended permit tcp 10.0.2.0 255.255.255.0 any eq 80
Rule 26| extended deny tcp host 10.0.2.4 any eq 80

- *Redundancy*:- this anomaly occurs when a firewall rule Rx performs the same action on a network traffic that another firewall rule Ry can perform as well. All the packets that satisfy Rx also satisfy Ry such that when either of the two rules Rx or Ry is removed, the security policy of the firewall will not be affected. This anomaly is considered an error because it increases the size of the rule base and hence increases the search time and space of packet filtering. The following two firewall rules in Listing 6.1 show an intra firewall redundancy anomaly.
Rule 13| extended permit tcp 10.0.1.0 255.255.255.0 host 10.0.3.5 eq 21
Rule 14| extended permit tcp 10.0.1.0 255.255.255.0 any eq 21
To fix this anomaly, the shadowed rule should be removed from the firewall rule base.

2. **Inter-Firewall Anomalies**[12][35] [106]

The distributed nature of firewalls in large enterprises has the potential to significantly increase the anomalies in the network. An inter-firewall anomaly occurs if any two firewalls in the network path have different filtering actions on the same traffic. When dealing with inter-firewall anomalies, the preceding firewall between two subdomains is called **upstream firewall** whereas the following firewall is called a **downstream firewall**. The closest firewall to the source domain is called the **most upstream firewall**. The closest firewall to the destination domain is called the **most downstream firewall**. A anomaly exists between two hosts in different subdomains crossing multiple firewalls if: (1) the most downstream firewall permits traffic that is denied by any of the upstream firewalls; (2) the most

upstream firewall permits traffic that is denied by any of the downstream firewalls; (3) a downstream firewall denies traffic that is already denied by the most upstream firewall.

All upstream firewalls should permit any traffic that is permitted by the most downstream firewall so as to enable network traffic reach its destination. The following are the various types of anomalies that occur between inter-firewall networks.

- *Redundancy*:- this anomaly occurs when a downstream firewall denies the network traffic already denied by an upstream firewall. This unnecessarily adds the search time on the downstream firewall due to the presence of an unnecessary rule and hence might have an impact on the performance of the firewall.
- *Spuriousness*:- this anomaly occurs when unwanted network traffic is allowed by an upstream firewall to flow in the network when a downstream firewall has been configured to deny such network traffic. Complete spuriousness is said to occur when the fields of the two rules exactly match. However when the fields of the two rules inconclusively match, partial spuriousness is said to have occurred. Spuriousness is a critical anomaly because it allows unwanted traffic to flow across the network increasing the vulnerability of the network to attacks such as port scanning and denial of service.
- *Shadowing*:- this anomaly occurs if an upstream firewall rule denies some network traffic that has been allowed by a downstream firewall rule. Complete shadowing anomaly occurs when the firewall rule fields of both upstream and downstream firewalls exactly match each other. On the other hand, if the rules in the two firewalls inconclusively match each other, partial shadowing is said to have occurred. This is considered an anomaly because it prevents the traffic desired by the destination node from flowing through the network.
- *Correlation*:- this anomaly occurs as a result of having two correlated rules in the upstream and downstream firewalls of the network. Correlated rules having any action are always a source of anomaly in distributed firewalls because of the implied rule resulting from the conjunction of the correlated rules. This creates not only ambiguity in the inter-firewall policy but also spuriousness and shadow anomalies.

6.2 Firewall Policy Intention Abstraction

This section outlines how the firewall policy intention abstraction for the research thesis was implemented. The nodes in this policy intention abstraction are used to represent a group of network devices that have identical firewall relationships in any experiment. The labels on the nodes will be used in the network layout to determine the firewall realm (or group) a network device belongs to during an experiment. The edges in this policy intention abstraction are used to indicate the action to be taken in a firewall relationship between the two interconnected realms. The label permit on the edges in the graph is used to abstract firewall relationships between two interconnected realms that allows communication based on certain protocols in any experiment. The label deny on the edges in the graph is used to abstract firewall relationships between two interconnected realms that deny communication base on certain protocols in any experiment. It should be noted that color of the edges in the graph has no effect during the execution of the experiment as our system uses the labels and custom properties that have been defined by the network administrator during the design of any experiment.

Realms (or nodes) in the policy intention abstraction are used to represent a group of devices that have identical firewall relationship in relation to other devices within the proposed network. The name of the realm is very crucial as it will be used in the network layout phase to specify the firewall relationships a firewall device in the network layout has in relation to other firewall devices within the proposed network. The realms in a firewall policy intention graph do not have any node options and hence no abstractions have been provided. Realms labeled with the keyword *any* have been implemented to apply the abstracted firewall rule(s) on all firewall enabled devices in the proposed network. A realm labeled *any* with a link looping back to itself depicts an any-to-any firewall rule according to NePAS. The example provided in Figure 6.1 below depicts the high-level firewall policy intention abstraction with three realms (lab, campus and any) for a proposed hypothetical university network.

The edges in a firewall policy intention graph are used to represent the action to be taken by the firewall rule between the two realms. The action in this case is to either deny or permit traffic between the two realms the edge connects according to the security policy intention. The edges between firewall policy intentions cannot have default values and any edge between two realms that has no label (either permit or deny) will not be processed by NePAS. There are ten rules abstracted for the proposed university network depicted in Figure 6.2 below. There are a total of six permit and four deny rules in the proposed high-level firewall policy intention.

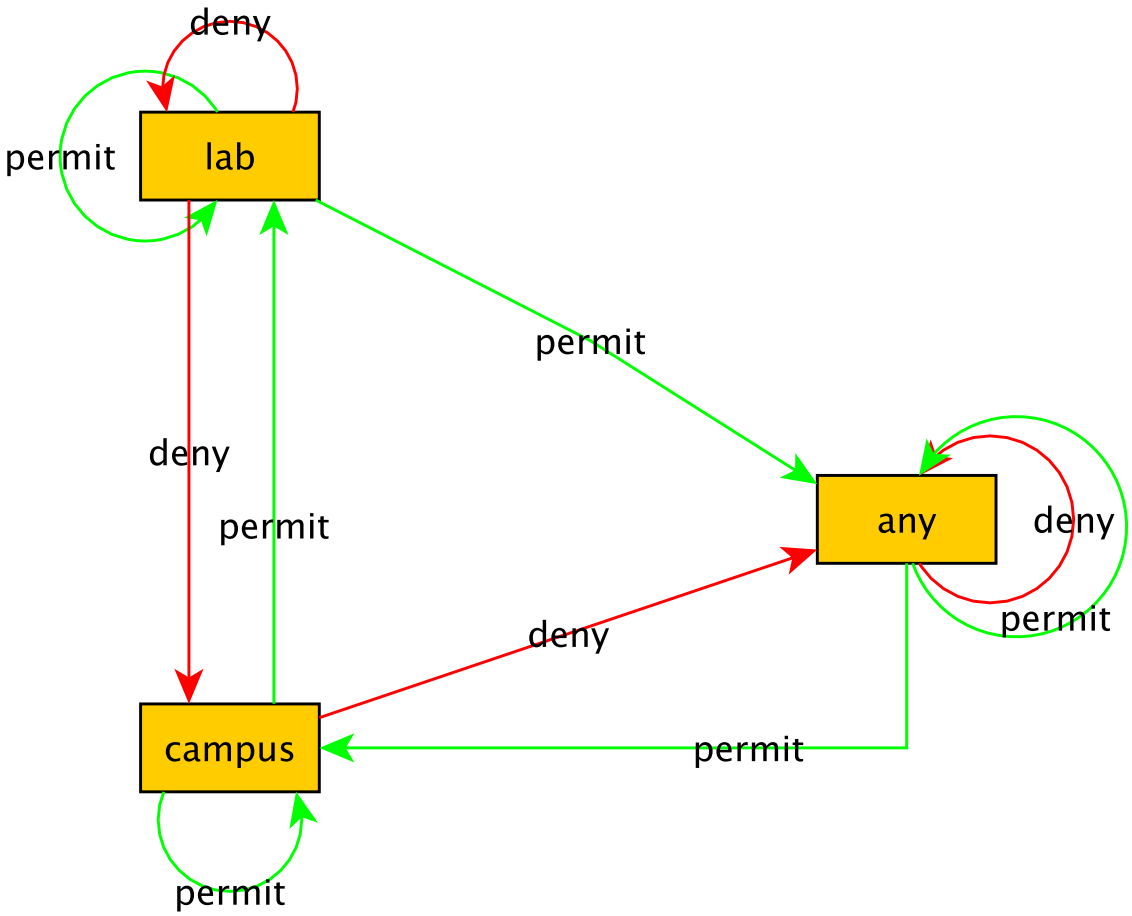

Figure 6.2: Proposed University Firewall Policy Intention

The edges in a firewall policy intention graph have the following custom properties that will give network administrators a way of specifying the remaining firewall tuple options: *protocol*; *dPort*; *dest*; and *sPort*.

The first edge option that was implemented for NePAS is called *protocol* and is used to state the transport layer protocol to be used for the firewall rule. Some of the options that can be selected include amongst others: TCP, UDP, IP, GRE, ICMP amongst others. All the protocols supported by Cisco ASA firewalls are also supported by NePAS. The default value for this option is IP except otherwise stated by the user. Multiple protocols can be specified by using a comma to separate each protocol option. This feature enables the network administrator to specify firewall rules with protocols during experiments.

The second edge option is called *dPort*. This option is used to state the destination port number or name the firewall rule will be using to communicate. This option can either be expressed using a service name such as www, dns, dhcp or a port number such as 80, 22, 443. This option has a default value of 'NULL' which will make NePAS configure a firewall rule that the two communicating devices in a proposed network

can use any port. Multiple service names, port numbers or a combination of service names and port numbers can be specified for the firewall rule. The multiple destination ports specified here must be separated using a comma symbol so as to enable NePAS process the firewall rule properly. It should be noted that for a firewall rule with multiple destination ports, the ports must be using identical protocol if not the firewall rule configured on the firewall will not behave as intended by the security policy.

The third edge option implemented for firewall policy abstraction in NePAS is called *dest*. This option is used when dealing with rules that have a destination or source outside the proposed network. The default value for this option is set to 'NULL' except otherwise stated by the user. This option takes a string comprising of a character of either *s* or *d*, a white space and then a string of IP address or IP network that represents the destination or source of the rule in question respectively. This option can also take the web address of the rule instead of an IP address. It should be noted that if the character of either *s* or *d* is omitted from the edge option, NePAS will generate a warning message and not process the firewall rule. Multiple destinations can be specified by using a comma to separate the various options. An example of how this option works is when a network administrator wants to represent a firewall rule between a realm and some device or network on the Internet. A realm must have an edge linking back to it in order to achieve this abstraction technique.

The last edge option implemented by NePAS is called *sPort*. This option is used to specify the source port number of the given firewall rule. This edge option is optional and has a default value of 'NULL' and hence overlooked by NePAS except when otherwise specified during the design process. This option is implemented to make NePAS very flexible and give users the option of specifying low level firewall details of the port the source device(s) of the realm will be using to communicate.

Table 6.2: Proposed University Firewall Policy Intention Details

S Realm	D Realm	Action	Protocol	D Port	Destination Address	S Port
any	any	permit	ICMP	NULL	NULL	NULL
any	any	deny	TCP	80, 8080	d www.gorillavid.in, d www.twitter.com	NULL
lab	any	permit	TCP	40728, 3689	NULL	NULL
lab	lab	deny	ICMP	NULL	NULL	NULL
lab	lab	permit	IP	NULL	s 1.2.3.4	NULL
lab	campus	deny	IP	NULL	NULL	NULL
campus	lab	permit	TCP	54321	NULL	54321
campus	any	deny	TCP	telnet, ftp	NULL	NULL
campus	campus	permit	IP	80, 8080	d www.gorillavid.in, d www.twitter.com	NULL
any	campus	permit	TCP	ssh	NULL	NULL

Table 6.2 above details the edge options abstracted for each individual firewall rule relation for the proposed university network presented in Figure 6.1 above.

From the example in Figure 6.1 above, there are ten firewall rules abstracted for the various realms in the proposed university network introduced. The first rule R1 is used to permit all the devices both within the university network and outside the network to communicate over ICMP protocol. The second rule R2 is used to deny any device within the university network from accessing social media (represented as *d www.twitter.com*) and online streaming websites (represented as *d www.gorillavid.in*) on ports 80 and 8080 over TCP protocol. The third rule R3 is used permit the any device that is within the lab realm to communicate using Skype with any device both within and outside the university on port 40728 or 3689 over TCP protocol. The fourth rule R4 is used to deny devices within the lab realm from communicating with each other over ICMP protocol. The fifth rule R5 is used to permit an external host (represented as *s 1.2.3.4*) to initiate communication with devices within the lab realm over IP protocol. The sixth rule R6 is used to deny communication from devices within the lab realm to devices within the campus realm over IP protocol. The seventh rule R7 is used to permit communication of devices within the campus realm on port 54321 to devices within the lab realm on port 54321 over TCP protocol. The eighth rule R8 is used to any device within the campus realm from communicating with any other device both within the university network and outside using telnet and ftp over TCP protocol. The ninth rule R9 is used to permit the devices within the campus realm from accessing social media (represented as *d www.twitter.com*) and online online streaming websites (represented

as *d www.gorillavid.in*) on ports 80 and 8080 over TCP protocol. The tenth rule R10 is used to permit any device both within and outside the university from accessing the campus realm using ssh over TCP protocol.

6.3 Firewall Network Layout Abstraction

This section outlines how the network layout abstraction for firewall-enabled networks in NePAS was implemented. The ellipse nodes in this network layout abstraction is used to represent routers; rectangular nodes are used to represent user computing devices; 6-pointed star are used to represent firewalls; trapeziod nodes are used to represent backend network servers. It should however be emphasised that all these devices need to specified using the custom property, ***dtype***, explained Chapter 4.2.1 before our system will accept and know what type of network device it will be dealing with during execution. There are two types of edges in this abstraction, these are - edges that represent logical links and edges used to abstract additional firewall rules. The first type of edges or pointed edges and edges that have no labels are used to represent logical links between the two devices in the graph. The second type of edges or the edges with labels are used to abstract firewall relationships in the network layout graph between the two interconnected nodes. The firewall relationships are abstracted in the same manner as the firewall policy intention. The label permit on the edges in the graph is used to abstract firewall relationships between two interconnected nodes that allows communication based on certain protocols in any experiment. The label deny on the edges in the graph is used to abstract firewall relationships between two interconnected nodes that deny communication base on certain protocols in any experiment. It should be noted that color of the edges in the graph has no effect during the execution of the experiment as our system uses the labels and custom properties that have been defined by the network administrator during the execution of the experiment.

The firewall devices are connected to routers which then connect to switches, servers or other routers when designing the proposed network layout. This is because connecting firewall devices directly to servers or to switches will cause the firewall to disrupt network connectivity within the proposed network layout. The disruption will be caused by the firewall device not being able to route traffic adequately to its proposed destination.

The first set of policy intention node options is called *fwpol* and is used to state the firewall realm a network device belongs to in the policy intention of the proposed network. The node option requires a string value of the realm's policy intention label for the network device. The value entered must be an exact match of the characters used as the realm's label from the network policy intention graph if not NePAS will not process the proposed network adequately. A network device can belong to multiple firewall policy intention realms. This is implemented by having multiple firewall policy intention

realms attached to the network device.

All the sequence of steps discussed in the previous Chapter 6.2 can also be used in the network layout layout graph. This is to further give the network administrator flexibility of representing firewall rule(s) on individual network device(s). Edge labels such as deny or permit are also used in the network layout graph. All edges that have labels will not be considered as physical links and will not be allocated any IP addresses. This is because the edges are considered to be a part of the network layout firewall policy intention abstraction. Edge options used in Chapter 6.2 such as *protocol*; *dPort*; *dest*; and *sPort* are used to represent firewall rule tuple options in this section as well. This enables specific devices within the proposed network to have unique or isolated firewall rules that other devices in the same realm as the device in question will not have. For example, if a device belongs to a firewall realm that has a rule denying secure communication (ssh), this abstract representation will enable network administrators represent an isolated firewall rule to enable a device with the realm to communicate using ssh. It should be noted that NePAS will not work properly if the edges representing firewall rules are not adequately labeled as either permit or deny and the edge options not specified adequately as discussed in Chapter 6.2.

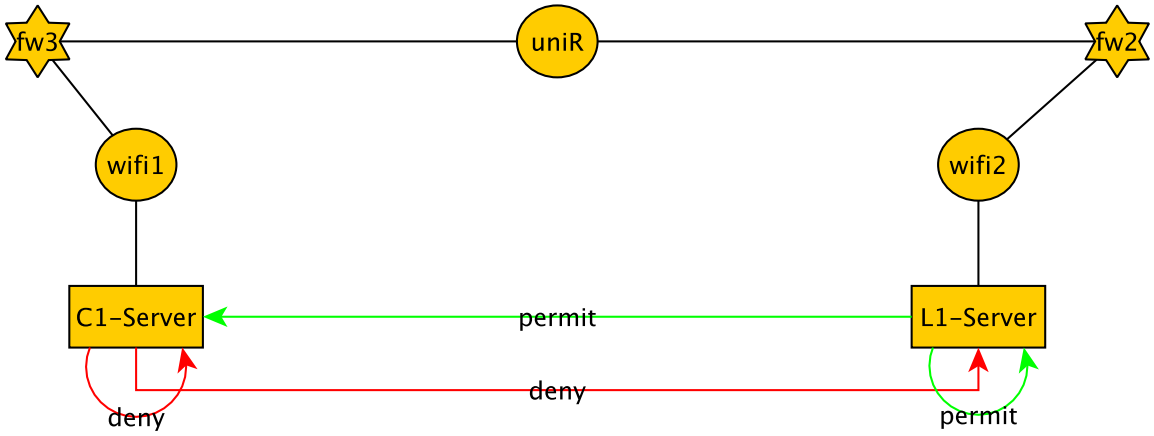

Figure 6.3: Proposed University Network Topology

Figure 6.3 above depicts the network layout of the proposed hypothetical university that have its security policy intention represented in the previous section. Table 6.3 below shows the firewall-enabled devices and the realm(s) each of them belongs to in the example.

Table 6.3: Firewall Device-Realm Mapping in Proposed University Network

S/No	Devices	Realm
1	C1-Server	campus
2	L1-Server	lab

The example in Figure 6.3 above shows the abstraction of four additional firewall rule relations for the proposed hypothetical university network. Table 6.4 below shows the detailed firewall tuple options for the four additional firewall rule relations. The four firewall rules abstracted in the network layout of the proposed university network are detailed below.

Table 6.4: Proposed University Network-Level Firewall Policy Detail

Rule No	S Device	D Device	Action	Protocol	D Port	S/D Address	S Port
R1	L1-Server	C1-Sever	permit	TCP	65432	NULL	23456
R2	C1-Server	L1-Server	deny	TCP, UDP	sftp	NULL	NULL
R3	L1-Server	L1-Server	permit	UDP	62300	s 2.3.4.5	56431
R4	C1-Server	C1-Server	deny	IP	NULL	d 4.5.6.7	NULL

The first rule R1 is used to permit communication from L1-Server on port 23456 to C1-Server on port 65432 over TCP protocol. The second rule is used to deny communication from C1-Server to L1-Server using secure file transfer protocol (sftp) over TCP and UDP protocols. The third rule R3 is used to permit an external host, 2.3.4.5, on port 56431 to communicate to L1-Server on port 62300 over UDP protocol. The fourth or last rule R4 is used to deny C1-Server from communicating with an external host, 4.5.6.7, over IP protocol.

6.4 Firewall Anomaly Resolution

This section outlines the techniques that have been implemented for NePAS to ensure the final low level firewall configurations to be generated for deployment are anomaly-free. The techniques employ a range of algorithms to resolve well-known firewall anomalies that can amongst others deny or exclude a significant part of a network from communicating due to various anomalies in the firewall rules as discussed in Section 6.1.2 above. NePAS uses the techniques discussed in the following section to ensure firewalls are anomaly-free and consistent before the final low level configurations are generated for deployment. The techniques adopted by NePAS in this phase are also to ensure final network deployments are completely accessible behind firewalls as the policy intention dictates. The techniques developed here are embedded within NePAS

and can be classified into two major parts: (i) intra firewall anomalies and (ii) inter firewall anomalies.

6.4.1 Intra Firewall Anomaly Resolution

This section looks at how NePAS resolves anomalies within a single firewall rule base. The following discussion does not imply the sequential order in which NePAS goes about resolving anomalies within a proposed network.

- *Shadowing Anomaly* - NePAS does not group IP addresses of devices within the proposed network into a combination of network address and mask when generating firewall rules for deployment. Due to the design concept, there is no way shadow anomalies can occur during low level firewall rule generation except when dealing with rules that have either the source or destination address set to *any*. To mitigate the possibility of shadow anomalies, NePAS has been designed to ensure device rules (which are specific or subset rules) come before rules that have their source and/or destination addresses of *any* (which are general or superset rules). NePAS thus employs address rule ordering during low level compilation as follows :
  <permit> <protocol> <host> <host> ...
  <deny> <protocol> <host> <host> ...
  <permit> <protocol> <host> <any> ...
  <deny> <protocol> <host> <any> ...
  <permit> <protocol> <any> <host> ...
  <deny> <protocol> <any> <host> ...
  <permit> <protocol> <any> <any> ...
  <deny> <protocol> <any> <any> ...
- *Redundancy Anomaly* - NePAS generates low level firewall rules on a per device basis between all firewall-enabled devices within a proposed network. This method on its own cannot ensure no additional or redundant firewall rules are generated during low level compilation. There are two ways in which redundant firewall rules can manifest during low level NePAS compilation of configuration files:
  (i) when a node belongs to multiple realms and the realms it belongs to have some identical firewall rule policy abstraction intention; and
  (ii) when a node has a network layout firewall rule abstraction that is identical to one of the firewall rule abstraction of the realm it belongs to in the policy intention.
  NePAS however checks every firewall rule that is being generated to ensure no identical copy of that firewall rule exists in the internal data structure it keeps them before the final configuration files are deployed. This technique ensures the final low level configuration generated for NePAS does not have any redundant firewall

rule.

- *Irrelevance Anomaly* - NePAS handles this anomaly conveniently because it generates firewall rule relations from either high level policy intentions or network layout intention abstractions on a per device basis. It uses the IP addresses it has automatically generated or specified by the user for each device when generating low level firewall rules and hence will not use an IP address outside the proposed network layout.

6.4.2 Inter Firewall Anomaly Resolution

This section discusses the techniques implemented for NePAS to ensure low level firewall rules generated are anomaly free during compilation:

- *Spuriousness Anomaly* - NePAS during low level firewall rule generation gets all the firewall devices between any two firewall-enabled network devices. It then proceeds to determine whether the action to be taken is to permit or deny network traffic as indicated in the policy intention. NePAS then adds the appropriate firewall rule in all the firewall devices between the two firewall-enabled network devices accordingly if its a permit rule. This technique ensures all firewall devices between two network devices are adequately configured to ensure no spuriousness anomaly occurs during low level firewall rule compilation. The redundancy anomaly resolution section below details what happens in the event the action to be taken is to deny network traffic.
- *Redundancy Anomaly* - As stated in the spuriousness anomaly resolution section above, NePAS gets all the firewall devices between any two firewall-enabled network devices. NePAS when dealing with a deny rule in an inter-firewall proposed network, configures a firewall rule relation only on the most upstream firewall only. This technique ensures no redundant firewall rules are configured in subsequent downstream firewalls between the two firewall-enabled network devices.
- *Shadowing Anomaly* - NePAS generates low level firewall rules for all firewall devices between any two firewall-enabled devices concurrently. This is to ensure all downstream firewalls between the two devices are adequately configured to permit communication between them. As already stated in the Intra Firewall Shadowing Anomaly above, all firewall devices within a distributed firewall proposed network will undergo the intra firewall shadow anomaly resolution technique. This will ensure no downstream firewall between the two firewall-enabled devices will be affected by this anomaly after low level compilation.

6.5 Firewall Compilation

This section discusses how low level configurations of firewall devices are generated by NePAS. To generate low-level platform specific firewall configurations for any proposed network, the following have been implemented. There are two sets of firewall rule abstraction as discussed above, these are as follows: (i) network layout rules, and (ii) policy intention rules. It should be noted that for all the two groups of firewall rule abstractions before they are generated, NePAS ensures no such identical rule has been generated. This technique is used by NePAS to ensure final firewall deployments are intra-firewall redundancy anomaly free.

6.5.1 Network Layout Rules

This section discusses the implementation methodology for generating firewall rules that have been abstracted within the firewall network layout graph as detailed in Chapter 6.3 above.

6.5.1.1 Device-Device Rules

This is the first stage of generating low-level firewall configuration commands for any proposed network using NePAS. Two network devices or nodes that are connected by an edge that has been labeled either deny or permit will be selected to start the configuration process. NePAS checks the neighbours between the two nodes and if they both have the same switch or router, a warning message will be logged and no firewall rule will be generated. This is because it will be a futile effort as the switch or router will forward packets before it reaches the firewall device. NePAS moves on to generate the associated firewall rule if both nodes are not behind the same switch or router. The system moves on to get the following tuple information from the connected edge options: protocol, destination port(dPort) and source port(sPort) as specified in the network layout graph. A list of all firewall devices along any path between the two selected firewall-enabled devices is compiled. At this point there are two options NePAS can take depending on the action to be taken as specified in the edge label.

Deny - this option occurs if the action to be taken as specified by the edge label is to deny communication between the two nodes. NePAS when dealing with an:

(i) intra firewall rule relation configures a firewall rule on the firewall device with the associated tuple options as specified in the edge option values. The firewall rule will be configured with an ACL name of **nepas-out** and will be applied on the outside interface of the firewall device.

(ii) inter firewall rule relation configures a firewall rule on the most upstream firewall with the associated tuple options as specified in the edge option values. This is to ensure inter

firewall redundancy anomaly does not occur when the proposed network is deployed. The firewall rule will be configured with an ACL name of **nepas-out** and will be applied on the outside interface of the most upstream firewall device.

The firewall rule abstraction R2 in Table 6.4 above illustrate this abstraction for the proposed university distributed firewall experiment in Figure 6.3 above. Two firewall rules with different protocols were configured with an access list named nepas-out in firewall fw3 as shown in Listing 6.2 below. The firewall rules were only configured on firewall fw3 so as to avoid inter firewall redundancy anomaly.

Listing 6.2: Proposed University Device-Device Deny Rules

```
access-list nepas-out extended deny tcp host 20.0.0.2 host 20.0.1.3
    eq sftp
access-list nepas-out extended deny udp host 20.0.0.2 host 20.0.1.3
    eq sftp
```

Permit - this option occurs if the action to be taken as specified by the edge label is to permit communication between the two nodes. NePAS when dealing with an:
(i) intra firewall rule relation configures a firewall rule on the firewall device with the associated tuple options as specified in the edge option values. The firewall rule will be configured with an ACL name of **nepas-out** and will be applied on the outside interface of the firewall device.
(ii) inter firewall rule relation configures a firewall rule with the associated tuple options as specified in the edge option values. The first set of firewall rule relations will be configured on the most upstream firewall device with an ACL name of **nepas-out** and is applied on its outside interface. The second set of firewall rule relations will be configured on subsequent firewalls between the two nodes and the most downstream firewall with an ACL name of **nepas-in** and is applied on the inside interfaces of all the firewalls.

The firewall rule abstraction R1 in Table 6.4 above illustrate this abstraction for the proposed university distributed firewall experiment in Figure 6.3 above. Two sets of firewall rules were configured in the firewalls in the university network. The first rule configured on firewall fw2 using an access list named nepas-out is used to permit L1-Server traffic going out of fw2 towards C1-Server. The second firewall rule configured on firewall fw3 using an access list named nepas-in is used to to permit C1-server to receive traffic from L1-Server. Listing 6.3 shows an excerpt of both from firewalls fw2 and fw3 respectively.

Listing 6.3: Proposed University Device-Device Permit Rules

```
Firewall fw2
access-list nepas-out extended permit tcp host 20.0.1.3 host 20.0.0.2
    eq 65432
```

```
Firewall fw3
access-list nepas-in extended permit tcp host 20.0.1.3 host 20.0.0.2
    eq 65432
```

6.5.1.2 Device-External or External-Device Rules

This is the second stage of generating low-level firewall configuration commands for any proposed network using NePAS for this group of abstractions. NePAS during the configuration process ensures that both the source and destination node of an edge labeled either deny or permit is the same value when going through the network layout graph of the proposed network in order to process these types of firewall rules. The edge options: protocol, dPort, sPort and dest are then processed by assigning them into the configuration snippet so as to generate the low-level platform specific firewall rule. It should be noted that when the *dest* value is not specified or specified does not have characters s or d preceding the IP address or IP network, NePAS will generate a warning message and not process the firewall rule abstraction.The firewall rule configuration to be generated come in one of two variants depending on the *dest* value specified:
(i) the first variant of this firewall rule relation occurs when the dest value specified has a character *d* preceding the IP address or IP network and net mask specified. In this situation, the destination address of the firewall rule will be the specified IP address or network in the edge option *dest* provided while the source address will be the IP address of the network device. The generated firewall rule based on the firewall tuple information above is then applied on the most upstream firewall of the network device involved. This firewall rule will be configured using an ACL name of **nepas-out** and applied on the outside interface because the host device that has this firewall rule is the source trying to communicate with an outside IP Address or network.

The firewall rule abstraction R4 in Table 6.4 above illustrate this abstraction for the proposed university distributed firewall experiment in Figure 6.3 above. A firewall rule was configured with an access list named nepas-out on firewall fw3 as shown in Listing 6.4 below.

Listing 6.4: Proposed University Device-External Rules

```
access-list nepas-out extended deny ip host 20.0.0.2 host 4.5.6.7
```

(ii) the second variant of the firewall rule relation occurs when the *dest* value specified has a character *s* preceding the specified IP address or network. In this situation, the source address of the firewall rule will be the specified IP address or network in the edge option *dest* provided while the destination address will be the IP address of the network device. The generated firewall rule based on the firewall tuple information above is then applied on the most upstream firewall device. This firewall rule will be configured using

an ACL name of **nepas-in** and applied on the inside interface because the host device that has this firewall rule is the destination.

The firewall rule abstraction R3 in Table 6.4 above illustrate this abstraction for the proposed university distributed firewall experiment in Figure 6.3 above. A firewall rule was configured with an access list named nepas-in on firewall fw2 as shown in Listing 6.5 below.

Listing 6.5: Proposed University External-Device Rules

```
access-list nepas-in extended permit tcp host 2.3.4.5 eq 56431 host
   20.0.1.3 eq 62300
```

6.5.1.3 Device-Any Rules

This is the third stage of generating low-level firewall rule configuration commands for any proposed network using NePAS for this group of abstractions. A specialised node labeled *any* will need to be created in the network layout graph to achieve this abstraction. NePAS gets all the firewall tuple information from the edge options that connects a network device with the node *any* that has been created. Depending on the action to be taken as specified by the edge label, there are two ways in which NePAS configures these types of firewall rule relations:

Deny: this option occurs if the action to be taken by the firewall rule is to deny communication. NePAS when dealing with an:

(i) intra firewall rule relation configures a firewall rule with the associated tuple options as indicated in the edge option values. The firewall rule is configured with an ACL named **nepas-out** and is applied on the outside interface of the firewall device.

(ii) inter firewall rule relation however configures the command on the most upstream firewall to the device in question. This is to ensure that no inter firewall redundancy anomaly occurs by configuring the other firewalls in the proposed network. The firewall rule is configured using an ACL named **nepas-out** which will be applied on the outside interface during low level configuration deployment.

Permit: this option occurs if the action to be taken by the firewall rule is to permit communication. NePAS when dealing with an:

(i) intra firewall rule relation configures a firewall rule with the associated tuple options as indicated in the edge option values on the firewall device. The firewall rule is configured with an ACL named **nepas-out** which will be applied on the outside interface of the firewall device.

(ii) inter firewall rule relation configures on all the firewall devices within the network layout a firewall rule with the associated tuple options as indicated in the edge option values. The firewall rule in the most upstream firewall device is configured using an ACL named **nepas-out** which will be applied on the outside interface of the device. For all

the other firewalls in the proposed network, the firewall rule is configured using an ACL named **nepas-in** and will be applied on their inside interfaces.

6.5.1.4 Any-Device Rules

This abstraction is the inverse of the rule discussed above. NePAS gets all the firewall tuple information from the edge options that connects a network device with the node *any* that has been created. Depending on the action to be taken by firewall rule as indicated by the edge label, there are two ways in which NePAS configures these types of firewall rule relations:
Deny: this option occurs if the action to be taken by the firewall rule is to deny communication. NePAS when dealing with an:
(i) intra firewall rule relation configures a firewall rule with the associated tuple options as indicated in the edge option values. The firewall rule is configured with an ACL named **nepas-in** and is applied on the inside interface of the firewall device.
(ii) inter firewall rule relation however configures the command on all the firewall devices in the proposed network except the most upstream firewall. This is to ensure that no inter firewall redundancy anomaly occurs by configuring the most upstream firewall device in the proposed network. The firewall rule is configured with an ACL named **nepas-in** and is applied on the inside interface of all firewall devices within the proposed network except the most upstream firewall.
Permit: this occurs if the action to be taken by the firewall rule is to permit communication. NePAS when dealing with an:
(i) intra firewall rule relation configures a firewall rule with the associated tuple options as indicated in the edge option values on the firewall device. The firewall rule is configured with an ACL named **nepas-in** and is applied on the inside interface of the firewall device.
(ii) inter firewall rule relation configures a firewall rule with the associated tuple options as indicated in the edge option values on all the firewall devices including the most upstream firewall in the proposed network. The firewall rule is configured with an ACL named **nepas-in** and is applied on the inside interface of the most upstream firewall device. All the other firewalls in the proposed network will however have a firewall rule configured with an ACL named **nepas-out** and applied on the outside interface.

6.5.2 Policy Intention Rules

This section discusses the implementation methodology of firewall rules that have been generated within the firewall policy intention graph detailed in Chapter 6.2 above.

6.5.2.1 Realm-Realm Rules

This is the first stage of generating low-level firewall configuration commands for any proposed network using NePAS for this group of firewall abstractions. Two network layout nodes that have a firewall policy assigned using the *fwpol* option in the network layout graph are selected to start the configuration process. NePAS checks the neighbours of the two nodes and if they both have the same switch or router, a warning message will be logged and no firewall rule will be generated. This is because it will be a futile effort as the switch or router will forward packets before it reaches the firewall device. NePAS moves on to generate the associated firewall rule if both nodes are not behind the same switch or router by getting the following firewall tuple options from the policy intention graph: protocol, dPort and sPort between the two realms. Next, a list of all firewall devices between the two selected firewall enabled devices is compiled. NePAS then checks whether the edge label between the two firewall realms in the policy intention graph is deny or permit. There are two ways in which NePAS will deal with such an abstraction depending on the edge label:
Deny: this option occurs if the action to be taken by the firewall rule is to deny communication between the two nodes. NePAS when dealing with an:
(i) intra firewall rule relation, a firewall rule is configured on the firewall device with the associated tuple options as specified in the edge (connecting the realms of the two nodes in the policy intention graph) option values. The firewall rule is configured using an ACL named **nepas-out** and applied on the outside interface of the firewall device.
(ii) inter firewall rule relation configures a firewall rule on the most upstream firewall with the associated tuple options as specified in the edge (connecting the realms of the two nodes in the policy intention graph) option values. This is to ensure inter firewall redundancy anomaly does not occur when the proposed network is deployed. The firewall rule will be configured with an ACL name of **nepas-out** and will be applied on the outside interface of the most upstream firewall device.

The firewall rule abstraction R6 in Table 6.2 above illustrate this abstraction for the proposed university distributed firewall experiment in Figure 6.3 above. A firewall rule was configured with an access list named nepas-out on firewall fw2 as shown in Listing 6.6 below. The firewall rule was only configured on firewall fw2 so as to avoid inter firewall redundancy anomaly.

Listing 6.6: Proposed University Realm-Realm Deny Rules

```
access-list nepas-out deny ip host 20.0.1.3 host 20.0.0.2
```

Permit: this option occurs if the action to be taken by the firewall rule is to permit communication between the two nodes. NePAS when dealing with an:
(i) intra firewall rule relation, a firewall rule is configured on the firewall device with the

associated tuple options as specified in the edge option values. The firewall rule will be configured with an ACL name of **nepas-out** and applied on the outside interface of the firewall device.
(ii) inter firewall rule relation, an access rule is placed in all firewall devices in the compiled list of firewalls between the two devices. This is to ensure inter firewall spuriousness anomaly is resolved during deployment of low level firewall rule relations. The firewall rule is configured using an ACL named **nepas-out** and applied on the outside interface of the most upstream firewall. All the other firewalls between the two devices will have a firewall rule relation with ACL named **nepas-in** and applied on their various inside interfaces.

The firewall rule abstraction R7 in Table 6.2 above illustrate this abstraction for the proposed university distributed firewall experiment in Figure 6.3 above. Two sets of firewall rules were configured in the firewalls of the university network. The first rule configured on firewall fw3 using an access list named nepas-out is used to permit C1-Server traffic going out of fw3 towards L1-Server. The second rule configured on firewall fw2 using an access list named nepas-in so as to permit L1-server to receive traffic from C1-Server. Listing 6.7 shows an excerpt of both from firewalls fw2 and fw3 respectively.

Listing 6.7: Proposed University Realm-Realm Permit Rules

```
Firewall fw2
access-list nepas-in permit tcp host 20.0.0.2 eq 54321 host 20.0.1.3
    eq 54321

Firewall fw3
access-list nepas-out permit tcp host 20.0.0.2 eq 54321 host 20.0.1.3
     eq 54321
```

6.5.2.2 Realm-External or External-Realm Rules

This is the second stage of generating low-level firewall configuration commands for any proposed network using NePAS for this group of firewall abstractions. NePAS checks the *fwpol* of network devices in a proposed network and ensures that both the source and destination realms of an edge in the firewall policy intention graph are the same. The edge options such as: protocol, dPort, sPort and dest are then processed by assigning them into the configuration snippet so as to generate the low level platform specific firewall rule. It should be noted that when the *dest* value is not specified, NePAS will generate a warning message and not process the firewall rule abstraction. The firewall rule configuration to be generated can come in one of two variants depending on the *dest* value specified:
(i) the first variant of this firewall rule relation occurs when the *dest* value specified has

a character *d* preceding the IP address or IP network and net mask specified. In this situation, the destination address of the firewall rule will be the specified IP address or network in the edge option *dest* provided while the source address will be the IP address of the network device with the assigned *fwpol* value. A firewall rule is generated based on the firewall tuple information above and applied to the most upstream firewall of the network device involved. This firewall rule will be configured using an ACL name of **nepas-out** and applied on the outside interface because the host device that has this firewall rule is the source.

The firewall rule abstraction R9 in Table 6.2 above illustrate this abstraction for the proposed university distributed firewall experiment in Figure 6.3 above. Four sets of firewall rules are configured on firewall fw3 with an access list named nepas-out for this rule abstraction. Each destination port will have a firewall rule configured for each destination IP address as shown in Listing 6.8 below.

Listing 6.8: Proposed University Realm-External Rules

```
access-list nepas-out permit tcp host 20.0.0.2 host www.facebook.com
    eq 80
access-list nepas-out permit tcp host 20.0.0.2 host www.facebook.com
    eq 8080
access-list nepas-out permit tcp host 20.0.0.2 host www.gorillavid.in
     eq 80
access-list nepas-out permit tcp host 20.0.0.2 host www.gorillavid.in
     eq 8080
```

(ii) the second variant of the firewall rule relation occurs when the *dest* value specified has a character *s* preceding the IP address or IP network specified. In this situation, the source address of the firewall rule will be the IP address or IP network specified in the edge option *dest* provided while the destination address will be the IP address of the network device with the assigned *fwpol* value. The generated firewall rule based on the firewall tuple information above and applied on the most upstream firewall device. This firewall rule will be configured using an ACL name of **nepas-in** and applied on the inside interface because the host device that has this firewall rule is the destination.

The firewall rule abstraction R5 in Table 6.2 above illustrate this abstraction for the proposed university distributed firewall experiment in Figure 6.3 above. A firewall rule was configured with an access list named nepas-in on firewall fw2 as shown in Listing 6.9 below.

Listing 6.9: Proposed University External-Realm Rules

```
access-list nepas-in extended permit ip host 1.2.3.4 host 20.0.1.3
```

6.5.2.3 Realm-Any Rules

This is the third stage of generating low-level firewall configuration commands for any proposed network using NePAS for this group of firewall abstractions. All firewall policy intention realms that have some relationship with the any realm from the policy intention graph are added into a list. As we go through the network layout graph, NePAS gets network nodes that have been assigned *fwpol* value that are contained in the list created. NePAS then takes the firewall policy intention realm of the network node and goes through the policy intention graph till we get a match with the any realm. NePAS then gets all the firewall tuple properties from the edge (connecting the node to any) options of the policy intention graph. Depending on the action to be taken by the firewall rule, there are two ways in which NePAS configures these types of firewall rule relations:
Deny: this option occurs if the action to be taken by the firewall rule is to deny communication. NePAS when dealing with an:
(i) intra firewall rule relation configures a firewall rule with the associated tuple options as indicated in the edge option values on the firewall device. The firewall rule is configured using an ACL named **nepas-out** and applied on the outside interface of the firewall device. This is because the node that has the realm label as its *fwpol* will be the source address of the firewall rule while *any* will be the destination address.
(ii) inter firewall rule relation configures a firewall rule with the associated tuple options as indicated in the edge option values on the most upstream firewall device of the node that carries the *fwpol* value. This is to ensure that no inter firewall redundancy anomaly occurs by configuring any other firewall device within the proposed network with the same rule that has already been denied communication by the most upstream firewall. The firewall rule is configured using an ACL named **nepas-out** and applied on the outside interface of the most upstream firewall device.

The firewall rule abstraction R8 in Table 6.2 above illustrate this abstraction for the proposed university distributed firewall experiment in Figure 6.3 above. Two sets of firewall rules will be configured on firewalls fw3 with an access list named nepas-out for the two protocols specified. The firewall rules were only configured on firewall fw3 so as to avoid inter firewall redundancy anomaly. The firewall rules configured are shown in Listing 6.10 below.

Listing 6.10: Proposed University Realm-Any Rules

```
access-list nepas-out deny tcp host 20.0.0.2 any eq ftp
access-list nepas-out deny tcp host 20.0.0.2 any eq telnet
```

Permit: this option occurs if the action to be taken by the firewall rule is to permit communication. NePAS when dealing with an:
(i) intra firewall rule relation configures a firewall rule with the associated tuple options as indicated in the edge option values on the firewall device. The firewall rule is configured

with an ACL named **nepas-out** and applied on the outside interface of the firewall device.
(ii) inter firewall rule relation configures a firewall rule with the associated tuple options as indicated in the edge option values on all the firewall devices within the network layout. The firewall rule is configured using an ACL named **nepas-out** and applied on the outside interface of the most upstream firewall device. For all the other firewalls in the proposed network, the firewall rule is configured using an ACL named **nepas-in** and applied on their inside interfaces.

The firewall rule abstraction R3 in Table 6.2 above illustrate this abstraction for the proposed university distributed firewall experiment in Figure 6.3 above. Four different firewall rules will be on the firewalls within the proposed network layout. The first two firewall rules will be configured on firewall fw2 using an access list named nepas-out that will enable L1-Server to initiate communication over TCP protocol on ports either 40728 or 3689. The other two firewall rules will be configured on firewall fw3 using an access list named nepas-in that will enable C1-Server receive communication over TCP protocol on ports 40728 or 3689. The firewall rules configured are shown in Listing 6.11 below.

Listing 6.11: Proposed University Realm-Any Rules

```
Firewall fw2
access-list nepas-out permit tcp host 20.0.1.3 any eq 40728
access-list nepas-out permit tcp host 20.0.1.3 any eq 3689

Firewall fw3
access-list nepas-in permit tcp host 20.0.1.3 any eq 40728
access-list nepas-in permit tcp host 20.0.1.3 any eq 3689
```

6.5.2.4 Any-Realm Rules

This is the fourth stage of generating low-level firewall rule configuration commands for any proposed network using NePAS for this group of firewall abstractions. The approach carried out in the initial stages of this approach is the same as the one discussed above. Depending on the action to be taken by the firewall rule, there are two ways in which NePAS configures these types of firewall rule relations:
Deny: this option occurs if the action to be taken by the firewall rule is to deny communication. NePAS when dealing with an:
(i) intra firewall rule relation, a firewall rule is configured with the associated tuple options as indicated in the edge option values. The firewall rule is configured with an ACL named **nepas-in** and applied on the inside interface of the firewall device. This is because the node that has the realm label as its *fwpol* will be the destination address of the firewall rule while *any* will be the source address.
(ii) inter firewall rule relation however configures the command on all the firewall devices

in the proposed network except the most upstream firewall. This is to ensure that no inter firewall redundancy anomaly occurs by configuring the most upstream firewall device in the proposed network. The firewall rule is configured with an ACL named **nepas-in** and applied on the inside interface of all firewall devices within the proposed network except the most upstream firewall. This is because the firewall rule is aimed at denying communication from devices both within and outside the network to the network device with the *fwpol* value.

Permit: this option occurs if the action to be taken by the firewall rule is to permit communication. NePAS when dealing with an:

(i) intra firewall rule relation, a firewall rule configured with the associated tuple options as indicated in the edge option values on the firewall device. The firewall rule is configured with an ACL named **nepas-in** and applied on the inside interface of the firewall device.

(ii) inter firewall rule relation configures a firewall rule with the associated tuple options as indicated in the edge option values on all the firewall devices in the proposed network. The firewall rule is configured with an ACL named **nepas-in** and applied on the inside interface of the most upstream firewall device. All the other firewalls in the proposed network will however have a firewall rule configured with an ACL named **nepas-out** and applied on their outside interfaces.

The firewall rule abstraction R10 in Table 6.2 above illustrate this abstraction for the proposed university distributed firewall experiment in Figure 6.3 above. There are two sets of firewall rules configured for this rule abstraction in the proposed network topology. The first firewall rule was configured on firewall fw3 using an access list named nepas-in which is meant to permit L1-Server to initiate communication with C1-Server over TCP using ssh. The second firewall rule to be configured on firewall fw2 using an access list named nepas-out which is meant to permit L1-Server to initiate communication with C1-Server in the proposed network. The firewall rules configured are shown in Listing 6.12 below.

Listing 6.12: Proposed University Any-Realm Rules

```
Firewall fw2
access-list nepas-out permit tcp any host 20.0.0.2 eq ssh

Firewall fw3
access-list nepas-in permit tcp any host 20.0.0.2 eq ssh
```

6.5.2.5 Any-Any Rules

This is the last stage of generating low-level firewall configuration commands for any proposed network using NePAS for this group of firewall abstractions. The rules generated in this part will be configured in all firewall devices within the proposed network.

NePAS goes through the policy intention graph and match the edges where both the source and destination nodes are *any*. The firewall tuple information from the edge options are then gotten and a rule based on them is appended into all the firewalls of the proposed network. There are three types of firewall rules that can be configured for this group of policy intention firewall rules:

1. This type of *any-any* rule is generated when both source and destination addresses that will be configured for the firewall rule relation will be set to *any* and assigned to all firewalls in the proposed network. These firewall rules are configured when the *dest* option of the edge option (tuple details) is set to 'NULL'. These firewall rules will be configured using an ACL named **nepas-any** and applied to the global interface on all the firewall devices within the proposed network.

   The firewall rule abstraction R1 in Table 6.2 above illustrate this abstraction for the proposed university distributed firewall experiment in Figure 6.3 above. A firewall rule was configured with an access list named nepas-any on both firewalls fw2 and fw3 as shown in Listing 6.13 below.

   Listing 6.13: Proposed University Any-Any Rules

   ```
   Firewall fw2
   access-list nepas-any permit icmp any any

   Firewall fw3
   access-list nepas-any permit icmp any any
   ```

2. This type of *any-any* rule is generated when the *dest* option of the edge options (tuple details) has been specified. The first set of these firewall rules occurs when the *dest* value specified has a character *d* preceding the IP address or IP network specified. In this situation, the destination address of the firewall rule will be the specified IP address or network in the edge option *dest* provided while the source address will be set to *any*. This firewall rule will be configured on all firewall in the proposed network using an ACL name of **nepas-out** and applied on their outside interfaces.

   The firewall rule abstraction R2 in Table 6.2 above illustrate this abstraction for the proposed university distributed firewall experiment in Figure 6.3 above. A firewall rule was configured with an access list named nepas-any on both firewalls fw2 and fw3 as shown in Listing 6.14 below.

   Listing 6.14: Proposed University Any-External Rules

   ```
   Firewall fw2
   access-list nepas-out deny tcp any host www.facebook.com eq 80
   access-list nepas-out deny tcp any host www.gorillavid.in eq 80
   ```

```
access-list nepas-out deny tcp any host www.facebook.com eq 8080
access-list nepas-out deny tcp any host www.gorillavid.in eq
   8080

Firewall fw3
access-list nepas-out deny tcp any host www.facebook.com eq 80
access-list nepas-out deny tcp any host www.gorillavid.in eq 80
access-list nepas-out deny tcp any host www.facebook.com eq 8080
access-list nepas-out deny tcp any host www.gorillavid.in eq
   8080
```

3. This type of *any-any* rule is generated when the *dest* option of the edge options (tuple details) has been specified. The first set of these firewall rules occurs when the *dest* value specified has a character *s* preceding the IP address specified. In this situation, the source address of the firewall rule will be the specified IP address or network in the edge option *dest* provided while the destination address will be set to *any*. This firewall rule will be configured on all firewall in the proposed network using an ACL name of **nepas-in** and applied on their inside interfaces.

6.6 Closing Remarks

This chapter of the thesis gives a thorough insight into firewalls, their classifications and how they work. The chapter proceeds to showcase how we implemented our firewall relationships abstractions using two sets of firewall specifications: policy intention rules and network layout rules. The chapter then looks at how the various phases of NePAS are used for expressing high level firewall abstractions down to generating anomaly-free low level platform specific configurations. A small example is used to give a step by step assessment of the how NePAS handles firewall policy intention abstractions. The next chapter will be used to discuss how cyber security competitions are organised and our proposed high level competition intention abstraction.

Chapter 7

Cyber Security Competition Abstractions

7.1 Background

Cyber security competitions (CDX) are computer security exercises where participants compete in information assurance. During such competitions, participants are given a piece of infrastructure composed of vulnerable services (also called flags). The participants are sometimes required to defend these flag by fixing the flaws and/or exploiting the the flags of the opponents depending on the competition's objectives. Such game environments are deployed in well controlled sandpits due to the consequences of activities carried out during the competitions. These competitions provide a very effective way of gaining hand-on practical computer security training for participants involved. These competitions have been wildly adopted by various educational institutions as part of their curricula and other professional competitions. As highlighted by Kenneth Geers [48], robust cyber security competitions need a team-oriented approach. The participants of a CDX competition can be broadly grouped into two: non-combatants and combatants.

1. Noncombatant - this category of participants is composed of people who design, deploy, monitor and evaluate the infrastructure before, during and after the competition. There are two distinct groups of participants with different responsibilities under this group which are as follows:

   - *Organising teams*
     This group of participants are usually called the *white team* and are responsible for among others:-

     – design and implementation of the competition infrastructure,

  - deciding whether the scoring mechanism of the competition should be automated or manual. This group are responsible for creating the scoring engine if an automated system is chosen.
  - recording the network traffic of the infrastructure during the competition.

- *Management teams*
  This group of participants is usually composed of information security professionals or academic faculty members as the case maybe. These participants are usually called the *green team* and are responsible for among others:-
  - assessing the teams participating in the competition in terms of their ability to carry out their objectives[76]
  - maintaining the equipment of the infrastructure during the competition[67]
  - handling disputes between red and blue teams such as flag definition when the teams are at an impasse[67]
  - administering competitions from a list of scenarios[76]
  - injecting business tasks to be carried out by the participating teams of the competition[76]

2. Combatants - this category of participants is composed of people who actively participate in exploiting and/or fixing vulnerabilities during the competition. There are two teams here which are as follows:

   - *Defensive Teams*
     This group of participants are usually composed of students or people trying to learn or improve their cyber security skills. These participants are usually not information assurance experts. These participants are usually called the *blue team* and their objectives include among others:
     - setting up and securing hardware provided by the green team[24]
     - carrying out business injects such as network redesign, account updates, blocking protocols and writing reports or presentation[24]
     - applying appropriate patches to the services in their infrastructure so as to prevent other teams from exploiting such services[30]
     - ensure all network services are always available as more service uptime equal greater score in some competitions.[48]

     Some of the benefits of being in blue team include among others:
     - give participants a sense of risk associated with hostile network environments[8]
     - reinforce importance of good communication skills[76]
     - platform to apply computer and network security related concepts and how it helps solve real-world problems[76]

  - helps participants improve team work skills [76]
  - provides motivation for participants to win hence investing time to ensuring their services are working securely.

  To avoid the blue team from completely patching or locking down their network infrastructure, the blue teams are not allowed to filter or block entire network traffic during compeitions[96].

- *Offensive teams*
  This group of participants are usually composed of penetration testing professionals from the military, government organisation or cyber security industry representatives. However in some cases these team members are composed of students [68]. These participants are usually called the *red team* and their objectives include among others:

  - disrupting the confidentiality, integrity and availability of the blue team's infrastructure.
  - effectively penetrating the blue team's infrastructure and performing some tasks as a way of demonstrating they have achieved their objectives such as creating a user or root owned text file after exploiting some of the blue teams' services.

  Red teams are expected to attempt the same exploit they have carried out on a particular blue team on the other blue teams accordingly. In some competitions, the red teams are given detailed prior knowledge of the blue teams infrastructure called *white box test* or they gather their own information independently called *black box test*. The red teams in some cases have some attack limitations that they are expected not to carry out such as:-

  - in most competitions, red teams are strictly prohibited from attacking the management infrastructure such as the scoring server[48]
  - not disrupting services, bringing down hosts or deleting files that could damage the effectiveness if the detection systems
  - red teams are barred from launching denial of service (DoS) attacks on the blue teams or management infrastructures in some competitions [24]

In the following section, the various components that are needed to organise cyber security competitions will be studied in a fashion similar to Furtuna et al's [44] proposed seven stage process for organising such competitions. The seven stage process are as follows: exercise objectives, approach, technical specifications, scenario, rules, metrics and lessons learned.

7.1.1 Approach

The white team members after identifying the set of objectives for a competition will then move on to deciding the approach to be used. This generally sets the tone for the subsequent sections of a cyber security competition. The approach a competition can take can be classified into three - defensive, offensive and comprehensive.

- Defensive Approach Competitions
  This competition approach features blue teams that are expected to in some cases setup or build their network infrastructure and defend it from internal and/or external red team attacks as the case maybe. The emphasis here is for the participants to ensure confidentiality, integrity and availability of their network infrastructure for the entire duration of the competition. This competition approach only appraises the blue team's ability to defend their network as red teams are usually brought in to exploit the blue team infrastructure. No score is awarded for blue teams that launch attacks on either the red team or management servers during the competition. In fact some competitions might decide to punish blue teams by deducting their points for launching any type of exploits during the competition. Some competitions that have used this approach in the past include:

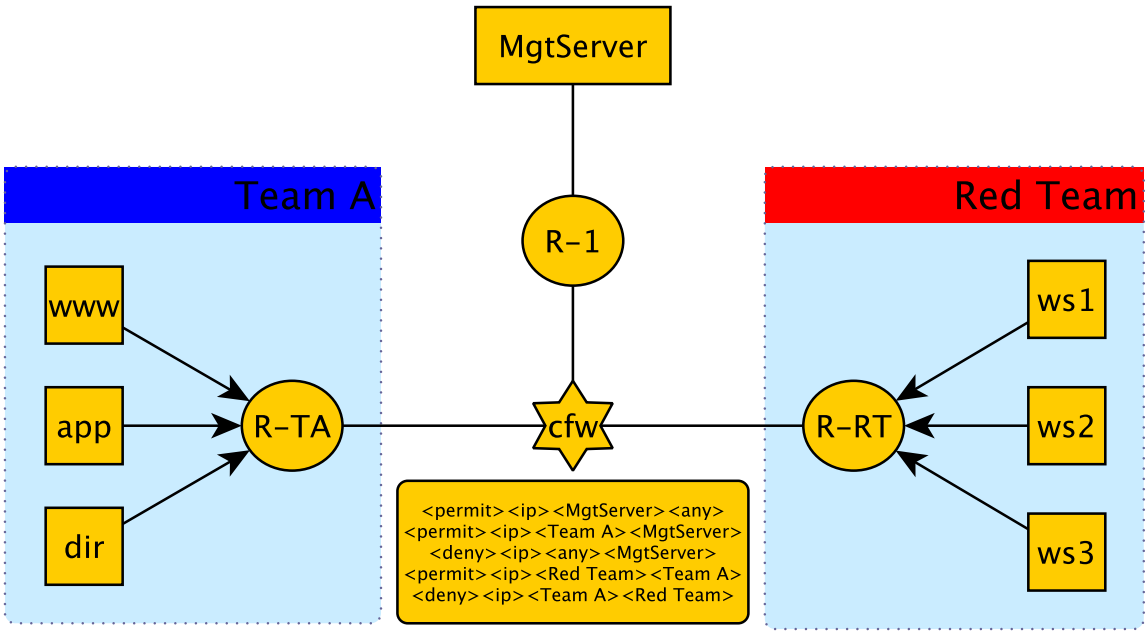

Figure 7.1: Example of a Typical CDX competition with a Defensive Approach

  - NSA CDX [8] which pits various US military academies is one of the oldest and most popular competitions that uses a defensive approach. Some of the objectives of the participants in the competition include the following:

(i) confidentiality, integrity and availability; (ii) complete various business injects during the competition; (iii) submit reports with all the required documentation of how they defended their infrastructure.

– The Baltic Cyber Shield 2010 [48] used this approach whereby the participants of the competition were required to defend the computer network of a power supply company that was under sophisticated cyber attacks sponsored by a terrorist group (the red team).

There are many other competitions that have chosen to use this approach in the past, these include [78] [36] [24] [76].

- Offensive Approach Competitions
  These competitions feature only red teams that are expected to launch attacks on the infrastructure that have been provided to them by the white team. This approach does not feature a blue team as all participating teams are expected to capture the flags in the infrastructure supplied to them. The emphasis of this approach is predominantly based on how well the teams are able to successfully exploit vulnerabilities or capture the flags within their opponent's network infrastructure and no score is awarded for defending the their own infrastructure.

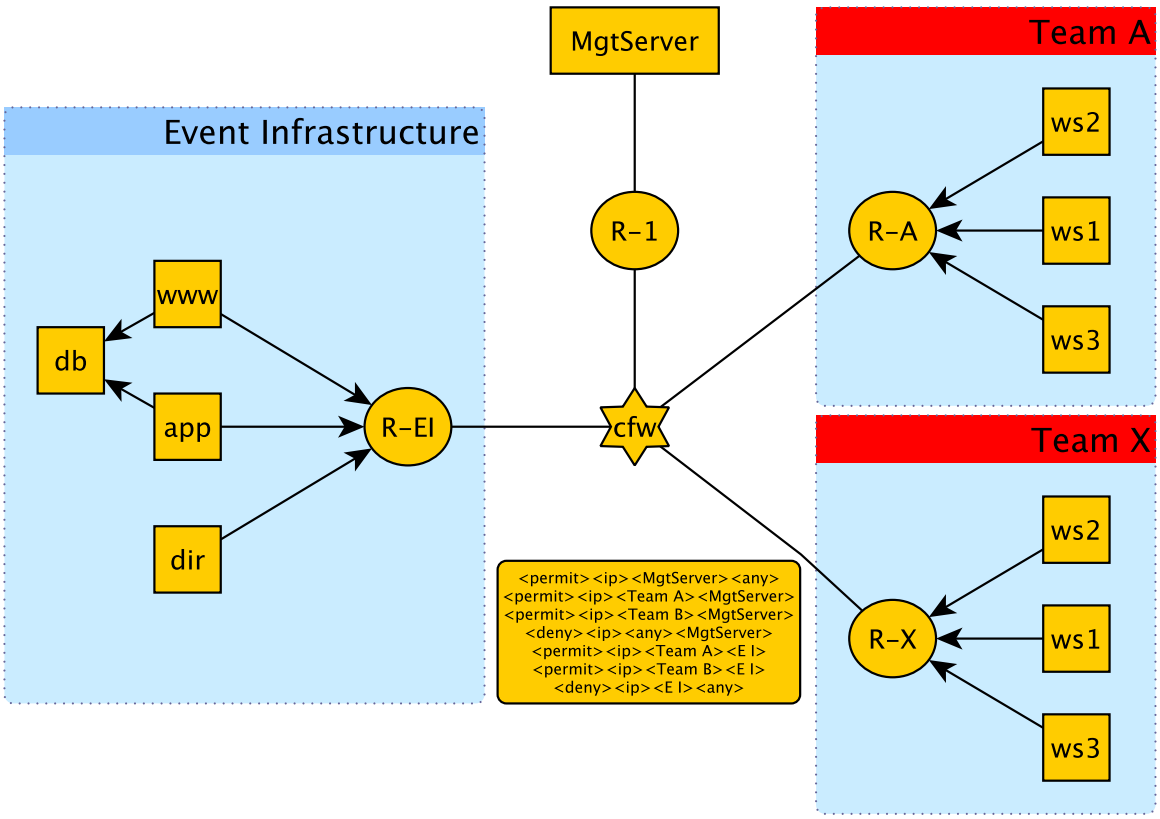

Figure 7.2: Example of a Typical CDX competition with an Offensive Approach

An example of a competition that uses this kind of approach is the Blunderdome exercise [68], where participants are expected to capture the flags within a given infrastructure by performing various exploits from kernel-level exploits to creating root-owned files in order to achieve part of their objectives. Many competitions such as [57] [38] [27] [3] have chosen to use this approach in the past.

This approach comes with some legal issues where by participants need to be aware of the ethical and legal repercussions of trying some of the attack methods they have learnt during the competition in the real-world.

- Comprehensive Approach Competitions
  These competitions feature teams that are expected to defend their network infrastructure while at the same time exploiting the infrastructure of opponents. The teams are scored for defending their infrastructure and also scored for each successful exploit they were able to achieve against their opponents.

  Some competitions that have used this approach include among others:-

  – University of Wisconsin developed a "cyberwar" competition using a comprehensive approach where participants competed in a variant of capture the flag called *plant the flag*. During the competition participants were expected to fulfil the following objectives: (i) defending the infrastructure that has been provided to them (ii) try to penetrate the infrastructure of opponents by leveraging weak accounts and passwords and placing a file with certain contents on the opponents' machines.

  – In [99], the organisers also used a comprehensive approach of cyber security competitions were the objectives of the participants include among others: (i) install, configure and secure their systems with services such as telnet, SMTP, HTTP and many others; (ii) ensuring competition opponents do not exploit the services in their infrastructure; (iii) exploit the services in the infrastructure of their opponents.

  Many other competitions such as [99] [79] [102] have chosen to use this approach in the past to improve the particiants' defensive and penetration testing skills.

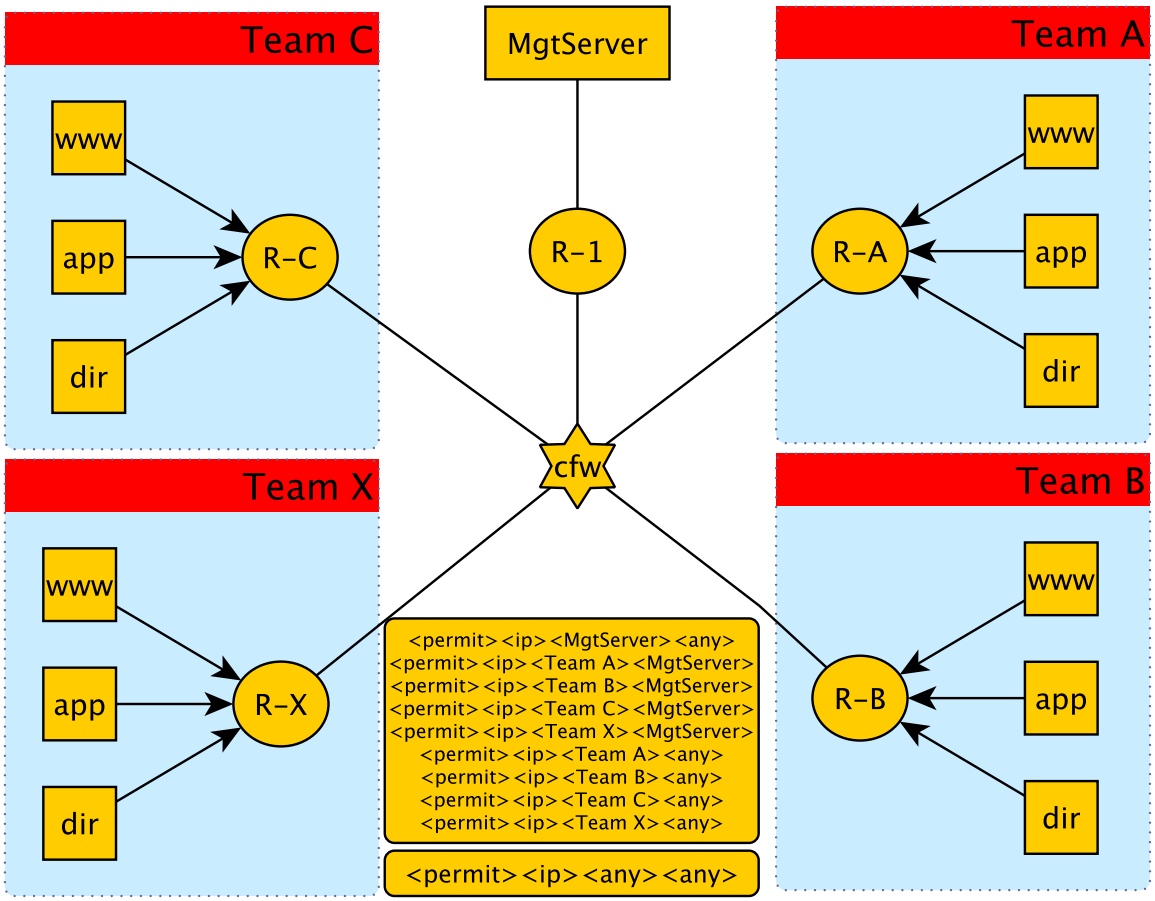

Figure 7.3: Example of a Typical CDX competition with a Comprehensive Approach

7.1.2 Technical Specification

The technical specification of a competition details the infrastructural requirement for the design, configuration and deployment by white team. The technical specification of a competition infrastructure details the various network devices, network policies and deployment methodology to be used. This section is where the competition approach detailed in 7.1.1 above is implemented.

All other participating teams whether red or blue teams are expected to have an identical infrastructure, equal bandwidth and maintain reliable communication during the competition so as to ensure a level playing field amongst the teams. Some white teams have chosen to isolate the competition infrastructure from other networks including the Internet so as to avoid any collateral damage that may arise. Most competitions prohibit participants from launching (i) denial of service (DoS) attacks as it affects the entire infrastructure and may disrupt the competition, and (ii) attacks against the management network especially the scoring and monitoring systems are prohibited and in some cases are filtered using firewalls.

The following are the major network devices; interconnection methods; configuration processes; and deployment platforms that have been used by various white teams during past competitions. We categorise this section into four major parts: servers, networking, configuration and deployment.

7.1.2.1 Servers

The servers used in a competition is a combination of services and applications installed on various kinds of operating systems that will be used for various purposes and with different security requirements as defined by the white team. We have grouped servers used by white teams during past competitions into three broad categories: attack servers; vulnerable servers; and management servers. These server groups and their respective requirements and usage are discussed in the following section:

1. **Management Servers**
   These servers are used by the green team members for the entire duration of the competition. The servers here are strictly for management purposes. The servers typically have access to the entire infrastructure of all the teams (blue and red) throughout the competition duration. Its is generally good practice to ensure the servers here are not part of the actual competition network if possible. This is so as to eliminate instances where participants try to filter the management servers from monitoring their infrastructure. The decision on whether to have distributed or centrally managed servers is taken by the white team. The servers deployed here include amongst others:

   (i) Repository servers - these servers are mainly used as a sort of warehouse for the participants to download applications and tools as required by the objectives of the competition. Many white teams have provided their participants such servers during competitions.

   (ii) Monitoring servers - these servers are mainly used to help automate green team responsibilities by using networking tools to monitor the infrastructure. These servers are considered so vital that Vasireaddy [94] sought to automate the configuration of the network monitoring tool Nagios for such competitions. In some competitions, a web dashboard is provided to help green teams to visualise and perform snapshots and revert servers to previous states.

   (iii) Scoring servers - these are the servers that host the various scoring scripts that are used for appraising the participating teams during a competition. The organisers create the various scripts that will be used here depending on the objectives of the competition. The server can also incorporate a submission mechanism where participants submit flags they have gotten through exploitation

of vulnerable services. The scoring server can also have a dashboard to show the points acquired by all participating teams during and after the competition. The use of scoring servers is highly popular amongst organisers of CDX competitions due to the ease with which points are awarded and displayed.

(iv) Situational awareness servers - these systems are used to visualise network traffic moving across the competition infrastructure. This functionality does not visualise real-time traffic but has some few minutes delay so as not to negatively impact the exercise. The system has a delay functionality so that participants will not be able to take pre-emptive measures when they realise opponents are trying to exploit flaws within their infrastructure.

(v) Illegal attacks: some traffic patterns such as denial of service (DoS) amongst others need to be mitigated as they can choke the entire competition infrastructure if not dropped accordingly. This functionality should be implemented using a firewall or an intrusion prevention system strategically placed at the exit points of all the network of the various teams.

2. **Vulnerable Servers**
   These servers are very important as they are composed of various vulnerabilities (or flags) to be used during the competition. Some competitions have chosen to go with operating systems that have not been patched so as to have more flags during the competition. These vulnerable operating systems host the flags that needs to be captured or planted as the requirement of the competition dictates. As already stated the white team is responsible for selecting the operating system type and version; service type and version, and vulnerable application to be used for the competition. Traffic generators have also being used by white teams such as [38] to simulate virtual user and may have some vulnerabilities in their make up as well. Mirkovic et al [75] decided to use a Skaion traffic generator to generate telnet, SSH and web application traffic which was considered legitimate for their Distributed Denial of Service(DDoS) competition coming from semi autonomous agents(bots). Table 7.1 above shows some of the flags used in some of the competitions reviewed for this thesis.

Team sheet		
Competition	Type of Service	Vulnerability
iCTF 2009	Stormlog	Off-by-One Overflow
	WeirdTCP	TCP IP Spoofing
	LityaHot	Session Fixation
BCS 2010	Apache	Path Traversal
	Linux	Accounts and Password Hacks
	Wordpress	File inclusion
oCTF	BlooperSurf	Port Knocking
	WarpZone	Format String
	Dr Mario	Input Checking Injection
SecLab	SMB	LSA Heap Overflow
	SMTP	
	WEP	WEP Replay Attack
2009 NSA CDX	SSH Server	SSH Reverse Tunnel
	Web server	Cross Site Scripting
	Windows Domain Cont.	Windows DNS Stack Overflow
Small Scale	FTP	Misconfiguration
	VNC	Starts silently at boot
iCTF 2008	PHP web server	Command Injection
	MySQL database	SQL Injection
	Application Server	Format String
CyberAttacks	Sohos Antivirus	Batch file viruses
		Poor definitions
	Lavasoft Anti Spyware	Default configurations

Table 7.1: Vulnerable services used in previous CDX competitions

3. **Attack Servers**

   These servers are very important because they contain the various tools and scripts that will enable participants (blue or red team members) to either exploit their opponent's infrastructure or fix vulnerabilities within their own infrastructure. These servers have access rules that will enable them access both their infrastructure and that of their opponents as the competition policy dictates. A good example of a server that can be used here is Kali Linux due to its wide range of specialised tools for penetration testing. There are instances where participants are expected to develop custom exploits or tools to fix vulnerabilities during the competition. There are also instances where participants are expected to download scripts or

tools that will be used for exploitation or fixing vulnerabilities from a repository server hosted within the management network of the competition. These servers are used by participants and sometimes the green teams to have access to the infrastructure as configured by the white teams with the help of documentations. Example of access methods used in previous competitions include: (i) Willems and Meinel [104] used a VNC connection embedded in web applications to allow participants to join competitions through their web browsers, (ii) During the NSA CDX, the white team used a virtual private network (VPN) technology such as SSH. Many other competitions[86] have used this VPN technology for enabling the participants access their infrastructure.

7.1.2.2 Networking

The network infrastructure of competitions need to be networked to achieve a number of objectives depending on the approach and in extension objectives. Competitions of an offensive nature such as the ones mentioned in 7.1.1 above need the individual teams to be isolated from each other so that the traffic from one team does not interfere with other teams for the entire duration. The open capture the flag(oCTF) [3] had a Fedora 13 router that was used to isolate the various participating teams and direct traffic across various VLANs during the competition. Another major networking concept used by white teams during competitions and sometimes integrated into the scenarios or objectives are sub networks (or firewall zones) within a participant team's infrastructure. During the Baltic Cyber Shield 2010 [48], the white team used a Linux-based firewall to create four zones comprising a DMZ, HMI, Internal and PLC within each participating team's infrastructure and strategically placed with various vulnerable servers and applications. Many other competitions such as [27] have used zones in their infrastructure for a variety of reasons in the past.

7.1.2.3 Configuration

After the white team has made a decision about the various types and specification of servers to be used and how to network the competition infrastructure, the next thing to think of is how to go about configuring the various devices in the infrastructure. Competition infrastructures are configured in one of the following two methods to achieve outlined objectives:

(i) *manually* by having the participants configure their respective network infrastructure using devices (such as routers, switches, firewalls, e.t.c.) provided and as required by the white teams during the competition. During some competitions, participants are required to configure their infrastructure to the specifications given by the white teams as part of tasks to be accomplished and are scored accordingly.

(ii) another means of configuration is the *semi automated* configuration of the competition infrastructure in which the white team only configures one virtual machine and replicates it using tools such as Norton Ghost. Many competitions have used this method of creating a virtual machine image and then replicating and distributing it to the various participating teams. This configuration methodology is not the same as the one proposed in this work as this configuration is only partially automated.

7.1.2.4 Deployment

At some point, usually before the configuration step discussed above, the white team needs to decide how the competition infrastructure will be deployed. There are two ways in which white teams deploy the competition infrastructure - physical and virtual. A number of factors tilt their decision in picking which way to go including but not limited to: (i) costs of acquiring the equipment; (ii) number of participants; (iii) availability of physical space for the equipment to be used e.t.c.

- *Physical*
  A number of white teams have decided to deploy their competitions using physical network devices. The main reason given by most white teams for this is to give the participants a more realistic environment to conduct competitions. For example in some competitions, all the participating teams were hosted within the same building so as to make it more realistic. Some other competitions that have used physical network devices for their infrastructure include but not limited to NSA CDX, DETER project, CCDC, DEFCON [5] e.t.c.

- *Virtualisation*
  Some other white teams have utilised virtualisation technology due to cost factors, availability of physical space and the flexibility it provides for taking snapshots of machines in a known working state as highlighted by the organisers of oCTF [3]. There are two types of virtualisation technologies used by white teams: para virtualisation and full virtualisation. Some examples of para virtualisation technologies used by white teams during their competitions include among others: OpenVZ [27], user mode linux(UML) [99] e.t.c. Some examples of full virtualisation technologies used by white teams during their competitions include among others: VMware [3] [79] [78] [30], KVM/QEMU [103]. Some limitations of virtualisation to deploy such competition infrastructures include: (i) they require real hardware firewalls, routers and switches cannot be achieved using virtualisation, (ii) performance of virtual machines running on the host machine gradually reduces as more virtual machines are created on it. (iii) in the case of para virtualisation technology, it limits the operating system to be used to the one used by the underlying host machine

7.2 CDX Policy Intention Abstraction

This section details the implementation methodology of CDX policy intention abstraction developed for NePAS. The nodes in this policy intention abstraction are used to represent group of network devices that belong to a particular cyber security competition team. The labels on the nodes are used in the cyber security competition infrastructure of a particular team during the compilation phase of any proposed experiment. The labels on the edges which includes comprehensive, defensive and offensive are used to represent the relationship between the two interconnected teams. It should be noted that color of the edges in the graph has no effect during the execution of the experiment as our system uses the labels and custom properties that have been defined by the network administrator during the execution of the experiment.

A realm in this policy intention abstraction is used to represent an entire cyber security competition infrastructure of a particular competing combatant team. Specifying a red team has been abstracted for NePAS by having the label of the realm prepended with *RT-* before the name of the team. Any team that has *RT-* before its name is considered a red team and no vulnerable infrastructure will be deployed for the team during the compilation phase of NePAS. The realms in CDX policy intention graph have two custom options: management and member.

The first realm option called *management* is itself an abstraction of all typical management servers described in section 1 above. As stated, the management servers depending on the underlying base image used to deploy it can include amongst others: scoring server, monitoring server, repository server, situational awareness server, etc. The management option takes a boolean value of either "TRUE" or "FALSE" and is used to enable management services for the particular team in question. The default value for this option is set to "TRUE" and management services is provided for the team except otherwise indicated.

The second realm option is called *members* and has been implemented for NePAS to identify the number of team members within a particular competing team. The number of team members indicated here is used to determine the number of attack servers (discussed in section 3 above) to be deployed. This option takes an integer value and has a default value of 1 when no other value has been specified. It should be noted that when an integer value of 0 is assigned to any team, the team is considered to be a blue team and only a vulnerable infrastructure will be deployed.

The example in Figure 7.4 below has one red team, RT-TeamX; and two teams that have a mixture of both defensive and offensive teams called Team A and Team B which will be engaged in competitive cyber security competitions amongst themselves.

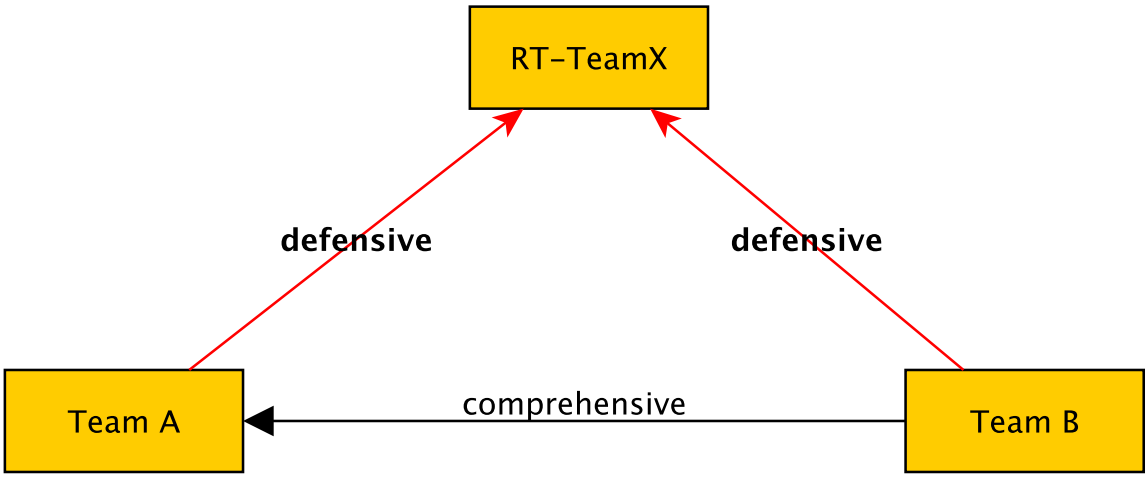

Figure 7.4: Proposed CDX Policy Intention

Table 7.2 below details the *management* options and the number of team mates (*members*) for the various teams in the example of Figure 7.4 above.

Table 7.2: Proposed CDX Policy Intention Details

Team	Management	Members
RT-TeamX	False	1
Team A	True	1
Team B	True	1

The edges between teams in the CDX intention graph is used to specify the competition approach between them. During the design of CDX intention policies, the edges can be labeled with any of the competition approaches (discussed in 7.1.1 above) such as: comprehensive, offensive or defensive. The labels on the edges goes a long way in determining the access rules between the attack servers and vulnerable servers of the various participating teams. Edges labeled comprehensive indicate a relationship whereby the attack servers of the teams connected by the edge can access each others' respective vulnerable servers. Edges labeled offensive indicate a relationship whereby only the attack servers of the source team connected by the edge can access the destination team's vulnerable servers. The destination team in this relationship is in defence formation and only has access to it's own vulnerable servers. Edges labeled defensive indicate a relationship whereby only the attack servers of the destination team connected by the edge can access the source team's vulnerable servers. The source team in this relationship is in defence formation and only has access it's own vulnerable servers. It should be noted that any edge that does not have any of these three labels will not be adequately processed by NePAS during low level compilation of the CDX competition

infrastructure. Also edges in a cyber security competition policy intention do not require any edge options to abstract details and hence none are provided.

As indicated by the defensive label in Figure 7.4, Team A and Team B are expected to defend their infrastructure from attacks being launched by RT-TeamX during the competition. During the competition, the comprehensive label on the edge between Team A and Team B implies they will be tasked with both fixing their vulnerable infrastructure from other team while trying to exploit the vulnerabilities in the opposing team's infrastructure except for the red team (RT-TeamX) that does not have any vulnerable infrastructure.

7.3 CDX Network Layout Abstraction

This section details how we implemented the network layout abstraction for CDX competitions in NePAS. The ellipse nodes in this network layout abstraction is used to represent routers; rectangular nodes are used to represent user computing devices; 6-pointed star are used to represent firewalls; and trapeziod nodes are used to represent backend network servers. It should however be emphasised that all these devices need to specified using the custom property, ***dtype***, explained Chapter 4.2.1 before our system will accept and know what type of network device it will be dealing with during the development of any cyber security competition infrastructure. The edges (both directional and non-directional) used in the graph are used to represent the logical links between the two interconnected nodes (or network devices) in any experiment.

The entire network layout graph represents a single participating team infrastructure for the competition. This is due to the fact that the network layout graph will be replicated across all participating teams. This is so as to make sure all the participating teams have exactly the same network devices and a level playing field. It is imperative that all network layout abstraction graphs of a CDX competition have a central firewall represented as a node labelled *cfw*. The central firewall node is designed to indicate how a team infrastructure will be connected to other sections of the competition network once deployed. This is because the central firewall node is where the attack servers and various vulnerable servers of the various participating team's infrastructure are connected.

The network layout graph in a CDX competition can have both BGP and firewall policies that will determine network connectivity and access rules for the devices within the proposed network. Hence all the network layout devices specified here can take all the custom property options used in specifying firewall policies in exactly the same way as the ones discussed in 6.2 above. The routers in the network layout graph of a CDX competition can also have various BGP business relationships and hence can take the *bgppol* option similar to the ones discussed in 5.2 above.

Figure 7.5 below depicts the cyber security competition infrastructure that has been designed for the high-level competition policy proposed in Figure 7.4. At the core of the

proposed competition infrastructure is a central firewall, cfw, that will be used to connect the various attack servers, various team vulnerable infrastructure and management server. The devices within the proposed network layout infrastructure except the central firewall will be replicated for the participating teams A and B during low level NePAS compilation phase.

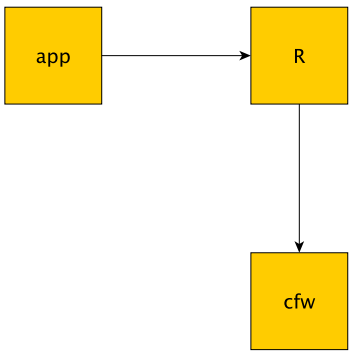

Figure 7.5: Proposed CDX Team Infrastructure

7.4 Network Anomaly Resolution

This section outlines the network anomaly resolution that is carried out to ensure the CDX infrastructure deployed is conflict free. This section can be split into two groups:

- The central firewall anomaly resolution - this section has to do with the access rules between the management server, attack servers and vulnerable servers. The anomaly resolution mechanism to be conducted here will be based on algorithms developed in 6.4 above. This will ensure all low level firewall rules generated for the management server to vulnerable servers and attack servers to vulnerable servers are conflict free. This will also ensure that these firewall rules are generated and deployed as dictated by the policy intention of the cyber security competition.

- The network layout anomaly resolution - this has to do with the network anomaly resolution of the CDX network layout of the vulnerable infrastructure of individual teams. The algorithms that will be employed in this section will ensure conflict free BGP and firewall rule configurations are generated during the deployment of the cyber security competition infrastructure. The algorithms used to achieve this are the same ones discussed in 5.4 for BGP and 6.4 for firewall rule configurations.

7.5 CDX Compilation

This section outlines the details of how NePAS generates the low level configuration commands of a competition infrastructure using the policy intention and network layout graphs designed by white team members. NePAS processes the policy intention graph and creates a specialist list of realms that have CDX relationships when dealing with a policy intention graph abstracting multiple network policies. This is achieved by looking at realms connected by edges with labels comprehensive, defensive or offensive. Next a new graph is created that will be used to create the specialised servers and replicate the infrastructure for the various teams participating in the competition. All the following procedures are done using a graph created for this reason. In order to fully explain the processes involved, we have divided this phase into three sections, these are: creation of specialised servers; replicating the team infrastructure; and infrastructure access rules for the various servers within the infrastructure.

7.5.1 Creation of Specialised Servers

1. Management Server

   A node called **MgtServer** which is based on an image of a server containing all the management services needed for the competition as discussed in 1 above will created be for the infrastructure. As only routers are designed to be connected to firewalls, a router will be created called **MgtRouter** that will be used to connect the management server to the central firewall which in turn gives it access to the entire competition infrastructure.

2. Attack Servers

   Once the management server, management router and central firewall have been created, the next step is the creation of attack servers that will be used by members of various teams during the competition. The attack servers are created using a base image that has a number of specialised tools for attacking the vulnerable systems of the opponent's network or fixing the vulnerabilities within a team's infrastructure. These attack servers will be created and labeled using the specialised list of

teams/realms created above and the node option members determines the number of servers to be created. The servers created here will be be connected to the router called *Attack-Router* so as to link them to the competition infrastructure via the central firewall. The image currently used by NePAS to deploy the attack router used here has support for only fourteen (14) devices to be connected to it during any experiment. This means multiple routers will be created so as to connect the attack servers to the competition infrastructure. As explained in 7.2 above, blue team realms are those that have a value of 0 for *members* option and hence no attack servers will be created for such a team at this stage of the compilation process.

7.5.2 Replicate Team Infrastructure

After NePAS has created the specialised attack servers, it proceeds to make a duplicate copy of the entire network layout graph which includes the nodes, edges and their various attributes for the various teams as indicated in the policy intention graph. This is done because the entire premise of cyber security competitions is to have a level playing field and hence the competition infrastructure must be identical for all the participating teams. The node labels of the new duplicates will have their respective team names prepended to the original node labels. The only network layout node that will not be duplicated from the network layout graph is the central firewall.

NePAS moves on to deep copy the new graph created with the duplicate infrastructure, management server, central firewall, attack router(s) and attack servers back to the original graph. Next the IP addresses of the various interfaces of respective devices are then allocated by NePAS according to the option selected in the configuration file which can be either random or sequential. As mentioned earlier in 7.2 above, any realm that has **RT-** prepended before its name is considered to be a red team and hence no vulnerable infrastructure will be deployed for it at this point.

Figure 7.6 below shows the final replicated infrastructure of the proposed competition introduced in Figure 7.4 and Figure 7.5 above. As can be seen, there is only one attack router labeled *Attack-Router* that has been created in order to link the three (3) participants based on the *member* option from Table 7.2 and the policy intention of the competition. All the teams have an exact infrastructure replica of Figure 7.5 as discussed above. The management router is used to link the management server to the central firewall and entire competition infrastructure as indicated in the diagram.

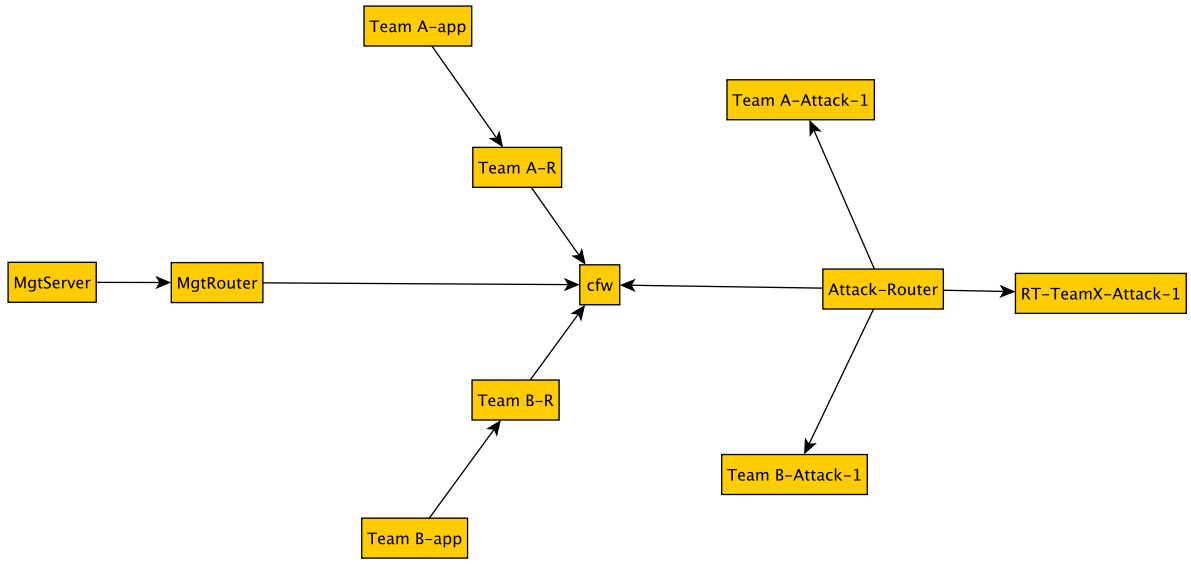

Figure 7.6: CDX Infrastructure with Replicated Teams

7.5.3 Infrastructure Access Rules

Once the IP addresses have been allocated, configuring access rules for the competition infrastructure will be the next thing to be carried out. There are three categories of access rules that have been implemented for the central firewall by NePAS so as to ensure proper security access control for any proposed competition. These categories of access rules are as follows:

1. Management Server Rules

   The first group of access rules has to do with the management server having access to the various team infrastructure during the competition. There are two different sets of firewall rules that have been implemented for the management server functionality. These sets of firewall rules are as follows:

   (a) The firewall rules contained in this section are used to allow the management server, MgtServer, to communicate with all the vulnerable servers of the various participating teams during the competition. The firewall rules that will be configured on the central firewall's rule base for this set of access rules have an access-list name called **nepas-mgt-out**. These set if firewall rules will be applied on the outside interface of the central firewall device that is facing the management server in the infrastructure. These firewall rules will enable the management server to monitor and check flags of teams for scoring during any competition.

(b) The firewall rules contained in this section are used to allow the vulnerable servers of various participating teams to communicate with the management server, MgtServer during the competition. This firewall rule is only configured for participating teams which have their *management* node option value set to "TRUE" in the competition policy intention. A deny firewall rule will be configured if the *management* node option value is set to "FALSE" in the competition policy intention. The fact that red team's do not have any vulnerable infrastructure means this option will be overlooked even if it is specified. The firewall rules that will be configured on the central firewall's rule base for this set of access rules have an access-list name called **nepas-mgt-in**. These set if firewall rules will be applied on the inside interface of the central firewall device that is facing the management server.

The two sets of management rules configured in the central firewall for the replicated competition infrastructure in Figure 7.6 above is shown in Listing 7.1 below.

Listing 7.1: Excerpt of Management Access Rules for Proposed CDX

```
'access-list nepas-mgt-out extended permit ip host
   20.0.0.1 any'
'access-list nepas-mgt-in extended permit ip host
   20.0.1.1 host 20.0.0.1'
'access-list nepas-mgt-in extended permit ip host
   20.0.2.1 host 20.0.0.1'
```

2. Vulnerable Server Rules

The second category of central firewall rules allow all attack servers of a particular team to access their respective vulnerable infrastructure during the competition. An allow rule is configured in the central firewall's rule base so as to enable team members have access to and fix their vulnerable servers accordingly. The firewall rules that will be configured on the central firewall's rule base for this set of access rules have an access-list name called **attack-own-out**. These set if firewall rules will be applied on the outside interface of the central firewall device that is facing the attack servers in the competition infrastructure.

The access rules that have been configured in the central firewall to allow the two teams in the proposed CDX in Figure 7.6 above is shown in Listing 7.2 below.

Listing 7.2: Excerpt of Team-to-Vulnerable Access Rules for Proposed CDX

```
'access-list attack-own-out extended permit ip host
   20.0.8.2 host 20.0.1.1'
'access-list attack-own-out extended permit ip host
   20.0.6.2 host 20.0.2.1'
```

3. Attack Server Rules

   The last category of access rules implemented for cyber security competitions in NePAS determines the access rules between the attack servers of a team and its opponents. This category of access rules is implemented using the relationship between any two teams as indicated in the CDX policy intention graph. The access rules are therefore categorised according to the competition approach as follows:

   (a) A comprehensive relationship between two realms in the intention graph permits access to the vulnerable infrastructure of both teams by their respective attack servers. This translates to an allow rule between the attack servers of both teams to their opponent's vulnerable servers in the central firewall's rule base of the competition infrastructure.

   (b) An offensive relationship between two realms in the intention graph allows the source team attack servers to access the vulnerable infrastructure of the destination team while the destination team attack servers are denied access to the vulnerable infrastructure of the source team.

   (c) A defensive relationship relationship between two realms in the intention graph translates into an allow rule in the central firewall's rule base so as to enable the destination team attack servers to access the vulnerable infrastructure of the source team while the source team attack servers are denied access to the vulnerable infrastructure of the destination team.

   The firewall rules that will be configured on the central firewall's rule base for this set of access rules have an access-list name called **attack-opp-out**. These set of firewall rules will be applied on the outside interface of the central firewall device that is facing the attack servers in the competition infrastructure. The access rules that have been configured in the central firewall to allow the two teams and the red team attack server to be able to access and exploit the opponent's vulnerable servers in the proposed CDX in Figure 7.6 above is shown in Listing 7.3 below.

Listing 7.3: Excerpt of Team-to-Opponent Access Rules for Proposed CDX

```
'access-list attack-opp-out extended permit ip host
   20.0.8.2 host 20.0.2.1'
'access-list attack-opp-out extended permit ip host
   20.0.6.2 host 20.0.1.1'
'access-list attack-opp-out extended permit ip host
   20.0.7.2 host 20.0.1.1'
'access-list attack-opp-out extended permit ip host
   20.0.7.2 host 20.0.2.1'
```

It should be noted that whenever NePAS is dealing with a cyber security competition with firewall device(s) within the individual team infrastructure, these rules will be

adequately added into all firewall device(s) as the policy intention dictates. Therefore when two teams are involved in a comprehensive approach competition, access rules will be configured into the various firewall device(s) in the infrastructure to allow the respective attack servers of the various teams have access to the vulnerable servers of an opponent in order to achieve set objectives.

7.6 Closing Remarks

This chapter of the thesis gives a thorough insight into how cyber security competitions are organised and our proposed abstractions. An example is showcased to analyse how NePAS is used to support our proposed abstractions and evaluate their limitations. The next chapter will be used to critically analyse our high level abstractions using complex experiments. The chapter concludes by evaluating the limitations of the developed system's handling of our proposed abstractions.

Chapter 8

Critical Analysis

8.1 Experimentation

This section attempts to provide verification of how our abstraction models can be used to deploy network experiments from sets of high-level policy intention through to low level device configurations for each of the network policies discussed in the previous three chapters. There are three experiments that will be conducted in this section and each experiment will be focusing on a particular network policy that has been discussed in the previous three chapters. The experiments will have a couple of policy anomalies within the proposed abstraction models so as to see how our system, NePAS, will resolve them before low level device configurations are generated for deployment.

The first experiment to be conducted details how our abstraction models can be used to deploy a multi-tiered ISP network running BGP between various autonomous systems from a high-level inter-domain routing policy intention. The second experiment to be conducted details how our abstraction models can be used to implement a distributed firewall network for a hypothetical financial service organisation from a high-level access control policy document. The last experiment to be conducted details how our abstraction models can be used in designing and running a multi-approach cyber security competition. Each of these experiments will be discussed using four segments as follows:

1. Experiment rationale - this segment is used to briefly explain the reason for conducting the experiment and its policy requirement.

2. Policy intention - this segment is used to explain how the policy requirement from the above segment is translated into a NePAS policy intention graph. Policy anomalies contained in these abstraction models will be highlighted in this segment.

3. Network layout - this segment is used to explain the second phase of our abstraction model or the actual network layout that will be used during the proposed experi-

ment. Policy anomalies contained in these abstraction models will be highlighted in this segment.

4. Experiment overview - the two segments, policy intention and network layout, are the basis for the proposed experimental setup. This segment will be used to explain the relationships between the nodes during the experiment.

5. Expected outcome - this segment is used to analyse the expected outcome (or expected result) of the experiments conducted using the system, NePAS, developed from our abstraction models. The segment will also be used to explain how NePAS handles the various policy anomalies that have been introduced in our abstraction models in the various experiments. The segment ends with an analysis of the strengths and weaknesses of our abstraction models and our system, NePAS.

8.1.1 BGP Routing Experiment

The experiment conducted in this section hopes to show the effectiveness of our inter domain routing abstraction models in deploying networks from high-level BGP business relationships. The first part of this section is used to discuss the rationale for conducting this experiment and detail the routing policies that have been prepared. The BGP business relationships between the autonomous systems that will be used to model a mini-Internet experiment will be discussed in-depth. The second part of this section will be used to discuss the high-level policy intention of the experiment based on the routing policies described in the first section. The business relationships between the various autonomous systems and how they are implemented using NePAS will be discussed in-depth in the second section. The third section details the network layout of the mini-Internet experiment that will be deployed in order to show the effectiveness of NePAS in mapping high-level policy intention of a BGP business relationship on to a proposed network layout. The last section of the experiment will be detailing the expected outcome of the experiment conducted. This part will detail how our system, NePAS, will handle the various anomalies that have been injected and discussed in the previous section. This part concludes with an examination of some the strengths and weaknesses of our inter domain routing abstraction models.

1. **Experiment Rationale**

   The purpose for conducting the hypothetical mini-Internet experiment in this section section is to showcase how our BGP abstraction models can be used to deploy inter domain routing networks. This section of the BGP experiment details the routing policies of the various autonomous systems prepared by a group of service providers called the Internet Council. The Internet Council will be tasked

with deploying the mini-Internet experiment using NePAS based on policies within this document and network layout design of the various service provider networks. There are a total of six autonomous systems that have been identified: Tata, Sprint, FiberRing, TDC, Local ISP (LISP) and Regional ISP (RISP). There are two sets of: tier-1, tier-2 and tier-3 AS networks in this experiment. Tata and Sprint are tier-1 AS networks, they have recently signed an agreement to freely share routes between their networks. The two tier-1 networks, Tata and Sprint, will be providing transit services to FiberRing and TDC respectively for an undisclosed amount. FiberRing realised they need to improve their service quality in order to keep a growing number of disgruntled customers from leaving and hence have signed a transit agreement with Sprint for an undisclosed amount as a way of load balancing customer traffic. The management of FiberRing and TDC have signed two contracts, the first contract is for an IPv4 agreement that states the two companies will freely share routes between their networks and the second contract is for TDC to provide FiberRing IPv6 traffic transit services for an undisclosed amount. The two tier-2 AS networks, FiberRing and TDC, will be providing transit services to LISP and RISP networks respectively for an agreed price that will be announced at a joint press conference. The management of Regional ISP in their expansion plans supported by their shareholders will be buying Local ISP for an industry record amount and will be looking to merge the two networks over a three year period.

2. **Policy Intention**

   This section of the BGP experiment details how the routing policy described in the section above will be converted into a BGP policy intention that will fit our abstraction model. The NePAS BGP policy intention for the experiment will be composed of six realms that will be used to represent the various ISP networks. The ASes for the proposed experiment are as follows: Tata, Sprint, FiberRing, TDC, LISP (representing Local ISP) and RISP (representing Regional ISP). Figure 8.1 below shows the high-level business relationship between the ASes in the proposed experiment. As indicated in the diagram, Tata and Sprint are tier-1 AS networks with AS numbers 6435 and 1239 respectively. Typical with tier-1 ASes and as discussed in the routing policy section, these networks have a peering business relationship between them. FiberRing and TDC are tier-2 AS networks with AS numbers 38930 and 3292 respectively. These two networks have an IP version dependent hybrid relationships between themselves with a peering relationship on IPv4 and a transit (provider to customer) relationship on IPv6. These two ASes are also customers of Tata and Sprint respectively. FiberRing is a multi-homed AS and has another transit relationship with Sprint for load

balancing and redundancy. This is a business decision taken by the management of FiberRing in order to provide their customers with superior service. LISP has an AS number 65432 while RISP will depend on NePAS to automatically generate a public AS number for it during the compilation process. The use of a private AS number as stated earlier is a BGP anomaly and has been intentionally injected into our abstraction model in order to see how our system, NePAS, will resolve it. These two ASes are tier-3 ASes and have a sibling business relationship due to the fact that LISP was just acquired by RISP's parent company. Both RISP and LISP have a transit business relationship with TDC and FiberRing respectively.

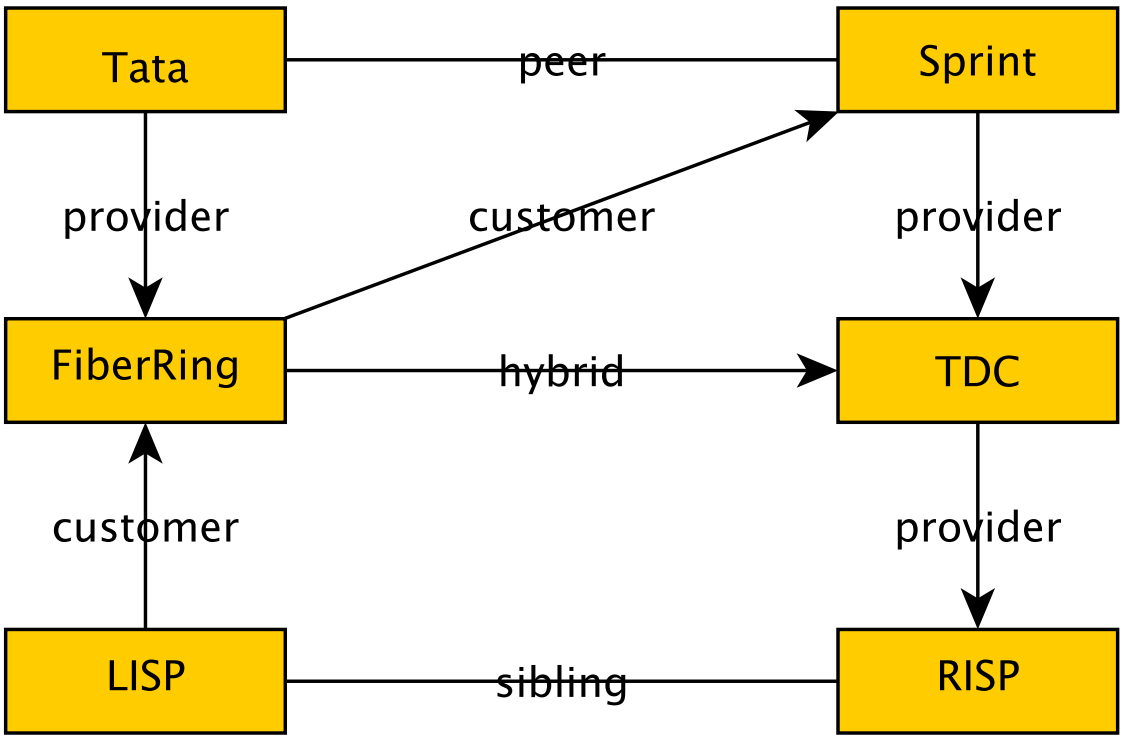

Figure 8.1: Experiment 1 Policy Intention

3. **Network Layout**

   This section of the BGP experiment discusses the network layout of the mini-Internet being developed by the Internet Council based on the network requirements of the various ISPs. The network layout is also where the high level BGP policy intention of the proposed experiment will be mapped on routers. Figure 8.2 below shows the network layout for the proposed experiment.

   As indicated in the diagram above, there are three sets of network devices - servers, routers and firewalls. The network devices are grouped according to the AS network they belong to within the proposed mini-Internet experiment. The

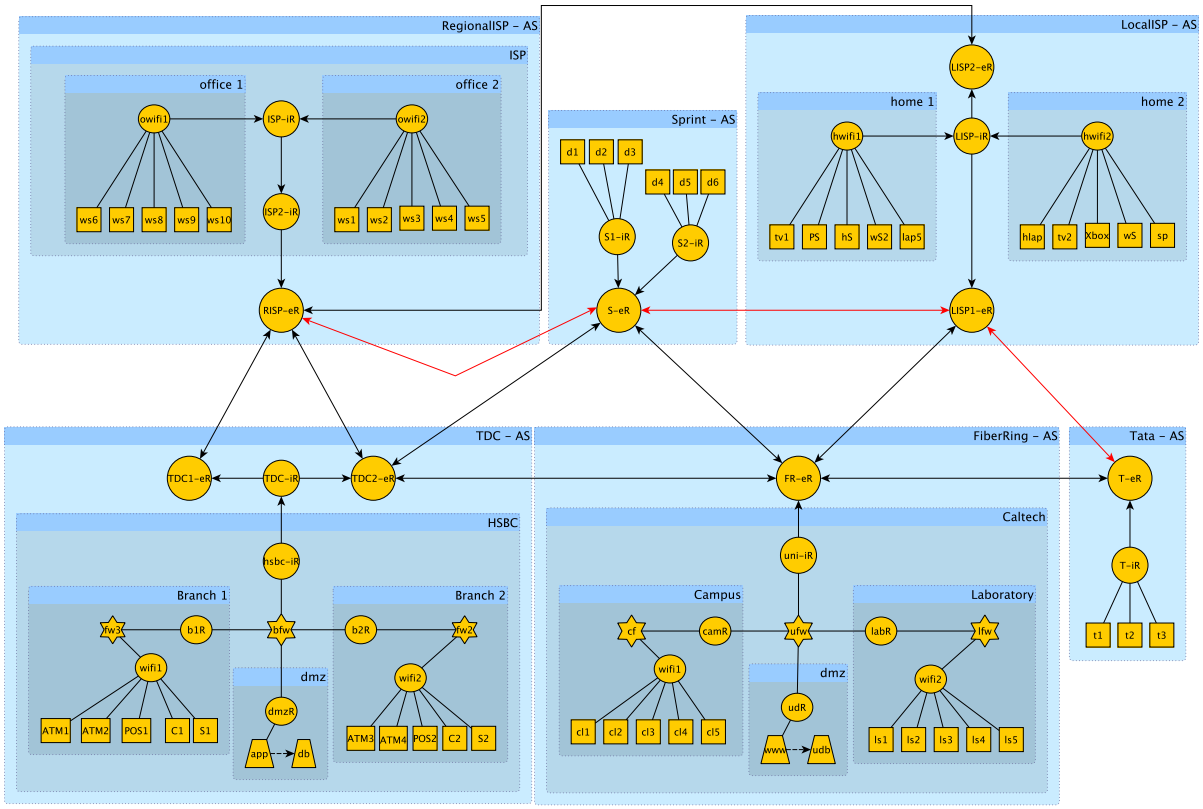

Figure 8.2: Experiment 1 Network Layout

members of a group will have the same AS number and the edge routers will be configured with low-level configurations of the business relationship described in the previous segment with their neighbours. Routers with 'iR' appended to their labels are to be used as internal routers. These routers have directed links terminating at the edge routers of their respective AS networks. Router with 'eR' appended to their labels are used as external routers. They will be configured with external BGP business relationships as dictated in the policy intention discussed above. The table 8.1 below shows a listing of additional routes that will be configured on external routers using the *network* node option.

Router	***network***
RISP-eR	50.0.0.0, 60.0.0.0
S-eR	20.0.0.0, 15.0.0.0
LISP1-eR	30.0.0.0, 25.0.0.0
FR-eR	15.0.0.0, 40.0.0.0
T-eR	50.0.0.0, 55.0.0.0

Table 8.1: Additional Routes on Routers

There are two instances of multiple origin autonomous system (MOAS) from the

table above. Routers RISP-eR and T-eR have been designed to advertise network 50.0.0.0 which is a BGP anomaly. Also, routers S-eR and FR-eR have been designed with MOAS anomaly so as to see how our system, NePAS, will handle it.

4. **Experiment Overview**

   There are a total of six service providers in the experiment with eleven BGP routing connections in the experiment. The following is a general overview of both the BGP policy intention and network layout setup of the experiment.

   From Figure 8.1 above, the nodes such as Tata, Sprint, TDC and others are used to represent a group of routers that belong to a particular service providers or identical inter-domain routing policies. The labels on the nodes will be used in the network layout to determine what service provides or inter-domain routing realms a network router belongs to during the experiment. For example, a router that has an inter-domain routing realm of LISP belongs to the LISP service provider during the experiment. The edges in the Figure 8.1 above are used to indicate the relationship between the two interconnected realms. The labels such as provider, sibling, peer and others on the edges states the exact inter-domain routing relationship between the two realms connected by the edge in the experiment. For example, the edge with label ***provider*** between nodes Tata and FiberRing is used to represent a transit relationship between the two nodes in question.

   From Figure 8.2 in the graph, ellipse nodes such as RISP-eR, wifi1 and others represent routers; rectangular nodes such as ATM1, C1 and others are used to represent user computing devices; 6-pointed star such as bfw, fw3 and others are used to represent firewalls; and trapeziod nodes such as www, app and others are used to represent backend network servers. It should however be emphasised that these devices need to specified using the custom property, ***dtype***, explained Chapter 4.2.1 before our system will accept and know what type of network device it will be dealing with during execution. Pointed edges are used to represent both internal and external BGP connections in the graph. It should be noted that the edges in the network layout abstraction does not use any labels and any labels added will be ignored by our system during the execution. Three of these pointed edges or connections marked with red links in 8.2 above have no predefined BGP relationships. The remaining eight connections marked with black links have routing relationships defined in the policy intention graph of the experiment.

   The following section discusses the connections in the experiment both those with predefined routing relationships and the connections that have no routing relationships defined in the policy intention graph.

   - The nodes, RISP-eR and LISP2-eR, from Figure 8.2 belong to realms RISP

and LISP from Figure 8.1 above. The realms, RISP and LISP, have a sibling relationship hence is expected that they share all routes in the experiment.

- The nodes, TDC2-eR and FR-eR, from Figure 8.2 belong to realms TDC and FiberRing from Figure 8.1 above. The realms, TDC and FiberRing, have a hybrid relationship which includes a peering relationship on IPv4 and a transit relationship on IPv6.
- The nodes, TDC2-eR and S-eR, from Figure 8.2 belong to realms TDC and Sprint from Figure 8.1 above. The realms, TDC and Sprint, have a transit relationship between them. Sprint has a provider relationship to TDC while TDC is a customer of Sprint. Therefore, it is expected that Sprint will advertise all its routes to TDC to enable it provide quality service to TDC. However, TDC is only expected advertise its own routes and the routes of its customers but not its peer or other provider routes to Sprint.
- The nodes, FR-eR and T-eR , from Figure 8.2 belong to realms FiberRing and Tata from Figure8.1 above. The realms, FiberRing and Tata, have a transit relationship between them. Tata has a provider relationship to FiberRing while FiberRing is a customer of Tata. Therefore, it is expected that Tata will advertise all its routes to FiberRing to enable it provide quality service and ensure it is globally reachable. However, FiberRing is only expected to advertise its own routes and routes of its customers but not its peer or other provider routes to Tata.
- The nodes, RISP-eR and TDC1-eR, from Figure 8.2 belong to realms TDC and RISP from Figure8.1 above. The realms, TDC and RISP, have a transit relationship between them. TDC has a provider relationship to RISP while RISP is a customer of TDC. Therefore, it is expected that TDC will advertise all its routes to RISP to enable it be globally reachable. However, RISP is only expected to advertise its own routes and routes of its customers but not its peers or other provider routes to TDC.
- The nodes, S-eR and FR-eR, from Figure 8.2 belong to realms Sprint and FiberRing from Figure8.1 above. The realms, Spring and FiberRing, have a transit relationship between them. FiberRing is a customer of Sprint while Sprint is a provider of FiberRing. Therefore, it is expected that Sprint will will advertise all its routes to FiberRing to enable it be globally reachable. However, FiberRing is only expected to advertise its own routes and routes of its customers but not its peers or other its other provider routes to Sprint.
- The nodes, FR-eR and LISP1-eR, from Figure 8.2 belong to realms FiberRing and LISP from Figure8.1 above. The realms, FiberRing and LISP, have a transit relationship between them. FiberRing has a provider to LISP while

LISP is a customer of FiberRing. Therefore, it is expected that FiberRing will advertise all its routes to LISP to enable it to be globally reachable. However, LISP is only expected to advertise its own routes and routes of its customers but not its peers or other provider routes to FiberRing.

- The nodes, TDC2-eR and RISP-eR, from Figure 8.2 belong to realms TDC and RISP from Figure8.1 above. The realms, TDC and RISP, have a transit relationship between them. TDC has a provider relationship to RISP while RISP is a customer of TDC. Therefore, it is expected that TDC will advertise all its routes to RISP to enable it be globally reachable. However,RISP is only expected to advertise its own routes and routes of its customers but not its peers or other provider routes to TDC.
- The nodes, LISP1-eR and T-eR, from Figure 8.2 belong to realms LISP and Tata from Figure 8.1 above. The realms, LISP and Tata have no relations in the policy intention graph 8.1 above and is considered an anomaly. The way our system handles such anomalies will be detailed in the Anomaly Resolution segment in the following section of the experiment.
- The nodes, RISP-eR and S-eR, from Figure 8.2 belong to realms RISP and Sprint from Figure 8.1 above. The realms, RISP and Sprint have no relations in the policy intention graph 8.1 above and is considered an anomaly. The way our system handles such anomalies will be detailed in the Anomaly Resolution segment in the following section of the experiment.
- The nodes, S-eR and LISP1-eR, from Figure 8.2 belong to realms Sprint and LISP from Figure 8.1 above. The realms, Sprint and LISP have no relations in the policy intention graph 8.1 above and is considered an anomaly. The way our system handles such anomalies will be detailed in the Anomaly Resolution segment in the following section of the experiment.

5. **Expected Outcomes**

   This section of the experiment will be used to analyse and discuss the expected outcome of the mini-Internet experiment. This section of the experiment will be split into three subsections that deal with - business relationships between the routers; how NePAS handles injected anomalies in the experiment; and the strengths and weaknesses of our abstraction models and system developed for testing the models.

   Business Relationships

   The external BGP link between TDC1-eR and RISP-eR is a backup link and will only be activated when the TDC2-eR and RISP-eR link fails for whatever reason. To configure this link, a lower local preference value was configured on

the BGP session on the link between routers TDC2-eR to RISP-eR. The routers TDC1-eR and TDC2-eR have a provider BGP business relationship with router RISP-eR. The routers have been configured with an export policy to RISP-eR advertising their own routes (composed of IP addresses of network devices within their AS) and routes router TDC2-eR has learned from S-eR (its customer). The router TDC2-eR has a BGP customer and peering business relationships with S-eR and FR-eR respectively. Therefore, TDC2-eR has been configured to advertise its own routes and routes it has learned from RISP-eR (its customer) to both S-eR and FR-eR. It has however not been configured advertise to FR-eR routes it has learned from S-eR and it was likewise not configured to advertise routes it learned from S-eR to FR-eR because it has a customer and peer relationship with both routers respectively. The router RISP-eR has a BGP customer, customer and sibling relationship with TDC1-eR, TDC1-eR and LISP2-eR respectively. The router been configured to advertise its own routes and routes it has learned from TDC AS. It was also configured to advertise all its routes and routes it has learned from LISP2-eR to both routers in TDC AS. The BGP sessions in Figure 8.2 above with red coloured edges indicate links between two ASes with no high level BGP business relationship in the policy intention graph. The external BGP links between RISP-eR to S-eR; LISP1-eR to S-eR and LISP1-eR to T-eR are examples of BGP sessions that have no BGP business relationships. In situations such as these, NePAS will in addition to generating a warning log message, no BGP session has been configured between the two routers.

Anomaly Resolution

The first anomaly that has been highlighted in this experiment is the use of a private AS number by LISP during the deployment of the network. A private AS number when detected in an abstraction model is converted to a public AS number automatically by our system, NePAS. The system when it generates a public that is already in use for a particular experiment, keeps on generating public AS numbers until a unique one is generated.

The second anomaly that has been highlighted in this experiment is potential multiple origin autonomous system (MOAS). There are two cases of MOAS present in the experiment conducted. The first instance of MOAS anomaly in the experiment is between routers RISP-eR and T-eR advertising network 50.0.0.0 respectively. The second instance of MOAS anomaly in the experiment is between routers S-eR and FR-eR advertising network 15.0.0.0 respectively. Our system, NePAS, generates an error log message and does not generate low level configurations for advertising the routes with such an anomaly on all the affected routers. This ensures low level configurations generated for networks being developed using

our abstraction models do not have MOAS anomalies.

Strengths and Weaknesses

The major strength of our abstraction models and NePAS is the ability to ensure final low level configuration generations are free of anomalies such as origin misconfigurations, private AS numbers announcement, multiple origin autonomous system (MOAS) and export misconfigurations. This goes a long way in ensuring network experiments behave in conformity with high level policy specifications. The major weakness of our abstraction model and/or NePAS is its inability to verify route connectivity to networks before such routes are configured on routers during experiments. Our abstraction models and system, NePAS, in their current state do not have any measures built-in to prevent BGP wedgies during network experiments as well due it being a post deployment anomaly which is not part of the research conducted in this thesis.

8.1.2 Firewall Relationships Experiment

This experiment hopes to show the effectiveness of using our firewall abstraction model for deploying a hypothetical financial service provider (bank) network. The security policy that has been prepared by the management for the organisation's ICT unit which responsible for configuration will be discussed in the first part of this section. This security document will be used to guide the policy intention abstraction of the bank's network. It should be noted that the policy document reproduced in this section is focused mainly on the firewall relationships between devices hosted within the network. The NePAS policy intention based on the proposed security policy will be discussed in the second segment of this section. This segment will also be used to discuss the various firewall policy anomalies that are in our firewall abstraction models. The bank's network layout discussed in the third part of the experiment. The bank's network layout is composed of various devices (and/or users) in two different office locations. This section also details some of the features of NePAS firewall rule abstraction. The last section of the experiment will be detailing the expected outcome of the experiment to be conducted. This part will detail how our system, NePAS, will handle the various firewall anomalies that have been injected and discussed in the previous sections of the experiment. This part concludes with an examination of some the strengths and weaknesses of our firewall abstraction models.

1. **Experiment Rationale**

   The rationale for conducting this experiment is to showcase how our firewall abstraction model can be used to deploy a multiple firewall network. This section of the firewall experiment details the bank's proposed network security policy as

prepared by the management composed of a specialised committee comprising security personnel; human resource department staff; IT department staff and some senior managers of the bank. The proposed security policy is expected to be implemented by the bank's IT department. There are seven separate zones of users or devices that have been identified by the committee for the bank's network in this document. These zones are: external de-militarised zone, internal de-militarised zone, trading office, management office, front desk, counter and the general staff office. The various zone outlined will have various devices that share identical security policies. Each of the zones will be briefly discussed in the following section:

- *External De-Militarised Zone* (eDMZ) - this zone of the network is used to host devices that are openly accessible to any user both within the bank network and outside it. The devices within this zone are not allowed to initiate communication with any device both within and outside the bank network for security reasons.
- *Internal De-Militarised Zone* (iDMZ) - this zone is used to host devices that are accessible to users within the bank network only. Therefore, zones permitted to access the resources within this zone are: management office, traders, front desk, counter and general staff office. There are network servers within this zone that backup data from network devices in the external de-militarised zone.
- *Trading Office* - this zone is used to host users that work in the stockbroking unit of the bank. The users within this group are allowed to access the network resources that are hosted in the internal de-militarised zone of the bank network. The users are only allowed to access the network of the Stock Exchange but any other communication both within or outside the bank network is not allowed by this group of users.
- *Management Office* - this zone is used to host devices that the management staff of the bank use. The users in this zone have unfettered access to any device both within the bank network. The devices within this zone can be accessed from any device both within and outside the bank network using virtual private networks such as secure shell (ssh). No one both within the bank network and outside it is allowed to initiate communication with the devices in this zone using any other means for security reasons.
- *Front Desk* - this zone is used by users in the customer care and marketing departments of the bank network. The users within this zone are permitted to communicate with any device both within and outside the network using

web, Skype and WhatsApp. The users in this zone are not allowed to use any other social media applications so as not to distract the staff members from work. The users of this zone are allowed to communicate with devices within the internal de-militarised zone so as serve customers better. No one both within the bank network and outside it is allowed to initiate communication with the devices in this zone for security reasons.

- *Counter* - this zone is used to indicate workstations of the bank that are used to satisfy customer operations of depositing and withdrawing money from their respective accounts. The devices in this zone are only permitted to communicate with a specific application server in the internal de-militarised zone. No one both within the bank network and outside it is allowed to initiate communication with the devices in this zone for security reasons.
- *General Staff Office* - this zone is used for users in the business development, research, advertising and risk management departments of the bank. The users in this zone are allowed unfettered access to any device both within and outside the bank network so as to make their jobs more effective and efficient. The users within this zone collaborate with another bank on a number of projects and hence there is a need to punch a firewall hole in the bank network to allow both groups to communicate with each other seamlessly using any medium they choose.
- The human resources department staff have realised that there are two highly addictive websites (1.2.3.4 and 5.6.7.8) that a lot of the staff members are fund of going and spending a considerable amount of time which impacts the work rate of the staff. This has been highlighted by the committee and the IT department is required to deny all staff members from accessing the websites. The founder of the bank has mandated that his home workstation should be allowed unfettered access to the entire bank network in a memo to the committee.

2. **Policy Intention**

This section of the firewall experiment discusses the NePAS firewall policy intention that has been designed based on the network security policy detailed in the above section. The NePAS firewall policy intention will have six firewall realms based on the possible rule composition and zones from the network security policy adopted. The firewall realms that have been developed for the experiment are: **mgt** is used to abstract firewall rule relations for the management office; **iDMZ** abstract internal de-militarised zone rules; **eDMZ** abstract external de-militarised zone rules; **traders** abstract trading office rules; **fdesk** abstract front desk rules;

general abstract general staff office rules. Figure 8.3 below shows the high level firewall rule relationships between the various realms of the proposed experiment.

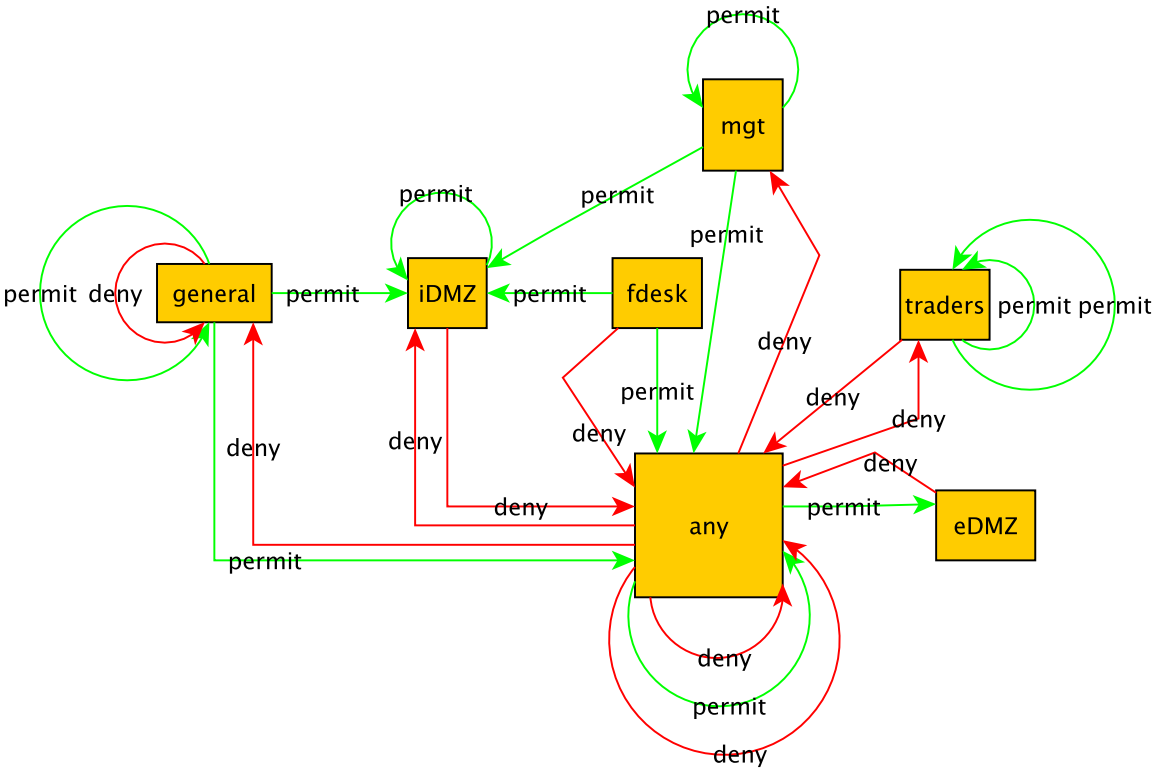

Figure 8.3: Experiment 2 Policy Intention

In Table 8.2 below, all the firewall tuple details for the various rule relationships between the realms that cannot be shown in Figure 8.3 are presented. A total of twenty-four firewall rule relations have been abstracted from the network security policy statement described in the first segment of this experiment. The firewall rule relations encompasses everything that is needed to ensure the network security policy statement of the bank is implemented.

Table 8.2: Firewall Rule Abstractions for Experiment 2

R/No	S Realm	D Realm	Action	Protocol	D Port	S/D Address	S Port
R1	any	any	permit	icmp	NULL	NULL	NULL
R2	any	any	deny	tcp,udp	NULL	d 1.2.3.4, d 5.6.7.8	NULL
R3	any	any	deny	ah, esp gre, 41	NULL	NULL	NULL
R4	any	general	deny	ip	NULL	NULL	NULL
R5	any	iDMZ	deny	ip	NULL	NULL	NULL
R6	any	eDMZ	permit	ip	NULL	NULL	NULL
R7	any	mgt	deny	ip	NULL	NULL	NULL
R8	any	traders	deny	ip	NULL	NULL	NULL
R9	eDMZ	any	deny	ip	NULL	NULL	NULL
R10	iDMZ	any	deny	ip	NULL	NULL	NULL
R11	iDMZ	iDMZ	permit	tcp	www, 8080	NULL	NULL
R12	mgt	any	permit	ip	NULL	NULL	NULL
R13	mgt	iDMZ	permit	ip	NULL	NULL	NULL
R14	mgt	mgt	permit	ip	NULL	s 1.2.3.4, c 5.6.7.8	NULL
R15	fdesk	any	permit	tcp, udp	www, 8080 80, 5222-5230	NULL	NULL
R16	fdesk	iDMZ	permit	ip	NULL	NULL	NULL
R17	fdesk	any	deny	ip	NULL	NULL	NULL
R18	traders	any	deny	ip	NULL	NULL	NULL
R19	general	any	permit	ip	NULL	NULL	NULL
R20	general	iDMZ	permit	ip	NULL	NULL	NULL
R21	general	general	deny	tcp	6881-6999	NULL	6000-7000
R22	general	general	permit	tcp	ftp, ftp-data ssh, sftp	124.0.0.23	NULL
R23	traders	traders	permit	ip	NULL	d 7.8.9.10	NULL
R24	traders	traders	permit	ip	NULL	s 132.122.76.0 255.255.255.0	NULL

The firewall rules, R12 and R18 in the table above are likely to cause inter-firewall correlation anomaly during low level configuration of the experiment. Also, firewall rules, R7 and R8, from the above table are likely to cause inter-firewall redundancy anomaly during low level configuration. These two anomalies have been added to our abstraction models in this experiment so as to see how NePAS

will resolve them during low level configuration.

3. **Network Layout**

   This section of the firewall experiment discusses the network layout that will be used. The network layout is split into five different networks - the network of the bank's head office, the home network of the bank's founder, the network where the datacenter of the bank's resources are hosted and two networks representing the banks' branches. The network layout is where the high level firewall policy intention of the proposed experiment will be implemented. Figure 8.4 below shows the layout of the proposed hypothetical financial network.

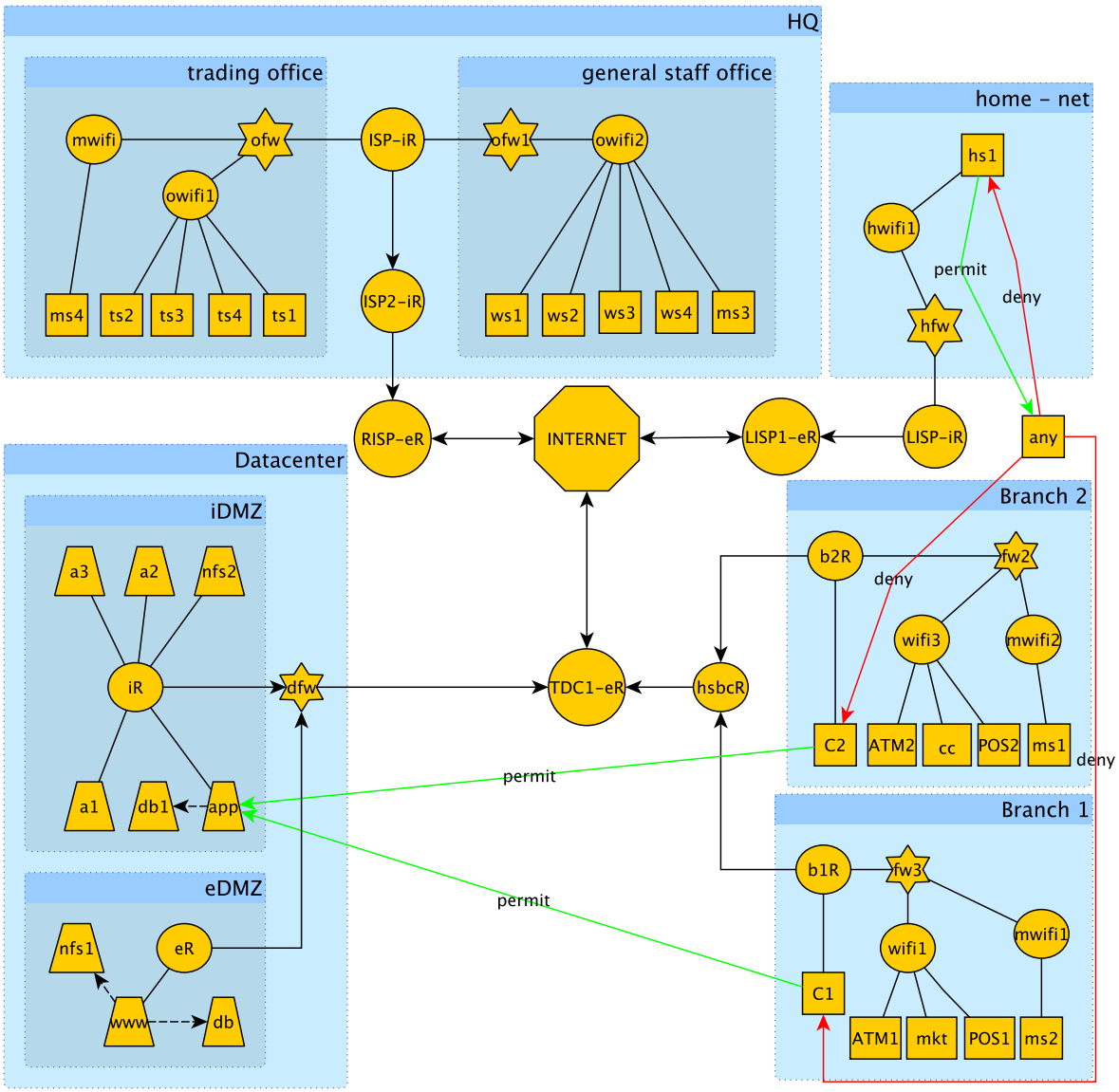

Figure 8.4: Experiment 2 Network Layout

Table 8.3 below shows a break down of the various policy intention realms dis-

cussed above and the firewall-enabled nodes with their respective realms.

Table 8.3: Network Topology Detail for Experiment 2

S/No	Realm	Devices
1	iDMZ	a1, a2, a3, app, db1, nfs2
2	eDMZ	www, db, nfs1,
3	mgt	ms1, ms2, ms3, ms4
4	fdesk	cc, mkt
5	traders	ts1, ts2, ts3, ts4, ms4
6	general	ws1, w2, ws3, ws4
7	-	hs1, C1, C2

As indicated in Figure 8.4 above, there are some additional firewall rules that have been implemented so as to conform with the network security policy detailed in the first section of the experiment. Table 8.4 below shows the detailed firewall tuple options for the six additional firewall rule relations. The six firewall rules abstracted in the network layout of the hypothetical financial institution network are detailed below.

Table 8.4: Network Layout Firewall Rules for Experiment 2

R/No	S Device	D Device	Action	Protocol	D Port	S/D Address	SPort
R1	C1	app	permit	tcp	34567	NULL	45678
R2	C2	app	permit	tcp	43200	NULL	45678
R3	any	app	deny	tcp	43200	NULL	45678
R4	hs1	hs1	permit	tcp	22	d 188.124.125.3	NULL
R5	hs1	any	permit	ip	NULL	NULL	NULL
R6	hs1	hs1	permit	tcp	22	d 188.124.125.3	NULL
R7	ms1	any	deny	icmp	NULL	NULL	NULL
R8	any	hs1	deny	ip	NULL	NULL	NULL
R9	any	C1	deny	ip	NULL	NULL	NULL
R10	any	C2	deny	ip	NULL	NULL	NULL

The first rule R1 is used to permit communication between the machine used on the bank's counter floor C1 on port 45678 and a server app within the iDMZ on port 34567 over TCP protocol as required in the counter section of the policy statement discussed above. The second rule R2 is used to permit communication between the machine used on the bank's counter floor C2 on port 45678 and a server app within iDMZ on port 43200 over TCP protocol. The third rule R3 is

used to permit communication to all the servers and machines within the multiple bank networks and the workstation of the founder in his home network over IP protocol. Rules 4-6 are used to deny communication from any device within the proposed network to devices hs1, C1 and C2 over IP protocol.

The firewall rules, R4 and R6, in the above table will cause intra-firewall redundancy anomaly during low level configuration of the experiment. Firewall rules, R2 and R3, in the above table will cause intra-firewall shadowing anomaly during low level configuration. Finally, the firewall rules, R7 in Table 8.4 and R1 in Table8.2 above will cause intra-firewall correlation during low level configuration.

4. **Experiment Overview**

   There are a total of six firewalls in the network layout of the experiment and five networks comprising Branch 1, Branch 2, home - net, HQ and Datacenter. The following is a general overview of both the firewall policy intention and network layout setup of the experiment.

   From Figure 8.3 above, the nodes such as fdesk, mgt, general and others are used to represent a group of network devices that have identical firewall relationships in the experiment. The labels on the nodes will be used in the network layout to determine the firewall realm a network device belongs to during the experiment. For example, network devices that belong to the eDMZ firewall realm will have firewall relationships abstracted in the policy intention graph with other devices in other realms during the experiment. The edges in Figure 8.4 above are used to indicate the action to be taken by the firewall relationship between the two interconnected realms. The label permit on the edges in the graph are used to abstract firewall relationships between two interconnected realms that allows communication based on certain protocols in the experiment. The label deny on the edges in the graph are used to abstract firewall relationships between two interconnected realms that deny communication base on certain protocols in the experiment. It should be noted that color of the edges in the graph has no effect during the execution of the experiment as our system uses the labels and custom properties that have been defined by the network administrator during the execution of the experiment.

   From Figure 8.4 above, ellipse nodes such as TDC1-eR, owifi1 and others represent routers; rectangular nodes such as ATM1, ts1 and others are used to represent user computing devices; 6-pointed star such as hfw, fw3 and others are used to represent firewalls; trapeziod nodes such as www, app and others are used to represent backend network servers; and octagon shaped nodes labeled "Internet" is used to represent an interconnection to the Internet. It should however be

emphasised that all these devices need to specified using the custom property, ***dtype***, explained Chapter 4.2.1 before our system will accept and know what type of network device it will be dealing with during execution. There are two types of edges in the graph above, these are - edges that represent logical links and edges used to abstract additional firewall rules. The first type of edges or pointed edges and edges that have no labels are used to represent logical links between the two devices in the graph. The second type of edges or the edges with labels are used to abstract firewall relationships in the network layout graph between the two interconnected nodes. The firewall relationships are abstracted in the same manner as the firewall policy intention. The label permit on the edges in the graph are used to abstract firewall relationships between two interconnected nodes that allows communication based on certain protocols in the experiment. The label deny on the edges in the graph are used to abstract firewall relationships between two interconnected nodes that deny communication base on certain protocols in the experiment. It should be noted that color of the edges in the graph has no effect during the execution of the experiment as our system uses the labels and custom properties that have been defined by the network administrator during the execution of the experiment.

5. **Expected Outcome**

   This section of the experiment will be used to analyse and discuss the expected outcome of the hypothetical financial institution network to be deployed. This section of the experiment will be split into three subsections that deal with - analysis of some complex firewall relationships in the experiment; how NePAS handles the various injected anomalies in the various firewall relationship abstraction models in the experiment; the section concludes with the analysis of the strengths and weaknesses of both our abstraction models and system used for testing these models.

   Firewall Relationship

   This section looks at some interesting sets of firewall relationship abstraction models expresses for the financial network. We begin by looking at the firewall devices and network devices with some interesting relationship expressions and properties.

   As indicated in Table 8.3 above, network devices hs1, C1 and C2 do not belong to any firewall realm in the policy intention. Next, firewall rule relations R1 and R2 from Table 8.4 have been configured in firewall ofw to enable C1 and C2 to communicate with app server in iDMZ of the proposed bank network.Two firewall rules, R5 and R6 from Table 8.4, were configured in all the firewall devices

within the proposed network to deny any device from initiating communication with C1 and C2 respectively. The firewall hfw has been configured with firewall rule R1 from Table 8.4 above to enable the workstation in the bank founder's house to communicate with any device within the proposed financial network. A firewall rule that denies any device from communicating with the founder's home workstation is configured on all the firewall devices within the proposed network. All the devices within the proposed network belong to only one firewall intention realm except ms4 which belongs to both mgt and traders. Firewall ofw has been configured with firewall rules of both realms so as to handle the network security policy requirement for ms4 network device.

All the rules in Table 8.2 above can be handled by NePAS except for R11, R14, R22 due to the following reasons. The first of the three rules, R11, was not configured for devices belonging to iDMZ realm because NePAS needed a *dest* value which was omitted in the firewall rule abstraction model. With the R14 rule, the second part of the *dest* value, c 5.6.7.8 will not be configured on the most most upstream firewall devices of devices within the mgt firewall realm because the character c is not what NePAS is expecting in the firewall rule abstraction model. However, the first part, s 1.2.3.4, with the appropriate notation that our system understands will be configured during the experiment. The third rule, R22, was not configured on any device that belong to the general firewall realm because the initial character s or d as required by NePAS was omitted in the firewall rule abstraction model.

Anomaly Resolution

This subsection looks at the various firewall anomalies injected into our abstraction models and how our system handles them during the experiment. There are a total of five intra and inter firewall anomalies that have been injected into the financial network's model. Intra-firewall anomalies will be discussed in the beginning and the inter-firewall anomalies discussed in the latter part.

The first intra-firewall anomaly that will analysed is the redundancy anomaly in R4 and R6 in Table 8.4. This anomaly in the abstraction model is handled by NePAS using its automatic removal of identical firewall rule relations within the financial network during deployment. Therefore in this case, only R4 will be configured in the affected firewall devices during deployment and hence no intra-firewall redundancy anomaly will occur. The second intra-firewall anomaly that will analysed is the shadowing anomaly in R2 and R3 in Table 8.4. This anomaly in the abstraction model is handled by NePAS using its rule ordering mechanism at deployment. As indicated in Chapter 6.4.1, R2 will be placed on top of R3 in the firewall rule table in all affected firewall devices during deployment

and hence no intra-firewall shadowing anomaly will occur. The third intra-firewall anomaly that will analysed is the correlation anomaly in R7 of Table8.4 and R1 in Table 8.2. This anomaly in the abstraction model is currently not resolved by NePAS during deployment. Therefore, both rules will be configured on all affected firewall devices within the financial network during deployment.

The first inter-firewall anomaly that will analysed is the correlation anomaly in R12 and R18. The fact that device ms4 belongs to both mgt and traders firewall intention realms introduced a correlation anomaly due to the presence of rules R12 and R18 from Table 8.2 above in all the firewall devices except hfw of the proposed network. NePAS for now does not support have a way of resolving such anomalies and hence both rules were configured in all the firewall rules except firewall ofw. The second inter-firewall anomaly that will analysed is the redundancy anomaly in Table 8.4 R7 and R8. NePAS however resolves the inter-firewall redundancy anomaly as both R7 and R8 from Table 8.2 specify the same firewall relationship and one is expected to be removed upon deployment on all the firewall devices within the financial network.

Strengths and Weaknesses

The major strength of our abstraction model is in its ability to totally avoid intra-firewall irrelevance anomaly as highlighted in 6.4.1. Our abstraction models also avoid inter-firewall spuriousness and shadowing anomalies during deployment. NePAS draws its major strength in the fact that it can handle any intra-firewall redundancy and shadowing anomalies; and also inter-firewall redundancy in any abstraction model at deployment.

The major weakness of our abstraction model is the lack of support for network address translation (NAT). Also, NePAs not being able to handle intra-firewall correlation anomaly is another major shortcoming especially when designing complex firewall networks. Lastly, while the device central firewall rule generation makes it easier for NePAS to handle intra-firewall shadowing anomaly, it invariably makes populated firewall tables to have too many rules and hence raises the issue of efficiency in rule traversal.

8.1.3 Cyber Security Competition Experiment

The experiment conducted in this section hopes to show the effectiveness of using our cyber security abstraction model deploying a cyber security competition. The experiment conducted in this section will be using a combination of high level policy intentions using all of the three network policies discussed in this thesis - BGP routing, Firewall rule relations and Cyber security competitions. The first part of this section is used to

discuss the rationale for conducting the hypothetical competition experiment taking place including the composition of participants and their respective objectives. The second part of the section will be used to discuss the NePAS policy intention based on the above section. As the proposed competition will encompass routing and firewall intentions within the cyber security competition, this part of the experiment will be split into three subparts so as to fully explain the various policy intentions for the competition and its infrastructure. The third part of this section will be used to detail the network layout of the proposed cyber security competition. This part will also be used to show the routing and firewall mapping of high level policies to network devices. This section concludes with an analysis of the expected outcome of the experiment presented. This will also be used to detail the final rendering of the cyber security competition infrastructure with additional focus on the firewall and routing devices.

1. **Experiment Rationale**

   The rationale for conducting this experiment is to showcase how our cyber security competition abstraction model can be used to deploy a complex competition network. This section of the cyber security competition experiment details the competitions participation and objectives as drawn up by members of the white team.

   There are heightened tension in the Merman region over a decade long dispute about the ownership of River Westeros between the four adjoining countries of Stormland Republic, Westerlands Republic, Dorne Republic and Vale Republic. The hostilities have led the intelligence agencies of the various countries to create a specialised computer emergency response team composed of three members. The teams have stepped up effort to ensure the cyber space of their respective countries are fortified from external attacks from the other countries. The teams in recent times have taken the approach "the best form of defence is to attack" and hence are looking at exploiting vulnerabilities within the other countries in order to destabilise and subdue them.

   A peace mission group, Baratheon, constituted by world leaders has been instructed to setup a replica cyber infrastructure that can be used by the various security teams as a training ground. All the security teams of the various countries are expected to exploit the replica infrastructure and gain cyber currency that can be used to purchase updates for their vulnerable servers. The green team is therefore expected to provide a management server where the various security teams can download updates using their cyber currency to upgrade their vulnerable infrastructure. The green team is also mandated to monitor and keep track of various exploits so as to have evidence of what transpired during this critical period for future arbitration purposes.

As tensions between the four countries flare, a loosely formed group of hackers going by the name "Recusants" have threatened cyber nuclear apocalypse on the the cyber infrastructure of the four countries. The multi-user virtual group has a decentralised membership of five with no known location of operation. The security team of the various countries have to contain the attacks being launched by this nefarious group.

2. **Policy Intention**

This section of the cyber security competition is used to discuss the high level relationship of various network policies. As already stated, in addition to the cyber security competition policy intention, the cyber infrastructure of the various countries has a combination of firewall and routing policies. The various policies (competition, inter domain routing and firewall) are taken one at a time and discussed starting with the cyber security competition intention.

Figure 8.5 below shows the high level competition relationship between the participating teams involved in the competition. The black edges are used for representing competition policy relationships between the various participating teams.

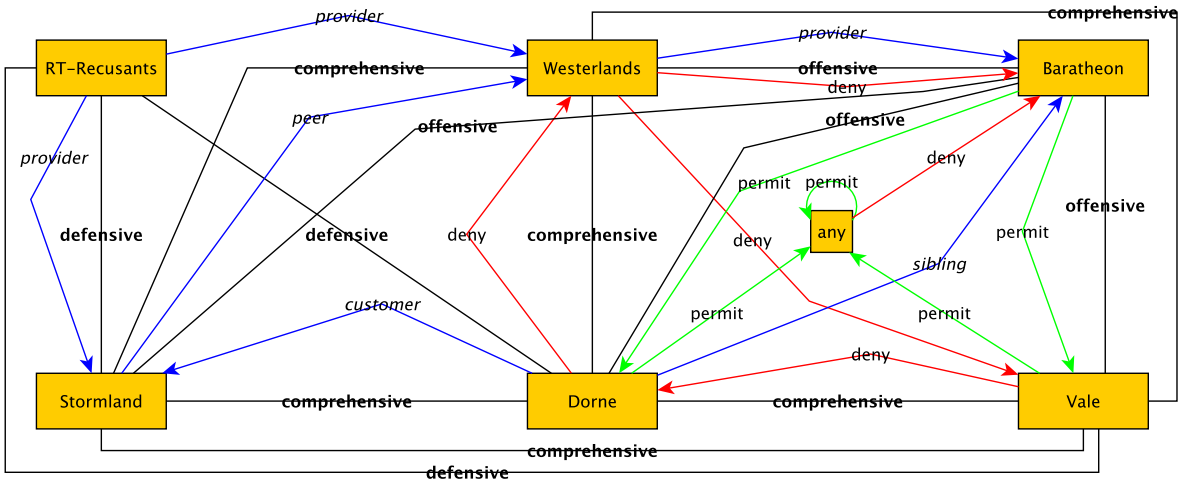

Figure 8.5: Experiment 3 Policy Intention

Competition Policy Intention - this subpart is used to discuss the high level cyber security competition policy intention of the relationships between the various combatant teams and other policy details. The specialised computer emergency response teams of the various countries are labelled: Stormland, Westerlands, Dorne and Vale respectively. These teams have a comprehensive competition approach between themselves as they are expected to exploit their opponents' vulnerable infrastructures while they fix the vulnerabilities in their infrastructure

so as to stop opponents from exploiting them. The blue team called Baratheon will be deployed as a replica cyber infrastructure for the peace mission group while a red team will be labelled RT-Recusants. The four specialised teams have an offensive competition approach between them and team Baratheon and hence are expected to exploit the vulnerabilities in that network. Lastly, the various teams have a defensive competition approach with RT-Recusants and are expected to stop them from exploiting their respective vulnerable infrastructures.

Table 8.5 below provides a detailed enumeration of all the competition's policy details such as number of participants and whether or not the various infrastructure of the teams are allowed to access the management server.

Table 8.5: Policy Details for Experiment 3

Team	Members	Managment	AS Numbers
RT-Recusants	5	False	123
Stormland	3	True	456
Westerlands	3	True	789
Dorne	3	True	135
Vale	3	True	-
Baratheon	0	False	246

BGP Policy Intention - this subpart is used to discuss the high level BGP routing policy intention between the AS networks within the individual competition infrastructure. The realms used for representing the cyber security competition teams will be used to represent AS networks. The ASes used for the proposed competition's routing are: RT-Recusants, Stormland, Westerlands, Dorne and Baratheon. The relationships between the various ASes is as follows RT-Recusants provide transit services to Westerlands and Stormland. Westerlands provides transit service to Baratheon while it has a peering relationship with Stormland. Dorne is a customer of Stormland while it has a sibling relationship with Baratheon. It should be noted that the realm Vale is not part of the BGP business relationship abstraction for the competition. The blue edges in Figure 8.5 above are used for representing BGP business relationships between the various ASes. Table 8.5 above shows the AS numbers that will be used for configuring the routing policies of the proposed competition routing infrastructure.

Firewall Policy Intention - this subpart is used to discuss the high level firewall policy intention for the proposed competition infrastructure. All the realms used in Figure 8.5 above will be used for abstracting high level firewall rule relations. The firewall rule relations will be applied on firewall-enabled devices within the cyber

space of the various countries. High level firewall rule relations will be abstracted using red and green coloured edges between the realms for deny and permit rules respectively. Table 8.6 below shows the full tuple details of the firewall rule relations abstracted. A total of ten firewall rule relations have been abstracted for the various firewall-enabled devices for the cyber space of the countries.

Table 8.6: Realm Rule Relations for Experiment 3

S/N	S Realm	D Realm	Action	Protocol	D Port	S/D Address	S Port
R1	any	any	permit	ip	NULL	d 10.0.0.1	NULL
R2	any	Baratheon	deny	tcp,udp	NULL	NULL	NULL
R3	Baratheon	Vale	permit	ip	NULL	NULL	NULL
R4	Dorne	any	permit	ip	NULL	NULL	NULL
R5	Westerlands	Baratheon	deny	ip	NULL	NULL	NULL
R6	Dorne	Westerlands	deny	ip	NULL	NULL	NULL
R7	Westerlands	Vale	deny	ip	NULL	NULL	NULL
R8	Vale	Dorne	deny	ip	NULL	NULL	NULL
R9	Vale	any	permit	tcp, udp	NULL	NULL	NULL
R10	Baratheon	Dorne	permit	ip	NULL	NULL	NULL

3. **Network Layout**

This section of the cyber security competition is used to discuss the network layout of the various hostile countries. There are two sets of high level policies that will be mapped to the network devices within their network layout in this section. The first set of mappings deal with routing policies that will be configured on the routers within the respective cyber spaces in the countries. Table 8.7 below shows the mapping of high-level routing policies on the various routers.

Table 8.7: Routing Policy Mappings for Experiment 3

S/No	Realm	Devices
1	RT-Recusants	R
2	Stormland	R1
3	Westerlands	R2
4	Dorne	R3
5	Baratheon	R4

Figure 8.6 below shows the proposed cyber space of the various countries that will be used for the competition.

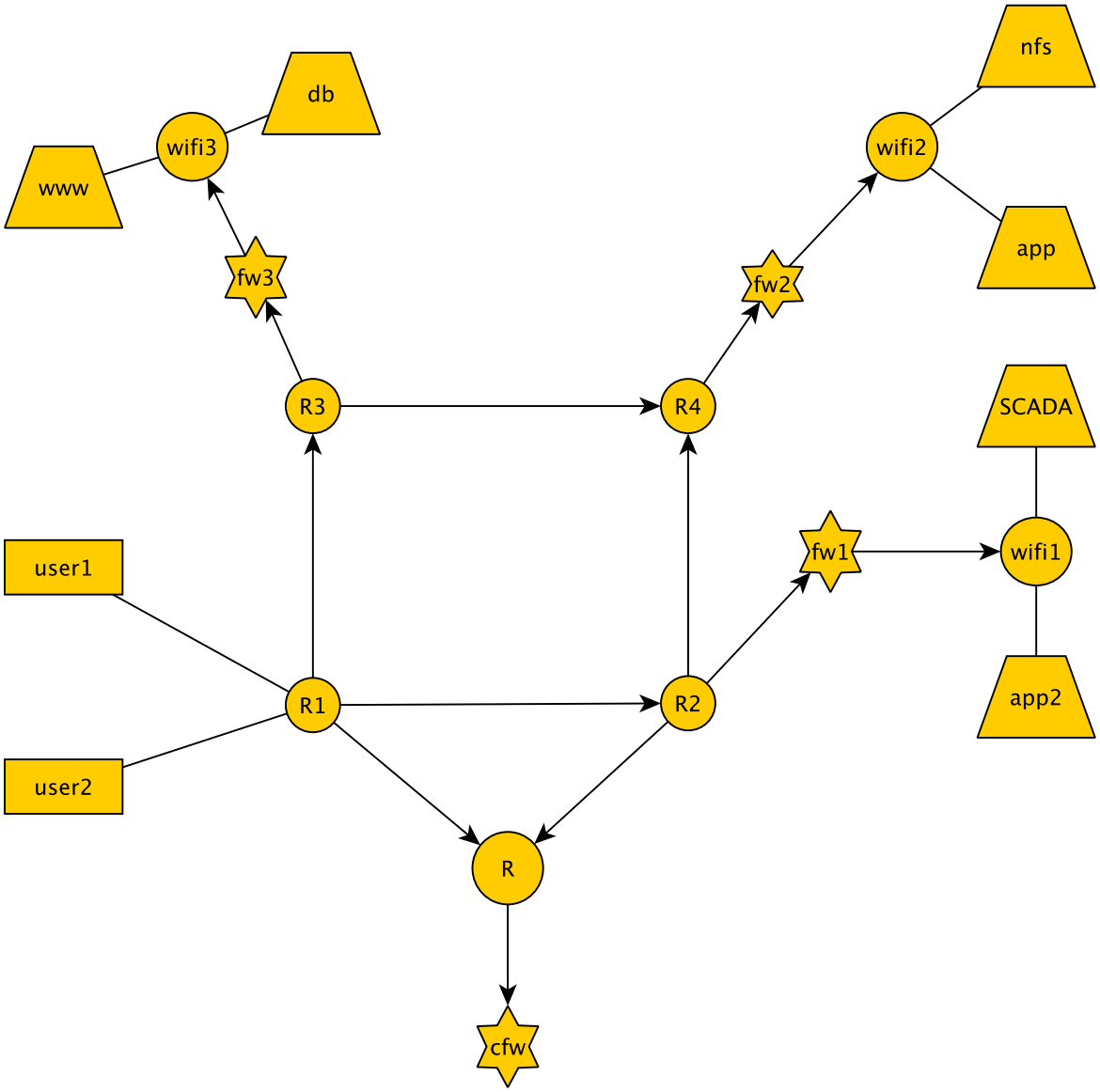

Figure 8.6: Experiment 3 Network Layout

The second set of mappings deal with firewall policies that will be configured on firewall-enabled devices within the respective cyber spaces of the countries. Table 8.8 below shows the mapping of high-level firewall policies on the various firewall-enabled devices.

Table 8.8: Firewall Policy Mappings for Experiment 3

S/No	Realm	Devices
1	Baratheon	SCADA, app2
2	Vale	app, nfs
3	Westerlands	www, db
4	Dorne	user1, user2

4. **Experiment Overview**

   The following is a general overview of both the policy intention and network layout setup of the cyber security challenge proposed in this experiment.

   From Figure 8.5 above, the nodes such as Dorne, RT-Recusants, any and others are used to represent group of network devices with identical policy intentions in the experiment. It should be noted that the graph uses a combination of BGP, firewall and cyber security competition policy intentions. The labels on the nodes are used to specify the policy realm of a network device during the design of the network layout. There are three sets of edges used in the policy intention graph of this experiment that have been used to abstract BGP relationships, firewall relationships and cyber security competition relationships. The first set of edges colored blue are used to represent the BGP business or inter-domain routing relationships between two interconnected service providers in the policy intention graph. The labels such as peer, providers, customer and others on these edges are used to represent the kind of BGP business relationships between the two interconnected service providers. The second set of edges are colored red and green and used to represent firewall relationships between two interconnected realms in the policy intention graph. The edges that are colored red are labelled deny and are used to abstract firewall rules that deny communication based on certain protocols between the two interconnected realms. The edges that are colored green are labelled permit and are used to abstract firewall rules that permit communication based on certain protocols between the two interconnected realms. The last set of edges colored black are used to represent the relationships between two interconnected cyber security competition teams. The labels on the edges which includes comprehensive, defensive and offensive are used to represent the relationship between the two interconnected teams. It should be noted that color of the edges in the graph has no effect during the execution of the experiment as our system uses the labels and custom properties that have been defined by the network administrator during the execution of the experiment.

   From Figure 8.6 above, ellipse nodes such as R, R1, wifi3 and others represent routers; the two rectangular nodes user1 and user2 are used to represent user computing devices; 6-pointed star such as cfw, fw3 and others are used to represent firewalls; and trapeziod nodes such as nfs, SCADA and others are used to represent backend network servers. It should however be emphasised that all these devices need to specified using the custom property, ***dtype***, explained Chapter 4.2.1 before our system will accept and know what type of network device it will be dealing with during execution. The edges (both directional and non-directional) used in the graph are used to represent the logical links between the two interconnected

nodes (or network devices) in the experiment.

5. **Expected Outcome**

   This section of the experiment will be used to analyse and discuss the deployed cyber security competition layout which is a combination of the entire individual team infrastructure. It should be noted that only the firewall within the proposed infrastructure will be discussed in detail as similar or more complex routing abstractions have already been discussed in previous sections of the thesis. Figure 8.7 below shows the complete cyber infrastructure that will be used during the competition.

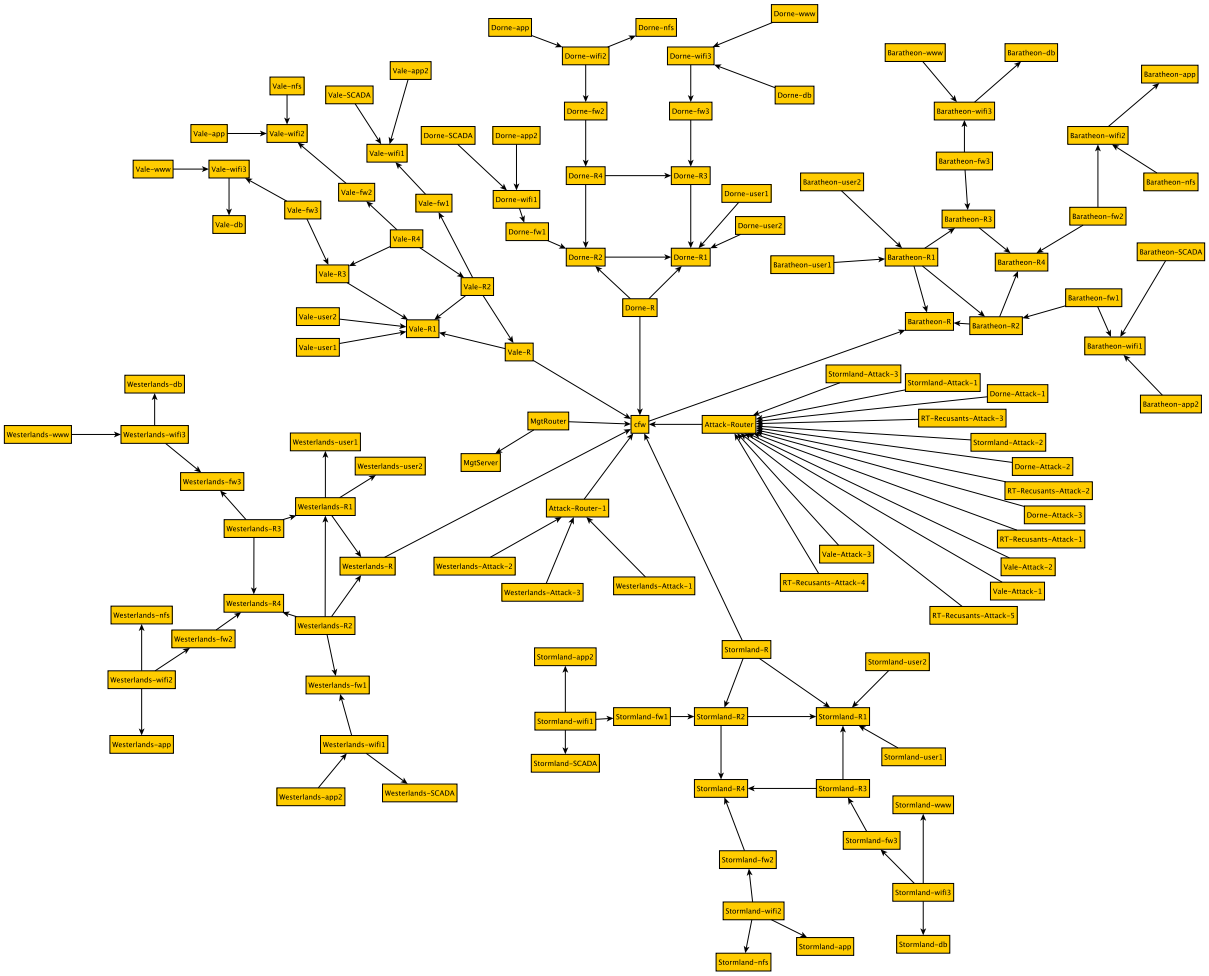

Figure 8.7: Experiment 3 Replicated Infrastructure

The respective firewall enabled devices of the various teams will belong to firewall realms as indicated in Table 8.8 above. The firewall rule generation for this set of policy intention abstractions are as indicated in Table 8.6 above and the NePAS compilation process carried here is similar to the one discussed in previous experiments in this research thesis.

The management server rules for the competition infrastructure are configured in the central firewall and all other firewall devices so as to enable the management server access all the devices within the infrastructure. The vulnerable server of the four combatant teams (Stormland, Westerlands, Dorne and Vale) that have the management option value specified as TRUE have access rules configured in both their most upstream firewall devices and the central firewall. This is to enable the vulnerable servers to have access to the management server.

The attack servers of the competition also have firewall rule relations configured in both the central firewall and all other firewalls in their respective cyber infrastructure. This is so as to ensure they have access to all vulnerable servers in their infrastructure so they can fix the vulnerabilities. The attack servers of all the teams (Stormland, Westerlands, Dorne, Vale and RT-Recusants) also have access rules in both the central firewall and other firewall devices in the cyber infrastructure. This is so as to give them access to the vulnerable servers of their opponents.

8.2 Evaluation of Abstractions and NePAS

The work presented is this thesis has been motivated by the need to provide network administrators a mechanism for expressing high-level network policies independent of the network layouts. The goal of any such mechanism is to get rid of low level details of network policy configuration so that administrators and researchers can focus on the bigger picture of analysing, measuring and evaluating various network models. A system has been implemented in this thesis using a methodology based on the template-design approach of network abstraction called NePAS. The implemented system accepts two or more graphs, one for high level network policy intention representations and the other(s) for the proposed network layout. NePAS uses the graphs to generate low level configuration commands after resolving known anomalies. In this section, we evaluate our abstraction models based on the system, NePAS developed by listing some limitations and deficiencies it has based on three areas - scalability, flexibility and how it eases the network management process. NePAS currently supports abstract models of three policies - inter domain routing, firewalls and cyber security competitions.

8.2.1 Scalability

This section hopes to assess the scalability of our proposed abstraction and to what extent can our system handle such specifications. The assessment of scalability will be done in respect to two aspects as follows - (i) the scalability and limitation of our abstraction models in terms of the number of policies it can handle, (ii) the scalability of the network layout for any proposed experiment. For the first aspect which hopes to abstract

high level network policies, there are no limitations for BGP and firewall relationships. Therefore network administrators can abstract as much BGP business relationships and firewall rule relations as their routing or security policy entails. There is however a limitation on the number of attack servers and hence *members* option in the cyber security competition policy abstraction. Our cyber security competition abstraction model while highly scalable, the system developed, NePAS currently supports the deployment of a maximum of seventy attack servers for all teams in a competition that will be deployed during the compilation process. The second aspect of the scalability evaluation deals with the size or scaling of network layouts that are proposed for experiments to be conducted. There are no network layout limitations for experiments based on firewall rule relationships and cyber security competitions except when they have BGP business relationships within. This is because when no BGP business relationship is running within a cyber security competition or a firewall experiment, all the network devices are by default assigned an AS number of 1 and regarded to be within the same AS network. On the other hand however, when an experiment has BGP business relationships, the network devices and routers (that have no *bgppol* values) will automatically belong to the AS network of the nearest router with a *bgppol* as stated in Chapter 4.3 above. There is a limitation of three layers of routers that NePAS supports before it breaks and not be able to go further in search of routers with *bgppol* value assigned. This can adversely affect the scalability of network layouts that are deployed by our system. As far as we know, our abstractions are highly scalable in terms of design but suffer some limitations when experiments are to be deployed due to system (or NePAS) limitations. It should also be noted that as with any virtualised experiment, the hardware resources can affect the scale of experiments.

8.2.2 Flexibility

This section is used to assess the flexibility of our proposed policy intention abstractions. The flexibility of the our proposed abstraction models used in this context implies how various network administrators can express any supported policy intention for any proposed network experiment. The flexibility of our abstraction models will be evaluated on a per network policy basis in the following section:

- **BGP** - Our inter domain routing abstraction model is able to provide network administrators the ability to express high level BGP business relationships. It however is not flexible enough to abstract lower level BGP routing policies such as (i) partial transit relationships or ability to provide transit to customers over a selected set of routes, (ii) preferring an exit from an AS network amongst multiple options is not currently supported by our abstraction model as well. However, our

proposed abstraction model supports the changing of routing relationships in an experiment by changing the edge label in the policy intention graph.

- **Firewall** - Our firewall abstraction model is able to provide network administrators the ability to express high level firewall rule relationships. Our abstraction model is highly modular and can express any type of rule relation a security policy requires. The abstraction of firewall rule relationship on the network layout level gives network administrators greater flexibility to express unique firewall rule relations. The only limitation of our abstraction model is that it does not support network address translation (NAT) currently.

- **Cyber Security Competitions** - Our cyber security competition abstraction model is able to provide white team members an avenue of implementing the designs of their competition infrastructure in an easy and modular way. The major limitation of our abstraction model has to do with its lack of support for white team members to assign specific IP Addresses on vulnerable machines of the various participating. This is because only one team infrastructure is given and NePAS automatically replicates it into the number of participating team infrastructures with IP addresses generated automatically. However, our proposed abstraction model give organisers of such competitions an avenue of changing competition infrastructures within short periods of time compared to other configuration processes or tools.

8.2.3 Ease of Network Management Process

This section hopes to assess how our proposed sets of abstractions make the network management process less cumbersome for network administrators. The section will be comparing the manual configuration process to our set of network management process abstractions. It should be noted that even current projects such as Cisco VIRL require administrators to configure advanced policies manually on network devices which has the same set of issues as the manual configuration process. This section will be evaluated on a per network policy basis in the following section:

- **Firewall** - The firewall rule abstractions proposed in this research are used to create multiple firewall rules for network devices in a proposed experiment. For example, the hypothetical bank experiment discussed in the above section uses 7 realms to abstract 24 high level firewall rule intentions. These realms that the various network devices in the proposed experiment belong to are used to generate more than 100 low level Cisco ASA firewall rules at deployment. This goes to show that low level firewall commands can be easily specified using a high level language represented as graphs and deployed in an efficient and time sensitive manner.

- **Cyber Security Competitions** - The CDX abstractions proposed in this research enables organisers of cyber security competitions to design and deploy complex sandpit environments using two or more graphs depending on the requirements. For example, the competition proposed in the above section encompasses a network with both firewalls and routers in addition to vulnerable servers. The above competition was deployed using 17 Kali Linux servers for the various teams, a total of 40 linux server that were used as web, database and application servers amongst others, 15 firewall devices were deployed for the entire teams, 25 network routers, 15 wireless routers, 2 core routers that were used to connect the Kali Linux servers of the various teams to the competition, a core firewall device that was used to restrict access to competition for the various teams and a core management router and server used for monitoring the competition. A total of 117 network devices were deployed in a virtual sandpit environment for the above proposed competition using only graphs - one for the policy intention and the other for the network layout specification. The policy intention graph was used to specify the relationship between the various teams and also specify the firewall and inter domain routing relationships for the network devices that were deployed. The network layout graph was used to specify the slice of network infrastructure that was replicated for all teams except the red team, Recusants. This goes to show that complex network experiments can be specified in an easy, understandable manner and deployed without too the organisers having to configure hundreds of network devices which would have taken hours or even days to do.

8.3 Closing Remarks

This chapter of the thesis was used to critically analyse our proposed abstraction models using complex networks experiments that has policy anomalies injected so as to see how our system, NePAS will handle them. The first part of the chapter described experiments of a mini-Internet, a bank's security policy and a multi-country computer emergency response team competition. These experiments provided the opportunity to discuss some complex features of our proposed network policy abstractions and features that do not work. The second part of the chapter was used to evaluate the usability, flexibility and scalability of the various network policy abstractions proposed in this research thesis.

Chapter 9

Conclusion

9.1 Review and Achievements

The Internet has transformed the way we communicate, learn and trade with each across the world. Behind the revolutionary Internet is however a set of complex network policies woven together to make it efficient, reliable and secure for users to interconnect. These network policies used in interconnecting various autonomous networks that make up the Internet are usually designed on a high level but have to be implemented using low level vendor specific languages on network devices. The configuration of these network devices is a labour intensive, tedious and an error-prone process due to human complacency and the time consuming nature of the process. Network abstractions have elicited attention in recent years as a way of minimising some of the issues related to the manual configuration process. Network abstraction aids automated network configuration by hiding some configuration processes away from network administrators so they can focus on high level network policy problems. There are two known approaches used by researchers for abstracting the network management process - template design approach and language design approach. The proposed set of abstractions in this research thesis uses template design approach where configuration snippets and contents from a network information database (NIDB) are used to generate configuration files for networks. The network information database used in this research uses graphs to allow network administrators provide both policy intentions and network layouts for any given experiment. The current network abstraction projects that have been developed by researchers are at the network device level. The projects such as Autonetkit, Cisco VIRL and many others only give users the ability to model network devices with basic network configurations. The research that was presented in this thesis enables administrators to abstract the network management process on a higher level than any other project currently available.

The first objective of this thesis was to provide an avenue where network admin-

istrators will be able to specify high level network policy intentions independent of network layouts. This objective was achieved by providing a set of graph specifications for expressing policy intention such as firewall rule relationships. The abstractions developed use the nodes of a graph as a way of grouping network devices with identical policy requirements called realms during an experiment. All devices within a realm will have identical network policies implemented for them during deployment. The edge labels of the graphs were used to represent the relationship between two realms in an experiment. Both nodes and edges in the graph can have additional policy details that have either mandatory or optional property values required in order to fully express policy intentions.

The second objective of this thesis was to provide an avenue where network administrators can specify the network infrastructure that will be used during any proposed experiment. This was achieved by providing network administrators an avenue for expressing network layouts using graphs. The nodes of the graph were used to represent network devices while the edges were used as physical links between the two devices in any given experiment. Multiple graphs can be used to model complex network layouts during experiments. For example, when modelling a network layout of a financial institution, individual graphs can be used to represent branch, trading and other networks as required for the experiment. Node and edge properties are used to provide additional details such as which realm a network device belongs to and flexibility of allocating IP addresses by network administrators amongst others.

The third objective of this thesis was to integrate a policy anomaly resolution algorithm so as to ensure all network experiments that will be generated by our abstractions models behave as intended by the network administrators. This has been and before low level configuration commands are generated, a policy anomaly resolution algorithm goes through the proposed network experiment models and resolves some of the major policy anomalies for both inter-domain routing and firewall relationships.

The fourth objective achieved for this thesis was the development of a system based on the template design approach to validate our sets of proposed network abstractions. The system developed, called network abstraction system or NePAS, was made up of four phases: policy intention; network layout; anomaly resolution and compilation. The **policy intention phase** of the system was used to specify high level network policy intentions abstracts on a graph independent of the proposed network layout for the various experiments conducted in this thesis. The nodes in this phase were used to represent realms of network devices with identical network policies. The edges are used to specify the relationships between the realms. Node and edge optional values are used to abstract network policy details. The **network layout phase** of the system was used to specify the proposed network topologies of the various experiments conducted in this thesis in a graph-like format similar to the first phase. This phase also provided another

layer of abstraction even though it was on a lower level compared to the first phase. The nodes in this phase were used to represent actual network devices and edges were used to represent the physical links between the devices. NePAS in this phase was used to specify the network policy intention realm of each device and the type of network device in all the experiments conducted. The **anomaly resolution phase** was used to ensure a number of policy that can manifest during experiments were resolved. The **compilation phase** was used to generate low level anomaly free VIRL configurations for all the experiments undertaken in this thesis. The system developed, NePAS, can be used to model an existing or proposed network and deployed in a virtual environment. For example, an ISP can use our BGP abstraction models to design and deploy a replica of its network in a virtual sandpit so as to analyse possible route leaks or other anomalies within the network. To do this, the network administrator of the ISP will design both the policy intention and network topology of their network. Our abstraction models can also be used to model network experiments before they are deployed in the real-world. For example, an enterprise network can model their distributed firewall network so as to ensure there are no firewall anomalies in the design before deploying such configuration commands on the intended network devices.

The last objective of this thesis was to conduct a set of network experiments deployed in a virtual environment using our abstraction models with known policy anomalies purposefully injected so as to see how NePAS will resolve them. A series of experiments were conducted to critically examine the flexibility, scalability and correctness of the configuration commands generated by NePAS from our abstraction models.

- The first experiment was a mini-Internet simulation conducted to ensure strict compliance of BGP business relationships between six autonomous systems. In this experiment, some eBGP links were designed with no BGP business relationships and NePAS did not generate any configuration commands for such link between BGP neighbours. The experiment highlighted the strengths of our abstraction models in ensuring anomalies such as origin misconfiguration, private AS announcements and export configurations are avoided during low level deployments.

- The second experiment conducted was a financial service provider network with two branch locations and the bank founder's home. In this experiment, the policy intention between some realms were designed with some firewall anomalies. The various firewall anomalies were resolved by NePAS in accordance to the policy intention intended. The experiment highlighted the strengths of our abstractions in avoiding intra-firewall anomalies such as irrelevance, redundancy and shadowing; and inter-firewall redundancy during low level deployments.

- The third experiment conducted was a comprehensive cyber war between four countries while a nefarious group of hackers (Red Team) were also launching

attacks on the cyber infrastructure of the various countries. A peace mission cyber infrastructure (Blue Team) was also deployed in order to help the countries improve there information assurance skills. The cyber infrastructure of the various countries was composed of firewall rules between the various network devices and BGP business relationships between the routers in the experiment.

As stated the research contribution of this thesis is premised on a set of high level network policy abstractions based on template design approach. The contributions detailed in this research are as follows:

The first contribution of this thesis is the specification of a set of highly scalable and easy to implement high level BGP business relationship abstractions. The BGP abstractions proposed in this thesis enables network administrators to specify business relationships such as: provider, customer, peer, sibling and hybrid. These business relationships are specified as edge labels in the graphs between autonomous systems with identical routing policies called realms. The nodes of the graph are used to represent these realms in any proposed network experiment. The edges of the graph are used to express the relationship between the realms by labelling the edge with the appropriate business relationship as contained in the routing policy document of the experiment to be conducted. The only optional node property proposed has to do with the specification of the autonomous system number which can be automatically generated by our system if a network administrator omits it when designing high level policy intentions. These high level BGP business relationships are specified independent of the network layout. BGP community attribute is used to to label incoming advertised provider, customer and peer routes while local preference attribute is used in preferring routes based on the following order - customer, sibling, peer, provider. This makes it easier to export a realms routes according to the high level policy specification.

The second contribution of the thesis is the specification of a set high level firewall relationship abstractions that have varying degrees of flexibility and granularity. There are two ways in which firewall relationships can be expressed using our abstractions - policy level and network level. The policy level firewall abstractions are independent of network layouts and group firewall enabled devices with identical security policies into realms represented as nodes in the firewall policy intention graph. The edges of firewall policy intention graphs are used to express the firewall rule action that will be taken. There are various edge properties that have been implemented to get firewall tuple details from the network administrator. These edge properties have default values in case any of them are omitted by the network administrator. The second way of specifying firewall abstractions is at the network layout level and gives network administrators a more granular way of specifying security policies between two firewall enabled devices in an experiment. The edge properties of network level firewall abstractions are specified

in the same way as the policy level firewall abstractions discussed. The firewall rules generated in both the policy intention and network layout rules go through a number of anomaly resolution algorithms so as to ensure the deployed firewall configurations do not have any anomalies.

The third contribution of this thesis is the expression of cyber security competition high level abstractions in such a way that eases the deployment of proposed infrastructure in a virtual sandpit environments. The high level cyber security competition abstractions group cyber infrastructures based on the participating teams. The nodes of the policy intention graph are used to represent the various participating teams while the edge labels between the teams are used in specifying the competition approach. The edges of the graph can have any of the following labels depicting the competition approach - comprehensive, defensive or offensive. The network layout graph design by organisers of such competitions will be replicated for all participating teams with the name of each team prepended before all network devices. A central firewall device is used to separate the various participating teams and management systems from each other in the competition infrastructure. A members node option is used to specify the number of team mates within each participating team. This option influences the number of attack servers that will be deployed during low level configuration of the competition infrastructure. The competition approach influences the firewall relationship between the attack servers and the vulnerable infrastructure of an opponent.

9.2 Future Work

The critical evaluation of our work in Chapter 8 above has helped us to identify several issues for future work. The issues can be categorised into two groups: (i) existing network policy abstractions and (ii) additional network policy abstractions.

9.2.1 Existing Network Policy Abstractions

This category deals with improvements on existing network management process abstractions proposed in this research thesis. The future work discussed in this section is used to proffer possible improvements on the existing sets of abstractions that have been researched and developed in this thesis. The following section takes all the contributions made in this research work in order to analyse how they can be improved.

- BGP

  We have researched network management process abstraction techniques for configuring BGP business relationships such as peering, sibling, provider and customer relations. Trying to extend the abstraction research so as to provide

additional flexibility to the current research work is a likely avenue for any future work.

Future work is needed to extend our system to ensure there is adequate network connectivity to routes inserted using the *network* node option so as to avoid the non-connectivity anomaly that may likely arise in abstractions models.

Future work is needed to extend the flexibility of current BGP business relationship abstraction research to support partial transit abstraction. This functionality will help provide a more flexible avenue for network administrators to be able to select the routes that will be advertised by an AS depending on the routing policy intention of the experiment.

Future work is also needed to extend the flexibility of BGP policies to support lower levels of abstraction such as choosing a particular exit from a group within an AS. This functionality will provide network administrators an avenue for deploying experiments that are a true representation of their respective policy intentions.

- Firewall

  We have researched abstraction techniques for firewall relationships in this thesis. Our abstractions allow for the specification of all kinds of rule relations with relative ease. However the system implemented to showcase these abstractions is not very efficient in terms of rule traversal. This is because the system generates firewall rule relations based on a per device basis without regard for IP address aggregation which makes firewall traversal cumbersome.

  Future work is needed to improve the performance of firewall devices by ensuring all firewall rule relations that have been abstracted will have the source and destination IP addresses aggregated whenever it is possible. This will ensure fewer firewall rules are generated during the low level compilation process and hence faster rule traversal.

  Future work is needed to improve on the anomaly resolution of NePAS to handle intra and inter firewall correlation anomalies. This will further optimise the performance of firewall devices configured using our proposed abstractions.

  Additional future work is needed to extend the firewall abstraction research to include Network Address Translation (NAT). As an important policy concept, including NAT abstraction will make it easier for network administrators in both the industry and academia to deploy networks that require address translation.

- Cyber Security Competitions

  We have researched abstraction techniques for cyber security competitions in this thesis. However the system built to evaluate the performance and limitations of our

abstractions is not very scalable at the moment as it only allows for the specification of seventy participants during any competition. This limitation restricts the number of team mates and hence the number of teams that can participate in any cyber security competition.

Future work is needed to handle a higher number of attack servers by adding more attack routers in the cyber security competition infrastructure. This will help organisers to have many participating teams and team members during such competitions.

Researching abstractions that will give organisers of such competitions the flexibility of manually choosing IP addresses for different vulnerable servers will be an important improvement.

9.2.2 Additional Network Policy Abstractions

This category deals with future work that hopes to abstract more network policies that will improve the network management process by eliminating issues highlighted with the manual configuration process.

Researching high level policy intention abstractions for network policies such as virtual private networks (VPNs), content data networks, software defined networks and many others will be an important direction of any future work to be carried out. Abstracting such important network concepts and evaluating the limitations and performance implication of such abstractions on networks will greatly help the network management community.

9.3 Closing Remarks

The research work presented in this thesis has been motivated by the need to provide network policy abstractions so as to simplify the network management process and prevent issues that arise from the manual configuration process. The results of our research work will benefit the entire network management community. The set of network policy abstractions presented will benefit those in the industry and academia with an avenue to experiment, measure and evaluate various network projects. Our network policy abstraction ideas can be successfully used in both simulated and real-world network deployments. It constitutes a background basis for helping researchers in the network policy abstractions area.

References

[1] Configuring ip access lists. http://www.cisco.com/c/en/us/support/docs/security/ios-firewall/23602-confaccesslists.html?referring_site=RE&pos=1&page=http://www.cisco.com/c/en/us/td/docs/ios/12_2/security/configuration/guide/fsecur_c/scfacls.html. Accessed: 2017-02-12.

[2] Firewall builder | simplifying firewall management. http://www.fwbuilder.org/. Accessed: 2016-08-17.

[3] (2010). Open capture the flag 6 whitepaper. .

[4] (2013a). Autonetkit. http://www.autonetkit.org/.

[5] (2013b). Defcon hacking conference. http://www.defcon.org/html/links/dc-faq/dc-faq.html.

[6] (2013c). The ucsb international capture the flag (ictf).

[7] Abbes, T., Bouhoula, A., and Rusinowitch, M. (2016). Detection of firewall configuration errors with updatable tree. *International Journal of Information Security*, 15(3):301–317.

[8] Adams, W. J., Lacey, T., Leblanc, S. P., and Gavas, E. (2009). Collective views of the nsa/css cyber defense exercise on curricula and learning objectives. In *Proceedings of the 2nd conference on Cyber security experimentation and test*, page 2.

[9] Adao, P., Bozzato, C., Rossi, G. D., Focardi, R., and Luccio, F. (2014). Mignis: A semantic based tool for firewall configuration. In *IEEE 27th Computer Security Foundations Symposium (CSF)*.

[10] Agarwal, A., Luniya, R., Bhatnagar, M., Gaikwad, M., and Inamdar, V. (2012). Reviewing the world of virtualization. In *2012 Third International Conference on Intelligent Systems Modelling and Simulation*, pages 554–557.

[11] Al-Musawi, B., Branch, P., and Armitage, G. (2017). Bgp anomaly detection techniques: A survey. *IEEE Communications Surveys Tutorials*, 19(1):377–396.

[12] Al-Shaer, E., Hamed, H., Boutaba, R., and Hasan, M. (2005). Conflict classification and analysis of distributed firewall policies. *IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS*, 23(10):2069–2084.

[13] Al-Shaer, E. S. and Hamed, H. H. (2004a). Discovery of policy anomalies in distributed firewalls. In *Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*.

[14] Al-Shaer, E. S. and Hamed, H. H. (2004b). Modeling and management of firewall policies. *IEEE Transactions on Network and Service Management*, 1(1):2–10.

[15] Anwar, R., Niaz, H., Choffnes, D., Cunha, Í., Gill, P., and Katz-Bassett, E. (2015). Investigating interdomain routing policies in the wild. In *ACM Conference on Internet Measurement Conference*, pages 71–77.

[16] Balakrishnan, H. (2009). Wide-area internet routing. Massachusetts Institute of Technology Department of Electrical Engineering and Computer Science.

[17] Bartal, Y., Mayer, A., Nissim, K., and Wool, A. (2004). Firmato: A novel firewall management toolkit. *ACM Transactions on Computer Systems (TOCS)*, 22(4):381–420.

[18] Bellovin, S. M. and Bush, R. (2009). Configuration management and security. *IEEE Journal on Selected Areas in Communications*, 27(3):268 – 274.

[19] Bhagchandka, D. (2003). *Classification of Firewalls and Proxies*.

[20] Bonaventure, O., Uhlig, S., and Quoitin, B. (2004). The case for more versatile bgp route reflectors.

[21] Bukhatwa, F. and Patel, A. (2003). Effects of ordered access lists in firewalls. In *in: Proceedings of IADIS WWW/Internet International Conference (W3I 2004*, pages 257–264.

[22] Butler, K., Farley, T. R., McDaniel, P., and Rexford, J. (2010). A survey of bgp security issues and solutions. *Proceedings of the IEEE*, 98(1):100–122.

[23] Caesar, M. and Rexford, J. (2005). Bgp routing policies in isp networks. In *IEEE Network*, volume 19, pages 5–11. IEEE.

[24] Carlin, A., Manson, D. P., and Zhu, J. (2010). Developing the cyber defenders of tomorrow with regional collegiate cyber defense competitions (ccdc). *Information Systems Education*, 8(14):1–10.

[25] Chandra, R. and Traina, P. (1996). BGP Communities Attribute). RFC 1997, RFC Editor.

[26] Chen, X., Mao, Y., Mao, Z. M., and der Merwe, J. V. (2010). Decor: Declarative network management and operation. *ACM SIGCOMM Computer Communication Review*, 40(1):61 – 66.

[27] Childers, N., Boe, B., Cavallaro, L., Cavedon, L., Cova, M., Egele, M., and Vigna, G. (2010). Organizing large scale hacking competitions. In *Detection of Intrusions and Malware, and Vulnerability Assessment*, pages 132–152. Springer.

[28] Chowdhury, N. M. K. and Boutaba, R. (2010). A survey of network virtualization. *Computer Networks*, 54:862–876.

[29] Chowdhury, N. M. M. K. and Boutaba, R. (2009). Network virtualization: state of the art and research challenges. *IEEE Communications Magazine*, 47(7):20–26.

[30] Chu, B.-T., Ahn, G.-J., Blanchard, S., Deese, J., Kelly, R., Yu, H., and Young, A. (2007). Collegiate cyber game design criteria and participation. In *The 6th IEEE/ACIS International Conference on Computer and Information Science*, pages 1036 – 1041. IEEE.

[31] Cisco. Cisco adaptive security device manager. http://www.cisco.com/c/en/us/products/security/adaptive-security-device-manager/index.html. Accessed: 2017-02-12.

[32] Cisco. Virl - virtual internet routing lab. http://virl.cisco.com/. Accessed: 2017-02-12.

[33] Cittadini, L., Vissicchio, S., and Battista, G. D. (2010). Doing don'ts: Modifying bgp attributes within an autonomous system. In *2010 IEEE Network Operations and Management Symposium - NOMS 2010*, pages 293–300.

[34] Coffman, E. G., Ge, Z., Misra, V., and Towsley, D. (2006). Network resilience: Exploring cascading failures within bgp?

[35] Cuppens, F., Cuppens-Boulahia, N., Garcia-Alfaro, J., Moataz, T., and Rimasson, X. (2012). *Handling Stateful Firewall Anomalies*, pages 174–186. Springer Berlin Heidelberg, Berlin, Heidelberg.

[36] Dodge, R. C., Ragsdale, D. J., and Reynolds, C. (2003). Organization and training of a cyber security team. In *IEEE International Conference on Systems, Man and Cybernetics*, volume 5, pages 4311– 4316. IEEE.

[37] Donnet, B. and Bonaventure, O. (2008). On bgp communities. *SIGCOMM Comput. Commun. Rev.*, 38(2):55–59.

[38] Doupe, A., Egele, M., Caillat, B., Stringhini, G., Yakin, G., Zand, A., Cavedon, L., and Vigna, G. (2011). Hit 'em where it hurts: A live security exercise on cyber situational awareness. In *Proceedings of the 27th Annual Computer Security Applications Conference*, pages 51–61. ACM.

[39] Dube, R. (1999). A comparison of scaling techniques for bgp. *SIGCOMM Comput. Commun. Rev.*, 29(3):44–46.

[40] Enck, W., McDaniel, P., Sen, S., Sebos, P., Spoerel, S., Greenberg, A., Rao, S., and Aiello, W. (2009). Configuration management at massive scale: System design and experience. *IEEE Journal on Selected Areas in Communications*, 27:323 – 335.

[41] Ernits, M., Tammekänd, J., and Maennel, O. (2015). i-tee: A fully automated cyber defense competition for students. In *ACM Conference on Special Interest Group on Data Communication*, pages 113–114.

[42] Feamster, N., Winick, J., and Rexford, J. (2004). A model of bgp routing for network engineering. In *Proceedings of the joint international conference on Measurement and modeling of computer systems*, pages 331–342.

[43] Frantzen, M., Kerschbaum, F., Schultz, E., and Fahmy, S. (2001). A framework for understanding vulnerabilities in firewalls using a dataflow model of firewall internals1. *Computers and Security*, 20(3):263 – 270.

[44] Furtuna, A., Patriciu, V.-V., and Bica, I. (2010). A structured approach for implementing cyber security exercises. In *The 8th International Conference on Communications (COMM)*, pages 415–418. IEEE.

[45] Gao, L. (2001). On inferring autonomous system relationships in the internet. In *IEEE/ACM Transactions on Networking*, volume 9, pages 733–745.

[46] Gao, L. and Rexford, J. (2001). Stable internet routing without global coordination. *IEEE/ACM Transactions on Networking (TON)*, 9(6):681–692.

[47] Garcia-Alfaro, J., ÌĄde ÌĄric Cuppens, F., Cuppens-Boulahia, N., Martinez, S., and Cabot, J. (2013). Management of stateful firewall misconfiguration. *Journal Computers and Security*, 39:64–85.

[48] Geers, K. (2010). Live fire exercise: Preparing for cyber war. *Journal of Homeland Security and Emergency Management*, 7(1):8–24.

[49] Giotsas, V., Luckie, M., Huffaker, B., and kc claffy (2014). Inferring complex as relationships. In *Conference on Internet Measurement Conference*, pages 23–30.

[50] Giotsas, V. and Zhou, S. (2011). Detecting and assessing the hybrid ipv4/ipv6 as relationships. *ACM SIGCOMM Computer Communication Review - SIGCOMM '11*, 41(4):424–425.

[51] Gottlieb, J., Greenberg, A., Rexford, J., and Wang, J. (2003). Automated provisioning of bgp customers. In *IEEE Network: The Magazine of Global Internetworking*, volume 17, pages 44–55.

[52] Griffin, T. G. and Wilfong, G. (1999). An analysis of bgp convergence properties. *ACM SIGCOMM Computer Communication Review*, 29(4):277–288.

[53] Griffin, T. G. and Wilfong, G. (2002). Analysis of the med oscillation problem in bgp. In *10th IEEE International Conference on Network Protocols, 2002. Proceedings.*, pages 90–99.

[54] Hamed, H. and Al-Shaer, E. (2006a). Dynamic rule-ordering optimization for high-speed firewall filtering. In *ACM Symposium on Information, computer and communications security*, pages 332–342.

[55] Hamed, H. and Al-Shaer, E. (2006b). Taxonomy of conflicts in network security policies. *IEEE Communications Magazine*, 44(3):134–141.

[56] Henry, P. (2001). An examination of firewall architectures. Technical report, CyberGuard Corporation.

[57] Holland-Minkl, A. M. (2006). Cyberattacks: A lab-based introduction to computer security. In *Proceedings of the 7th Conference on Information Technology Education*, pages 39–46. ACM.

[58] Hucaby, D. (2005). *Cisco ASA and PIX Firewall Handbook*. Cisco Press.

[59] Huston, G. (2006). Exploring autonomous system numbers. *The Internet Protocol Journal*, 9(1):2–23.

[60] Hyun, Y., Broido, A., and kc claffy (2003). Traceroute and bgp as path incongruities. Technical report, Cooperative Association for Internet Data Analysis (CAIDA).

[61] Karlin, J., Forrest, S., and Rexford, J. (2008). Autonomous security for autonomous systems. *Computer Networks*, 52(15):2908 – 2923. Complex Computer and Communication Networks.

[62] Knight, S., Maennel, O., Roughan, M., Phillips, I., and Jaboldinov, A. Network configuration: from abstraction to emulation.

[63] Kruegel, C., Mutz, D., Robertson, W., and Valeur, F. (2003). *Topology-Based Detection of Anomalous BGP Messages*, pages 17–35. Springer Berlin Heidelberg, Berlin, Heidelberg.

[64] Kumar, M. and Kumar, S. (2014). Improving routing in large networks inside autonomous system. *International Journal of System Assurance Engineering and Management*, 5(3):383–390.

[65] Lee, C. P., Uluagac, A. S., Fairbanks, K. D., and Copeland, J. A. (2011). The design of netseclab: A small competition-based network security lab. *IEEE Transactions on Education*, 54(1):149–155.

[66] Lee, S., Wong, T., and Kim, H. S. (2008). To automate or not to automate: On the complexity of network configuration. In *IEEE International Conference on Communications*, pages 5726 – 5731. IEEE.

[67] Levin, D. (2003). Lessons learned in using live red teams in ia experiments. In *Proceedings of the DARPA Information Survivability Conference and Exposition*, volume 1, pages 110–119. IEEE.

[68] Louthan, G., Roberts, W., Butler, M., and Hale, J. (2010). The blunderdome: An offensive exercise for building network, systems and web security awareness. In *Proceedings of the 3rd International Conference on Cyber Security Experimentation and Test*, pages 1–7. USENIX Association.

[69] Luckie, M., Huffaker, B., Dhamdhere, A., Giotsas, V., and claffy, k. (2013). As relationships, customer cones, and validation. In *Proceedings of the 2013 Conference on Internet Measurement Conference*, IMC '13, pages 243–256, New York, NY, USA. ACM.

[70] M, S., M, H. G., A, N., and J, U. (2013). Performance analysis of kernel-based virtual machine. *International Journal of Computer Science and Information Technology (IJCSIT)*, 5(1):137–144.

[71] Mahajan, R., Wetherall, D., and Anderson, T. (2002). Understanding bgp misconfiguration. In *ACM SIGCOMM Computer Communication Review*, volume 32, pages 3–16.

[72] Manadhata, P. and Sekar, V. Understanding bgp anomalies: Detection, analysis, and prevention. 15-744 Class Project Report.

[73] Mayer, A., Wool, A., and Ziskind, E. (2000). Fang: A firewall analysis engine. In *IEEE Symposium on Security and Privacy*, page 177.

[74] Mazloum, R., Buob, M.-O., Augè, J., Baynat, B., Rossi, D., and Friedman, T. (2014). Violation of interdomain routing assumptions. In *International Conference on Passive and Active Network Measurement*, volume 8362.

[75] Mirkovic, J., Reiher, P., Papadopoulos, C., Hussain, A., Shepard, M., Berg, M., and Jung, R. (2008). Testing a collaborative ddos defense in a red team/blue team exercise. *IEEE Transactions on Computers*, 57(8):1098–1112.

[76] MS, P. S., PhD, S. T., Dantu, R., and Cankaya, E. C. (2010). Experiences during a collegiate cyber defense competition. *Journal of Applied Security Research*, 5(3):382–396.

[77] Nguyen, H., Roughan, M., Knight, S., Falkner, N., Maennel, O., and Bush, R. (2011). How to build complex, large-scale emulated networks. In *Testbeds and Research Infrastructures. Development of Networks and Communities*, pages 3 – 18. Springer.

[78] O'Leary, M. (2006). A laboratory based capstone course in computer security for undergraduates. In *Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education*, volume 38, pages 2–6. ACM.

[79] O'Leary, M. (2012). Small-scale cyber security competitions. In *Proceedings of the 16th Colloquium for Information Systems Security Education*, pages 103–110.

[80] PATRICIU, V.-V. (2009). Guide for designing cyber security exercises. In *Proceedings of the 8th WSEAS International Conference on E-Activities and information security and privacy*, pages 172 – 177. World Scientific and Engineering Academy and Society (WSEAS).

[81] Peng, Z., Lu, Q., Liping, Z., and Feng, L. (2014). Internet routing policies verification method based on as relationships. In *International Conference on Logistics Engineering, Management and Computer Science*.

[82] Quoitin, B. and Uhlig, S. (2005). Modeling the routing of an autonomous system with c-bgp. *IEEE Network*, 19(6):12–19.

[83] Rekhter, Y., Li, T., and Hares, S. (2006). A Border Gateway Protocol 4 (BGP-4). RFC 4271, RFC Editor.

[84] Roschke, S., Willems, C., and Meinel, C. (2010). A security laboratory for ctf scenarios and teaching ids. In *2010 2nd International Conference on Education Technology and Computer (ICETC)*, volume 1, pages 433–437. IEEE.

[85] Sahoo, J., Mohapatra, S., and Lath, R. (2010). Virtualization: A survey on concepts, taxonomy and associated security issues. In *2010 Second International Conference on Computer and Network Technology*, pages 222–226.

[86] Sangster, B., O'Connor, T. J., Cook, T., Fanelli, R., Dean, E., Adams, W. J., Morrell, C., and Conti, G. (2009). Toward instrumenting network warfare competitions to generate labeled datasets. In *Proceedings of the 2nd conference on Cyber security experimentation and test*.

[87] Sharmila, M. V. and Kalimuthu, M. M. (2014). An survey on interdomain routing using border gateway protocol. *INTERNATIONAL JOURNAL OF ADVANCED INFORMATION AND COMMUNICATION TECHNOLOGY*, 1:597–600.

[88] Sheth, C. and Thakker, R. (2011). Performance evaluation and comparative analysis of network firewalls. In *2011 International Conference on Devices and Communications (ICDeCom)*, pages 1–5.

[89] Stone, G. N., Lundy, B., and Xie, G. G. (2001). Network policy languages: a survey and a new approach. *IEEE Network*, 15(1):10–21.

[90] Systems, D. I. (2005). Firewalls - overview and best practices. White Paper.

[91] Tozal, M. E. (2016). The internet: A system of interconnected autonomous systems. In *2016 Annual IEEE Systems Conference (SysCon)*, pages 1–8.

[92] Vanbever, L., Pardoen, G., and Bonaventure, O. (2008). Towards validated network configurations with ncguard. In *IEEE Internet Network Management Workshop*, pages 1 – 6. IEEE.

[93] Vanbever, L., Quoitin, B., and Bonaventure, O. (2009). A hierarchical model for bgp routing policies. In *Proceedings of the 2nd ACM SIGCOMM workshop on Programmable routers for extensible services of tomorrow*, pages 61–66.

[94] Vasireddy, J. (2009). *Network Monitoring Using Nagios and Autoconfiguration for Cyber Defense Competitions*. PhD thesis, California State University, Sacramento.

[95] Vasu, A. K., Ganesh, A., Ayyappan, P., and Sudarsan, A. (2014). Improving firewall performance by eliminating redundancies in access control lists. *International Journal of Computer Networks (IJCN)*, 6:92–107.

[96] Vigna, G. (2003). Teaching hands-on network security: Testbeds and live exercises. *Journal of Information Warfare*, 2(3):8–24.

[97] Voellmy, A. and Hudak, P. (2009). Nettle: A language for configuring routing networks. In *Domain-Specific Languages*, pages 211 – 235. Springer.

[98] Vohra, Q. and Chen, E. (2012). BGP Support for Four-octet AS Number Space. RFC 6793, RFC Editor.

[99] Walden, J. (2005). A real-time information warfare exercise on a virtual network. In *Proceedings of the 36th SIGCSE technical symposium on Computer science education*, volume 37, pages 86–90.

[100] Wang, A., Iyer, M., Dutta, R., Rouskas, G. N., and Baldine, I. (2013). Network virtualization: Technologies, perspectives, and frontiers. *Journal of Lightwave Technology*, 31(4):523–537.

[101] Wang, Y., Schapira, M., and Rexford, J. (2009). Neighbor-specific bgp: More flexible routing policies while improving global stability. In *Proceedings of the Eleventh International Joint Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '09, pages 217–228, New York, NY, USA. ACM.

[102] Werther, J., Zhivich, M., Leek, T., and Zeldovich, N. (2011). Experiences in cyber security education: The mit lincoln laboratory capture-the-flag exercise. In *Proceedings of the 4th conference on Cyber Security Experimentation and Test*, volume 8, pages 1–9.

[103] Willems, C. and Meinel, C. (2011). Practical network security teaching in an online virtual laboratory. In *Proceedings of the 2011 International conference on Security & Management*.

[104] Willems, C. and Meinel, C. (2012). Online assessment for hands-on cyber security training in a virtual lab. In *2012 IEEE Global Engineering Education Conference (EDUCON)*, pages 1–10. IEEE.

[105] Xavier, M. G., Neves, M. V., Rossi, F. D., Ferreto, T. C., Lange, T., and Rose, C. A. F. D. (2013). Performance evaluation of container-based virtualization for high performance computing environments. In *2013 21st Euromicro International Conference on Parallel, Distributed, and Network-Based Processing*, pages 233–240.

[106] Zhang, Y., Zhang, Y., and Wang, W. (2005). Optimization of firewall filtering rules by a thorough rewriting. In *4th Latin American Network Operations and Management Symposium*.

[107] Zhao, X., Pei, D., Wang, L., Massey, D., Mankin, A., Wu, S. F., and Zhang, L. (2001). An analysis of bgp multiple origin as (moas) conflicts. In *Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*, IMW '01, pages 31–35, New York, NY, USA. ACM.

Appendix A

NePAS User Manual

In this section, the system dependencies, requirements, commands and various the various experiments conducted in this research are discussed in detail. This user manual is split into three sections - applications, dependencies and NePAS. The three components of the user manual are discussed in the following section:

A.1 Applications

This section looks at the application required to run experiments using NePAS. There are two applications needed for designing and deploying NePAS experiments - yED and Cisco VIRL.

A.1.1 Graphical Editor

This section looks at the graphical editor, yED, is very important because it is used for populating the network information database needed by NePAS network policy template snippets for any proposed experiment. The high level policy and network layout graphs are designed using yED graphical editors and saved using file extension **graphml**. The official link to yED including the documentation and how to use it is as follows:-
https://www.yworks.com/yed

A.1.2 Network Simulator

This section looks at the network simulator, Cisco VIRL, which can be used for the deployment of proposed experiments. After designing the yED graph files, our system is used to generate Cisco VIRL files that are used for deploying the proposed network experiments. The following section shows the step-by-step process of loading and deploying proposed virtual sandpit environments using Cisco VIRL. The official link to

Cisco VIRL including the documentation and how to use it is as follows:-
virl.cisco.com/

A.2 Dependencies

This section details the various Linux tools or software needed by python in order to run our system and generate VIRL files for the network simulator described in the above section. The following shows the various dependencies and the various links that will help in configuring them:

- NetworkX - NetworkX is a Python library that was used for specification of both high level policy intentions and network layouts. The official link to NetworkX including the documentation and how to use it is as follows:-
  https://networkx.github.io/
- Netaddr - This is a network address manipulation library for Python. It was used for issuing IP addresses to interfaces of various machines in an experiment. The official link to Netaddr including the documentation and how to use it is as follows:
  https://pypi.python.org/pypi/netaddr
- LXML - This is a python library used for processing XML and HTML. It was used in the processing network information database gotten from yED graphs into generating VIRL files used for deploying network experiments. The official link to LXML including the documentation and how to use it is as follows:
  lxml.de/

A.3 NePAS

This section shows the sample usage of the system developed in order to evaluate the abstraction proposed during this research. It is expected that both the high level policy intention of a proposed experiment has been generated and the network layout as well. The following is the command that is used to run system: python nepas.py <highlevelpolicy>.graphml <networklayout>.graphml <outputfile>.virl

Appendix B

Result of Experiments

This section shows the complete low level Cisco configuration commands of the various network devices in the experiments described in the BGP, firewall and cyber security competition chapters of this research.

B.1 Firewall Chapter Experiment

Listing B.1: Firewall fw2 Configuration for Proposed University

```
! ASAv Config generated by NePAS
!
hostname fw2
username cisco password cisco privilege 15
enable password cisco
passwd cisco
names
!
interface GigabitEthernet0/1
  description to uniR
  duplex full
  nameif nepas-outside
  security-level 0
  no shutdown
  ip address 20.0.2.5 255.255.255.0
interface GigabitEthernet0/2
  description to wifi2
  duplex full
  nameif nepas-outside-1
  security-level 0
  no shutdown
  ip address 20.0.4.6 255.255.255.0
interface Management0/0
 description OOB Management
 duplex full
 management-only
 nameif mgmt
 security-level 100
 no shutdown
```

```
 ! Configured on launch
 no ip address
access-list nepas-out extended permit tcp host 20.0.1.3 host 20.0.0.2
     eq 65432
access-list nepas-in extended permit tcp host 2.3.4.5 eq 56431 host
    20.0.1.3 eq 62300
access-list nepas-in extended permit ip host 1.2.3.4 host 20.0.1.3
access-list nepas-in extended permit tcp host 20.0.0.2 eq 54321 host
    20.0.1.3 eq 54321
access-list nepas-out extended deny ip host 20.0.1.3 host 20.0.0.2
access-list nepas-out extended permit tcp host 20.0.1.3 any eq 40728
access-list nepas-out extended permit tcp host 20.0.1.3 any eq 3689
access-list nepas-out extended permit tcp any host 20.0.0.2 eq ssh
access-list nepas-out extended deny tcp any host www.facebook.com eq
    80
access-list nepas-out extended deny tcp any host www.gorillavid.in eq
     80
access-list nepas-out extended deny tcp any host www.facebook.com eq
    8080
access-list nepas-out extended deny tcp any host www.gorillavid.in eq
     8080
access-list nepas-any extended permit icmp any any
access-group nepas-out out interface nepas-outside
access-group nepas-in in interface nepas-outside-1
access-group nepas-any global
!
same-security-traffic permit inter-interface
logging enable
logging asdm informational
user-identity default-domain LOCAL
aaa authentication ssh console LOCAL
http server enable
http 0.0.0.0 0.0.0.0 mgmt
ssh 0.0.0.0 0.0.0.0 mgmt
telnet 0.0.0.0 0.0.0.0 mgmt
http 0.0.0.0 0.0.0.0 nepas-outside
ssh 0.0.0.0 0.0.0.0 nepas-outside
telnet 0.0.0.0 0.0.0.0 nepas-outside
http 0.0.0.0 0.0.0.0 nepas-outside-1
ssh 0.0.0.0 0.0.0.0 nepas-outside-1
telnet 0.0.0.0 0.0.0.0 nepas-outside-1
ssh version 2
crypto key generate rsa modulus 768
telnet timeout 15
console timeout 0
username cisco password cisco privilege 15
!
class-map inspection_default
 match default-inspection-traffic
!
!
policy-map type inspect dns preset_dns_map
 parameters
  message-length maximum client auto
  message-length maximum 512
policy-map global_policy
 class inspection_default
  inspect ip-options
```

```
  inspect netbios
  inspect rtsp
  inspect sunrpc
  inspect tftp
  inspect xdmcp
  inspect dns preset_dns_map
  inspect ftp
  inspect h323 h225
  inspect h323 ras
  inspect rsh
  inspect esmtp
  inspect sqlnet
  inspect sip
  inspect skinny
  inspect icmp
  inspect http
!
service-policy global_policy global
no call-home reporting anonymous
call-home
 profile CiscoTAC-1
  no active
end
```

Listing B.2: Firewall fw3 Configuration for Proposed University

```
! ASAv Config generated by NePAS
!
hostname fw3
username cisco password cisco privilege 15
enable password cisco
passwd cisco
names
!
interface GigabitEthernet0/1
  description to uniR
  duplex full
  nameif nepas-outside
  security-level 0
  no shutdown
  ip address 20.0.3.6 255.255.255.0
interface GigabitEthernet0/2
  description to wifi1
  duplex full
  nameif nepas-outside-1
  security-level 0
  no shutdown
  ip address 20.0.5.8 255.255.255.0
interface Management0/0
 description OOB Management
 duplex full
 management-only
 nameif mgmt
 security-level 100
 no shutdown
 ! Configured on launch
 no ip address
```

```
access-list nepas-in extended permit tcp host 20.0.1.3 host 20.0.0.2
   eq 65432
access-list nepas-out extended permit tcp host 20.0.0.2 eq 54321 host
    20.0.1.3 eq 54321
access-list nepas-out extended permit tcp host 20.0.0.2 host www.
   facebook.com eq 80
access-list nepas-out extended permit tcp host 20.0.0.2 host www.
   facebook.com eq 8080
access-list nepas-out extended permit tcp host 20.0.0.2 host www.
   gorillavid.in eq 80
access-list nepas-out extended permit tcp host 20.0.0.2 host www.
   gorillavid.in eq 8080
access-list nepas-out extended deny ip host 20.0.0.2 host 4.5.6.7
access-list nepas-out extended deny tcp host 20.0.0.2 host 20.0.1.3
   eq sftp
access-list nepas-out extended deny udp host 20.0.0.2 host 20.0.1.3
   eq sftp
access-list nepas-in extended permit tcp host 20.0.1.3 any eq 40728
access-list nepas-in extended permit tcp host 20.0.1.3 any eq 3689
access-list nepas-out extended deny tcp host 20.0.0.2 any eq ftp
access-list nepas-out extended deny tcp host 20.0.0.2 any eq telnet
access-list nepas-in extended permit tcp any host 20.0.0.2 eq ssh
access-list nepas-out extended deny tcp any host www.facebook.com eq
   80
access-list nepas-out extended deny tcp any host www.gorillavid.in eq
    80
access-list nepas-out extended deny tcp any host www.facebook.com eq
   8080
access-list nepas-out extended deny tcp any host www.gorillavid.in eq
    8080
access-list nepas-any extended permit icmp any any
access-group nepas-out out interface nepas-outside
access-group nepas-in in interface nepas-outside-1
access-group nepas-any global
!
same-security-traffic permit inter-interface
logging enable
logging asdm informational
user-identity default-domain LOCAL
aaa authentication ssh console LOCAL
http server enable
http 0.0.0.0 0.0.0.0 mgmt
ssh 0.0.0.0 0.0.0.0 mgmt
telnet 0.0.0.0 0.0.0.0 mgmt
http 0.0.0.0 0.0.0.0 nepas-outside
ssh 0.0.0.0 0.0.0.0 nepas-outside
telnet 0.0.0.0 0.0.0.0 nepas-outside
http 0.0.0.0 0.0.0.0 nepas-outside-1
ssh 0.0.0.0 0.0.0.0 nepas-outside-1
telnet 0.0.0.0 0.0.0.0 nepas-outside-1
ssh version 2
crypto key generate rsa modulus 768
telnet timeout 15
console timeout 0
username cisco password cisco privilege 15
!
class-map inspection_default
 match default-inspection-traffic
```

```
!
!
policy-map type inspect dns preset_dns_map
 parameters
  message-length maximum client auto
  message-length maximum 512
policy-map global_policy
 class inspection_default
  inspect ip-options
  inspect netbios
  inspect rtsp
  inspect sunrpc
  inspect tftp
  inspect xdmcp
  inspect dns preset_dns_map
  inspect ftp
  inspect h323 h225
  inspect h323 ras
  inspect rsh
  inspect esmtp
  inspect sqlnet
  inspect sip
  inspect skinny
  inspect icmp
  inspect http
!
service-policy global_policy global
no call-home reporting anonymous
call-home
 profile CiscoTAC-1
  no active
end
```

B.2 CDX Chapter Experiment

Listing B.3: Central Firewall Device Configuration for Chapter 6 Proposed CDX

```
! ASAv Config generated by NePAS
!
hostname cfw
username cisco password cisco privilege 15
enable password cisco
passwd cisco
names
!
interface GigabitEthernet0/1
  description to Team B-R
  duplex full
  nameif nepas-Team B-R
  security-level 0
  no shutdown
  ip address 20.0.9.2 255.255.255.0
interface GigabitEthernet0/2
  description to Team A-R
  duplex full
  nameif nepas-Team A-R-1
```

```
  security-level 0
  no shutdown
  ip address 20.0.3.2 255.255.255.0
interface GigabitEthernet0/3
  description to MgtRouter
  duplex full
  nameif nepas-MgtRouter-2
  security-level 0
  no shutdown
  ip address 20.0.4.2 255.255.255.0
interface GigabitEthernet0/4
  description to Attack-Router
  duplex full
  nameif nepas-Attack-Router-3
  security-level 0
  no shutdown
  ip address 20.0.5.2 255.255.255.0
interface Management0/0
 description OOB Management
 duplex full
 management-only
 nameif mgmt
 security-level 100
 no shutdown
 ! Configured on launch
 no ip address
access-list nepas-mgt-out extended permit ip host 20.0.0.1 any
access-list nepas-mgt-in extended permit ip host 20.0.1.1 host
    20.0.0.1
access-list nepas-mgt-in extended permit ip host 20.0.2.1 host
    20.0.0.1
access-list attack-own-out extended permit ip host 20.0.8.2 host
    20.0.1.1
access-list attack-opp-out extended permit ip host 20.0.8.2 host
    20.0.2.1
access-list attack-opp-out extended permit ip host 20.0.6.2 host
    20.0.1.1
access-list attack-own-out extended permit ip host 20.0.6.2 host
    20.0.2.1
access-list attack-opp-out extended permit ip host 20.0.7.2 host
    20.0.1.1
access-list attack-opp-out extended permit ip host 20.0.7.2 host
    20.0.2.1
!
same-security-traffic permit inter-interface
logging enable
logging asdm informational
user-identity default-domain LOCAL
aaa authentication ssh console LOCAL
http server enable
http 0.0.0.0 0.0.0.0 mgmt
ssh 0.0.0.0 0.0.0.0 mgmt
telnet 0.0.0.0 0.0.0.0 mgmt
http 0.0.0.0 0.0.0.0 nepas-outside
ssh 0.0.0.0 0.0.0.0 nepas-outside
telnet 0.0.0.0 0.0.0.0 nepas-outside
http 0.0.0.0 0.0.0.0 nepas-outside-1
ssh 0.0.0.0 0.0.0.0 nepas-outside-1
```

```
telnet 0.0.0.0 0.0.0.0 nepas-outside-1
http 0.0.0.0 0.0.0.0 nepas-outside-2
ssh 0.0.0.0 0.0.0.0 nepas-outside-2
telnet 0.0.0.0 0.0.0.0 nepas-outside-2
http 0.0.0.0 0.0.0.0 nepas-outside-3
ssh 0.0.0.0 0.0.0.0 nepas-outside-3
telnet 0.0.0.0 0.0.0.0 nepas-outside-3
ssh version 2
crypto key generate rsa modulus 768
telnet timeout 15
console timeout 0
username cisco password cisco privilege 15
!
class-map inspection_default
 match default-inspection-traffic
!
!
policy-map type inspect dns preset_dns_map
 parameters
  message-length maximum client auto
  message-length maximum 512
policy-map global_policy
 class inspection_default
  inspect ip-options
  inspect netbios
  inspect rtsp
  inspect sunrpc
  inspect tftp
  inspect xdmcp
  inspect dns preset_dns_map
  inspect ftp
  inspect h323 h225
  inspect h323 ras
  inspect rsh
  inspect esmtp
  inspect sqlnet
  inspect sip
  inspect skinny
  inspect icmp
  inspect http
!
service-policy global_policy global
no call-home reporting anonymous
call-home
 profile CiscoTAC-1
  no active
```