STAKEHOLDER SECURITY ANALYSIS – A NEW APPROACH TO SECURITY DESIGN WITH EXAMPLE APPLICATION

A Thesis submitted by

Nabeel Mahdy Hadaad Hadaad

For the award of

Doctor of Philosophy

2021

Abstract

Stakeholder security analysis (SSA) is a rigorous approach to analysing and designing systems from the point of view of cybersecurity which is defined and applied in this dissertation. SSA starts by identifying the *objectives* of the stakeholders, and then seeks to find rules which can be enforced to ensure that these objectives are met. It is shown by several detailed examples in this dissertation, and proved theoretically, by means of Hilbert's thesis, that first order logic is able to express any mathematical model and correctly explains the concept of logical proof; and that stakeholder security analysis can be used systematically to design secure systems. The relationship between the different cybersecurity rules is illustrated by means of inference graphs, which show how the rules which are enforced ensure that the objectives are met.

Chapter 1 provides an introduction, background, and presents outcomes of research significance. Chapter 2 reviews the relevant literature on the philosophy of security design that is applied to the application areas of web security, network security, and emergency networks. Chapter 3 defines stakeholder security analysis, including its theoretical justification, by means of Hilbert's thesis, and explains the use of *inference graphs*, which were developed as part of this research. *Service protection rules* are defined, in this chapter, as rules which, without appearing to define or ensure security, are nevertheless essential because they ensure that a service fulfills its objectives. Examples of these are provided in subsequent chapters, where it becomes clear that unless this type of rule is included, the system being designed is

logically incomplete. In Chapter 4, stakeholder security analysis is applied to web services, and, in particular, to the Netml system for network analysis, design and simulation. It is used to design and prove the security of certain aspects of the system.

In Chapter 5, the design of network filters and firewalls is considered, together with the security implications of virtual private networks. The use of simulation for security analysis of networks is explored practically, and the capability and limitations of simulation as a tool for security analysis of networks are investigated, using stakeholder security analysis as a rigorous framework that underpins all the proposed methods. It is shown that simulation can be rigorously used to prove the consistency of policies, and the sense in which simulation is able to prove the validity of cybersecurity is identified. In Chapter 6, the stakeholder security analysis is applied to emergency networks. The purpose of emergency networks is to save lives. The possibility of misuse and attacks upon an emergency network is also considered. A key consideration in the management of power for the devices which form the network. Five experiments concerned with the management of battery life to save lives in emergency situations are presented. Conclusions are presented in Chapter 7.

Certification of Thesis

This Thesis is entirely the work of **Nabeel Mahdy Hadaad Hadaad** except where otherwise acknowledged. The work is original and has not previously been submitted for any other award, except where acknowledged.

Student and supervisors signatures of endorsement are held at the University.

Principal Supervisor: Assoc Prof Ron Addie.

Associate Supervisor: Prof Yan Li.

Acknowledgments

By the Name of Allah, the Most Gracious and the Most Merciful At the outset, I would like to thank the Allah of creation, for everything because they made to me owner strength and guidance made me able to have the courage and faith and patience to complete this study. First, my sincere thanks and gratitude to my supervisor dear ASSOC.PROF.RON ADDIE for the support and guidance and keen on clarify and help me to take the right steps to build the project. My appreciation to him for his insightful remarks, valuable comments and ideas for all these years. He has provided me excellent guidance to work and develop critical thinking abilities. He taught me many other things apart from technical matters.

I would like to express my gratitude to PROF. YAN LI to support, motivation, encouragement, advice during my study time. My thanks go to Iraq Government for my sponsors for giving me the opportunity to complete my PhD study.

Offer deep thanks to my family in general, and to my brothers and sisters, especially my mother and my wife and dedicate my thanks to my kids: HUSSIN, HUDA, NADA, ZANIAB and ABBAS. Offer my thanks and appreciation to all of my friends who helped me in Australia and in Iraq with sincere thanks and appreciation to each of the fatigue for the tutorial to my professors respected and USQ staff, in University Southern Queensland, especially, DR SHAHAB ABDULLA. My sincere gratitude and appreciation go to the Australian Government for Research Training Program

(RTP).

Praise be to Allah

Nabeel Mahdy Hadaad Hadaad

Associated Publications

Hadaad, N., Drury, L. & Addie, R. G. (2015), Protecting services from security misconfiguration, in Telecommunication Networks and Applications Conference (ITNAC), 2015 International, IEEE, pp.120–125.

Hadaad, N., Pitsillides, A., Kolios, P., Kuras, A. & Addie, R. G. (2016), Emergency network design-saving lives by saving power, in 2016 26th International Telecommunication Networks and Applications Conference (ITNAC), IEEE, pp.19–21.

Sheniar, D., Hadaad, N., Martin, D., Addie, R. & Abdullah, S. (2018), Experiments and proofs in web-service security, in 2018 28th International Telecommunication Networks and Applications Conference (ITNAC), IEEE, pp.1–6.

Sheniar, D., Hadaad, N., Addie, R.(2019), The Inference Graph of Cybersecurity Rules, in 2019 29th International Telecommunication Networks and Applications Conference (ITNAC), IEEE, pp.3–10.

Table of Contents

List of Figures

List of Tables

List of Examples

List of Experiments

List of Listings

Acronyms & Abbreviations

ICT	Information and Communication Technology
DAC	Discretionary Access Control
MAC	Mandatory Access Control
RBAC	Role-Based Access Control
OMNET++	Optical Micro-Networks Plus
NS-3	Network Simulator 3
D.JSIM	Java-based simulation
NGFW	Next-generation Firewall
UTM Firewall	Unified Threat Management Firewall
ECN	Emergency communication networks
API	Application Programmer Interface
SOE	Standard Operating Environment
SSA	stakeholder security analysis
NL	Natural language
D2DC	device to device communication
NGFWs	next-generation firewalls
ACL	Access Control List
Vis4Sec	visualization for security

Chapter 1

Introduction

1.1 General Overview

This thesis introduces the new concept of *stakeholder security analysis* as a general method for analysing cybersecurity issues in networks, and ICT systems in general, and designing solutions to the cybersecurity problems which arise in them.

Stakeholder security analysis is defined and explained in Chapter 3, and then applied to web services in Chapter 4, to networks in Chapter 5, and to emergency networks in Chapter 6.

In all three application areas, what makes the approach of stakeholder security analysis effective is that rules are included which define and protect the service experienced by users. Cybersecurity practitioners often neglect such rules because they do not appear to be *about security*. These are termed *service protection rules* (Hadaad et al., 2015).

In the analysis of web services, our objectives include that, *users can create a new*

user identity and password associated with a specific email address, and *users can access the services associated with the user account by providing their password*. Such rules are so obvious that they are overlooked, but doing so undermines the logical analysis of the system being considered.

In the network security application area, from the viewpoint of stakeholder security analysis, our goal is to guarantee a set of rules which describe satisfactory behaviour and performance. It is already common to express network security by means of rules. Some well-known examples of rules are: (i) firewall rules, e.g. as in IPTables; and (ii) access control rules. A less traditional rule is the one typically included in the service level agreement for users: "Users should not pass on their username or password to anyone else".

Defining security by rules is a practical approach to safety, and potentially provides also a sound basis for a theory of network security. The term rule is used informally, for example, in relation to the rules to be observed by users and formally, for example, the rules embedded in systems that are enforced by the configuration of servers and routers. This research assumes that security is specified by formally defined and systematically enforced rules. Defining security by rules (sometimes called policies, or agreements) has recently emerged as an approach for defining and designing security (Bauer et al., 2002; Schneider, 2000).

In this dissertation, any network used to manage the rescue and recovery of survivors trapped in a disaster or emergency site will be termed an *emergency network*. Emergency networks can significantly help to reduce the severity of trauma and loss of life by more effective communication between emergency workers, volunteers, and survivors. Typically, almost all of these parties hold mobile phones (in addition to special-purpose wireless communication devices, in the case of emergency workers), which are likely to be used intensively to seek and to provide aid to those in need.

The utility of mobile phones may, however, be limited by battery life. The usefulness of these mobile phones can, therefore, be enhanced by extending battery life. One of the ways to extend battery life is to introduce a duty cycle.

The use of a common language avoids conflicts and inconsistency for defining access control policies between all network devices (Sabelfeld and Myers, 2003). A candidate language for this purpose has been defined as XACML (OASIS, 2010). It is discussed in more detail in Section 2.4 of Chapter 2. We use ns-3 and Click in simulations to check the consistency of an aggregate security policy by checking that service protection rules are valid, as shown in Chapter 5. Also, we use R in simulations to check the consistency, as shown in Chapter 6. This study shows that these can improve the performance of the network experienced by users and increase network security.

1.2 Research Problem

In the security of networks and computer systems, there is often a poor linkage between the requirements analysis and the design. Supposing, for the moment, that it is enough to say that requirements are specified by rules, there is no clear algorithm for finding a design that meets those rules rigorously, effectively, and efficiently. Developing such an algorithm, or procedure, is a major undertaking. However, the components of this algorithm will be outlined below, in the introduction, rigorously defined methods will be specified in Chapters 3–5, and these are illustrated by examples in Chapters 3–6.

All systems have *stakeholders* (Reed et al., 2009; Maguire et al., 2012), and it can be easier to visualise the rules a system must satisfy if these rules are classified according to which stakeholder's requirements they meet. This approach will be developed later

in Chapters 3, 4, 5, 6.

This thesis considers the unforeseen consequences of security rules, in preventing legitimate use, and how to avoid these unexpected consequences from occurring. There are many challenges (Caulfield and Pym, 2015) to overcome to achieve correctness and consistency of security rules: First, due to complexity, some rules have errors on web service, network security, and emergency network. The increasing size of networks or the size of web service inevitably leads to increased complexity. In particular, inconsistent rule matching between firewalls can result in illegitimate traffic being allowed into the network, leading to serious security threats (Hamed and Al-Shaer, 2006; Lupu and Sloman, 1999).

Second, devices from different vendors (Mayer et al., 2000) may require different methods of configuration. Third, a great variety of different actions (e.g., bypass, discard, encrypt/tunnel, authenticate/transport) need to be envisaged when analyzing network security policies (Hamed et al., 2005). In web service, Sometimes, users can not access the services they need. When this occurs users may feel entitled to use insecure methods to obtain the services they need, which introduces risks that personal information of the user and overall security of the system could be compromised.

In some cases, an emergency network fails to provide assistance to survivors because they have lost communication with central emergency control, for example, survivors cannot send text messages or call if the battery of their mobile phone has died. When a disaster or emergency occurs, the survivors, emergency professionals, and helpers will use their mobile phones to enable rescues to occur and assistance to be provided (Hossain et al., 2019). In effect, their phones form an *emergency network*. It remains unclear how best to manage and use this informal emergency network.

The research undertaken in relation to emergency networks in this dissertation seeks

to extend the time during which a mobile device can be used to rescue and recover survivors by carefully designed power management.

1.3 Research Objective

The main objective of this study is to create a philosophy and strategy which enables us to design and manage ICT security through the use of rules. In doing so, the following sub-objectives will be accomplished first.

1. To introduce and investigate a new philosophy of security design, namely that security is defined by the rules which stakeholders operate by and require, and that a security design is *valid* if and only if (a) its rules are consistent and (b) the rules which are enforced logically imply the remaining rules (Hadaad et al., 2015; Sheniar et al., 2018).

   Stakeholder analysis takes into account all those who have influence on decisions, events, or outcomes related to the system and also those who are influenced by such decisions and outcomes (Wu et al., 2018; Rose, 2013).

2. To determine the necessity in ICT Security of using rules which define the services provided. This type of rule might not be used in the implementation, for example (Hadaad et al., 2015).

3. To apply the methodology to web services. This methodology applies particularly well to all web services.

4. To apply the methodology to network security. Network security is one of the most important of all applications of cybersecurity, and since networks continue to be a source of problems, it is of special interest to find more rigorous methods that can be used in this area (Hadaad et al., 2015).

5. To investigate the use of simulation for validation of network security. In particular, network security and the performance needs of users will be investigated by Netml (Addie et al., 2011b), ns3 (Riley and Henderson, 2010), Click (Kohler, 2001) or other tools for modeling and analysis of networks. Simulation is not necessarily the best tool for analysis of an ICT security model. It has the advantage that it is readily understood by a wide audience, and existing tools are available and can be used. If other tools are thought to be needed, it probably is necessary to develop them.

6. To evaluate a model of network security and performance by means of ns-3 (Riley and Henderson, 2010), Click (Kohler, 2001; P. and Merz, 2011), or other network analysis tools.

7. To apply the methodology to emergency networks. This is an excellent application for the security methodology because it is quite different from the other examples considered (Hadaad et al., 2016).

1.4 Outcomes of research

This dissertation introduces a philosophy of design for ICT security (Hadaad et al., 2015; Sheniar et al., 2018). It defines a model of security for networks, emergency networks (in particular) and web services that includes a new type of rule which we call *service protection rules*. These are additional to the traditional rules which are concerned with controlling risk. Service protection rules list all the qualities of the services which are to be provided, irrespective of whether they concern "security". More specifically, the research in this project has achieved:

1. Stakeholder security analysis (SSA) is a new methodology for cybersecurity architecture which was introduced in (Hadaad et al., 2015; Sheniar et al., 2018)

which has been defined and explained in Chapter 3 and applied to web services in Chapter 4, to networks in Chapter 5 and to emergency networks in Chapter 6.

2. Inference graphs were introduced in (Sheniar et al., 2019). They are defined in Section 3.4 of Chapter 3. They are used in Subsection 4.2 of Chapter 4, Section 5.4 of Chapter 5, and Section 6.3 of Chapter 6. They are shown, in examples, how we can prove that objectives are achieved by enforcement of other rules, which is then illustrated by an inference graph.

3. Service protection rules were introduced in (Hadaad et al., 2015). They are defined in subsection 3.3.3 of Chapter 3. They are used in examples on web service in Subsection 4.2.3 of Chapter 4, network security in Section 5.3.3 of Chapter 5, and emergency network in Subsection 6.3.3 of Chapter 6.

4. The stakeholder security analysis is applied to the emergency network in Section 6.3 of Chapter 6.

5. The dynamic duty cycle for smartphone and devices taking is designed when use an emergency network in Subsection 6.4.1 of Chapter 6.

.

1.5 Scope of the study

Networks provide a great variety of services to users. Security is often seen as inevitably reducing the convenience of access. For example, it is a common practice, supposedly for the purpose of increasing security, for managers of servers to be forced to adopt elaborate and inconvenient management procedures. Without a simple, elegant explanation of what constitutes ideal security design, it is difficult to refute

the idea that inconvenience is a necessary byproduct of tight security. This research aims to show that in some cases, at least, first-rate network security does not have to compromise convenience or speed of access to any services.

1.6 Structure of the dissertation

This dissertation has seven chapters: Chapter 1 is an introduction to the study, which presents research problems, research objectives, scope, and Outcomes of research significance. Chapter 2 presents the background information of security design and reviews of the published literature. Chapter 3 introduces the stakeholder security analysis. Chapter 4 explains web service security design, including the design of a password reset system with SSA to prove that its objectives are met. Chapter 5 presents network security design, rules of security design, and some examples to use these rules with SSA to improve network security and reduce problems. Chapter 6 explains emergency scenarios, emergency network design, stakeholder security analysis (SSA), and Power management of smartphones. Chapter 7 provides research conclusions and future work with suggested recommendations for further work.

Chapter 2

Literature Review

2.1 Rules in cybersecurity

In (Caulfield and Pym, 2015), construction of executable models is examined, for example, and the policies of the organizations, which describe the organization's objectives are made up of *rules*.

According to (Koppel et al., 2016), the perceptions of general users and cybersecurity professionals expressed dissatisfaction with security rules. These researchers identified misunderstandings and misdirected approaches to achieve improved security. This study is a step towards improving user behaviour and cybersecurity policies and then developing better security rules.

It appears, therefore, that the security of networks, and of ICT systems in general, can be expressed, analysed, and designed by means of *rules* (statements which make assertions about the system considered). This idea is investigated further in this chapter, by examination of the literature on cybersecurity, and is further developed in the subsequent chapters of this dissertation by experiments.

2.1.1 Network security requirements

According to (Stallings and Brown, 2015, §1.1), there are three security objectives for information and for information systems: confidentiality, integrity and availability. Let us consider each of these objectives, as specified by Stallings and Brown, in more detail. The definitions from (Stallings and Brown, 2015, §1.1) are as follows:

- Confidentiality: this term covers two related concepts:
  - Data confidentiality: assures that private or confidential information is not made available or disclosed to unauthorized individuals.
  - Privacy: assures that individuals control or influence what information related to them may be collected and stored and by whom and to whom that information may be disclosed.
- Integrity: this term covers two related concepts:
  - Data integrity: assures that information and programs are changed only in a specified and authorized manner.
  - System integrity: assures that a system performs its intended function in an unimpaired manner, free from deliberate or inadvertent unauthorized manipulation of the system.
- Availability: assures that systems work promptly and service is not denied to authorized users.

Later in the same section Stallings and Brown expand the definition of availability as follows: *Ensuring timely and reliable access to and use of information. A loss of availability is the disruption of access to or use of information or an information system.*

Even with this clarification of the availability objective, it is clear that Stallings and Brown view availability as primarily concerned with preserving the *normal* service of the network in the face of security risks.

2.1.2 Security of networks and security rules

Another widely adopted convention is to express network security by means of rules. This is a practical approach to security, and potentially provides also a sound basis for a theory of network security. In (Jasser, 2019), it is asserted that security is often taken into account *too late* in the process of software engineering and development. Architectural rules for security can support software architects and developers in consciously taking security into account during the design and implementation phase. Architectural security rules are identified, in (Jasser, 2019), through expert interviews.

In this dissertation, we generally assume that security is specified by *formally and systematically defined* rules. The task of network security designers is to choose these rules and implement procedures that enforce them. A key idea to explore in this dissertation is how to improve the design of network security, emergency, and web services by adding some rules whose purpose is to protect the service experienced by users. These *service protection rules* are in addition to the conventional rules which are concerned with protecting data from attack. That is to say, these rules can't logically be put into any of the three categories of Stallings and Brown, as discussed in §2.1.1.

2.2 Stakeholders and security analysis

2.2.1 Stakeholders

The paper (Diver, 2007) advises that *stakeholders* should be involved in development and review of security policy, where (Diver, 2007),

> "Stakeholders would typically include anyone who is a user of the information or systems covered by the policy."

According to (Rose, 2013), stakeholders play a key role in any project success, and managing stakeholder interactions is more than a matter of excellent communication. The processes are: (i) identify stakeholders; (ii) develop a stakeholder management plan; (iii) manage stakeholder engagement; and (iv) control stakeholder engagement. Their approach not only improves stakeholder management but also enables better emphasis on communications by focusing on the collection of project information.

2.2.2 Stakeholder security analysis (SSA)

According to (Reed et al., 2009), stakeholder analysis means different things to different people. Various methods and approaches have been developed in different fields for different purposes. Stakeholder analysis processes include, according to (Reed et al., 2009): (i) identifying stakeholders; (ii) differentiating between and categorising stakeholders; and (iii) investigating relationships between stakeholders.

The paper (Samonas et al., 2020) explains the convergence and divergence of stakeholder perceptions about security policy. Moreover, it highlights the practical utility of *the Repertory Grid Analysis* in helping information security researchers and managers pinpoint: (i) the aspects of a security policy that are well-received by stake-

holders; and (ii) the variance in the perceptions of stakeholders. It can capitalize on the well-received aspects of the policy.

The paper (Zinsmaier et al., 2020) proposes and applies a requirements approach that focuses on the security, privacy properties, and takes into account different stakeholder interests. The proposed methodology facilitates the integration of security and privacy by design into the requirements engineering process. Thus, specific, individual security and privacy requirements can be implement from the very beginning of a software project the. The approach includes the application of threat and risk rating methodologies, a technique to derive technical requirements from legal texts (Zinsmaier et al., 2020).

According to (Hanauer et al., 2018), visualization for security (Vis4Sec), a process framework for the generation and distribution of stakeholder-specific visualizations of security metrics, which assists in closing the gap between theoretical and practical information security by respecting the different points of view of the involved security report audiences. The initiation of Vis4Sec consists of the following steps (Hanauer et al., 2018): (i) the environment is analyzed; (ii) the requirements are stated; (iii) the stakeholders are defined; (iv) the actions are planned.

The paper (Faily, 2015) introduces an approach that assumption personas are used to engage stakeholders in the elicitation and specification of security requirements at a late stage of a system design. The methodology of the author has devised an approach for developing assumption personas for use in participatory design sessions during the later stages of a system design. The author validates this approach using a case study in the e-Science domain (Faily, 2015).

According to (Flechais and Sasse, 2009), the threats to resources from online attacks require robust and effective security to mitigate the risks faced. They raise two issues ensuring that: (i) the security mechanisms put in place are usable by the different

users of the system; (ii) the security of the overall system satisfies the security needs of all its various stakeholders. A failure to address either of these issues can seriously jeopardise the success of e-Science projects. However, these case studies highlight the importance of involving all stakeholders in the process of identifying security needs and designing secure and usable systems (Flechais and Sasse, 2009).

SSA identifies the stakeholders, the goals of the stakeholders, and then finds rules that can be enforced which ensure that their objectives are achieved (Sheniar et al., 2018, 2019).

SSA is carried out by a cybersecurity architect or designer; hence, least initially, it has been done in natural language, i.e., the rules which state the objectives are expressed in English; also, the enforced rules are expressed in English. When enforced rules are enforced by computer systems, they need to be expressed in computer languages. Natural language, when used to express cybersecurity objectives and rules, can easily be translated or interpreted, as formal logic, by introducing *predicates* in place of the natural language verbs and associated language constructs. Examples in which this is done occur in Chapter 5. The choice of language in which to express cybersecurity rules is considered in more detail in §2.4.

Stakeholder security analysis is explored in more detail in Chapter 3. Chapter 4 shows how SSA is used to make web services more secure and to provide effective and efficient services. In Chapter 5, the design of network security is shown to benefit from SSA, and in Chapter 6, SSA is applied to emergency networks, which leads to some valuable and interesting insights.

2.3 Simulation and Modeling Tools

The paper (Breslau et al., 2000) explains that network simulation allows researchers to test scenarios that are difficult or expensive to implement in the real world.

The paper (Aggarwal et al., 2020) suggests that simulation is particularly useful to test new networking protocols or for changes the existing protocols in a controlled and reproducible environment. Using simulation, it is possible to design and test different networks with various types of nodes (hosts, hubs, bridges, routers and mobile devices) and transmission equipment, whether or not they currently exist.

This study discusses five different modeling tools: ns3, Omnet++, D.Jsim, Packet Tracer, and Petri.NET. Of these tools, the first four are network simulation tools, and the last tool is a logical modeling and analysis tool; Omnet++, ns3, and Petri.NET are open source while D.Jsim and Packet Tracer are commercial products.

2.3.1 The Netml system

The Netml system (Addie, 2010) was developed at the University of Southern Queensland for teaching and research into network protocols and technology. Its objective is to enable students and users from industry to create networks quickly and to understand the full complexity of a multilayer network easily by means of highly configurable visualisation tools. It is freely available for use (in the cloud) at `http://netml.org`.

According to (Addie et al., 2006), The Netml system is used in this dissertation to construct network examples and simulate them. Simulations in Netml are carried out by generating an ns-3 C++ program and together with Click scripts for each firewall. The main tools provided in the Netml system are analytic and do not rely on

simulation. However, simulation is an essential technique in research and teaching. More than, half of all users introduced to a system for network analysis assume that simulation is the only possible way to analyse a network. Even users who understand that there are effective mathematical models and for analysis expect the reassurance of confirming simulations to validate them.

In (Addie et al., 2011b), the Netml system provides services for analysis, design, and implementation of networks. These services are provided by means of a web site; no software is installed on users' computers (except in the form of cached javascript). This system is used in teaching, by computer science students, and in research into network analysis and design. Most of the users of the Netml system do not use it for an extended period of time and, therefore, do not develop complex networks that incorporate a significant investment. However, the security requirements of users of the system, and its owners, are nevertheless important.

2.3.2 ns-3 and Click

According (Riley and Henderson, 2010), ns-3 is written in C++ and has Python scripting capability. To ease the creation of basic facilities and define their inter-relationships, ns-3 has a system of containers. The ns-3 simulator is a discrete-event network simulator targeted primarily for research and educational use. The ns-3 project started in 2006, and is an open-source project. The ns-3 system provides accurate and fast simulation of communication systems, with emphasis on the TCP/IP protocols. The range of technologies, protocols, and applications which may be of interest to include in simulations is very considerable, and it is, therefore, essential to facilitate modular extension of any simulation system.

In (Kohler, 2001; P. and Merz, 2011), in particular, the ns-3 system does not include it's own native model of routers or router protocols, but instead can model

routers using the Click modular router system, or to use other router implementations, including commercial software. Ns-3 is able to include a mixture of simulation, emulation, and implementation of networking software and hardware, including interfacing with software running in virtual machines.

2.3.3 Optical Micro-Networks Plus Plus(OMNET++)

OMNeT++ is not strictly a network simulator, but rather a general-purpose, component-based, modular, and open framework for discrete event-based simulation. However, according to (Weingartner et al., 2009), this tool is usually applied to the field of networks simulation, to know the truth with its INET package. It provides a comprehensive collection of Internet protocol models. OMNeT++ simulations consist of so-called simple modules which realize the atomic behavior of a model. In the paper (Varga, 2010), it is an extensible, modular, component-based C++ simulation library and framework, primarily for building network simulators. The most common use of OMNeT++ is for simulation of computer networks, but it is also used for queuing network simulations and other areas as well (Siraj et al., 2012).

2.3.4 Java-based simulation (D.JSIM)

D.JSIM is a Java-based simulation for building models that are difficult to check them in fact and analyzing them to extract experimental data (Gupta et al., 2013). J-Sim is an environment for application development, which depends on components of software architecture. This framework is built upon the autonomous component architecture (ACA) and the extensible internetworking framework (INET) of J-Sim. J-Sim has been developed by a team at the Distributed Real-time Computing Laboratory (DRCL) (Kumar et al., 2012).

2.3.5 Packet Tracer

Packet Tracer is a simulation program (Zhang et al., 2012), that allows users to simulate network behavior, and Cisco network academy uses it free. In the education field, by use of this tool, students can access different network devices such as router, switch, and other devices. Also, it plays a role in teaching students about complex technology concepts. Ciscos systems claim that Packet Tracer is useful for network experimentation. It can also be used for collaboration and to simulate the operation of the network. Packet Tracer supports a multi-user system that enables multiple users to connect various topologies by a computer network (Trabelsi and Saleous, 2019).

2.3.6 Petri Nets

Petri.NET is one of simulation tools (Thong and Ameedeen, 2015) is a tool that can be used for modeling, simulation, and analysis for components behaviour of Networks. Also, it can be freely used to simulate manufacturing systems, and discrete event systems. Petri nets designed a sequence of modules, with each module containing a single data element and communicating with its two neighbors. Users can use Petri nets to simulate any system which they can create described graphically like flow charts that need some mean of representing activities. Since Petri nets can be applied to most systems to characterize them graphically, users can model and check their system by Petri net tools in detail and analyse the logical correctness and performance of their systems. By using Petri net tools, users can analyze the performance of a system and use it as a graphical editor and code generator (Baez et al., 2019).

2.3.7 Discussion

This dissertation has used exclusively ns-3 in conjunction with the Netml user-interface. The concept of traffic is not supported explicitly in most simulation systems, but is a key feature of the Netml/ns-3 system. Ns-3 itself does not support the traffic concept but instead provides *sources* and *applications*, which can be used together to implement traffic. In a Netml/ns-3 simulation, a report can be generated, which shows whether traffic is carried or not. From this traffic report, it is usually straightforward to see whether the intended rules are satisfied or not. For this reason, and because any desired new features for Netml/ns-3 can be developed in-house, the Netml/ns-3 system was adopted for modeling and simulation of networks in this research.

2.3.8 Use of R to simulate lifetimes

In this paper (Kaya et al., 2019) defines that R program is open source *free software for statistical computing and graphics*. It can run on different systems, for example, a variety of UNIX platforms, Windows, and MacOS.

According to (Beaujean, 2013), there are three advantages for R. First, it is a free and open source. Secondly, it is a programming language that is expressive and efficient for quantitative analysis. Finally, it allows users to submit their own packages to another user through the R server (`https://cran.r-project.org/`).

This program has been in this research to determine a lifetime to survivors in an emergency situation, by using some diagrams to explain how this research can extend the battery life of a smartphone.

2.4 Languages

2.4.1 XACML

The use of a common language for defining access control policies between all network devices in order to avoid conflicts and inconsistency (Sabelfeld and Myers, 2003). A candidate language for this purpose has been defined XACML (OASIS, 2010).

XACML was investigated as a language for defining security rules in (Addie et al., 2011a), and it was shown there that it lacks some of the features of logic which are required for defining the rules needed in stakeholder security analysis. In essence, this shortcoming of XACML appears to be due to the perception that security rules are only about expressing *restrictions* or *constraints*, but, as shown in (Hadaad et al., 2015), this study also needs to express some rules which express *objectives*.

2.4.1.1 The use of formal rules for defining ICT security

In the literature on ICT security, it is sometimes implied that security is either fully, or partially, defined by formal rules, recorded in a language such as XACML (OASIS, 2010). The precise semantics of how security can be managed in this way is, however, unclear, and there is currently no widely accepted industry standard for the management of formal rules even for access control alone.

2.4.2 Logic

The model proposed in (Zhang, 2005; Zhang et al., 2004, 2005, 2008) is a network (nodes, links, and traffic) together with rules (service level agreements, firewall rules, and access control rules) associated with each element. The association of rules with

network elements was previously discussed in (Abadi et al., 1993). These rules are formal statements, expressed in a rigorous form. Mathematical logic has been used in the analysis of security for many years, for example, (Manin, 1977; Wolf, 2005). The requirement of consistency might be all that is necessary to ensure that a collection of rules is valid (Cori and Lascar, 2000; Glasgow et al., 1992). Gödel's completeness theorem tells us how to check consistency. Intuitively, it says "to prove consistency is equivalent to finding a model" (Cohen, 1966). In many cases, a model is equivalent to a simulation (Hadaad et al., 2015).

This dissertation uses mathematical logic to express the rules associated with networks and web service systems. Since it is the traditional tool for formal expression and reasoning about statements, in general, it is also applicable to the rules used to define security. The entire collection of rules needs to be consistent (Guelev et al., 2004). It is usual to assume that individual stakeholders associated with a network or system being analysed have made sensible choices of the rules they wish to be enforced, and that they are willing to abide by.

2.4.3 Natural language (NL)

Natural language (English) can be used to express logical statements needed for design. This language should be formal, so that computers can read it, but it should also have an informal version. Some aspects of access control cannot be formalised (Badger et al., 1995; Shi and Chadwick, 2011), and all aspects need to be understood by humans as well as by computers.

Is it necessary to adopt a universal standard language for expressing security?

Translation of natural language into logic can be obvious. Examples: see later Chapters 3, 4, 5, 6 (Karjoth and Schunter, 2002; Zhang et al., 2008; Hadaad et al., 2015,

2016).

2.5 Application areas

This section reviews the literature on the three application areas to which SSA has been applied in this dissertation, namely, web services, network security, and emergency networks.

2.5.1 Web security

According to (Ranchal et al., 2019), enforcement of access control policies and preventing unwanted data leakage in composite web services is a challenge due to: (i) Inability of the client on the selection of services in an orchestration; (ii) Vulnerabilities caused by improper implementation of access control in web services; (iii) Insufficient options for the client to specify their access control policies; (iv) Improper communication of the clients access control policies by the services in an orchestration.

According to (Erdogan, 2009), web applications provide convenient access to information and services. However, their vulnerabilities have increased steadily in recent times. They suggest that an important software security practice, which addresses this, is security testing, to reduce the increasing number of vulnerabilities.

In the paper (Addie and Colman, 2010), it is suggested that lack of a satisfactorily standardised secure access model may be the critical road-block preventing wider deployment and use of web services. They suggest five criteria for web-services security: (i) Policy-sufficiency is defined as the requirement that any meaningful statements can be expressed in policy definitions of the architecture; (ii) Protocol neutrality is

the requirement that a protocol exchange which is logically equivalent to a valid protocol sequence is also valid; (iii) Predicate boundedness is the constraint that a fixed, finite set of predicates (or language constructs) will be sufficient for security policy definitions, i.e., the language does not need to be incrementally extended indefinitely; (iv) Protocol-closure requires that security protocols can be combined arbitrarily to make new protocols; (v) Processing complexity constrains algorithms for evaluating security rules to be of satisfactory (low) complexity.

The methodology of stakeholder security analysis investigated in this dissertation is evaluated according to these criteria in Section 3.1 of Chapter 3.

According to (Addie et al., 2011a), in formulating security architectures for web services, there appears to be a conflict between: (i) The need for expressive power in expressing policies; (ii) Computational simplicity in access algorithms; and (iii) A natural desire to use the same language for policies and access rules. Researchers provide five examples of security in web services, which illustrate this tension. These examples highlight the need for more expressiveness in the rules used to express policies in three cases, and in the other. A tentative solution to this problem is to define two different languages, one for defining policies, and another for defining algorithms for access.

The paper (Bhardwaj and Goundar, 2018) explains that attackers scan the environment, seek vulnerable points, and analyse the attack surface. To solve these problems, they recommend that we should reduce the threat surface to minimise the impact of cyberattacks. For example, a web application server can have a reduced attack surface by reducing the available web resources. Moreover, reducing the threat security score if it reduce the exposed surface area. This is achieved by: (i) Identifying attack surface resources to measure; (ii) Calculate a surface area impact score – a low score indicates good coverage; (iii) Reducing surface area and incentives for

cyberattack; (iv) Redesign the architecture and review the security score; (v) review the surface area security score. The design is reviewed and design modifications are performed on the application architecture.

2.5.2 Network security

According to (Hamed and Al-Shaer, 2006), there are many challenges to overcome to achieve the correctness and consistency of security rules. For example, rules complexity, different vendors, and different actions (e.g., bypass, discard, encrypt/tunnel, authenticate/transport). These authors claim that the deployment of a network security system requires a global analysis of policy configurations of all network security devices. They address these problems and to avoid policy conflicts and inconsistency use analysis of policy configurations.

The paper (Zhang et al., 2008) claims that what has long been neglected by much previous work is the analysis and detection of security holes in policies caused by interactions of rules, co-operations between agents and multi-step actions. It is not enough to know: (i) whether a single rule behaves correctly, but that all rules, working together, behave correctly; (ii) what a single agent can do by herself, but what a set of agents can achieve through co-operations, including perhaps overwriting each others privileges; (iii) what an agent can do in a single action, but what he can achieve through a sequence of actions, especially when agents can change permissions to give themselves or others privileges.

The paper (Hamelin, 2010) shows that security networks face problems due to complex rule configurations that routers and other devices are required to monitor and maintain continuously. They claim that focusing efforts on the right firewall at the right time can mitigate risks before they become a problem. Also, it prevents network security from experiencing a meltdown.

Just as standard firewalls are difficult to manage, according to (Erdheim, 2013), next-generation firewalls (NGFWs) have their own set of challenges. The need is increased greatly with NGFWs and their application control because of the complexity of having thousands of rule sets and the potential for errors.

The paper (JAIDI, 2019) explains that firewall rule bases may rise in complexity and size over time which can introduce misconfiguration problems, non-compliant rules, and excessive permissions. To address these issues, several researchers worked on a variety of approaches that allow: (i) detecting and removing firewall misconfigurations; (ii) conducting firewall policy reviews for compliance reasons; and (iii) analyzing the coherence of firewall rules.

This paper (JAIDI, 2019) introduces the novel concept of an FW-TR firewall which aims to (i) strengthen the quality of the firewall filtering service; (ii) facilitate the identification of firewall misconfigurations; (iii) reduce the complexity of the analysis of firewall rules for detecting existing anomalies; and (iv) configure the firewall to automatically change its behavior facing critical and malicious scenarios.

According to (Pozo et al., 2012), the design, development, and maintenance of firewall ACLs are very hard and error-prone tasks. Two of the reasons for these difficulties are: (i) the significant gap that exists between the access control requirements and the complex and heterogeneous firewall platforms. (ii) the absence of ACL design, development, and maintenance environments that integrate inconsistency and redundancy diagnosis. Also, two of the most important problems firewall administrators have to face are: (i) the high complexity of firewall-specific ACL development and maintenance; and (ii) ACL inconsistencies (contradictions), and redundancies introduced during these life-cycle tasks. These authors suggest a model-driven design, development and maintenance framework for layer-3 firewall ACLs to address these issues(Pozo et al., 2012).

This dissertation introduces a methodology – stakeholder security analysis, with service protection rules – which is able to address the challenges of security design discussed in the preceding papers. Stakeholder analysis and service protection rules are explored in more detail in Section 5.2 and Subsection 5.3.3 of Chapter 5, respectively.

2.5.3 Emergency networks

Emergency communication networks play an important role in saving lives and reducing loss of resources in emergencies and disasters (ETSI, 2014).

There are four important types of emergency communication (ETSI, 2014): authorities to authorities, authorities to citizens, citizens to authorities, and citizens to citizens.

Authorities to Authorities: this part of emergency communications involves government authorities communicating with each other, and with other agencies, for example, to create preparedness in advance of an emergency, or for the purpose of coordinating services during an emergency.

Authorities to Citizens: communications between authorities and citizen to address some problems arising during a national emergency. For example, an Emergency Alert System (EAS) is a national public warning system.

Citizen to Authorities: a person sends an emergency message to an appropriate authority via whatever service is available, for example, a public mobile phone network.

Citizen to Citizen: a person directly sends a message or calls another person when a disaster is happening.

According to (Whittaker et al., 2013), on Saturday 7 February 2009, 173 people lost their lives, and more than 2000 houses were destroyed in bushfires (wildfires) in the Australian State of Victoria. The authors of this report concluded that inadequate planning and preparedness and the tendency for people to wait until they are directly threatened before taking action were major factors leading to late evacuation, failed defence. Just over half of the respondents (53%) stayed to defend their homes and properties, whereas the remainder left before or when the fires arrived (43%) or sheltered in a house, structure, vehicle, or outside (4%). Results reveal a survival rate of (77%) for houses that were defended by one or more household members, compared to (44%) for unattended houses.

The final report (Queensland Government, 2012), into the 2010/2011 floods, in which more than 78 percent of Queensland was declared to be a disaster zone (Queensland Government, 2012, p32), stated that 33 people died, and another three were still missing (at the time the report was written). While flood-related fatalities occurred throughout the state, the highest concentration of deaths occurred in the Lockyer Valley area, where 16 people died and the three people who remain missing were lost. According to this report, one reason for the loss of life was the issue of *black spots, areas which are not covered by a radio communications network and within which radio communications are consistently difficult or impossible.* This means that communication is a weakness between the emergency centre and the survivors in this area.

In report (Coyle and Childs, 2005), the impact that the widespread availability of mobile phones has had in response to specific disasters and atrocities, such as the Indian Ocean tsunami, Hurricane Katrina, the summer floods in central Europe, and terrorist attacks in Istanbul and London. They observed that mobile phones could be used to send messages in preference to making calls, during an emergency. Text messages are more likely to get through (as they use less network capacity or can be

held in a queue and sent when there is free capacity), and their use also help ease congestion in the network.

Mobile phones can also play an important role in helping recovery from a disaster. For example, in the United States, both the Federal Communications Commission and the local operators have posted consumer advisories, telling customers to ensure their handset batteries are charged ahead of an emergency, to have a back-up battery, to keep their phones dry, and to expect the network to be busy in the aftermath of an event such as a hurricane (Coyle and Childs, 2005), Also, mobile phones tend to play a supplementary role in early warning systems. They can be a useful mechanism for individuals to relay that information on to friends and family who may have missed the initial broadcast. Mobile phones can help in the process of recovery because they uniquely give affected people and aid agencies the means to find and receive information specific to their needs.

2.5.3.1 Device to device communication (D2DC)

According to (Albalawi, 2019), 5G technology supports direct communication between two devices. By this technology, the survivor mobile can connect with another mobile near him. According to (Deepak et al., 2019), two users connect between them through (D2D) communication without need base station in the emergency area. Although device to device communication has the potential to be highly beneficial in emergencies, there currently does not appear to be any standard which could be described as a *language* for such communication. Since the efficiency of the communication might be critical to the effectiveness of D2DC in emergencies, this appears to be an important gap in the literature on emergency networks.

2.5.3.2 Power management

Power management is critically important for smartphones in emergency conditions if they are to fulfill their potential for saving lives and helping with recovery. Power management is very important in smartphone design for achieving the battery life that users need in their normal operation and has therefore received considerable attention from both researchers and manufacturers. The paper (Awal et al., 2018) reports the development and design of power-efficient client server architecture software by applying wake lock techniques that depend on statically defined power saving profiles to manage the smartphone features such as Wi-Fi, GPS, display brightness and background running applications. Each profile will have a predefined control on several features of android smartphones which include: turning off GPS, reducing the brightness level and dynamically shutting down of some background running applications to make optimal use of the battery life. In addition, the client-server (app) concept will be applied to collect usage information of clients and sending them to a remote server that generates the power saving profiles of the smartphones.

2.5.3.3 A Dynamic Duty Cycle

Emergency networks cannot provide assistance to survivors when they have lost communication due to the loss of stored battery power. In the rare and special situation of an emergency or a disaster, battery power management techniques of a more specialized nature, which restricts the service provided by a smartphone.

Dynamic duty cycle *is a processor in a smartphone, which enables it to switch off for a period of time. And then switch on again, for a short period, during an emergency. Hence, a dynamic duty cycle can extend battery life for a smartphone to improve their contribution to the saving of human lives.*

Power Nap is a mobile device operating system power management module (Olsen and Narayanaswarni, 2006). It utilizes the processor power states more efficiently and modifies the timing of certain tasks.

Stakeholder security analysis is applied to emergency networks in section 6.3 of Chapter 6, and in particular the design of a power management system for a smartphone in an emergency network, and in particular the design of a dynamic duty cycle are explored in more detail in Section 6.4 and Subsection 6.4.1, respectively, of Chapter 6.

2.6 Examples

Many examples of the cybersecurity methodology developed and applied in this dissertation are presented below. These are listed in the List of Examples.

Tables of rules for each example are presented below, in the examples, and these tables, objectives are distinguished by names like O1–O4, and enforced rules by names starting with "E", such as E1, EV1, etc.

2.7 Summary

This chapter reviewed concepts relevant to a new philosophy (stakeholder security analysis) of design of security in networks and web services and the literature on these concepts. Later chapters explain how to use these concepts to create new designs of the security for web services, for networks and emergency networks, by using new and different methods.

Chapter 3

Stakeholder Security Analysis

3.1 Introduction

The main result of this chapter is to introduce and describe such a methodical design philosophy, namely, stakeholder security analysis(SSA). SSA identifies the stakeholders, the goals of the stakeholders, and then finds rules that can be enforced which ensure that their objectives are achieved in web service, security network and emergency network (Sheniar et al., 2018, 2019).

For example, vulnerabilities of web systems take a great variety of forms, and new ones appear to emerge regularly, so it can seem an endless process to manage and maintain web site or web service security (Jøsang and Pope, 2005). The approach of searching for weaknesses and fixing them by use stakeholder analysis is so widely used that it might reasonably be regarded as a design philosophy by use SSA to achieve objectives.

3.2 Stakeholders

Stakeholders include all those who have an influence on decisions, events, or outcomes related to the system (Rose, 2013). There are several ways to classify stakeholders. The author of (Rose, 2013) discussed the roles of stakeholders by looking at the results obtained from stakeholder analysis and make suggestions for managing stakeholder participation.

Stakeholder analysis refers to a set of tools for identifying and describing stakeholder based on their relationships and interests related to this issue. Goals are identified by all stakeholders who have a relationship with or who influence the system. In addition, we seek to understand the changes, and challenges within the system in order to determine the best type of stakeholder communication and to define the relationships, changes and challenges in order to suggest the best management system (Maguire et al., 2012).

The paper (Savage et al., 1991) explains that the evaluation of stakeholders in cooperating with organizations and managing stakeholders appropriately, gives a positive result. In some cases, managers can use a comprehensive strategy to change relationships between stakeholders and transform them from unproductive groups into productive groups (Savage et al., 1991). Maynard, Ruighaver, and Ahmad assume that stakeholders in organizations should be involved in the Information security police ISP. They investigate the ISP development process to determine stakeholders roles in any organisation (Maynard et al., 2011). In (Bourne, 2016; Grimble and Wellard, 1997; Billgren and Holmén, 2008), some of the stakeholder roles depend on preventing threats potential (Almorsy et al., 2016; Scholl, 2005). Stakeholder analysis helps in determining: needs of Stakeholders and how security rules should be changed to make the system safer (Diver, 2007).

However, the most important reason for a careful stakeholder analysis, from the point

of view of this chapter, is that if we are able to identify a sufficient set of rules that ensure the willing participation of each stakeholder, and if we can enforce these rules, then the system is self-evidently secure. This does not rule out the possibility that through experience stakeholders may, during the lifetime of a system, discover that there are rules which were not initially obvious and which needed to be added to their required rule set. If sufficient care is taken with the stakeholder analysis, such events should be rare.

3.3 Definition of Stakeholder Security analysis

Definition 1 *Stakeholder security analysis is the process of identifying the objectives of all stakeholders, finding axioms, assumptions and conditions which can be enforced to ensure that these objectives must be true, and proving that the objectives follow from these conditions.*

Stakeholder security analysis (Sheniar et al., 2018) proceeds as follows:

1. Identify the key stakeholders.
2. For each stakeholder, identify a set of rules *required* by these stakeholders. Note: the collected rules required by all stakeholders must be consistent.
3. Implement procedures which ensure that all rules are enforced.

The *goal* of cybersecurity is to guarantee that certain objectives are maintained. For example, it is likely that a bank will have, as an objective, that no transactions – transfers of money from one account to another – occur except with valid authorization.

Cybersecurity *design* aims to discover or instantiate axioms, assumptions, and enforced rules which enable us to *prove* that the *objectives* are true/false. Along the way to doing this, there may be some *intermediate propositions* that we also wish to prove.

Thus, *objectives* and *intermediate propositions* have proof. On the other hand, *axioms* do not have proofs because these are fundamental truths that are true from logical principles or, in some cases, because their expressions follow from the definition of the predicates they contain. *Assumptions* are true by assumption (which might not always hold, but at present we adopt them), and *enforced rules* are true because we make sure, in the system, that they are true, so none of these rule types have proofs.

3.3.1 Theoretical justification of Stakeholder Security Analysis

If all the objectives of the stakeholders are met, then the system is valid. By Hilbert's thesis (Barwise, 1982), any system which can be described mathematically can be defined by a set of axioms expressed in symbolic logic:

> *(a) when one is forced to make all one's mathematical (extra-logical) assumptions explicit, the axioms of a theory can always be expressed in first-order logic (the Predicate calculus), and*
>
> *(b) the informal notion of* provable *in mathematics is made precise by the notion of* provable in first order logic.

Hilbert's thesis is not a theoretical observation. It tells us something both surprising, and yet very practical. When the rules of the agents participating in a workflow are fully expressed in logic, which will require us to introduce a number of predicates

that are used to explain everyday actions and conditions that arise in the course of the workflow, the semantics (the meaning) of all these predicates can be fully expressed by logic. This typically requires identifying rules (axioms) which are not associated specifically with one agent or another, but which, instead, *define* the predicates or, in some cases, express assumptions that we feel we must make, or which are *convenient* to make, as part of the workflow design. Hilbert's thesis, that any mathematical system can be defined in first order logic does not immediately apply to any web service or business system. However, with good reason, it is generally assumed that any real-world system can be modeled mathematically, so it follows that any real world system can also be modeled by first order logic.

3.3.2 Evaluation of stakeholder security analysis

Let us evaluate stakeholder security analysis as in Definition 1 by means of the five criteria for web service security architecture set out in (Addie and Colman, 2010), which was discussed in §2.5.1:

(i) *Sufficiency* – any system can be described; stakeholder security analysis places no limits on the rules which are allowed, except, in some cases, that they can be expressed in first-order logic (which is not really a constraint anyway), so this criterion is met.

(ii) *neutrality* – there is no bias toward one solution or another; stakeholder security analysis is not committed to any specialised techniques or constructs, so this criterion is met;

(iii) *predicate boundedness* – this is unclear and probably cannot be determined except by exploration of more examples.

(iv) *protocol-closure* – there are no limitations or constraints imposed by stakeholder security analysis, so this criterion is met.

(v) *complexity* – there are no limitations or constraints on choice of algorithms in stakeholder security analysis, so this criterion is not relevant to it. Stakeholder security analysis is a methodology for analysis, and some aspects of design, but does not impose too many conditions on implementation.

3.3.3 Service Protection Rules

Service protection rules are defined, hereby, as rules which, without appearing to define or ensure security, are nevertheless essential because unless they are included, a design which meets all other requirements might fail to ensure these objectives. They are the objectives which ensure that a service is provided, satisfac- torily, without having any direct connection with IT or security. Examples of these are provided in Chapters 4, 5, 6, where it becomes clear that unless this type of rule is included, the system being designed is logically incomplete.

A good example of a service protection rule occurs in networking: rules which ensure what services *must be provided* need to be included, when defining a firewall, or filtering. It is very common for firewall or filter maintenance to accidentally disable essential services (especially less prominent ones) because the concept that security is about *blocking unwanted traffic*, rather than allowing wanted traffic, is widespread. The firewall might not actually *use* such rules, but they can be used in testing to ensure that the firewall is properly configured.

3.4 Inference graphs

An inference graph (Sheniar et al., 2019) shows the relationship of *inference* (what implies what), which applies between the different cybersecurity rules which apply in a system. An example of an inference graph is shown in Figure 3.2, and this graph will be explained later in Example 3.1. The following types of rules arise in cybersecurity analysis: *objective*, *enforced rule*, *axiom*, *proposition*, and *assumption*. An objective is a rule which is required to be true at all times. An enforced rule is a condition which it is possible to implement, and which it has been decided to enforce, by technical means. For example, access to many systems is only provided if a user is able to enter a valid username and password.

An axiom is a condition which is held to be true *a priori*; an assumption is a condition which we *choose* to believe. For example, under some conditions, we assume that users do not reveal their password to other users. A proposition is a rule or statement which we define in order to express a useful stage of reasoning.

An *edge* in an inference graph connects each of the rules which are referenced in proof to the rule, which is proved. Objectives are typically the destination of edges, while enforced rules usually occur only as the origin of an edge.

The appearance of a reference to a proposition, assumption, axiom, objective, or enforced rule, in a proof, constitutes a relationship between that rule and the rule which is proved. The *inference graph* has vertices or nodes corresponding to the rules, and directed edges or links from any rule which is referenced in a proof to the rule which is proved. In all the inference graphs included in this dissertation, the different types of rules are represented by nodes of different shapes and colours, as indicated in the Legend, shown in Figure 5.2.

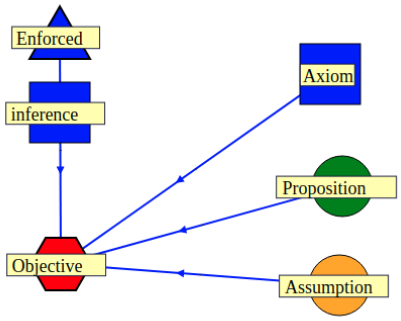

Figure 3.1: Legend for Inference graphs

3.5 Example of Stakeholder Analysis

Stakeholder security analysis is best explained by means of examples, so a simple example is now presented. More complex examples will be given in Chapters 4–6.

Example 3.1 A parcel box

Consider a box for delivering parcels: which can be opened only by the owner and those delivering parcels. The following example was first published in (Sheniar et al., 2019)

The objectives of the system as a whole are:

- O1: parcels cannot be stolen (taken from the box by someone other than the owner) from the parcel box.
- O2: parcels can be retrieved from the parcel box by the owner of the box.
- O3: deliverers are able to store delivered goods in the box whenever they visit the property with an item to be delivered.

To achieve these objectives, O1–O3, we use enforced Rules E1–E7, as defined in Table 3.1, and assumption A1, which is also defined in Table 3.1.

3.5.1 Proofs of objectives for Parcel Box

Proof of O1

Because of E4, parcels can only be stolen by someone with a code. But, by E3, codes are only sent to the parcel box owner and a deliverer. By A1, the owner does not pass on their code to anyone else. Furthermore, by E1, the code sent to the owner is sent by a secure path, and by E2, the code sent to the parcel box is sent by a secure path also, so, together, these conditions make it impossible for anyone except the owner to have the code sent to the owner. By definition, if the owner uses their code, it is not stealing, so the only remaining way for the parcel to be stolen is if a deliverer steals it or passes on their code to someone else. But, by E6, a code sent to a deliverer can not be used to open a box unless it is empty, so deliverers can't steal the parcel, and neither can anyone who receives a code sent to a deliverer. □

Proof of O2

Because E1 is true, we can suppose the owner has the code for the parcel in the box. In addition, E5 holds so the owner can open the parcel box. This proves O2. □

Proof of O3

To prove O3, we need to be sure that deliverers receive a code. This is ensured by E3. By E7, deliverers visit the parcel box when it is empty. By E6, the parcel box can be opened by a deliverer who has the code. □

From these proofs, we can conclude that the inference graph for these rules is as shown in Figure 3.2.

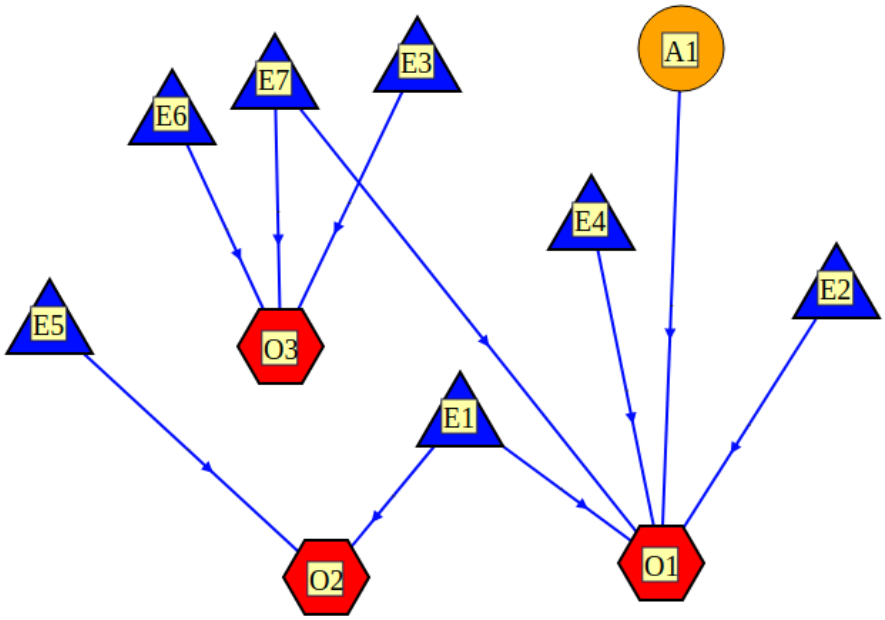

Figure 3.2: Inference graph of rules for a parcel box

This example illustrates the way in which assumptions can and should influence design of a system. If we assume that deliverers can be trusted implicitly, there is no need to ensure that deliverers can only open a parcel box when it is empty. If, on the other hand, we do not make this assumption, the system will need to ensure that deliverers cannot open a parcel box which contains a parcel. This approach is more

secure but a little more difficult to manage. In the past, assuming that deliverers can be implicitly trusted would not seem unreasonable, but in future, as the range of delivery options increases, such an assumption might come to seem unnecessary and unrealistic. □

3.6 Summary

In this chapter, the concepts of stakeholder, stakeholder roles, and stakeholder rules were introduced. Also, the definition of Stakeholder Security Analysis is achieved.

The needs of stakeholders were analysed as their *objectives*, which take the form of *statements about the system* which they require to be true. A system which ensures that all the objectives of its participants are met, self-evidently, thereby ensures the willing participation of its stakeholders and can be regarded as secure.

A list of rules stated as the requirements of each of the stakeholders are defined as enforced rules were E1–En and assumptions are A1–An. Then *objectives* and *intermediate propositions* have proofs and achieved by use stakeholder rules analysis. We assume cybersecurity design was improved when all objective be achieved.

Inference graphs shows the relationship between different cybersecurity rules in a system. Also, types of rules in cybersecurity analysis are *objective*, *enforced rule*, *axiom*, *proposition*, and *assumption* and use link between these rules. According to nodes and links in this graph, we can prove all objectives.

An example – of a parcel box with careful cybersecurity design – was used to illustrate the concepts introduced in this chapter.

Stakeholder security analysis (SSA) is explored in this chapter. SSA has been applied

in this dissertation, namely, web services, network security, and emergency networks. Also, more detail in Chapter 4 shows how SSA used to make web services more secure and to provide effective and efficient services. Moreover, in Chapter 5, the design of network security is shown to benefit from SSA, and in Chapter 6, SSA is applied to emergency networks, which leads to some valuable and interesting insights.

Table 3.1: Rules for a parcel box

Rule Name	**Details**
O1	Parcels cannot be stolen (taken from the box by someone other than the owner) from the parcel box.
O2	Parcels can be retrieved from the parcel box by the owner of the box.
O3	Deliverers are able to store delivered goods in the box whenever they visit the property with an item to be delivered.
A1	The owner of the parcel box does not allow access to the codes they receive to anyone else.
E1	Codes are sent by a secure path to the owner of the parcel box.
E2	Codes are sent by a secure path to the parcel box.
E3	Codes for access to the box are generated and stored on the server in a system which does not provide read access to any person, agent, or process, except for a process which sends them to the owner of the parcel box, and to deliverers, and this process cannot be used to send the codes to anyone else.
E4	Parcels cannot be removed from a locked box without a code for opening it.
E5	Parcels can be removed from a locked box by anyone with the code for opening associated with the parcel it contains
E6	The parcel box can be opened by the code sent to a deliverer, when, and only when, it is empty.
E7	Deliverers are scheduled to visit the parcel box only when it is empty.

Chapter 4

Web Service Security Design

4.1 Web Service Security

Because web services (including services provided via apps on mobile phones) are a recent development and continue to evolve in both details and fundamentals, principles of secure design of these services is also a new and evolving area of research and development (Addie and Colman, 2010; Addie et al., 2011a; Sheniar et al., 2018).

This section reviews three different approaches for securing web sites/services. Each of these approaches is usually expressed as a completely independent philosophy for achieving good security. These approaches are actually complementary, and to achieve rigorous security all three approaches are needed. Note that although we describe a design philosophy which is able, formally, to prove, i.e. guarantee, security, because no logical system can claim certainty in an absolute sense (in mathematical logic, this fact is expressed in Gödel's incompleteness theorem), the strategy of attacking the system remains useful, even after it has been methodically proved to be correct.

The present chapter does not apportion equal emphasis on all approaches because the original contribution of this chapter is in the third of them, together with the way the second approach joins with the third to form a more comprehensive whole. The second approach is the one summarised in Subsection 4.1.2. The third approach is summarised in Section 4.2 and applied to the Netml password reset system in Sections 4.2 and 5.5.6.

4.1.1 Good Security Design Practice

Good design takes security, ease of access, and usability into account, striking a balance between protecting the system and ease of use. Good practice has evolved a number of practical approaches like minimizing attack surface area (Bhardwaj and Goundar, 2018), establish secure defaults (Lai et al., 2018), using the principle of defence in depth (Toch et al., 2018), not trusting services (Ghirardello et al., 2018), keeping simple security (Thomsen and Bertino, 2018), and fixing security issues correctly (Ali and Alaa, 2018; Tabassum et al., 2018). These approaches are used for maintaining and improving security which they are so natural and important that they should be adopted as a first layer of protection as a matter of standard practice, even when more sophisticated approaches are also in use (Ross et al., 2018).

4.1.2 Security Auditing

Strategies for breaking into web systems or services are under continuous development by government and non-government organisations and individuals, both those with friendly intentions and those who wish to exploit security weaknesses for their advantage. When a new exploit is discovered, if it is discovered first by those with friendly intentions, defences against the exploit are usually developed quickly and published. Exploits discovered by attackers with ill intent can, of course, be de-

ployed before web managers have the opportunity to defend against them. Also, in the period of time immediately after the defence against a new exploit has been published, there is still an opportunity to attack web sites which have not deployed the newly developed defences. This time can be somewhat extended due to the limited expertise of web-site owners and because the sequence of steps required to address a weakness in a high-level framework can be quite lengthy.

A widely used strategy for improving web site or web service security is to attempt to attack the site by using the strategies which are currently known to be effective one-by-one, or simultaneously, to discover if the site is vulnerable to any of these strategies. Since all of the strategies tried are known, the defences against all of them are also almost certainly known, and hence can be adopted by the web site.

4.2 Stakeholder security analysis of a web service

In this section, the stakeholder security analysis is applied to a particular web service, namely the Netml system (Sheniar et al., 2018, 2019), however, it will be clear how the methods apply to other web systems.

A more fundamental strategy which is not well-developed at present is to seek to develop provably secure protocols and software for all aspects of a web service (Whitman and Mattord, 2011; Mailloux et al., 2018; Bishop, 2005).

The first step in this approach, which is developed further in Section 4.2 is to consider the point of view of all legitimate stakeholders in relation to the service, and to enumerate a complete set of rules required by each of these stakeholders, sufficient to ensure that they agree to participate actively in the service. Both a security audit and a stakeholder security analysis are applied in this chapter to a specific subsystem of a web service system being developed and managed by the authors.

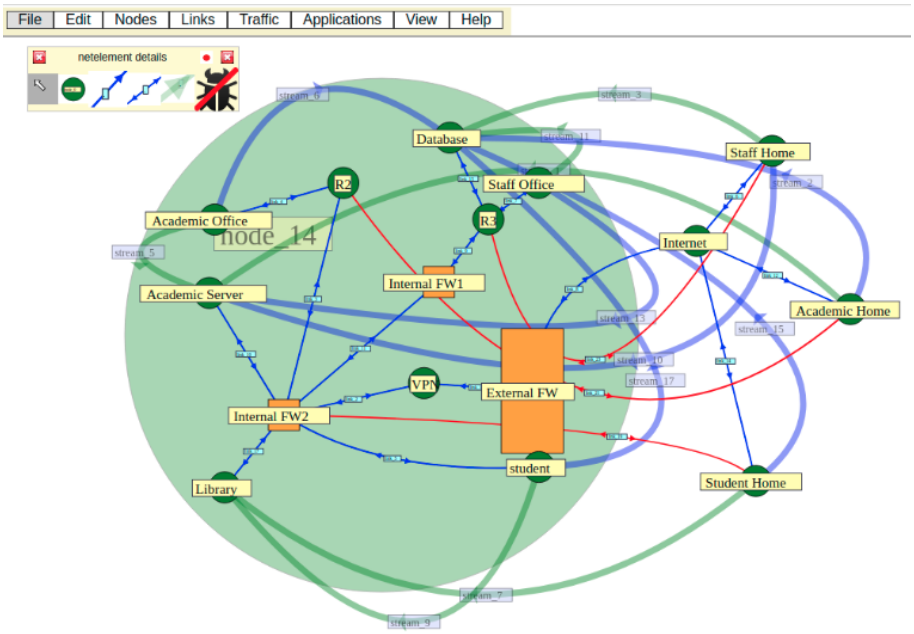

Figure 4.1: Netml system to apply network with firewall

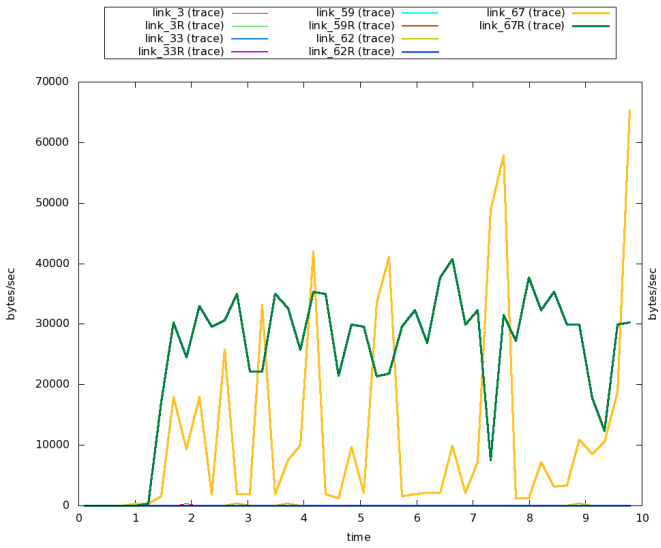

Figure 4.2: Traffic traces

Example 4.1 The Netml system

The Netml system provides services for analysis, design, and implementation of networks (Addie et al., 2011b). These services are provided by means of a web site, for example, as shown in Figure 4.1. No software is installed on users' computers (except in the form of cached javascript). This system is used in teaching, by computer science students, and in research into network analysis and design. For example, a plot from Netml website shown in Figure 4.2. Most of the users of the Netml system do not use it for an extended period of time and therefore, do not develop complex networks that incorporate a significant investment. However, the security requirements of users of the system, and its owners, are nevertheless important.

A key requirement of this system is that users can readily create their own accounts and can reset their password if necessary when they have forgotten it. Users can share networks that they create or load with each other, but by default users cannot access the data of other users.

4.2.1 Netml stakeholder roles

Stakeholder roles in the Netml system and examples of stakeholders are listed respectively in Table 4.1. The study considers four stakeholder roles in this chapter.

User

Anyone can become a user. Users are permitted to access and change the information they have stored including information about their identity, such as their email address and password.

Table 4.1: Netml stakeholder roles

Stakeholder role	**Example of stakeholder**
User	Teacher or Student
IT Admin	The System Administrator, IT Department personnel
Netml Admin	An administrator of the Netml system
Guest	Visitor of Netmal system
Attacker	External attacker,internal attacker(administration member)

Admin

Admin users are able to access all the same services and information as ordinary users and, in addition, are able to see a variety of reports which are not available to ordinary users. There are different *levels* of admin access, and at a sufficiently high admin access level, data stored by other users is visible. This is a feature which is convenient for users when they need help with their use of the system, although it may be desirable under some circumstances to allow users to have privacy from admin users as well as from other users. Administration role in this system consists of Individuals that can play the role of attackers.

Guest

Guest users are able to access, and run algorithms on, networks which are publicly accessible.

Attackers

Although attackers have no inherent rights, it is useful to consider the objectives and motivations of attackers to better understand the strategies most effective in thwarting them. In particular, this role includes who can achieve a successful attack on the password reset system. As mentioned previously, the attacker may be an administrative member.

4.2.2 Netml stakeholder rules

In this subsection, a subset of the rules for two of the key stakeholders have been identified and are shown in Tables 4.2 and 4.4. Rules are classified as objectives, enforced rules, and assumptions.

□

4.2.3 Service protection rules

Objectives *O1, O2, O3, O6, and O7* from Table 4.2 and Table 4.4 are service protection rules for web services. These rules ensure users to access their services easily without break web services security. Although these rules are "obvious", without them, the definition of these services is incomplete, and the process of designing the security cannot be methodically completed. The design of a service requires all rules to be proved; if all rules are proved, the web services work security (Hadaad et al., 2015). For example, we will use A1, E4, and E5 to prove Objective O3 in Subsection 4.2.4.

Example 4.2 A password reset system

Table 4.2: Netml user objectives

Rule name	**Explanation**
O1	Users can create a new user identity and password associated with a specific email address.
O2	Users can access the services associated with the user account by providing their password.
O3	Users are able to reset their password by receiving a ticket by email and using this ticket at the website within 30 minutes, assuming that they provide an incorrect ticket at most twice.
O4	Agents/persons other than a valid user cannot use this system to reset the password of a user.
O5	Users can not change the password of a user other than themselves.

Table 4.3: Netml user assumptions

Rule name	**Explanation**
A1	Password reset tickets are sent to each user correct email address and are received by email within 1 minute.
A2	Only persons/agents who know a user email password can access email sent to the user within the last 30 minutes.
A3	The user password cannot be guessed.
A4	The user is the only person/agent who knows their email password.
A5	The possibility of an attacker intercepting a ticket on the user email client, during its 30 minutes of valid life is negligible.
A6	The possibility of guessing a ticket in fewer than 10,000 attempts is negligible.
A7	The server administrator never acts in a way to subvert the intentions of the system he/she administers.
A8	User does not disseminate the supplied email address that contains the Password notification email.
A9	User should not reveal their Netml password to anyone else, or store it unsafely.
A10	User cannot obtain another user password.

Table 4.4: Administration objectives

Rule name	Explanation
O6	Admins can ensure that password problems are only resolved after adequate user identification.
O7	Admins can enable users to access the right resources at the right times and for the right reasons.
O8	Admins can not access to the inbox of the account user.

Table 4.5: Objectives for the password reset subsystem

Rule Name	Explanation
O3	Users are able to reset their password by receiving a ticket by email and using this ticket at the website within 30 minutes, assuming that they provide an incorrect ticket at most twice.
O4	Agents/persons other than a valid user cannot use this system to reset the password of a user.
O5	Users cannot change the password of a user other than themselves.

Consider a portal that maintains accounts for users of its services and which has a *password reset service*, such as discussed in (Sheniar et al., 2019). Rules from Tables 4.2, 4.3 and 4.4 which refer to the password reset subsystem are listed in Tables 4.5, 4.6, and 4.7. Focussing on this subset of rules, finding the proofs of the objectives, and the inference graph for this subsystem enables us to reduce the complexity of the analysis task.

```
<!-- this is the case of updating details of an existing user -->
<c:if test="${(! empty param.resetpwd) and
↪  (param.ticket==target_ticket)}">
<sql:update dataSource="jdbc/MySQLDB">
UPDATE USERS set name=?, user_name=?, organisation=?, email=?,
↪  password='', user_pass=?
where user_name= ?
<sql:param value="${param.fullName}"/>
<sql:param value="${username}"/>
<sql:pastakeholdersram value="${param.organisation}"/>
<sql:param value="${param.useremail}"/>
<sql:param value="${hexshapass}"/>
<sql:param value="${username}"/>
</sql:update>
...
</c:if>
```

Listing 4.1: Code for changing passwords

4.2.4 Proofs of objectives for a password reset subsystem

Proof of O3

To prove O3, we assume A1 *users receive an email that includes a ticket within 1 minute*, and once they have a valid ticket, by E4, they can use this ticket to set their password on the web site. In addition, by E5, users can generate a ticket and have it sent to their email address by some action on the web site. □

```
/**
* Returns a random salt to be used to hash a password.
*
* @return a 32 bytes random salt
*/
privatestakeholders static final Random RANDOM = new
↪  SecureRandom();
public static byte[] nextSalt() {
byte[] salt = new byte[32];
RANDOM.nextBytes(salt);
return salt;
}
/**
*
* @return a random Hex string of length 64 bytes
*/
public static String ticket() {
StringBuilder sb = new StringBuilder();
byte[] salt = nextSalt();
for (int k=0; k<salt.length; k++) {
sb.append(String.format("%02x",salt[k]));
}
return sb.toString();
}
```

Listing 4.2: Algorithm for tickets

```
                <!-- this is the case of updating password of an existing user
                ↪  -->
                <c:if test="${! empty param.resetpwd &&
                ↪  (param.ticket!=target_ticket
                or param.minutes>(ticket_minutes+30) or
↪  param.date!=ticket_date)}">
                <jsp:forward page="login.jsp" >
                <jsp:param name="errorMsg"
                value="An invalid or out-of-date ticket was used to reset a
                ↪  password. Please try again." />
                </jsp:forward>
                </c:if>
```

Listing 4.3: Algorithm for checking ticket timeliness

Proof of O4

By E6, supplying a valid password reset ticket is the *only* method to reset a password. Hence, to prove O4, it will be sufficient to show that an attacker cannot generate or gain access to a valid password reset ticket.

By E7, only the password reset service can generate a password reset ticket.

We should now consider two possibilities: (i) the attacker generates the password reset ticket themselves, and: (ii) the attacker intercepts a password reset ticket generated by the user being attacked.

Case (i)

By A1, any password reset ticket, generated by any method, will be sent to the correct of user email whose password is being reset, within 1 minute. By A2–A4, the attacker cannot intercept the email message with the reset ticket. By E8 and A7, it

Table 4.6: Assumptions for a password reset system

Rule Name	Explanation
A1	Ticket sent to users are received by email within 1 minute.
A2	Only persons/agents who know a user's email password can access email sent to the user within the last 30 minutes.
A3	The user's password cannot be guessed.
A4	The user is the only person/agent who knows their email password.
A5	The possibility of an attacker intercepting a ticket on the user's email client, during its 30 minutes of valid life is negligible.
A6	The possibility of guessing a ticket in fewer than 10,000 attempts is negligible.
A7	The server administrator never acts in a way to subvert the intentions of the system he/she administers.

is also not possible for an attacker to access a reset ticket on the server. Guessing the reset ticket is ruled out by A6, and E1-E3. Hence, the attacker cannot access the reset ticket and change the user's password in this case.

Case (ii)

In this case, also, by A2–A4, A7, A6, E1-E3, and E8, the attacker cannot access the email containing the reset ticket, and therefore cannot change the user's password. □

Proof of O5

The proof of O5 is also applicable in this case because nowhere is the fact that the attacker is *not* a user-relevant in the proof of O4. □

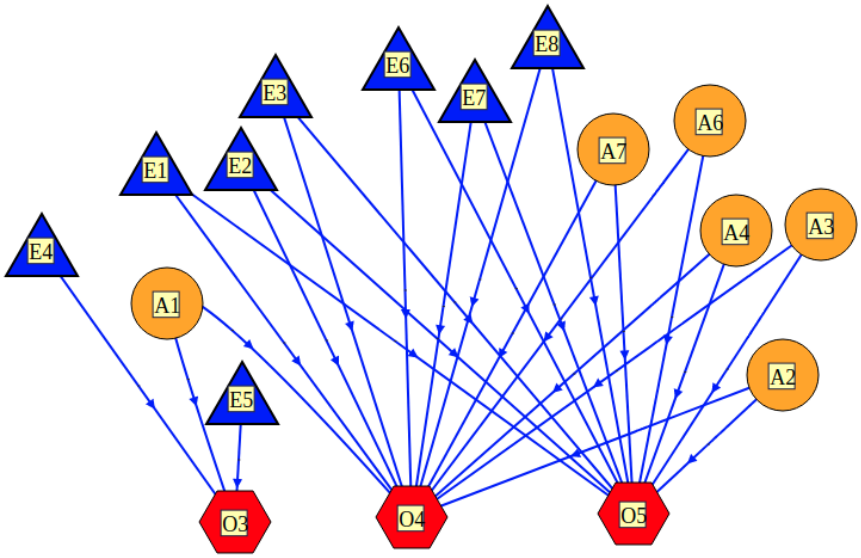

Figure 4.3: Inference graph of rules for a password reset system

The key requirements for this system are O1 and O2 in this table, i.e., the service can be used to change passwords, but users can only change passwords for *their own* accounts. Such systems are vulnerable to logical errors, as discussed in (Sheniar et al., 2019). An inference graph for this system, which can be used to guide its validation, is shown in Figure 4.3. □

4.3 Summary

In Section 4.1, three complementary approaches for achieving rigorous web service security were reviewed. The first of these is a pragmatic list of good practices in Subsection 4.1.1 that help to minimise the effort required to create and maintain good security. If these practices are not followed, the other two approaches, which are more specific in addressing security requirements, will require too much effort to be put into practice. The second approach is to attempt a series of attacks by methods which are known to be currently active and to use the known (and usually published) techniques to address them. These methods are used by attackers as they are easy to get the worked attacked vector, and they know they will work as the uptake of fixes and patches is slow.

The third approach is to apply stakeholder security analysis, as described in the previous chapter. The objectives of stakeholders are identified, and then the steps which are taken to address these objectives are formulated sufficiently clearly, and in such a way, that the stakeholder objectives are provable. This makes a critical connection between the code implementing a web service and the security rules it is expected to conform to. It is not necessary to adopt a special-purpose programming language, or methodology, in order to achieve this level of rigour (although adopting special methods may make this task easier).

Table 4.7: Enforced rules for a password reset system

Rule Name	**Explanation**
E1	Use of an invalid ticket to reset a password three times or more in 10 minutes, for a certain user, causes any outstanding tickets for that user to become invalid.
E2	tickets sent to users who are resetting their password are valid for at least 30 minutes and at most 31 minutes.
E3	When a ticket is successfully used, it will not be valid in future.
E4	If a user supplies a valid ticket, at the service web site, they can use this site to set their password to a new setting, without knowing the existing password.
E5	Any user can cause a ticket to be generated and sent to the user's previously registered email address, by appropriate actions at the service web site.
E6	The *only way* to reset a password is by supplying a valid password reset ticket.
E7	The *only way* to generate a password reset ticket is by the password reset service.
E8	Password reset tickets cannot be accessed by any user other than the administrator, on the server where they are generated.

Chapter 5

Network Security Design

5.1 Introduction

Network and ICT security is traditionally viewed as the protection of the services and resources of an organisation from unauthorized modification, unauthorized access to information, and from network activity which interferes with operations. However, a broader definition has recently emerged (Hadaad et al., 2015), which is that security is the practice of defining and enforcing appropriate *policies* which define the allowed behaviour of all participants in the target organisation (Schneider, 2000; Bauer et al., 2002).

There are many challenges to overcome to achieve the correctness and consistency of security rules. For example rules complexity (Erdheim, 2013), different vendors (Mayer et al., 2000) and different actions (e.g., bypass, discard, encrypt/tunnel, authenticate/transport) (Hamed et al., 2005; Hamelin, 2010). These problems explained early in section 1.2. One approach which has been suggested to overcome these difficulties is the use of a common language for defining access control policies

between all network devices to avoid conflicts and inconsistency(Sabelfeld and Myers, 2003; Pozo et al., 2012). A candidate language for this purpose has been defined (Jajodia et al., 2001; Fong and Siahaan, 2011), but it has not been widely adopted (Zhang et al., 2008).

In this chapter we investigate the role of *logical consistency* of the complete set of rules defining the security of a network. This study shows that consistency is of both theoretical and practical relevance to the greater aim of *validating* security rules. The consistency is a requirement of the collection of security policies of any organization. Also, investigating consistency may reveal and correct errors of some rules of web service, network security, or emergency network. Moreover, conflict between rules can lead to unpredictable outcomes. Note: logical consistency means that there are no logical contradictions. It does not mean that the rules stay the same over time, or that they are the same for everyone. The research shows that logical consistency can be checked by simulating a network and show how these simulations can be conducted, both in examples and by providing an online tool in which all these examples can be viewed in full detail, including re-running the simulations to check rules. Also, add other rules to improve security in the future. See Table 5.11.

This tool can be used to simulate, and validate the consistency, of a wide variety of networks. We fully investigate the limitations of consistency, and its validation by simulation, and show that these limitations are not sufficient to prevent this approach from being used to validate security of networks.

The remainder of the chapter is organized as follows. In Section 5.2, present Stakeholder security analysis, describe stakeholder roles, and who are stakeholders?. In Section 5.3, we define service protection policies and discuss introducing a common language between devices of a network. In section 5.4, several example problems are considered with partial solutions in some cases, including internal filtering and

firewalls, virtual private networks, printer access, and single-sign-on. In Section 5.5, we discuss validation of security policies and a formal correspondence between simulation models. Also, models in the sense of mathematical logic and uses this to show that simulations are able to prove the consistency of security policies. In section 5.6 describes examples of Validation by Simulation. In section 6.6 gives some summary about problems of access service and how they can fix them.

5.2 Stakeholder Security Analysis

The idea of *Stakeholder Security Analysis* in Chapter 5 is an extension for Chapter 3 by which is meant that we should identify the stakeholders, then the goals of the stakeholders, then find rules that can be enforced, and then choose which of these rules to enforce, and, finally, prove that all the rules of the stakeholders can be proved to be true whenever the rules which are enforced are true. The stakeholders of a network should be spelled out. In all the examples, either there is no specific business, or, in some cases, the network owner is a university. Hence the stakeholders should be described fairly generically; There are *four stakeholder roles*:

- Network owner(s) are university or any organisation.
- Authorized clients (users) of the network are all persons use university service or company service.
- Non-authorized users (the general public) are all people who do not have authorization or not be part of working in these universities or organisations.
- Attackers are all persons who use illegal methods to obtain services in universities, organisations and companies.

Once the stakeholders have been defined, naturally, they need to determine their

objectives, generically. In the specific examples, the objectives will be defined more specifically also.

5.3 Security Policies

Information security police ISP protects within organisations. Set of rules which determine who is authorised to access service in organisations or companies. This access should under conditions which authorised is allow or block users to use these services.

5.3.1 Firewall and filtering rules

A firewall is a system that checks incoming and outgoing packets to ensure only "safe" traffic is allowed through it. Firewalls can work in several of ways: A packet-filter examines each packet to determine whether it is safe or not. After examining a packet, the filter will either allow through or drop the packet, depending on whether it is safe or not. A disadvantage of using a packet-filter firewall is that some packets that are safe may be blocked by accident. In addition, a disadvantage of a firewall is that it can reduce packet throughput and add to latency.

5.3.2 Access control rules

Many systems require access control. Printers, for example, sometimes require authentication of users wishing to print. Requiring users to authenticate *every* time they print a document is poor practice, however providing un-protected access to printers is also, in some situations, unsatisfactory. For other systems – databases, for example – the sensitivity of the resources being accessed readily justifies the in-

convenience of requiring authentication. Nevertheless, even when restricting access is clearly important, the careful management of access control to avoid unnecessary authentication dialogs with users is important.

5.3.3 Service Protection Policies

A network of ICT devices is like a community of interacting agents, which need to talk to each other (Kalam et al., 2003; Verma, 2002). These devices make requests, which must be checked for correctness. Each device has its own requirements. This checking can be viewed as "access control", but it also includes details of the service requested. For these devices to work together securely *and* effectively, the research contend that the following are needed:

(i) a common language for expressing access control (Jajodia et al., 2001; Sabelfeld and Myers, 2003; OASIS, 2010; Fong and Siahaan, 2011);

(ii) this language should be formal, so that computers can read it, but it should also have an informal version. Some aspects of access control cannot be formalised (Badger et al., 1995; Shi and Chadwick, 2011), and all aspects need to be understood by humans as well as by computers;

(iii) the expressive power of the access control language (in regard to logic) should be equivalent to first-order formal logic (Bandara et al., 2003; Zhang et al., 2008). The objects referenced needs to include all the devices, services, and users in the network;

(iv) policies should be defined for all participating users and devices, which define precisely how each device may be accessed to provide its services;

(v) policies should specify not only what is *not* allowed, but also what *is* allowed (these are the *service protection rules*) (Damianou et al., 2000).

Once we have a common access control language with these properties, these benefits accrue:

(i) a rigorously defined security design strategy (Ribeiro et al., 2001) becomes clear: policies should be defined for all users, and devices, which specify what *is* allowed (the service protection rules) as well as what is *not* allowed. The entire collection of policies of all devices must be logically consistent (Ribeiro et al., 2000).

(ii) Similarly, it becomes clear what it means to *validate* the security policies of an organisation or network (Alfaro et al., 2008): validation of an organisations policies consists of ensuring that a subset of rules are enforced and enforcing the rules in this subset is sufficient to ensure that the other formal rules are also true (Brodie et al., 2005; Nentwich et al., 2002; Kropiwiec et al., 2011). There may also be informal rules (such as restrictions on how staff use their personal computers), which either can't be enforced or guaranteed or perhaps even if enforcement was possible, good relations with employees dictates that strict enforcement is not appropriate. Such rules will not be considered in the same way when validating a security policy.

5.3.4 Dynamic Rules

Dynamic rules take the same form as static rules, but they can be added or removed by specific events. Network address translation (NAT) is an example of this behaviour which occurs in many gateway routers; another example, which does not occur regularly, but which could be employed to tighten security, is the dynamic filtering of VPN packets. The purpose of the dynamic rule changes would be to allow *only* VPN packets of *registered* VPN sessions. The additional expressive power afforded by dynamic rules is very significant. This additional expressive power is too

important to neglect. However, the additional *complexity* of systems of rules which incorporate dynamic rules makes them much more prone to mistakes or unintended consequences and makes such systems much harder to analyse. Consequently, best practice is to avoid the use of dynamic rules unless they are *essential*, i.e. unless it is impossible, or at least quite difficult, to achieve the desired goal without their use.

5.3.5 Validity

This research has identified that security policies include rules which are enforced, and also rules which are not enforced but which they nevertheless expect to hold. Some service protection rules are *required* to hold in order that our system is satisfactory at all. The subset of service protection rules which they *require* to be true is called the *mandatory* security protection rules. An implementation of a security policy in which all essential service protection rules are necessarily true is termed *valid*.

More formally, suppose (i) the enforced rules are E_1, E_2, ..., E_n; (ii) the *mandatory* service protection rules are S_1, S_2, ..., S_m; Then, the security policy is valid if and only if

$$E_1, E_2, \ldots, E_n \Rightarrow S_1, S_2, \ldots S_m.$$

5.4 Examples of solutions for service Problems

5.4.1 Firewall design

Example 5.1 Internal Filtering and Firewalls

An example network with internal firewalls is shown in Figure 5.1. The legend is shown in Figure 5.2.

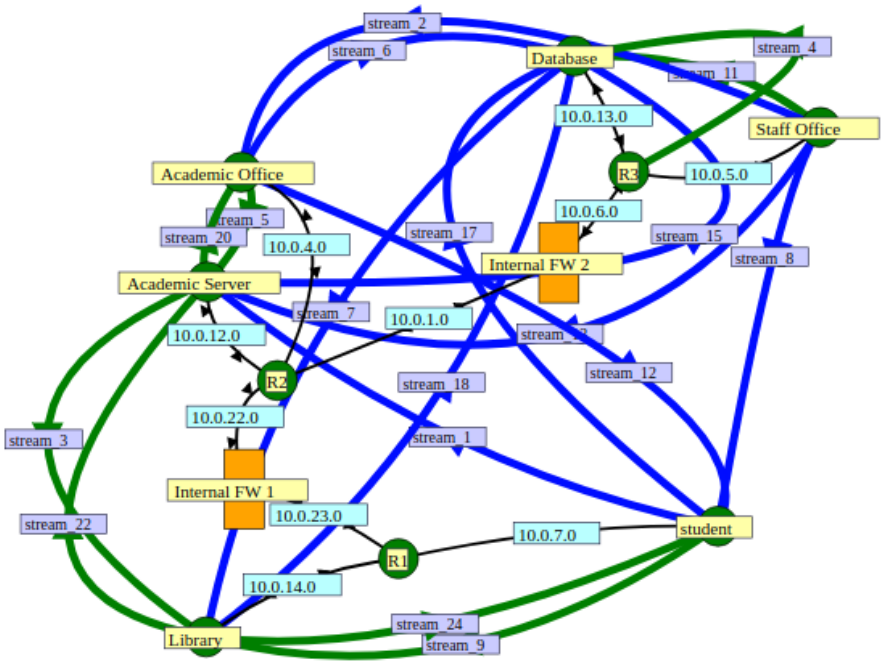

Figure 5.1: Example of a filtering configuration for example 5.1(See Figure 5.2 for legend).

Internal filtering has the potential to enhance security in environments where there are several classes of users with very different security profiles. For example, at a university, the administrative, academic and student users have quite different roles, responsibilities, and are quite reasonably perceived as introducing different risks.

Filtering prevents unwanted access in a comprehensive manner, which is fairly safe from unwanted interference. Traffic from different classes of users can be segregated from each other. In universities, there are three main classes of users: academics, students, administrators. Other classes of users than just these three will, in general, be needed, but for simplicity, in this chapter, we limit discussion to these three classes.

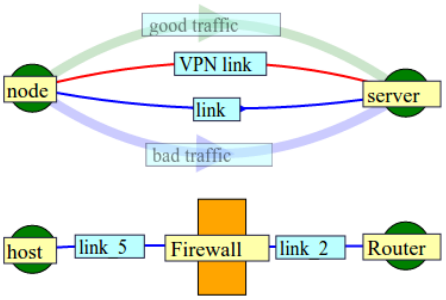

Figure 5.2: Legend for Figure 5.1.

Firewalls and filtering routers are configured by specifying a series of patterns to which packets are compared one-by-one. When a match is found, the matching packet may be dropped or accepted. Otherwise, the next pattern is checked. The patterns used in internal FW1 in our example network are listed in Table 5.1 and the patterns used in internal FW2 are listed in Table 5.2.

Problem

Because of the complexity of the firewall and filtering router configurations, it is common for certain pathways through the organizations network to be unintentionally blocked. This might be regarded "merely" as a mistake. However, correcting a mistake like this can be very costly.

The internal firewalls in a moderate to a large organisation like a university will usually be much more complex than these. It is not unlikely that these firewalls will contain errors that prevent access, which is needed and allow access, which is undesirable, by mistake. The mistakes will be logical errors in some cases, and simple blunders in others. Users affected by these firewall configuration errors will, in many cases, be unaware that their difficulties are caused by the firewalls, and the firewall administrators will, therefore, not receive any feedback about the firewall design from the affected users. The majority of users, therefore, rely on the existence of a small group of technically aware users who discover firewall configuration problems and

Table 5.1: Filtering rules for Internal Firewall 1

Name	SRC IP	DEST IP	DEST PORT	VERDICT
E1	10.0.14.0/24	*	*	accept
E2	*	10.0.14.0/24	*	accept
E3	10.0.12.0/24	10.0.4.0/24	*	accept
E4	10.0.4.0/24	10.0.12.0/24	*	accept
E5	*	*	*	drop

Table 5.2: Filtering rules for Internal Firewall 2

Name	SRC IP	DEST IP	DEST PORT	VERDICT
E6	10.0.5.0/24	10.0.13.0/24	*	accept
E7	10.0.13.0/24	10.0.5.0/24	*	accept
E8	*	*	*	drop

pass on their concerns to the administrators.

The firewalls in the network depicted in Figure 5.1 were tested by simulation. The traffic used in these tests corresponds to the rules shown in Table 5.3. It was discovered that there were many errors in the firewall rules, as first implemented. For example, IP addresses which were meant to be entered as `10.0.14.0` were entered instead as `10.0.0.14`. Traffic which was supposed to be blocked was not blocked, and traffic which was supposed to be allowed was blocked.

After the errors were found and corrected, the simulation produced the results shown in Figure 5.3. The simulation was carried out using ns3 (Riley and Henderson, 2010), using the Click system (Kohler et al., 2000) to implement the firewalls, and to use the Netml (Addie et al., 2011b) system to construct the simulation program and generate plots from the results. A tabular display of the throughput is also shown in Table 5.4. All the traffic streams which were supposed to be blocked are now

Table 5.3: Service Protection Policy rules (rules for testing)

Name	SRC IP	DEST IP	DEST PORT	VERDICT
O1	10.0.7.0/24	10.0.14.0/24	*	accept
O2	10.0.4.0/24	10.0.14.0/24	*	accept
O3	10.0.4.0/24	10.0.12.0/24	*	accept
O4	10.0.7.0/24	10.0.4.0/24	*	drop
O5	10.0.7.0/24	10.0.13.0/24	*	drop
O6	10.0.4./24	10.0.13.0/24	*	drop
O7	10.0.5.0/24	10.0.14.0/24	*	accept
O8	10.0.5.0/24	10.0.12.0/24	*	drop
O9	10.0.5.0/24	10.0.13.0/24	*	accept
O10	10.0.14.0/24	10.0.7.0/24	*	accept
O11	10.0.14.0/24	10.0.4.0/24	*	accept
O12	10.0.12.0/24	10.0.4.0/24	*	accept
O13	10.0.4.0/24	10.0.7.0/24	*	drop
O14	10.0.13.0/24	10.0.7.0/24	*	drop
O15	10.0.13.0/24	10.0.4./24	*	drop
O16	10.0.14.0/24	10.0.5.0/24	*	accept
O17	10.0.12.0/24	10.0.5.0/24	*	dO4rop
O18	10.0.13.0/24	10.0.5.0/24	*	accept

blocked, and the traffic streams which are meant to be allowed, are allowed.

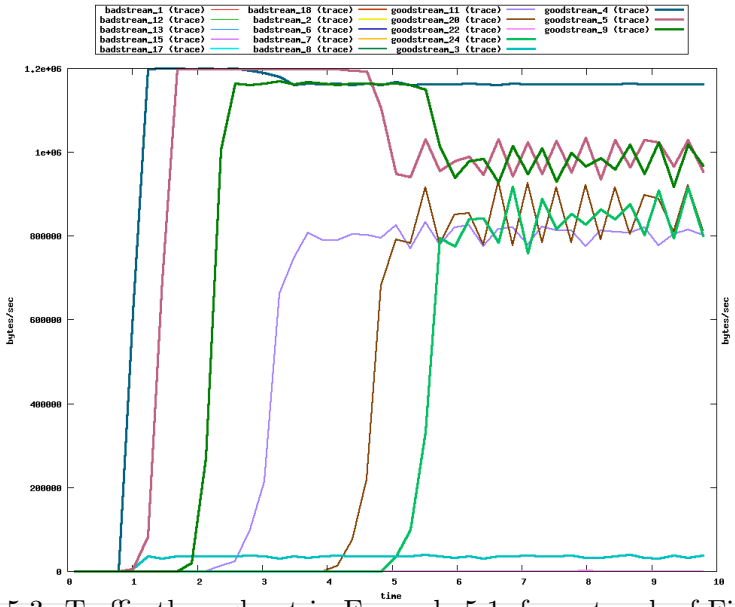

Figure 5.3: Traffic throughput in Example 5.1, for network of Figure 5.1

The rules in Table 5.3 form the service protection policy for the network firewalls and filters. They take the same form as the firewall rules which are used to implement the firewall, but they express desired outcomes rather than the mechanism of implementation(De Santis et al., 2013).

□

5.4.2 VPN design

In this section, VPN cybersecurity design is considered, in a series of examples. The first example – Example 5.2 – provides a basic design; in Example 5.3, the risk that a firewall can be used as a backdoor by attackers is addressed. And finally, in Example 5.4, the risk that VPNs from outside the university can be used as a backdoor is addressed.

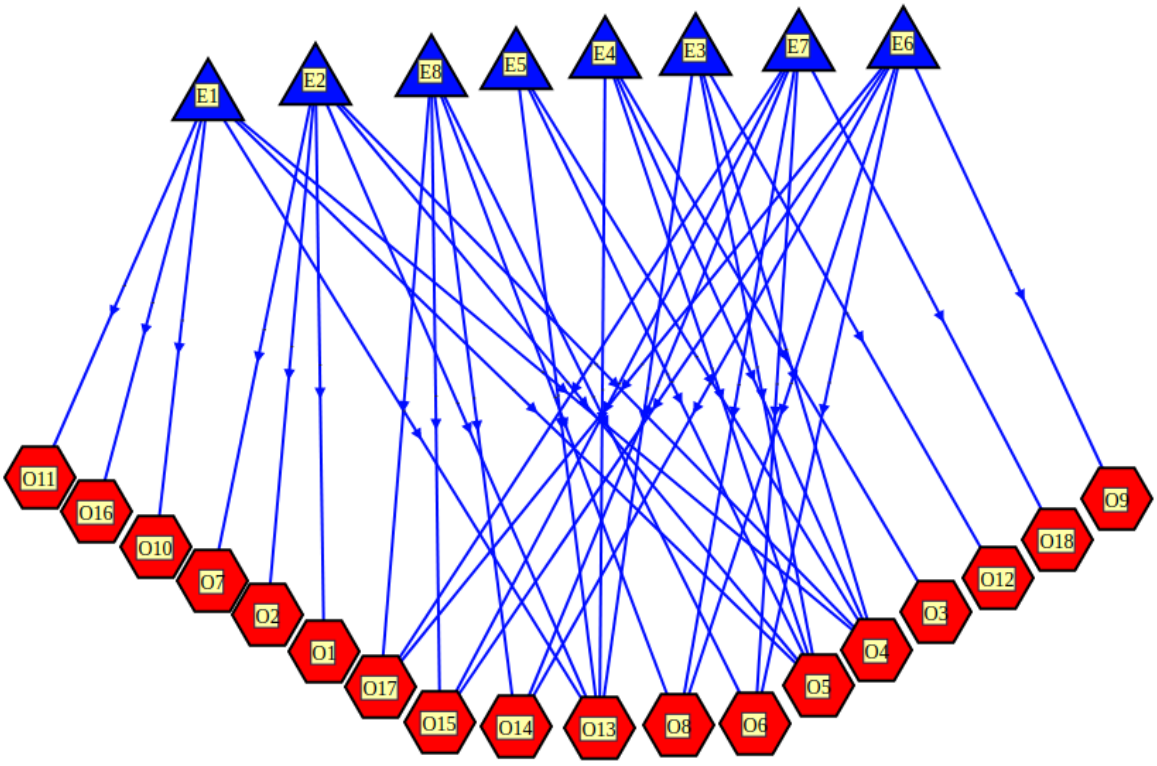

Figure 5.4: Inference graph for Example 5.1

Table 5.4: Throughput of traffic streams in Example 5.1 as reported by Netml/ns-3 simulation

Trace	Time-averaged Mean	Sample count	Standard Deviation
badstream1 (avg wdw = 0.2250s)	0.0000	45.0	0.0000
badstream12 (avg wdw = 0.2250s)	0.0000	45.0	0.0000
badstream13 (avg wdw = 0.2250s)	0.0000	45.0	0.0000
badstream15 (avg wdw = 0.2250s)	0.0000	45.0	0.0000
badstream17 (avg wdw = 0.2250s)	0.0000	45.0	0.0000
badstream18 (avg wdw = 0.2250s)	0.0000	45.0	0.0000
badstream2 (avg wdw = 0.2250s)	0.0000	45.0	0.0000
badstream6 (avg wdw = 0.2250s)	0.0000	45.0	0.0000
badstream7 (avg wdw = 0.2250s)	0.0000	45.0	0.0000
badstream8 (avg wdw = 0.2250s)	0.0000	45.0	0.0000
goodstream11 (avg wdw = 0.2250s)	992.3716	45.0	657.6029
goodstream20 (avg wdw = 0.2250s)	798.8264	45.0	792.7667
goodstream22 (avg wdw = 0.2250s)	19.2665	45.0	0.6551
goodstream24 (avg wdw = 0.2250s)	686.4548	45.0	759.6388
goodstream3 (avg wdw = 0.2250s)	54.1809	45.0	24.5555
goodstream4 (avg wdw = 0.2250s)	1843.8142	45.0	677.5971
goodstream5 (avg wdw = 0.2250s)	1643.3252	45.0	723.1831
goodstream9 (avg wdw = 0.2250s)	1452.5183	45.0	836.1965

The first two examples constitute a description and analysis of current practice in the use of VPN's whereas the last example goes beyond common practice, and therefore should be regarded as a recommendation.

In each case, the rules which the design seeks to achieve (objectives), are identified, the rules chosen to enforce to achieve these objectives are described, a model of the system is constructed and simulated, thereby verifying that all the rules are correctly implemented, and also inference graphs which explain how the enforced rules ensure the objectives have been constructed.

Example 5.2 Basic VPN Configuration

Problem

The first problem we consider in this example is that of *defining* precisely what access is enabled by a VPN. Users should be provided with this information, however, in practice, this is rarely done. The second problem is how to prevent undesirable or un-allowed VPNs from being set up. Since a VPN is usually encrypted, if we allow arbitrary VPN traffic, it could potentially be used to mask highly undesirable forms of access, like data theft.

Solution

What should users be able to access via a VPN? Specifying this detail would be error-prone and unmanageable. A better approach is to adopt the following rule, which is an excellent example of a service protection policy:

VPN Service Rule

> *Users accessing the organisation via the VPN have access to the same services as from their normal place of work.*

We suppose that the network in this example is similar to Example 5.1, as shown in Figure 5.1 but with the addition of a VPN server and an external firewall. The network of this example is shown in Figure 5.5.

The rules in the internal firewalls are the same as before, and the rules in the external firewall are shown in Table 5.10.

The VPN Service Rule can be interpreted as a logical rigorously defined rule. It has a precise meaning. But, a rule like this is difficult to implement. So,

(i) how can this rule be implemented, in such a way that we can *prove* that it

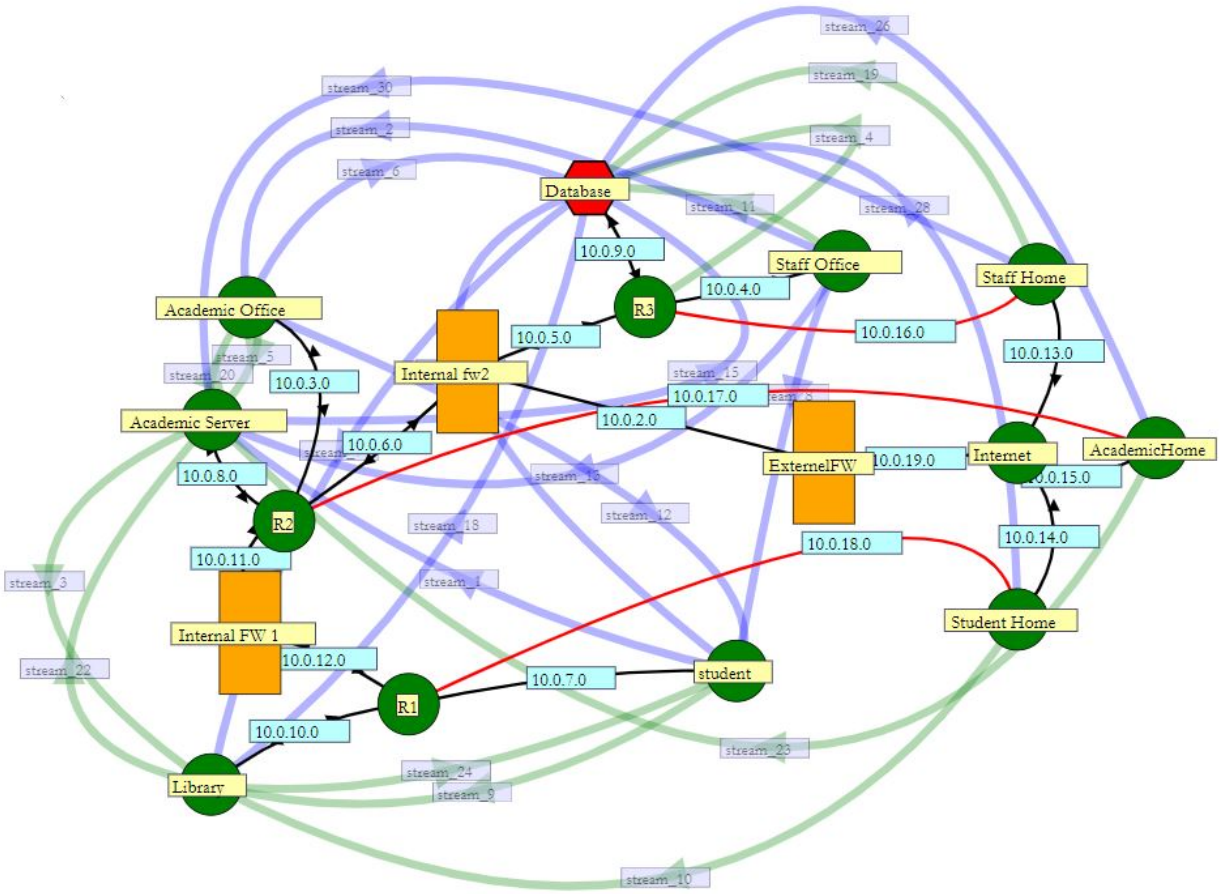

Figure 5.5: Filtering and VPN configuration for Example 5.2

holds?

(ii) is it possible that security is compromised by implementing this rule?

Implementation

The VPN Service Rule can be enforced as follows: when a user authenticates with the organisation's VPN, a VPN link is set up between the user's external computer and the *same router* to which the user normally connects when at work. The external end of this VPN link is then equipped with an IP address in the same range as their computer at work (which might be a desktop computer, or it could be the same actual computer as used from outside the organisation).

Finally, the external firewall of the organisation must be configured to allow all of the VPN traffic to pass through without interference.

The objectives in this example are formally stated in Table 5.6. The conditions which have been enforced, in this example, in order to ensure that these objectives hold, are also listed in Table 5.6. For example, the condition that all VPN traffic is allowed through the firewall is stated formally in EV4, in Table 5.6. Figure 5.7 is an inference graph that shows how the enforced conditions can be used to prove that the objectives are met.

Validation

In effect, the VPN service sets up a new *virtual link* between the VPN client and his or her home router. If this link freely carries all traffic from their place of residence, where the VPN client is located, the VPN service protection rule will hold. However, setting up such a link has the potential to affect other security policies. To ensure that this can't happen, we need to install rules governing the use of the virtual link,

namely that it can *only* be used by traffic generated by the VPN client.

Preventing all use of this link by others can be achieved by blocking all UDP traffic and blocking the sending of `tcp SYN` packets, except in the direction from the VPN user.

This design of a VPN service appears to meet all the criteria discussed up to now. However, it still needs to be validated if we wish to be truly confident that it provides the desired services and mandates all the rules we have adopted for this network.

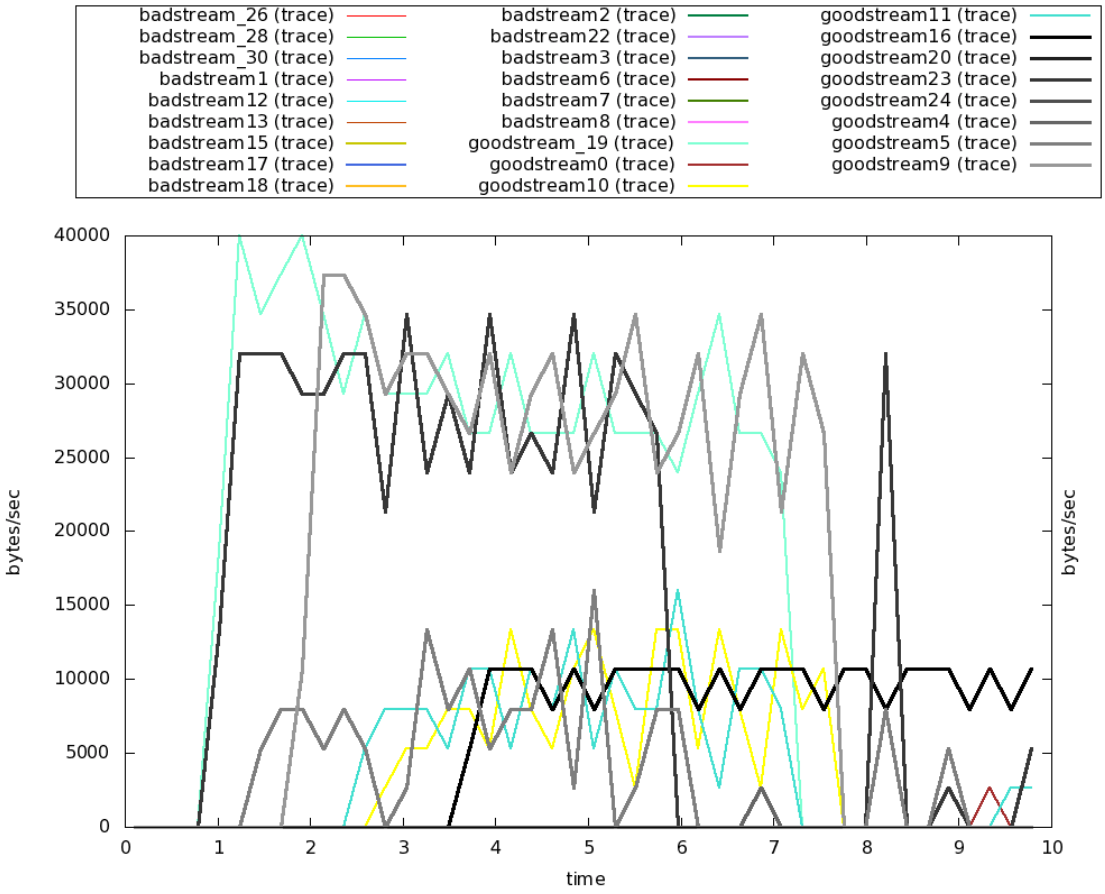

Figure 5.6: Carried traffic for Example 5.2, for the network in Figure 5.5

Let us adopt all the rules from Example 5.1 in this network, as a starting point. The rules for the main firewall, between the institution and the Internet, are shown in Ta-

Table 5.5: Traffic throughput in Example 5.2, for the network of Figure 5.5

Trace	Time-averaged Mean	Sample count	Standard Deviation
badstream_26 (avg wdw = 0.2250s)	0.0000	45.0	0.0000
badstream_28 (avg wdw = 0.2250s)	0.0000	45.0	0.0000
badstream_30 (avg wdw = 0.2250s)	0.0000	45.0	0.0000
badstream1 (avg wdw = 0.2250s)	0.0000	45.0	0.0000
badstream12 (avg wdw = 0.2250s)	0.0000	45.0	0.0000
badstream13 (avg wdw = 0.2250s)	0.0000	45.0	0.0000
badstream15 (avg wdw = 0.2250s)	0.0000	45.0	0.0000
badstream17 (avg wdw = 0.2250s)	0.0000	45.0	0.0000
badstream18 (avg wdw = 0.2250s)	0.0000	45.0	0.0000
badstream2 (avg wdw = 0.2250s)	0.0000	45.0	0.0000
badstream22 (avg wdw = 0.2250s)	0.0000	45.0	0.0000
badstream3 (avg wdw = 0.2250s)	0.0000	45.0	0.0000
badstream6 (avg wdw = 0.2250s)	0.0000	45.0	0.0000
badstream7 (avg wdw = 0.2250s)	0.0000	45.0	0.0000
badstream8 (avg wdw = 0.2250s)	0.0000	45.0	0.0000
goodstream_19 (avg wdw = 0.2250s)	16774.5721	45.0	13258.4748
goodstream0 (avg wdw = 0.2250s)	104.8411	45.0	351.1528
goodstream10 (avg wdw = 0.2250s)	3826.6993	45.0	4395.4709
goodstream11 (avg wdw = 0.2250s)	3774.2787	45.0	4188.4847
goodstream16 (avg wdw = 0.2250s)	5504.1565	45.0	4331.8907
goodstream20 (avg wdw = 0.2250s)	52.4205	45.0	0.0000
goodstream23 (avg wdw = 0.2250s)	13524.4988	45.0	12927.7821
goodstream24 (avg wdw = 0.2250s)	52.4205	45.0	0.0000
goodstream4 (avg wdw = 0.2250s)	157.2616	45.0	351.1528
goodstream5 (avg wdw = 0.2250s)	3669.4377	45.0	3986.2203
goodstream9 (avg wdw = 0.2250s)	14939.8533	45.0	13276.8513

ble 5.10. In addition, we adopt the VPN Service Rule, and one additional rule, which is that users other than the VPN user can't make use of the link set up dynamically for VPN traffic. The simulation of this network produced the throughput results shown in Figure 5.6, with the numerical results shown in Table 5.5. The simulation was carried out using Netml/ns-3/Click, as discussed above. All the traffic streams which were supposed to be blocked are blocked, and the traffic streams which are meant to be allowed, are allowed. In this example, the network of Figure 5.1 is extended to include locations *outside* the university from where the students and staff wish to continue using university services via the Internet. The resulting network, in which external clients of the university have been added, is shown in Figure 5.5. When a university student or employee wishes to access the university, we assume they make use of a VPN server, which sets up a *VPN link* from their computer to a location inside the university. The challenge in VPN design is to configure it so that users are able to use the services they need in a manner that does not compromise

Table 5.6: Rules of normal VPN for Example 5.2

Rule Name	**Details**
O1	Academic staff can access university resources just as at work.
O2	Admin staff can access university resources just as at work.
O3	Students can access university resources just as at work.
EV1	The academic VPN connects an academic client directly to the same router which their computer is connected to.
EV2	The admin VPN connects an admin client directly to the same router which their computer is connected to.
EV3	The student VPN connects a student client directly to the same router which their computer is connected to.
EV4	Firewall allows all VPN traffic.

security.

The red links in this diagram represent VPN links which are being used by academic staff, administrative staff and students to make use of services of the university which are normally available only to clients on campus. □

Example 5.3 VPN Client Rule

Because a VPN can potentially be used as a back-door into a network, a rule which is commonly adopted, for the operation of VPN's, is that when a VPN link has been set up, for a client,

All traffic from a VPN client must be routed through the VPN.

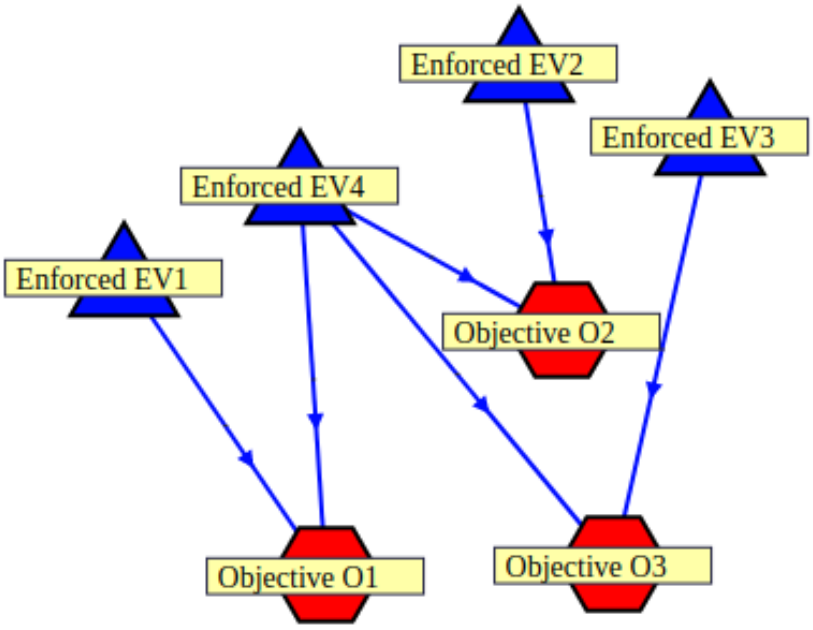

Figure 5.7: Inference graph for Example 5.2

The additional rules for this example are also shown in Table 5.8. In Figure 5.8,

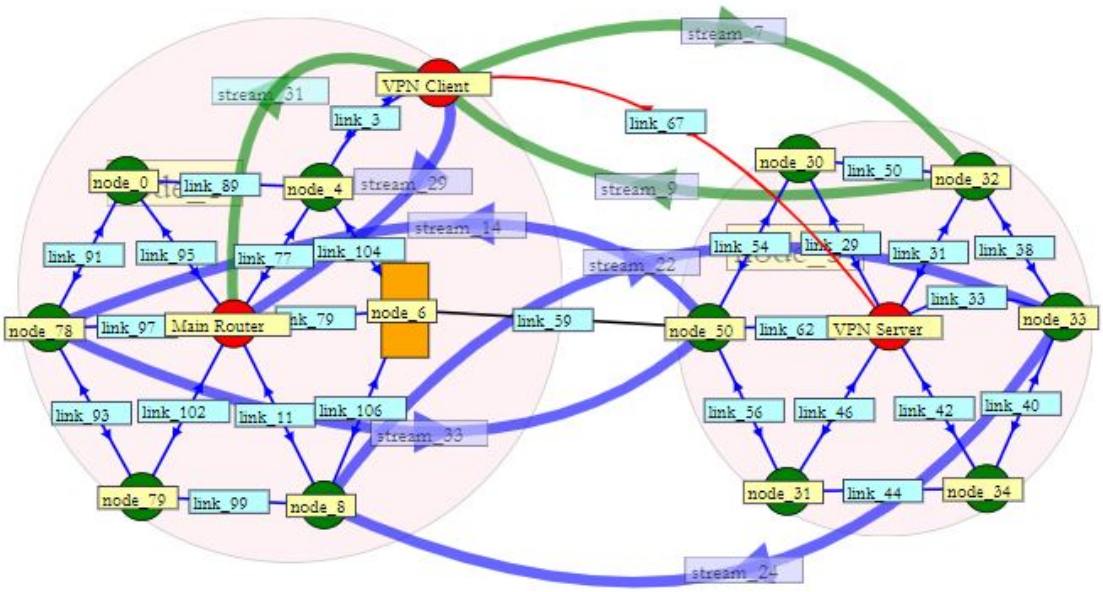

Figure 5.8: A VPN Connection from one intranet to another for Example 5.3

when the VPN between VPN Client and VPN Server is active, according this rule *all* communication with VPN Client *must* pass through VPN Server. Simulation of this network produced the throughput results shown in Figure 5.9 and Table 5.7.

The simulation was carried out using Netml/ns-3/Click, as discussed above. All the traffic streams which were supposed to be blocked are blocked, and the traffic streams which are meant to be allowed, are allowed.

Table 5.7: Traffic throughput in Example 5.3, for the network of Figure 5.8

Trace	Time-averaged Mean	Sample count	Standard Deviation
badstream_22 (avg wdw = 0.2250s)	0.0000	45.0	0.0000
badstream_24 (avg wdw = 0.2250s)	0.0000	45.0	0.0000
badstream_29 (avg wdw = 0.2250s)	0.0000	45.0	0.0000
badstream_31 (avg wdw = 0.2250s)	0.0000	45.0	0.0000
goodstream_14 (avg wdw = 0.2250s)	17193.9364	45.0	15459.7886
goodstream_33 (avg wdw = 0.2250s)	2725.8680	45.0	6117.9789
goodstream_7 (avg wdw = 0.2250s)	7653.3985	45.0	8084.8373
goodstream_9 (avg wdw = 0.2250s)	16879.4132	45.0	12802.3089

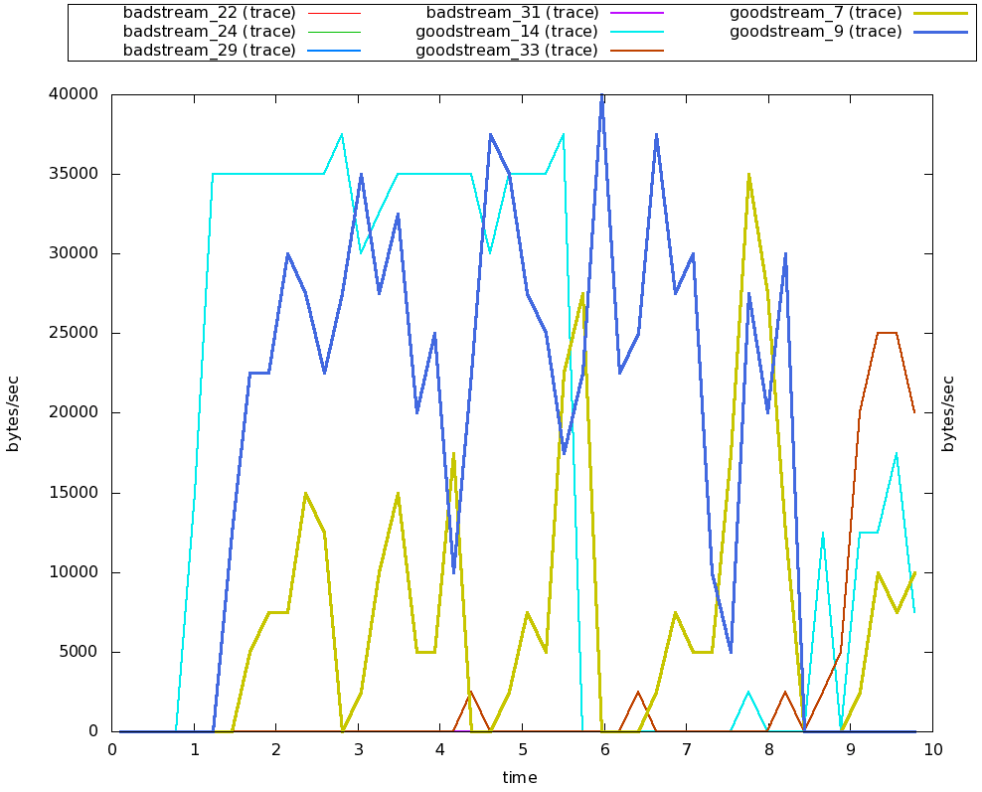

Figure 5.9: Traffic throughput for security of two companies in Example 5.3

Table 5.8: Additional Rule to protect networks from VPN as backdoor, for Example 5.3

Rule Name	**Details**
O4	Attackers can not use a VPN of academic staff, admin staff and Students to again access to university resources.
O5	Users cannot modify the routing of their client computers to allow routing to anywhere except the VPN server, except as used by the VPN client software itself.
EV5	All routing of a client node using a VPN goes through the VPN.
EV6	VPN client software and client routing is validated, to ensure that routing *must* pass all traffic, except the VPN implementation traffic, through the VPN link.

One problem remains with the rule EV5 for VPN clients. That is that VPN clients are normally computers belonging to users. There is, therefore, nothing to stop users modifying the VPN client software, or developing their own versions of this software, which allows routing between their computer and other hosts in their network. If they do so, the security of their own home network, and the network they are connecting to and the computers in those networks, will be compromised.

One solution to this problem is to mandate that VPN client software has to be *validated*. That is to say, VPN client software must be *tamper proof* and the proof that the software has not been altered will need to be provided every time it is used. The checking could be conducted by the VPN server software, or by the gateway of the client domain, or by both of these parties. Without such checking, every VPN is a potential security risk. □

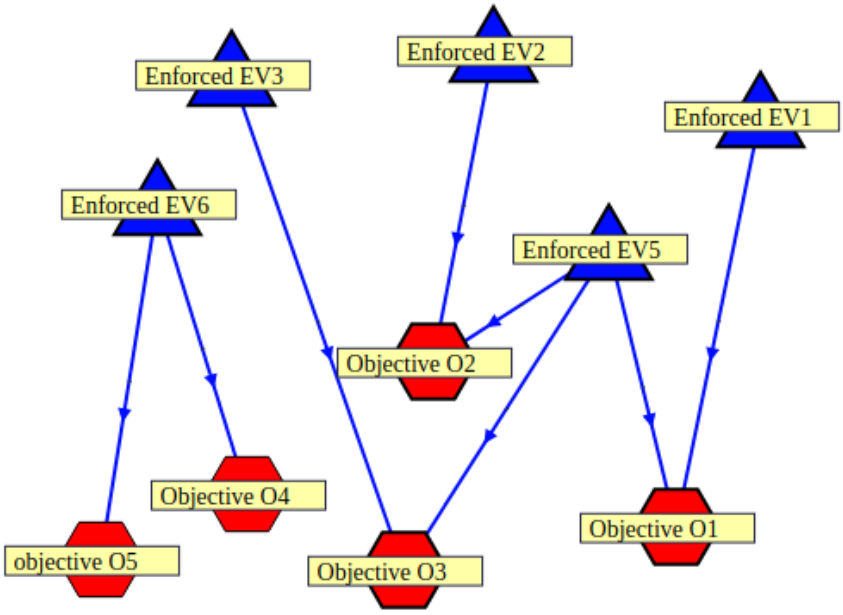

Figure 5.10: Inference graph for Example 5.3

Example 5.4 VPN Firewall Dynamic Reconfiguration

This example explains VPN links going *out* of the network need to be allowed through the firewall, but only after checking, as shown in Figure 5.11. To the inference graph in 5.12 use rules in Table 5.9 and Table 5.10. Also, mention that all the rules of the previous VPN examples, except EV4, are still active. Moreover, all rules in Table 5.10 are sub-rules from rules in Table 5.9. And we can see orange VPN link not allow because VPN Firewall Dynamic.

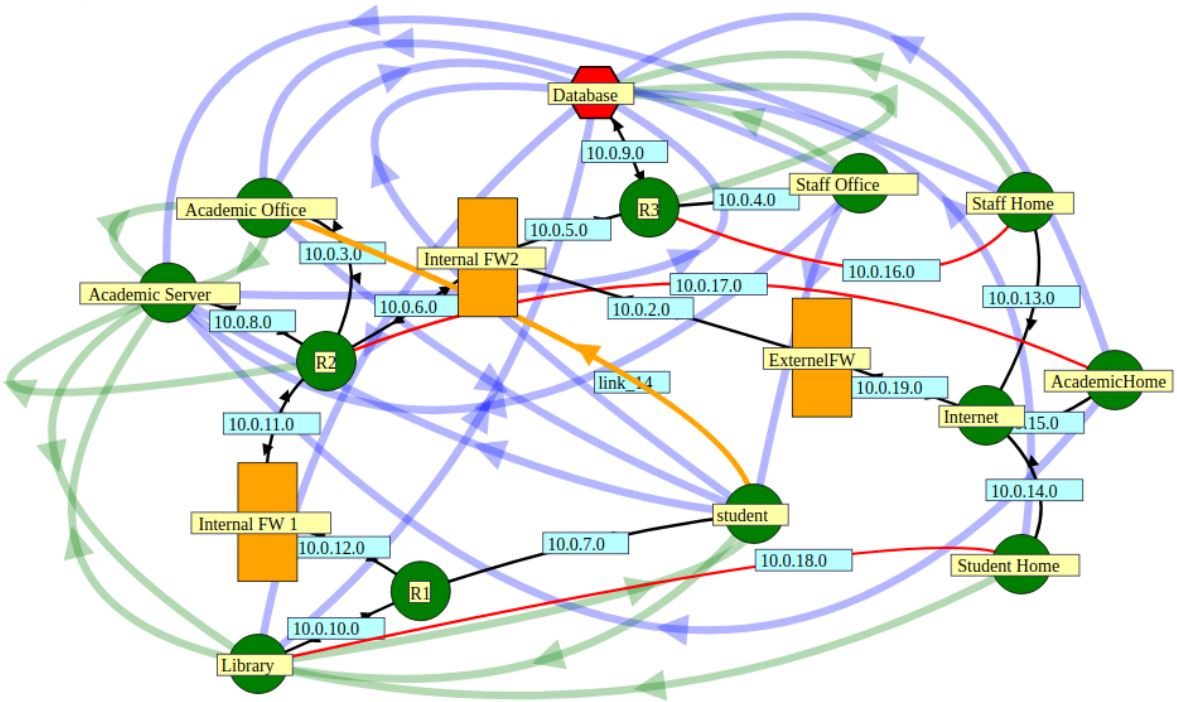

Figure 5.11: Network for VPNs in Example 5.4

□

Example 5.5 Printer Access

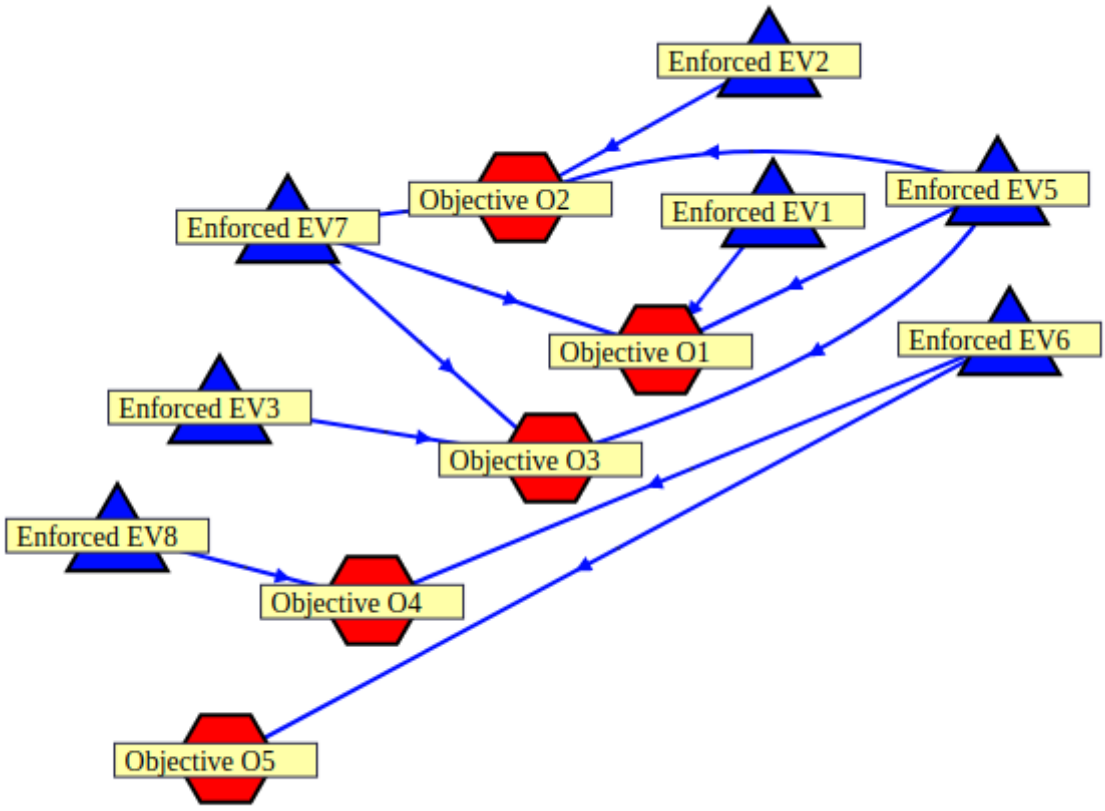

Figure 5.12: Inference graph for Example 5.4

Table 5.9: Dynamic Rules for VPNs in Example 5.4

Rule Name	**Details**
EV7	For each VPN, v, set up by VPN server: VPN traffic from v-specific source to v-specific destination is allowed.
EV8	All other VPN traffic is blocked.

Table 5.10: Filtering rules for External Firewall for Example 5.4

Name	SRC IP	DEST IP	DEST PORT	VERDICT
E7a	10.0.15.0/24	10.0.14.0/24	443	accept
E7b	10.0.14.0/24	10.0.15.0/24	443	accept
E7c	10.0.16.0/24	10.0.12.0/24	443	accept
E7d	10.0.12.0/24	10.0.16.0/24	443	accept
E7f	10.0.17.0/24	10.0.13.0/24	443	accept
E7e	10.0.13.0/24	10.0.17.0/24	443	accept
E8	*	*	443	drop

Problem

Security rules sometimes prevent legitimate users from being able to print. This can be very frustrating because sometimes printed documents are required urgently, but security re-configuration may require days.

Security configuration problems of this sort, which cause printing to fail mysteriously, are more likely to affect users with *non-standard* computer configurations: for example, users of Linux or Macintosh computers, or short-term guests. For this reason, some organisations adopt the "solution" of adopting a Standard Operating Environment (SOE), which is supported by the ICT Department. However, this really hides the problem rather than solving it.

Solution

The natural, and obvious, rule for printer access is:

> *Users should (with availability 99.9%) be able to print on their nearest accessible printer, without having to undertake additional security configuration.* (5.1)

Protocols for printer access have evolved along several parallel paths and it is surprisingly difficult to validate this simple and natural rule, particularly if it is required to include a full variety of operating systems.

The security configuration for a printer can vary widely. Direct access to a printer could be blocked completely, except for access via a printer spooler, or it could be allowed subject to authentication at the printer using the printer's control language, or it could be freely accessible to all. It is not unlikely that each printer (of which there may be dozens) has its own different configuration, which is chosen to suit its situation. A key element in ensuring that rule (5.1) holds is to make the security configuration of each printer known to users. Hence, in this instance, the application of Principle ((ii)) dictates that a description of the security configuration of the printer, in English, should appear next to each printer. This will improve the chances of realizing (5.1) considerably. □

Example 5.6 Single Sign-on

Problem

Single sign-on has the potential to reduce security or access to services severely if it is misconfigured:

(a) If the duration of the single sign-on authority is too long, a user's computer might enable an attacker to gain access to valuable resources while its owner is out of their office or otherwise indisposed;

(b) If the single sign-on server itself is overloaded, or vulnerable to deliberate or accidental damage, it might become a bottleneck preventing users from accessing many resources;

(c) When a user's authentication details (identity and password, for example), are obtained by an attacker, the range of services compromised is greatly increased by the use of single sign-on.

Solution

Single-sign-on is a complex concept that requires careful design. It is likely that users will continue to keep certain domains of activity isolated from others. On the other hand, even when two domains of user activity are not linked by single-sign-on authority, they can provide auxiliary security mechanisms. For example, mobile phones are often used to provide secondary authentication for banking transactions.

Here are some possible service protection policies for single sign-on:

(a-1) The duration of an authentication session never exceeds a certain system-wide limit;

(a-2) Users can configure the session duration within certain limits;

(a-3) Users can invalidate the current session (logout) at any time;

(b) The authority transfer network and the single sign-on servers are carefully monitored and always configured to handle the current load;

(c) Users will be notified of all changes to the configuration of their single sign-on facility.

Because users have different practices, allowing users to customise the key features of a single sign-on system has the potential to considerably improve performance and security. □

5.5 Validation of Security and Service Protection Rules

Validation of security policies has been considered in (Zhang et al., 2008; Nan Zhang and Guelev, 2005; Hamed and Al-Shaer, 2006) and it is either asserted or implied in these papers that validation of security rules is equivalent to or implied by overall logical consistency of policies. However, (Hamed and Al-Shaer, 2006) implies that this is not a practical approach for validation of the entire collection of policies, and (Zhang et al., 2008; Nan Zhang and Guelev, 2005) approach the issue of validation from a theoretical perspective so the question of whether it is practical is not addressed. Informally, a set of firewall rules is *valid* if it achieves its intended goal. In particular, we expect all rules to be enforced at all times. Logical inconsistency undermines this expectation because nothing can be proved in an inconsistent system. Proof ceases to be meaningful in an inconsistent system. Let us, therefore, define *validity* of a system based on rules, for a network (such as a firewall), as the property that the collection of rules is *consistent*, and also, that the rules are *always true*. It follows, as asserted in Section 5.3, in Subsection 5.3.5, that the mandatory rules must logically follow from the rules which are enforced by the network.

5.5.1 Procedural vs Declarative Policies

In firewalls rules have an "order", but in logic, they do not. In firewalls, the language adopted to express policies is usually *procedural*. Although the individual rules from which a firewall configuration is composed are logical statements about the processing of each packet, it is common to connect successive rules by the "connective":

and if the previous rules did not decide the fate of this packet, then...

This connective combines *sets* of rules together, and refers to the outcome from a *set* of "previous" rules. Once all the members of the list of rules of a firewall has been combined together in this way, the concept of "ordering" is no longer necessary, because the statement which combines the rules together encodes it. This explains why, at first sight, consistency does not appear to be important for firewalls. However, when service protection rules are included in the analysis, consistency is, after all, important for firewalls.

5.5.2 Simulation Tools

Network simulation allows researchers to test scenarios that are difficult or expensive to implement in the real world.

The simulation and modeling tools ns-3, Omnet++, D.Jsim, Packet Tracer and Petri.NET have been considered for the type of simulation and validation under discussion. All of these are simulation tools, except the last, which seeks to validate a protocol design by state exploration. In this chapter, we have used exclusively ns-3 (Riley and Henderson, 2010) in conjunction with the Netml user-interface. The concept of traffic is not supported explicitly in most simulation systems but is a key feature of the Netml/ns-3 system. Ns-3 itself does not support the traffic concept but

instead provides *sources* and *applications*, which can be used together to implement traffic.

In a Netml/ns-3 (Addie et al., 2006; Addie, 2010; Addie et al., 2011b) simulation a report can be generated which shows whether traffic is carried or not. Figure 5.3 and Table 5.4 show this report for Example 5.1, Figure 5.6 and Table 5.5 show this report for Example 5.2, and Figure 5.9 and Table 5.7 show this report for Example 5.3. From this traffic report it is straightforward to see whether the intended rules are satisfied or not. Availability of such a report is a key requirement of any system used for validating a security design. For this reason, and because any desired new features for Netml/ns-3 can be developed in-house, the Netml/ns-3 system was adopted in this chapter.

5.5.3 Models in the sense of mathematical logic

In this section, it is proved that certain types of simulation are able to *prove* the consistency of a collection of security rules. The proof is given in Subsection F.

Proving consistency of security rules is important, in general. For some sets of rules, however, consistency may be a very weak condition. In Section 5.6, an example (Example 5.11) is given in which a simulation with no traffic can prove the consistency of the rules. For this reason, it is useful to include rules which assert that certain types of traffic do occur. Also there are cases which it is important to prove that certain rules are enforced *as well as proving consistency.*

Proving that rules are enforced cannot always be done by simulation. There is no systematic procedure for proving mathematical theorems, and since the system of rules which define security is essentially open-ended, it is extremely unlikely that any systematic procedure can exist for proving rules. In particular, simulation cannot

be expected to provide such a procedure. However, creating networks and systems in general, which include rules of this sort, which are difficult to prove, is more indicative of bad design than weak analysis.

The term *model* is used with several quite different meanings in mathematics. In this chapter, we need to use both models as defined in mathematical logic and models in the sense of simulation. The former will henceforth be referred to as *formal models*.

Definition 2 *Suppose S is a set of statements involving constants, c_α, $\alpha \in I$, I an arbitrary index set, and relations (or predicates) R_α, $\alpha \in J$. A set M together with elements, $\widetilde{c}_\alpha$ and relations, $\widetilde{R}_\alpha$ on M, for which there is a mapping $\phi : c_\alpha \to \widetilde{c}_\alpha$, $R_\alpha \to \widetilde{R}_\alpha$ is termed* a model *for S if all the statements in S are true in M (under the interpretation ϕ)(Cohen, 1966, I.4) .*

Example 5.7 A simple logical system

This logical system has a predicate C, with two parameters, such that $C(a, b)$ says a is a client of b, a predicate S with the interpretation that $S(s)$ means s is a server, and a rule: $\forall\, x\, \exists\, y C(y, x)$ (all servers have clients). This set of rules is consistent and a model for it is provided by a set, X, with two elements $\widetilde{s}$, $\widetilde{c}$. by setting $\widetilde{C} = \{(\widetilde{c}, \widetilde{s})\}$, and $\widetilde{S} = \{\widetilde{s}\}$. □

5.5.4 Simulation Models

The concept of simulation technology for computer networks is used frequently in the communications and research of computer networks on the basis of a model to know the behaviour of the computer networks that might exist in the future. All the features of a real network can be modelled in a simulation and these details can

be monitored, and statistics concerning their occurrence collected (Law, 1998; Yang et al., 2009).

5.5.5 Validation

In a logical system which is inconsistent, it is possible to prove any statement at all, and its opposite. Consequently, it is not possible to know how such a system will behave in practice. Therefore, logical consistency is a requirement of the collection of security policies of any organisation.

However, consistency is not always sufficient to ensure that a security policy is valid. *In addition* to consistency, as discussed in Section 5.3, this study need to ensure that if all rules which the policy requires to be enforced *are* enforced then all the mandatory service protection rules will hold. Theorem 1 identifies how simulations can be used to check validity. In the present context, this theorem says that for a mandatory rule to be provably true, it is necessary and sufficient if it is true in all simulations. Thus, one simulation is not always enough to prove validity. However, in many cases (for example, when firewall rules are not state-dependent) we can infer, even from one simulation, that a mandatory rule will be true in all simulations and hence is valid.

Prior to discussing how to prove statements about a network we need to choose the *language* in which these statements will be made. In the present work, it will not be necessary to provide an exhaustive list of the type of statements which can be made. We merely assume that there is a collection of predicates that can be applied to the network under consideration. The language, usually denoted by $\mathcal{L}$, is then assumed to include all statements which can be composed from these predicates, together with logical symbols such as the logical connectives, variables, and quantifiers.

Definition 3 *A simulation relative to the language $\mathcal{L}$ is* defined *to be a computer program which can be observed in such a way that during any* run*, of the program, it is possible to observe whether any statement expressed in the language $\mathcal{L}$ is true or false.*

If a statement, ϕ is true in a simulation run, S, in this research, we write $S \Vdash \phi$. In words: S simulates ϕ. Given a finite set of statements, T, and another statement, ϕ, let us write $T \Vdash \phi$ if in every simulation run in which T holds, ϕ also holds. In words: in all simulation runs, T implies ϕ.

In mathematical logic, it is conventional to use *models* rather than simulations. When a statement, ϕ, is valid in a model, $\mathcal{M}$, we write $\mathcal{M} \models \phi$ and say "$\mathcal{M}$ models ϕ. Also, if a statement, ϕ, is true in *all* models, we write $\models \phi$, or, if it is true in all models for which the theory T is true, we write $T \models \phi$.

Suppose we have a formal language $\mathcal{L}$ based on the predicate calculus and a sequence of sentences, $\phi_1, \ldots, \phi_n$ expressed this language. The predicate calculus allows only predicates, functions, and constants as methods for extending the language, e.g. P_1, P_2, $\ldots$, f_1, $\ldots$, and c_1, $\ldots$. Each of these predicates, etc. has an associated interpretation. For example, $P_1(x, y, z)$ could mean that x is the source node, and y the destination node of a link z. The predicate calculus allows sentences to be formed by using the predicates, functions and constants together with logical connectives ($\wedge$, $\vee$, $\neg$, $\rightarrow$, variables ($x_1, \ldots, y_1, \ldots$), and quantifiers $\forall$, $\exists$. An example of a sentence might be $\forall\, x \,\forall\, z \neg P_1(x, x, z)$, meaning there are no links connecting a node to itself.

The predicate calculus includes one more symbol, $\vdash$, which we can pronounce "Valid", or "provable". This symbol ($\vdash$) is not used within sentences, but only in front of them, with the meaning that the sentence in question has been proved. For example, $\vdash \forall\, x, x = x$ is the statement "It is provable that for all x, $x = x$."

Example 5.8 A simple model

Suppose the language has two predicates: $N(x)$ which says x is a node, and $L(u, v, w)$ which says u is a link from v to w. An example of a model is as follows (and is illustrated in Figure 5.13): the domain is $\{A, B, L\}$. The relation, a set, $N = \{A, B\}$, and the relation $L = \{(L, A, B)\}$, where $(L, (A, B)) \doteq (L, (A, B))$ and $(x, y) \doteq \{\{x\}, \{x, y\}\}$. The definitions of n-tuple and ordered-pair given here are conventional and given here only for completeness. The underlying objects in the domain have been denoted by A, B, L, but in order for the model to be entirely constructed from sets we can choose $A = 0$, $B = 1$, and $L = 2$, adopting the standard definition of numbers as sets. □

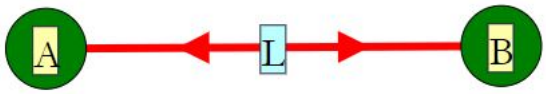

Figure 5.13: A Simple model,network for Example 5.8

The predicate calculus is not a particularly expressive language. However, it includes all the features necessary to express logical deductions, and this is the key requirement in the present instance. Note, however, that we will not consider ourselves bound to use precisely the symbols listed above if a more natural terminology which is logically equivalent proves to be more suitable. A more expressive and flexible language, such as English, is also satisfactory so long as it is sufficiently precise. The process of reducing statements in such a language to the Predicate Calculus is, for the most part mechanical and tedious.

Suppose our simulations and models are concerned with checking a sequence of sentences $\phi_1, \ldots, \phi_n$. Together this collection of sentences will be termed a *theory*. Since

a theory contains only finitely many sentences, we can actually confine ourselves to theories with a single sentence, ϕ, by setting $\phi = \phi_1 \wedge \cdots \wedge \phi_n$.

Note that by the nature of simulations, corresponding to every statement about the simulation there is a statement about real networks, and conversely. The intention of simulations is that these statements are true or false in exact correspondence with the real network under consideration, however, the accuracy or verisimilitude of the simulations is of no importance in this proposition. All we are concerned to show is the concept of *recursive* model (as in Theorem 35 from (Kleene et al., 1952)) is synonymous with simulation.

Church's thesis, which cannot be proved, but which is generally accepted, is that the concept"recursive" captures the concept "computable", in all its forms. We therefore use these two terms (computable and recursive) synomymously. Since, by Theorem 35 of (Kleene et al., 1952), the models in Godel's theorem may be taken to be *primitive recursive* (a restricted class of recursive functions) *or* general recursive, it is not important whether we require simulations to be primitive recursive or just recursive.

Proposition 1 *For every simulation, S, and language, $\mathcal{L}$, of statements about S, there exists a recursive model, M such that for all sentences $\phi \in \mathcal{L}$, $S \Vdash \phi \Leftrightarrow M \models \phi$. Conversely, for every recursive model, M, $\exists\, S, S \Vdash \phi \Leftrightarrow M \models \phi$.*

Proof

($\Rightarrow$)

Suppose we have a computer program, S, which simulates a network, and the theory T expresses statements about the network, which is simulated by S. As noted above,

each such statement can be interpreted as a statement about the simulation, also , Let us assume the output of the simulation is a sequence of statements about packets, each of which expresses the contents of the packet and what happened to it. Let us now construct a model from the output from the simulation. A model is a collection of sets (including sets of sets, etc.). For each predicate in the language $\mathcal{L}$ in the model, we construct a *set* of n-tuples (which are also set). The n-tuple $\langle 3, 7, 2 \rangle$ is included in the set representing predicate P_1 whenever, in the simulation, node 3 is the source and node 7 the destination, of link 2.

The *domain* of these relations and the corresponding sets needs to include packets and numbers. In statements which refer to times, for example `packet` p `arrived at time` 0.1 the times will be encoded as numbers. We can assume, for example, that the number refers to the number of seconds since the start of the simulation. Packet identities will also be encoded as numbers, and the contents of packets are, of course, already numbers. Similarly, links, nodes, and where necessary ports of nodes will be encoded as numbers. Numbers, in turn, are encoded as sets in the usual way. In this way, every detail of a simulation can be represented as a collection of sets. The sets which represent the packets, nodes, links, etc., make up the domain of the model, and the sets which represent the relations, functions, and constants comprise the remaining features of the model.

This entire collection of sets is a model for the theory T. By construction, if a statement in the language $\mathcal{L}$ is true in the simulation when the statement is interpreted in terms of the model it will also be true, and conversely when a statement constructed in this way is true in the model it must have been true in the simulation. Notice that since this model was derived from the output of a computer simulation, it is clearly computable.

($\Leftarrow$)

Now suppose we are first given a recursive model. This is a collection of sets, just as in the preceding part of the proof. Since the model is computable, it has, in effect, been generated by a computer program. This computer program can be interpreted as a simulation. In particular, it generates sets corresponding to all the predicates, functions, and constants of the language. To render this computer program more conventional, we can encode all the sets of the domain as numbers. (The most famous example of this idea of encoding mathematical objects as numbers is the scheme introduced by Gödel.) The computer program which generates the model might not do so in order of time, the way a conventional simulation would, however, we can extend the computer program to sort the generated details by time to achieve something more conventional, since we are not in the least concerned with efficiency.

Since the computer output generated in this way has been derived from the model, by construction, each statement must be true in the model if and only if it is true in the simulation created from the model. □

Gödel's completeness theorem for simulations (instead of models) says: ϕ can be logically deduced from the theory T if and only if in every simulation where T is true, ϕ is also true, or, formally:

Theorem 1

$$T \vdash \phi \Leftrightarrow T \Vdash \phi.$$

Proof

This result follows from Proposition 1 and Gödel's Completeness Theorem (Barwise, 1982, p35). □

5.5.6 Consistency proof by simulation

Theorem 2 *Corresponding to every simulation in which all the security rules for a network are implemented, there is a model for the system, in the sense of mathematical logic. Hence, one simulation in which all the rules are valid constitutes a proof of the consistency of the rules.*

Proof

This follows from Gödel's completeness theorem together with Proposition 1. There is one complication, however. In Proposition 1, the model is required to be *computable*. (Sometimes *effective* is the adjective used for this property.) Not all models are computable. However, in the present instance, the set of rules being modelled, which are the rules adopted in a network, must be computable. Otherwise, they could not be implemented. A refinement of Gödel's completeness theorem includes the condition that if the theory being modelled is computable, the models referred to in the theorem may also be assumed to be primitive recursive (Kleene et al., 1952, Theorem 35, p394). This resolves the complication. □

This theorem is applied or illustrated in Examples 5.9–5.13, in Section 5.6.

In a sense, a single simulation is no more than an *example* of how a network might operate. One of the most important and dangerous fallacies of informal human reasoning is that one example can prove a statement. This is true of simulations also. Thus, a statement such as *All packets from A to B reach their destination*

cannot be proved by a simulation. In fact, what is needed is the much stronger requirement: *packets from A to B reach their destination in all simulations.* This follows from Theorem I.1 in (Cohen, 1966), if we accept that every model in the sense of logic is also, effectively, a simulation.

However, as Theorem 2 shows, one simulation is sufficient to show consistency of the rules.

In some cases, exploring all possible states will be impossible. However exploring the states which occur most frequently may be adequate, as a *confirmation* of a statement to a satisfactory level. Also, there may be cases where the number of states which need to be explored in order to prove the validity of rules is finite and even relatively small. For example, if all security rules are static, it will not be necessary to explore all states.

5.6 Examples of Validation by Simulation

Examples in this section clarify the relationship between all models in the sense of mathematical logic and simulation models.

Example 5.9 illustrates the use of simulation to check consistency; Example 5.10 shows how simulation can reveal inconsistency; Example 5.11 illustrates that consistency can be trivial, and hence is of little value.

Example 5.12 considers a case where validation by simulation maybe difficult due to the presence of dynamic rules, and, finally, Example 5.13 illustrates a case where *proof* that a network configuration is valid is not possible, nevertheless, network validity is correctly identified by simulation. The point of this last example is not that proof is inferior to simulation, but rather that *seeking a universal strategy for*

proving or disproving validity is a mistaken goal – it is unachievable. In particular, simulation can be applied to any network, but it cannot be expected to prove of disprove validity in all cases.

Example 5.9 Consistency of firewalls

A firewall implemented by procedural rules cannot be inconsistent by itself because whatever the procedure decides to do with a packet is, by definition, the intention of the firewall. However, if we include service protection rules, which assert behaviour that we *expect* of the firewall, inconsistency is possible and needs to be checked. Security policy rules can be checked one-by-one or all at once. Simulation is one way to do this. It might seem that checking by simulation is less rigorous than logical checking, but simulation tools need to implement the logic of firewall rules. Consequently, the only way that simulation might be "inferior" is because of a lower efficiency, ie, simulation is slower. But speed is not an issue in this context.

We can, therefore, apply Theorem 2 to check consistency of the firewall rules, and service protection rules, by simulation.

Examples 5.1 and 5.2 provided above have been checked by simulation for consistency. To do so, we must confirm that during at least one simulation, all the rules required in the model are valid. Table 5.4 shows the throughput of all the traffic streams, in Example 5.1 and Table 5.5 shows the throughput of all the traffic streams, in Example 5.2. These tables show that all rules are satisfied, and hence the firewall and service protection rules are consistent in both examples. □

Example 5.10 Inconsistent Rules

Suppose the operational rules of a network, including all its firewalls, are set up in such a way that one of the network administrators is sometimes unable to access one of the servers that they are required to administer. This is an undesirable configuration. This can be avoided by stating, and checking, a rule which says: "Administrator A can always access server S".

Assuming this rule (a service protection rule), has been added to the system of rules to be checked, a simulation will be able to discover this problem, and the invalidity of the complete set of rules, by checking whether these rules hold and discovering cases where it does not.

Note: a simulation in which administrator A never attempts to access server S would fail to discover this problem. It is therefore essential to add not only the rule that A can always access S, but also the rule that on at least one occasion A does access S. If there are firewall with state-dependent (dynamic) rules it will also be necessary to add a rule that A accesses S during all possible states of the firewalls. □

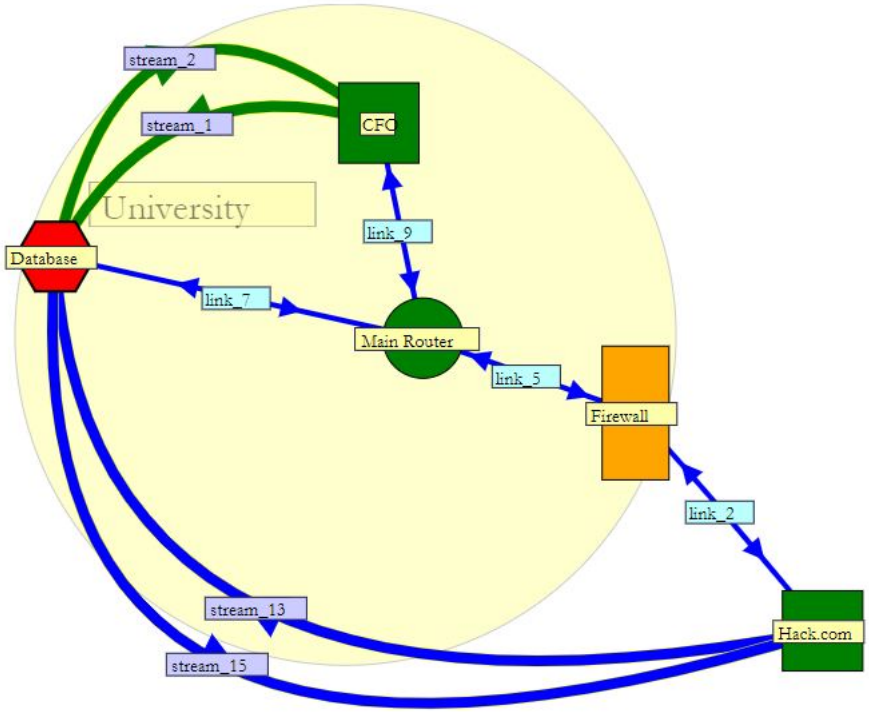

Figure 5.14: A network with simple filtering rules, network for Example 5.11

Example 5.11 Consistency is trivial

Consider the network shown in Figure 5.14. The router and the firewall in this network are expected to implement two rules:

(a) CFO can always access DB;

(b) Hacker can never access DB.

These rules can be shown to be consistent merely by adding an assumption that there is no traffic between CFO and DB and also no traffic between Hacker and DB. This shows that consistency, by itself, does not ensure that a network configuration is valid.

If we add the rules:

(c) There is non-zero traffic between CFO and DB;

(d) There is non-zero traffic between Hacker and DB;

the condition of consistency of the complete set of rules (a)–(d) is no longer trivial. □

Example 5.12 State-dependent rules

Consider an organisation network with a firewall that implements network address translation. In this case, the firewall *state* changes whenever a user inside the network makes a TCP connection to an external server. Consequently, without further consideration, we can't assert that a single simulation in which all rules are valid shows that all rules are always valid, or even that the rules are consistent.

Suppose we add a rule which asserts that during the simulation, all possible states of the firewall have been explored. Under these circumstances it is valid that if all the rules have been true, in one simulation, then the network's system of rules is consistent.

Exploring all possible states of a firewall's NAT table will usually be infeasible due to the large number of possible states. However, exploring every possible state of a NAT table is not necessary. For example, the order in which NAT entries are added to a table should not have any effect on its operation, and the presence of one entry in a NAT table should not affect traffic, which does not have the same source and destination. Taking this into consideration means that the range of states of the NAT table which need, practically, to be explored, is quite small. It should be sufficient to explore states in which NAT applies, or fails to apply, to any traffic which otherwise must appear in the simulation. Exploring all such states is still a non-trivial task, but one which has manageable complexity. □

Example 5.13 Rules which cannot be validated

Consider the network shown in Figure 5.15. The router, R, in this network counts all the packets as they are routed, starting at 0, and routes packets according to their *packet number*. If there exists n bigger than the packet number such that both n and $n+2$ are prime numbers, then it routes the packet to B, otherwise it routes the packet to A.

It has been shown in (Forbes, 1997; Rezgui, 2017) that $2996863034895 \times 2^{1290000} - 1$ and $2996863034895 \times 2^{1290000} + 1$ are both primes. Hence, for all practical purposes, the router R will always route packets to B. However, at the present time there is no proof that there are infinitely many *twin primes*, i.e., two numbers, p, q such that p-q=2, which are both primes (Guy, 2004). Hence there is *no proof* that the router

will always send packets to B. For more than 150 years it has been conjectured but has not been proved that there are infinitely many twin primes(Dijk, 2019).

Supposing that the router is *required* to send all packets to B (i.e., either this is a service protection rule, or it follows from other service protection rules), in this example, proof of the validity of the system is not possible. A simulation, however, will show that the system is valid, i.e., the router will send all packets to B.

In this example, a simulation in which we check whether all rules hold, throughput the simulation, gives us the right answer. This does not *prove* that the system is valid. However, because there is no known proof of the twin prime conjecture, no proof that this system is valid is possible. Thus, seeking a *proof*, in this case, is a waste of time.

Of course, a routing algorithm defined in this way is impractical and unnecessary. However, this nevertheless shows that attempting to develop a *general method* for proving correctness of routing, or any other aspect of networking or security, is a mistake. Proving correctness is not mistaken, but attempting to find *universal* approaches for proving correctness is, as shown in this example, inappropriate.

□

5.7 Summary

Stakeholder security analysis, from Chapter 3 is applied, in this chapter, to networks. The concept of *service protection rules* was also introduced in this chapter and a number of examples were explored, which showed that well-known problems could potentially be better managed if these rules are introduced.

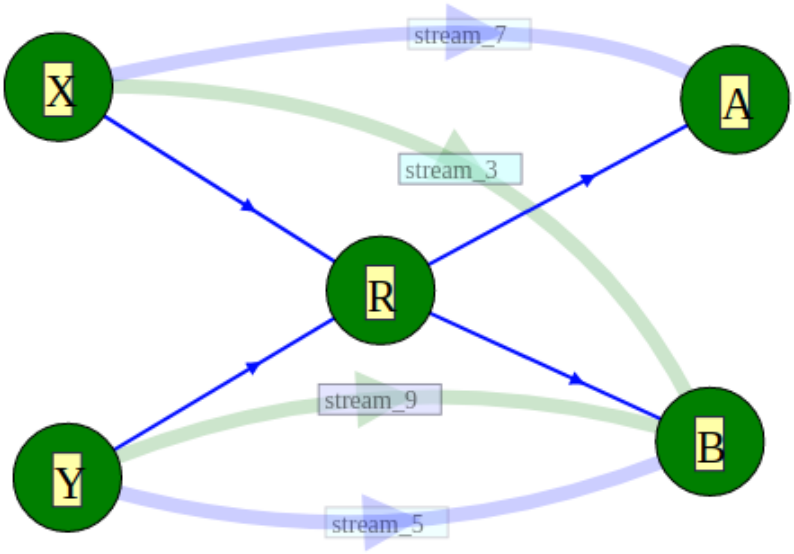

Figure 5.15: A network with an undecidable rule, network for Example 5.13

Table 5.11: URLS of examples

Example	**URL**
Exmple 1	`http://netml.usq.edu.au/netml4_63/index.jsp?netname=filternet3qb3&location=Demo&userid=nabeel`
Examle 2	...
Example 5	...
Example 9	`https://netml.usq.edu.au/netml4_63/index.jsp?netname=CfoDbandHack&location=Demo&userid`

Thirteen network examples were considered in which objectives were identified, and rules to be enforced to achieve these objectives were found. Inference graphs were used to illustrate how the enforced rules ensured that the objectives were met. Several network security policies were improved in this way in this chapter.

Theorem 1 shows that if sufficiently many simulations have been conducted, and the objectives are met in all of them, then the security rules are *valid*, i.e., the rules operate exactly as intended. Strictly speaking, the number of simulations required to ensure validity is infinite – all possible simulations, which is, of course, impossible. However, in the following sense, this theorem can give us confidence in the results of simulations: if our rules are *invalid*, there will be a simulation in which one or more of the rules is false.

The theorem 2 has shown that the consistency of rules can be proved by simulation. Furthermore, the definition of a network model of a network in a way that can be simulated is feasible for real networks.

Chapter 6

Emergency Network Design

6.1 Introduction

A common feature of all serious emergencies is the high risk of loss of human life. The main objective of this chapter is to contribute to the design methodology for emergency networks and show how this methodology helps us to create networks which will reduce the loss of lives during emergencies, by rescuing and recovering as many affected survivors in the emergency, as possible. Before analysing emergency network design or developing any strategies, there are five typical emergency scenarios, some of which are major recent events. In some of these scenarios which many people lost their lives are reviewed in this chapter. So the role of communication technology in emergency response and recovery is well understood.

It is logical to take the objective of system design for an emergency network as the minimization of loss of life. Mobile phones can save lives during emergencies (Queensland Government, 2010).

Outcomes for survivors will obviously benefit if we improve their ability to connect

with others in the emergency area, and especially with authorized emergency workers. In an emergency situation, it may be the case that communication infrastructure, such as mobile phone towers is disabled. Hence, for this reason device-to-device communication features have been introduced in recent standards for mobile communication (Nishiyama et al., 2014). Security design demands to recognize the fact that attackers may be present in the same area. Moreover, we must prevent these attackers from disrupting communication during an emergency or using the emergency communication services and facilities to disrupt ordinary communication when there is no emergency.

In Section 6.2, some broadly typical scenarios for an emergency network are described. In Section 6.3, as described in general terms in Chapter 3 analyse stakeholders of an emergency network, including identification of objectives, enforced rules, and assumptions in Tables 6.2–6.4, respectively, and an inference graph connecting these rules, as shown in Figure 6.2. In Section 6.4, the power management design problem for an emergency network is defined. A preliminary design is outlined and justified. In Section 6.5, some thought experiments in which the design is tested are defined and analysed, and in Section 6.6, the conclusions of the chapter are presented.

6.2 Scenarios

6.2.1 Bush fire: Black Saturday

The bushfires in Victoria, Australia, which took place in and near Victoria, Australia, in 2009 (Commission et al., 2009; Herald), 2009), are among the worst ever to occur in Australia. In the vicinity of 170 deaths, and 400 injuries were incurred, and more than 2000 houses were destroyed. Emergency workers and volunteers affected many rescues.

Key Facts

The number of emergency workers involved in this emergency is approximately 1700; the number of volunteers involved is approximately 20,000.

From (Commission et al., 2009):

> "The flow of information from the fireground to the integrated Emergency Coordination Centre was at times seriously inadequate."

One of the key findings of the royal commission was that orders to evacuate should have been issued, and if such orders had been given, received by those to whom they were directed, acted upon and losses of lives would have been less. The royal commission report did not consider the technical challenge of how the necessary information could be gathered sufficiently quickly, and how the evacuation orders delivered adequately quickly to achieve this result.

6.2.2 Flood: Lockyer Valley and Toowoomba, January 2010

On 10 January 2010, flooding at an unprecedented level occurred in multiple locations in the Lockyer Valley and Toowoomba, in Queensland, Australia (Queensland Government, 2010):

> The flooding in Toowoomba and the Lockyer Valley on 10 January 2010 killed 21 people, 12 of them in Grantham. It was one of the most deadly natural disasters ever to hit Queensland.

Key Facts

Lessons

In both the Victorian bushfires of 2009 and the Queensland flood of 2010,

(a) Communication was identified as a key facility for improving emergency response, and

(b) The contribution of volunteers in response to the emergency was very important.

6.2.3 Earthquake

In this scenario, a *bystander* discovers a collapsed building and uses the emergency network application of their phone. The phone immediately takes action by searching for nearby phones, both to assist survivors and to notify authorities. At this stage, the emergency cannot be officially declared, but there is scope for a rescue to take place.

Let us now suppose that the action of the network discovers the presence of a nearby phone belonging to a survivor in the collapsed building. The survivor's phone notifies the survivor that there has been contact with a possible rescuer.

In particular, the survivor's phone initiates interaction by seeking guidance from the phone owner. The phone owner might: (a) not respond, or might respond by indicating that (b) the owner needs help; (c) the owner does not need help; (d) the owner is not aware of an emergency.

All these alternatives are important, but we first focus on (b), in which case a *normal*

emergency network is in place, but with only two devices. Although the presence of an emergency is not certain, at this stage, there are more evidences that a real emergency exists, since it has been confirmed by two parties. This fact should be taken into account in the subsequent behaviour of the network.

It is possible that the bystander becomes a volunteer and provides assistance to the survivor. What form this takes will not be considered here. However, it is clear that the speed and effectiveness of the rescue can potentially benefit from the ability to communicate with the survivor, which can be affected by the emergency network.

6.2.4 Single person emergency

On 23 October 2015, a walker became lost in the largely uninhabited South-west region of Tasmania, Australia, near Mount Anne (Examiner, 2015b; Police, 2015; News, 2015):

> *Mr Lane-Mullins managed to phone police about 3pm on Thursday and told them he was lost. However, his phone battery died mid-conversation.*

The walker was found, after two nights in the wilderness (Examiner, 2015a; News, 2015; Mercury, 2015). In the aftermath of this emergency, some of the comments by police and emergency worker referred explicitly to the importance of walkers for ensuring that their phones should be fully charged when setting out on a walk. They were concerned, quite reasonably, to avoid the unnecessary expense associated with search and rescue operations.

6.2.5 Hoax or Attack

Any individual, for whatever reason, decides to misuse the emergency network capability of the mobile phones in use in his or her vicinity. This individual is either completely unconcerned with the expectations of his community about the misuse of this network or is actively seeking to cause harm for reasons which do not concern us. We also assume that this individual has access to technical information, and even to security codes or passwords, enabling their abuse of the network to be relatively unrestricted.

An *attack* is a scenario in which we assume that an individual or group is seeking to cause harm to a community and simultaneously attempting to misuse the emergency network facility.

Because of these scenarios, which can not be completely ruled out, it is essential that the emergency-network duty cycle can be triggered, by a message, only when there are careful checks on the authenticity of the message. One way to check message authenticity would be to sign such messages with a certificate assigned to the emergency network managers, and to provide the public key of this certificate in the emergency network software, so that messages can be checked for authenticity.

Because of the attack scenario, even when a message has been authenticated, it is essential that the emergency function can be readily disabled by the phone user.

Triggering of the emergency network behaviour of a phone by a message sent from nearby phones has great potential to increase the benefits of an emergency network, but the risk of this function being misused are considerable, and must be guarded against.

6.2.6 General Observations

Some emergencies – earthquakes for example – occur so quickly that it is not possible to prevent any injuries or deaths by warning affected parties prior to its impact. However, even in these cases, there may be a prolonged period *after* the incident causing the emergency during which communication with affected survivors can reduce the number of casualties. In other cases – floods, tsunamis, and bushfires for example – there is sufficient time after the incident has begun to develop that rapid and precise communication with the individuals in the direct path of the incident can prevent injuries or death. Thus, in all cases, effective communication with individuals in the focal region of an incident has the potential to ameliorate its impact dramatically. Mobile phones can be introduced into the (ad-hoc) emergency network with careful design.

6.3 Emergency Network Stakeholders

In these days, emergency communication networks (ECN) help to exchange communication between people in a disaster or emergency with others inside or outside the area of the emergency. Mobile phones can be used to disseminate basic information such as the presence and location of a survivor, and also can be vital in actions taken to rescue survivors in the most effective way (Hadaad et al., 2016).

The key stakeholders (Ribeiro Soriano et al., 2012) for an emergency network include: (i) affected survivors; (ii) unaffected smart-phone users; (iii) emergency workers, government administrators, and medical staff; (iv) volunteers and bystanders, with potentially relevant skills; (v) families of survivors and bystanders; (vi) telephone companies; (vi) operating system vendors (in particular, Google, Apple, and Microsoft); (vii) hoaxers and attackers.

Each of the stakeholders have a significant role in the operation of the emergency network. Although the contribution of hoaxers and attackers is not helpful, their behaviour must be taken into account, or the role of the network will be severely compromised. In the remainder of this section the roles of these stakeholders is defined their objectives analysed.

There are eight stakeholder roles, and their stakeholders are listed in the first and second columns, respectively, in Table 6.1 and. In addition, stakeholder rules explain in subsection 6.3.2, and the stakeholder scenario is shown in Figure 6.1.

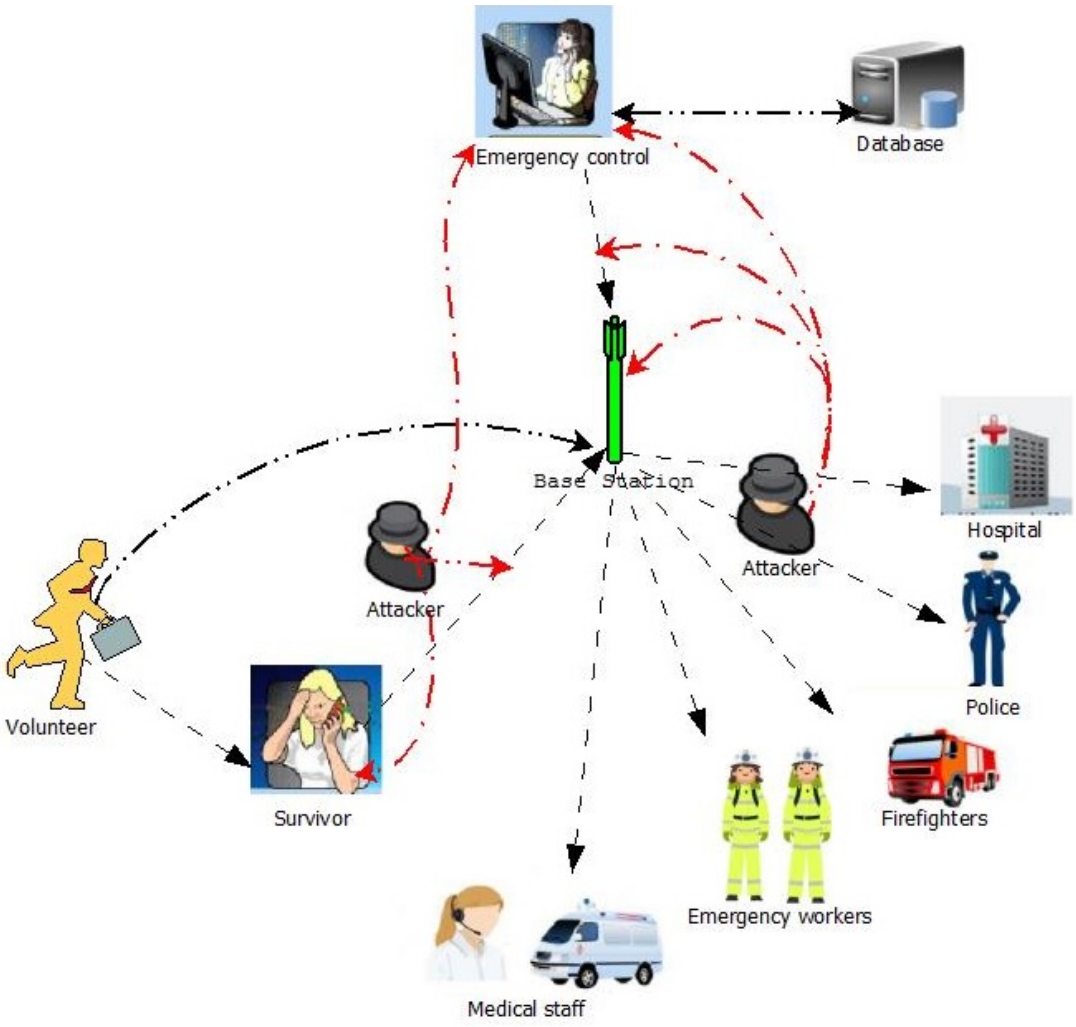

Figure 6.1: Emergency Scenario (with attackers)

6.3.1 Stakeholder roles

Table 6.1: Stakeholder roles for an emergency network

Stakeholder role	**Stakeholder**
User	Affected survivors, Unaffected phone-users
IT Admin	Telephone companies, Operating System Vendors
Emergency central	Emergency workers and officials,police
Guest	Families and friends, Volunteers and bystanders
Attacker	External attacker, Internal attacker (Administration member)

- Survivor. The survivor uses their mobile device to send sms and information about his/her location to emergency centre during few time when disaster happened. he/ she receive sms has first aid before get emergency and medical staff.

- Unaffected phone-users. Misuse, or abuse, of the emergency network technology will have little or no impact on ordinary users.

- Telephone companies. Operating companies should provide all possible support to other operating companies in their need to reconfigure to overcome damage due to the emergency.

- Operating System Vendors. Operating system vendors take responsibility for quality control of applications and phone software.

- Emergency centre (EC). For example: an emergency center will receive sms or location verification code from Survivor and then send all details about survivors to emergency staff and medical staff to rescue victim as soon as.in

addition, EC will update all details about emergency and medical staff with location during all time.

- Emergency workers. Emergency workers receive sms and location verification code from EC and after that they can get Survivor quickly. Emergency works include police, State Emergency Organisation staff, and staff of the Fire Brigade or authority which operates in this area.

- Medical staff. They get authentication when his or her device connects with emergency central and then receives sms from Emergency center or emergency staff to rescue victim. They send sms with sms verification code or call EC after resolve problem.

- Volunteers. Individual citizens who happen to be in the vicinity of an emergency are frequently willing to assist with rescue and aid to survivors. This assistance from volunteers has been found, in many emergencies, to be of great value.

- Attacker. An attacker is any person who enters a communication network and tries to change Survivor information including private details and location. In addition, he/ she might prevent survivors from accessing the communication network to ask any help from emergency centre or emergency and medical staff.

6.3.2 Stakeholder rules

In this subsection we explore how to achieve the objectives O1, O2, O3 and O4 by using enforced rules and assumptions, which are listed in Tables 6.2–6.4. The inference graph of stakeholder rules is shown in Figure 6.2. It shows how the objectives are achieved by the enforced conditions, as justified in the remainder of this subsection.

Here is an explanation of how O1 is achieved, we assume all emergency staff, medical

staff, firefighters and police are not attackers. They and anyone with certified credentials, provide only correct information, by A1 and A2. Survivors can communicate with others in same area, and with emergency professionals, even when infrastructure is damaged, by E3. E11 helps to ensure that friends and families of survivors do not overload the network with inquiries, which may be the most effective way to prevent them from overloading the network. That is why this rule is included as ensuring Objective O1.

Under emergency conditions the power management system cannot prevent the user from using their phone in whatever way they choose (in particular, using it in the usual way), by E9. Also, the power management system extends battery life for client devices as much as possible during emergencies, by E1. Finally, E12, improves the usefulness of the network for rescues and it does it efficiently because the location from which the message is being sent (and any other relevant information) is automatically included even with the first message from a survivor.

Traditionally, infrastructure components (including cellular base stations and WLAN access points) exercise full power control for all active devices. Unfortunately, infrastructure networks have shown to repeatedly fail during emergencies and large disasters. For instance, the Tohoku earthquake (that struck Japan in March 2011) damaged a total of 1.9 million communication lines and about 29,000 cellular base stations while the remaining network was only able to recover more than a month later (Docomo, 2012).

In response, and in order to reduce the dependency to the infrastructure during emergencies, proximate communication between (Deepak et al., 2019) nearby mobile devices is currently being considered for both the upcoming cellular standards through LTE-Direct (Qualcomm Technologies, 2013) and WLAN technologies through WiFi-Direct (WiFi Alliance, 2010). Both standards define two networking phases: a)

the discovery of devices and services, and b) the communication between proximate peers. Device and service availability is broadcast within the periodic beacons (using licensed bands provided by the overlaying base station in the case of LTE-Direct, and over unlicensed channels for WiFi-Direct). In the absence of any infrastructure nodes, this operation is conducted in a distributed fashion to support emergency services.

Clearly, the periodicity of broadcasted beacons governs the responsiveness of the network to changes and the longevity of individual nodes. Higher beaconing frequency improves the dissemination of information across the network, at the expense however of higher battery power consumption. Hence, the beacon periodicity should be optimized to achieve the best trade off (Hunukumbure et al., 2013).

The use of 5G innovations which support direct communication can be used to ensure, E3. By this technology, survivor mobil can connect with another mobel near him. According to Umar Albalawi, two users can connect between them through (D2D) communication in without need base station in emergency area (Albalawi, 2019).

To achieve O2 we use E10 (*any claim that there is an emergency, by a non-authorized party is confirmed by an authorized party before an emergency state of the network is initiated*). The network will therefore not immediately go into an emergency state when an attacker attempts to initiate an emergency. By E5, attackers cannot gain access to any information by any use of emergency services or facilities. And by E8, the power management features of devices cannot be used to modify their functionality unless there is an emergency. In particular, attackers are unable to trigger any significant modification of functions in nearby devices by notifying that there is an emergency. These conditions show that even though attackers can notify the existence of an emergency, this will have no affect on other users (unless there is a real emergency).

To achieve O3, we use E2, to ensure that activity by attackers cannot disrupt communication from authorized participants, like emergency workers, during an emergency. E4 ensures that the communication needs of survivors also cannot be disrupted by attackers. In addition, E5 and E6, ensure that sensitive information is not revealed to attackers during an emergency. These conditions ensure that all the necessary communication continues to take place during an emergency, even when it is under attack.

To achieve O4, we use E7, users may choose to leave the emergency network at any time.

The problem of how to enforce these conditions E1–E13 remains, and is not addressed in this dissertation. In a sense, we have merely classified the way in which the network, during an emergency, must be protected from attack. Nevertheless, this is an important step, and the means for achieving these more specific objectives are, to some degree, more obvious than the original objectives (O1–O4).

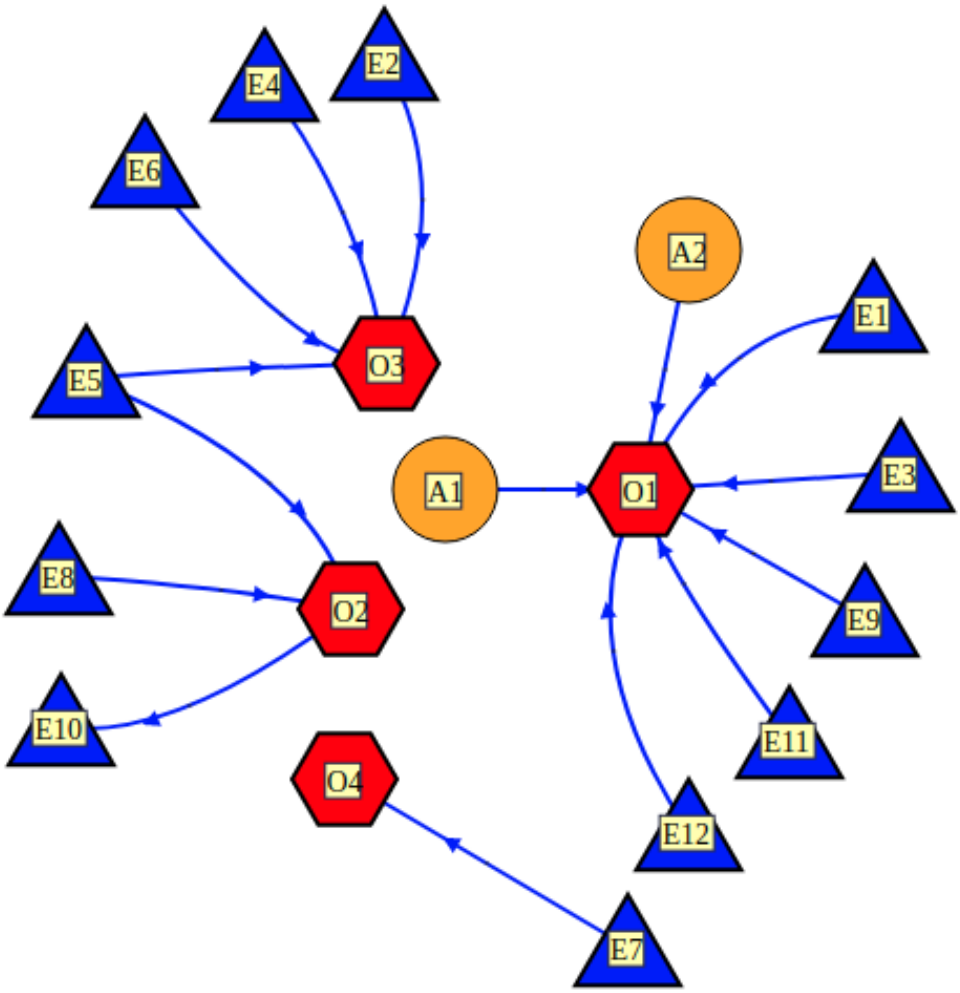

Figure 6.2: Inference graph of stakeholder rules from Tables 6.2–6.4

Table 6.2: Objectives of emergency stakeholders

Rule Name	**Explanation**
O1	Survivors are rescued and assisted at the earliest possible time and in the most effective way.
O2	Fraudulent claims of an emergency cause minimal disruption to the host network.
O3	Fraudulent claims during an emergency cause minimal disruption to the emergency network.
O4	If users are able to use their phone or device in a conventional way to efficiently be rescued, or rescue another survivor, the emergency features of the network can't prevent them from doing so.

Table 6.3: Enforced rules for Emergency networks, from Section 6.3

Rule Name	**Explanation**
E1	Power management extends battery life for client devices as much as possible during emergencies.
E2	Security mechanisms are defined which enable messages from authorities to be clearly identified and which prevent such messages from being modified, or blocked.
E3	Survivors can communicate with others in same area, and with emergency professionals, even when infrastructure is damaged.
E4	No communication activity from an attacker will cause the emergency response for genuine survivors to be significantly compromised.
E5	An attacker cannot use an emergency, or a claimed emergency, to gain access to information which would otherwise be restricted, or to create mis-information.
E6	An attacker cannot modify, delete, resend, or reroute valid messages between Survivors, workers, volunteers, or an emergency centre.
E7	Users may choose to leave the emergency network at any time.
E8	Under normal conditions (no emergency), the power management system cannot be used in such a way that other applications, or the operating system, operate differently.
E9	Under emergency conditions the power management system cannot prevent the user from using their phone in whatever way they choose (in particular, using it in the usual way).
E10	Any claim that there is an emergency, by a non-authorized party, is confirmed by an authorized party before an emergency state of the network is initiated.
E11	Families and friends will receive information about their affected members at the earliest opportunity.
E12	During an emergency, location, time, and any other relevant available information is appended to all messages from survivors.

Table 6.4: Assumptions for emergency networks

Rule Name	**Explanation**
A1	We assume all emergency staff, medical staff, firefighters and police are not attackers.
A2	It is assumed that police and emergency workers, and anyone with certified credentials, provide only correct information.

6.3.3 Service protection rules

Objectives *O1, O2, O3 and O4*, from Table 6.2, are service protection rules for emergency networks. These rules guarantee that survivors can communicate with others in same area, and with emergency professionals, even when infrastructure is damaged, without compromising emergency network security (Hadaad et al., 2015). If include these rules the design of the emergency make it more safety. For example, E9 from Table 6.3, it explains under emergency conditions the power management system cannot prevent the user from using their phone in whatever way they choose (in particular, using it in the usual way) in to ensure O4 is true.

6.4 Power Management Design

This section explores some specific strategies to manage the remaining power in mobile phones more effectively during emergencies (Coyle and Childs, 2005; Hadaad et al., 2016). A key power management strategy in all mobile battery-powered devices is the adoption and adjustment of a *duty-cycle*, i.e. a recurrent pattern of different operation modes, with different power requirements, through which the device progresses. Typically, duty cycles alternate active and sleep cycles of fixed durations but this is not mandatory, and in this chapter duty cycles with dynamically adjusting sleep period durations are analysed.

Since emergencies sometimes require urgent medical attention, there are cases where the preservation of battery life is *not* the most important objective, but instead, maintaining contactability is more important. This we shall call*intense-contact mode*. But there are other cases where urgency is no longer relevant, and another objective is to preserve the phone life. This we shall call *long-life mode*.

As a first and important step in improving power management for mobile phones during emergencies, we explore strategies in which duty-cycle management switches between intense-contact mode and long-life mode. It occur when the battery level reaches a specific threshold.

The effectiveness of the network will rely crucially on its ability to function efficiently for as long as possible. For this reason it is essential to manage the power of the participating devices very carefully. There are a number of ways in which the power consumption of a smart phone or other mobile device can be reduced without significantly reducing its effectiveness as a participant in an emergency network. The question of how to manage power of a mobile device most effectively, in an emergency or disaster, is considered in detail in Sections 6.4 and 6.5.

In line with (3GPP, 2014; Hossain et al., 2019) (which considers LTE enhancements for public safety applications), the overall network design includes security, protocol definitions, smartphone functionality, routing and relaying strategies for control and information packets, and power management strategies. Given the wide range of scenarios to which the emergency network architecture applies, the complexity defining the whole network design will be considerable. Therefore, the primary focus in the remainder of this chapter is on the power management of mobile phones in an ad-hoc emergency network.

Power consumption and battery capacity have retained high priority ever since the initial rollout of smartphones to the market about a decade ago. As indicated in Table 6.5, there is great disparity in the standby and talk times (where both the processor and communication circuitry are active) of these devices. Evidently, the gap is even wider when video playback is considered instead of voice communication since smartphone screens consume similar amounts of power with processing and communication units (as discussed in (Pentikousis, 2010)). The power consumption

Table 6.5: Smartphone power consumption

Device	Battery Capacity	Standby Time	Talk Time
SG S3	2100mAh	Up to 790h	Up to 11h
SG S4	2600mAh	Up to 370h	Up to 17h
SG S5	2800mAh	Up to 390h	Up to 21h
SG S6	2550mAh	Up to 370h	Up to 17h
SG Note 3	3200mAh	Up to 420h	Up to 21h
SG Note 4	3220mAh	Up to 355h	Up to 20h
SG Tab Pro	4800mAh	Up to 750h	Up to 25h

SG: Samsung Galaxy

problem is especially important during emergency situations whereby the "golden 72h" time window applies for rescue operations (Canton and Levy, 2004; James N. Marathas, 2007). As such, the power management has received considerable attention in emerging and future network designs.

6.4.1 A Dynamic Duty Cycle

Typically a duty cycle (Olsen and Narayanaswarni, 2006; Shih and Wang, 2012) is composed of a series of on and off periods. However, in the context of an emergency it may be appropriate to increase the length of the off periods successively, so that battery life can be extended to the greatest degree possible. In this subsection we discuss how such a cycle can be structured. A diagram illustrating a duty cycle with non-constant off periods is given in Figure 6.3.

Listings 6.1–6.4 show the most important functions in a model of fixed and dynamic

duty cycles (Hadaad et al., 2016), which has been developed in R (Simon, 2016) . These comprise approximately half of the total code. (The full code is available from the authors on request.) The remaining code is not relevant to explaining or modeling the duty cycle.

Listing 6.1 is a function which calculates the remaining battery level, at any time, when the system uses a dynamic duty cycle.Listing 6.2 is similar, but includes the feature that the phone uses a fixed duty cycle until a specified battery level is reached, and then uses the dynamic scheme.

Listing 6.3 is a function which uses a built in root-finding function of the language R to find the lifetime of a battery using the dynamic scheme with or without a threshold when the duty cycle becomes dynamic. Finally, Listing 6.4 shows a function which calculates the delay after a search has approached a survivor until the survivor's phone becomes active again.

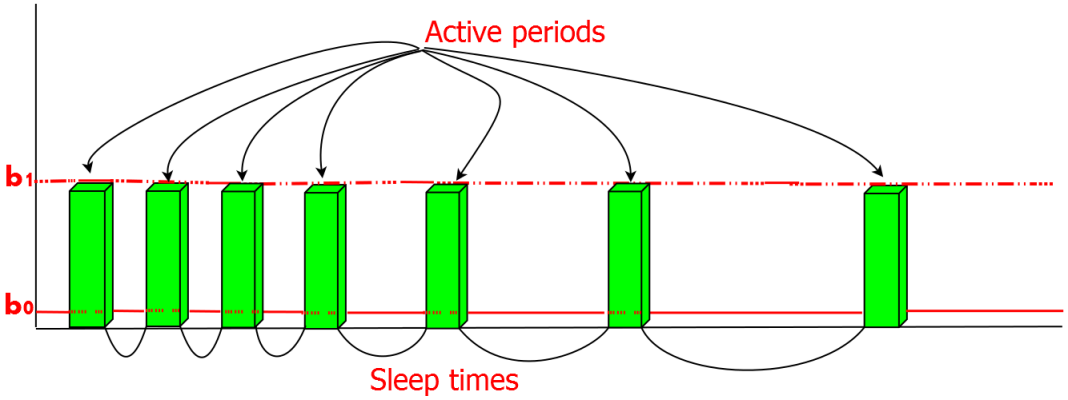

Figure 6.3: Power loss rate as a function of time when the duty-cycle is dynamic (b_0 is the power level during sleep and b_1 power level during active periods)

```
Edynamic = function(t, sleeptime, growthfactor, EnergyRemaining) {
  # energy level at time t assuming dynamic duty cycle, starting
  # at level EnergyRemaining
  if (EnergyRemaining<=0) {
    -10
  } else if (growthfactor==1) {
    Efixed(t, sleeptime, EnergyRemaining)
  } else if (t<sleeptime+activetime) {
    if(t<activetime) {
      EnergyRemaining-t*b1
    } else {
      EnergyRemaining-activetime*b1-(t-activetime)*b0
    }
  } else {
    Edynamic(t-activetime-sleeptime, growthfactor*sleeptime,
           growthfactor, EnergyRemaining-activetime*b1-sleeptime*b0)
  }
}
```

Listing 6.1: R function to calculate the remaining energy in a battery when the duty cycle is dynamic (as illustrated in Figure 6.3)

```
EnergyWithThresh = function(t, sleeptime, growthfactor, thresh,
                            EnergyRemaining) {
  # energy at t, starting at EnergyRemaining, with dynamic duty
  # cycle when battery level falls below thresh
  if (EnergyRemaining<=0) {
    -10
  } else if (EnergyRemaining>thresh && t<sleeptime) {
    EnergyRemaining-(b0*t)
  } else if (EnergyRemaining>thresh && t<activetime+sleeptime) {
    EnergyRemaining-b0*sleeptime-b1*(t-sleeptime)
  } else if (EnergyRemaining>thresh) {
    EnergyWithThresh(t-activetime-sleeptime, sleeptime, growthfactor,
          thresh, EnergyRemaining-(sleeptime*b0+activetime*b1))
  } else {
    if (t<sleeptime+activetime) {
      if(t<activetime) {
        EnergyRemaining-t*b1
      } else {
        EnergyRemaining - activetime*b1 - (t-activetime)*b0
      }
    } else {
      Edynamic(t-activetime-sleeptime, growthfactor*sleeptime,
         growthfactor, EnergyRemaining - activetime*b1-sleeptime*b0)
    }
  }
}
```

Listing 6.2: R function to calculate the remaining energy in a battery when the duty cycle is as dynamic after a certain battery level threshold is reached

```
lifetime = function(growthfactor, u, origsleep, threshfactor) {
  # finds the lifetime of a battery
  Ea = function(t) {
    if (threshfactor>0) {
      EnergyWithThresh(t, sleeptime=origsleep, growthfactor,
          threshfactor*OriginalEnergy, OriginalEnergy)
    } else {
      Edynamic(t, sleeptime=origsleep, growthfactor, OriginalEnergy)
    }
  }
  uniroot(Ea,lower=0,upper=1000000)\$root
}
```

Listing 6.3: R function to calculate the remaining lifetime of battery when the duty cycle is dynamic or becomes dynamic at a certain battery level

```
activeDelay = function(searchtime, sleeptime, growthfactor, thresh,
                       EnergyRemaining, longtime=Year) {
  # the extra time after a delay when the phone will next be active
  if (EnergyWithThresh(searchtime, sleeptime, growthfactor, thresh,
                       EnergyRemaining)<=0) {
    longtime    # a long time
  } else if (searchtime<activetime+sleeptime) {
    activetime+sleeptime - searchtime
  } else if (EnergyRemaining<thresh) {
    activeDelay(searchtime-activetime-sleeptime,
    ↪  growthfactor*sleeptime,
       growthfactor, thresh,
       ↪  EnergyRemaining-(b0*sleeptime+b1*activetime),
       longtime)
  } else {
    activeDelay(searchtime-activetime-sleeptime, sleeptime,
    ↪  growthfactor,
       thresh, EnergyRemaining-(b0*sleeptime+b1*activetime),
       ↪  longtime)
  }
}
```

Listing 6.4: R function to calculate the time delay after a search reaches a survivor before the phone is next active (assuming the duty-cycle becomes dynamic at a certain battery level). See Figure 6.7 for a plot of this function.

6.4.2 Threshold for Alteration of the Duty Cycle

It is intuitively clear that in many emergencies two modes of operation are likely to be useful: intensive-contact mode, and long-life mode. Once the transition has been made to long-life mode it is unlikely that the intensive-contact mode will be needed again, and so the only design choice we need to make is *when* the switch should be made from one to the other. For example, it is plausible that an optimal strategy for choosing when to introduce a non-trivial duty cycle must take this form: we need to identify a parameter which grows over time, and we need to choose a value of this parameter (the threshold), at which time (i.e. when the observed parameter reaches this value) the duty cycle is initiated.

6.5 Experiments

In this section, we present five experiments to explain how we can extend battery life for a smartphone to improve their contribution to the saving of human lives.

It seems reasonable to hypothesise that: (a) lifetimes during and for several days after emergencies are severely shortened, and (b) there are cases where survivors remain at risk, and in urgent need of assistance, for several days.

In these five experiments, key performance parameters are estimated and plotted as a function of certain design parameters. This study can then use these plots to choose the best value for the design parameters against which the performance is plotted.

All of the experiments have been written in the language R, which is effective for the type of models developed, and for creating the graphical representation of results appropriate for them.

In the first experiment, described in Subsection 6.1, power level is evaluated for the dynamic duty cycle as function of two parameters. In the second experiment, described in Subsection 6.2, battery life is plotted as a function to evaluate the battery life. In the third experiment, described in Subsection 6.3, dynamic adjustment of the sleep duration and a threshold for the battery life. In the fourth experiment, described in Subsection 6.4, the delay till a phone is next active. In the fifth experiment, described in Subsection 6.5, assumptions are the survival time till rescue and delay till arrival of aid of the survivors.

Experiment 6.1 Power level

In this experiment the remaining power level is evaluated for the dynamic duty cycle as a function of the two key parameters – the initial idle-time and the idle-time growth rate. The algorithm which evaluates the energy level when a dynamic duty cycle is used is shown in Listing 6.1. The Figure 6.4 shows a plot of the energy level for various choices of parameters. □

Experiment 6.2 Battery Life

In this experiment the lifetime of the phone is shown, assuming dynamic adjustment of the sleep duration. See Figure 6.5. The algorithm used to evaluate the lifetimes is shown in Listing 6.3, which uses in turn the algorithm shown in Listing 6.1 to evaluate the battery life. □

Experiment 6.3 A threshold for Dynamic sleep cycles

In this experiment the lifetime of the phone is shown, assuming dynamic adjustment of the sleep duration and a threshold for the battery life when dynamic adjustment

begins. See Figure 6.6. The algorithm used to evaluate the lifetimes is shown in Listing 6.3, which, in this case, in turn, the algorithm shown in Listing 6.2 to evaluate the battery life. □

Experiment 6.4 Delay till next active cycle

In this experiment we estimate the delay till a phone is next active, as a function of the time when the search for the phone starts. The R code to determine the additional time after the search has reached a survivor, till the phone is active again, is shown in Listing 6.4. This function is plotted, for various choices of the growth factor, in Figure 6.7. When the growth factor is 1, the sleep duration remains 1 for the entire experiment, and hence the battery expires after approximately one day. This is why the blue curve in Figure 6.7 rises vertically quite early. □

Experiment 6.5 Estimating lives saved

In this experiment we estimate lives "saved"(in the sense of communicating with them) under a various choices for the power-management strategy and under variety of assumptions concerning the survival time till rescue and delay till arrival of aid of the survivors

We assume that there is an initial sleep-duration of 1 minute, in all cases. Once dynamic adjustment of the duty cycle starts, the sleep-time increases by the growth-rate, which is the x-axis in Figures 6.8 and 6.9. The transition from the fixed sleep duration to the dynamic one occurs when the battery level reaches a certain proportion of the full battery life. The legend shows this threshold, as a proportion of the original battery life, for each of the plots in the figure. When the threshold is zero, it plays no role, i.e. there is no threshold in this case. Also, if the growth factor

is 1, it plays no role, so the duty cycle is fixed. Finally, the case where no duty cycle is used at all is shown as the last curve, in Figures 6.8 and 6.9.

The survivors were modelled in this experiment by drawing random numbers from either a gamma distribution with scale parameter 10000 and shape parameter 0.3, or from a log normal distribution with mean, of the logs, 6, and standard-deviation of the logs 2.1 (and hence, mean of the lifetimes is 2 days and 13 hours). Both the log-normal distribution and the gamma distribution are concentrated on the positive numbers are moderately heavy-tailed, and fits the situation where there are significant numbers of both small and large numbers in the population. Experiments confirmed that the choice of distribution type is not a major influence on the results.

The rescue process was also modelled by drawing random numbers, from the same distribution, to represent the time taken by the search to come near to a survivor. The survivor is deemed to be rescued if the time taken to reach the next active period of the survivor's phone, after the search has reached the survivor, is less than their lifetime without aid. This is why, in the experiments presented in Section 6.5, we adopt a distribution of survivor lifetimes until assistance which has a dramatically reduced mean, relative to normal lifetimes.

Figure 6.9 shows the proportion of survivors saved, according to this criterion. In this particular experiment, 2000 survivors were simulated.

□

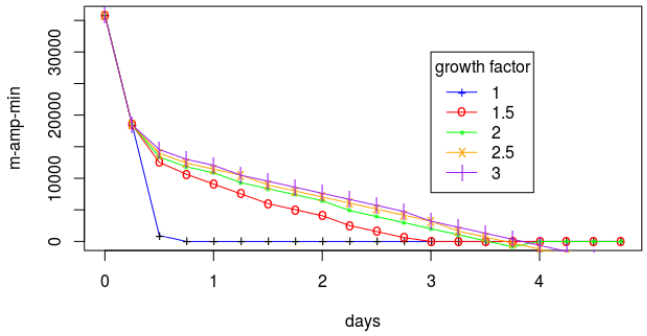

Figure 6.4: Remaining energy as a function of time and growth factor

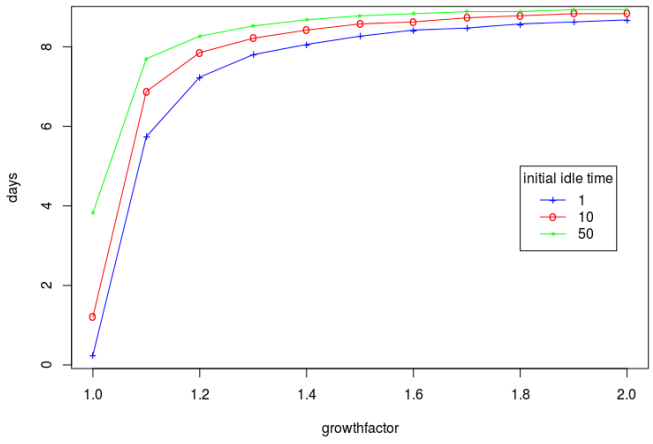

Figure 6.5: Phone lifetime as a function of growth factor and initial idle-time

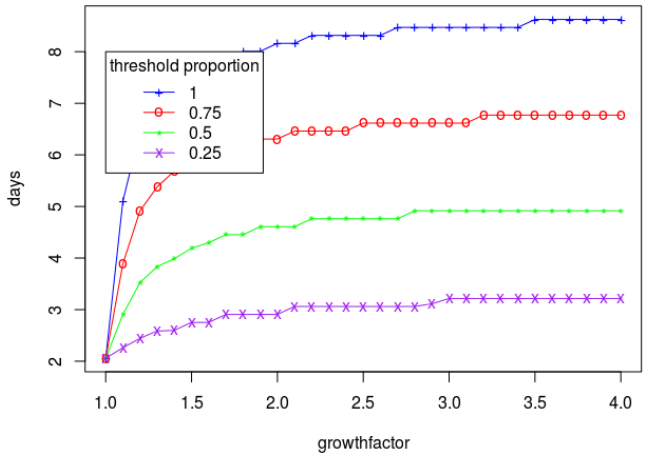

Figure 6.6: Phone lifetime as a function of growth factor and threshold for switching to a dynamic duty cycle

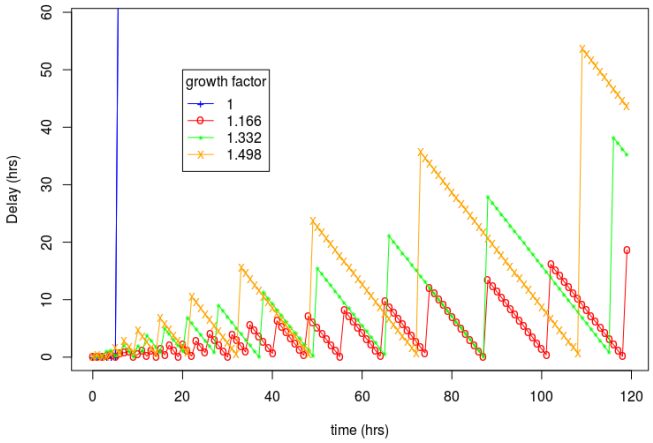

Figure 6.7: Delay till next active cycle as a function of the time of searching for a phone, for various choices of sleep-time growth factor

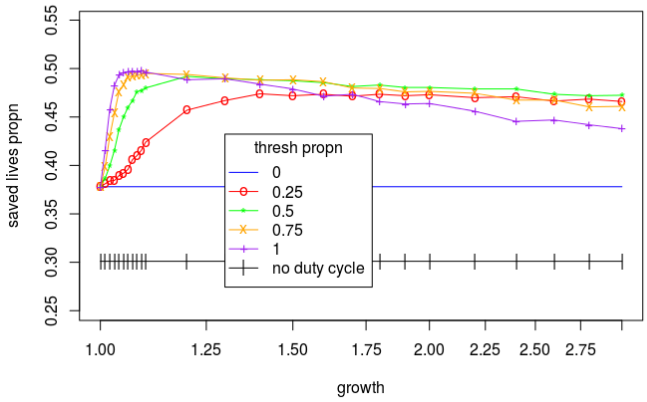

Figure 6.8: Proportion of lives saved under different choices of sleep duration growth and threshold for dynamic duty cycle (in this case the distribution of lifetimes is log-normal)

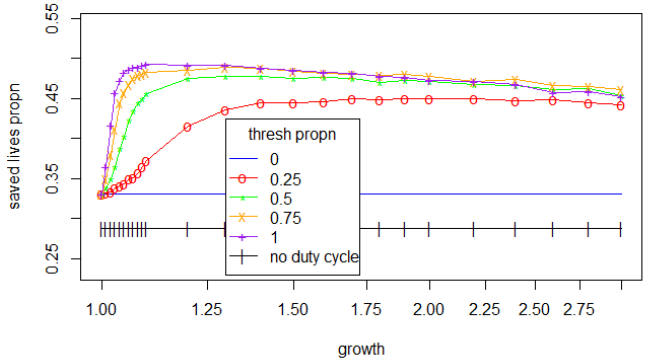

Figure 6.9: Proportion of lives saved under different choices (with gamma distribution)

6.6 Conclusion

6.6.1 Summary

In this chapter, stakeholder security analysis for emergency networks was carried out, including identifying the objectives of an emergency network and identification of specific rules to be enforced to achieve those objectives. The experiments in Section 6.5 have shown that with improved power management, the duration of time over which a mobile phone can remain useful and actively involved in rescue and recovery of those affected by an emergency can be dramatically extended.

Particular attention was paid to power management of mobile devices and the design of a power management system which achieves the objectives found in the stakeholder security analysis was developed.

6.6.2 Recommendations

It has been shown that the objectives of an emergency network based on user devices can be met if the conditions listed in Table 6.3 are enforced. Thefore, all of the rules listed in this table are recommended. Some of these rules are quite natural and others are less obvious. For example, it is recommended that all messages sent by survivors, during an emergency, should include their location, the time, and any other relevant information. If the device will be entering a duty cycle where it is only able to receive messages during certain times, in the future, the plan for this duty cycle should be included in each message, to make it easier to communicate with this device in future.

The power management strategies (Hadaad et al., 2016) explored can only be implemented and successfully deployed if the following innovations or something similar

are effected:

1. Changes are made to device power management systems;

2. Devices should include emergency network software which enables an *Emergency power management state* to be enabled and managed;

3. The power management system includes a dynamic duty cycle to achieve maximum extension of useful battery life during emergencies and the following recovery phase.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

This thesis has argued that the following new ideas are useful and important: (a) there is a logical security design philosophy, which is to identify the objectives required by all stakeholders, and ensure they are guaranteed; (b) we can ensure that the objectives all hold by enforcing certain rules of a simpler nature and then proving that the objectives must hold from these enforced rules.

For systems which need a security guarantee, it may be feasible to prove that all stakeholder rules hold. However, many systems rely on correct behaviour by the stakeholders, which will limit what level of security can be achieved. This reliance on good user behaviour can be modelled by adopting *assumptions*. Realistic assumptions can simplify the logical analysis of security without compromising security any further than is necessary or appropriate. However, it is always the case that weak assumptions reduce security and therefore avoiding such assumptions in the design of a system is a way to improve security.

For logical completeness it is necessary to state and ensure rules which express the desired behaviour of any system being analysed or designed. Such rules do not necessarily refer to risks or problems at all. Such rules have been termed, in this dissertation, *service protection rules*. Including these rules in the analysis appears to lead to more satisfactory designs. For example, in Subsection 4.2.3 of Chapter 4, Subsection 5.3.3 of Chapter 5 and Subsection 6.3.3 of Chapter 6.

The use of stakeholder security analysis in the web service system was able to achieve the desired objectives and thereby to make the web service more secure and to provide effective and efficient services.

It is reasonable to ask each stakeholder (or their representative): "is this list of rules sufficient, if guaranteed, for you to agree to actively participate in this service?" If all stakeholders are satisfied in this way, and these rules are guaranteed by the system's design, the web service is secure.

If a user, nevertheless, experiences a problem such as loss of personal data or password theft, they may realise that the list of rules they agreed to was incomplete, and will need to be revised.

Despite an agreement of this sort being established, naturally, if a new approach which undermines security in a way which one or more stakeholders were not able to anticipate is discovered, by an event or by a security audit carried out by the system designers, the rules which stakeholders require will need to change. At least, when this happens, we have an explanation: our fundamental understanding of the nature of the service has progressed.

The use of stakeholder rules analysis in security of networks also leads to achieving objectives of security network design. In some cases stakeholder rules analysis determines service protection rules that can be used to improve the design and change

management of ICT facilities like firewalls. In addition, analysing the combination of stakeholder rules and service protection rules may assist to avoid logical flaws in security design which otherwise might be regarded as unavoidable costs of good security. Since stakeholder rules and service protection rules are rules which can potentially be defined in the same language and context as security rules, they are well suited to enhancing the design, change management, and measurement of the performance of network security.

The use of stakeholder security analysis in emergency networks has achieved an emergency network design which meets the desired objectives. A set of twelve rules were identified which achieve these objectives. Some of these rules are quite natural, and others are not so obviously necessary. For example, it was identified that messages from survivors should always include the time, location, and possibly information about power management. Another example is that it was found that users should always be able to use their devices in the usual way, even during an emergency or disaster.

Furthermore, the stakeholder security analysis helped to find rules of power management which maximize lives saved during and after emergencies. The utility of mobile devices may be limited by battery life. The usefulness of these devices can be enhanced by extending battery life. One way to extend battery life is to introduce a duty cycle, i.e. a pattern of use where the phone is alternately sleeping and active.

Better management of power in mobile phones can save lives. Furthermore, it has been shown that in order to achieve this it will probably be necessary to introduce the unconventional concept of a dynamic duty cycle, which initially adopts a fixed duration sleep state, and when battery level falls to a certain level begins to increase the duration of the sleep state by a fixed growth ratio at each occurrence.

Algorithms which allow the battery life in a phone making use of a dynamic duty

cycle, with a threshold, to be modelled have been developed and used to estimate the best choice of dynamic duty cycle parameters. Experiments, using two different models for the lifetimes of survivors and of times taken to find survivors, showed that the most effective setting for the threshold when the duty cycle begins to grow, and for the growth rate after it begins growing, was for the growth to begin immediately (threshold=1), and for the growth rate to be slow enough (about 1.1) that the battery lasts nearly as long as possible (5-7 days).

In conclusion, this thesis explored some strategies of stakeholder security analysis to address some challenges in security network, emergency network and information and communication technology in general. Therefore, this work helps to improve the design of security of networks and other ICT systems.

7.2 Future Work

- To determine more specific rules of a web service – in particular, of the Netml system – which ensure that information generated or gathered by any user is not accessible to any other user, except as specified by the original user.
- To investigate the specification of objectives concerning user sharing of data generated by users of a web service and the specific rules which need to be enforced to ensure that these objectives are met.
- To define additional service protection rules for networks which prevent or minimize any problems and increase the effectiveness of their security design.
- To develop more specific technical specifications for how to protect an emergency network from an attack during an emergency (for example: under what circumstances should private data with the potential to assist survivors be encrypted)

- To define more specific protocols which enforce the rules proposed to apply to an emergency communication network, including its devices and participants, to ensure that survivors are rescued and receive treatment and care as quickly and efficiently as possible.

References

3GPP (2014). Study on architecture enhancements to support proximity-based services (prose). Technical Report TR 23.703 V12.0.0 (2014-02) (Release 12), 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TTA, TTC).

Abadi, M., Burrows, M., Lampson, B., and Plotkin, G. (1993). A calculus for access control in distributed systems. *ACM Trans. Program. Lang. Syst.*, 15(4):706–734.

Addie, R. (2010). Netml v4. 0–an online environment for access and sharing of network data and software for network analysis and design. Technical report, Tech. Rep. SC-MC-1001, University of Southern Queensland.

Addie, R., Braithwaite, S., and Zareer, A. (2006). Netml: a language and website for collaborative work on networks and their algorithms. In *Proceedings of the Australian Telecommunication Networks and Applications Conference (ATNAC 2006)*, pages 1–5. University of Melbourne.

Addie, R. and Colman, A. (2010). Five criteria for web-services security architecture. In *Network and System Security (NSS), 2010 4th International Conference on*, pages 521–526.

Addie, R., Moffatt, S., Dekeyser, S., and Colman, A. (2011a). Five examples of web-services for illustrating requirements for security architecture. In *2nd International Conference on Data and Knowledge Engineering (ICDKE 2011)*, pages 47–54.

Addie, R. G., Peng, Y., and Zukerman, M. (2011b). Netml: networking networks.

In *Dependable, Autonomic and Secure Computing (DASC), 2011 IEEE Ninth International Conference on*, pages 1055–1060. IEEE.

Aggarwal, P., Gonzalez, C., and Dutt, V. (2020). Hackit: A real-time simulation tool for studying real-world cyberattacks in the laboratory. In *Handbook of Computer Networks and Cyber Security*, pages 949–959. Springer.

Albalawi, U. (2019). A device-to-device system for safety and emergency services of mobile users. *IEEE Consumer Electronics Magazine*, 8(5):42–45.

Alfaro, J. G., Boulahia-Cuppens, N., and Cuppens, F. (2008). Complete analysis of configuration rules to guarantee reliable network security policies. *International Journal of Information Security*, 7(2):103–122.

Ali, R. S. and Alaa, K. F. (2018). Security protocol of keys management system for transmission encrypted data. *International Journal of Computer Network and Information Security*, 10(1):10.

Almorsy, M., Grundy, J., and Müller, I. (2016). An analysis of the cloud computing security problem. *arXiv preprint arXiv:1609.01107*.

Awal, A., Mishra, A., Joseph, E., and AbdulG, A. (2018). A power management model for android operated smart phones. *IJRP.ORG*, 3(1):15–15.

Badger, L., Sterne, D. F., Sherman, D. L., Walker, K. M., Haghighat, S., et al. (1995). Practical domain and type enforcement for unix. In *Security and Privacy, 1995. Proceedings., 1995 IEEE Symposium on*, pages 66–77. IEEE.

Baez, J. C., Foley, J., and Moeller, J. (2019). Network models from Petri Nets with catalysts. *Compositionality*, 1:4.

Bandara, A. K., Lupu, E. C., and Russo, A. (2003). Using event calculus to formalise policy specification and analysis. In *Policies for Distributed Systems and Networks, 2003. Proceedings. POLICY 2003. IEEE 4th International Workshop on*, pages 26–39. IEEE.

Barwise, J. (1982). *Handbook of mathematical logic*. Elsevier.

Bauer, L., Ligatti, J., and Walker, D. (2002). More enforceable security policies. In *Proceedings of the Workshop on Foundations of Computer Security (FCS), Copenhagen, Denmark*, pages 95–104. Citeseer.

Beaujean, A. A. (2013). Factor analysis using R. *Practical assessment, research, and evaluation*, 18(1):4.

Bhardwaj, A. and Goundar, S. (2018). Reducing the threat surface to minimise the impact of cyber-attacks. *Network Security*, 2018(4):15–19.

Billgren, C. and Holmén, H. (2008). Approaching reality: Comparing stakeholder analysis and cultural theory in the context of natural resource management. *Land Use Policy*, 25(4):550–562.

Bishop, M. A. (2005). *Introduction to computer security*, volume 50. Addison-Wesley Boston.

Bourne, L. (2016). *Stakeholder relationship management: a maturity model for organisational implementation*. Routledge.

Breslau, L., Estrin, D., Fall, K., Floyd, S., Heidemann, J., Helmy, A., Huang, P., McCanne, S., Varadhan, K., Xu, Y., et al. (2000). Advances in network simulation. *Computer*, 33(5):59–67.

Brodie, C., Karat, C.-M., Karat, J., and Feng, J. (2005). Usable security and privacy: a case study of developing privacy management tools. In *Proceedings of the 2005 symposium on Usable privacy and security*, pages 35–43. ACM.

Canton, L. and Levy, D. (2004). Disaster preparedness in a changing world.

Caulfield, T. and Pym, D. (2015). Improving security policy decisions with models. In *In Proc. IEEE Security & Privacy, Special Issue, SPSI: Economics of Cybersecurity, 2015*. RISCS.

Cohen, P. J. (1966). *Set theory and the continuum hypothesis*. W. A. Benjamin, Inc.

Commission, V. B. R. et al. (2009). The 2009 victorian bushfires royal commission final report summary. *Retrieved July*, 8:2013.

Cori, R. and Lascar, D. (2000). *Mathematical Logic: A course with exercises*. Oxford univeristy Press, Great Clarendon Street, Oxford OX2 6DP.

Coyle, D. and Childs, M. (2005). The role of mobiles in disasters and emergencies. *London: GSM Association*.

Damianou, N., Dulay, N., Lupu, E., and Sloman, M. (2000). A language for specifying security and management policies for distributed systems. *London: Department of Computing, Imperial College, Tech. Rep*.

De Santis, A., Castiglione, A., Fiore, U., and Palmieri, F. (2013). An intelligent security architecture for distributed firewalling environments. *Journal of Ambient Intelligence and Humanized Computing*, 4(2):223–234.

Deepak, G., Ladas, A., and Politis, C. (2019). Robust device-to-device 5g cellular communication in the post-disaster scenario. In *2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, pages 1–6. IEEE.

Dijk, S. (2019). Twin primes: classical results and new developments. B.S. thesis, University of Twente.

Diver, S. (2007). Information security policy–a development guide for large and small companies. *Sans Institute*, pages 1–37.

Docomo, N. (2012). Measures for recovery from the great east japan earthquake using ntt docomo r&d technology. *NTT DOCOMO Technical Journal*, 13(4):96–106.

Erdheim, S. (2013). Deployment and management with next-generation firewalls. *Network Security*, 2013(10):8–12.

Erdogan, G. (2009). Security testing of web based applications. Master's thesis, Institutt for datateknikk og informasjonsvitenskap.

ETSI (2014). GSC task force on emergency communications (GSC-EM). Technical report, European Telecommunications Standards Institute (ETSI).

Examiner (2015a). Found: Searchers find bushwalker.

Examiner (2015b). Missing bushwalker near Mount Anne.

Faily, S. (2015). Engaging stakeholders during late stage security design with assumption personas. *Information & Computer Security*.

Flechais, I. and Sasse, M. A. (2009). Stakeholder involvement, motivation, responsibility, communication: How to design usable security in e-science. *International Journal of Human-Computer Studies*, 67(4):281–296.

Fong, P. W. and Siahaan, I. (2011). Relationship-based access control policies and their policy languages. In *Proceedings of the 16th ACM symposium on Access control models and technologies*, pages 51–60. ACM.

Forbes, A. (1997). A large pair of twin primes. *Math. Comput.*, 66:451–455.

Ghirardello, K., Maple, C., Ng, D., and Kearney, P. (2018). Cyber security of smart homes: Development of a reference architecture for attack surface analysis. In *Living in the Internet of Things: Cybersecurity of the IoT-2018*, pages 1–10. IET.

Glasgow, J., Macewen, G., and Panangaden, P. (1992). A logic for reasoning about security. *ACM Trans. Comput. Syst.*, 10(3):226–264.

Grimble, R. and Wellard, K. (1997). Stakeholder methodologies in natural resource management: a review of principles, contexts, experiences and opportunities. *Agricultural systems*, 55(2):173–193.

Guelev, D. P., Ryan, M., and Schobbens, P. Y. (2004). Model-checking access control policies. In *Information Security*, pages 219–230. Springer.

Gupta, S. G., Ghonge, M. M., Thakare, P. D., and Jawandhiya, P. (2013). Open-

source network simulation tools: An overview. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, 2(4):1629–1635.

Guy, R. K. (2004). *Unsolved problems in number theory*. Springer.

Hadaad, N., Drury, L., and Addie, R. G. (2015). Protecting services from security mis-configuration. In *Telecommunication Networks and Applications Conference (ITNAC), 2015 International*, pages 120–125. IEEE.

Hadaad, N., Pitsillides, A., Kolios, P., Kuras, A., and Addie, R. G. (2016). Emergency network design-saving lives by saving power. In *2016 26th International Telecommunication Networks and Applications Conference (ITNAC)*, pages 19–21. IEEE.

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

Hamed, H., Al-Shaer, E., and Marrero, W. (2005). Modeling and verification of ipsec and vpn security policies. In *Network Protocols, 2005. ICNP 2005. 13th IEEE International Conference on*, pages 259–278. IEEE.

Hamelin, M. (2010). Preventing firewall meltdowns. *Network Security*, 2010(6):15–16.

Hanauer, T., Hommel, W., Metzger, S., and Pöhn, D. (2018). A process framework for stakeholder-specific visualization of security metrics. In *Proceedings of the 13th International Conference on Availability, Reliability and Security*, pages 1–10.

Herald), F. M. S. M. (Retrieved 11 February 2009). Horrific, but not the worst we've suffered.

Hossain, M., Ray, S., and Shahamiri, S. (2019). A context-aware and technology-assisted informal caregiver selection method to support medical emergency. In *2019 29th International Telecommunication Networks and Applications Conference (ITNAC)*, pages 4–9. IEEE.

Hunukumbure, M., Moulsley, T., Oyawoye, A., Vadgama, S., and Wilson, M. (2013). D2d for energy efficient communications in disaster and emergency situations. In *2013 21st International Conference on Software, Telecommunications and Computer Networks-(SoftCOM 2013)*, pages 1–5. IEEE.

JAIDI, F. (2019). Fw-tr: Towards a novel generation of firewalls based on trust-risk assessment of filtering rules and policies. In *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*, pages 1043–1048. IEEE.

Jajodia, S., Samarati, P., Sapino, M. L., and Subrahmanian, V. (2001). Flexible support for multiple access control policies. *ACM Transactions on Database Systems (TODS)*, 26(2):214–260.

James N. Marathas, Paul J. Alfano, R. S. (2007). A canton citizenshandbook for emergency preparedness.

Jasser, S. (2019). Constraining the implementation through architectural security rules: An expert study. In *International Conference on Product-Focused Software Process Improvement*, pages 203–219. Springer.

Jøsang, A. and Pope, S. (2005). User centric identity management. In *AusCERT Asia Pacific Information Technology Security Conference*, page 77. Citeseer.

Kalam, A., Baida, R., Balbiani, P., Benferhat, S., Cuppens, F., Deswarte, Y., Miege, A., Saurel, C., and Trouessin, G. (2003). Organization based access control. In *Policies for Distributed Systems and Networks, 2003. Proceedings. POLICY 2003. IEEE 4th International Workshop on*, pages 120–131.

Karjoth, G. and Schunter, M. (2002). A privacy policy model for enterprises. In *Computer Security Foundations Workshop, 2002. Proceedings. 15th IEEE*, pages 271–281.

Kaya, E., Agca, M., Adiguzel, F., and Cetin, M. (2019). Spatial data analysis with r programming for environment. *Human and ecological risk assessment: An International Journal*, 25(6):1521–1530.

Kleene, S. C., De Bruijn, N., de Groot, J., and Zaanen, A. C. (1952). *Introduction to metamathematics*, volume 483. van Nostrand New York.

Kohler, E. (2001). *The Click Modular Router*. PhD thesis, Massachusetts Institute of Technology.

Kohler, E., Morris, R., Chen, B., Jannotti, J., and Kaashoek, M. F. (2000). The click modular router. *ACM Trans. Comput. Syst.*, 18(3):263–297.

Koppel, R., Blythe, J., Kothari, V., and Smith, S. (2016). Beliefs about cybersecurity rules and passwords: A comparison of two survey samples of cybersecurity professionals versus regular users. In *Twelfth Symposium on Usable Privacy and Security ({SOUPS} 2016)*.

Kropiwiec, C. D., Jamhour, E., Penna, M. C., and Pujolle, G. (2011). Multi-constraint security policies for delegated firewall administration. *International journal of network management*, 21(6):469–493.

Kumar, A., Kaushik, S. K., Sharma, R., and Raj, P. (2012). Simulators for wireless networks: A comparative study. In *2012 International Conference on Computing Sciences*, pages 338–342. IEEE.

Lai, H., Hsu, J. S.-C., and Wu, M.-X. (2018). The impact s of requested permission on mobile app adoption: The insights based on an experiment in taiwan. In *Proceedings of the 51st Hawaii International Conference on System Sciences*.

Law, D. R. (1998). Scalable means more than more: a unifying definition of simulation scalability. In *Simulation Conference Proceedings, 1998. Winter*, volume 1, pages 781–788. IEEE.

Lupu, E. C. and Sloman, M. (1999). Conflicts in policy-based distributed systems management. *Software Engineering, IEEE Transactions on*, 25(6):852–869.

Maguire, B., Potts, J., and Fletcher, S. (2012). The role of stakeholders in the marine planning processstakeholder analysis within the solent, united kingdom. *Marine Policy*, 36(1):246–257.

Mailloux, L. O., Beach, P. M., and Span, M. T. (2018). Examination of security design principles from nist sp 800-160. In *Systems Conference (SysCon), 2018 Annual IEEE International*, pages 1–8. IEEE.

Manin, Y. I. (1977). *Graduate texts in mathematics: A course in mathematical logic*. Springer-Verlag, 175 Fifth Avenue, New York, New York 10010, USA.

Mayer, A., Wool, A., and Ziskind, E. (2000). Fang: a firewall analysis engine. In *Security and Privacy, 2000. S P 2000. Proceedings. 2000 IEEE Symposium on*, pages 177–187.

Maynard, S., Ruighaver, A., and Ahmad, A. (2011). Stakeholders in security policy development. In *9th Australian Information Security Management Conference*, page 182. Citeseer.

Mercury, H. (2015). Bushwalker found alive and well after two cold nights in tasmania.

Nan Zhang, M. D. R. and Guelev, D. (2005). Evaluating access control policies through model checking. In *Eighth Information Security Conference (ISC'05)*., pages 446–460. Springer-Verlag.

Nentwich, C., Capra, L., Emmerich, W., and Finkelsteiin, A. (2002). xlinkit: A consistency checking and smart link generation service. *ACM Transactions on Internet Technology (TOIT)*, 2(2):151–185.

News, S. (2015). Search continues for missing bushwalker.

Nishiyama, H., Ito, M., and Kato, N. (2014). Relay-by-smartphone: realizing multihop device-to-device communications. *IEEE Communications Magazine*, 52(4):56–65.

OASIS (2010). Oasis extensible access control markup language (XACML) TC. `http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml`.

Olsen, C. M. and Narayanaswarni, C. (2006). Powernap: An efficient power man-

agement scheme for mobile devices. *Mobile Computing, IEEE Transactions on*, 5(7):816–828.

P., L. S. and Merz, R. (2011). Ns-3-click: Click modular router integration for ns-3. In *Wns3*.

Pentikousis, K. (2010). In search of energy-efficient mobile networking. *IEEE Communications Magazine*, 48(1):95–103.

Police, T. (2015). Update lost bushwalker near mount anne.

Pozo, S., Gasca, R. M., Reina-Quintero, A., and Varela-Vaca, A. J. (2012). Confiddent: A model-driven consistent and non-redundant layer-3 firewall acl design, development and maintenance framework. *Journal of Systems and Software*, 85(2):425–457.

Qualcomm Technologies, I. (2013). Lte direct: The case for device-to-device proximate discovery.

Queensland Government (2010). Grantham floods commission of inquiry. `http://www.granthaminquiry.qld.gov.au/`.

Queensland Government (2012). Queensland floods commission of inquiry. `http://www.granthaminquiry.qld.gov.au/`.

Ranchal, R., Bhargava, B., Angin, P., and b. Othmane, L. (2019). Epics: A framework for enforcing security policies in composite web services. *IEEE Transactions on Services Computing*, 12(3):415–428.

Reed, M. S., Graves, A., Dandy, N., Posthumus, H., Hubacek, K., Morris, J., Prell, C., Quinn, C. H., and Stringer, L. C. (2009). Who's in and why? a typology of stakeholder analysis methods for natural resource management. *Journal of environmental management*, 90(5):1933–1949.

Rezgui, H. (2017). Conjecture of twin primes (still unsolved problem in number theory) an expository essay. *Surveys in Mathematics and its Applications*, 12:229–252.

Ribeiro, C., Zuquete, A., and Ferreira, P. (2000). Security policy consistency. *Technical Report, INESC*.

Ribeiro, C., Zuquete, A., Ferreira, P., and Guedes, P. (2001). Spl: An access control language for security policies and complex constraints. In *NDSS*, pages 89–107.

Ribeiro Soriano, D., Wagner Mainardes, E., Alves, H., and Raposo, M. (2012). A model for stakeholder classification and stakeholder relationships. *Management Decision*, 50(10):1861–1879.

Riley, G. and Henderson, T. (2010). The ns-3 network simulator. In Wehrle, K., Güne, s, M., and Gross, J., editors, *Modeling and Tools for Network Simulation*, pages 15–34. Springer Berlin Heidelberg.

Rose, K. H. (2013). A guide to the project management body of knowledge (pmbok® guide)fifth edition. *Project management journal*, 44(3):e1–e1.

Ross, R. S., McEvilley, M., and Oren, J. C. (2018). Systems security engineering: Considerations for a multidisciplinary approach in the engineering of trustworthy secure systems. Technical report, NIST.

Sabelfeld, A. and Myers, A. C. (2003). Language-based information-flow security. *Selected Areas in Communications, IEEE Journal on*, 21(1):5–19.

Samonas, S., Dhillon, G., and Almusharraf, A. (2020). Stakeholder perceptions of information security policy: Analyzing personal constructs. *International Journal of Information Management*, 50:144–154.

Savage, G. T., Nix, T. W., Whitehead, C. J., and Blair, J. D. (1991). Strategies for assessing and managing organizational stakeholders. *Academy of management perspectives*, 5(2):61–75.

Schneider, F. B. (2000). Enforceable security policies. *ACM Transactions on Information and System Security (TISSEC)*, 3(1):30–50.

Scholl, H. J. (2005). Interoperability in e-government: More than just smart middle-

ware. In *System Sciences, 2005. HICSS'05. Proceedings of the 38th Annual Hawaii International Conference on*, pages 123–123. IEEE.

Sheniar, D., Hadaad, N., and Addie, R. (2019). The inference graph of cybersecurity rules. In *2019 29th International Telecommunication Networks and Applications Conference (ITNAC)*, pages 3–8. IEEE.

Sheniar, D., Hadaad, N., Martin, D., Addie, R., and Abdullah, S. (2018). Experiments and proofs in web-service security. In *2018 28th International Telecommunication Networks and Applications Conference (ITNAC)*, pages 1–6. IEEE.

Shi, L. and Chadwick, D. W. (2011). A controlled natural language interface for authoring access control policies. In *Proceedings of the 2011 ACM Symposium on Applied Computing*, pages 1524–1530. ACM.

Shih, H.-C. and Wang, K. (2012). An adaptive hybrid dynamic power management algorithm for mobile devices. *Computer Networks*, 56(2):548–565.

Simon, S. D. (2016). R for big data analysis. *Big Data Analysis for Bioinformatics and Biomedical Discoveries*, page 35.

Siraj, S., Gupta, A., and Badgujar, R. (2012). Network simulation tools survey. *International Journal of Advanced Research in Computer and Communication Engineering*, 1(4):199–206.

Stallings, W. and Brown, L. (2015). *Computer security: principles and practice*. Pearson Education Upper Saddle River, NJ, USA, 3 edition.

Tabassum, M., Watson, S., Chu, B., and Lipford, H. R. (2018). Evaluating two methods for integrating secure programming education. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, pages 390–395. ACM.

Thomsen, D. and Bertino, E. (2018). Network policy enforcement using transactions: The neutron approach. In *Proceedings of the 23nd ACM on Symposium on Access Control Models and Technologies*, pages 129–136. ACM.

Thong, W. J. and Ameedeen, M. (2015). A survey of petri net tools. In *Advanced Computer and Communication Engineering Technology*, pages 537–551. Springer.

Toch, E., Bettini, C., Shmueli, E., Radaelli, L., Lanzi, A., Riboni, D., and Lepri, B. (2018). The privacy implications of cyber security systems: A technological survey. *ACM Computing Surveys (CSUR)*, 51(2):36.

Trabelsi, Z. and Saleous, H. (2019). Exploring the opportunities of cisco packet tracer for hands-on security courses on firewalls. In *2019 IEEE Global Engineering Education Conference (EDUCON)*, pages 411–418. IEEE.

Varga, A. (2010). Omnet++. In *Modeling and tools for network simulation*, pages 35–59. Springer.

Verma, D. C. (2002). Simplifying network administration using policy-based management. *Network, IEEE*, 16(2):20–26.

Weingartner, E., Vom Lehn, H., and Wehrle, K. (2009). A performance comparison of recent network simulators. In *2009 IEEE International Conference on Communications*, pages 1–5. IEEE.

Whitman, M. E. and Mattord, H. J. (2011). *Principles of information security*. Cengage Learning.

Whittaker, J., Haynes, K., Handmer, J., and McLennan, J. (2013). Community safety during the 2009 australian black saturdaybushfires: an analysis of household preparedness and response. *International journal of wildland fire*, 22(6):841–849.

WiFi Alliance (2010). Wi-fi peer-to-peer (p2p) technical specification v1.1.

Wolf, R. S. (2005). *A tour through mathematical logic*. Mathematical Association of America, Washington, DC 20090-1112.

Wu, W., Mayo, G., McCuen, T. L., Issa, R. R., and Smith, D. K. (2018). Building information modeling body of knowledge. i: Background, framework, and initial development. *Journal of Construction Engineering and Management*, 144(8):04018065.

Yang, C., Zhu, Z., Huang, W., Yang, C., and Zhang, W. (2009). Application of simulation technology in reliability measure of ad hoc network. In *Reliability, Maintainability and Safety, 2009. ICRMS 2009. 8th International Conference on*, pages 1137–1140. IEEE.

Zhang, N. (2005). *Generating Verified Access Control Policies through Model Checking*. PhD thesis, University of Birmingham. `http://www.cs.bham.ac.uk/~hxq/acpeg.php`.

Zhang, N., Ryan, M., and Guelev, D. P. (2004). Synthesising verified access control systems in xacml. In *Proceedings of the 2004 ACM workshop on Formal methods in security engineering*, pages 56–65. ACM.

Zhang, N., Ryan, M., and Guelev, D. P. (2005). Evaluating access control policies through model checking. In *Information Security*, pages 446–460. Springer.

Zhang, N., Ryan, M., and Guelev, D. P. (2008). Synthesising verified access control systems through model checking. *Journal of Computer Security*, 16(1):1–61.

Zhang, Y., Liang, R., and Ma, H. (2012). Teaching innovation in computer network course for undergraduate students with packet tracer. *IERI Procedia*, 2:504–510.

Zinsmaier, S. D., Langweg., H., and Waldvogel., M. (2020). A practical approach to stakeholder-driven determination of security requirements based on the gdpr and common criteria. In *Proceedings of the 6th International Conference on Information Systems Security and Privacy - Volume 1: ICISSP,*, pages 473–480. INSTICC, SciTePress.