ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Semester thesis for Stefan Hildenbrand

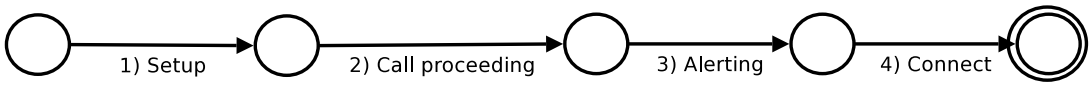
Generation of Test Cases

Programming of a tool for the generation of test cases from finite automata

Supervisor: Diana Senn
Professor: Prof. D. Basin
Issue Date: August 2005
Submission Date: February 2006

1 Introduction

We live in a world where all the company networks are connected to the Internet. Nobody can control the Internet, therefore a company has to protect their data from unauthorised access through the Internet. This is done by firewalls whose analogon in the physical world are locks. Everybody understands that doors need to be locked to prevent unauthorised access. It is the same in the digital world: unauthorised access to a company's network should be prevented, and this can be done by one or several firewalls.
Using the analogon of the door lock again, everybody understands that it is not enough to have a door lock. Only if the lock is locked properly and only authorised people have got a key to unlock it, we have what we want. It is the same in the digital world. It is not enough to have a firewall. We can only be satisfied if the firewall is doing what we expect from it. And to find out if a firewall satisfies our expectations (stated by a policy) we need to test it.



1) Setup

Source	A	
Destination	B	
Header	Field Name:	Protocol discriminator
	Field Value:	00001000
	Field Name:	Call reference
	Field Value:	X
	Field Name:	Message type
	Field Value:	00000101
	Field Name:	Sending complete
	Field Value:	[optional, Range = ?]
	...	
Payload	...	

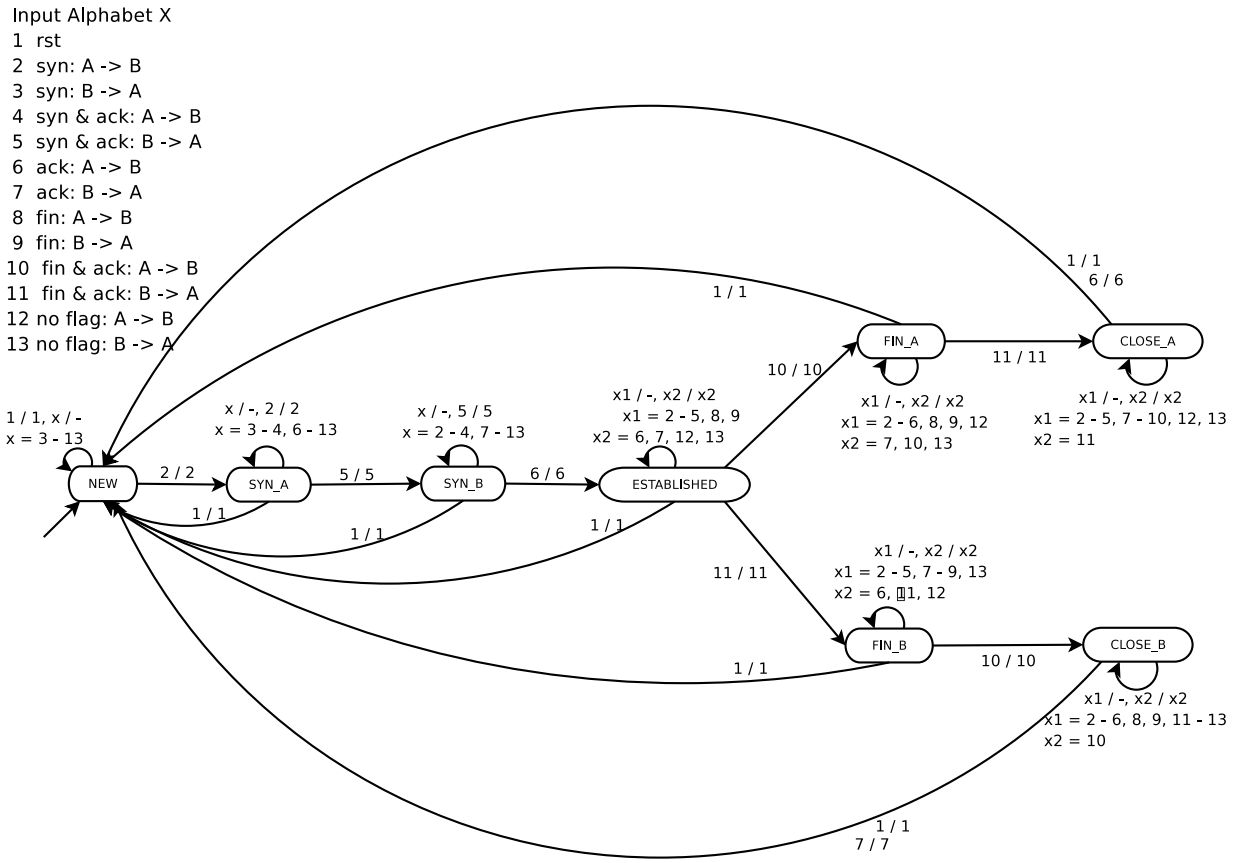
2) Call proceeding

Source	B	
Destination	A	
	Field Name:	Protocol discriminator
	Field Value:	00001000
	Field Name:	Call reference
	Field Value:	X
	Field Name:	Message type
	Field Value:	00000010
	Field Name:	Bearer capability
	Field Value:	[optional, Range = ?]
	...	
Payload	...	

Figure 1: Automaton for Simple Call Establishement in H.323

2 Motivation

When testing firewalls, one of many things that needs to be tested is the correct stateful handling of various protocols by the firewall. To do this, one needs a specification of the protocol which is to be tested. Such a specification can be written as a finite automata, see figures 1 and 2 for examples. From such automata, test cases can then be generated, using different methods [1, 2, 4, 5, 6, 7], and run against the firewall.



Some explanations:
I / O on transitions means Input and Output respectively.
The outputs are the reaction of a firewall to the given inputs. This is either accepting (forwarding) a packet or dropping it.
"x1 = 2-6" means $x1 \in 2..6$

Figure 2: Automaton for tcp

3 Assignement

3.1 Objectives

The goal of this project is to implement a tool which converts a graphical representation of an automaton into abstract test cases. These abstract test cases will then be instantiated with test tuples to generate concrete test cases [8], which then can be fed to fwtest [9].

3.2 Tasks

- Define criteria the tool has to satisfy (together with the supervisor)
- Evaluate tools for the graphical specification of automata, e.g. [3].
- Adapt the best suited tool to our needs
- Write a converter (for the tool chosen) between graphical and textual specifications of finite automata
- Generate abstract test cases from a textual representation of a finite automaton using an algorithm given by the supervisor

The whole software written during this thesis should rely on open source software (if possible) and should be modular and extendable. Particularly it should be easily possible to later on extend the software by other test generation algorithms.

3.3 Deliverables

- At the beginning of the semester thesis an agreement must be signed which allows the supervisor of this thesis, his project partners and ETH Zurich to use and distribute the software written during the thesis.
- At the end of the second week, a detailed time schedule of the semester thesis must be given and discussed with the supervisor.
- At the end of the semester thesis a presentation of 20 minutes must be given during an Infsec group seminar. It should give an overview as well as the most important details of the work.
- The final report may be written in English or German. It must contain an abstract written in both English and German, this assignement and the schedule. It should include an introduction, an analysis of related work, and a complete documentation of all used software tools. Three copies of the final report must be delivered to the supervisor.
- Software and configuration scripts developed during the thesis must be delivered to the supervisor on a CD-ROM.

16th August 2005

Prof. D. Basin

References

[1] Wendy Y. L. Chan, Son T. Vuong, and M. Robert Ito. An improved protocol test generation procedure based on UIOS. pages 283–294, 1989.

[2] Tsun S. Chow. Testing software design modeled by finite-state machines. In *IEEE Transactions on Software Engineering, Vol. SE-4, No 3*, pages 178–187, May 1978.

[3] Vincent Tscherter et al. Exorciser. http://www.educeth.ch/informatik/exorciser/.

[4] Susumu Fujiwara, Gregor von Bochmann, Ferhat Khendek, Mokhtar Amalou, and Abderrazak Ghedamsi. Test selection based on finite state models. *IEEE Trans. Softw. Eng.*, 17(6):591–603, 1991.

[5] A. Gill. State-identification experiments in finite automata. In *Information and Control, vol. 4*, pages 132 – 154, 1961.

[6] A. Gill. *Introduction to the Theory of Finite-state Machines*. McGraw-Hill, 1962.

[7] Krishan Sabnani and Anton Dahbura. A protocol test generation procedure. In *Computer Networks and ISDN Systems 15*, pages 285–297, 1988.

[8] Diana Senn, David Basin, and Germano Caronni. Firewall conformance testing. In Ferhat Khendek and Rachida Dssouli, editors, *Proceedings of TestCom 2005*, volume 3502 of *Lecture Notes in Computer Science*, pages 226–241. Springer-Verlag GmbH, May 2005.

[9] Gerry Zaugg. Firewall testing. http://www.infsec.ethz.ch/people/dsenn/DA_GerryZaugg_05.pdf.