Abstract

This is one of the more complex specifications, the catch is that most of the complexity is in the model!

1 Simple Firewall

1.1 The Specification

MODULE *Firewall*

EXTENDS *Naturals*
CONSTANT *AcceptList*,
RejectList,
Protocols,
IPs,
Ports

VARIABLE *forwarded* packets forwarded on through the firewall

---

$Address \triangleq [ip : IPs,\ port : Ports]$ address has port and ip

A Packet has a source and destination address, and a protocol

$Packet \triangleq [source : Address,\ dest : Address,\ protocol : Protocols]$

$Rule \triangleq [Packet \to \text{BOOLEAN}\ ]$ A Rule maps from packets to boolean values

---

$Init \triangleq forwarded = \{\}$

$Reject(p) \triangleq$
$\quad \exists\, rule \in RejectList :$
$\quad\quad \land\, rule[p]$

$Accept(p) \triangleq$
$\quad \exists\, rule \in AcceptList :$
$\quad\quad \land\, rule[p]$

$Firewall(p) \triangleq$ Firewall acts on a packet
$\quad$ IF $Accept(p) \land \neg Reject(p)$

THEN $forwarded' = forwarded \cup \{p\}$
ELSE UNCHANGED $forwarded$

$Next \triangleq$
$\exists\, packet \in Packet : Firewall(packet)$

Modification History
Last modified Tue Feb 16 16:16:29 GMT 2021 by alunm
Created Sat Feb 13 18:27:34 GMT 2021 by alunm

2 The Model

2.1 Model Overview

The Behaviour specification is an *Initial predicate and next-state relation*

Init
Next

The Model Is quite complex due to the nature of the problem and the data required.

Declared constants the constants in the specification are declared as follows.

Accept list A list of functions that match packets

$AcceptList \leftarrow \{$
$[p \in Packet \mapsto p.protocol = \text{“http”}],$
$[p \in Packet \mapsto p.dest.port = 4],$
$[p \in Packet \mapsto p.source.ip \in 1 \,..\, 3]$
$\}$

RejectList is the list of rules to match rejected packets.

$RejectList \leftarrow \{\}$

Protocols A set of strings of protocol names to match.

$Protocols \leftarrow \{\text{“http”}, \text{“smtp”}, \text{“ftp”}\}$

Ports The range of ports in use

$Ports \leftarrow 1 \mathinner{..} 5$

IPs The range if IP addresses to use

$IPs \leftarrow 1 \mathinner{..} 10$

2.2 Checks and verifications

The invariant is checked to see if all the packets that have been forwarded are either accepted or not expicitly rejected.

$\forall\, p \in forwarded : Accept(p) \land \neg Reject(p)$

2.3 Results

A summary of the results

Statistics a summaries of the actions and number of states found.

States found	7501
Distinct states	1

Action	Location	States Found	**Distinct states**
Init	Line 23	1	1
Firewall	Line 33	7500	0

2.4 Discussion

2.4.1 Model description

This is a complex model. In this case we have to model and specify the environment in which the specification operates.

The state of the system is... The state of the firewall is just the contents of the ACL, packets coming into the firewall are not really part of the firewall state, but are part of the environment state in which the firewall operates. The *forwarded* and *declined* variables model this state of the environment.

Rules The firewall uses rules to check packets and accept or reject them if they match. A rule can be thought of as a predicate function on the packets, i.e. it is a function that maps from packets to a boolean value.

It's type is therefore:

$Rule \triangleq [Packet \to \text{BOOLEAN}\ ]$

The accept-list and reject-list are simply sets of rules.

ACL The access control list is a set of accepting and rejecting rules. These are constants from the model. This could be a variable and managed by operations, but it adds nothing to the firewall behaviour.

Accept and Reject rules A packet is accepted (or rejected) if there is a rule that matches the packet, in other words

$\exists\, rule \in AcceptList : rule[p]$ This is true, if there is a rule that is true, and a rule is true if it matches a packet. A rule might be:

$[p \in Packet \mapsto p.dest.port = 8080]$ Which is true if the port field of the dest field in the packet record is 8080, and false otherwise.

Firewall Action The action of the fire wall is to forward a packet if it is in the accept list and not matched by a reject rule.

$Firewall(p) \triangleq$ Firewall acts on a packet

IF $Accept(p) \wedge \neg Reject(p)$

THEN $forwarded' = forwarded \cup \{p\}$

ELSE UNCHANGED $forwarded$

Next action The next action is that there is a packet that can be forwarded.

$Next \triangleq$

$\exists\, packet \in Packet : Firewall(packet)$

2.4.2 Interpretation of results

This is an interesting case in that the firewall itself has no state, all the states come from the environment, packets received, forwarded and rejected. It is another system where care is needed in determining the size of the model and consequently the size of the state-space.