

Verified Firewall Ruleset Verification

with Isabelle/HOL

Cornelius Diekmann



Introduction to Firewalls

Chain INPUT (policy ACCEPT)

target	prot	source	destination	
DOS_PROTECT	all	0.0.0.0/0	0.0.0.0/0	
ACCEPT	all	0.0.0.0/0	0.0.0.0/0	state RELATED,ESTABLISHED
DROP	tcp	0.0.0.0/0	0.0.0.0/0	tcp dpt:22
DROP	tcp	0.0.0.0/0	0.0.0.0/0	multiport dports 21,873,5005,5006,80,548,...
DROP	udp	0.0.0.0/0	0.0.0.0/0	multiport dports 123,111,2049,892,5353
ACCEPT	all	192.168.0.0/16	0.0.0.0/0	
DROP	all	0.0.0.0/0	0.0.0.0/0	

Chain DOS_PROTECT (1 references)

target	prot	source	destination	
RETURN	icmp	0.0.0.0/0	0.0.0.0/0	icmptype 8 limit: avg 1/sec burst 5
DROP	icmp	0.0.0.0/0	0.0.0.0/0	icmptype 8
RETURN	tcp	0.0.0.0/0	0.0.0.0/0	tcp flags:0x17/0x04 limit: avg 1/sec burst 5
DROP	tcp	0.0.0.0/0	0.0.0.0/0	tcp flags:0x17/0x04

Introduction to Firewalls

Chain **INPUT** (policy ACCEPT)

target	prot	source	destination	
DOS_PROTECT	all	0.0.0.0/0	0.0.0.0/0	
ACCEPT	all	0.0.0.0/0	0.0.0.0/0	state RELATED,ESTABLISHED
DROP	tcp	0.0.0.0/0	0.0.0.0/0	tcp dpt:22
DROP	tcp	0.0.0.0/0	0.0.0.0/0	multiport dports 21,873,5005,5006,80,548,...
DROP	udp	0.0.0.0/0	0.0.0.0/0	multiport dports 123,111,2049,892,5353
ACCEPT	all	192.168.0.0/16	0.0.0.0/0	
DROP	all	0.0.0.0/0	0.0.0.0/0	

Chain DOS_PROTECT (1 references)

target	prot	source	destination	
RETURN	icmp	0.0.0.0/0	0.0.0.0/0	icmptype 8 limit: avg 1/sec burst 5
DROP	icmp	0.0.0.0/0	0.0.0.0/0	icmptype 8
RETURN	tcp	0.0.0.0/0	0.0.0.0/0	tcp flags:0x17/0x04 limit: avg 1/sec burst 5
DROP	tcp	0.0.0.0/0	0.0.0.0/0	tcp flags:0x17/0x04

Introduction to Firewalls

Chain INPUT (policy ACCEPT)

target	prot	source	destination	
DOS_PROTECT	all	0.0.0.0/0	0.0.0.0/0	
ACCEPT	all	0.0.0.0/0	0.0.0.0/0	state RELATED,ESTABLISHED
DROP	tcp	0.0.0.0/0	0.0.0.0/0	tcp dpt:22
DROP	tcp	0.0.0.0/0	0.0.0.0/0	multiport dports 21,873,5005,5006,80,548,...
DROP	udp	0.0.0.0/0	0.0.0.0/0	multiport dports 123,111,2049,892,5353
ACCEPT	all	192.168.0.0/16	0.0.0.0/0	
DROP	all	0.0.0.0/0	0.0.0.0/0	

Chain DOS_PROTECT (1 references)

target	prot	source	destination	
RETURN	icmp	0.0.0.0/0	0.0.0.0/0	icmptype 8 limit: avg 1/sec burst 5
DROP	icmp	0.0.0.0/0	0.0.0.0/0	icmptype 8
RETURN	tcp	0.0.0.0/0	0.0.0.0/0	tcp flags:0x17/0x04 limit: avg 1/sec burst 5
DROP	tcp	0.0.0.0/0	0.0.0.0/0	tcp flags:0x17/0x04

Introduction to Firewalls

Chain INPUT (policy ACCEPT)

target	prot	source	destination	
DOS_PROTECT	all	0.0.0.0/0	0.0.0.0/0	
ACCEPT	all	0.0.0.0/0	0.0.0.0/0	state RELATED,ESTABLISHED
DROP	tcp	0.0.0.0/0	0.0.0.0/0	tcp dpt:22
DROP	tcp	0.0.0.0/0	0.0.0.0/0	multiport dports 21,873,5005,5006,80,548,...
DROP	udp	0.0.0.0/0	0.0.0.0/0	multiport dports 123,111,2049,892,5353
ACCEPT	all	192.168.0.0/16	0.0.0.0/0	
DROP	all	0.0.0.0/0	0.0.0.0/0	

Chain DOS_PROTECT (1 references)

target	prot	source	destination	
RETURN	icmp	0.0.0.0/0	0.0.0.0/0	icmptype 8 limit: avg 1/sec burst 5
DROP	icmp	0.0.0.0/0	0.0.0.0/0	icmptype 8
RETURN	tcp	0.0.0.0/0	0.0.0.0/0	tcp flags:0x17/0x04 limit: avg 1/sec burst 5
DROP	tcp	0.0.0.0/0	0.0.0.0/0	tcp flags:0x17/0x04

Introduction to Firewalls

Chain INPUT (policy ACCEPT)

target	prot	source	destination	
DOS_PROTECT	all	0.0.0.0/0	0.0.0.0/0	
ACCEPT	all	0.0.0.0/0	0.0.0.0/0	state RELATED,ESTABLISHED
DROP	tcp	0.0.0.0/0	0.0.0.0/0	tcp dpt:22
DROP	tcp	0.0.0.0/0	0.0.0.0/0	multiport dports 21,873,5005,5006,80,548,...
DROP	udp	0.0.0.0/0	0.0.0.0/0	multiport dports 123,111,2049,892,5353
ACCEPT	all	192.168.0.0/16	0.0.0.0/0	
DROP	all	0.0.0.0/0	0.0.0.0/0	

Chain DOS_PROTECT (1 references)

target	prot	source	destination	
RETURN	icmp	0.0.0.0/0	0.0.0.0/0	icmptype 8 limit: avg 1/sec burst 5
DROP	icmp	0.0.0.0/0	0.0.0.0/0	icmptype 8
RETURN	tcp	0.0.0.0/0	0.0.0.0/0	tcp flags:0x17/0x04 limit: avg 1/sec burst 5
DROP	tcp	0.0.0.0/0	0.0.0.0/0	tcp flags:0x17/0x04

Introduction to Firewalls

Chain INPUT (policy ACCEPT)

target	prot	source	destination	
DOS_PROTECT	all	0.0.0.0/0	0.0.0.0/0	
ACCEPT	all	0.0.0.0/0	0.0.0.0/0	state RELATED,ESTABLISHED
DROP	tcp	0.0.0.0/0	0.0.0.0/0	tcp dpt:22
DROP	tcp	0.0.0.0/0	0.0.0.0/0	multiport dports 21,873,5005,5006,80,548,...
DROP	udp	0.0.0.0/0	0.0.0.0/0	multiport dports 123,111,2049,892,5353
ACCEPT	all	192.168.0.0/16	0.0.0.0/0	
DROP	all	0.0.0.0/0	0.0.0.0/0	

Chain **DOS_PROTECT** (1 references)

target	prot	source	destination	
RETURN	icmp	0.0.0.0/0	0.0.0.0/0	icmptype 8 limit: avg 1/sec burst 5
DROP	icmp	0.0.0.0/0	0.0.0.0/0	icmptype 8
RETURN	tcp	0.0.0.0/0	0.0.0.0/0	tcp flags:0x17/0x04 limit: avg 1/sec burst 5
DROP	tcp	0.0.0.0/0	0.0.0.0/0	tcp flags:0x17/0x04

Introduction to Firewalls

Chain INPUT (policy ACCEPT)

target	prot	source	destination	
DOS_PROTECT	all	0.0.0.0/0	0.0.0.0/0	
ACCEPT	all	0.0.0.0/0	0.0.0.0/0	state RELATED,ESTABLISHED
DROP	tcp	0.0.0.0/0	0.0.0.0/0	tcp dpt:22
DROP	tcp	0.0.0.0/0	0.0.0.0/0	multiport dports 21,873,5005,5006,80,548,...
DROP	udp	0.0.0.0/0	0.0.0.0/0	multiport dports 123,111,2049,892,5353
ACCEPT	all	192.168.0.0/16	0.0.0.0/0	
DROP	all	0.0.0.0/0	0.0.0.0/0	

Chain DOS_PROTECT (1 references)

target	prot	source	destination	
RETURN	icmp	0.0.0.0/0	0.0.0.0/0	icmptype 8 limit: avg 1/sec burst 5
DROP	icmp	0.0.0.0/0	0.0.0.0/0	icmptype 8
RETURN	tcp	0.0.0.0/0	0.0.0.0/0	tcp flags:0x17/0x04 limit: avg 1/sec burst 5
DROP	tcp	0.0.0.0/0	0.0.0.0/0	tcp flags:0x17/0x04

Introduction to Firewalls

Chain INPUT (policy ACCEPT)

target	prot	source	destination	
DOS_PROTECT	all	0.0.0.0/0	0.0.0.0/0	
ACCEPT	all	0.0.0.0/0	0.0.0.0/0	state RELATED,ESTABLISHED
DROP	tcp	0.0.0.0/0	0.0.0.0/0	tcp dpt:22
DROP	tcp	0.0.0.0/0	0.0.0.0/0	multiport dports 21,873,5005,5006,80,548,...
DROP	udp	0.0.0.0/0	0.0.0.0/0	multiport dports 123,111,2049,892,5353
ACCEPT	all	192.168.0.0/16	0.0.0.0/0	
DROP	all	0.0.0.0/0	0.0.0.0/0	

Chain DOS_PROTECT (1 references)

target	prot	source	destination	
RETURN	icmp	0.0.0.0/0	0.0.0.0/0	icmptype 8 limit: avg 1/sec burst 5
DROP	icmp	0.0.0.0/0	0.0.0.0/0	icmptype 8
RETURN	tcp	0.0.0.0/0	0.0.0.0/0	tcp flags:0x17/0x04 limit: avg 1/sec burst 5
DROP	tcp	0.0.0.0/0	0.0.0.0/0	tcp flags:0x17/0x04

Introduction to Firewalls

Chain INPUT (policy ACCEPT)

target	prot	source	destination	
DOS_PROTECT	all	0.0.0.0/0	0.0.0.0/0	
ACCEPT	all	0.0.0.0/0	0.0.0.0/0	state RELATED,ESTABLISHED
DROP	tcp	0.0.0.0/0	0.0.0.0/0	tcp dpt:22
DROP	tcp	0.0.0.0/0	0.0.0.0/0	multiport dports 21,873,5005,5006,80,548,...
DROP	udp	0.0.0.0/0	0.0.0.0/0	multiport dports 123,111,2049,892,5353
ACCEPT	all	192.168.0.0/16	0.0.0.0/0	
DROP	all	0.0.0.0/0	0.0.0.0/0	

Chain DOS_PROTECT (1 references)

target	prot	source	destination	
RETURN	icmp	0.0.0.0/0	0.0.0.0/0	icmptype 8 limit: avg 1/sec burst 5
DROP	icmp	0.0.0.0/0	0.0.0.0/0	icmptype 8
RETURN	tcp	0.0.0.0/0	0.0.0.0/0	tcp flags:0x17/0x04 limit: avg 1/sec burst 5
DROP	tcp	0.0.0.0/0	0.0.0.0/0	tcp flags:0x17/0x04

Introduction to Firewalls

Chain INPUT (policy ACCEPT)

target	prot	source	destination	
DOS_PROTECT	all	0.0.0.0/0	0.0.0.0/0	
ACCEPT	all	0.0.0.0/0	0.0.0.0/0	state RELATED,ESTABLISHED
DROP	tcp	0.0.0.0/0	0.0.0.0/0	tcp dpt:22
DROP	tcp	0.0.0.0/0	0.0.0.0/0	multiport dports 21,873,5005,5006,80,548,...
DROP	udp	0.0.0.0/0	0.0.0.0/0	multiport dports 123,111,2049,892,5353
ACCEPT	all	192.168.0.0/16	0.0.0.0/0	
DROP	all	0.0.0.0/0	0.0.0.0/0	

Chain DOS_PROTECT (1 references)

target	prot	source	destination	
RETURN	icmp	0.0.0.0/0	0.0.0.0/0	icmptype 8 limit: avg 1/sec burst 5
DROP	icmp	0.0.0.0/0	0.0.0.0/0	icmptype 8
RETURN	tcp	0.0.0.0/0	0.0.0.0/0	tcp flags:0x17/0x04 limit: avg 1/sec burst 5
DROP	tcp	0.0.0.0/0	0.0.0.0/0	tcp flags:0x17/0x04

Introduction to Firewalls

Chain INPUT (policy ACCEPT)

target	prot	source	destination	
DOS_PROTECT	all	0.0.0.0/0	0.0.0.0/0	
ACCEPT	all	0.0.0.0/0	0.0.0.0/0	state RELATED,ESTABLISHED
DROP	tcp	0.0.0.0/0	0.0.0.0/0	tcp dpt:22
DROP	tcp	0.0.0.0/0	0.0.0.0/0	multiport dports 21,873,5005,5006,80,548,...
DROP	udp	0.0.0.0/0	0.0.0.0/0	multiport dports 123,111,2049,892,5353
ACCEPT	all	192.168.0.0/16	0.0.0.0/0	
DROP	all	0.0.0.0/0	0.0.0.0/0	

Chain DOS_PROTECT (1 references)

target	prot	source	destination	
RETURN	icmp	0.0.0.0/0	0.0.0.0/0	icmptype 8 limit: avg 1/sec burst 5
DROP	icmp	0.0.0.0/0	0.0.0.0/0	icmptype 8
RETURN	tcp	0.0.0.0/0	0.0.0.0/0	tcp flags:0x17/0x04 limit: avg 1/sec burst 5
DROP	tcp	0.0.0.0/0	0.0.0.0/0	tcp flags:0x17/0x04

Introduction to Firewalls

Chain INPUT (policy ACCEPT)

target	prot	source	destination	
DOS_PROTECT	all	0.0.0.0/0	0.0.0.0/0	
ACCEPT	all	0.0.0.0/0	0.0.0.0/0	state RELATED,ESTABLISHED
DROP	tcp	0.0.0.0/0	0.0.0.0/0	tcp dpt:22
DROP	tcp	0.0.0.0/0	0.0.0.0/0	multiport dports 21,873,5005,5006,80,548,...
DROP	udp	0.0.0.0/0	0.0.0.0/0	multiport dports 123,111,2049,892,5353
ACCEPT	all	192.168.0.0/16	0.0.0.0/0	
DROP	all	0.0.0.0/0	0.0.0.0/0	

Chain DOS_PROTECT (1 references)

target	prot	source	destination	
RETURN	icmp	0.0.0.0/0	0.0.0.0/0	icmptype 8 limit: avg 1/sec burst 5
DROP	icmp	0.0.0.0/0	0.0.0.0/0	icmptype 8
RETURN	tcp	0.0.0.0/0	0.0.0.0/0	tcp flags:0x17/0x04 limit: avg 1/sec burst 5
DROP	tcp	0.0.0.0/0	0.0.0.0/0	tcp flags:0x17/0x04

Introduction to Firewalls

Chain INPUT (policy ACCEPT)

target	prot	source	destination	
DOS_PROTECT	all	0.0.0.0/0	0.0.0.0/0	
ACCEPT	all	0.0.0.0/0	0.0.0.0/0	state RELATED,ESTABLISHED
DROP	tcp	0.0.0.0/0	0.0.0.0/0	tcp dpt:22
DROP	tcp	0.0.0.0/0	0.0.0.0/0	multiport dports 21,873,5005,5006,80,548,...
DROP	udp	0.0.0.0/0	0.0.0.0/0	multiport dports 123,111,2049,892,5353
ACCEPT	all	192.168.0.0/16	0.0.0.0/0	
DROP	all	0.0.0.0/0	0.0.0.0/0	

Chain DOS_PROTECT (1 references)

target	prot	source	destination	
RETURN	icmp	0.0.0.0/0	0.0.0.0/0	icmptype 8 limit: avg 1/sec burst 5
DROP	icmp	0.0.0.0/0	0.0.0.0/0	icmptype 8
RETURN	tcp	0.0.0.0/0	0.0.0.0/0	tcp flags:0x17/0x04 limit: avg 1/sec burst 5
DROP	tcp	0.0.0.0/0	0.0.0.0/0	tcp flags:0x17/0x04

Introduction to Firewalls

Chain INPUT (policy ACCEPT)

target	prot	source	destination	
DOS_PROTECT	all	0.0.0.0/0	0.0.0.0/0	
ACCEPT	all	0.0.0.0/0	0.0.0.0/0	state RELATED,ESTABLISHED
DROP	tcp	0.0.0.0/0	0.0.0.0/0	tcp dpt:22
DROP	tcp	0.0.0.0/0	0.0.0.0/0	multiport dports 21,873,5005,5006,80,548,...
DROP	udp	0.0.0.0/0	0.0.0.0/0	multiport dports 123,111,2049,892,5353
ACCEPT	all	192.168.0.0/16	0.0.0.0/0	
DROP	all	0.0.0.0/0	0.0.0.0/0	

Chain DOS_PROTECT (1 references)

target	prot	source	destination	
RETURN	icmp	0.0.0.0/0	0.0.0.0/0	icmptype 8 limit: avg 1/sec burst 5
DROP	icmp	0.0.0.0/0	0.0.0.0/0	icmptype 8
RETURN	tcp	0.0.0.0/0	0.0.0.0/0	tcp flags:0x17/0x04 limit: avg 1/sec burst 5
DROP	tcp	0.0.0.0/0	0.0.0.0/0	tcp flags:0x17/0x04

Introduction to Firewalls

Chain INPUT (policy ACCEPT)

target	prot	source	destination	
DOS_PROTECT	all	0.0.0.0/0	0.0.0.0/0	
ACCEPT	all	0.0.0.0/0	0.0.0.0/0	state RELATED,ESTABLISHED
DROP	tcp	0.0.0.0/0	0.0.0.0/0	tcp dpt:22
DROP	tcp	0.0.0.0/0	0.0.0.0/0	multiport dports 21,873,5005,5006,80,548,...
DROP	udp	0.0.0.0/0	0.0.0.0/0	multiport dports 123,111,2049,892,5353
ACCEPT	all	192.168.0.0/16	0.0.0.0/0	
DROP	all	0.0.0.0/0	0.0.0.0/0	

Chain DOS_PROTECT (1 references)

target	prot	source	destination	
RETURN	icmp	0.0.0.0/0	0.0.0.0/0	icmptype 8 limit: avg 1/sec burst 5
DROP	icmp	0.0.0.0/0	0.0.0.0/0	icmptype 8
RETURN	tcp	0.0.0.0/0	0.0.0.0/0	tcp flags:0x17/0x04 limit: avg 1/sec burst 5
DROP	tcp	0.0.0.0/0	0.0.0.0/0	tcp flags:0x17/0x04

Introduction to Firewalls

Chain INPUT (policy ACCEPT)

target	prot	source	destination	
DOS_PROTECT	all	0.0.0.0/0	0.0.0.0/0	
ACCEPT	all	0.0.0.0/0	0.0.0.0/0	state RELATED,ESTABLISHED
DROP	tcp	0.0.0.0/0	0.0.0.0/0	tcp dpt:22
DROP	tcp	0.0.0.0/0	0.0.0.0/0	multiport dports 21,873,5005,5006,80,548,...
DROP	udp	0.0.0.0/0	0.0.0.0/0	multiport dports 123,111,2049,892,5353
ACCEPT	all	192.168.0.0/16	0.0.0.0/0	
DROP	all	0.0.0.0/0	0.0.0.0/0	

Chain DOS_PROTECT (1 references)

target	prot	source	destination	
RETURN	icmp	0.0.0.0/0	0.0.0.0/0	icmptype 8 limit: avg 1/sec burst 5
DROP	icmp	0.0.0.0/0	0.0.0.0/0	icmptype 8
RETURN	tcp	0.0.0.0/0	0.0.0.0/0	tcp flags:0x17/0x04 limit: avg 1/sec burst 5
DROP	tcp	0.0.0.0/0	0.0.0.0/0	tcp flags:0x17/0x04

Problem

“there are no good high-complexity rule sets”

A. Wool, A Quantitative Study of Firewall Configuration Errors, Computer, IEEE, vol. 37, no. 6, pp. 62–67, Jun. 2004.

Problem

“there are no good high-complexity rule sets”

“firewalls are (still) poorly configured”

A. Wool, A Quantitative Study of Firewall Configuration Errors, Computer, IEEE, vol. 37, no. 6, pp. 62–67, Jun. 2004.

A. Wool, Trends in Firewall Configuration Errors: Measuring the Holes in Swiss Cheese, Internet Computing, IEEE, vol. 14, no. 4, pp. 58–65, Jul. 2010.

Problem

“there are no good high-complexity rule sets”

“firewalls are (still) poorly configured”

“tools do not understand real-world firewall rules”

A. Wool, A Quantitative Study of Firewall Configuration Errors, Computer, IEEE, vol. 37, no. 6, pp. 62–67, Jun. 2004.

A. Wool, Trends in Firewall Configuration Errors: Measuring the Holes in Swiss Cheese, Internet Computing, IEEE, vol. 14, no. 4, pp. 58–65, Jul. 2010.

C. Diekmann, L. Hupel, and G. Carle, Semantics-Preserving Simplification of Real-World Firewall Rule Sets, in Formal Methods (FM). Springer, pp. 195–212. Jun. 2015

A Tool for Ruleset Verification

Specification

- ▶ Documentation

Implementation

- ▶ Code, tool
- ▶ Performance

A Tool for Ruleset Verification

Specification

- ▶ Documentation
- ▶ *What is a correct ruleset?*

Implementation

- ▶ Code, tool
- ▶ Performance

A Tool for Ruleset Verification

Specification

- ▶ Documentation
- ▶ *What is a correct ruleset?*
- ▶ Goal: **Spoofing protection**

Implementation

- ▶ Code, tool
- ▶ Performance

A Tool for Ruleset Verification

Specification

- ▶ Documentation
- ▶ *What is a correct ruleset?*
- ▶ Goal: Spoofing protection
- ▶ Needs: **Model of iptables**

Implementation

- ▶ Code, tool
- ▶ Performance

A Tool for Ruleset Verification



Specification

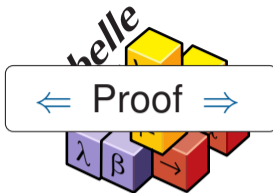
- ▶ Documentation
- ▶ *What is a correct ruleset?*
- ▶ Goal: Spoofing protection
- ▶ Needs: Model of iptables

Implementation

- ▶ Code, tool
- ▶ Performance

<http://isabelle.in.tum.de/>

A Tool for Ruleset Verification



Specification

- ▶ Documentation
- ▶ *What is a correct ruleset?*
- ▶ Goal: Spoofing protection
- ▶ Needs: Model of iptables

Implementation

- ▶ Code, tool
- ▶ Performance

<http://isabelle.in.tum.de/>

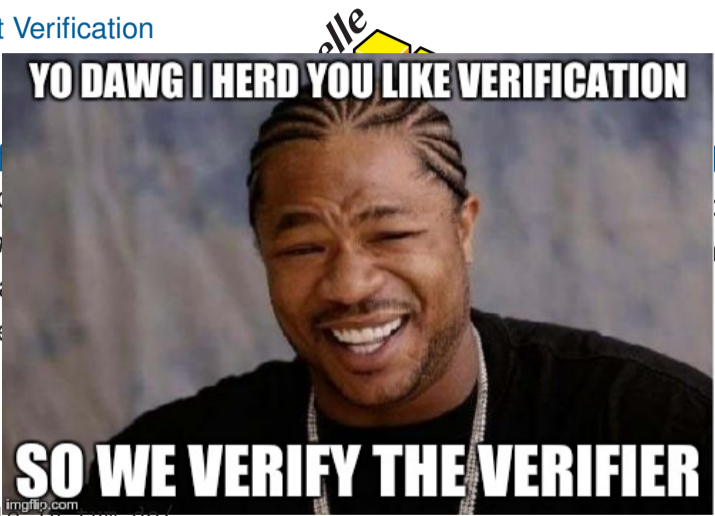
A Tool for Ruleset Verification

Specification

- ▶ Doc
- ▶ Wh
- ▶ Go
- ▶ Ne

Verification

- ▶ ol
- ▶ nce



<http://isabelle.in.tum.de/>



Match Expressions: Syntax and Semantics

Syntax

- ▶ How to represent match expressions?


$$\text{datatype 'a mexpr} = \text{Match 'a} \mid \text{MatchAny}$$
$$\mid \text{MatchNot 'a mexpr} \mid \text{MatchAnd 'a mexpr 'a mexpr}$$

Match Expressions: Syntax and Semantics

Syntax

- ▶ How to represent match expressions?

Polymorphic: arbitrary type 'a



`datatype 'a mexpr = Match 'a | MatchAny`
`| MatchNot 'a mexpr | MatchAnd 'a mexpr 'a mexpr`

Match Expressions: Syntax and Semantics

Syntax

- ▶ How to represent match expressions?

Polymorphic: arbitrary type 'a




datatype 'a mexpr = **Match 'a** | MatchAny
| MatchNot 'a mexpr | MatchAnd 'a mexpr 'a mexpr

Match Expressions: Syntax and Semantics

Syntax

- ▶ How to represent match expressions?

Polymorphic: arbitrary type 'a



datatype 'a mexpr = Match 'a | MatchAny
 | MatchNot 'a mexpr | MatchAnd 'a mexpr 'a mexpr

Match Expressions: Syntax and Semantics

Syntax

- ▶ How to represent match expressions?

Polymorphic: arbitrary type 'a

datatype 'a mexpr = Match 'a | MatchAny
| MatchNot 'a mexpr | MatchAnd 'a mexpr 'a mexpr

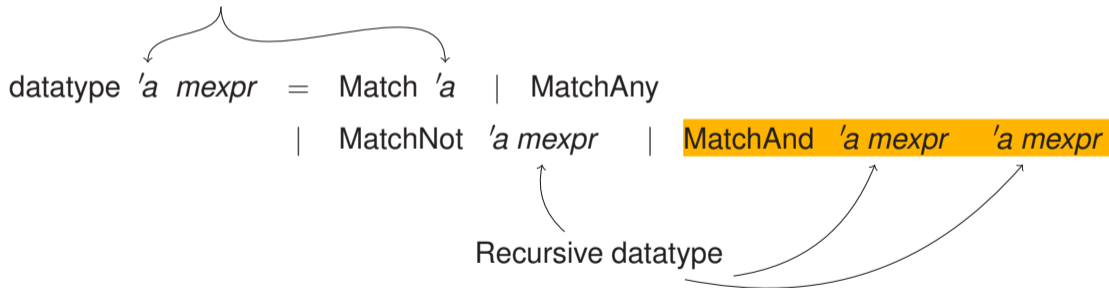
Recursive datatype

Match Expressions: Syntax and Semantics

Syntax

- ▶ How to represent match expressions?

Polymorphic: arbitrary type 'a





Match Expressions: Syntax and Semantics

Syntax

- ▶ How to represent match expressions?

```
datatype 'a mexpr = Match 'a | MatchAny  
                  | MatchNot 'a mexpr | MatchAnd 'a mexpr 'a mexpr
```

Example:

```
MatchAnd (Match ( DstIP 8.8.8.8 )) (Match ( Protocol TCP ))
```

Match Expressions: Syntax and Semantics

Syntax

- ▶ How to represent match expressions?

datatype 'a mexpr = Match 'a | MatchAny
 | MatchNot 'a mexpr | MatchAnd 'a mexpr 'a mexpr

Example:

MatchAnd (Match (DstIP 8.8.8.8)) (Match (Protocol TCP))





Match Expressions: Syntax and Semantics

Semantics

- ▶ What do match expressions mean?

$$\text{matches} :: ('a \Rightarrow 'p \Rightarrow \mathbb{B}) \Rightarrow 'a \text{ mexpr} \Rightarrow 'p \Rightarrow \mathbb{B}$$
$$\text{matches } \gamma \text{ (Match } a) p \quad \longleftrightarrow \quad \gamma \ a \ p$$
$$\text{matches } _ \text{ MatchAny } _ \quad \longleftrightarrow \quad \text{True}$$
$$\text{matches } \gamma \text{ (MatchNot } m) p \quad \longleftrightarrow \quad \neg \text{ matches } \gamma \ m \ p$$
$$\text{matches } \gamma \text{ (MatchAnd } m_1 \ m_2) p \quad \longleftrightarrow \quad \text{matches } \gamma \ m_1 \ p \ \wedge \ \text{matches } \gamma \ m_2 \ p$$



Match Expressions: Syntax and Semantics

Semantics

- ▶ What do match expressions mean?

$matches :: ('a \Rightarrow 'p \Rightarrow \mathbb{B}) \Rightarrow 'a\ mexpr \Rightarrow 'p \Rightarrow \mathbb{B}$

$matches\ \gamma\ (Match\ a)\ p \quad \longleftrightarrow \quad \gamma\ a\ p$

$matches\ _ \ MatchAny\ _ \quad \longleftrightarrow \quad True$

$matches\ \gamma\ (MatchNot\ m)\ p \quad \longleftrightarrow \quad \neg\ matches\ \gamma\ m\ p$

$matches\ \gamma\ (MatchAnd\ m_1\ m_2)\ p \quad \longleftrightarrow \quad matches\ \gamma\ m_1\ p \wedge matches\ \gamma\ m_2\ p$



Match Expressions: Syntax and Semantics

Semantics

- ▶ What do match expressions mean?

$matches :: ('a \Rightarrow 'p \Rightarrow \mathbb{B}) \Rightarrow 'a\ mexpr \Rightarrow 'p \Rightarrow \mathbb{B}$

$matches\ \gamma\ (Match\ a)\ p \iff \gamma\ a\ p$

$matches\ _ \ (MatchAny\ _)\ _ \iff True$

$matches\ \gamma\ (MatchNot\ m)\ p \iff \neg\ matches\ \gamma\ m\ p$

$matches\ \gamma\ (MatchAnd\ m_1\ m_2)\ p \iff matches\ \gamma\ m_1\ p \wedge matches\ \gamma\ m_2\ p$

Match Expressions: Syntax and Semantics

Semantics

- ▶ What do match expressions mean?

$$\text{matches} :: ('a \Rightarrow 'p \Rightarrow \mathbb{B}) \Rightarrow \text{'a mexpr} \Rightarrow 'p \Rightarrow \mathbb{B}$$
$$\text{matches } \gamma \text{ (Match } a) p \quad \longleftrightarrow \quad \gamma \ a \ p$$
$$\text{matches } _ \text{ MatchAny } _ \quad \longleftrightarrow \quad \text{True}$$
$$\text{matches } \gamma \text{ (MatchNot } m) p \quad \longleftrightarrow \quad \neg \text{ matches } \gamma \ m \ p$$
$$\text{matches } \gamma \text{ (MatchAnd } m_1 \ m_2) p \quad \longleftrightarrow \quad \text{matches } \gamma \ m_1 \ p \wedge \text{ matches } \gamma \ m_2 \ p$$

Match Expressions: Syntax and Semantics

Semantics

- ▶ What do match expressions mean?

$$\text{matches} :: ('a \Rightarrow 'p \Rightarrow \mathbb{B}) \Rightarrow 'a \text{ mexpr} \Rightarrow 'p \Rightarrow \mathbb{B}$$
$$\text{matches } \gamma \text{ (Match } a) \ p \quad \longleftrightarrow \quad \gamma \ a \ p$$
$$\text{matches } _ \text{ MatchAny } _ \quad \longleftrightarrow \quad \text{True}$$
$$\text{matches } \gamma \text{ (MatchNot } m) \ p \quad \longleftrightarrow \quad \neg \text{ matches } \gamma \ m \ p$$
$$\text{matches } \gamma \text{ (MatchAnd } m_1 \ m_2) \ p \quad \longleftrightarrow \quad \text{matches } \gamma \ m_1 \ p \wedge \text{ matches } \gamma \ m_2 \ p$$

Match Expressions: Syntax and Semantics

Semantics

- ▶ What do match expressions mean?

$$matches :: ('a \Rightarrow 'p \Rightarrow \mathbb{B}) \Rightarrow 'a\ mexpr \Rightarrow 'p \Rightarrow \mathbb{B}$$

$$matches\ \gamma\ (Match\ a)\ p \quad \longleftrightarrow \quad \gamma\ a\ p$$

$$matches\ _ \ MatchAny\ _ \quad \longleftrightarrow \quad True$$

$$matches\ \gamma\ (MatchNot\ m)\ p \quad \longleftrightarrow \quad \neg\ matches\ \gamma\ m\ p$$

$$matches\ \gamma\ (MatchAnd\ m_1\ m_2)\ p \quad \longleftrightarrow \quad matches\ \gamma\ m_1\ p \wedge matches\ \gamma\ m_2\ p$$



Match Expressions: Syntax and Semantics

Semantics

- ▶ What do match expressions mean?

$$\text{matches} :: ('a \Rightarrow 'p \Rightarrow \mathbb{B}) \Rightarrow 'a \text{ mexpr} \Rightarrow 'p \Rightarrow \mathbb{B}$$
$$\text{matches } \gamma \text{ (Match } a) p \quad \longleftrightarrow \quad \gamma \ a \ p$$
$$\text{matches } _ \text{ MatchAny } _ \quad \longleftrightarrow \quad \text{True}$$
$$\text{matches } \gamma \text{ (MatchNot } m) p \quad \longleftrightarrow \quad \neg \text{ matches } \gamma \ m \ p$$
$$\text{matches } \gamma \text{ (MatchAnd } m_1 \ m_2) p \quad \longleftrightarrow \quad \text{matches } \gamma \ m_1 \ p \ \wedge \ \text{matches } \gamma \ m_2 \ p$$



Match Expressions: Syntax and Semantics

Semantics

- ▶ What do match expressions mean?

$$matches :: ('a \Rightarrow 'p \Rightarrow \mathbb{B}) \Rightarrow 'a\ mexpr \Rightarrow 'p \Rightarrow \mathbb{B}$$

$$matches\ \gamma\ (Match\ a)\ p \iff \gamma\ a\ p$$

$$matches\ _ \ MatchAny\ _ \iff True$$

$$matches\ \gamma\ (MatchNot\ m)\ p \iff \neg\ matches\ \gamma\ m\ p$$

$$matches\ \gamma\ (MatchAnd\ m_1\ m_2)\ p \iff matches\ \gamma\ m_1\ p \wedge matches\ \gamma\ m_2\ p$$

Match Expressions: Syntax and Semantics

Semantics

- ▶ What do match expressions mean?

$$\text{matches} :: ('a \Rightarrow 'p \Rightarrow \mathbb{B}) \Rightarrow 'a \text{ mexpr} \Rightarrow 'p \Rightarrow \mathbb{B}$$
$$\text{matches } \gamma \text{ (Match } a) p \quad \longleftrightarrow \quad \gamma \ a \ p$$
$$\text{matches } _ \text{ MatchAny } _ \quad \longleftrightarrow \quad \text{True}$$
$$\text{matches } \gamma \text{ (MatchNot } m) p \quad \longleftrightarrow \quad \neg \text{ matches } \gamma \ m \ p$$
$$\text{matches } \gamma \text{ (MatchAnd } m_1 \ m_2) p \quad \longleftrightarrow \quad \text{matches } \gamma \ m_1 \ p \ \wedge \ \text{matches } \gamma \ m_2 \ p$$



Match Expressions: Syntax and Semantics

Semantics

- ▶ What do match expressions mean?

$$\text{matches} :: ('a \Rightarrow 'p \Rightarrow \mathbb{B}) \Rightarrow 'a \text{ mexpr} \Rightarrow 'p \Rightarrow \mathbb{B}$$
$$\text{matches } \gamma \text{ (Match } a) p \quad \longleftrightarrow \quad \gamma \ a \ p$$
$$\text{matches } _ \text{ MatchAny } _ \quad \longleftrightarrow \quad \text{True}$$
$$\text{matches } \gamma \text{ (MatchNot } m) p \quad \longleftrightarrow \quad \neg \text{ matches } \gamma \ m \ p$$
$$\text{matches } \gamma \text{ (MatchAnd } m_1 \ m_2) p \quad \longleftrightarrow \quad \text{matches } \gamma \ m_1 \ p \ \wedge \ \text{matches } \gamma \ m_2 \ p$$



Match Expressions: Syntax and Semantics

Semantics

- ▶ What do match expressions mean?

$$\text{matches} :: ('a \Rightarrow 'p \Rightarrow \mathbb{B}) \Rightarrow 'a \text{ mexpr} \Rightarrow 'p \Rightarrow \mathbb{B}$$
$$\text{matches } \gamma \text{ (Match } a) p \quad \longleftrightarrow \quad \gamma \ a \ p$$
$$\text{matches } _ \text{ MatchAny } _ \quad \longleftrightarrow \quad \text{True}$$
$$\text{matches } \gamma \text{ (MatchNot } m) p \quad \longleftrightarrow \quad \neg \text{ matches } \gamma \ m \ p$$
$$\text{matches } \gamma \text{ (MatchAnd } m_1 \ m_2) p \quad \longleftrightarrow \quad \text{matches } \gamma \ m_1 \ p \ \wedge \ \text{matches } \gamma \ m_2 \ p$$

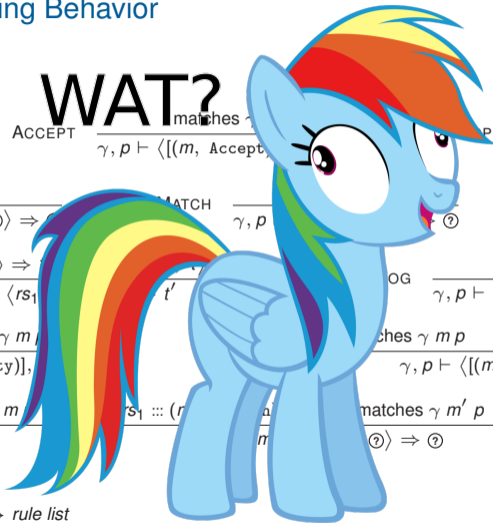
Iptables Semantics: Filtering Behavior

$$\begin{array}{c}
 \text{SKIP} \quad \frac{}{\gamma, p \vdash \langle [], t \rangle \Rightarrow t} \qquad \text{ACCEPT} \quad \frac{\text{matches } \gamma \ m \ p}{\gamma, p \vdash \langle [(m, \text{Accept})], \textcircled{?} \rangle \Rightarrow \checkmark} \qquad \text{DROP} \quad \frac{\text{matches } \gamma \ m \ p}{\gamma, p \vdash \langle [(m, \text{Drop})], \textcircled{?} \rangle \Rightarrow \otimes} \\
 \\
 \text{REJECT} \quad \frac{\text{matches } \gamma \ m \ p}{\gamma, p \vdash \langle [(m, \text{Reject})], \textcircled{?} \rangle \Rightarrow \otimes} \qquad \text{NOMATCH} \quad \frac{\neg \text{matches } \gamma \ m \ p}{\gamma, p \vdash \langle [(m, a)], \textcircled{?} \rangle \Rightarrow \textcircled{?}} \qquad \text{DECISION} \quad \frac{t \neq \textcircled{?}}{\gamma, p \vdash \langle rs, t \rangle \Rightarrow t} \\
 \\
 \text{SEQ} \quad \frac{\gamma, p \vdash \langle rs_1, \textcircled{?} \rangle \Rightarrow t \quad \gamma, p \vdash \langle rs_2, t \rangle \Rightarrow t'}{\gamma, p \vdash \langle rs_1 \dots rs_2, \textcircled{?} \rangle \Rightarrow t'} \qquad \text{LOG} \quad \frac{\text{matches } \gamma \ m \ p}{\gamma, p \vdash \langle [(m, \text{Log})], \textcircled{?} \rangle \Rightarrow \textcircled{?}} \\
 \\
 \text{EMPTY} \quad \frac{\text{matches } \gamma \ m \ p}{\gamma, p \vdash \langle [(m, \text{Empty})], \textcircled{?} \rangle \Rightarrow \textcircled{?}} \qquad \text{CALLRESULT} \quad \frac{\text{matches } \gamma \ m \ p \quad \gamma, p \vdash \langle \Gamma \ c, \textcircled{?} \rangle \Rightarrow t}{\gamma, p \vdash \langle [(m, \text{Call } c)], \textcircled{?} \rangle \Rightarrow t} \\
 \\
 \text{CALLRETURN} \quad \frac{\text{matches } \gamma \ m \ p \quad \Gamma \ c = rs_1 \dots (m', \text{Return}) \dots rs_2 \quad \text{matches } \gamma \ m' \ p \quad \gamma, p \vdash \langle rs_1, \textcircled{?} \rangle \Rightarrow \textcircled{?}}{\gamma, p \vdash \langle [(m, \text{Call } c)], \textcircled{?} \rangle \Rightarrow \textcircled{?}}
 \end{array}$$

Background ruleset Γ : *chain name* \rightarrow *rule list*

Iptables Semantics: Filtering Behavior

WAT?



SKIP	$\frac{}{\gamma, p \vdash \langle [], t \rangle \Rightarrow t}$	ACCEPT	$\frac{\text{matches } \gamma \ m \ p}{\gamma, p \vdash \langle [(m, \text{Accept})], t \rangle \Rightarrow t}$	DROP	$\frac{\text{matches } \gamma \ m \ p}{\gamma, p \vdash \langle [(m, \text{Drop})], \textcircled{?} \rangle \Rightarrow \textcircled{X}}$
REJECT	$\frac{\text{matches } \gamma \ m \ p}{\gamma, p \vdash \langle [(m, \text{Reject})], \textcircled{?} \rangle \Rightarrow \textcircled{X}}$	MATCH	$\frac{}{\gamma, p \vdash \langle [m], t \rangle \Rightarrow t}$	DECISION	$\frac{t \neq \textcircled{?}}{\gamma, p \vdash \langle rs, t \rangle \Rightarrow t}$
SEQ	$\frac{\gamma, p \vdash \langle rs_1, \textcircled{?} \rangle \Rightarrow t'}{\gamma, p \vdash \langle rs_1, t' \rangle \Rightarrow t'}$	LOG	$\frac{\text{matches } \gamma \ m \ p}{\gamma, p \vdash \langle [(m, \text{Log})], \textcircled{?} \rangle \Rightarrow \textcircled{?}}$		
EMPTY	$\frac{\text{matches } \gamma \ m \ p}{\gamma, p \vdash \langle [(m, \text{Empty})], \textcircled{?} \rangle \Rightarrow \textcircled{?}}$	CALL	$\frac{\text{matches } \gamma \ m \ p \quad \gamma, p \vdash \langle \Gamma \ c, \textcircled{?} \rangle \Rightarrow t}{\gamma, p \vdash \langle [(m, \text{Call } c)], \textcircled{?} \rangle \Rightarrow t}$		
CALLRETURN	$\frac{\text{matches } \gamma \ m \ p \quad rs_1 \dots (m, \text{Call } c) \dots (m, \text{Call } c) \dots}{\gamma, p \vdash \langle rs_1, \textcircled{?} \rangle \Rightarrow \textcircled{?}}$		$\frac{\text{matches } \gamma \ m' \ p \quad \gamma, p \vdash \langle rs_1, \textcircled{?} \rangle \Rightarrow \textcircled{?}}{\gamma, p \vdash \langle [(m, \text{Call } c)], \textcircled{?} \rangle \Rightarrow \textcircled{?}}$		

Background ruleset Γ : *chain name* \rightarrow *rule list*


Semantics Explained

$$\gamma, p \vdash \langle rs, s \rangle \Rightarrow t$$

Semantics Explained

$$\gamma, \mathbf{p} \vdash \langle \mathbf{rs}, \mathbf{s} \rangle \Rightarrow t$$

Packet

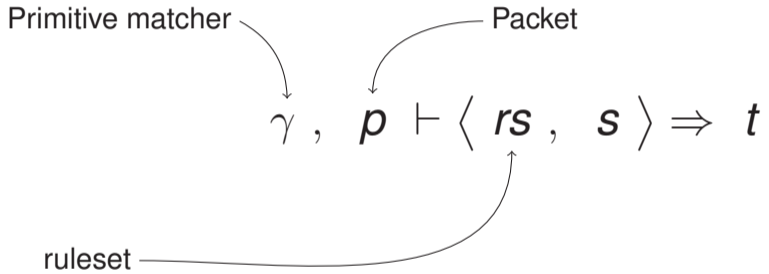


Semantics Explained

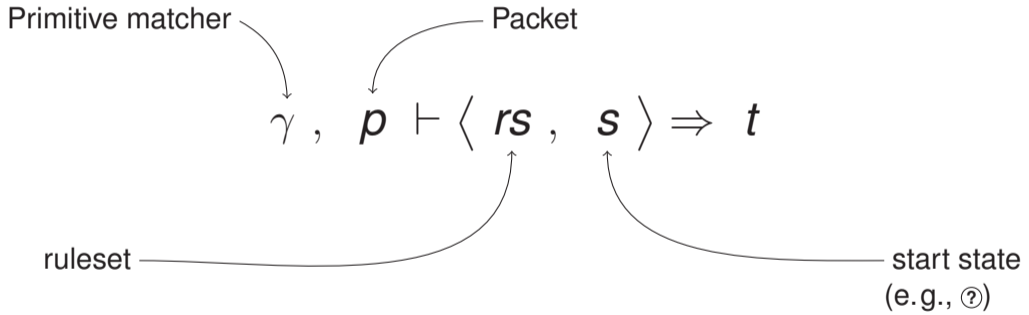
Primitive matcher γ Packet p

$$\gamma, p \vdash \langle rs, s \rangle \Rightarrow t$$

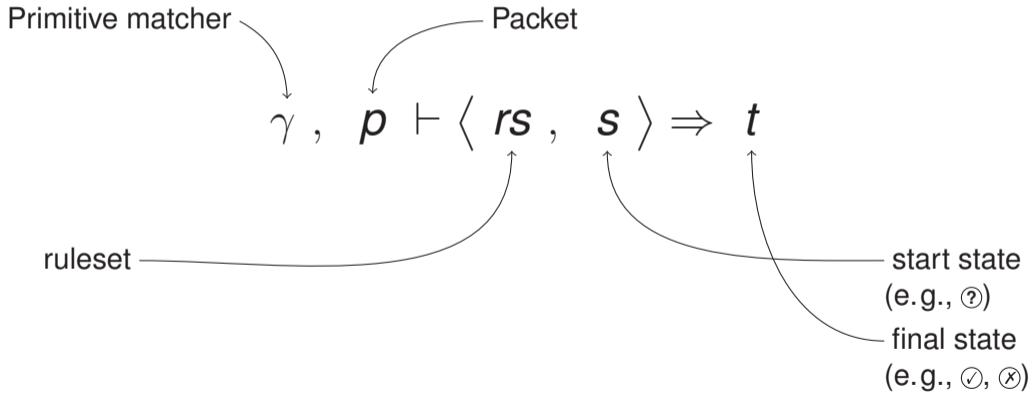
Semantics Explained



Semantics Explained



Semantics Explained



Semantics Explained: SKIP

$$\frac{}{\gamma, \rho \vdash \langle [], t \rangle \Rightarrow t}$$

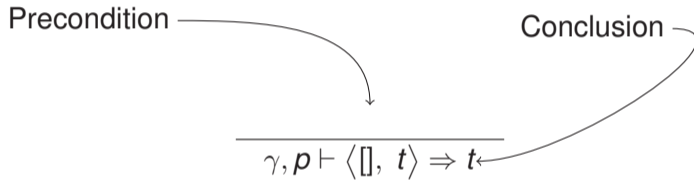
Semantics Explained: SKIP

Precondition

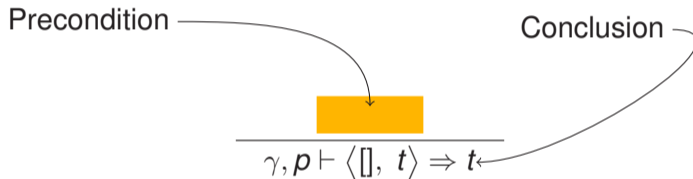


$$\frac{}{\gamma, \rho \vdash \langle [], t \rangle \Rightarrow t}$$

Semantics Explained: SKIP

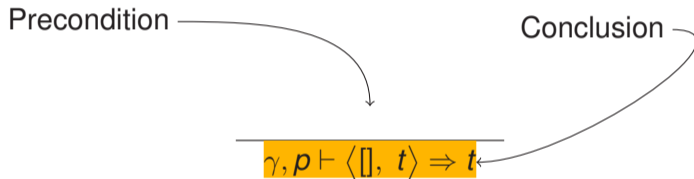


Semantics Explained: SKIP



- ▶ **no precondition**
 - ▶ Holds unconditionally

Semantics Explained: SKIP



- ▶ no precondition
 - ▶ Holds unconditionally
 - ▶ IF TRUE then $\gamma, p \vdash \langle [], t \rangle \Rightarrow t$

Semantics Explained: SKIP

$$\frac{}{\gamma, \rho \vdash \langle [], t \rangle \Rightarrow t}$$

Semantics Explained: SKIP

$$\frac{}{\gamma, \rho \vdash \langle \square, t \rangle \Rightarrow t}$$

- ▶ Empty Ruleset

Semantics Explained: SKIP

$$\frac{}{\gamma, p \vdash \langle [], t \rangle \Rightarrow t}$$

- ▶ Empty Ruleset
- ▶ Start state equals final state

Semantics Explained: SKIP

$$\frac{}{\gamma, \rho \vdash \langle [], t \rangle \Rightarrow t}$$

- ▶ Empty Ruleset
- ▶ Start state equals final state
- ▶ For the empty ruleset, the firewall does nothing

Semantics Explained: ACCEPT

$$\frac{\text{matches } \gamma \ m \ p}{\gamma, p \vdash \langle [(m, \text{Accept})], \text{?} \rangle \Rightarrow \checkmark}$$

Semantics Explained: ACCEPT

$$\frac{\text{matches } \gamma \ m \ p}{\gamma, p \vdash \langle [(m, \text{Accept})], (?) \rangle \Rightarrow \checkmark}$$

- ▶ Ruleset: single rule

Semantics Explained: ACCEPT

$$\frac{\text{matches } \gamma \ m \ p}{\gamma, p \vdash \langle [(m, \text{Accept})], (?) \rangle \Rightarrow \checkmark}$$

- ▶ Ruleset: single rule
- ▶ matches

Semantics Explained: ACCEPT

$$\frac{\text{matches } \gamma \ m \ p}{\gamma, p \vdash \langle [(m, \text{Accept})], (?) \rangle \Rightarrow \checkmark}$$

- ▶ Ruleset: single rule
- ▶ matches
- ▶ The action of the rule is Accept rule

Semantics Explained: ACCEPT

$$\frac{\text{matches } \gamma \ m \ p}{\gamma, \rho \vdash \langle [(m, \text{Accept})], \text{?} \rangle \Rightarrow \checkmark}$$

- ▶ Ruleset: single rule
- ▶ matches
- ▶ The action of the rule is `Accept` rule
- ▶ The firewall does not have a decision yet

Semantics Explained: ACCEPT

$$\frac{\text{matches } \gamma \ m \ p}{\gamma, p \vdash \langle [(m, \text{Accept})], \textcircled{?} \rangle \Rightarrow \checkmark}$$

- ▶ Ruleset: single rule
- ▶ matches
- ▶ The action of the rule is `Accept` rule
- ▶ The firewall does not have a decision yet
- ▶ It will accept the packet

Semantics Explained: ACCEPT

$$\frac{\text{matches } \gamma \ m \ p}{\gamma, p \vdash \langle [(m, \text{Accept})], (?) \rangle \Rightarrow \checkmark}$$

- ▶ Ruleset: single rule
- ▶ matches
- ▶ The action of the rule is `Accept` rule
- ▶ The firewall does not have a decision yet
- ▶ It will accept the packet
- ▶ A matching `Accept` rule accepts packets



Semantics Explained: CALLRETURN

$$\frac{\text{matches } \gamma \ m \ p \quad \Gamma \ c = rs_1 :: (m', \text{Return}) :: rs_2 \quad \text{matches } \gamma \ m' \ p \quad \gamma, p \vdash \langle rs_1, \textcircled{?} \rangle \Rightarrow \textcircled{?}}{\gamma, p \vdash \langle [(m, \text{Call } c)], \textcircled{?} \rangle \Rightarrow \textcircled{?}}$$



Semantics Explained: CALLRETURN

$$\frac{\text{matches } \gamma m p \quad \Gamma c = rs_1 ::: (m', \text{Return}) :: rs_2 \quad \text{matches } \gamma m' p \quad \gamma, p \vdash \langle rs_1, ? \rangle \Rightarrow ?}{\gamma, p \vdash \langle [(m, \text{Call } c)], ? \rangle \Rightarrow ?}$$

► matches

Semantics Explained: CALLRETURN

$$\frac{\text{matches } \gamma \ m \ p \quad \Gamma \ c = rs_1 ::: (m', \text{Return}) :: rs_2 \quad \text{matches } \gamma \ m' \ p \quad \gamma, p \vdash \langle rs_1, \textcircled{?} \rangle \Rightarrow \textcircled{?}}{\gamma, p \vdash \langle [(m, \text{Call } c)], \textcircled{?} \rangle \Rightarrow \textcircled{?}}$$

- ▶ matches
- ▶ The called chain c in the background ruleset Γ is defined as $rs_1 ::: (m', \text{Return}) :: rs_2$

Semantics Explained: CALLRETURN

$$\frac{\text{matches } \gamma \ m \ p \quad \Gamma \ c = rs_1 ::: (m', \text{Return}) :: rs_2 \quad \text{matches } \gamma \ m' \ p \quad \gamma, p \vdash \langle rs_1, \text{?} \rangle \Rightarrow \text{?}}{\gamma, p \vdash \langle [(m, \text{Call } c)], \text{?} \rangle \Rightarrow \text{?}}$$

- ▶ matches
- ▶ The called chain c in the background ruleset Γ is defined as $rs_1 ::: (m', \text{Return}) :: rs_2$
- ▶ First part rs_1 is processed without result

Semantics Explained: CALLRETURN

$$\frac{\text{matches } \gamma \ m \ p \quad \Gamma \ c = rs_1 ::: (m', \text{Return}) :: rs_2 \quad \text{matches } \gamma \ m' \ p \quad \gamma, p \vdash \langle rs_1, ? \rangle \Rightarrow ?}{\gamma, p \vdash \langle [(m, \text{Call } c)], ? \rangle \Rightarrow ?}$$

- ▶ matches
- ▶ The called chain c in the background ruleset Γ is defined as $rs_1 ::: (m', \text{Return}) :: rs_2$
- ▶ First part rs_1 is processed without result
- ▶ Then there is a matching Return

Semantics Explained: CALLRETURN

$$\frac{\text{matches } \gamma \ m \ p \quad \Gamma \ c = rs_1 ::: (m', \text{Return}) :: rs_2 \quad \text{matches } \gamma \ m' \ p \quad \gamma, p \vdash \langle rs_1, ? \rangle \Rightarrow ?}{\gamma, p \vdash \langle [(m, \text{Call } c)], ? \rangle \Rightarrow ?}$$

- ▶ matches
- ▶ The called chain c in the background ruleset Γ is defined as $rs_1 ::: (m', \text{Return}) :: rs_2$
- ▶ First part rs_1 is processed without result
- ▶ Then there is a matching Return
- ▶ Calling to user-defined chain and return without result

Semantics Explained: CALLRETURN

WHY?

$$\frac{\text{matches } \gamma \ m \ p \quad \Gamma \ c = rs_1 :: (m', \text{Return}) :: rs_2 \quad \text{matches } \gamma \ m' \ p \quad \gamma, p \vdash \langle rs_1, ? \rangle \Rightarrow ?}{\text{all } c], ? \rangle \Rightarrow ?}$$

- ▶ matches
- ▶ The called chain c in the base ruleset Γ is defined as $rs_1 :: (m', \text{Return}) :: rs_2$
- ▶ First part rs_1 is processed
- ▶ Then there is a matching rule
- ▶ Calling to user-defined chain m' with result



Semantics-Preserving Simplification

$$\gamma, p \vdash \langle rs, s \rangle \Rightarrow t \quad \text{iff} \quad \gamma, p \vdash \langle f rs, s \rangle \Rightarrow t$$

Semantics-Preserving Simplification

$$\gamma, p \vdash \langle rs, s \rangle \Rightarrow t \quad \text{iff} \quad \gamma, p \vdash \langle \mathbf{f} rs, s \rangle \Rightarrow t$$

- ▶ $\mathbf{f} :: 'a \text{ ruleset} \Rightarrow 'a \text{ ruleset}$

Semantics-Preserving Simplification

$$\gamma, p \vdash \langle rs, s \rangle \Rightarrow t \quad \text{iff} \quad \gamma, p \vdash \langle \mathbf{f} rs, s \rangle \Rightarrow t$$

- ▶ $f :: 'a \text{ ruleset} \Rightarrow 'a \text{ ruleset}$
- ▶ f does not change filtering behavior of firewall

Semantics-Preserving Simplification

$$\gamma, p \vdash \langle rs, s \rangle \Rightarrow t \quad \text{iff} \quad \gamma, p \vdash \langle \mathbf{f} rs, s \rangle \Rightarrow t$$

- ▶ $f :: 'a \text{ ruleset} \Rightarrow 'a \text{ ruleset}$
- ▶ f does not change filtering behavior of firewall
- ▶ Removing Log rules

Semantics-Preserving Simplification

$$\gamma, p \vdash \langle rs, s \rangle \Rightarrow t \quad \text{iff} \quad \gamma, p \vdash \langle \mathbf{f} rs, s \rangle \Rightarrow t$$

- ▶ $f :: 'a \text{ ruleset} \Rightarrow 'a \text{ ruleset}$
- ▶ f does not change filtering behavior of firewall
- ▶ Removing Log rules
- ▶ Unfolding of user-defined chains

Semantics-Preserving Simplification

$$\gamma, p \vdash \langle rs, s \rangle \Rightarrow t \quad \text{iff} \quad \gamma, p \vdash \langle \mathbf{f} rs, s \rangle \Rightarrow t$$

- ▶ $f :: 'a \text{ ruleset} \Rightarrow 'a \text{ ruleset}$
- ▶ f does not change filtering behavior of firewall
- ▶ Removing Log rules
- ▶ Unfolding of user-defined chains
- ▶ Normalizing match expressions, ...



Embedding in Ternary Logic

$$\mathbb{B} = \{True, False\}$$

$$Ternary = \{True, False, Unknown\}$$

$$\{p \mid \text{approx_firewall } \gamma \text{ stricter } rs = \checkmark\}$$

$$\subseteq$$

$$\{p \mid \gamma, p \vdash \langle rs, ? \rangle \Rightarrow \checkmark\}$$

$$\subseteq$$

$$\{p \mid \text{approx_firewall } \gamma \text{ permissive } rs = \checkmark\}$$

Embedding in Ternary Logic

$$\mathbb{B} = \{True, False\} \quad Ternary = \{True, False, Unknown\}$$

$$\{p \mid \text{approx_firewall } \gamma \text{ stricter } rs = \checkmark\}$$

$$\subseteq$$

$$\{p \mid \gamma, p \vdash \langle rs, ? \rangle \Rightarrow \checkmark\}$$

$$\subseteq$$

$$\{p \mid \text{approx_firewall } \gamma \text{ permissive } rs = \checkmark\}$$

- ▶ Set of packets accepted by the firewall

Embedding in Ternary Logic

$$\mathbb{B} = \{True, False\} \quad Ternary = \{True, False, Unknown\}$$

$$\{p \mid \text{approx_firewall } \gamma \text{ stricter } rs = \checkmark\}$$

$$\subseteq$$

$$\{p \mid \gamma, p \vdash \langle rs, ? \rangle \Rightarrow \checkmark\}$$

$$\subseteq$$

$$\{p \mid \text{approx_firewall } \gamma \text{ permissive } rs = \checkmark\}$$

- ▶ Set of packets accepted by the firewall

Embedding in Ternary Logic

$$\mathbb{B} = \{True, False\} \quad Ternary = \{True, False, Unknown\}$$

$$\{p \mid \text{approx_firewall } \gamma \text{ stricter } rs = \checkmark\}$$

$$\subseteq$$

$$\{p \mid \gamma, p \vdash \langle rs, ? \rangle \Rightarrow \checkmark\}$$

$$\subseteq$$

$$\{p \mid \text{approx_firewall } \gamma \text{ permissive } rs = \checkmark\}$$

- ▶ Set of packets accepted by the firewall

Embedding in Ternary Logic

$$\mathbb{B} = \{True, False\} \quad Ternary = \{True, False, Unknown\}$$

$$\{p \mid \text{approx_firewall } \gamma \text{ stricter } rs = \checkmark\}$$

$$\subseteq$$

$$\{p \mid \gamma, p \vdash \langle rs, ? \rangle \Rightarrow \checkmark\} \longleftarrow \text{not executable}$$

$$\subseteq$$

$$\{p \mid \text{approx_firewall } \gamma \text{ permissive } rs = \checkmark\}$$

- ▶ Set of packets accepted by the firewall
- ▶ We can specify a lot . . .

Embedding in Ternary Logic

$$\mathbb{B} = \{True, False\}$$

$$Ternary = \{True, False, Unknown\}$$

$$\{p \mid \text{approx_firewall } \gamma \text{ stricter } rs = \checkmark\}$$

executable

$$\subseteq$$

$$\{p \mid \gamma, p \vdash \langle rs, ? \rangle \Rightarrow \checkmark\}$$

not executable

$$\subseteq$$

$$\{p \mid \text{approx_firewall } \gamma \text{ permissive } rs = \checkmark\}$$

executable

- ▶ Set of packets accepted by the firewall
- ▶ We can specify a lot . . . but we also believe in running code

Spoofing Protection

ipassmt :: interface \Rightarrow IP set

Example:

$$ipassmt = [\text{eth0} \mapsto 192.168.0.0/24]$$

Spoofing Protection:

$$\{p.\text{src_ip} \mid p.\text{in_iface} = \text{eth0} \wedge \gamma, p \vdash \langle rs, \textcircled{?} \rangle \Rightarrow \textcircled{\checkmark}\} \subseteq 192.168.0.0/24$$



Spoofing Protection

ipassmt :: interface \Rightarrow IP set

Example:

$$ipassmt = [\text{eth0} \mapsto 192.168.0.0/24]$$

Spoofing Protection:

$$\{p.\text{src_ip} \mid p.\text{in_iface} = \text{eth0} \wedge \gamma, p \vdash \langle rs, \textcircled{?} \rangle \Rightarrow \textcircled{\checkmark}\} \subseteq 192.168.0.0/24$$

Spoofing Protection

$ipassmt :: interface \Rightarrow IP\ set$

Example:

$$ipassmt = [eth0 \mapsto 192.168.0.0/24]$$

Spoofing Protection:

$$\{p.src_ip \mid p.in_iface = eth0 \wedge \gamma, p \vdash \langle rs, \textcircled{?} \rangle \Rightarrow \textcircled{\checkmark}\} \subseteq 192.168.0.0/24$$

spoofing Protection

ipassmt :: interface \Rightarrow IP set

Example:

$$ipassmt = [eth0 \mapsto 192.168.0.0/24]$$

Spoofing Protection:

$$\{p.src_ip \mid p.in_iface = eth0 \wedge \gamma, p \vdash \langle rs, \textcircled{?} \rangle \Rightarrow \textcircled{\checkmark}\} \subseteq 192.168.0.0/24$$

spoofing Protection

ipassmt :: interface \Rightarrow IP set

Example:

$$ipassmt = [eth0 \mapsto 192.168.0.0/24]$$

Spoofing Protection:

$$\{p.src_ip \mid p.in_iface = eth0 \wedge \gamma, p \vdash \langle rs, \textcircled{?} \rangle \Rightarrow \textcircled{\checkmark}\} \subseteq 192.168.0.0/24$$

Spoofting Protection

ipassmt :: interface \Rightarrow IP set

Example:

$$ipassmt = [eth0 \mapsto 192.168.0.0/24]$$

Spoofting Protection:

$$\{p.src_ip \mid p.in_iface = eth0 \wedge \gamma, p \vdash \langle rs, \textcircled{?} \rangle \Rightarrow \textcircled{\checkmark}\} \subseteq 192.168.0.0/24$$

Spoofing Protection

ipassmt :: interface \Rightarrow IP set

Example:

$$ipassmt = [eth0 \mapsto 192.168.0.0/24]$$

Spoofing Protection:

$$\forall eth \in ipassmt.keys$$

$$\{p.src_ip \mid p.in_iface = eth \wedge \gamma, p \vdash \langle rs, \textcircled{?} \rangle \Rightarrow \textcircled{\checkmark}\} \subseteq ipassmt.get(eth)$$

Spoofing Protection

ipassmt :: interface \Rightarrow IP set

Example:

$$ipassmt = [\text{eth0} \mapsto 192.168.0.0/24]$$

Spoofing Protection:

check_spoofing_protection *ipassmt rs* \longrightarrow

$\forall \text{eth} \in ipassmt.keys$

$\{p.src_ip \mid p.in_iface = \text{eth} \wedge \gamma, p \vdash \langle rs, ? \rangle \Rightarrow \checkmark\} \subseteq ipassmt.get(\text{eth})$

Spoofing Protection

ipassmt :: interface \Rightarrow IP set

Example:

$$ipassmt = [eth0 \mapsto 192.168.0.0/24]$$

Spoofing Protection:

check_spoofing_protection *ipassmt* *rs* \longrightarrow

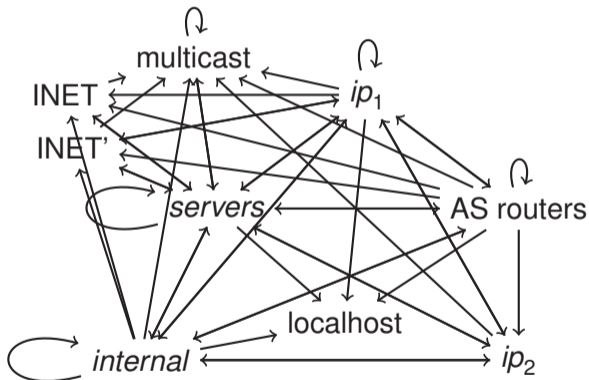
$\forall eth \in ipassmt.keys$

$$\{p.src_ip \mid p.in_iface = eth \wedge \gamma, p \vdash \langle rs, ? \rangle \Rightarrow \checkmark\} \subseteq ipassmt.get(eth)$$



```
diekmann@xps12: ~/glt/iptables_Semantics/beta_haskell_tool
$ iptables-save | ./check_spoofing_protection ipassmt_tumi8
loading ipassmt from `ipassmt_tumi8'
Parsed IpAssmt
sanity_wf_ruleset passed
Parsed 90 chains in table filter, a total of 4813 rules
Table `nat' : `Reading ruleset failed! sanity_wf_ruleset check failed.'.
Table `raw' : `Reading ruleset failed! sanity_wf_ruleset check failed.'.
sanity_wf_ruleset passed
(eth0,True)
(foo,False)
(eth1.96,True)
(eth1.108,True)
(eth1.109,True)
(eth1.110,False)
(eth1.116,True)
(eth1.152,True)
(eth1.171,True)
(eth1.173,True)
(eth1.1010,True)
(eth1.1011,True)
```

Service Matrix



- ▶ Partitions complete IPv4 space
- ▶ All IP addresses in each group have same access rights
- ▶ Cannot be compressed any further



Sources

Firewall Rulesets ← **plz contribute**

<https://github.com/diekman/net-network>

Isabelle Theories + Haskell Tool:

https://github.com/diekman/Iptables_Semantics