Chair of Network Architectures and Services
Department of Informatics
Technical University of Munich

Verified *iptables* Firewall Analysis

IFIP Networking 2016

Cornelius Diekmann, Julius Michaelis, Maximilian Haslbeck, and Georg Carle

Wednesday, May 18, 2016

Theorem Proving

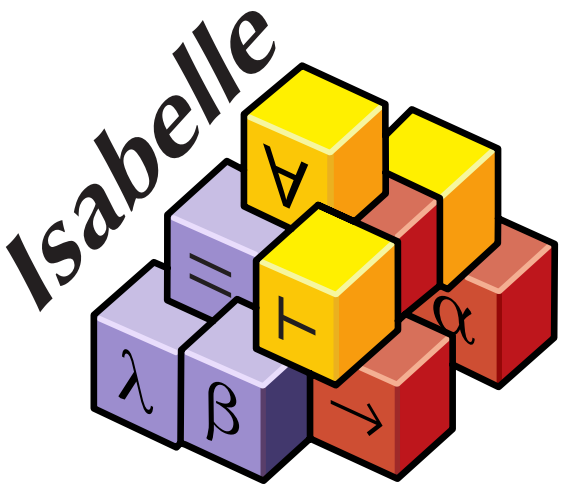
- This work: $\backsim$ 3 years, $\backsim$ 500 pages of manual proof

Theorem Proving

- This work: $\backsim$ 3 years, $\backsim$ 500 pages of manual proof

About Isabelle/HOL

- Interactive proof assistant

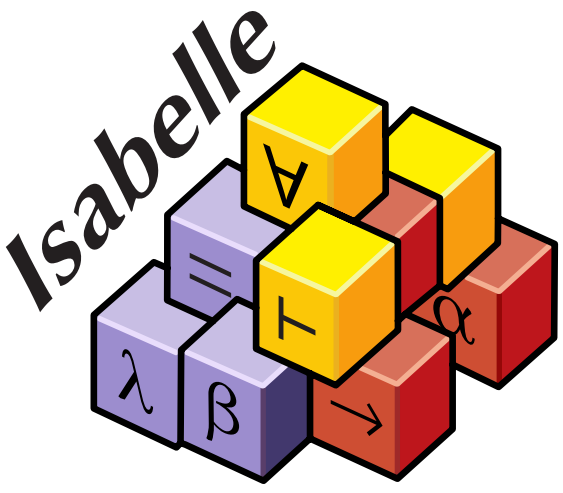


Theorem Proving

- This work: $\backsim$ 3 years, $\backsim$ 500 pages of manual proof

About Isabelle/HOL

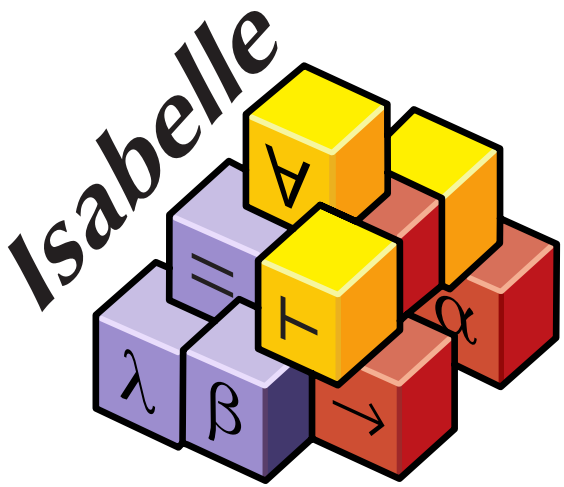
- Interactive proof assistant
- "theorem prover" $\neq$ automated theorem prover
  - Computers are **good** at **replaying** proofs
  - Computers are **bad** at **finding** proofs



Theorem Proving

- This work: $\backsim$ 3 years, $\backsim$ 500 pages of manual proof

About Isabelle/HOL



- Interactive proof assistant
- "theorem prover" $\neq$ automated theorem prover
  - Computers are **good** at **replaying** proofs
  - Computers are **bad** at **finding** proofs
- *Can we trust Isabelle?*
  - LCF-sytle mathematical micro kernel $\rightarrow$ code fits on screen
  - Over 20 years without a bug that affected a user's proof
  - Ask your formal methods colleague
  - How to *Common Criteria EAL 7*? $\rightarrow$ Use Isabelle (c.f. CC Appendix)

Problem Statement

Let's get practical

- Configuring firewalls is hard

Problem Statement

Let's get practical

- Configuring firewalls is hard
- Understanding ruleset of previous administrator $\rightarrow$ almost impossible

Problem Statement

Let's get practical

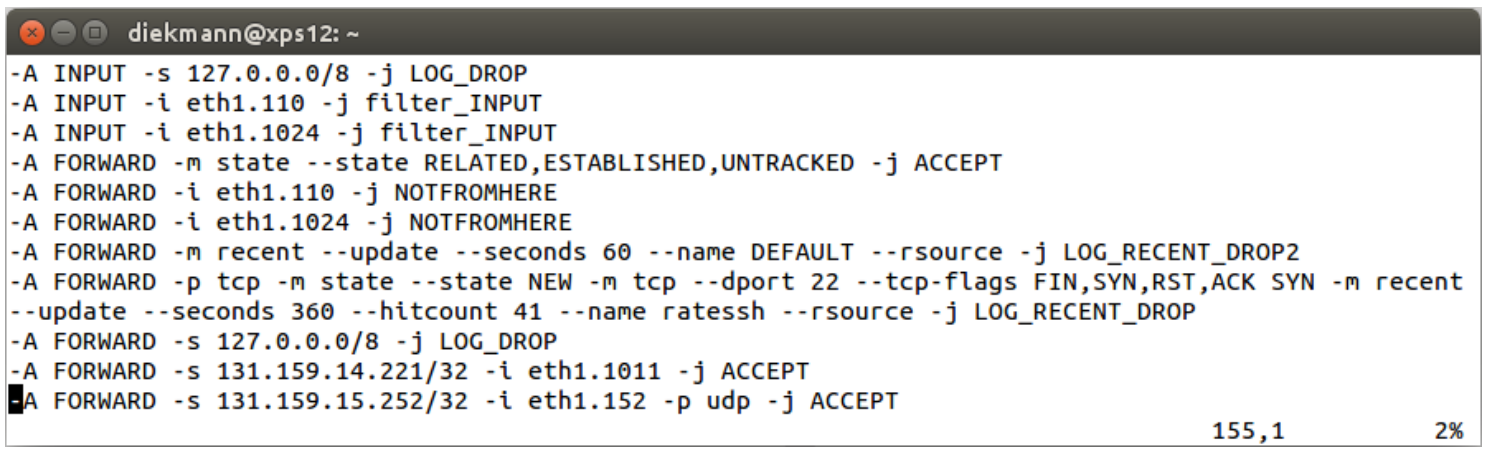
- Configuring firewalls is hard
- Understanding ruleset of previous administrator $\rightarrow$ almost impossible
- *Let's just consider packet filtering without modification*

Chair of Network Architectures and Services
Department of Informatics
Technical University of Munich

Problem Statement

Let's get practical

- Configuring firewalls is hard
- Understanding ruleset of previous administrator → almost impossible
- *Let's just consider packet filtering without modification*
- Linux/netfilter iptables firewall
  - In use for over 10 years → rulesets of that age
  - Over 200 packet matching options



Fun Examples

```
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [1745:334865]
:DEFAULT_INPUT - [0:0]
:DOS_PROTECT - [0:0]
-A INPUT -j DOS_PROTECT
-A INPUT -j DEFAULT_INPUT
-A DEFAULT_INPUT -i lo -j ACCEPT
-A DEFAULT_INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A DEFAULT_INPUT -p tcp -m tcp --dport 22 -j ACCEPT
-A DEFAULT_INPUT -p tcp -m multiport --dports 3262,3240:3259,21,... -j DROP
-A DEFAULT_INPUT -p tcp -m multiport --dports 22,23 -j DROP
-A DEFAULT_INPUT -s 192.168.0.0/16 -j ACCEPT
-A DEFAULT_INPUT -j DROP
-A DEFAULT_INPUT -i eth0 -j DROP
-A DOS_PROTECT -i eth0 -p icmp -m icmp --icmp-type 8 ... --limit 1/sec -j RETURN
-A DOS_PROTECT -i eth0 -p icmp -m icmp --icmp-type 8 -j DROP
-A DOS_PROTECT -i eth0 -p tcp -m tcp --tcp-flags ... --limit 1/sec -j RETURN
-A DOS_PROTECT -i eth0 -p tcp -m tcp --tcp-flags ... -j DROP
-A DOS_PROTECT -i eth0 -p tcp -m tcp --tcp-flags ... --limit-burst 100 -j RETURN
-A DOS_PROTECT -i eth0 -p tcp -m tcp --tcp-flags ... -j DROP
COMMIT
```

Fun Examples

```
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [1745:334865]
:DEFAULT_INPUT - [0:0]
:DOS_PROTECT - [0:0]
-A INPUT -j DOS_PROTECT
-A INPUT -j DEFAULT_INPUT
-A DEFAULT_INPUT -i lo -j ACCEPT
-A DEFAULT_INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A DEFAULT_INPUT -p tcp -m tcp --dport 22 -j ACCEPT
-A DEFAULT_INPUT -p tcp -m multiport --dports 3262,3240:3259,21,... -j DROP
-A DEFAULT_INPUT -p tcp -m multiport --dports 22,23 -j DROP
-A DEFAULT_INPUT -s 192.168.0.0/16 -j ACCEPT
-A DEFAULT_INPUT -j DROP
-A DEFAULT_INPUT -i eth0 -j DROP
-A DOS_PROTECT -i eth0 -p icmp -m icmp --icmp-type 8 ... --limit 1/sec -j RETURN
-A DOS_PROTECT -i eth0 -p icmp -m icmp --icmp-type 8 -j DROP
-A DOS_PROTECT -i eth0 -p tcp -m tcp --tcp-flags ... --limit 1/sec -j RETURN
-A DOS_PROTECT -i eth0 -p tcp -m tcp --tcp-flags ... -j DROP
-A DOS_PROTECT -i eth0 -p tcp -m tcp --tcp-flags ... --limit-burst 100 -j RETURN
-A DOS_PROTECT -i eth0 -p tcp -m tcp --tcp-flags ... -j DROP
COMMIT
```

Fun Examples

```
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [1745:334865]
:DEFAULT_INPUT - [0:0]
:DOS_PROTECT - [0:0]
-A INPUT -j DOS_PROTECT
-A INPUT -j DEFAULT_INPUT
-A DEFAULT_INPUT -i lo -j ACCEPT
-A DEFAULT_INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A DEFAULT_INPUT -p tcp -m tcp --dport 22 -j ACCEPT
-A DEFAULT_INPUT -p tcp -m multiport --dports 3262,3240:3259,21,... -j DROP
-A DEFAULT_INPUT -p tcp -m multiport --dports 22,23 -j DROP
-A DEFAULT_INPUT -s 192.168.0.0/16 -j ACCEPT
-A DEFAULT_INPUT -j DROP
-A DEFAULT_INPUT -i eth0 -j DROP
-A DOS_PROTECT -i eth0 -p icmp -m icmp --icmp-type 8 ... --limit 1/sec -j RETURN
-A DOS_PROTECT -i eth0 -p icmp -m icmp --icmp-type 8 -j DROP
-A DOS_PROTECT -i eth0 -p tcp -m tcp --tcp-flags ... --limit 1/sec -j RETURN
-A DOS_PROTECT -i eth0 -p tcp -m tcp --tcp-flags ... -j DROP
-A DOS_PROTECT -i eth0 -p tcp -m tcp --tcp-flags ... --limit-burst 100 -j RETURN
-A DOS_PROTECT -i eth0 -p tcp -m tcp --tcp-flags ... -j DROP
COMMIT
```

Fun Examples

```
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [1745:334865]
:DEFAULT_INPUT - [0:0]
:DOS_PROTECT - [0:0]
-A INPUT -j DOS_PROTECT
-A INPUT -j DEFAULT_INPUT
-A DEFAULT_INPUT -i lo -j ACCEPT
-A DEFAULT_INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A DEFAULT_INPUT -p tcp -m tcp --dport 22 -j ACCEPT
-A DEFAULT_INPUT -p tcp -m multiport --dports 3262,3240:3259,21,... -j DROP
-A DEFAULT_INPUT -p tcp -m multiport --dports 22,23 -j DROP
-A DEFAULT_INPUT -s 192.168.0.0/16 -j ACCEPT
-A DEFAULT_INPUT -j DROP
-A DEFAULT_INPUT -i eth0 -j DROP
-A DOS_PROTECT -i eth0 -p icmp -m icmp --icmp-type 8 ... --limit 1/sec -j RETURN
-A DOS_PROTECT -i eth0 -p icmp -m icmp --icmp-type 8 -j DROP
-A DOS_PROTECT -i eth0 -p tcp -m tcp --tcp-flags ... --limit 1/sec -j RETURN
-A DOS_PROTECT -i eth0 -p tcp -m tcp --tcp-flags ... -j DROP
-A DOS_PROTECT -i eth0 -p tcp -m tcp --tcp-flags ... --limit-burst 100 -j RETURN
-A DOS_PROTECT -i eth0 -p tcp -m tcp --tcp-flags ... -j DROP
COMMIT
```

Fun Examples

```
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [1745:334865]
:DEFAULT_INPUT - [0:0]
:DOS_PROTECT - [0:0]
-A INPUT -j DOS_PROTECT
-A INPUT -j DEFAULT_INPUT
-A DEFAULT_INPUT -i lo -j ACCEPT
-A DEFAULT_INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A DEFAULT_INPUT -p tcp -m tcp --dport 22 -j ACCEPT
-A DEFAULT_INPUT -p tcp -m multiport --dports 3262,3240:3259,21,... -j DROP
-A DEFAULT_INPUT -p tcp -m multiport --dports 22,23 -j DROP
-A DEFAULT_INPUT -s 192.168.0.0/16 -j ACCEPT
-A DEFAULT_INPUT -j DROP
-A DEFAULT_INPUT -i eth0 -j DROP
-A DOS_PROTECT -i eth0 -p icmp -m icmp --icmp-type 8 ... --limit 1/sec -j RETURN
-A DOS_PROTECT -i eth0 -p icmp -m icmp --icmp-type 8 -j DROP
-A DOS_PROTECT -i eth0 -p tcp -m tcp --tcp-flags ... --limit 1/sec -j RETURN
-A DOS_PROTECT -i eth0 -p tcp -m tcp --tcp-flags ... -j DROP
-A DOS_PROTECT -i eth0 -p tcp -m tcp --tcp-flags ... --limit-burst 100 -j RETURN
-A DOS_PROTECT -i eth0 -p tcp -m tcp --tcp-flags ... -j DROP
COMMIT
```

Fun Examples

```
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [1745:334865]
:DEFAULT_INPUT - [0:0]
:DOS_PROTECT - [0:0]
-A INPUT -j DOS_PROTECT
-A INPUT -j DEFAULT_INPUT
-A DEFAULT_INPUT -i lo -j ACCEPT
-A DEFAULT_INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A DEFAULT_INPUT -p tcp -m tcp --dport 22 -j ACCEPT
-A DEFAULT_INPUT -p tcp -m multiport --dports 3262,3240:3259,21,... -j DROP
-A DEFAULT_INPUT -p tcp -m multiport --dports 22,23 -j DROP
-A DEFAULT_INPUT -s 192.168.0.0/16 -j ACCEPT
-A DEFAULT_INPUT -j DROP
-A DEFAULT_INPUT -i eth0 -j DROP
-A DOS_PROTECT -i eth0 -p icmp -m icmp --icmp-type 8 ... --limit 1/sec -j RETURN
-A DOS_PROTECT -i eth0 -p icmp -m icmp --icmp-type 8 -j DROP
-A DOS_PROTECT -i eth0 -p tcp -m tcp --tcp-flags ... --limit 1/sec -j RETURN
-A DOS_PROTECT -i eth0 -p tcp -m tcp --tcp-flags ... -j DROP
-A DOS_PROTECT -i eth0 -p tcp -m tcp --tcp-flags ... --limit-burst 100 -j RETURN
-A DOS_PROTECT -i eth0 -p tcp -m tcp --tcp-flags ... -j DROP
COMMIT
```

Fun Examples

```
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [1745:334865]
:DEFAULT_INPUT - [0:0]
:DOS_PROTECT - [0:0]
-A INPUT -j DOS_PROTECT
-A INPUT -j DEFAULT_INPUT
-A DEFAULT_INPUT -i lo -j ACCEPT
-A DEFAULT_INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A DEFAULT_INPUT -p tcp -m tcp --dport 22 -j ACCEPT
-A DEFAULT_INPUT -p tcp -m multiport --dports 3262,3240:3259,21,... -j DROP
-A DEFAULT_INPUT -p tcp -m multiport --dports 22,23 -j DROP
-A DEFAULT_INPUT -s 192.168.0.0/16 -j ACCEPT
-A DEFAULT_INPUT -j DROP
-A DEFAULT_INPUT -i eth0 -j DROP
-A DOS_PROTECT -i eth0 -p icmp -m icmp --icmp-type 8 ... --limit 1/sec -j RETURN
-A DOS_PROTECT -i eth0 -p icmp -m icmp --icmp-type 8 -j DROP
-A DOS_PROTECT -i eth0 -p tcp -m tcp --tcp-flags ... --limit 1/sec -j RETURN
-A DOS_PROTECT -i eth0 -p tcp -m tcp --tcp-flags ... -j DROP
-A DOS_PROTECT -i eth0 -p tcp -m tcp --tcp-flags ... --limit-burst 100 -j RETURN
-A DOS_PROTECT -i eth0 -p tcp -m tcp --tcp-flags ... -j DROP
COMMIT
```

Fun Examples

```
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [1745:334865]
:DEFAULT_INPUT - [0:0]
:DOS_PROTECT - [0:0]
-A INPUT -j DOS_PROTECT
-A INPUT -j DEFAULT_INPUT
-A DEFAULT_INPUT -i lo -j ACCEPT
-A DEFAULT_INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A DEFAULT_INPUT -p tcp -m tcp --dport 22 -j ACCEPT
-A DEFAULT_INPUT -p tcp -m multiport --dports 3262,3240:3259,21,... -j DROP
-A DEFAULT_INPUT -p tcp -m multiport --dports 22,23 -j DROP
-A DEFAULT_INPUT -s 192.168.0.0/16 -j ACCEPT
-A DEFAULT_INPUT -j DROP
-A DEFAULT_INPUT -i eth0 -j DROP
-A DOS_PROTECT -i eth0 -p icmp -m icmp --icmp-type 8 ... --limit 1/sec -j RETURN
-A DOS_PROTECT -i eth0 -p icmp -m icmp --icmp-type 8 -j DROP
-A DOS_PROTECT -i eth0 -p tcp -m tcp --tcp-flags ... --limit 1/sec -j RETURN
-A DOS_PROTECT -i eth0 -p tcp -m tcp --tcp-flags ... -j DROP
-A DOS_PROTECT -i eth0 -p tcp -m tcp --tcp-flags ... --limit-burst 100 -j RETURN
-A DOS_PROTECT -i eth0 -p tcp -m tcp --tcp-flags ... -j DROP
COMMIT
```

Fun Examples

```
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [1745:334865]
:DEFAULT_INPUT - [0:0]
:DOS_PROTECT - [0:0]
-A INPUT -j DOS_PROTECT
-A INPUT -j DEFAULT_INPUT
-A DEFAULT_INPUT -i lo -j ACCEPT
-A DEFAULT_INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A DEFAULT_INPUT -p tcp -m tcp --dport 22 -j ACCEPT
-A DEFAULT_INPUT -p tcp -m multiport --dports 3262,3240:3259,21,... -j DROP
-A DEFAULT_INPUT -p tcp -m multiport --dports 22,23 -j DROP
-A DEFAULT_INPUT -s 192.168.0.0/16 -j ACCEPT
-A DEFAULT_INPUT -j DROP
-A DEFAULT_INPUT -i eth0 -j DROP
-A DOS_PROTECT -i eth0 -p icmp -m icmp --icmp-type 8 ... --limit 1/sec -j RETURN
-A DOS_PROTECT -i eth0 -p icmp -m icmp --icmp-type 8 -j DROP
-A DOS_PROTECT -i eth0 -p tcp -m tcp --tcp-flags ... --limit 1/sec -j RETURN
-A DOS_PROTECT -i eth0 -p tcp -m tcp --tcp-flags ... -j DROP
-A DOS_PROTECT -i eth0 -p tcp -m tcp --tcp-flags ... --limit-burst 100 -j RETURN
-A DOS_PROTECT -i eth0 -p tcp -m tcp --tcp-flags ... -j DROP
COMMIT
```

Fun Examples

```
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [1745:334865]
:DEFAULT_INPUT - [0:0]
:DOS_PROTECT - [0:0]
-A INPUT -j DOS_PROTECT
-A INPUT -j DEFAULT_INPUT
-A DEFAULT_INPUT -i lo -j ACCEPT
-A DEFAULT_INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A DEFAULT_INPUT -p tcp -m tcp --dport 22 -j ACCEPT
-A DEFAULT_INPUT -p tcp -m multiport --dports 3262,3240:3259,21,... -j DROP
-A DEFAULT_INPUT -p tcp -m multiport --dports 22,23 -j DROP
-A DEFAULT_INPUT -s 192.168.0.0/16 -j ACCEPT
-A DEFAULT_INPUT -j DROP
-A DEFAULT_INPUT -i eth0 -j DROP
-A DOS_PROTECT -i eth0 -p icmp -m icmp --icmp-type 8 ... --limit 1/sec -j RETURN
-A DOS_PROTECT -i eth0 -p icmp -m icmp --icmp-type 8 -j DROP
-A DOS_PROTECT -i eth0 -p tcp -m tcp --tcp-flags ... --limit 1/sec -j RETURN
-A DOS_PROTECT -i eth0 -p tcp -m tcp --tcp-flags ... -j DROP
-A DOS_PROTECT -i eth0 -p tcp -m tcp --tcp-flags ... --limit-burst 100 -j RETURN
-A DOS_PROTECT -i eth0 -p tcp -m tcp --tcp-flags ... -j DROP
COMMIT
```

Fun Examples

```
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [1745:334865]
:DEFAULT_INPUT - [0:0]
:DOS_PROTECT - [0:0]
-A INPUT -j DOS_PROTECT
-A INPUT -j DEFAULT_INPUT
-A DEFAULT_INPUT -i lo -j ACCEPT
-A DEFAULT_INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A DEFAULT_INPUT -p tcp -m tcp --dport 22 -j ACCEPT
-A DEFAULT_INPUT -p tcp -m multiport --dports 3262,3240:3259,21,... -j DROP
-A DEFAULT_INPUT -p tcp -m multiport --dports 22,23 -j DROP
-A DEFAULT_INPUT -s 192.168.0.0/16 -j ACCEPT
-A DEFAULT_INPUT -j DROP
-A DEFAULT_INPUT -i eth0 -j DROP
-A DOS_PROTECT -i eth0 -p icmp -m icmp --icmp-type 8 ... --limit 1/sec -j RETURN
-A DOS_PROTECT -i eth0 -p icmp -m icmp --icmp-type 8 -j DROP
-A DOS_PROTECT -i eth0 -p tcp -m tcp --tcp-flags ... --limit 1/sec -j RETURN
-A DOS_PROTECT -i eth0 -p tcp -m tcp --tcp-flags ... -j DROP
-A DOS_PROTECT -i eth0 -p tcp -m tcp --tcp-flags ... --limit-burst 100 -j RETURN
-A DOS_PROTECT -i eth0 -p tcp -m tcp --tcp-flags ... -j DROP
COMMIT
```

Fun Examples

```
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [1745:334865]
:DEFAULT_INPUT - [0:0]
:DOS_PROTECT - [0:0]
-A INPUT -j DOS_PROTECT
-A INPUT -j DEFAULT_INPUT
-A DEFAULT_INPUT -i lo -j ACCEPT
-A DEFAULT_INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A DEFAULT_INPUT -p tcp -m tcp --dport 22 -j ACCEPT
-A DEFAULT_INPUT -p tcp -m multiport --dports 3262,3240:3259,21,... -j DROP
-A DEFAULT_INPUT -p tcp -m multiport --dports 22,23 -j DROP
-A DEFAULT_INPUT -s 192.168.0.0/16 -j ACCEPT
-A DEFAULT_INPUT -j DROP
-A DEFAULT_INPUT -i eth0 -j DROP
-A DOS_PROTECT -i eth0 -p icmp -m icmp --icmp-type 8 ... --limit 1/sec -j RETURN
-A DOS_PROTECT -i eth0 -p icmp -m icmp --icmp-type 8 -j DROP
-A DOS_PROTECT -i eth0 -p tcp -m tcp --tcp-flags ... --limit 1/sec -j RETURN
-A DOS_PROTECT -i eth0 -p tcp -m tcp --tcp-flags ... -j DROP
-A DOS_PROTECT -i eth0 -p tcp -m tcp --tcp-flags ... --limit-burst 100 -j RETURN
-A DOS_PROTECT -i eth0 -p tcp -m tcp --tcp-flags ... -j DROP
COMMIT
```

Fun Examples

```
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [1745:334865]
:DEFAULT_INPUT - [0:0]
:DOS_PROTECT - [0:0]
-A INPUT -j DOS_PROTECT
-A INPUT -j DEFAULT_INPUT
-A DEFAULT_INPUT -i lo -j ACCEPT
-A DEFAULT_INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A DEFAULT_INPUT -p tcp -m tcp --dport 22 -j ACCEPT
-A DEFAULT_INPUT -p tcp -m multiport --dports 3262,3240:3259,21,... -j DROP
-A DEFAULT_INPUT -p tcp -m multiport --dports 22,23 -j DROP
-A DEFAULT_INPUT -s 192.168.0.0/16 -j ACCEPT
-A DEFAULT_INPUT -j DROP
-A DEFAULT_INPUT -i eth0 -j DROP
-A DOS_PROTECT -i eth0 -p icmp -m icmp --icmp-type 8 ... --limit 1/sec -j RETURN
-A DOS_PROTECT -i eth0 -p icmp -m icmp --icmp-type 8 -j DROP
-A DOS_PROTECT -i eth0 -p tcp -m tcp --tcp-flags ... --limit 1/sec -j RETURN
-A DOS_PROTECT -i eth0 -p tcp -m tcp --tcp-flags ... -j DROP
-A DOS_PROTECT -i eth0 -p tcp -m tcp --tcp-flags ... --limit-burst 100 -j RETURN
-A DOS_PROTECT -i eth0 -p tcp -m tcp --tcp-flags ... -j DROP
COMMIT
```

Fun Examples

```
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [1745:334865]
:DEFAULT_INPUT - [0:0]
:DOS_PROTECT - [0:0]
-A INPUT -j DOS_PROTECT
-A INPUT -j DEFAULT_INPUT
-A DEFAULT_INPUT -i lo -j ACCEPT
-A DEFAULT_INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A DEFAULT_INPUT -p tcp -m tcp --dport 22 -j ACCEPT
-A DEFAULT_INPUT -p tcp -m multiport --dports 3262,3240:3259,21,... -j DROP
-A DEFAULT_INPUT -p tcp -m multiport --dports 22,23 -j DROP
-A DEFAULT_INPUT -s 192.168.0.0/16 -j ACCEPT
-A DEFAULT_INPUT -j DROP
-A DEFAULT_INPUT -i eth0 -j DROP
-A DOS_PROTECT -i eth0 -p icmp -m icmp --icmp-type 8 ... --limit 1/sec -j RETURN
-A DOS_PROTECT -i eth0 -p icmp -m icmp --icmp-type 8 -j DROP
-A DOS_PROTECT -i eth0 -p tcp -m tcp --tcp-flags ... --limit 1/sec -j RETURN
-A DOS_PROTECT -i eth0 -p tcp -m tcp --tcp-flags ... -j DROP
-A DOS_PROTECT -i eth0 -p tcp -m tcp --tcp-flags ... --limit-burst 100 -j RETURN
-A DOS_PROTECT -i eth0 -p tcp -m tcp --tcp-flags ... -j DROP
COMMIT
```

More Fun Examples

```
-A FORWARD -p tcp -m tcp --sport 410:415 ↩
 -m time  --timestart 06:59 --timestop 23:59 ↩
 --days Sun,Mon,Tue,Wed,Thu,Fri,Sat -j DROP

-A FORWARD -p tcp -m time  --timestart 06:59 ↩
 --timestop 23:59 --days Sun,Mon,Tue,Wed,Thu,Fri,Sat ↩
 -m string --string X-Kazaa-User -j DROP

-A FORWARD -s  192.168.1.1 -p tcp --syn ↩
 -m mac --mac 00:60:08:76:35:51 ↩
 -m connlimit --connlimit-above 15 -j REJECT
```

Chair of Network Architectures and Services
Department of Informatics
Technical University of Munich

More Fun Examples

```
-A FORWARD -p tcp -m tcp --sport 410:415 ↩
 -m time  --timestart 06:59 --timestop 23:59 ↩
 --days Sun,Mon,Tue,Wed,Thu,Fri,Sat -j DROP

-A FORWARD -p tcp -m time  --timestart 06:59 ↩
 --timestop 23:59 --days Sun,Mon,Tue,Wed,Thu,Fri,Sat ↩
 -m string --string X-Kazaa-User -j DROP

-A FORWARD -s  192.168.1.1 -p tcp --syn ↩
 -m mac --mac 00:60:08:76:35:51 ↩
 -m connlimit --connlimit-above 15 -j REJECT
```

Chair of Network Architectures and Services
Department of Informatics
Technical University of Munich

More Fun Examples

```
-A FORWARD -p tcp -m tcp --sport 410:415 ↩
 -m time  --timestart 06:59 --timestop 23:59 ↩
 --days Sun,Mon,Tue,Wed,Thu,Fri,Sat -j DROP

-A FORWARD -p tcp -m time  --timestart 06:59 ↩
 --timestop 23:59 --days Sun,Mon,Tue,Wed,Thu,Fri,Sat ↩
 -m string --string X-Kazaa-User -j DROP

-A FORWARD -s  192.168.1.1 -p tcp --syn ↩
 -m mac --mac 00:60:08:76:35:51 ↩
 -m connlimit --connlimit-above 15 -j REJECT
```

Get all the data at

`https://github.com/diekmann/net-network/`

Requirements

- Requirement 1: A simple model for packet filtering

Requirements

- Requirement 1: A simple model for packet filtering
- Requirement 2: Applicable to real-world

Simple Firewall Model

$$\begin{aligned}
\textsf{simple-fw}\ []\ p &= ? \\
\textsf{simple-fw}((m,\ \texttt{Accept}) :: rs)\ p &= \text{if } \textsf{match}\ m\ p \text{ then } ✓ \text{ else } \textsf{simple-fw}\ rs\ p \\
\textsf{simple-fw}((m,\ \texttt{Drop}) :: rs)\ p &= \text{if } \textsf{match}\ m\ p \text{ then } ✗ \text{ else } \textsf{simple-fw}\ rs\ p
\end{aligned}$$

where match can only match on

- in/out interface, including support for the '+' wildcard
- src/dst IP address range in CIDR notation, e.g. 192.168.0.0/24
- protocol ($\star$, or any numeric protocol identifier)
- src/dst interval of ports, e.g. 0:65535

Main Theorem

$$\{p.\ \ \text{new}\ p\ \wedge\ \Gamma, \gamma, p \vdash \langle rs,\ ⓘ \rangle \Rightarrow ✓\}$$
$$\subseteq$$
$$\{p.\ \ \text{new}\ p\ \wedge\ \text{simple-fw}\ (\text{translate-oapprox}\ rs) = ✓\}$$

Chair of Network Architectures and Services
Department of Informatics
Technical University of Munich

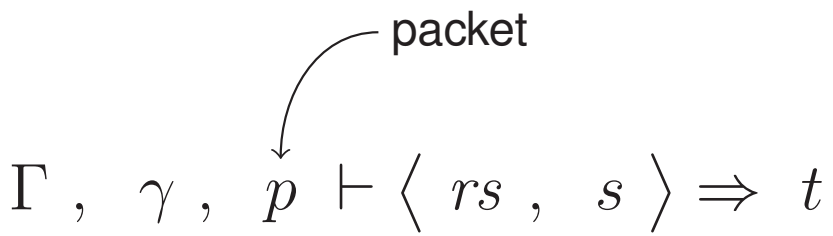
Main Theorem

$$\{p.\ \text{new}\ p\ \wedge\ \Gamma, \gamma, p \vdash \langle rs,\ ?\rangle \Rightarrow \checkmark\}$$
$$\subseteq$$
$$\{p.\ \text{new}\ p\ \wedge\ \text{simple-fw}\ (\text{translate-oapprox}\ rs) = \checkmark\}$$

Iptables Semantics

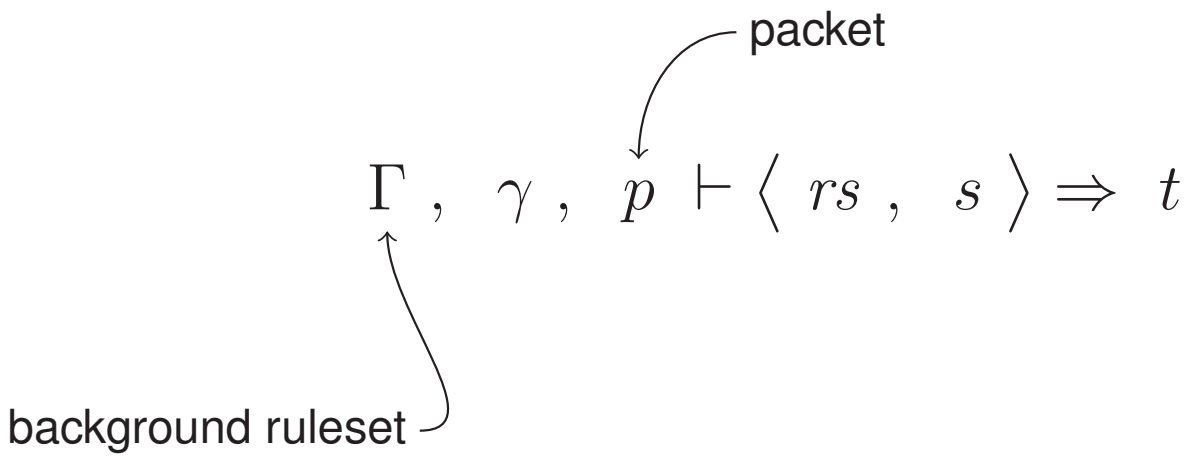
$$\Gamma\ ,\ \ \gamma\ ,\ \ p\ \vdash \left\langle\ rs\ ,\ \ s\ \right\rangle \Rightarrow\ t$$

Iptables Semantics

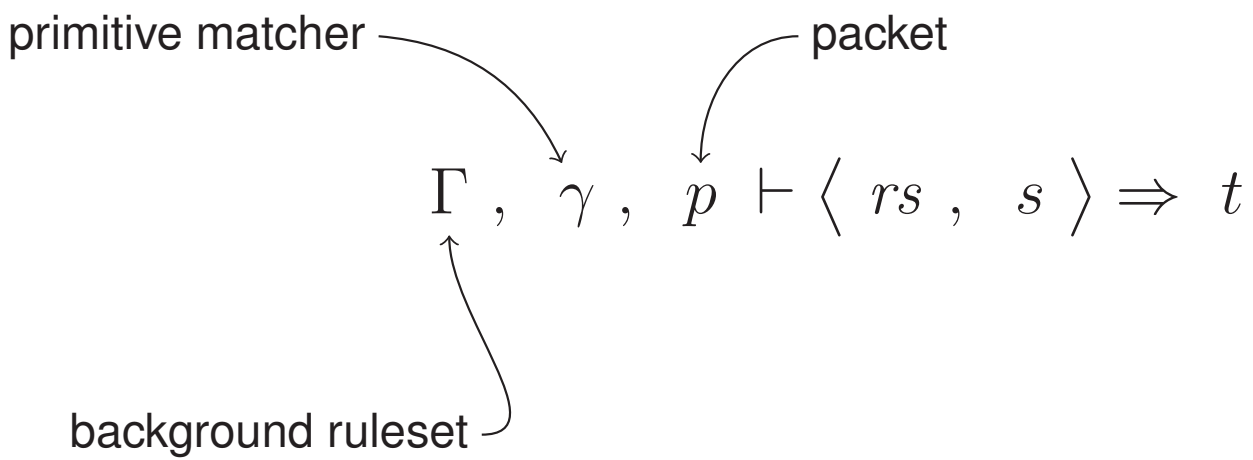


Chair of Network Architectures and Services
Department of Informatics
Technical University of Munich

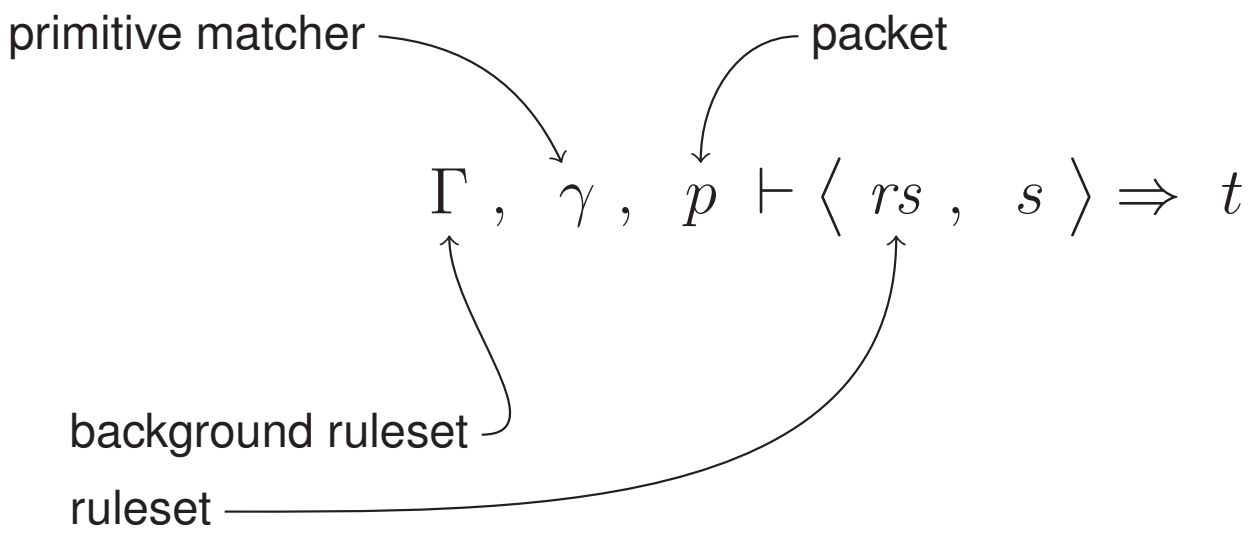
Iptables Semantics



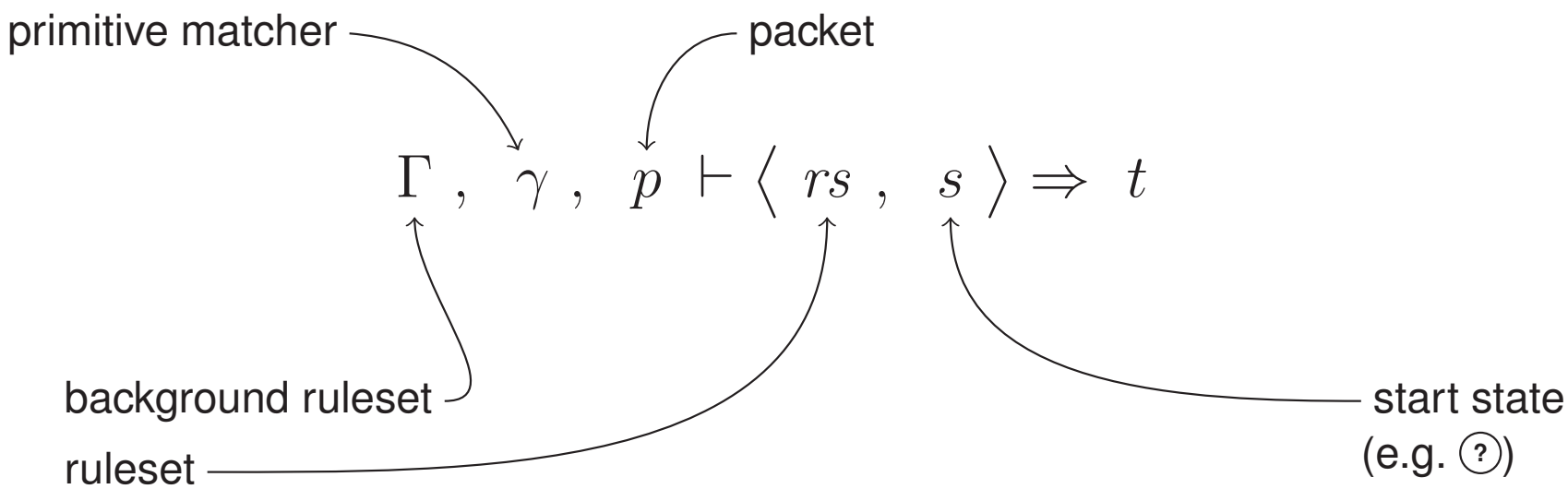
Iptables Semantics



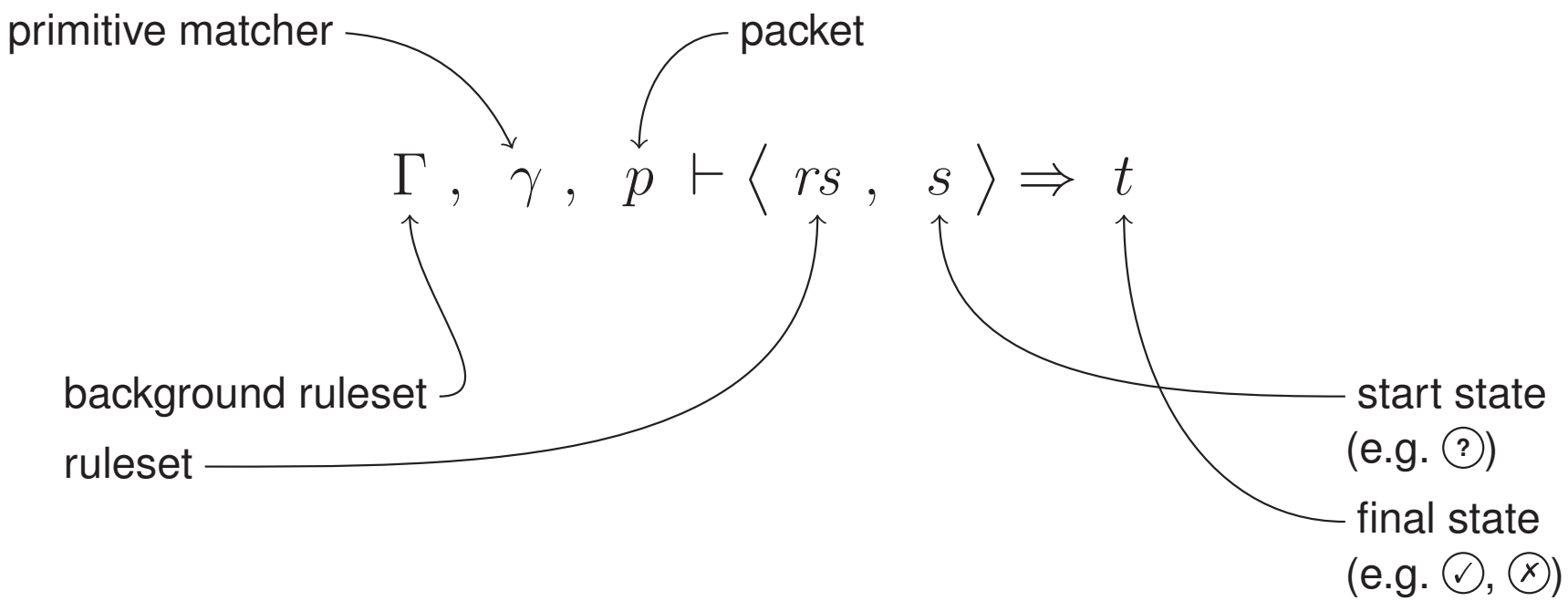
Iptables Semantics



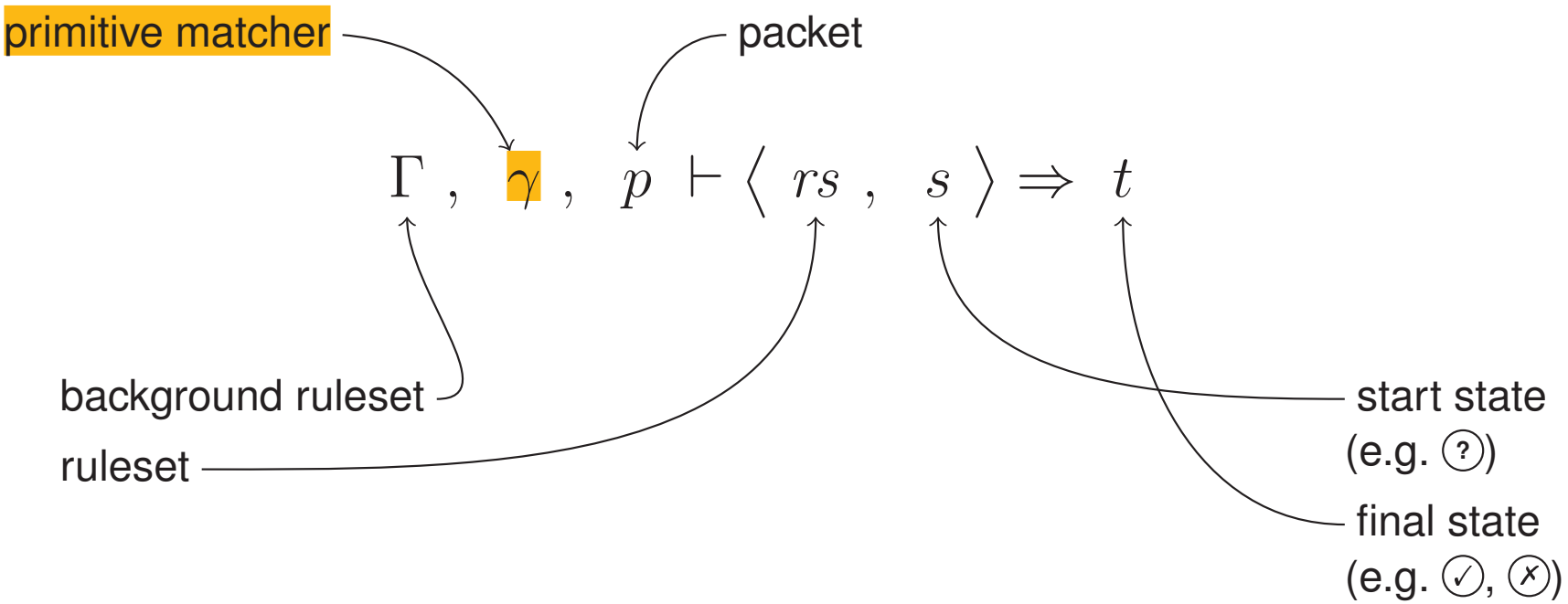
Iptables Semantics



Iptables Semantics

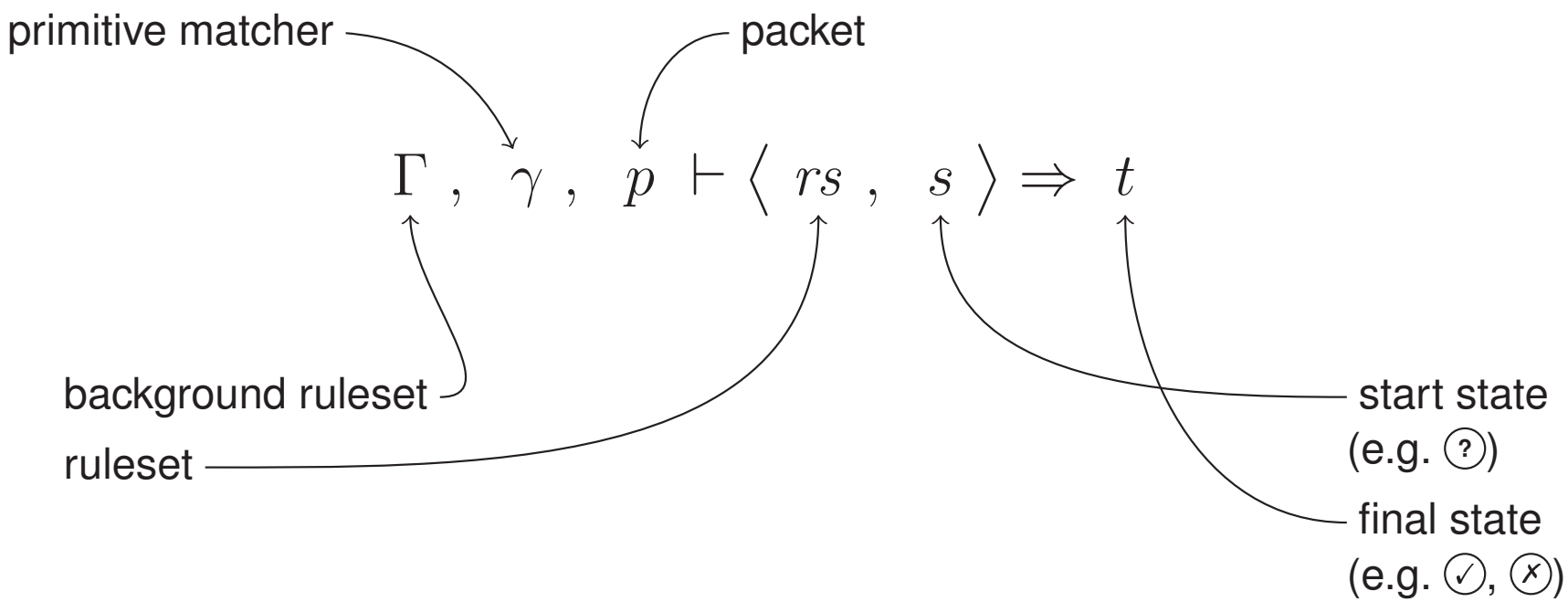


Iptables Semantics



- Arbitrary function: $\gamma :: (primitive \Rightarrow packet \Rightarrow \mathbb{B})$

Iptables Semantics



- Arbitrary function: $\gamma :: (primitive \Rightarrow packet \Rightarrow \mathbb{B})$
- 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

Chair of Network Architectures and Services
Department of Informatics
Technical University of Munich

Towards the Main Theorem

$$\{p.\quad \Gamma, \gamma, p \vdash \langle rs,\ ⓘ \rangle \Rightarrow ✓\}$$

- Set of all packets accepted by the **real** firewall

Towards the Main Theorem

$$\{p.\quad \Gamma, \gamma, p \vdash \langle rs,\ ? \rangle \Rightarrow \checkmark\}$$

- Set of all packets accepted by the **real** firewall

$$\{p.\quad \text{simple-fw } rs' = \checkmark\}$$

- Set of all packets accepted by the **simple** firewall

Chair of Network Architectures and Services
Department of Informatics
Technical University of Munich

Towards the Main Theorem

$$\{p.\quad \Gamma, \gamma, p \vdash \langle rs,\ ? \rangle \Rightarrow \checkmark\}$$

- Set of all packets accepted by the **real** firewall

$$\{p.\quad \text{simple-fw } rs' = \checkmark\}$$

- Set of all packets accepted by the **simple** firewall

- $rs \neq rs'$

Towards the Main Theorem

$$\{p.\ \ \Gamma, \gamma, p \vdash \langle rs,\ ? \rangle \Rightarrow \checkmark\}$$
$$\subseteq$$
$$\{p.\ \ \text{simple-fw}\ (\ \text{translate-oapprox}\ \ rs) = \checkmark\}$$

Chair of Network Architectures and Services
Department of Informatics
Technical University of Munich

Towards the Main Theorem

$$\{p.\quad \Gamma, \gamma, p \vdash \langle rs,\ ? \rangle \Rightarrow \checkmark\}$$

$$\subseteq$$

$$\{p.\quad \text{simple-fw}\ (\ \text{translate-oapprox}\ \ rs) = \checkmark\}$$

Chair of Network Architectures and Services
Department of Informatics
Technical University of Munich

Towards the Main Theorem

$$\{p.\quad \Gamma, \gamma, p \vdash \langle rs,\ ? \rangle \Rightarrow \checkmark\}$$
$$\subseteq$$
$$\{p.\quad \text{simple-fw}\ (\ \text{translate-oapprox}\ \ rs) = \checkmark\}$$

Chair of Network Architectures and Services
Department of Informatics
Technical University of Munich

Towards the Main Theorem

$$\{p.\ \ \Gamma, \gamma, p \vdash \langle rs,\ ? \rangle \Rightarrow \checkmark\}$$
$$\subseteq$$
$$\{p.\ \ \text{simple-fw}\ (\ \text{translate-oapprox}\ \ rs) = \checkmark\}$$

Main Contribution (#1)

Main Theorem

$$\{p.\ \ \text{new}\ p\ \wedge\ \Gamma, \gamma, p \vdash \langle rs,\ ⓘ \rangle \Rightarrow ✓\}$$
$$\subseteq$$
$$\{p.\ \ \text{new}\ p\ \wedge\ \text{simple-fw}\ (\text{translate-oapprox}\ rs) = ✓\}$$

Chair of Network Architectures and Services
Department of Informatics
Technical University of Munich

Main Theorem

$$\{p.\ \ \text{new}\ p \wedge \Gamma, \gamma, p \vdash \langle rs,\ ? \rangle \Rightarrow \checkmark\}$$
$$\subseteq$$
$$\{p.\ \ \text{new}\ p \wedge \text{simple-fw}\ (\text{translate-oapprox}\ rs) = \checkmark\}$$

Example

```
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT DROP [0:0]
:FOO - [0:0]
-A FORWARD -s 10.0.0.0/8 -j FOO
-A FOO ! -s 10.0.0.0/9 -j DROP
-A FOO -p tcp --bar -j ACCEPT
COMMIT
```

Example

```
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT DROP [0:0]
:FOO - [0:0]
-A FORWARD -s 10.0.0.0/8 -j FOO
-A FOO ! -s 10.0.0.0/9 -j DROP
-A FOO -p tcp --bar -j ACCEPT
COMMIT
```

Example

```
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT DROP [0:0]
:FOO - [0:0]
-A FORWARD -s 10.0.0.0/8 -j FOO
-A FOO ! -s 10.0.0.0/9 -j DROP
-A FOO -p tcp --bar -j ACCEPT
COMMIT
```

Example

```
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT DROP [0:0]
:FOO - [0:0]
-A FORWARD -s 10.0.0.0/8 -j FOO
-A FOO ! -s 10.0.0.0/9 -j DROP
-A FOO -p tcp --bar -j ACCEPT
COMMIT
```

Example

```
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT DROP [0:0]
:FOO - [0:0]
-A FORWARD -s 10.0.0.0/8 -j FOO
-A FOO ! -s 10.0.0.0/9 -j DROP
-A FOO -p tcp --bar -j ACCEPT
COMMIT
```

Example

```
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT DROP [0:0]
:FOO - [0:0]
-A FORWARD -s 10.0.0.0/8 -j FOO
-A FOO ! -s 10.0.0.0/9 -j DROP
-A FOO -p tcp --bar -j ACCEPT
COMMIT
```

Example

```
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT DROP [0:0]
:FOO - [0:0]
-A FORWARD -s 10.0.0.0/8 -j FOO
-A FOO ! -s 10.0.0.0/9 -j DROP
-A FOO -p tcp --bar -j ACCEPT
COMMIT
```

```
$ ./fffuu iptables-save.txt
target     prot   source                destination
DROP       all    10.128.0.0/9          0.0.0.0/0
ACCEPT     tcp    10.0.0.0/8            0.0.0.0/0
DROP       all    0.0.0.0/0             0.0.0.0/0
```

Part 2: Ruleset Analysis

Ruleset Analysis

- *Who can access whom over ssh?*

Ruleset Analysis

- *Who can possibly access whom over ssh?*

Ruleset Analysis

- *Who can possibly access whom over ssh?*

- Visualize as matrix or graph

Ruleset Analysis

- *Who can possibly access whom over ssh?*

- Visualize as matrix or graph

- Proven properties
  - Sound

Ruleset Analysis

- *Who can possibly access whom over ssh?*

- Visualize as matrix or graph

- Proven properties
  - Sound: If some flow is **not** in the graph, your firewall definitely blocks it

Ruleset Analysis

- *Who can possibly access whom over ssh?*

- Visualize as matrix or graph

- Proven properties
  - Sound: If some flow is **not** in the graph, your firewall definitely blocks it
  - Covers complete IPv4 address space

Ruleset Analysis

- *Who can possibly access whom over ssh?*

- Visualize as matrix or graph

- Proven properties
  - Sound: If some flow is **not** in the graph, your firewall definitely blocks it
  - Covers complete IPv4 address space
  - Minimal

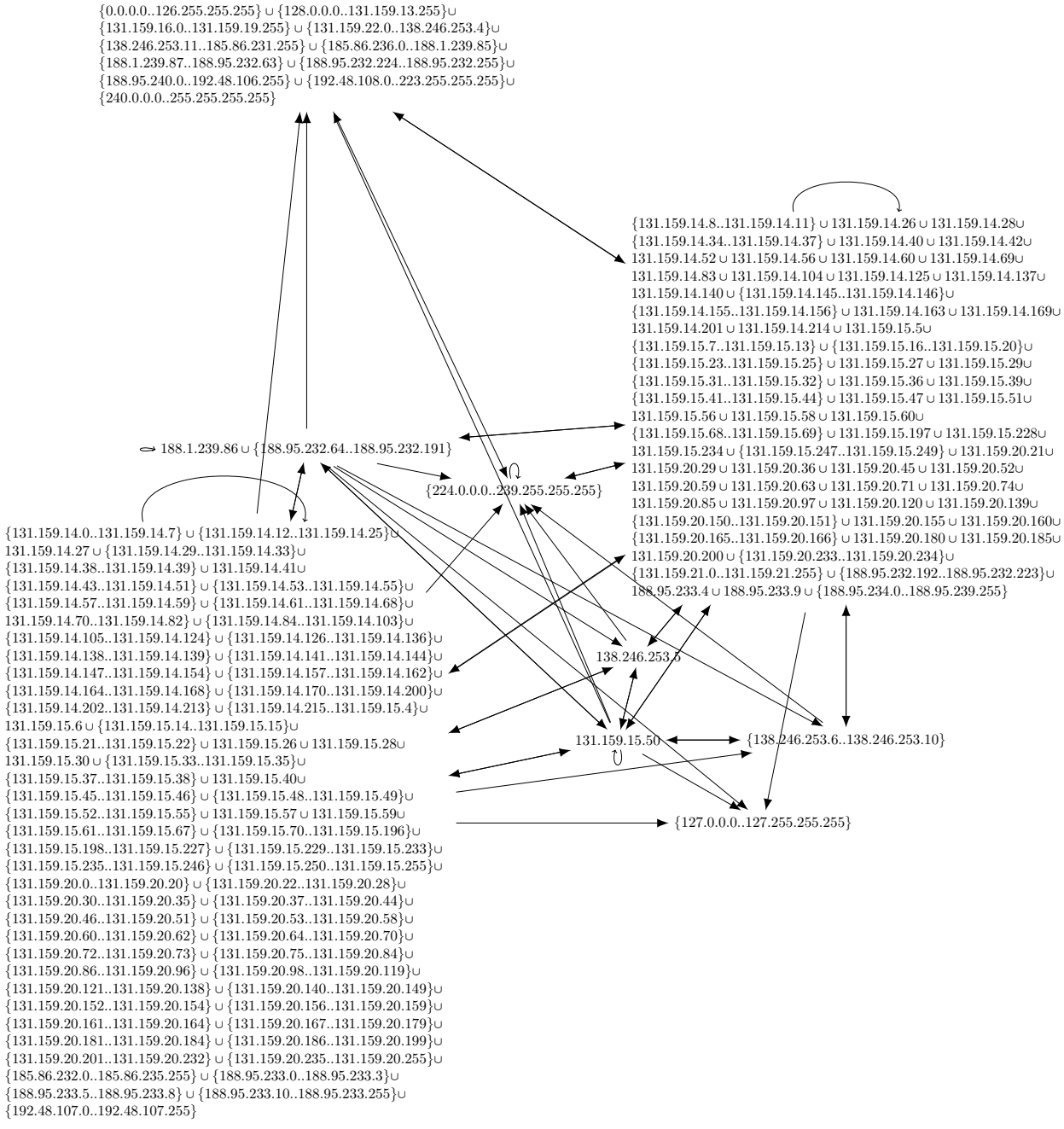
Ruleset Analysis

- *Who can possibly access whom over ssh?*

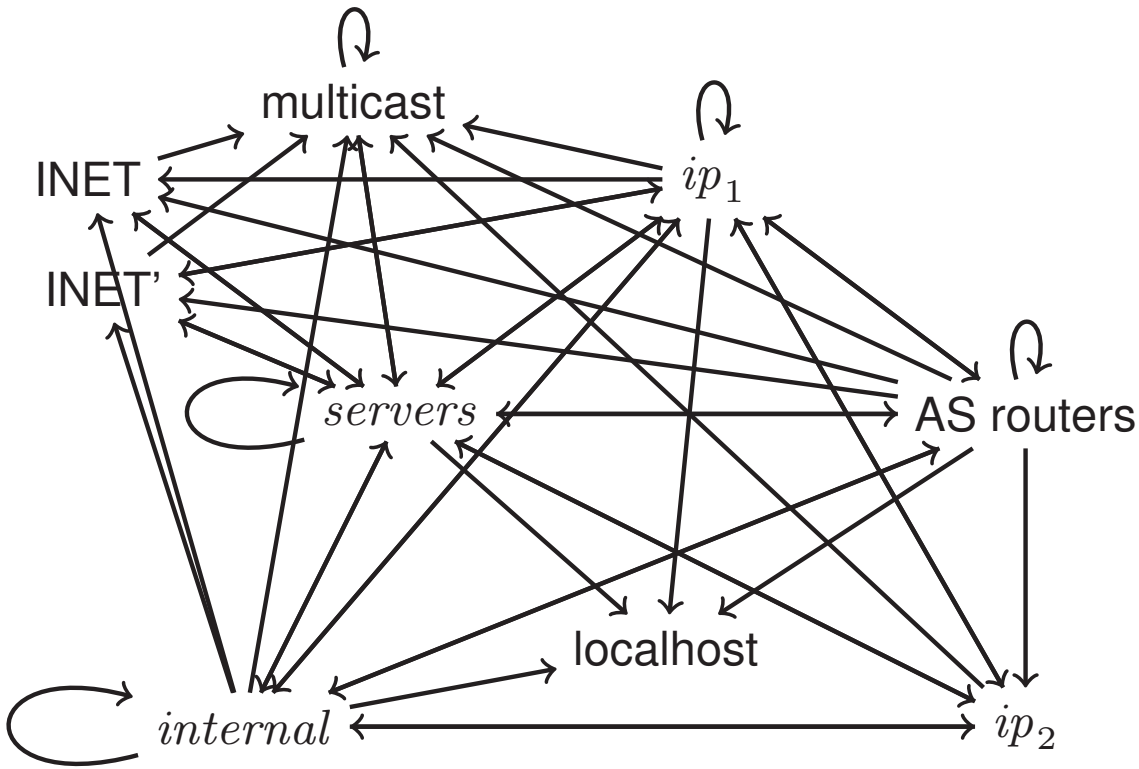
- Visualize as matrix or graph

- Proven properties
  - Sound: If some flow is **not** in the graph, your firewall definitely blocks it
  - Covers complete IPv4 address space
  - Minimal: Cannot be compressed further

Example: Firewall of our lab (2016)



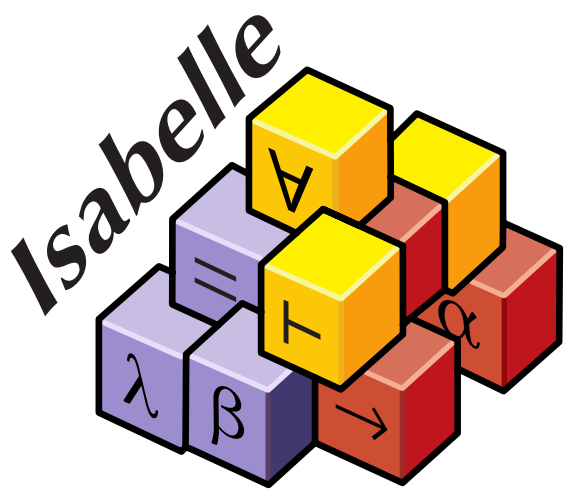
Example: Firewall of our lab (pre-2016)



Part 3: Evaluation & Related Work

Correctness

- Proven



Related Work

- *ITVal*: iptables ruleset analysis

Related Work

- *ITVal*: iptables ruleset analysis
- Not formally proven correct

Related Work

- *ITVal*: iptables ruleset analysis
- Not formally proven correct
  - *Yes, there are bugs*

Related Work

- *ITVal*: iptables ruleset analysis
- Not formally proven correct
  - *Yes, there are bugs*
  - Number of significant bits in IP addresses in CIDR notation is not a multiple of 8.
  - Example: 188.95.232.0/21

Related Work

- *ITVal*: iptables ruleset analysis
- Not formally proven correct
  - *Yes, there are bugs*
  - Number of significant bits in IP addresses in CIDR notation is not a multiple of 8.
  - Example: 188.95.232.0/21
  - Logical negations induced by `RETURN`

Related Work

- *ITVal*: iptables ruleset analysis
- Not formally proven correct
  - *Yes, there are bugs*
  - Number of significant bits in IP addresses in CIDR notation is not a multiple of 8.
  - Example: 188.95.232.0/21
  - Logical negations induced by `RETURN`
  - Unknown primitives

Related Work

- *ITVal*: iptables ruleset analysis
- Not formally proven correct
  - *Yes, there are bugs*
  - Number of significant bits in IP addresses in CIDR notation is not a multiple of 8.
  - Example: 188.95.232.0/21
  - Logical negations induced by `RETURN`
  - Unknown primitives
  - . . .

Related Work

- *ITVal*: iptables ruleset analysis
- Not formally proven correct
  - *Yes, there are bugs*
  - Number of significant bits in IP addresses in CIDR notation is not a multiple of 8.
  - Example: 188.95.232.0/21
  - Logical negations induced by `RETURN`
  - Unknown primitives
  - . . .
- Performance

Related Work

- *ITVal*: iptables ruleset analysis
- Not formally proven correct
  - *Yes, there are bugs*
  - Number of significant bits in IP addresses in CIDR notation is not a multiple of 8.
  - Example: 188.95.232.0/21
  - Logical negations induced by `RETURN`
  - Unknown primitives
  - . . .
- Performance
  - Firewall with 4946 rules (the one from before)
  - 53h and almost 100GB RAM

Related Work

- *ITVal*: iptables ruleset analysis
- Not formally proven correct
  - *Yes, there are bugs*
  - Number of significant bits in IP addresses in CIDR notation is not a multiple of 8.
  - Example: 188.95.232.0/21
  - Logical negations induced by `RETURN`
  - Unknown primitives
  - . . .
- Performance
  - Firewall with 4946 rules (the one from before)
  - 53h and almost 100GB RAM
- Disclaimer

Related Work

- *ITVal*: iptables ruleset analysis
- Not formally proven correct
  - *Yes, there are bugs*
  - Number of significant bits in IP addresses in CIDR notation is not a multiple of 8.
  - Example: 188.95.232.0/21
  - Logical negations induced by `RETURN`
  - Unknown primitives
  - . . .
- Performance
  - Firewall with 4946 rules (the one from before)
  - 53h and almost 100GB RAM
- Disclaimer
  - ITval academic **open source** prototype

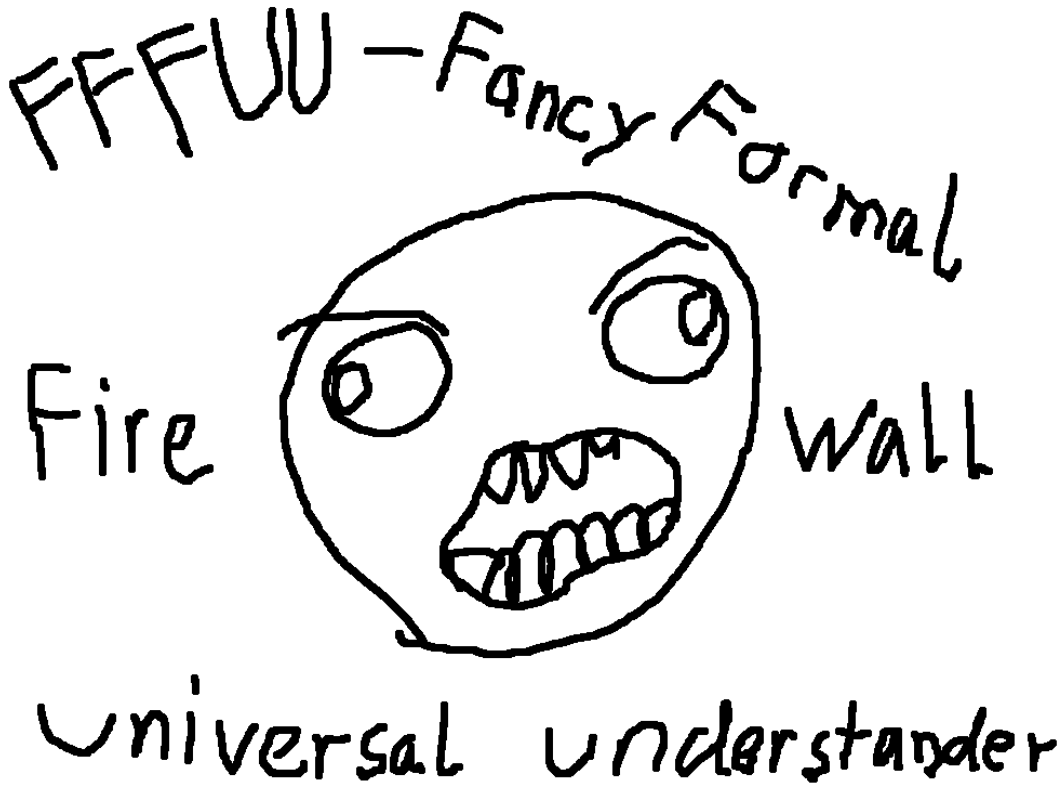
Related Work

- *ITVal*: iptables ruleset analysis
- Not formally proven correct
  - *Yes, there are bugs*
  - Number of significant bits in IP addresses in CIDR notation is not a multiple of 8.
  - Example: 188.95.232.0/21
  - Logical negations induced by `RETURN`
  - Unknown primitives
  - . . .
- Performance
  - Firewall with 4946 rules (the one from before)
  - 53h and almost 100GB RAM
- Disclaimer
  - ITval academic **open source** prototype
  - Introduces idea of IP address range partitioning

Related Work

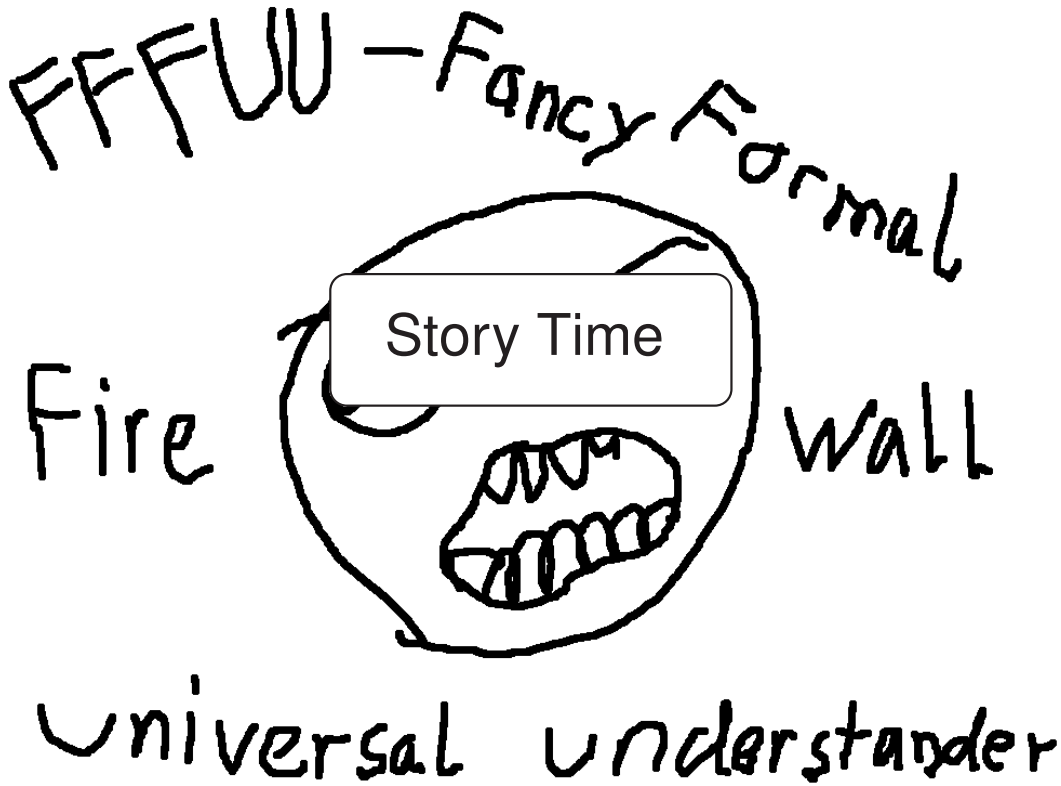
- *ITVal*: iptables ruleset analysis
- Not formally proven correct
  - *Yes, there are bugs*
  - Number of significant bits in IP addresses in CIDR notation is not a multiple of 8.
  - Example: 188.95.232.0/21
  - Logical negations induced by `RETURN`
  - Unknown primitives
  - . . .
- Performance
  - Firewall with 4946 rules (the one from before)
  - 53h and almost 100GB RAM
- Disclaimer
  - ITval academic **open source** prototype
  - Introduces idea of IP address range partitioning
  - We are standing on the shoulders of giants

FFFUU: Get It While It's Hot



`http://iptables.isabelle.systems/` & don't forget to publish your rulesets!

FFFUU: Get It While It's Hot



`http://iptables.isabelle.systems/` & don't forget to publish your rulesets!

Appendix

Fw	Rules	Chain	Simple rules	Use	Parts (ITVal)	ssh	http	Time (ITVal)
A	2784	FW (2376)	2381 (1920)	✓	246 (1)	13	9	172s (3h*)
	-	FW (2376)	2837 (581)	✗r	522 (1)	1	1	194s (9h*)
A	4113	FW (2922)	3114 (2862)	✓	334 (2)	11	11	302s (27h*)
	-	FW (2922)	3585 (517)	✗r	490 (1)	1	1	320s (8h)
A	4814	FW (4403)	3574 (3144)	✓	364 (2)	9	12	477s (46h*)
	-	FW (4403)	5123 (1601)	✗r	1574 (1)	1	1	618s (3h*)
A	4946	FW (4887)	4004 (3570)	✓	371 (2)	9	12	477s (53h*)
	-	FW (4887)	5563 (1613)	✗r	1585 (1)	1	1	820s (4h*)
B	88	FW (40)	110 (106)	✓	50 (4)	4	2	3s (2s)
	-	FW (40)	183 (75)	✓	40 (1)	1	1	2s (1s)
C	53	FW (30)	29 (12)	✓	8 (1)	1	1	1s (1s)
	-	FW (30)	27 (1)	✓	1 (1)	1	1	1s (1s)
	-	IN (49)	74 (46)	✓	38 (1)	1	1	1s (1s)
	-	IN (49)	75 (21)	✓	6 (1)	1	1	1s (1s)
D	373	FW (2649)	3482 (166)	✓	43 (1)	1	1	22s (3s)
	-	FW (2649)	16592 (1918)	✗	67 (1)	1	1	49s (33min*)
E	31	IN (24)	57 (27)	✓	4 (3)	1	2	10s (1s)
	-	IN (24)	61 (45)	✗r	3 (1)	1	1	1s (1s)
F	263	IN (261)	263 (263)	✓	250 (3)	3	3	80s (2min)
	-	IN (261)	265 (264)	✓	250 (3)	3	3	57s (3min)
G	68	IN (28)	20 (20)	✓	8 (5)	1	2	8s (1s)
	-	IN (28)	19 (19)	✗	8 (2)	2	2	1s (1s)
H	19	FW (20)	10 (10)	✗	9 (1)	1	1	8s (1s)
	-	FW (20)	8 (8)	✗r	3 (1)	1	1	1s (1s)

Fw	Rules	Chain	Simple rules	Use	Parts (ITVal)	ssh	http	Time (ITVal)
I	15	FW (5)	4 (4)	✓	4 (4)	4	4	8s (1s)
	-	FW (5)	4 (4)	✓	4 (4)	4	4	1s (1s)
J	48	FW (12)	5 (5)	✓	3 (2)	2	2	6s (1s)
	-	FW (12)	8 (2)	✓	1 (1)	1	1	1s (1s)
K	21	FW (9)	7 (6)	✓	3 (1)	1	1	12s (1s)
	-	FW (9)	4 (3)	✓	2 (1)	1	1	1s (1s)
L	27	IN (16)	19 (19)	✓	17 (3)	2	2	1s (1s)
	-	IN (16)	18 (18)	✓	17 (3)	2	2	1s (1s)
M	80	IN (92)	64 (16)	✓	2 (2)	1	2	6s (1s)
	-	IN (92)	58 (27)	✗	11 (1)	1	1	1s (1s)
N	34	FW (14)	12 (12)	✓	10 (6)	6	6	2s (2s)
	-	FW (14)	12 (12)	✓	10 (6)	6	6	1s (2s)
O	8	IN (7)	9 (9)	✓	3 (3)	1	2	1s (1s)
	-	IN (7)	8 (8)	✓	3 (3)	1	2	1s (1s)
P	595	IN (15)	8 (8)	✓	3 (2)	2	2	?s (1s)
	-	IN (15)	9 (9)	✓	3 (2)	2	2	?s (1s)
	595	IN (66)	64 (64)	✓	60 (5)	5	4	?s (22s)
	-	IN (66)	63 63)	✓	60 (5)	5	4	?s (22s)
Q	58	IN (59)	65 (65)	✓	21 (1)	1	1	?s (2s)
	-	IN (59)	62 (62)	✓	21 (2)	2	1	?s (2s)