

Verified iptables Ruleset Verification

Cornelius Diekmann diekmann@net.in.tum.de

Packet Filtering In Simple Terms

Matching a packet

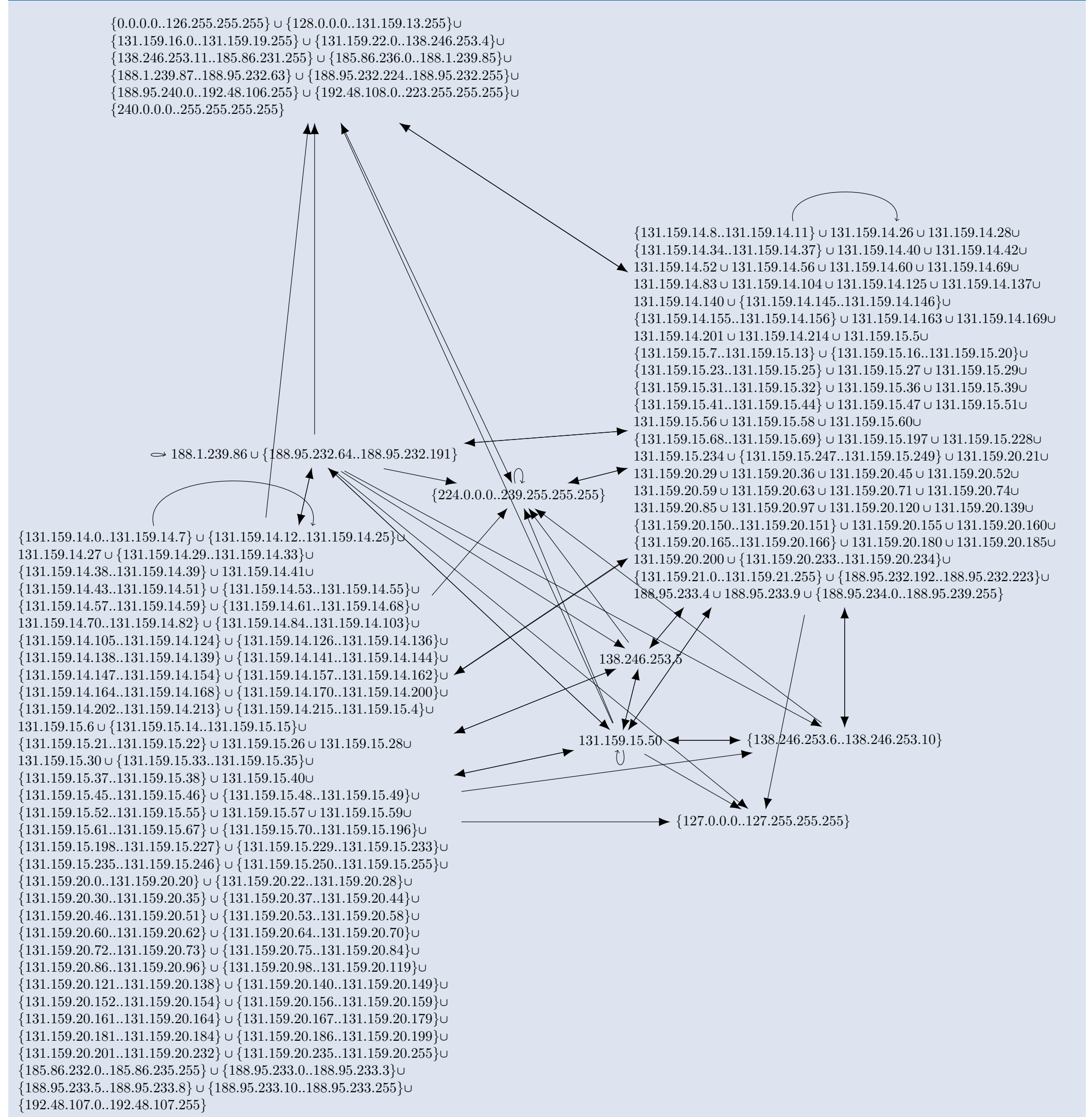
$\text{match } \gamma \text{ (Match } a) p \iff \gamma a p$
 $\text{match } _ \text{ MatchAny } _ \iff \text{True}$
 $\text{match } \gamma \text{ (MatchNot } m) p \iff \neg \text{match } \gamma m p$
 $\text{match } \gamma \text{ (MatchAnd } m_1 m_2) p \iff \text{match } \gamma m_1 p \wedge \text{match } \gamma m_2 p$

Processing a ruleset

Skip $\frac{}{\gamma, p \vdash \langle [], t \rangle \Rightarrow t}$ **Accept** $\frac{\text{match } \gamma m p}{\gamma, p \vdash \langle [(m, \text{Accept})], \emptyset \rangle \Rightarrow \checkmark}$
Drop $\frac{\text{match } \gamma m p}{\gamma, p \vdash \langle [(m, \text{Drop})], \emptyset \rangle \Rightarrow \times}$ **Reject** $\frac{\text{match } \gamma m p}{\gamma, p \vdash \langle [(m, \text{Reject})], \emptyset \rangle \Rightarrow \times}$
NoMatch $\frac{\neg \text{match } \gamma m p}{\gamma, p \vdash \langle [(m, a)], \emptyset \rangle \Rightarrow \emptyset}$ **Decision** $\frac{t \neq \emptyset}{\gamma, p \vdash \langle rs, t \rangle \Rightarrow t}$
Seq $\frac{\gamma, p \vdash \langle rs_1, \emptyset \rangle \Rightarrow t \quad \gamma, p \vdash \langle rs_2, t \rangle \Rightarrow t'}{\gamma, p \vdash \langle rs_1 :: rs_2, \emptyset \rangle \Rightarrow t'}$
CallResult $\frac{\text{match } \gamma m p \quad \gamma, p \vdash \langle \Gamma c, \emptyset \rangle \Rightarrow t}{\gamma, p \vdash \langle [(m, \text{Call } c)], \emptyset \rangle \Rightarrow t}$
CallReturn $\frac{\text{match } \gamma m p \quad \Gamma c = rs_1 :: (m', \text{Return}) :: rs_2 \quad \text{match } \gamma m' p \quad \gamma, p \vdash \langle rs_1, \emptyset \rangle \Rightarrow \emptyset}{\gamma, p \vdash \langle [(m, \text{Call } c)], \emptyset \rangle \Rightarrow \emptyset}$
Log $\frac{\text{match } \gamma m p}{\gamma, p \vdash \langle [(m, \text{Log})], \emptyset \rangle \Rightarrow \emptyset}$ **Empty** $\frac{\text{match } \gamma m p}{\gamma, p \vdash \langle [(m, \text{Empty})], \emptyset \rangle \Rightarrow \emptyset}$

- ▶ For any primitive matcher $\gamma :: (\text{primitive} \Rightarrow \text{packet} \Rightarrow \mathbb{B})$
- ▶ For any well-formed ruleset Γ
- ▶ Specification not executable but deterministic

Service Matrix: SSH Connectivity

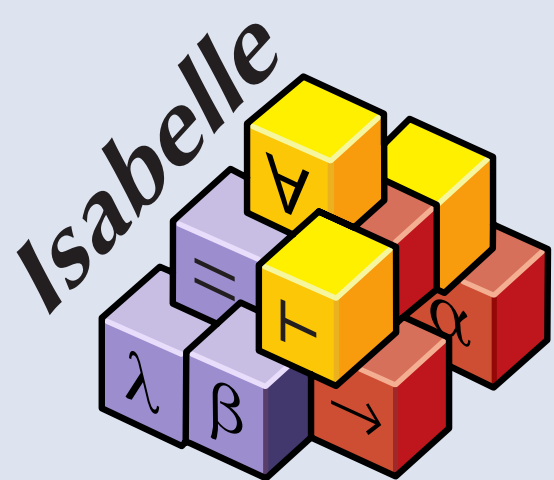


“Which machines are accessible via ssh?”

Table filter: 4911 rules, 94 user-defined chains.

Formal Verification

All the algorithms and translations are machine-verifiably proven sound with the Isabelle proof assistant.



Using Isabelle’s code generation feature, a stand-alone Haskell tool is derived from the theory.

Easy to Use

```

adm@fw# iptables-save | ./check ipassmt.txt
preprocessing ruleset
sanity checking ipassmt
checking spoofing protection:
eth1.96 True
eth1.109 False
...
[time] real 0m38.439s
    
```

Sound, Permissive Ruleset Simplification

For arbitrary iptables matching features:

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

where simple-fw 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 (*, or any numeric protocol identifier)
- ▶ src/dst interval of ports, e.g. 0:65535



Free & Open Source



<http://iptables.isabelle.systems/>