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

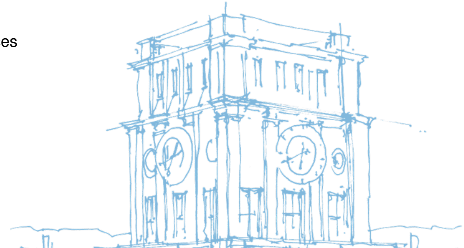
Provably Secure Networks: Methodology and Toolset for Configuration Management

Cornelius Diekmann, M. Sc.

PhD Thesis Defense

July 27, 2017

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



Motivational Quote

“*there are no good high-complexity rule sets*”

— A. Wool, 2004

Motivational Quote

“*there are no good high-complexity rule sets*”
— A. Wool, 2004

“*firewalls are (still) poorly configured*”
— A. Wool, 2010

Motivational Quote

“*there are no good high-complexity rule sets*”
— A. Wool, 2004

“*firewalls are (still) poorly configured*”
— A. Wool, 2010

. . .

Research Question

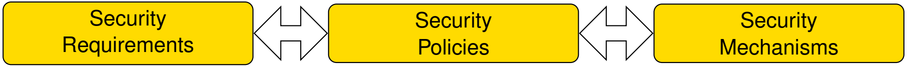
How can we help administrators to configure secure networks
and
verify the security of existing network configurations?

Agenda

1. My Contributions & My Thesis
2. Selected Topic: Case Study
3. State-of-the-Art & Related Work

Structured Approach: Security Components

Inspired by Bishop [S&P vol. 1, 2003] and taught in “Network Security” at TUM.



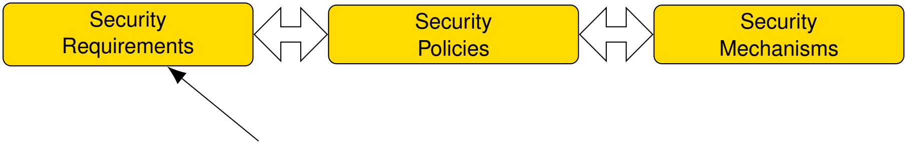
Security Requirements

Security Policies

Security Mechanisms

Structured Approach: Security Components

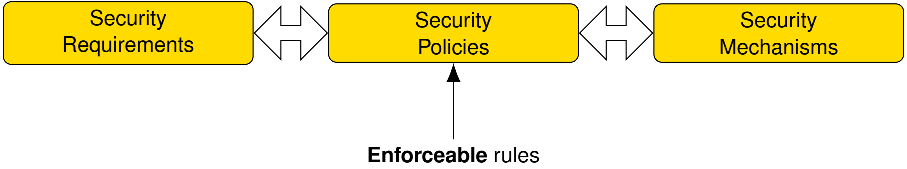
Inspired by Bishop [S&P vol. 1, 2003] and taught in “Network Security” at TUM.



Scenario-specific security goals (high level of abstraction)

Structured Approach: Security Components

Inspired by Bishop [S&P vol. 1, 2003] and taught in “Network Security” at TUM.



Security Requirements

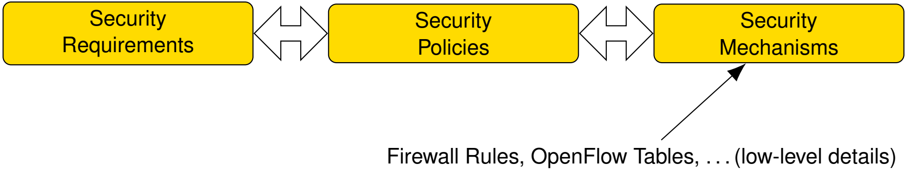
Security Policies

Security Mechanisms

Enforceable rules

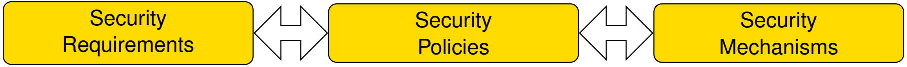
Structured Approach: Security Components

Inspired by Bishop [S&P vol. 1, 2003] and taught in "Network Security" at TUM.



Structured Approach: Security Components

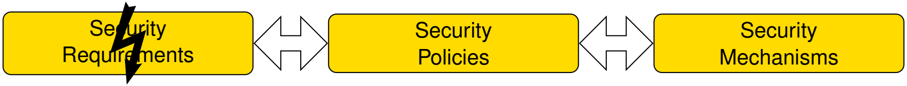
Inspired by Bishop [S&P vol. 1, 2003] and taught in “Network Security” at TUM.



Security Problems

Structured Approach: Security Components

Inspired by Bishop [S&P vol. 1, 2003] and taught in "Network Security" at TUM.

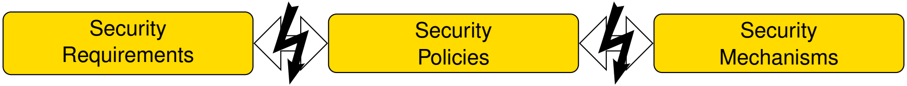


Security Problems

- Unsuitable requirements

Structured Approach: Security Components

Inspired by Bishop [S&P vol. 1, 2003] and taught in “Network Security” at TUM.

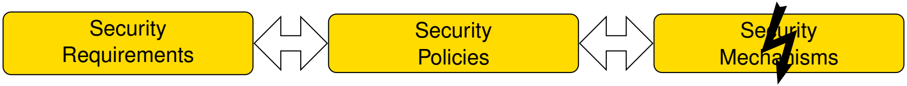


Security Problems

- Unsuitable requirements
- Translation error

Structured Approach: Security Components

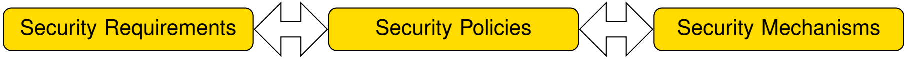
Inspired by Bishop [S&P vol. 1, 2003] and taught in “Network Security” at TUM.



Security Problems

- Unsuitable requirements
- Translation error
- Bug in the mechanism (not part of this thesis)

Contents of My Thesis

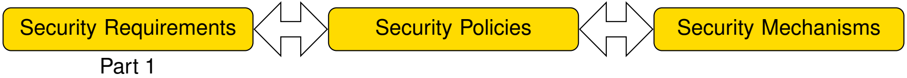


Security Requirements

Security Policies

Security Mechanisms

Contents of My Thesis

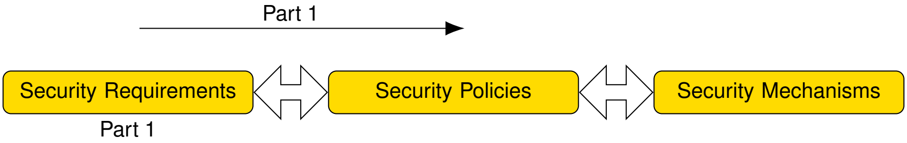


Part 1

- Specifying security requirements
  - Chapters 5, 6

- Security Invariants
  - Generic part: template
  - Generic proofs, e. g., “prohibiting more does not decrease security”
  - Template library
  - Scenario-specific part: user assigns attributes to hosts
- Policy verification

Contents of My Thesis

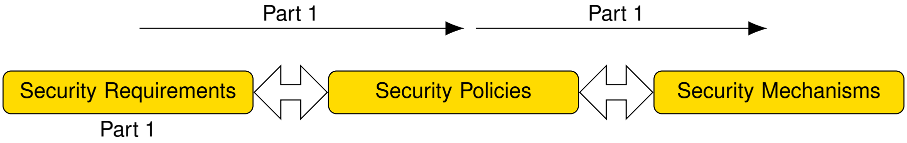


Part 1

- Specifying security requirements
  - Chapters 5, 6
- Translating to an enforceable policy
  - Chapters 7, 8, 9

- Visual feedback
- Policy uniquely defined?
- Sound & Complete
- Performance
- Connection level vs. network level and stateful flows

Contents of My Thesis

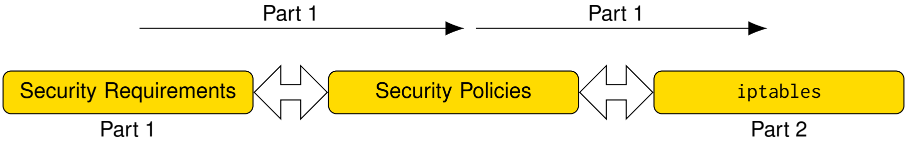


Part 1

- Specifying security requirements
  - Chapters 5, 6
- Translating to an enforceable policy
  - Chapters 7, 8, 9
- Deploying to devices
  - Chapter 10

- Discussing assumptions and implementation details
- Central `iptables` firewall, OpenVPN setup, OpenFlow

Contents of My Thesis



Part 1

Part 1

Security Requirements

Security Policies

`iptables`

Part 1

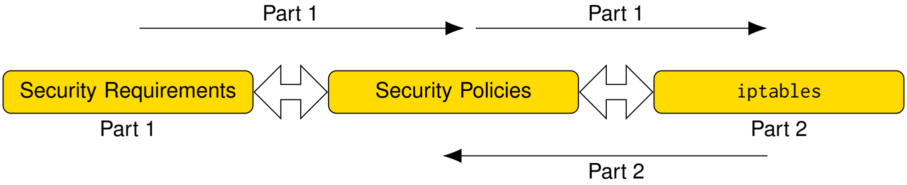
Part 2

Part 2

- Focus on `iptables`
  - Chapters 12, 13

- Formal semantics of `iptables`
  - Arbitrary match conditions
- IPv4 & IPv6
- Verify spoofing protection

Contents of My Thesis

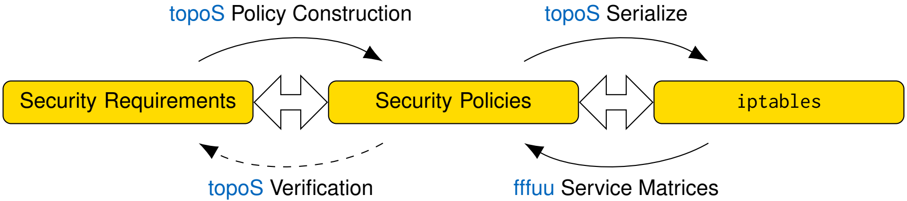


Part 2

- Focus on `iptables`
  - Chapters 12, 13
- Inferring policy from low-level rules
  - Chapter 14

- Translate to a simplified firewall model
  - Abstract over low-level details by overapproximation
- Infer high-level policy
- If a firewall (probably) accepts a connection $\longrightarrow$ then the connection is (definitely) shown in our inferred policy

Contents of My Thesis

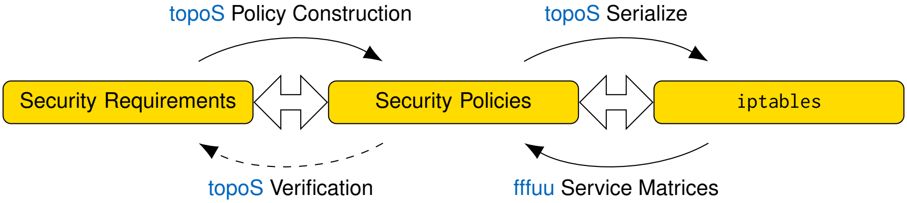


Part 3

- Demonstrate applicability
  - Chapter 16: Docker
  - Chapter 17: MeasrDroid
- Summary of scientific results, comparison to state-of-the-art
  - Chapters 18, 19, 20, 21

- Further evaluation
  - Cabin data network
  - MeasrDroid privacy audit
  - Largest collection of public `iptables` dumps
- Comparison to state-of-the-art

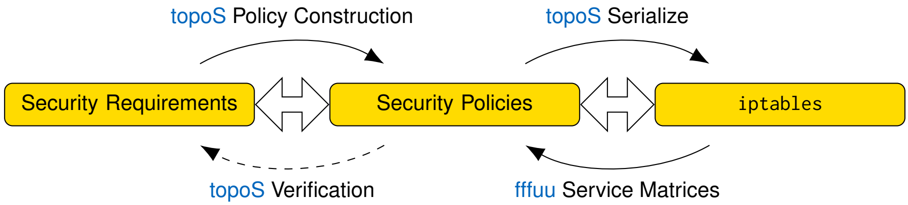
Contents of My Thesis



Part 3

- Demonstrate applicability
  - Chapter 16: Docker ⟵ next
  - Chapter 17: MeasrDroid
- Summary of scientific results, comparison to state-of-the-art
  - Chapters 18, 19, 20, 21
- Further evaluation
  - Cabin data network
  - MeasrDroid privacy audit
  - Largest collection of public `iptables` dumps
- Comparison to state-of-the-art

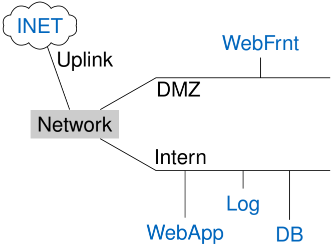
Contents of My Thesis



Part 3

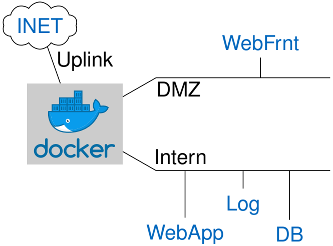
- Demonstrate applicability
  - Chapter 16: Docker ⟵ next
  - Chapter 17: MeasrDroid
- Summary of scientific results, comparison to state-of-the-art
  - Chapters 18, 19, 20, 21
- Further evaluation
  - Cabin data network
  - MeasrDroid privacy audit
  - Largest collection of public `iptables` dumps
- Comparison to state-of-the-art ⟵ This talk later

Example: Specifying Security Requirements



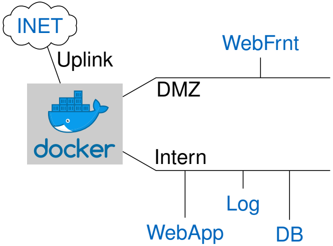
1. Logging data must not leave the Log server.
2. DB, Log and WebApp are internal hosts. WebFrnt must be accessible from outside.
3. DB, Log contain confidential information. WebApp is trusted and allowed to declassify.
4. Only WebApp may access the DB.

Example: Specifying Security Requirements



1. Logging data must not leave the Log server.
2. DB, Log and WebApp are internal hosts. WebFrnt must be accessible from outside.
3. DB, Log contain confidential information. WebApp is trusted and allowed to declassify.
4. Only WebApp may access the DB.

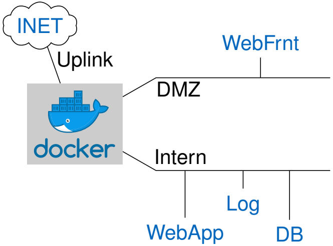
Example: Specifying Security Requirements



1. Logging data must not leave the Log server.
2. DB, Log and WebApp are internal hosts. WebFrnt must be accessible from outside.
3. DB, Log contain confidential information. WebApp is trusted and allowed to declassify.
4. Only WebApp may access the DB.

Sink $\{$Log $\mapsto$ *Sink*$\}$

Example: Specifying Security Requirements

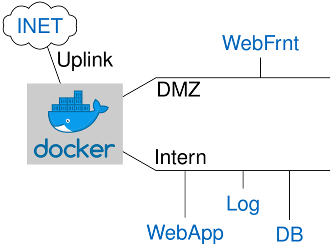


1. Logging data must not leave the Log server.
2. DB, Log and WebApp are internal hosts. WebFrnt must be accessible from outside.
3. DB, Log contain confidential information. WebApp is trusted and allowed to declassify.
4. Only WebApp may access the DB.

Sink $\{\text{Log} \mapsto \textit{Sink}\}$

SubnetsInGW $\{\text{DB} \mapsto \textit{internal}, \ \text{Log} \mapsto \textit{internal}, \ \text{WebApp} \mapsto \textit{internal}, \ \text{WebFrnt} \mapsto \textit{InboundGateway}\}$

Example: Specifying Security Requirements



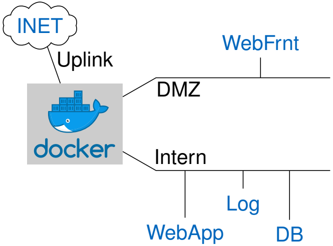
1. Logging data must not leave the Log server.
2. DB, Log and WebApp are internal hosts. WebFrnt must be accessible from outside.
3. DB, Log contain confidential information. WebApp is trusted and allowed to declassify.
4. Only WebApp may access the DB.

Sink {Log ↦ *Sink*}

SubnetsInGW {DB ↦ *internal*, Log ↦ *internal*, WebApp ↦ *internal*, WebFrnt ↦ *InboundGateway*}

Bell LaPadula {DB ↦ *confidential*, Log ↦ *confidential*, WebApp ↦ *declassify* (*trusted*)}

Example: Specifying Security Requirements



1. Logging data must not leave the Log server.
2. DB, Log and WebApp are internal hosts. WebFrnt must be accessible from outside.
3. DB, Log contain confidential information. WebApp is trusted and allowed to declassify.
4. Only WebApp may access the DB.

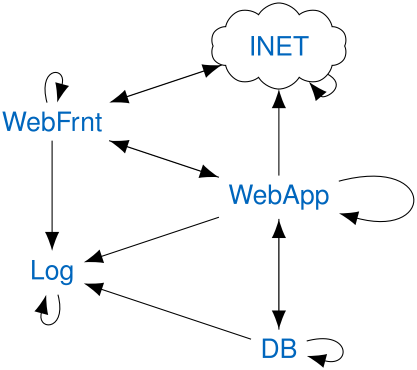
Sink {Log ↦ *Sink*}

SubnetsInGW {DB ↦ *internal*, Log ↦ *internal*, WebApp ↦ *internal*, WebFrnt ↦ *InboundGateway*}

Bell LaPadula {DB ↦ *confidential*, Log ↦ *confidential*, WebApp ↦ *declassify* (*trusted*)}

Communication Partners {DB ↦ *Access allowed by* : WebApp}

Translation to Security Policy



INET

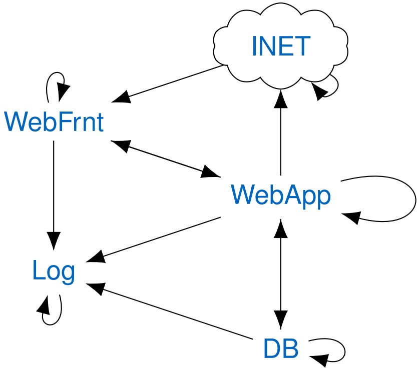
WebFrnt

WebApp

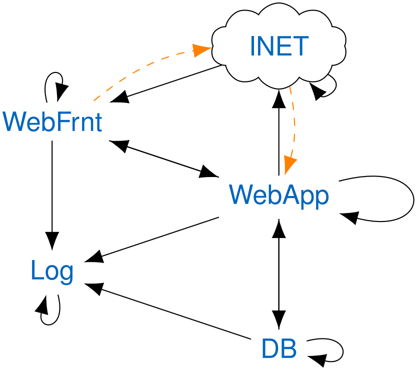
Log

DB

Translation to Security Policy



Translation to Security Policy

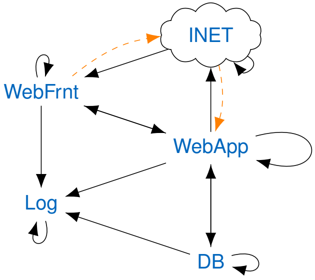


Translation to Security Mechanism (`iptables` Firewall)

```
-A FORWARD -i $WebFrnt_iface -s $WebFrnt_ipv4 -o $WebFrnt_iface -d $WebFrnt_ipv4 -j ACCEPT
-A FORWARD -i $WebFrnt_iface -s $WebFrnt_ipv4 -o $Log_iface -d $Log_ipv4 -j ACCEPT
-A FORWARD -i $WebFrnt_iface -s $WebFrnt_ipv4 -o $WebApp_iface -d $WebApp_ipv4 -j ACCEPT
-A FORWARD -i $DB_iface -s $DB_ipv4 -o $DB_iface -d $DB_ipv4 -j ACCEPT
-A FORWARD -i $DB_iface -s $DB_ipv4 -o $Log_iface -d $Log_ipv4 -j ACCEPT
-A FORWARD -i $DB_iface -s $DB_ipv4 -o $WebApp_iface -d $WebApp_ipv4 -j ACCEPT
-A FORWARD -i $Log_iface -s $Log_ipv4 -o $Log_iface -d $Log_ipv4 -j ACCEPT
-A FORWARD -i $WebApp_iface -s $WebApp_ipv4 -o $WebFrnt_iface -d $WebFrnt_ipv4 -j ACCEPT
-A FORWARD -i $WebApp_iface -s $WebApp_ipv4 -o $DB_iface -d $DB_ipv4 -j ACCEPT
-A FORWARD -i $WebApp_iface -s $WebApp_ipv4 -o $Log_iface -d $Log_ipv4 -j ACCEPT
-A FORWARD -i $WebApp_iface -s $WebApp_ipv4 -o $WebApp_iface -d $WebApp_ipv4 -j ACCEPT
-A FORWARD -i $WebApp_iface -s $WebApp_ipv4 -o $INET_iface -d $INET_ipv4 -j ACCEPT
-A FORWARD -i $INET_iface -s $INET_ipv4 -o $WebFrnt_iface -d $WebFrnt_ipv4 -j ACCEPT
-A FORWARD -i $INET_iface -s $INET_ipv4 -o $INET_iface -d $INET_ipv4 -j ACCEPT
-I FORWARD -m state --state ESTABLISHED -i $INET_iface -s $INET_ipv4 -o $WebApp_iface -d $WebApp_ipv4 -j ACCEPT
-I FORWARD -m state --state ESTABLISHED -i $WebFrnt_iface -s $WebFrnt_ipv4 -o $INET_iface -d $INET_ipv4 -j ACCEPT
-P FORWARD DROP
```

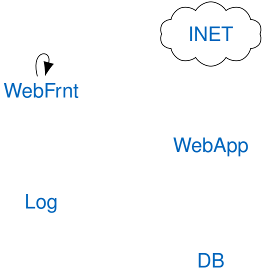
Translation to Security Mechanism (iptables Firewall)

```
-A FORWARD -i $WebFrnt_iface -s $WebFrnt_ipv4 -o $WebFrnt_iface -d $WebFrnt_ipv4 -j ACCEPT
-A FORWARD -i $WebFrnt_iface -s $WebFrnt_ipv4 -o $Log_iface -d $Log_ipv4 -j ACCEPT
-A FORWARD -i $WebFrnt_iface -s $WebFrnt_ipv4 -o $WebApp_iface -d $WebApp_ipv4 -j ACCEPT
-A FORWARD -i $DB_iface -s $DB_ipv4 -o $DB_iface -d $DB_ipv4 -j ACCEPT
-A FORWARD -i $DB_iface -s $DB_ipv4 -o $Log_iface -d $Log_ipv4 -j ACCEPT
-A FORWARD -i $DB_iface -s $DB_ipv4 -o $WebApp_iface -d $WebApp_ipv4 -j ACCEPT
-A FORWARD -i $Log_iface -s $Log_ipv4 -o $Log_iface -d $Log_ipv4 -j ACCEPT
-A FORWARD -i $WebApp_iface -s $WebApp_ipv4 -o $WebFrnt_iface -d $WebFrnt_ipv4 -j ACCEPT
-A FORWARD -i $WebApp_iface -s $WebApp_ipv4 -o $DB_iface -d $DB_ipv4 -j ACCEPT
-A FORWARD -i $WebApp_iface -s $WebApp_ipv4 -o $Log_iface -d $Log_ipv4 -j ACCEPT
-A FORWARD -i $WebApp_iface -s $WebApp_ipv4 -o $WebApp_iface -d $WebApp_ipv4 -j ACCEPT
-A FORWARD -i $WebApp_iface -s $WebApp_ipv4 -o $INET_iface -d $INET_ipv4 -j ACCEPT
-A FORWARD -i $INET_iface -s $INET_ipv4 -o $WebFrnt_iface -d $WebFrnt_ipv4 -j ACCEPT
-A FORWARD -i $INET_iface -s $INET_ipv4 -o $INET_iface -d $INET_ipv4 -j ACCEPT
-I FORWARD -m state --state ESTABLISHED -i $INET_iface -s $INET_ipv4 -o $WebApp_iface -d $WebApp_ipv4 -j ACCEPT
-I FORWARD -m state --state ESTABLISHED -i $WebFrnt_iface -s $WebFrnt_ipv4 -o $INET_iface -d $INET_ipv4 -j ACCEPT
-P FORWARD DROP
```



Translation to Security Mechanism (`iptables` Firewall)

```
-A FORWARD -i $WebFrnt_iface -s $WebFrnt_ipv4 -o $WebFrnt_iface -d $WebFrnt_ipv4 -j ACCEPT
-A FORWARD -i $WebFrnt_iface -s $WebFrnt_ipv4 -o $Log_iface -d $Log_ipv4 -j ACCEPT
-A FORWARD -i $WebFrnt_iface -s $WebFrnt_ipv4 -o $WebApp_iface -d $WebApp_ipv4 -j ACCEPT
-A FORWARD -i $DB_iface -s $DB_ipv4 -o $DB_iface -d $DB_ipv4 -j ACCEPT
-A FORWARD -i $DB_iface -s $DB_ipv4 -o $Log_iface -d $Log_ipv4 -j ACCEPT
-A FORWARD -i $DB_iface -s $DB_ipv4 -o $WebApp_iface -d $WebApp_ipv4 -j ACCEPT
-A FORWARD -i $Log_iface -s $Log_ipv4 -o $Log_iface -d $Log_ipv4 -j ACCEPT
-A FORWARD -i $WebApp_iface -s $WebApp_ipv4 -o $WebFrnt_iface -d $WebFrnt_ipv4 -j ACCEPT
-A FORWARD -i $WebApp_iface -s $WebApp_ipv4 -o $DB_iface -d $DB_ipv4 -j ACCEPT
-A FORWARD -i $WebApp_iface -s $WebApp_ipv4 -o $Log_iface -d $Log_ipv4 -j ACCEPT
-A FORWARD -i $WebApp_iface -s $WebApp_ipv4 -o $WebApp_iface -d $WebApp_ipv4 -j ACCEPT
-A FORWARD -i $WebApp_iface -s $WebApp_ipv4 -o $INET_iface -d $INET_ipv4 -j ACCEPT
-A FORWARD -i $INET_iface -s $INET_ipv4 -o $WebFrnt_iface -d $WebFrnt_ipv4 -j ACCEPT
-A FORWARD -i $INET_iface -s $INET_ipv4 -o $INET_iface -d $INET_ipv4 -j ACCEPT
-I FORWARD -m state --state ESTABLISHED -i $INET_iface -s $INET_ipv4 -o $WebApp_iface -d $WebApp_ipv4 -j ACCEPT
-I FORWARD -m state --state ESTABLISHED -i $WebFrnt_iface -s $WebFrnt_ipv4 -o $INET_iface -d $INET_ipv4 -j ACCEPT
-P FORWARD DROP
```



INET

WebFrnt

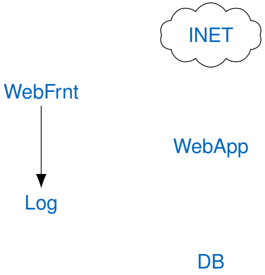
WebApp

Log

DB

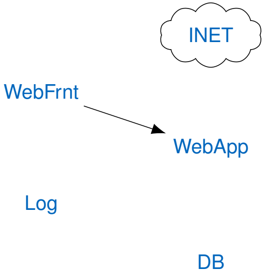
Translation to Security Mechanism (iptables Firewall)

```
-A FORWARD -i $WebFrnt_iface -s $WebFrnt_ipv4 -o $WebFrnt_iface -d $WebFrnt_ipv4 -j ACCEPT
-A FORWARD -i $WebFrnt_iface -s $WebFrnt_ipv4 -o $Log_iface -d $Log_ipv4 -j ACCEPT
-A FORWARD -i $WebFrnt_iface -s $WebFrnt_ipv4 -o $WebApp_iface -d $WebApp_ipv4 -j ACCEPT
-A FORWARD -i $DB_iface -s $DB_ipv4 -o $DB_iface -d $DB_ipv4 -j ACCEPT
-A FORWARD -i $DB_iface -s $DB_ipv4 -o $Log_iface -d $Log_ipv4 -j ACCEPT
-A FORWARD -i $DB_iface -s $DB_ipv4 -o $WebApp_iface -d $WebApp_ipv4 -j ACCEPT
-A FORWARD -i $Log_iface -s $Log_ipv4 -o $Log_iface -d $Log_ipv4 -j ACCEPT
-A FORWARD -i $WebApp_iface -s $WebApp_ipv4 -o $WebFrnt_iface -d $WebFrnt_ipv4 -j ACCEPT
-A FORWARD -i $WebApp_iface -s $WebApp_ipv4 -o $DB_iface -d $DB_ipv4 -j ACCEPT
-A FORWARD -i $WebApp_iface -s $WebApp_ipv4 -o $Log_iface -d $Log_ipv4 -j ACCEPT
-A FORWARD -i $WebApp_iface -s $WebApp_ipv4 -o $WebApp_iface -d $WebApp_ipv4 -j ACCEPT
-A FORWARD -i $WebApp_iface -s $WebApp_ipv4 -o $INET_iface -d $INET_ipv4 -j ACCEPT
-A FORWARD -i $INET_iface -s $INET_ipv4 -o $WebFrnt_iface -d $WebFrnt_ipv4 -j ACCEPT
-A FORWARD -i $INET_iface -s $INET_ipv4 -o $INET_iface -d $INET_ipv4 -j ACCEPT
-I FORWARD -m state --state ESTABLISHED -i $INET_iface -s $INET_ipv4 -o $WebApp_iface -d $WebApp_ipv4 -j ACCEPT
-I FORWARD -m state --state ESTABLISHED -i $WebFrnt_iface -s $WebFrnt_ipv4 -o $INET_iface -d $INET_ipv4 -j ACCEPT
-P FORWARD DROP
```



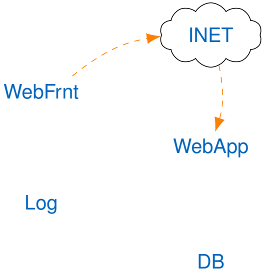
Translation to Security Mechanism (`iptables` Firewall)

```
-A FORWARD -i $WebFrnt_iface -s $WebFrnt_ipv4 -o $WebFrnt_iface -d $WebFrnt_ipv4 -j ACCEPT
-A FORWARD -i $WebFrnt_iface -s $WebFrnt_ipv4 -o $Log_iface -d $Log_ipv4 -j ACCEPT
-A FORWARD -i $WebFrnt_iface -s $WebFrnt_ipv4 -o $WebApp_iface -d $WebApp_ipv4 -j ACCEPT
-A FORWARD -i $DB_iface -s $DB_ipv4 -o $DB_iface -d $DB_ipv4 -j ACCEPT
-A FORWARD -i $DB_iface -s $DB_ipv4 -o $Log_iface -d $Log_ipv4 -j ACCEPT
-A FORWARD -i $DB_iface -s $DB_ipv4 -o $WebApp_iface -d $WebApp_ipv4 -j ACCEPT
-A FORWARD -i $Log_iface -s $Log_ipv4 -o $Log_iface -d $Log_ipv4 -j ACCEPT
-A FORWARD -i $WebApp_iface -s $WebApp_ipv4 -o $WebFrnt_iface -d $WebFrnt_ipv4 -j ACCEPT
-A FORWARD -i $WebApp_iface -s $WebApp_ipv4 -o $DB_iface -d $DB_ipv4 -j ACCEPT
-A FORWARD -i $WebApp_iface -s $WebApp_ipv4 -o $Log_iface -d $Log_ipv4 -j ACCEPT
-A FORWARD -i $WebApp_iface -s $WebApp_ipv4 -o $WebApp_iface -d $WebApp_ipv4 -j ACCEPT
-A FORWARD -i $WebApp_iface -s $WebApp_ipv4 -o $INET_iface -d $INET_ipv4 -j ACCEPT
-A FORWARD -i $INET_iface -s $INET_ipv4 -o $WebFrnt_iface -d $WebFrnt_ipv4 -j ACCEPT
-A FORWARD -i $INET_iface -s $INET_ipv4 -o $INET_iface -d $INET_ipv4 -j ACCEPT
-I FORWARD -m state --state ESTABLISHED -i $INET_iface -s $INET_ipv4 -o $WebApp_iface -d $WebApp_ipv4 -j ACCEPT
-I FORWARD -m state --state ESTABLISHED -i $WebFrnt_iface -s $WebFrnt_ipv4 -o $INET_iface -d $INET_ipv4 -j ACCEPT
-P FORWARD DROP
```



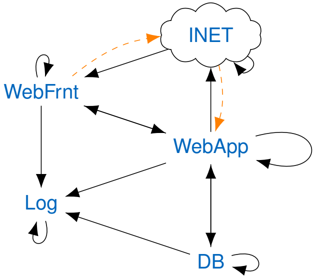
Translation to Security Mechanism (iptables Firewall)

```
-A FORWARD -i $WebFrnt_iface -s $WebFrnt_ipv4 -o $WebFrnt_iface -d $WebFrnt_ipv4 -j ACCEPT
-A FORWARD -i $WebFrnt_iface -s $WebFrnt_ipv4 -o $Log_iface -d $Log_ipv4 -j ACCEPT
-A FORWARD -i $WebFrnt_iface -s $WebFrnt_ipv4 -o $WebApp_iface -d $WebApp_ipv4 -j ACCEPT
-A FORWARD -i $DB_iface -s $DB_ipv4 -o $DB_iface -d $DB_ipv4 -j ACCEPT
-A FORWARD -i $DB_iface -s $DB_ipv4 -o $Log_iface -d $Log_ipv4 -j ACCEPT
-A FORWARD -i $DB_iface -s $DB_ipv4 -o $WebApp_iface -d $WebApp_ipv4 -j ACCEPT
-A FORWARD -i $Log_iface -s $Log_ipv4 -o $Log_iface -d $Log_ipv4 -j ACCEPT
-A FORWARD -i $WebApp_iface -s $WebApp_ipv4 -o $WebFrnt_iface -d $WebFrnt_ipv4 -j ACCEPT
-A FORWARD -i $WebApp_iface -s $WebApp_ipv4 -o $DB_iface -d $DB_ipv4 -j ACCEPT
-A FORWARD -i $WebApp_iface -s $WebApp_ipv4 -o $Log_iface -d $Log_ipv4 -j ACCEPT
-A FORWARD -i $WebApp_iface -s $WebApp_ipv4 -o $WebApp_iface -d $WebApp_ipv4 -j ACCEPT
-A FORWARD -i $WebApp_iface -s $WebApp_ipv4 -o $INET_iface -d $INET_ipv4 -j ACCEPT
-A FORWARD -i $INET_iface -s $INET_ipv4 -o $WebFrnt_iface -d $WebFrnt_ipv4 -j ACCEPT
-A FORWARD -i $INET_iface -s $INET_ipv4 -o $INET_iface -d $INET_ipv4 -j ACCEPT
-I FORWARD -m state --state ESTABLISHED -i $INET_iface -s $INET_ipv4 -o $WebApp_iface -d $WebApp_ipv4 -j ACCEPT
-I FORWARD -m state --state ESTABLISHED -i $WebFrnt_iface -s $WebFrnt_ipv4 -o $INET_iface -d $INET_ipv4 -j ACCEPT
-P FORWARD DROP
```



Translation to Security Mechanism (`iptables` Firewall)

```
-A FORWARD -i $WebFrnt_iface -s $WebFrnt_ipv4 -o $WebFrnt_iface -d $WebFrnt_ipv4 -j ACCEPT
-A FORWARD -i $WebFrnt_iface -s $WebFrnt_ipv4 -o $Log_iface -d $Log_ipv4 -j ACCEPT
-A FORWARD -i $WebFrnt_iface -s $WebFrnt_ipv4 -o $WebApp_iface -d $WebApp_ipv4 -j ACCEPT
-A FORWARD -i $DB_iface -s $DB_ipv4 -o $DB_iface -d $DB_ipv4 -j ACCEPT
-A FORWARD -i $DB_iface -s $DB_ipv4 -o $Log_iface -d $Log_ipv4 -j ACCEPT
-A FORWARD -i $DB_iface -s $DB_ipv4 -o $WebApp_iface -d $WebApp_ipv4 -j ACCEPT
-A FORWARD -i $Log_iface -s $Log_ipv4 -o $Log_iface -d $Log_ipv4 -j ACCEPT
-A FORWARD -i $WebApp_iface -s $WebApp_ipv4 -o $WebFrnt_iface -d $WebFrnt_ipv4 -j ACCEPT
-A FORWARD -i $WebApp_iface -s $WebApp_ipv4 -o $DB_iface -d $DB_ipv4 -j ACCEPT
-A FORWARD -i $WebApp_iface -s $WebApp_ipv4 -o $Log_iface -d $Log_ipv4 -j ACCEPT
-A FORWARD -i $WebApp_iface -s $WebApp_ipv4 -o $WebApp_iface -d $WebApp_ipv4 -j ACCEPT
-A FORWARD -i $WebApp_iface -s $WebApp_ipv4 -o $INET_iface -d $INET_ipv4 -j ACCEPT
-A FORWARD -i $INET_iface -s $INET_ipv4 -o $WebFrnt_iface -d $WebFrnt_ipv4 -j ACCEPT
-A FORWARD -i $INET_iface -s $INET_ipv4 -o $INET_iface -d $INET_ipv4 -j ACCEPT
-I FORWARD -m state --state ESTABLISHED -i $INET_iface -s $INET_ipv4 -o $WebApp_iface -d $WebApp_ipv4 -j ACCEPT
-I FORWARD -m state --state ESTABLISHED -i $WebFrnt_iface -s $WebFrnt_ipv4 -o $INET_iface -d $INET_ipv4 -j ACCEPT
-P FORWARD DROP
```



Translation to Security Mechanism (`iptables` Firewall)

- Copy & Paste without verification into existing Docker rules

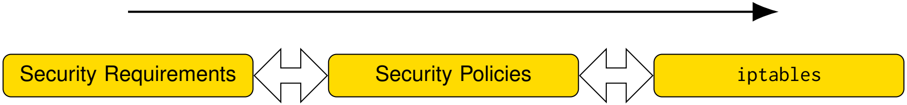
Existing, Docker-generated:

```
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:DOCKER - [0:0]
:DOCKER-ISOLATION - [0:0]
:MYNET - [0:0]
-A FORWARD -j DOCKER-ISOLATION
-A FORWARD -j MYNET
-A FORWARD -o dbr -j DOCKER
-A FORWARD -o dbr -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -i dbr ! -o dbr -j ACCEPT
-A FORWARD -o docker0 -j DOCKER
-A FORWARD -o docker0 -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -i docker0 ! -o docker0 -j ACCEPT
-A FORWARD -i docker0 -o docker0 -j ACCEPT
-A FORWARD -i dbr -o dbr -j DROP
-A DOCKER-ISOLATION -i docker0 -o dbr -j DROP
-A DOCKER-ISOLATION -i dbr -o docker0 -j DROP
-A DOCKER-ISOLATION -j RETURN
```

New, topoS-generated:

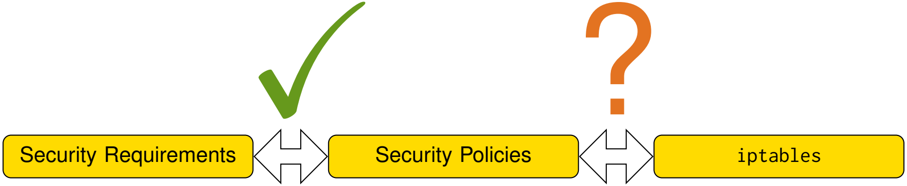
```
-A MYNET -m state --state ESTABLISHED                                          ↩
         ! -i dbr -o dbr -d 10.0.0.4 -j ACCEPT
-A MYNET -m state --state ESTABLISHED                                          ↩
         -i dbr -s 10.0.0.1 ! -o dbr -j ACCEPT
-A MYNET -i dbr -s 10.0.0.1 -o dbr -d 10.0.0.1 -j ACCEPT
-A MYNET -i dbr -s 10.0.0.1 -o dbr -d 10.0.0.2 -j ACCEPT
-A MYNET -i dbr -s 10.0.0.1 -o dbr -d 10.0.0.4 -j ACCEPT
-A MYNET -i dbr -s 10.0.0.3 -o dbr -d 10.0.0.3 -j ACCEPT
-A MYNET -i dbr -s 10.0.0.3 -o dbr -d 10.0.0.2 -j ACCEPT
-A MYNET -i dbr -s 10.0.0.3 -o dbr -d 10.0.0.4 -j ACCEPT
-A MYNET -i dbr -s 10.0.0.2 -o dbr -d 10.0.0.2 -j ACCEPT
-A MYNET -i dbr -s 10.0.0.4 -o dbr -d 10.0.0.1 -j ACCEPT
-A MYNET -i dbr -s 10.0.0.4 -o dbr -d 10.0.0.3 -j ACCEPT
-A MYNET -i dbr -s 10.0.0.4 -o dbr -d 10.0.0.2 -j ACCEPT
-A MYNET -i dbr -s 10.0.0.4 -o dbr -d 10.0.0.4 -j ACCEPT
-A MYNET -i dbr -s 10.0.0.4 ! -o dbr -j ACCEPT
-A MYNET ! -i dbr -o dbr -d 10.0.0.1 -j ACCEPT
-A MYNET -i dbr -j DROP
COMMIT
```

So Far



- So far
  - Serializing new configurations

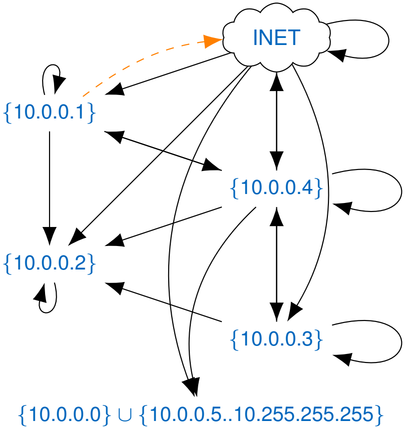
So Far



- So far
  - Serializing new configurations
- Missing
  - Verify `iptables` filtering rules
  - Understanding arbitrary `iptables` filtering rules
  - `man iptables-extensions` over 200 match options

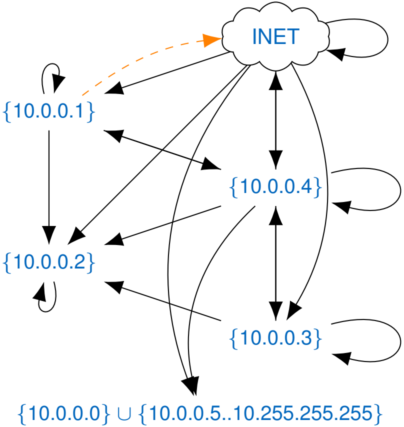
Understanding `iptables` with fffuu

- Input: `iptables-save`
- Output:

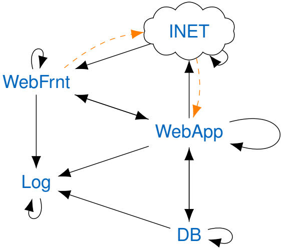


Understanding `iptables` with fffuu

- Input: `iptables-save`
- Output:

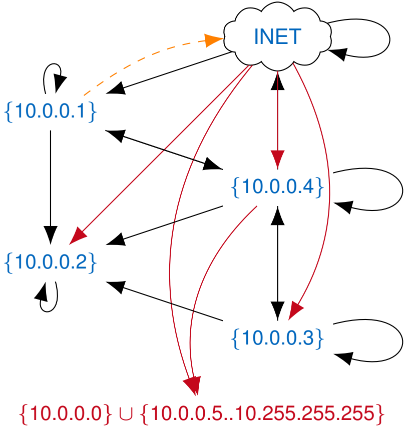


- Recall the policy:

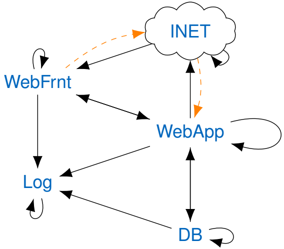


Understanding `iptables` with fffuu

- Input: `iptables-save`
- Output:

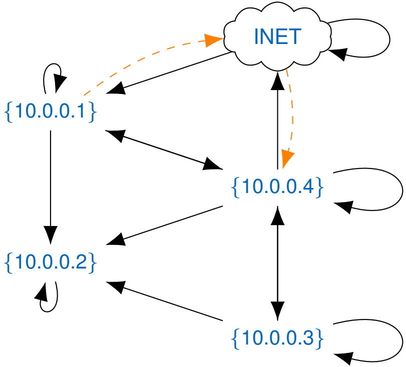


- Recall the policy:

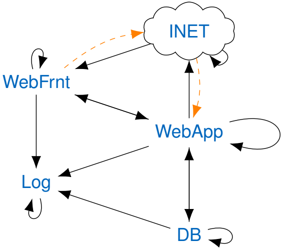


Understanding `iptables` with fffuu

- Input: `iptables-save`
- Output:



- Recall the policy:

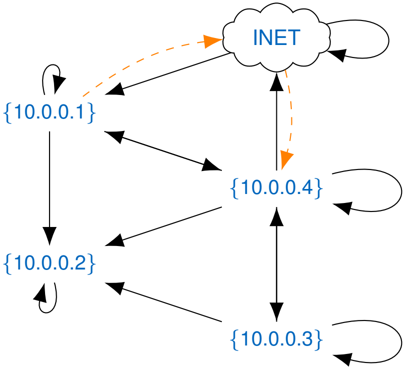


- Changing `iptables` rules

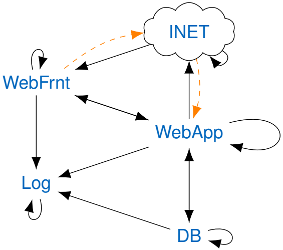
```
--A MYNET -i dbr -s 10.0.0.4 ! -o dbr -j ACCEPT
--A MYNET ! -i dbr -o dbr -d 10.0.0.1 -j ACCEPT
+-A MYNET -i dbr -s 10.0.0.4 ! -o dbr ! -d 10.0.0.0/8 -j ACCEPT
+-A MYNET ! -i dbr ! -s 10.0.0.0/8 -o dbr -d 10.0.0.1 -j ACCEPT
 -A MYNET -i dbr -j DROP
+-A MYNET -o dbr -j DROP
+-A MYNET -s 10.0.0.0/8 -j DROP
+-A MYNET -d 10.0.0.0/8 -j DROP
```

Understanding `iptables` with fffuu

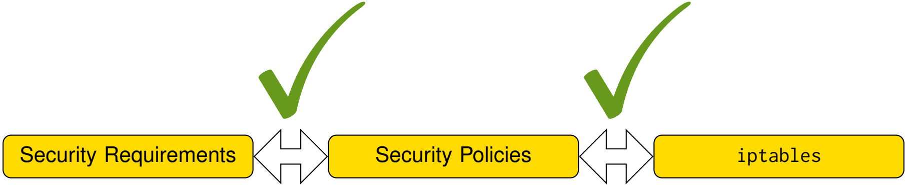
- Input: `iptables-save`
- Output:



- Recall the policy:



Ultimately



Security Requirements

Security Policies

`iptables`

Related Work

	Formal Semantics	Formal Verification	High-Level Language	Built-In Verification	Connection Semantics	Legacy Support	Low Level Access

Related Work

	Formal Semantics	Formal Verification	High-Level Language	Built-In Verification	Connection Semantics	Legacy Support	Low Level Access

Formal Semantics

- Management language
- Target security mechanism
- Type checked by theorem prover

Related Work

	Formal Semantics	Formal Verification	High-Level Language	Built-In Verification	Connection Semantics	Legacy Support	Low Level Access

Formal Verification

- Compilation/translation verified: Language → Mechanism
- Proof machine-checked

Related Work

	Formal Semantics	Formal Verification	High-Level Language	Built-In Verification	Connection Semantics	Legacy Support	Low Level Access

High-Level Language

- Security requirements, not policy

Related Work

	Formal Semantics	Formal Verification	High-Level Language	Built-In Verification	Connection Semantics	Legacy Support	Low Level Access

Built-In Verification

- Of the **specification**
- Feedback
- Automated

Related Work

	Formal Semantics	Formal Verification	High-Level Language	Built-In Verification	Connection Semantics	Legacy Support	Low Level Access

Stateful Connection Semantics

- `conntrack`
- Connection level vs. network level
- “I can connect to the Internet, but not the other way round”

Related Work

	Formal Semantics	Formal Verification	High-Level Language	Built-In Verification	Connection Semantics	Legacy Support	Low Level Access

Legacy Support

- Read `iptables`, Cisco, . . . and make available to high-level abstraction

Related Work

	Formal Semantics	Formal Verification	High-Level Language	Built-In Verification	Connection Semantics	Legacy Support	Low Level Access

Low Level Access

- Administrator may tune low-level configuration
- Soundness check

Related Work

	Formal Semantics	Formal Verification	High-Level Language	Built-In Verification	Connection Semantics	Legacy Support	Low Level Access

Related Work

	Formal Semantics	Formal Verification	High-Level Language	Built-In Verification	Connection Semantics	Legacy Support	Low Level Access
NetCore	✓	✓	✗	✓	✗	✗	✗
NetKAT family	✓	✓	✗	✓	✗	✗	✗

Related Work

	Formal Semantics	Formal Verification	High-Level Language	Built-In Verification	Connection Semantics	Legacy Support	Low Level Access
NetCore	✓	✓	×	✓	×	×	×
NetKAT family	✓	✓	×	✓	×	×	×
VALID	✓	×	✓	×	×	×	×
Zhao et al.	×	×	✓	×	×	×	×

Related Work

	Formal Semantics	Formal Verification	High-Level Language	Built-In Verification	Connection Semantics	Legacy Support	Low Level Access
NetCore	✓	✓	×	✓	×	×	×
NetKAT family	✓	✓	×	✓	×	×	×
VALID	✓	×	✓	×	×	×	×
Zhao et al.	×	×	✓	×	×	×	×
Flowlog & co	×	×	×	✓	✓	✓	×

Related Work

	Formal Semantics	Formal Verification	High-Level Language	Built-In Verification	Connection Semantics	Legacy Support	Low Level Access
NetCore	✓	✓	✗	✓	✗	✗	✗
NetKAT family	✓	✓	✗	✓	✗	✗	✗
VALID	✓	✗	✓	✗	✗	✗	✗
Zhao et al.	✗	✗	✓	✗	✗	✗	✗
Flowlog & co	✗	✗	✗	✓	✓	✓	✗
Mignis	✓	✓	✗	✗	✓	✗	✓

Related Work

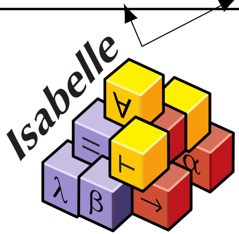
	Formal Semantics	Formal Verification	High-Level Language	Built-In Verification	Connection Semantics	Legacy Support	Low Level Access
NetCore	✓	✓	×	✓	×	×	×
NetKAT family	✓	✓	×	✓	×	×	×
VALID	✓	×	✓	×	×	×	×
Zhao et al.	×	×	✓	×	×	×	×
Flowlog & co	×	×	×	✓	✓	✓	×
Mignis	✓	✓	×	×	✓	×	✓

Related Work

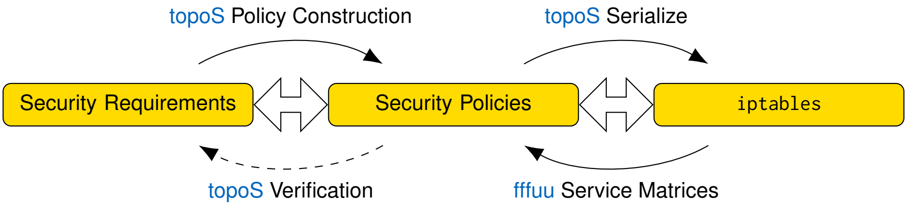
	Formal Semantics	Formal Verification	High-Level Language	Built-In Verification	Connection Semantics	Legacy Support	Low Level Access
NetCore	✓	✓	×	✓	×	×	×
NetKAT family	✓	✓	×	✓	×	×	×
VALID	✓	×	✓	×	×	×	×
Zhao et al.	×	×	✓	×	×	×	×
Flowlog & co	×	×	×	✓	✓	✓	×
Mignis	✓	✓	×	×	✓	×	✓
topoS + fffuu	✓	✓	✓	✓	✓	✓	✓

Related Work

	Formal Semantics	Formal Verification	High-Level Language	Built-In Verification	Connection Semantics	Legacy Support	Low Level Access
NetCore	✓	✓	✗	✓	✗	✗	✗
NetKAT family	✓	✓	✗	✓	✗	✗	✗
VALID	✓	✗	✓	✗	✗	✗	✗
Zhao et al.	✗	✗	✓	✗	✗	✗	✗
Flowlog & co	✗	✗	✗	✓	✓	✓	✗
Mignis	✓	✓	✗	✗	✓	✗	✓
topoS + fffuu	✓	✓	✓	✓	✓	✓	✓



Conclusion



- First, fully machine-verified tools for bridging the above gaps in both directions
- Evaluated on
    - Cabin data network
    - Android Measurement System
    - Largest collection of public `iptables` dumps
- Published in
    - AFP (x6), FM, FORTE (x2), IFIP NETWORKING, CNSM, CNSM Workshop, ESSS

Backup

Backup Slides

Publications (AFP)

- J. Michaelis and C. Diekmann. LOFT – Verified Migration of Linux Firewalls to SDN. *Archive of Formal Proofs*, Oct. 2016. Formal proof development
- C. Diekmann and L. Hupel. Iptables Semantics. *Archive of Formal Proofs*, Sept. 2016. Formal proof development
- J. Michaelis and C. Diekmann. Routing. *Archive of Formal Proofs*, Aug. 2016. Formal proof development
- C. Diekmann, J. Michaelis, and M. Haslbeck. Simple Firewall. *Archive of Formal Proofs*, Aug. 2016. Formal proof development
- C. Diekmann, J. Michaelis, and L. Hupel. IP Addresses. *Archive of Formal Proofs*, June 2016. Formal proof development
- C. Diekmann. Network Security Policy Verification. *Archive of Formal Proofs*, July 2016. Formal proof development

Publications

- M. von Maltitz, C. Diekmann, and G. Carle. Privacy Assessment using Static Taint Analysis (Tool Paper). In *Formal Techniques for Distributed Objects, Components, and Systems: 37th IFIP WG 6.1 International Conference (FORTE)*, Neuchâtel, Switzerland, June 2017
- M. von Maltitz, C. Diekmann, and G. Carle. Taint Analysis for System-Wide Privacy Audits: A Framework and Real-World Case Studies. 1st Workshop for Formal Methods on Privacy, Nov. 2016. workshop without proceedings
- C. Diekmann, A. Korsten, and G. Carle. Demonstrating topoS: Theorem-prover-based synthesis of secure network configurations. In *11th International Conference on Network and Service Management (CNSM)*, pages 366–371, Barcelona, Spain, Nov. 2015
- C. Diekmann, S.-A. Posselt, H. Niedermayer, H. Kinkelin, O. Hanka, and G. Carle. Verifying Security Policies using Host Attributes. In *Formal Techniques for Distributed Objects, Components, and Systems: 34th IFIP WG 6.1 International Conference (FORTE)*, pages 133–148, Berlin, Germany, June 2014. Springer Berlin Heidelberg
- C. Diekmann, L. Hupel, and G. Carle. Directed Security Policies: A Stateful Network Implementation. In *Engineering Safety and Security Systems (ESSS)*, volume 150 of *Electronic Proceedings in Theoretical Computer Science*, pages 20–34, Singapore, May 2014. Open Publishing Association
- C. Diekmann, J. Michaelis, M. Haslbeck, and G. Carle. Verified iptables Firewall Analysis. In *IFIP Networking 2016*, Vienna, Austria, May 2016
- C. Diekmann, L. Schwaighofer, and G. Carle. Certifying Spoofing-Protection of Firewalls. In *11th International Conference on Network and Service Management (CNSM)*, pages 168–172, Barcelona, Spain, Nov. 2015
- C. Diekmann, L. Hupel, and G. Carle. Semantics-Preserving Simplification of Real-World Firewall Rule Sets. In *Formal Methods*, June 2015

Central Theorem of fffuu

Assumes

- Unfolded rs for Γ
- p is `NEW`
- $\Gamma, \gamma, p \vdash \langle rs, \textcircled{?} \rangle \Rightarrow \textcircled{\checkmark}$
- Let $(V, E) = \text{matrix}\ (\text{iifce}\ p, \text{oifce}\ p, \text{prot}\ p, \text{sport}\ p, \text{dport}\ p)\ (\text{simplify}\ rs)$

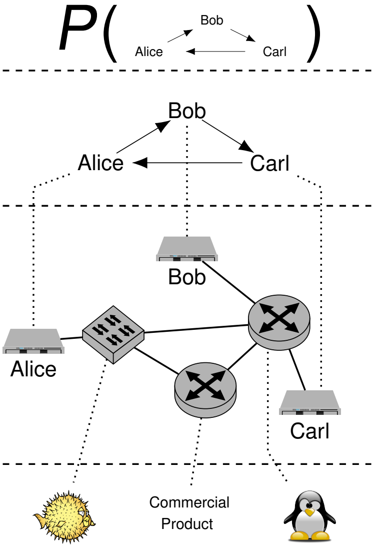
Shows

$$\begin{aligned}\exists s_{\text{repr}}\ d_{\text{repr}}\ s_{\text{range}}\ d_{\text{range}}.\ & (s_{\text{repr}}, d_{\text{repr}}) \in \text{set}\ E\ \land \\ & (\text{map\_of}\ V)\ s_{\text{repr}} = \text{Some}\ s_{\text{range}}\ \land\ (\text{src}\ p) \in s_{\text{range}}\ \land \\ & (\text{map\_of}\ V)\ d_{\text{repr}} = \text{Some}\ d_{\text{range}}\ \land\ (\text{dst}\ p) \in d_{\text{range}}\end{aligned}$$

Reads: If the firewall accepts a packet, we can look up source and destination IP in the graph.

- Unfolding may fail (it is successful if the kernel accepts it and it has no ‘strange’ actions)
- We can ignore interfaces if we have spoofing protection

Abstraction Layers



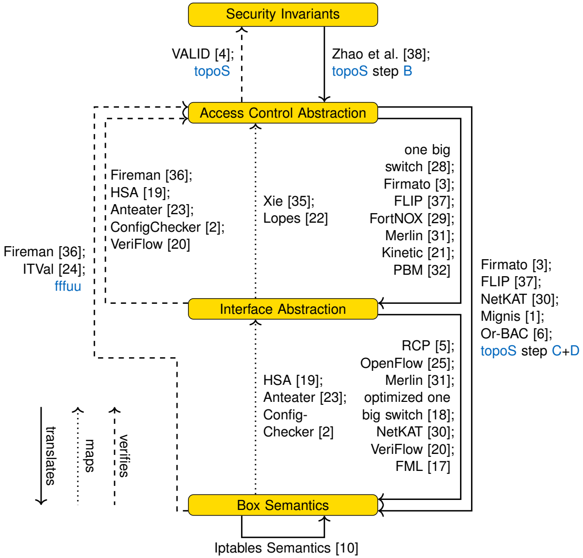
Security Invariants

Access Control Abstraction

Interface Abstraction

Box Semantics

Related Work



Security Invariants

VALID [4]; topoS

Zhao et al. [38]; topoS step B

Access Control Abstraction

Fireman [36]; HSA [19]; Anteater [23]; ConfigChecker [2]; VeriFlow [20]

Xie [35]; Lopes [22]

one big switch [28]; Firmato [3]; FLIP [37]; FortNOX [29]; Merlin [31]; Kinetic [21]; PBM [32]

Fireman [36]; ITVal [24]; fffuu

Firmato [3]; FLIP [37]; NetKAT [30]; Mignis [1]; Or-BAC [6]; topoS step C+D

Interface Abstraction

HSA [19]; Anteater [23]; Config-Checker [2]

RCP [5]; OpenFlow [25]; Merlin [31]; optimized one big switch [18]; NetKAT [30]; VeriFlow [20]; FML [17]

translates

maps

verifies

Box Semantics

Iptables Semantics [10]

Bibliography

[1] P. Adão, C. Bozzato, G. Dei Rossi, R. Focardi, and F. L. Luccio.
Mignis: A Semantic Based Tool for Firewall Configuration.
In *27th Computer Security Foundations Symposium*, CSF, pages 351–365. IEEE, July 2014.

[2] E. Al-Shaer, W. Marrero, A. El-Atawy, and K. Elbadawi.
Network Configuration in A Box: Towards End-to-End Verification of Network Reachability and Security.
In *International Conference on Network Protocols (ICNP)*, pages 123–132. IEEE, Oct. 2009.

[3] Y. Bartal, A. Mayer, K. Nissim, and A. Wool.
Firmato: A Novel Firewall Management Toolkit.
In *Symposium on Security and Privacy*, pages 17–31. IEEE, May 1999.

[4] S. Bleikertz and T. Groß.
A Virtualization Assurance Language for Isolation and Deployment.
In *International Symposium on Policies for Distributed Systems and Networks (POLICY)*, pages 33–40. IEEE, June 2011.

[5] M. Caesar, D. Caldwell, N. Feamster, J. Rexford, A. Shaikh, and J. van der Merwe.
Design and Implementation of a Routing Control Platform.
In *2nd USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, NSDI'05, pages 15–28, Boston, MA, May 2005. USENIX Association.

[6] F. Cuppens, N. Cuppens-Boulahia, T. Sans, and A. Miège.
A Formal Approach to Specify and Deploy a Network Security Policy.
In *Formal Aspects of Security and Trust (FAST)*, pages 203–218. Springer US, Aug. 2004.

[7] C. Diekmann.
Network Security Policy Verification.
Archive of Formal Proofs, July 2016.
Formal proof development.

Bibliography

[8] C. Diekmann and L. Hupel.
Iptables Semantics.
Archive of Formal Proofs, Sept. 2016.
Formal proof development.

[9] C. Diekmann, L. Hupel, and G. Carle.
Directed Security Policies: A Stateful Network Implementation.
In *Engineering Safety and Security Systems (ESSS)*, volume 150 of *Electronic Proceedings in Theoretical Computer Science*, pages 20–34, Singapore, May 2014. Open Publishing Association.

[10] C. Diekmann, L. Hupel, and G. Carle.
Semantics-Preserving Simplification of Real-World Firewall Rule Sets.
In *Formal Methods*, June 2015.

[11] C. Diekmann, A. Korsten, and G. Carle.
Demonstrating topoS: Theorem-prover-based synthesis of secure network configurations.
In *11th International Conference on Network and Service Management (CNSM)*, pages 366–371, Barcelona, Spain, Nov. 2015.

[12] C. Diekmann, J. Michaelis, and M. Haslbeck.
Simple Firewall.
Archive of Formal Proofs, Aug. 2016.
Formal proof development.

[13] C. Diekmann, J. Michaelis, M. Haslbeck, and G. Carle.
Verified iptables Firewall Analysis.
In *IFIP Networking 2016*, Vienna, Austria, May 2016.

Bibliography

[14] C. Diekmann, J. Michaelis, and L. Hupel.
IP Addresses.
Archive of Formal Proofs, June 2016.
Formal proof development.

[15] C. Diekmann, S.-A. Posselt, H. Niedermayer, H. Kinkelin, O. Hanka, and G. Carle.
Verifying Security Policies using Host Attributes.
In *Formal Techniques for Distributed Objects, Components, and Systems: 34th IFIP WG 6.1 International Conference (FORTE)*, pages 133–148, Berlin, Germany, June 2014. Springer Berlin Heidelberg.

[16] C. Diekmann, L. Schwaighofer, and G. Carle.
Certifying Spoofing-Protection of Firewalls.
In *11th International Conference on Network and Service Management (CNSM)*, pages 168–172, Barcelona, Spain, Nov. 2015.

[17] T. L. Hinrichs, N. S. Gude, M. Casado, J. C. Mitchell, and S. Shenker.
Practical Declarative Network Management.
In *1st ACM workshop on Research on enterprise networking*, WREN'09, pages 1–10. ACM, Aug. 2009.

[18] N. Kang, Z. Liu, J. Rexford, and D. Walker.
Optimizing the "One Big Switch" Abstraction in Software-defined Networks.
In *9th ACM Conference on Emerging Networking Experiments and Technologies*, CoNEXT, pages 13–24. ACM, Dec. 2013.

[19] P. Kazemian, G. Varghese, and N. McKeown.
Header Space Analysis: Static Checking for Networks.
In *9th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, NSDI'12, pages 113–126, San Jose, CA, Apr. 2012. USENIX Association.

Bibliography

[20] A. Khurshid, X. Zou, W. Zhou, M. Caesar, and P. B. Godfrey.
VeriFlow: Verifying Network-Wide Invariants in Real Time.
In *10th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, NSDI'13, pages 15–27, Lombard, IL, Apr. 2013. USENIX Association.

[21] H. Kim, J. Reich, A. Gupta, M. Shahbaz, N. Feamster, and R. Clark.
Kinetic: Verifiable Dynamic Network Control.
In *12th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, NSDI'15, pages 59–72, Oakland, CA, May 2015. USENIX Association.

[22] N. P. Lopes, N. Bjørner, P. Godefroid, and G. Varghese.
Network Verification in the Light of Program Verification.
Technical report, Sept. 2013.

[23] H. Mai, A. Khurshid, R. Agarwal, M. Caesar, P. B. Godfrey, and S. T. King.
Debugging the Data Plane with Anteater.
In *ACM SIGCOMM*, pages 290–301, Toronto, Ontario, Canada, Aug. 2011.

[24] R. Marmorstein and P. Kearns.
Firewall Analysis with Policy-based Host Classification.
In *20th USENIX Large Installation System Administration Conference (LISA)*, volume 6, Washington, D.C., Dec. 2006. USENIX Association.

[25] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner.
OpenFlow: Enabling Innovation in Campus Networks.
ACM SIGCOMM Computer Communication Review, 38(2):69–74, Apr. 2008.

[26] J. Michaelis and C. Diekmann.
LOFT – Verified Migration of Linux Firewalls to SDN.
Archive of Formal Proofs, Oct. 2016.
Formal proof development.

Bibliography

[27] J. Michaelis and C. Diekmann.
Routing.
Archive of Formal Proofs, Aug. 2016.
Formal proof development.

[28] C. Monsanto, J. Reich, N. Foster, J. Rexford, and D. Walker.
Composing Software Defined Networks.
In *10th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, NSDI'13, pages 1–13, Lombard, IL, Apr. 2013. USENIX Association.

[29] P. Porras, S. Shin, V. Yegneswaran, M. Fong, M. Tyson, and G. Gu.
A Security Enforcement Kernel for OpenFlow Networks.
In *First Workshop on Hot Topics in Software Defined Networks*, HotSDN '12, pages 121–126, Helsinki, Finland, Aug. 2012. ACM.

[30] S. Smolka, S. A. Eliopoulos, N. Foster, and A. Guha.
A Fast Compiler for NetKAT.
In *International Conference on Functional Programming (ICFP)*, pages 328–341. ACM, Sept. 2015.

[31] R. Soulé, S. Basu, P. J. Marandi, F. Pedone, R. Kleinberg, E. G. Sirer, and N. Foster.
Merlin: A Language for Provisioning Network Resources.
CoRR, abs/1407.1199, 2014.

[32] D. C. Verma.
Simplifying Network Administration Using Policy-Based Management.
IEEE Network, 16(2):20–26, Mar. 2002.

[33] M. von Maltitz, C. Diekmann, and G. Carle.
Taint Analysis for System-Wide Privacy Audits: A Framework and Real-World Case Studies.
1st Workshop for Formal Methods on Privacy, Nov. 2016.
workshop without proceedings.

Bibliography

[34] M. von Maltitz, C. Diekmann, and G. Carle.
Privacy Assessment using Static Taint Analysis (Tool Paper).
In *Formal Techniques for Distributed Objects, Components, and Systems: 37th IFIP WG 6.1 International Conference (FORTE)*, Neuchâtel, Switzerland, June 2017.

[35] G. G. Xie, J. Zhan, D. A. Maltz, H. Zhang, A. G. Greenberg, G. Hjálmtýsson, and J. Rexford.
On Static Reachability Analysis of IP Networks.
In *24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, volume 3, pages 2170–2183. IEEE, Mar. 2005.

[36] L. Yuan, H. Chen, J. Mai, C.-N. Chuah, Z. Su, and P. Mohapatra.
FIREMAN: A Toolkit for FIREwall Modeling and ANalysis.
In *IEEE Symposium on Security and Privacy*, pages 199–213, May 2006.

[37] B. Zhang, E. Al-Shaer, R. Jagadeesan, J. Riely, and C. Pitcher.
Specifications of a High-level Conflict-free Firewall Policy Language for Multi-domain Networks.
In *12th ACM symposium on Access control models and technologies*, SACMAT'07, pages 185–194. ACM, June 2007.

[38] H. Zhao, J. Lobo, A. Roy, and S. M. Bellovin.
Policy Refinement of Network Services for MANETs.
In *12th IFIP/IEEE International Symposium on Integrated Network Management (IM 2011)*, Dublin, Ireland, May 2011.