Demonstrating *topoS*: Theorem-Prover-Based Synthesis of Secure Network Configurations

manSDN/NFV 2015

Cornelius Diekmann Andreas Korsten Georg Carle

Chair for Network Architectures and Services
Technische Universität München
Munich, Germany

Agenda

1. Goals & Overview
2. Example
3. Pros & Cons
4. Discussion

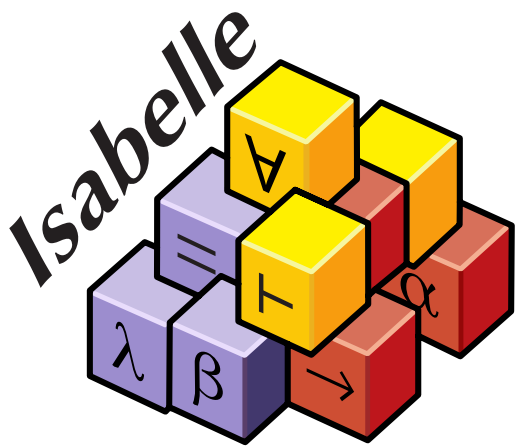
Demonstrating *topoS*: Theorem-Prover-Based Synthesis of Secure Network Configurations

topoS: a Constructive, Top-Down, Greenfield Approach for Network Security Management

- Translates high-level security goals to network security device configurations
- Easy-to-use^{TM}
- Automatic
- Visualizes intermediate steps
- Allows manual intervention
- Fully formally verified

topoS: a Constructive, Top-Down, Greenfield Approach for Network Security Management

- Translates high-level security goals to network security device configurations
- Easy-to-use^{TM}
- Automatic
- Visualizes intermediate steps
- Allows manual intervention
- Fully formally verified



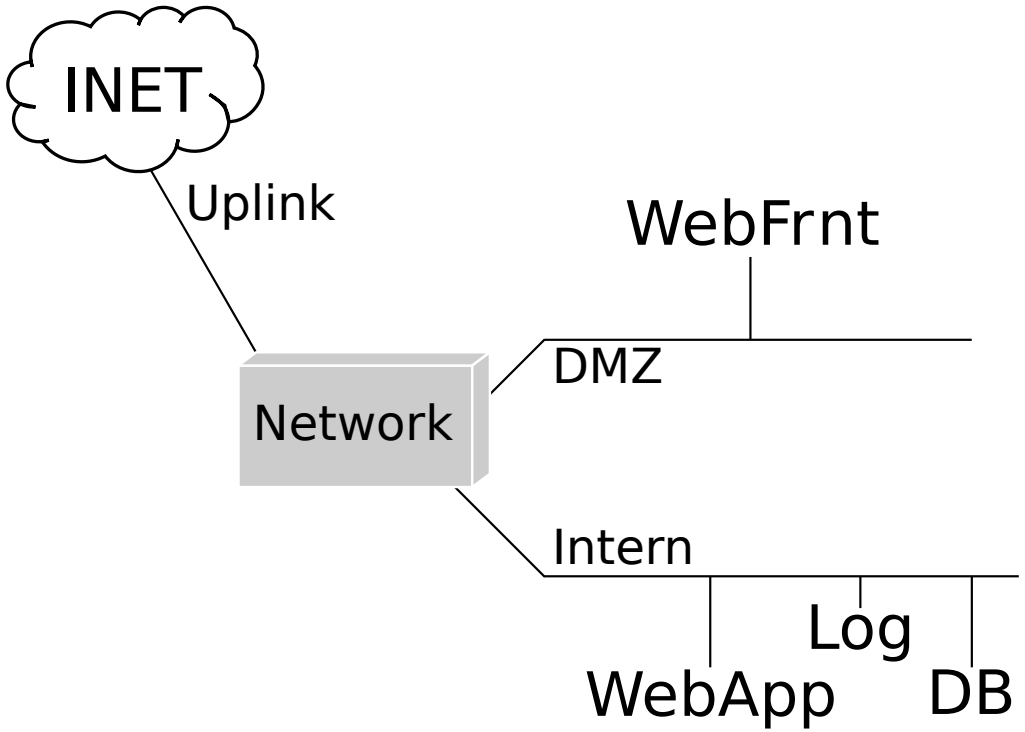
Overview

1. Formalize high-level security goals
   1. Categorize security goals
   2. Add scenario-specific knowledge
   3. ⋆ Auto-complete information
2. ⋆ Construct security policy
3. ⋆ Construct stateful policy
4. ⋆ Serialize security device configurations

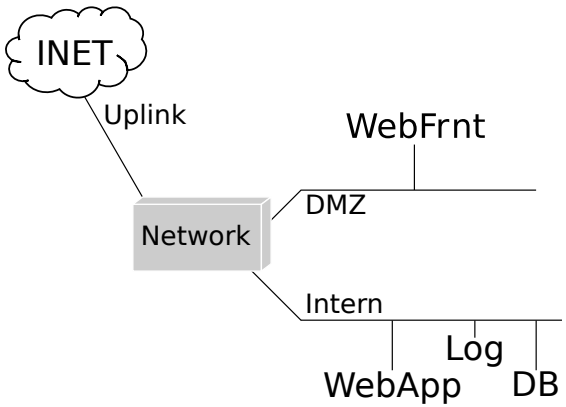
⋆ = automatic

Example

Overview – Network Schematic

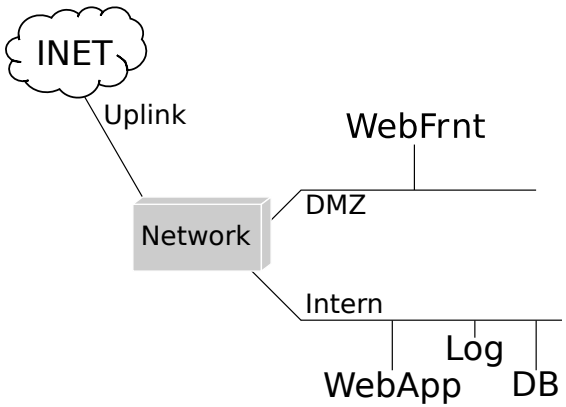


Overview



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

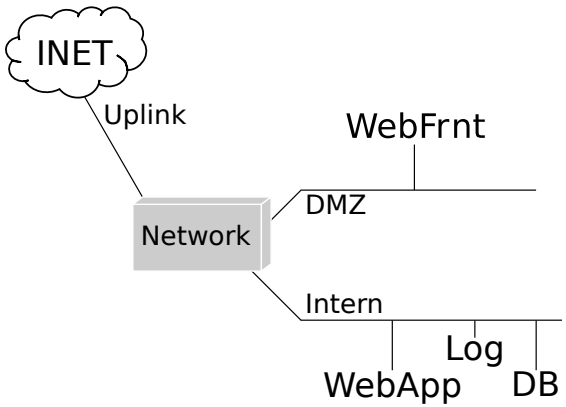
Overview



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

Subnets $\{\mathrm{DB} \mapsto \textit{internal}, \mathrm{Log} \mapsto \textit{internal}, \mathrm{WebApp} \mapsto \textit{internal}, \mathrm{WebFrnt} \mapsto \textit{DMZ}\}$

Overview

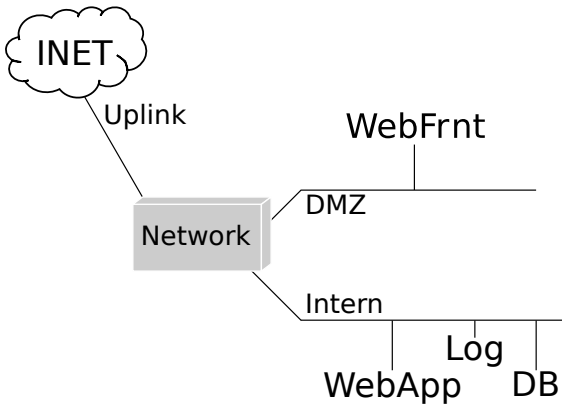


1. *DB*, *Log* and *WebApp* are internal hosts. *WebFrnt* must be accessible from outside.
2. Logging data must not leave the log server.
3. *DB*, *Log* contain confidential information. *WebApp* is trusted and allowed to declassify.
4. Only *WebApp* may access the *DB*.

Subnets $\{\mathrm{DB} \mapsto \textit{internal}, \mathrm{Log} \mapsto \textit{internal}, \mathrm{WebApp} \mapsto \textit{internal}, \mathrm{WebFrnt} \mapsto \textit{DMZ}\}$

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

Overview



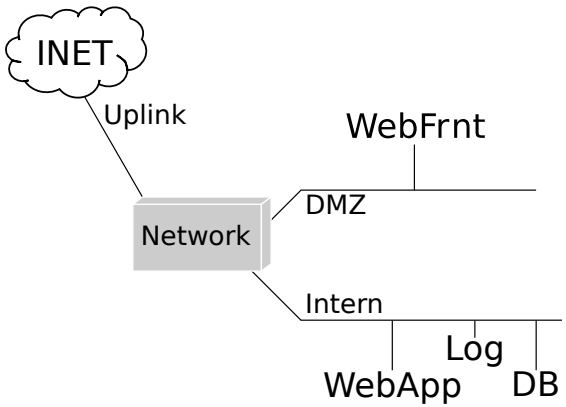
1. *DB*, *Log* and *WebApp* are internal hosts. *WebFrnt* must be accessible from outside.
2. Logging data must not leave the log server.
3. *DB*, *Log* contain confidential information. *WebApp* is trusted and allowed to declassify.
4. Only *WebApp* may access the *DB*.

Subnets $\{\mathrm{DB} \mapsto \textit{internal},\ \mathrm{Log} \mapsto \textit{internal},\ \mathrm{WebApp} \mapsto \textit{internal},\ \mathrm{WebFrnt} \mapsto \textit{DMZ}\}$

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

Bell LaPadula $\{\mathrm{DB} \mapsto \textit{confidential},\ \mathrm{Log} \mapsto \textit{confidential},\ \mathrm{WebApp} \mapsto \textit{declassify}\ (\textit{trusted})\}$

Overview



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

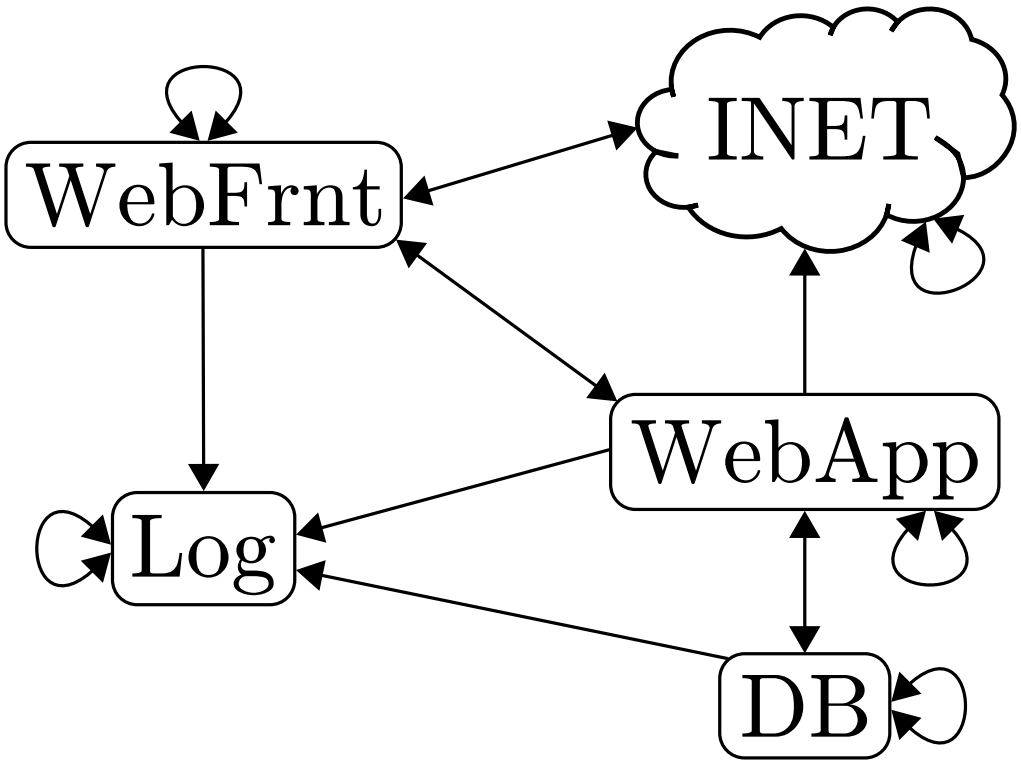
Subnets $\{\mathrm{DB} \mapsto \textit{internal}, \mathrm{Log} \mapsto \textit{internal}, \mathrm{WebApp} \mapsto \textit{internal}, \mathrm{WebFrnt} \mapsto \textit{DMZ}\}$

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

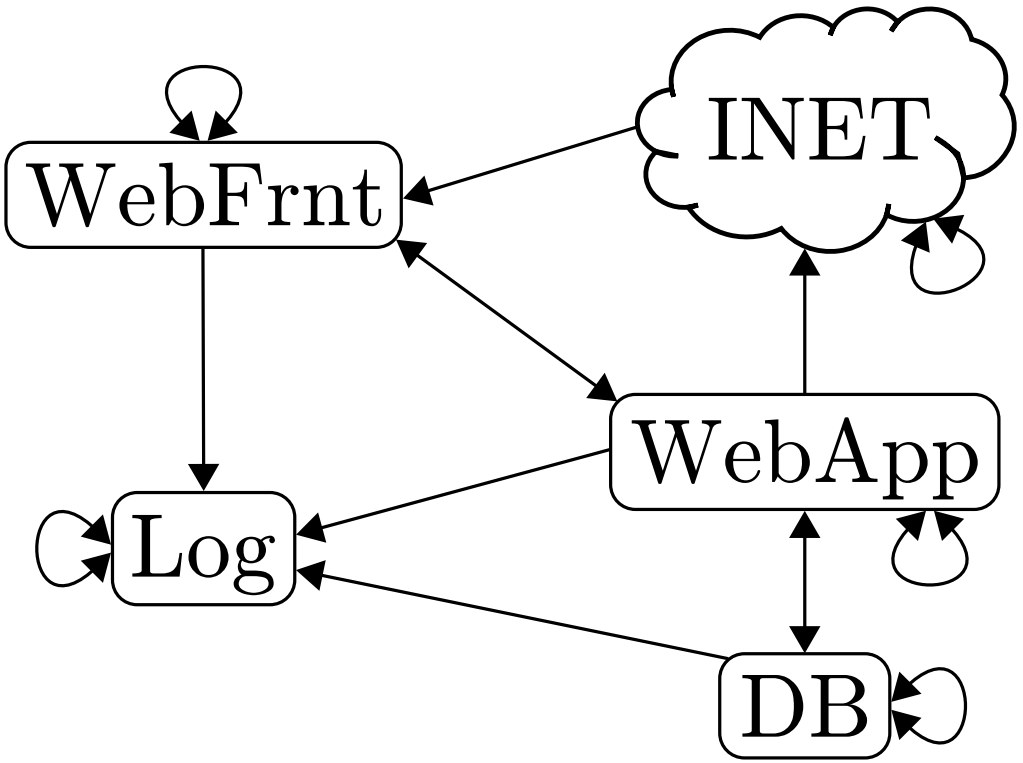
Bell LaPadula $\{\mathrm{DB} \mapsto \textit{confidential}, \mathrm{Log} \mapsto \textit{confidential}, \mathrm{WebApp} \mapsto \textit{declassify}\ (\textit{trusted})\}$

Comm. Partners $\{\mathrm{DB} \mapsto \textit{Access allowed by} : \mathrm{WebApp}\}$

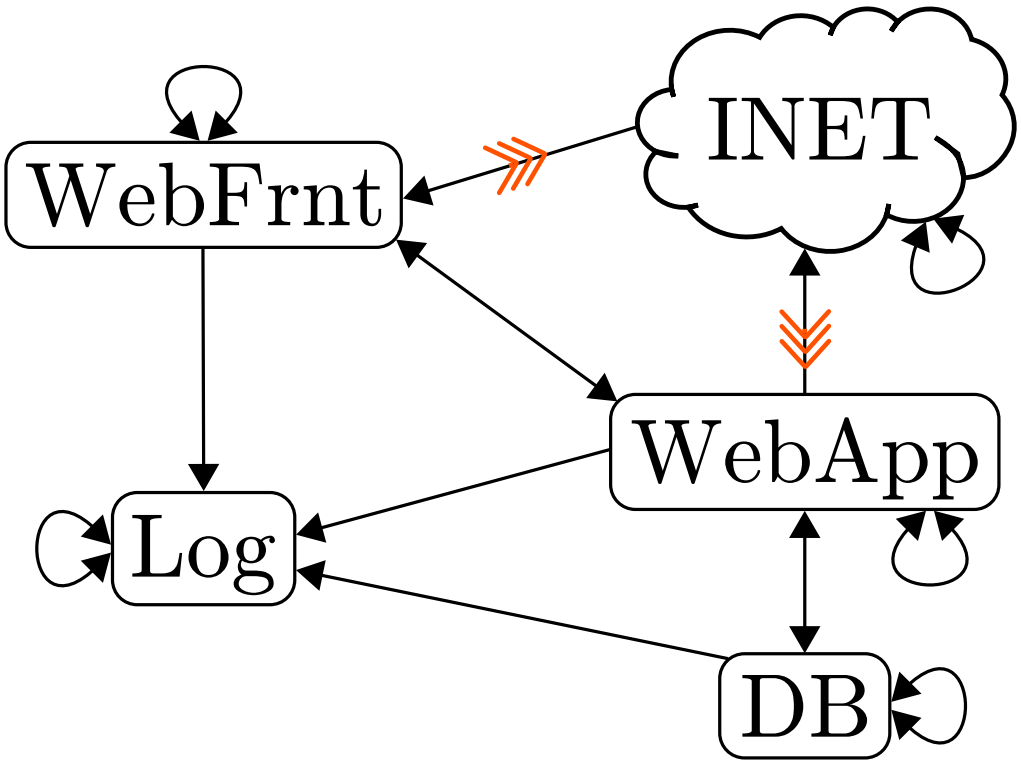
Security Policy



Security Policy – Manually Edited



Stateful Policy



Firewall Rules

```
FORWARD DROP
-A FORWARD -i tun0 -s $WebFrnt_ipv4 -o tun0 -d $Log_ipv4 -j ACCEPT
-A FORWARD -i tun0 -s $WebFrnt_ipv4 -o tun0 -d $WebApp_ipv4 -j ACCEPT
-A FORWARD -i tun0 -s $DB_ipv4 -o tun0 -d $Log_ipv4 -j ACCEPT
-A FORWARD -i tun0 -s $DB_ipv4 -o tun0 -d $WebApp_ipv4 -j ACCEPT
-A FORWARD -i tun0 -s $WebApp_ipv4 -o tun0 -d $WebFrnt_ipv4 -j ACCEPT
-A FORWARD -i tun0 -s $WebApp_ipv4 -o tun0 -d $DB_ipv4 -j ACCEPT
-A FORWARD -i tun0 -s $WebApp_ipv4 -o tun0 -d $Log_ipv4 -j ACCEPT
-A FORWARD -i tun0 -s $WebApp_ipv4 -o eth0 -d $INET_ipv4 -j ACCEPT
-A FORWARD -i eth0 -s $INET_ipv4 -o tun0 -d $WebFrnt_ipv4 -j ACCEPT
-I FORWARD -m state --state ESTABLISHED -i eth0 -s $INET_ipv4 -o tun0
-d $WebApp_ipv4 -j ACCEPT
-I FORWARD -m state --state ESTABLISHED -i tun0 -s $WebFrnt_ipv4 -o eth0
-d $INET_ipv4 -j ACCEPT
-P FORWARD DROP
```

OpenFlow Flow Table Template

```
# ARP Request
in_port=$port_src dl_src=$mac_src dl_dst=ff:ff:ff:ff:ff:ff
    arp arp_sha=$mac_src arp_spa=$ip4_src arp_tpa=$ip4_dst
    priority=40000 action=mod_dl_dst:$mac_dst,output:$port_dst

# ARP Reply
dl_src=$mac_dst dl_dst=$mac_src arp arp_sha=$mac_dst arp_spa=$ip4_dst
    arp_tpa=$ip4_src priority=40000 action=output:$port_src

# IPv4 one-way
in_port=$port_src dl_src=$mac_src ip nw_src=$ip4_src nw_dst=$ip4_dst
    priority=40000 action=mod_dl_dst:$mac_dst,output:$port_dst

# if src (resp. dst) is INET, replace $ip4_src (resp. $ip4_dst) with *
# and decrease the priority
```

```
ovs-vsctl set-fail-mode $switch secure && ovs-ofctl add-flows
```

Translation

Security Goals to Security Policy

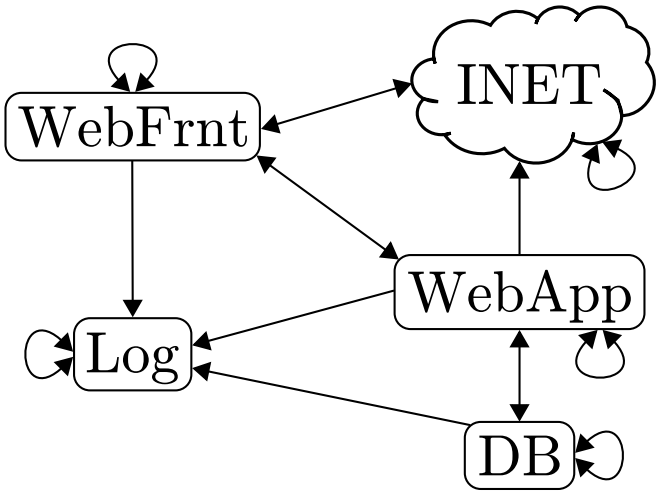
Subnets $\{\mathrm{DB} \mapsto \textit{internal}, \mathrm{Log} \mapsto \textit{internal}, \mathrm{WebApp} \mapsto \textit{internal}, \mathrm{WebFrnt} \mapsto \textit{DMZ}\}$

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

Bell LaPadula
$\{\mathrm{DB} \mapsto \textit{confidential}, \mathrm{Log} \mapsto \textit{confidential}, \mathrm{WebApp} \mapsto \textit{declassify}\ (\textit{trusted})\}$

Comm. Partners
$\{\mathrm{DB} \mapsto \textit{Access allowed by} : \mathrm{WebApp}\}$

$\Longrightarrow$



1. Complete Security Goals
2. Compute Security Policy

Security Goals to Security Policy (1)

- Completing Security Goals

Subnets $\{\mathrm{DB} \mapsto \textit{internal},\ \mathrm{Log} \mapsto \textit{internal},\ \mathrm{WebApp} \mapsto \textit{internal},\ \mathrm{WebFrnt} \mapsto \textit{DMZ},\ \mathrm{INET} \mapsto \bot\}$

Sink $\{\mathrm{Log} \mapsto \textit{Sink},\ \mathrm{DB} \mapsto \bot,\ \mathrm{WebApp} \mapsto \bot,\ \mathrm{WebFrnt} \mapsto \bot,\ \mathrm{INET} \mapsto \bot\}$

Bell LaPadula $\{\mathrm{DB} \mapsto \textit{confidential},\ \mathrm{Log} \mapsto \textit{confidential},\ \mathrm{WebApp} \mapsto \textit{declassify}\ (\textit{trusted}),\ \mathrm{WebFrnt} \mapsto \bot,\ \mathrm{INET} \mapsto \bot\}$

Comm. Partners $\{\mathrm{DB} \mapsto \textit{Access allowed by} : \mathrm{WebApp},\ \mathrm{Log} \mapsto \bot,\ \mathrm{WebApp} \mapsto \bot,\ \mathrm{WebFrnt} \mapsto \bot,\ \mathrm{INET} \mapsto \bot\}$

- $\bot$ can never lead to an unnoticed security problem, given enough information is provided

Security Goals to Security Policy (2)

- Computing Security Policy

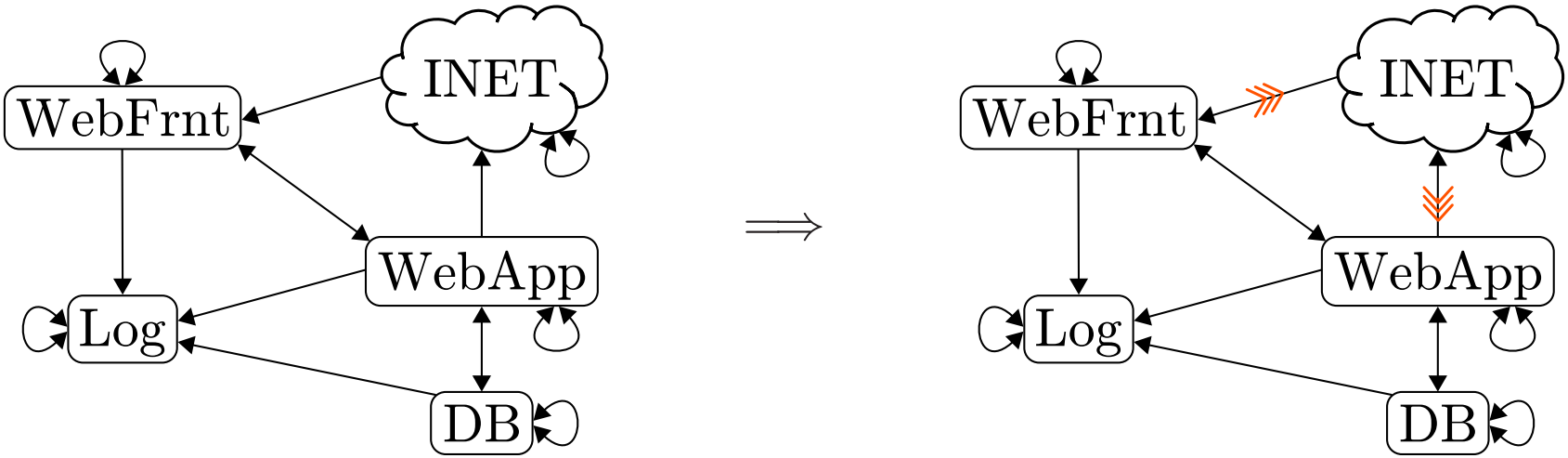
1. Start with allow-all policy:

   $\{\text{Log}, \text{DB}, \text{WebApp}, \text{WebFrnt}, \text{INET}\} \times$
   $\{\text{Log}, \text{DB}, \text{WebApp}, \text{WebFrnt}, \text{INET}\}$

2. Remove all rules which contradict the Security Goals

- Sound
- Complete: Maximum permissive policy
  (only for certain invariant templates)

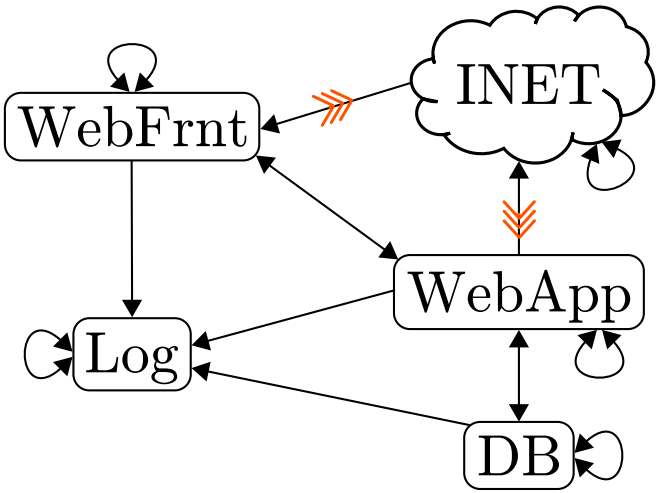
Security Policy to Stateful Policy



Consistency:

1. No information flow violation must occur
2. No access control side effects must be introduced

Stateful Policy to Firewall/SDN Rules



```
FORWARD DROP
-A FORWARD -i tun0 -s $WebFrnt_ipv4 -o tun0 -d $Log_ipv4 -j
-A FORWARD -i tun0 -s $WebFrnt_ipv4 -o tun0 -d $WebApp_ipv4
-A FORWARD -i tun0 -s $DB_ipv4 -o tun0 -d $Log_ipv4 -j ACCEP
-A FORWARD -i tun0 -s $DB_ipv4 -o tun0 -d $WebApp_ipv4 -j A
-A FORWARD -i tun0 -s $WebApp_ipv4 -o tun0 -d $WebFrnt_ipv4
-A FORWARD -i tun0 -s $WebApp_ipv4 -o tun0 -d $DB_ipv4 -j A
-A FORWARD -i tun0 -s $WebApp_ipv4 -o tun0 -d $Log_ipv4 -j
-A FORWARD -i tun0 -s $WebApp_ipv4 -o eth0 -d $INET_ipv4 -j
-A FORWARD -i eth0 -s $INET_ipv4 -o tun0 -d $WebFrnt_ipv4 -j
-I FORWARD -m state --state ESTABLISHED -i eth0 -s $INET_ipv
-I FORWARD -m state --state ESTABLISHED -i tun0 -s $WebFrnt
```

- Term rewriting
- Translating assumptions

Structure Enforced network connectivity structure = policy. Links: confidential and integrity protected.

Authenticity Policy's entities must match their network representation (e.g., IP/MAC addresses).

State The stateful connection handling must match the stateful policy's semantics.

Enforcement Assumptions

Firewall & Central VPN Server

- Structure: central OpenVPN server (`tun`) + iptables ✓
- Authenticity: X.509 certificates ✓
- State: iptables ✓

SDN (layer 2 network)

- Structure: Known ports, MAC, IP addresses + MAC broadcast rewriting ✓
  ARP information leak → needs controller to answer ARP
- Authenticity: No ARP attacks, enforced port/MAC/IP mapping ✓
- State: ✗
  add iptables firewall ✓

Pros & Cons

Pros & Cons

Pros

- Fully formally verified
- Executable
- ‘Deployable’ security goals
- Manual intervention on intermediate results

Cons

- Only one security device
- Static & needs ‘names’ of entities
- No specification of paths, bandwith, QoS, ...

  $\Rightarrow$ Merlin, NetKAT, ...

Pros & Cons

Pros

- Fully formally verified
- Executable
- ‘Deployable’ security goals
- Manual intervention on intermediate results

Cons

- Only one security device
- Static & needs ‘names’ of entities
- No specification of paths, bandwith, QoS, ...

  $\Rightarrow$ Merlin, NetKAT, ...

Solves access-control-matrix-related issues in security management

Availability

topoS and the correctness proofs can be obtained at

`https://github.com/diekmann/topoS/`

or

`http://afp.sourceforge.net/entries/Network_Security_Policy_Verification.shtml`

Formalized Example: `Distributed_WebApp.thy`

Runs live at: `http://otoro.net.in.tum.de/goals2config/`