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™
- Automatic
- Visualizes intermediate steps
- Allows manual intervention
- Fully formally verified

- Translates high-level security goals to network security device configurations
- Easy-to-use™
- 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.
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 \{DB \rightarrow internal, \ Log \rightarrow internal, \ WebApp \rightarrow internal, \ WebFrnt \rightarrow 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 \{DB \leftrightarrow \textit{internal}, \ Log \leftrightarrow \textit{internal}, \ WebApp \leftrightarrow \textit{internal}, \ WebFrnt \leftrightarrow \textit{DMZ}\}

Sink \{\textit{Log} \leftrightarrow \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 \{ DB \mapsto \textit{internal}, \ Log \mapsto \textit{internal}, \\
WebApp \mapsto \textit{internal}, \ WebFrnt \mapsto \textit{DMZ} \}\n
Sink \{ Log \mapsto \textit{Sink} \}\n
Bell LaPadula \{ DB \mapsto \textit{confidential}, \ Log \mapsto \textit{confidential}, \\
WebApp \mapsto \textit{declassify (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.
4. Only *WebApp* may access the *DB*.

Subnets \{DB \mapsto internal, \ Log \mapsto internal, \\
    WebApp \mapsto internal, \ WebFrnt \mapsto DMZ\}\n
Sink \{Log \mapsto Sink\}\n
Bell LaPadula \{DB \mapsto confidential, \ Log \mapsto confidential, \\
    WebApp \mapsto declassify (trusted)\}\n
Comm. Partners \{DB \mapsto Access allowed by : WebApp\}
Security Policy
Security Policy – Manually Edited
Stateful Policy

manSDN/NFV'15, Demonstrating topoS:, Theorem-Prover-Based Synthesis of Secure Network Configurations
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_spa=$ip4_src nw_dst=$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 \{ DB \mapsto \textit{internal}, \ Log \mapsto \textit{internal}, \ WebApp \mapsto \textit{internal}, \ WebFrnt \mapsto \textit{DMZ} \}

Sink \{ \Log \mapsto \textit{Sink} \}

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

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

1. Complete Security Goals
2. Compute Security Policy
Security Goals to Security Policy (1)

- Completing Security Goals

Subnets \{DB \mapsto \text{internal}, \ Log \mapsto \text{internal},
\ WebApp \mapsto \text{internal}, \ WebFrnt \mapsto \text{DMZ},
\ INET \mapsto \bot\} 

Sink \{Log \mapsto \text{Sink},
\ DB \mapsto \bot, \ WebApp \mapsto \bot, \ WebFrnt \mapsto \bot, \ INET \mapsto \bot\} 

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

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

   \[\{\text{Log, DB, WebApp, WebFrnt, INET}\} \times \{\text{Log, DB, WebApp, WebFrnt, INET}\}\]

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

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.

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

▶ 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.

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, bandwidth, QoS, ...
  ⇒ 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, bandwidth, QoS, ...
  ⇒ 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


Formalized Example: Distributed_WebApp.thy

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