Network Verification
Solvers, Symmetries, Surgeries

Nikolaj Bjørner

NetPL, August, 2016
Networking needs:
Configuration Sanity/Synthesis, Programming, Provisioning

Network Design Automation

Z3 advances:
Bit-vector Reasoning ~ Header Spaces
Reachability Checking, Quantitative Reasoning
Symbolic Analysis with Z3

$x^2 + y^2 < 1 \text{ and } xy > 0.1$

$sat, x = \frac{1}{8}, y = \frac{7}{8}$

$x^2 + y^2 < 1 \text{ and } xy > 1$

$unsat, Proof$

Is execution path $P$ feasible?

Does Policy Satisfy Contract?

SAGE

Z3 solved more than 10 billion constraints created by SymEx tools including SAGE checking Win8,10 and Office

Z3 used by Pex, Static Driver Verifier, many other tools
Our competition also likes symbolic solving 😊

Microsoft Azure and MSR are always hiring.

Top engineering and research orgs with big and long term bets.
# Calculus and Solvers

<table>
<thead>
<tr>
<th>Application</th>
<th>Calculus</th>
<th>Solver</th>
</tr>
</thead>
<tbody>
<tr>
<td>SecGuru:</td>
<td>Satisfiability</td>
<td>SAT</td>
</tr>
<tr>
<td>Access Control</td>
<td>Modulo Theories</td>
<td></td>
</tr>
<tr>
<td>Routing Validation</td>
<td>for Bit-vectors</td>
<td></td>
</tr>
<tr>
<td>Static configurations for</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Border Gateway Protocol</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Checking <em>beliefs</em> in networks</td>
<td>Network Optimized Datalog</td>
<td>Datalog for</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Header Spaces</td>
</tr>
<tr>
<td></td>
<td>Network Symmetries and Surgeries</td>
<td>Tries for</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Header Space</td>
</tr>
<tr>
<td></td>
<td></td>
<td>partitioning</td>
</tr>
<tr>
<td>Verifying SDN controllers</td>
<td>Quantified logical formulas</td>
<td>Instantiation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>based reasoning</td>
</tr>
</tbody>
</table>
## Verification: Values and Obstacles

<table>
<thead>
<tr>
<th></th>
<th>Hardware</th>
<th>Software</th>
<th>Networks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chips</td>
<td>Burned into silicone</td>
<td>Exploitable, workarounds</td>
<td>Latent, Exposed</td>
</tr>
<tr>
<td>Bugs are:</td>
<td>Costly recalls</td>
<td>Online updates</td>
<td>Live site incidents</td>
</tr>
<tr>
<td>Dealing with bugs:</td>
<td>Design Complexity</td>
<td>Code churn, legacy, false positives</td>
<td>Topology, configuration churn</td>
</tr>
<tr>
<td>Obstacles to eradication:</td>
<td>Cut time to market</td>
<td>Safety/OS critical systems, Quality of code base</td>
<td>Meet SLA, Utilize bandwidth, Enable richer policies</td>
</tr>
<tr>
<td>Value proposition</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---
SecGuru
Policies as Logical Formulas

Precise Semantics as formulas

\[ (10.20.0.0 \leq srcIp \leq 10.20.31.255) \land \\
Allow: \ (157.55.252.0 \leq dstIp \leq 157.55.252.255) \land \\
(\text{protocol} = 6) \]

\[ (65.52.244.0 \leq dstIp \leq 65.52.247.255) \land \\
Deny: \ (\text{protocol} = 4) \]

Combining semantics

\[ \left( \bigvee_i \text{Allow}_i \right) \land \left( \bigwedge_j \neg \text{Deny}_j \right) \]
Access Control

**Contract:**

DNS ports on DNS servers are **accessible** from tenant devices over both TCP and UDP.

**Contract:**

The SSH ports on management devices are **inaccessible** from tenant devices.
SecGuru workflow

StreamInsight Complex Event Processing (CEP) Application

Contract Stream

Configuration Stream

SecGuru ACL Validation

Contract Stream

Z3 Theorem Prover

Device Validation Stream

SQL Server StreamInsight

Reports Database

Alerts + Reporting in WANetmon

Windows Azure Network Monitoring Infrastructure
Regression test suite + SecGuru check correctness of Edge ACL prior to deployment

Several major Edge ACL pushes

no major impact on any services

Stable state

SecGuru for GNS edge ACLs

Size of AS8075-EDGE-PROTECT-IN ACL

AS8075-EDGE-PROTECT-IN ACL Line Modifications
Beyond Z3: a new idea to go from one violation to all violations

\[
\left( \bigvee_i \text{Allow}_i \right) \land \left( \bigwedge_j \neg \text{Deny}_j \right) \Rightarrow Z3 \Rightarrow \neg \left[ \left( \bigvee_m \text{Allow}_m \right) \land \left( \bigwedge_n \neg \text{Deny}_n \right) \right]
\]

Semantic Diffs

\[
\text{srcIp} = 10.20.0.0/16, 10.22.0.0/16
\]

\[
\text{dstIp} = 157.55.252.000/24, 157.56.252.000/24
\]

\[
\text{port} = 80, 443
\]

Representing solutions
- \(2 \times 2^{16} \times 2 \times 2^8 \times 2 = 2^{27}\) single solutions, or
- 8 products of contiguous ranges, or
- A single product of ranges

SecGuru contains optimized algorithm for turning single solutions into all (product of ranges)
Verifying Forwarding Rules with SecGuru

Cluster\text{\( (dst) \Rightarrow \)}

\text{\( \text{Router}_1 (dst) \equiv \text{Router}_2 (dst) \)}
Network Reachability
Checking **beliefs** in Dynamic Networks

Which packets can reach B from A?

Datalog useful for encoding a broad range of queries. We use **belief** for a class of general properties that one may expect to hold of networks.

**Sample belief:** packets flow through middle-box

[Lopes, B, Godefroid, Jayaraman, Varghese NSDI’15]

\[
\begin{align*}
G_{12} & := \ dst = 10* \land src = 01* \\
G_{13} & := \neg G_{12} \land dst = 1** \\
G_{2B} & := \ dst = 10* \\
G_{3D} & := \ src = 1** \\
G_{32} & := \neg G_{3D} \land dst = 1** \\
Id & := src' = src \land dst' = dst \\
Set0 & := src' = src \land dst' = dst[2] 0 dst[0]
\end{align*}
\]
Applying NoD to P4\textsubscript{14}

reach(S') :-
reach(S),
router\_processing(S, S').

router\_processing(S, S') :-
reset\_local\_data(S, S0),
start(S0, S1),
egress(S1.local.addr, S1.std_md.egress_spec, Next, Port),
S' = { S1 with std_md.ingress_port = Port, local.addr = Next }.

reset\_local\_data(S, S') :-
S' = { S with local_md = 0, std_md = 0, parsed = 0 }.

table table \{ reads actions \} ∈ Prog
act ∈ actions
vals = add\_entry\_table\_act(S.reads)

\[
\begin{align*}
S, E & \xrightarrow{\text{act}(\text{vals})} S' \\
S', E & \xrightarrow{\text{stmt}} S''
\end{align*}
\]

apply \frac{\text{apply(\text{table}\{\text{act}\{\text{stmt}\}\}})}{S, E} \rightarrow S''

+ P4 code + Config \rightarrow NoD

[Lopes, Rybalchenko, B, McKeown, Talayco, Varghese]
Scaling Network Verification using Symmetry and Surgery

A Theory of Network Dataplanes

- $\text{out} : \text{Nodes} \rightarrow 2^\text{Ports}$
- $\text{Port} \equiv \{n \cdot i \mid n \in \text{Nodes}, i \in \text{out}(n)\}$
- $\text{links}: \text{Port}_N \rightarrow \text{Nodes}$
- $h@n. i \longrightarrow h'@n'. i'$
  $\in \text{Trans}$
  $\subseteq (\text{Header} \times \text{Port}) \times (\text{Header} \times \text{Port})$
  Such that $n' = \text{links}(n. i), i' \in \text{out}(n')$

A basis for defining bisimulation relations:

$h@n. i \sim h'@n'. i'$

[Plotkin, B, Lopes, Rybalchenko, Varghese, POPL 16]
Scaling Network Verification using Symmetry and Surgery

A Toolbox of Network Transformations

Example: Replace a core of a network by a single hub:

[Plotkin, B, Lopes, Rybalchenko, Varghese, POPL 16]
Scaling Network Verification using Symmetry and Surgery

Scaling comprehensive Network Verification

**Example:** Move rules from B to C if forwarding is the same.

Relies on efficient representation of header equivalence classes.
Forwarding rules

<table>
<thead>
<tr>
<th>Rule</th>
<th>Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>1**</td>
<td>port1</td>
</tr>
<tr>
<td><em>1</em></td>
<td>port2</td>
</tr>
<tr>
<td>**1</td>
<td>port3</td>
</tr>
<tr>
<td>***</td>
<td>port2</td>
</tr>
</tbody>
</table>

Original guards

Intersection

[B, Juniwal, Mahajan, Seshia, Varghese MSR-TR]
Summary

Much is about Configuration Correctness:

• Is intent captured? (SecGuru)
• Usage (NoD + P4)
• Synthesis (Control Plane)
• Bandwidth Use and Provisioning (QNA)

Modern packet switched networks a good use case for PL + Symbolic Methods