SNAP: Stateful Network-Wide Abstractions for Packet Processing

Mina Tahmasbi Arashloo¹, Yaron Koral¹, Michael Greenberg², Jennifer Rexford¹, and David Walker¹

¹ Princeton University, ² Pomona College
Two-tiered Programming Model

1. **Stateless** OpenFlow rules
   - process each packet independently

2. **Stateful** component
   - on top of controller
   - changes rules based on current state
Example - Detecting DNS Reflection Attacks

Attacker-controlled botnet

Small spoofed DNS request

Amplified DNS response from open resolver

http://ddosandbotnets.blogspot.com
Example - Detecting DNS Reflection Attacks [1]

- For each host
  - Record DNS requests
  - Increment a counter for unmatched responses
  - Mark as suspicious after a threshold

Practical Concerns

• Cannot send every packet to the controller

• Cannot have per-packet stateful processing
  
    • decide what to do with the packet based on packets seen so far!
Opportunity: Local State on Data Plane

- Programmatic control over local state
  - P4, OpenState, POF, Open vSwitch
- Simple stateful network functions can be offloaded to programmable switches!
Opportunity:
Local State on Data Plane

- Programmatic control over local state
- P4, OpenState, POF, Open vSwitch
- Simple stateful network functions can be offloaded to programmable switches!

Too low level!
Our Goal

1. **High-level programming language**
   - one big **stateful** switch!

2. **Compiler**
   - to automatically distribute stateful programs
Our Contribution

SNAP Language and Compiler
SNAP - Language

- One Big Stateful Switch (OBSS)
- Composition
DNS Reflection Detection in SNAP

```plaintext
if srcip in CSNET & dstport = 53 then
  seen[srcip][dns.id] ← True
else if dstip in CSNET & srcport = 53 then
  if ~seen[dstip][dns.id] then
    unmatched[dstip]++;
    if unmatched[dstip] = threshold then
      susp[dstip] ← True
  else
    id
else
  id
```
DNS Reflection Detection in SNAP

```python
if srcip in CSNET & dstport = 53 then
    seen[srcip][dns.id] ← True
else if dstip in CSNET & srcport = 53 then
    if ~seen[dstip][dns.id] then
        unmatched[dstip]++;
        if unmatched[dstip] = threshold then
            susp[dstip] ← True
    else id
else id
```
Composition
SNAP - Compiler

**where** to place state variables

**how to forward** packets through them
Program Analysis

- Each flow should go through switches with
  - all the state variables that it needs
  - in the correct order
Mixed-Integer Linear Program (MILP)
Mixed-Integer Linear Program (MILP)
Mixed-Integer Linear Program (MILP)
Intermediate Representation (IR)
Intermediate Representation (IR)

Composable and Easily Partitioned IR
Extended Forwarding Decision Diagrams (xFDDs)

- Intermediate node: test on header fields and state
- Leaf: set of action sequences

\[
\begin{align*}
\text{dstip} &= 10.0.0.1 \\
\text{srcip} &= \text{dstip} \\
\text{s[srcip]} &= 2 \\
\{\text{s[dstip]} &\leftarrow 2\} &\quad \{\text{drop}\}
\end{align*}
\]
Distributing the xFDD
Distributing the xFDD
Putting It All Together

ISP1

ISP2

CS

EE

srcip in CSNET

dstip in CSNET

srcport = 53

dstip in EENET

seen[dstip] [dns.id]

unmatched[dstip]++;
outport ← CS

outport ← EE

{unmatched[dstip]++;
susp[srcip][dstip] ← True;
outport ← CS}
Putting It All Together

ISP1

ISP2

CS

EE

srcip in CSNET

dstip in CSNET

seen[dstip]
[dns.id]

srcport = 53

dstip in EENET

unmatched[dstip] = threshold - 1

outport ← CS

{unmatched[dstip]++;
susp[srcip][dstip] ← True;
outport ← CS}
Putting It All Together

ISP1 -> CS -> ISP2

srcip in CSNET

ISP1

ISP2

EE

dstip in CSNET

seen[dstip] [dns.id]

unmatched[dstip] = threshold - 1

outport ← CS

{unmatched[dstip]++; susp[srcip][dstip] ← True; outport ← CS}

srcport = 53

dstip in EENET

outport ← EE

unmatched[dstip]++

24
Compiler Evaluation - Benchmarks and Scenarios

- Topologies with 100s of switches and edges
  - 7 campus and ISPs

- Scenarios
  - Cold start
  - Policy change
  - Topology/TM change
Compiler Evaluation - Results

![Graph showing time (sec.) vs. topology changes and policy changes for different institutions.]

- Stanford
- Berkley
- Purdue
- ISP 1755
- ISP 1221
- ISP 6461
- ISP 3257

Legend:
- Topology/TM Change
- Policy Change
- Cold Start
Thank You!

Questions?