

Security and SDN

Roy H Campbell

University of Illinois at Urbana-Champaign

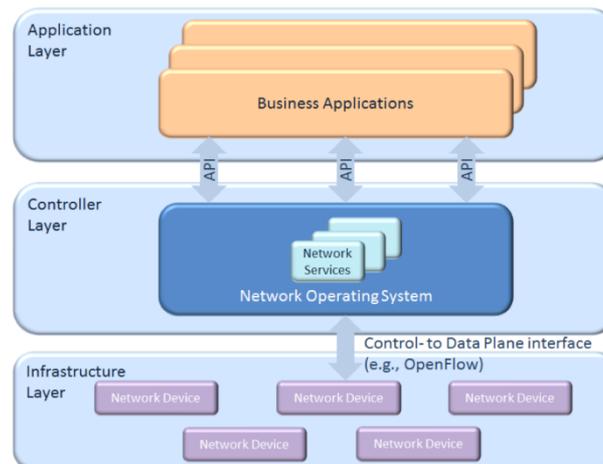
12/17/2013

Goal

SDN offers many potential advantages for better security in networks. What research needs to be accomplished to secure SDN deployment?

Introduction

- Ultimate goal (according to the Shenker perspective)
 - Provide abstractions necessary to treat the network as a single entity.
 - Similar to an operating system's abstractions of hardware
 - Hence, Network Operating System (NOS)



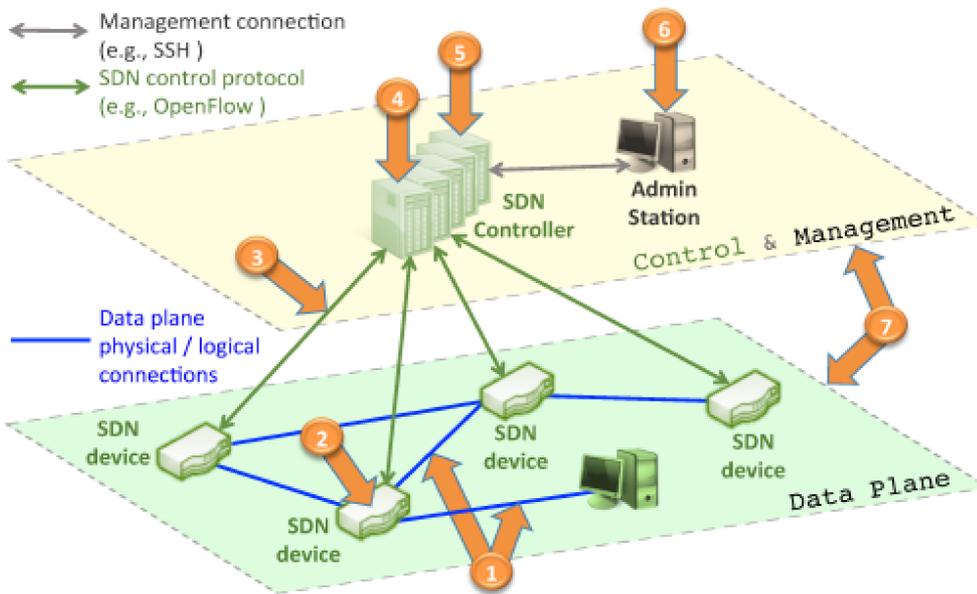
Outline

- Security advantages of SDN?
- Security disadvantages of SDN?
- Is SDN technology mature wrt security?
- Privacy, Confidentiality?

Advantages of SDN (Kreutz et al)

- Separation of control plane from data plane
- Network switches - simple forwarding devices
- Control logic in a logically centralized controller
- Controller dictates network behavior
 1. it is simpler and less error-prone to modify network policies through software, than via low-level device configurations.
 2. a control program can automatically react to spurious changes of the network state and thus maintain the high-level policies in place.
 3. the centralization of the control logic in a controller with global knowledge of the network state simplifies the development of more sophisticated network functions.

Disadvantages of SDN (Kreutz et al)



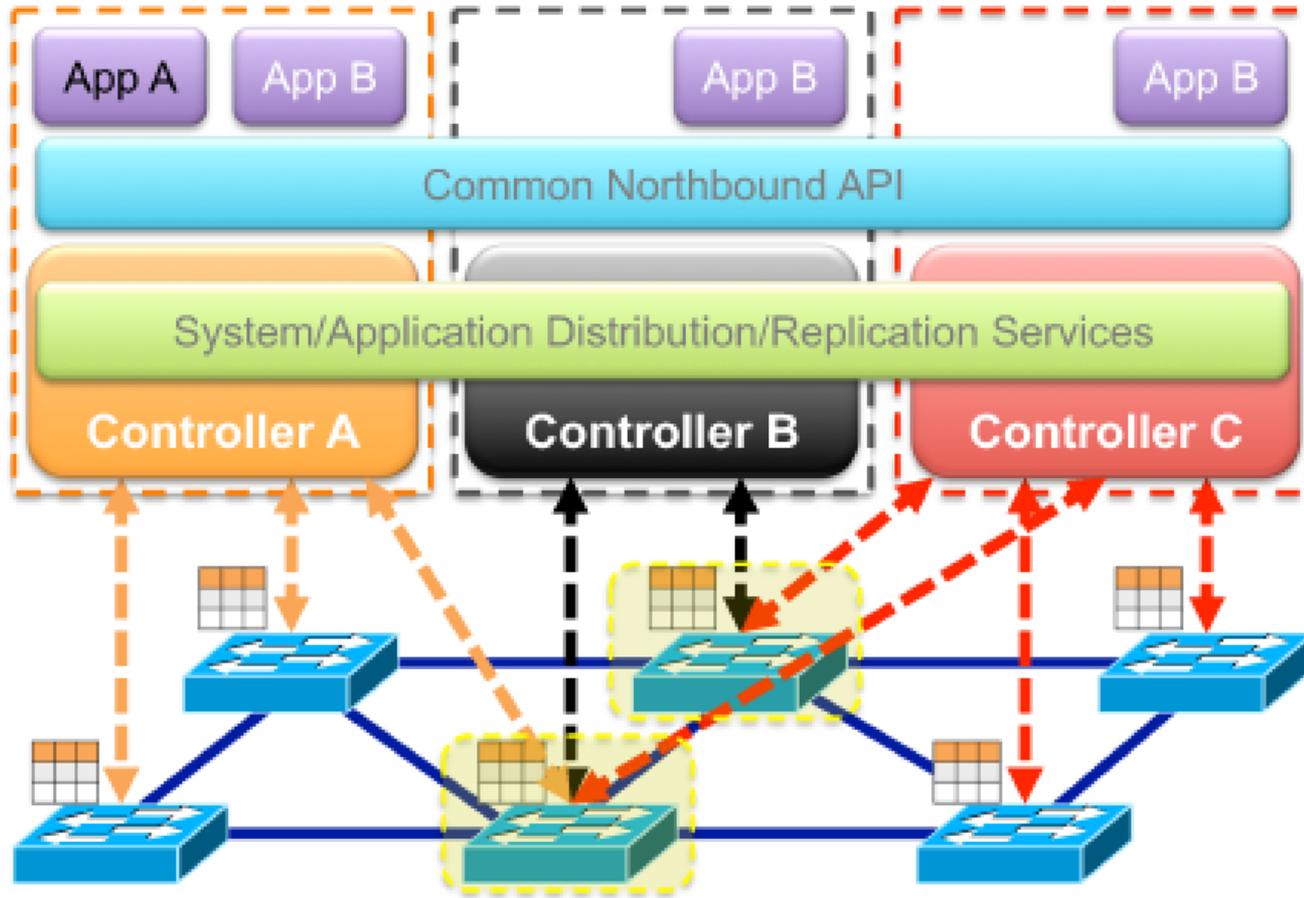
- 1 forged or faked traffic flows
- 2 attacks on and vulnerabilities in controllers
- 3 attacks on control plane communications
- 4 attacks on and vulnerabilities in controllers
- 5 lack of mechanisms to ensure trust between the controller and management applications
- 6 attacks on and vulnerabilities in administrative stations
- 7 lack of trusted resources for forensics and remediation

D. Kreutz, F. Ramos, and P. Verissimo, "Towards secure and dependable software-defined networks," Proc. of the second ACM SIGCOMM workshop on Hot topics in software defined networking, 2013, pp. 55–60.

Specific/Non-specific Threats to SDN (Kreutz et al)

Threats	Specific to SDN?	Consequences in SDN
Vector 1	no	can be a door for DoS attacks
Vector 2	no	but now the impact is potentially augmented
Vector 3	yes	communication with logically centralized controllers can be explored
Vector 4	yes	controlling the controller may compromise the entire network
Vector 5	yes	malicious applications can now be easily developed and deployed on controllers
Vector 6	no	but now the impact is potentially augmented
Vector 7	no	it is still critical to assure fast recovery and diagnosis when faults happen

Secure and Dependable SDN? (Kreutz et al)



Solutions to Threat Vectors

(Kreutz et al)

Solution/mechanism	Threat vectors
Replication	1, 4, 5, 7
Diversity	3, 4, 6
Self-healing	2, 4, 6
Dynamic switch association	3, 4
Trust between controllers & devices	1, 2, 3
Trust between controllers & apps	4, 5
Security domains	4, 5
Secure components	4, 5, 7
Fast and reliable update & patching	2, 4, 6

Lack of TLS and Man in the Middle? (Kreutz et al)

Switch Vendor	TLS Support
<i>HP</i>	No [?]
<i>Brocade</i>	No
<i>Dell</i>	No
<i>NEC</i>	Partial ²
<i>Indigo</i>	No
<i>Pica8</i>	No
<i>OpenWRT</i>	Yes [?]
<i>Open vSwitch</i>	Yes [?]

OpenFlow Controller	TLS Support
<i>NOX</i>	Controller only ³
<i>POX</i>	No [?]
<i>Beacon</i>	No [?]
<i>Floodlight</i>	No [?]
<i>MuL</i>	No [?]
<i>FlowVisor</i>	No ⁴
<i>Big Network Controller</i>	No ⁵
<i>Open vSwitch Controller</i>	Yes [?]

K. Benton, L. J. Camp, and C. Small, "OpenFlow vulnerability assessment," in Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking, 2013, pp. 151–152.

Is SDN technology mature wrt security?

- Will only address a few areas; too much to cover it all.
- Early work was focused on the NOS/controller, implementing basic functionality, and the APIs they exposed
 - NOX
 - Low-level API
 - FlowVisor
 - Network slicing
- Also, much effort has been put into languages for describing network-wide flow-level policies
 - Flow-based Management Language
 - Resonance
- Note. SDN is not secure “by design”

Research Trends: Scalability

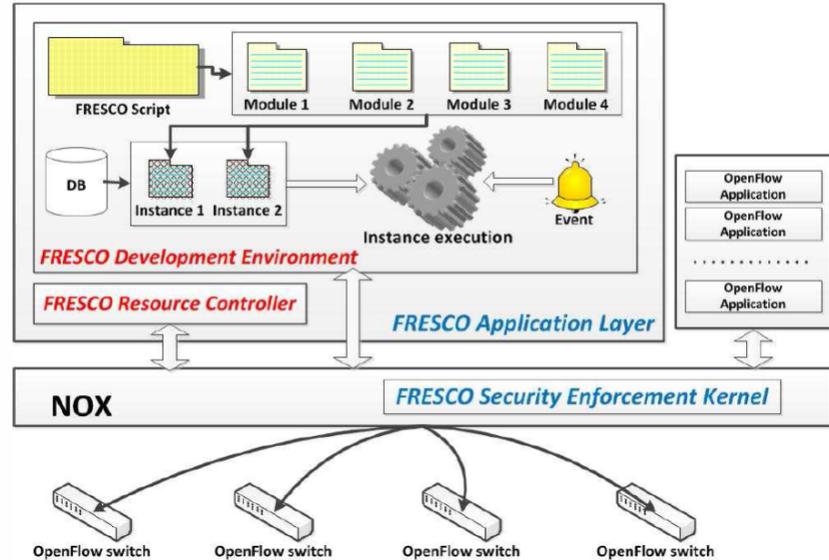
- Became clear NOX would not scale
 - Could handle up to about 30k flows/sec
 - High capacity networks could have between 100k and 10M flows/sec
- Several multi-threaded solutions have emerged
 - Beacon
 - Maestro
 - NOX-MT
- Other solutions are physically distributed while remaining logically centralized
 - Onix - DHT
 - HyperFlow - DFS
 - Kandoo – Hierarchical structure
- BUT: Can SDN control state be made consistent?

Research Trends: Security

- Until recently, security was largely overlooked as a research topic
 - OpenFlow controllers may use SSL/TLS to secure their communications with switches.
- Several surveys have considered potential weaknesses
- Kreutz, et al., consider how SDN in general changes security considerations
 - Examines how OpenFlow model introduces new threat vectors or requires a different approach to mitigating old threat vectors.
 - For example, attacks on centralized control plane represent a new threat vector
 - Inadequate forensics is an old problem that is slightly augmented in SDN

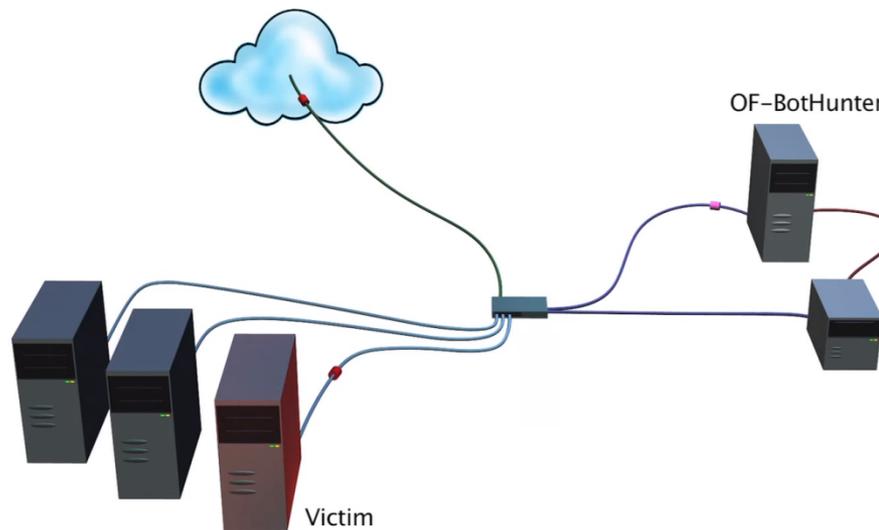
Research Trends: Security

- www.openflowsec.org
 - SRI and Texas A&M
 - Led by Phil Porras
- Security Extended Kernels
 - Native code extensions to existing OpenFlow controllers
 - NOX → FortNOX /FRESKO
 - Floodlight → SE-Floodlight
 - Detect and reconcile conflicting flow
 - Implements IPC wrapper to allow applications to run in individual user accounts on host.
 - Implements security authorization for applications running on top of the



Research Trends: Security

- Dynamic Quarantine with SE-Floodlight and OF-BotHunter
 - Passive analysis system that detects traffic patterns consistent with malware
 - Issues quarantine directives to Security Actuator, which translates them to flow rules



OCEAN Cluster

(Brighten Godfrey, Matthew Caesar)

- VeriFlow: Verifying Network-Wide Invariants in Real Time
 - Designed to help network operators understand network behavior
 - Verifies network-wide correctness and security
 - Real time operation
- AntEater
 - Finds networking bugs via data plane analysis
 - Models data plane behavior as instances of satisfiability problems
 - Uses formal analysis techniques to systematically analyze the network

Security-Enhanced SDN...or is it SDN-Enhanced Security

[www.openflowsec.\[org|net|com\]](http://www.openflowsec.[org|net|com])

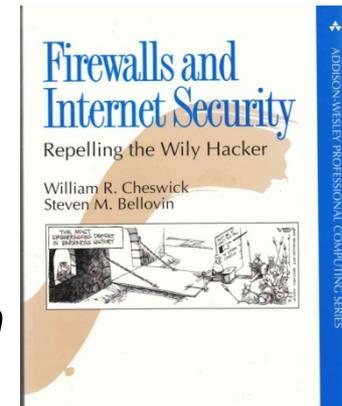
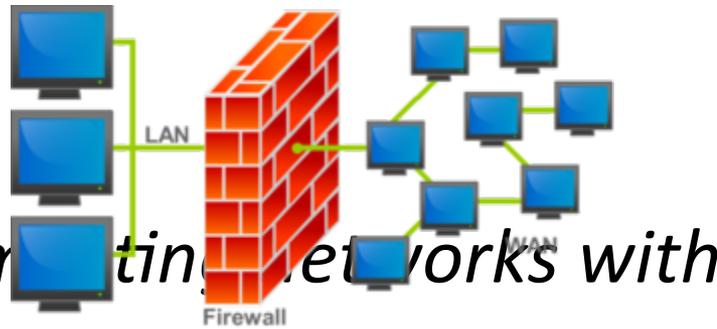
Phillip Porras (phillip.porras@sri.com)

Computer Science Laboratory

SRI International

Classic network security

- '91 – Weismann – Blacker
- '93 – First NDSS Symposium - *The goal of this workshop is to bring together individuals who have built, are building, or will soon build software and hardware concerned with the provision of network or distributed system security services...*
- '94 –
- '97 – *instrumenting networks with sensors*



Classic network perimeter defense

- Provide a well-defined security policy instantiated for a target topology
- Vet both policy and network for compliance
- Deploy policy enforcement consistently across the network
- Test and monitor the network for violations
- Those elements of the network that can alter the security policy must be trusted, and these events must be audited

Challenging the Perimeter Mentality

Challenge 1: Networks are not static, and neither should policy

Challenge 2: Policy is distributed (gateways, internal routers, domain layer). What is the emergent Enterprise policy?

Challenge 3: Virtualization

We don't have perimeters
We have accordions

Challenge 4: BYOD

Challenge 5: Clouds and SaaS – distributing data, computation and services

Network Threat Analysis

- Lots of past work

Malicious Packet Stream ←

Policy Violations ←

A Network Wide Anomaly ←

Diagnosing Infected Host ←

Remote Shell / C&C
Detection ←

Floods / Service Denials ←

Malicious Object /
Logic Injection ←

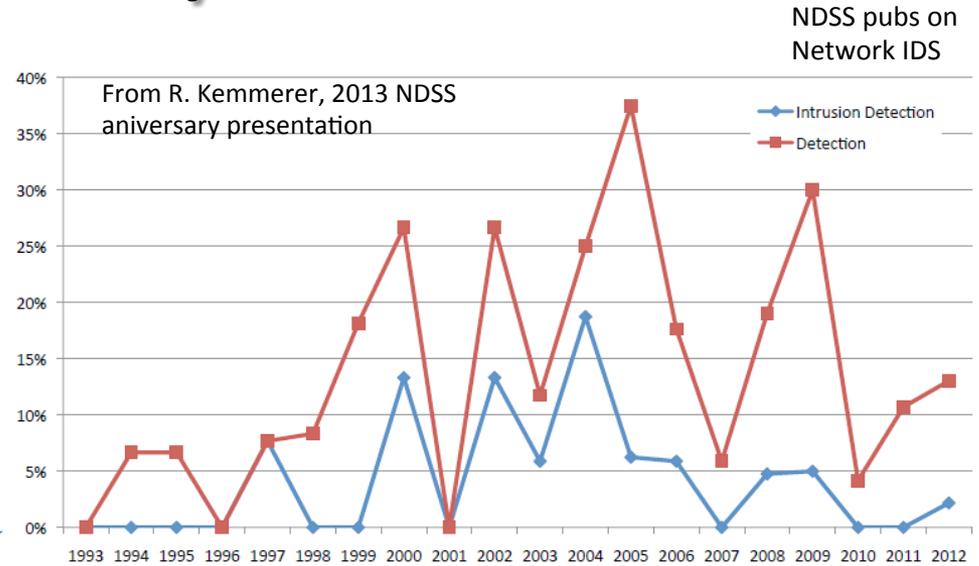
Suspicious
Behavior
Deviations

Network
Reconnaissance

Threat
Reputation

Stepping Stone
Tunneling

Correlation of
all of the above



Challenging our notion of mitigation

	What we do	What we'd like to do....
Malicious Packet Stream	Drop	Block Insider to Malicious Source Redirect Malicious Source to Honeynet
Policy Violations	Drop	Redirect User to a Notification Server
Network Wide Anomaly	Drop	Selective Filtering or Reprovision assets
Infected Host	Drop	Quarantine
Floods and Service Denials	Drop	Block, Migrate Mission Critical services, Redirect
Malicious Object and Logic injection	Drop	Redirect into Sandnet
Remote Shell or C&C	Drop	Redirect Outbound and Inbound to separate Reverse Engineering Services, Set Network Wide Triggers
Suspicious Behavioral Deviations	Drop	Dynamic quota adjustment, fishbowl and reprovision new
Network Reconnaissance	Drop	Proactively redirect
Threat Reputation	Drop	Selectively limit network privileges
Stepping Stone Tunneling	Drop	Selective interruption to validate that tunnel exists

Challenging out ACL Mindset

+20 years of Network Security Policy

Based on the principles of ACLs/Circuits/Application FW, which filter traffic at egress positions based on static packet-layer policies or coarse-grained flow policies.

But an ACL is NOT A SUICIDE PACT

(modern cyber-espionage has taught us its not just about keeping out the bad...)

Network Privilege Management

The right of a device to communicate over the network *is conditionally granted and dynamically revocable*

- Privilege to establish or receive a connection that is otherwise allowed by an ACL, must be revoked based a range of dynamically computed conditions

SDN Security

Challenges

- Flow rules == policy :: what was it 5mins ago? what will it be in 5mins?
- Security must not depend on the absence of SDN App vulnerabilities or complex interactions among apps
- Policy enforcement across switches must be synchronized and consistent across the network
- Everything is a TARGET for attack: the SDN stack (all layers) are a direct target of interest for adversaries



Are solving these challenges a prerequisite for adoption?
well, in sensitive computing environments...Yes

SDN Security

Opportunities

- Embrace Policy Dynamism: We may reconcile how to compute (least privilege preserving) network policies based on many dynamic factors
- Data-plane layer threat mitigation –SDNs enable us to lower the cost of integrating complex threat mitigation logic
- WAN/Internet Security: explore new concepts in WAN security



Are new SDNSec Apps a second adoption prerequisite?
well, in sensitive computing environments...Yes

Control Layer Security

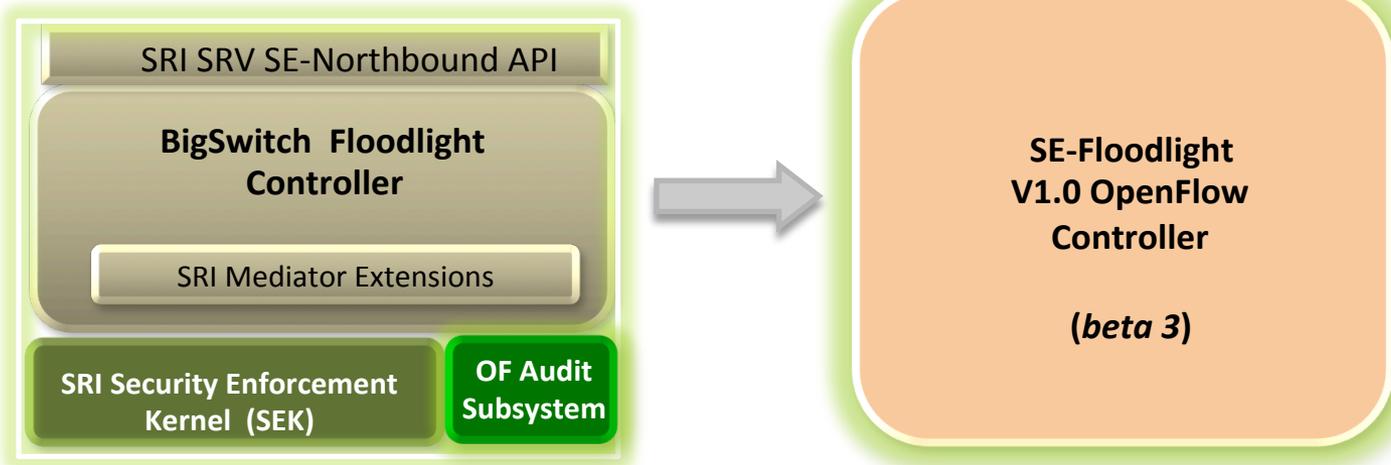
“Toward an adoptable SDN Stack for sensitive network environments”

SE-Floodlight?

An implementation of a security mediation service in an openflow Stack

- Recognizes and resolve conflicts between the existing security policy and candidate flow rules
- Allows the dynamism of OpenFlow applications to produce optimal flow routing decision ... as long as there is no conflict
- Empowers *OpenFlow security applications* and operators to dynamically assert extensions to the security policy when new threats are perceived

Free Download: SE-Floodlight



www.openflowsec.org

Inline flow rule
conflict detection

Role-based Authorization
(conflict resolution)

Digital Authentication
of FlowRule Source

Least Privilege
(OF Apps)

Security
Audit

Distributed policy
synchronization

Building SDN Security Tech Suite

www.openflowsec.org

SE-Floodlight is an OpenFlow Security Mediation Service for Network Policy Enforcement

SE-Floodlight

Server-side SE-Northbound API
SRI Mediator Extension
SRI Security Enforcement
Kernel

OpenFlow Security Actuator

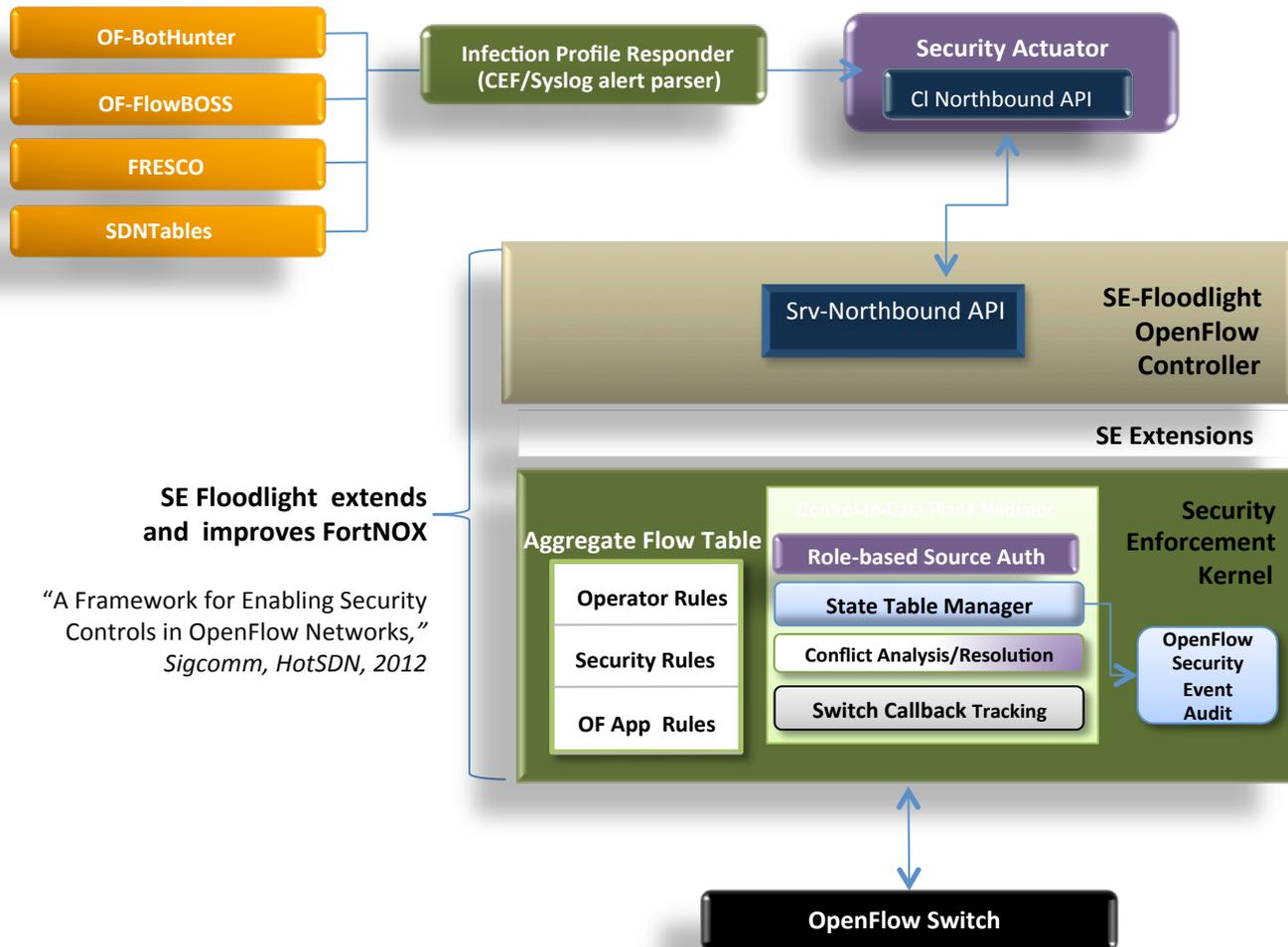
SRI Security Response Director

OF-BotHunter

Pre-requisites: package 1 and 2
BotHunter v1.7.2
Infection Profile Responder

FlowBOSS

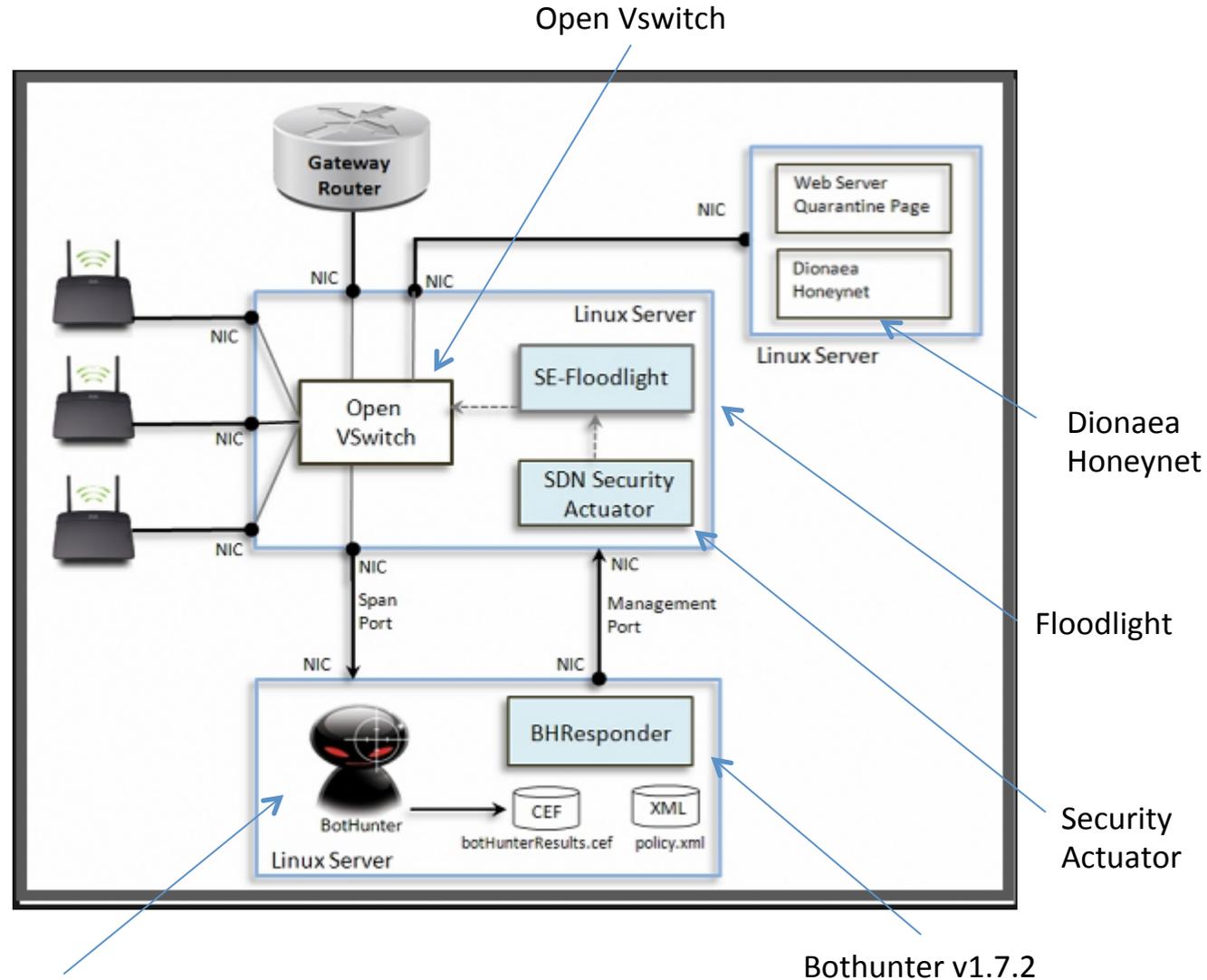
Network Privilege Management
(Jan 2014)



SE Floodlight extends and improves FortNOX

"A Framework for Enabling Security Controls in OpenFlow Networks,"
Sigcomm, HotSDN, 2012

A Self Defending Wireless Network



BotHunter v1.7.2

Bothunter v1.7.2

Avant-Guard

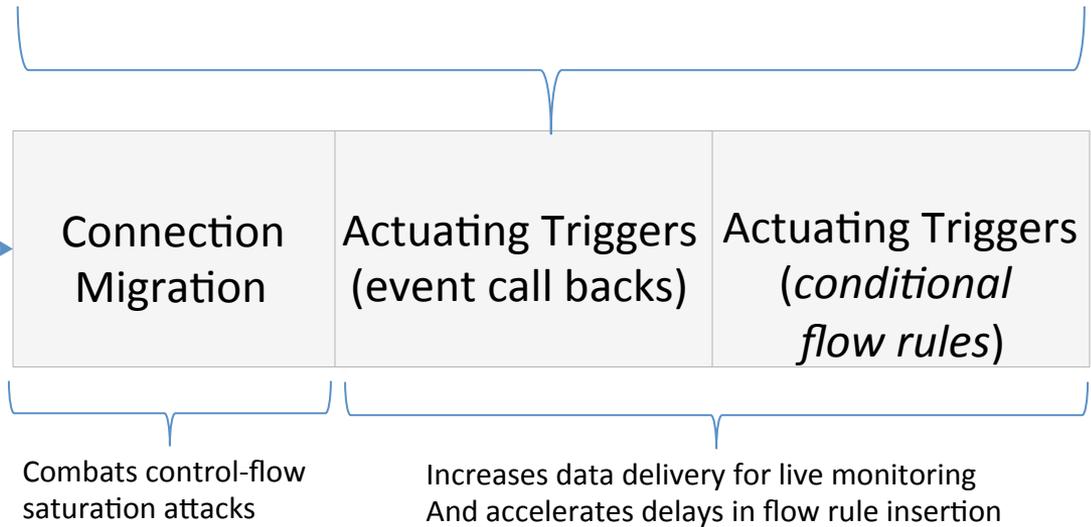
SE Floodlight +
Connection Migration

Southbound API
with Avant-Guard extensions

Open VSwitch
Reference Implementation

Avant-Guard

S. Shin, V. Yegneswaran, P.A. Porras, and G. Gu, "**AVANT-GUARD: Scalable and Vigilant Switch Flow Management in Software-Defined Networks**", in *Proceedings of the 20th ACM Conference on Computer and Communications Security (CCS)*, Berlin, Germany, 2013



FlowBoss



available January 2014

An embodiment of Network Privilege Management - makes intelligent contextual flow policy enforcement easy to express and enforce on any OpenFlow network

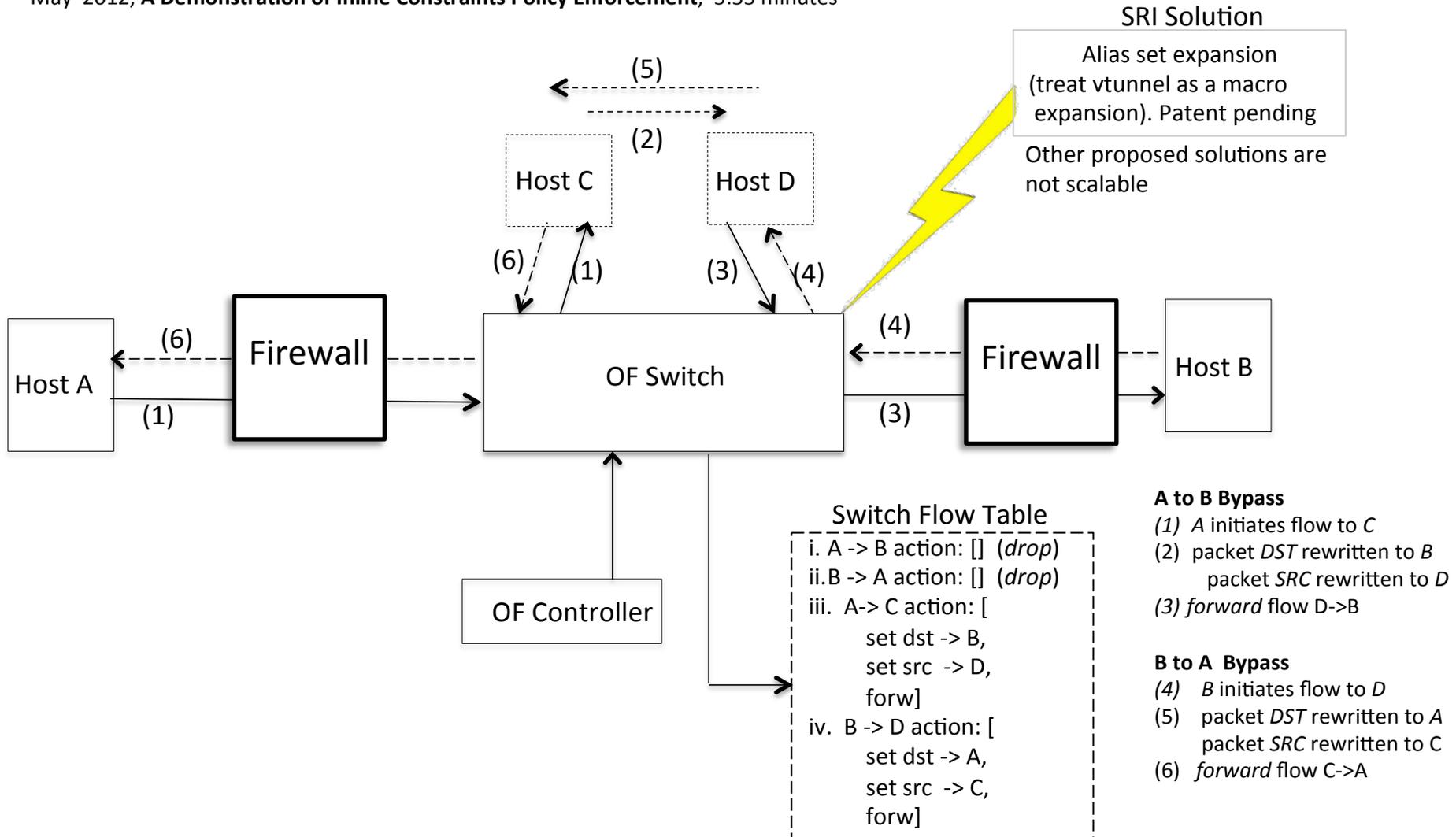
Specified a wide range of unique network security policies prevent

- Unauthorized data exfiltration
- Handling policies for IP reputation violations
- Specification-based policies that limit a network to "approved flows only"
- Modal policies - evening and weekend flow policies,
- Geo-aware flow policies
- Conditional flow policies that depend on current overall network statistics
- Policies for a wide range of malicious or prohibited network flows.

Example SDN Challenge

http://www.openflowsec.org/OpenFlow_Security/Demo_Vids.html

May 2012, A Demonstration of Inline Constraints Policy Enforcement, 5:55 minutes



VeriFlow: Verifying Network-Wide Invariants in Real Time

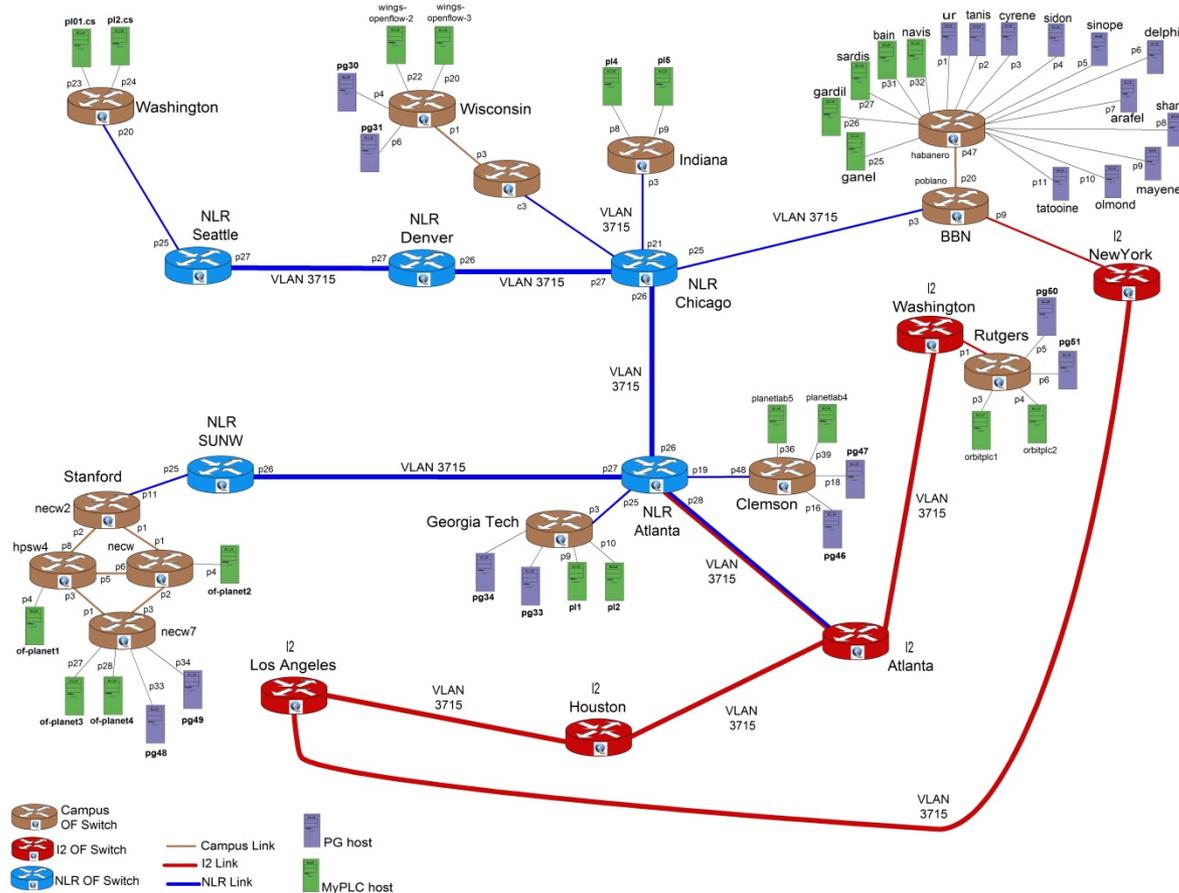
Ahmed Khurshid, Xuan Zou, Wenxuan Zhou,
Matthew Caesar, P. Brighten Godfrey
University of Illinois at Urbana-Champaign (UIUC)

April 3, 2013

NSDI 2013

10th USENIX Symposium on Networked Systems Design and Implementation

Challenges in Network Debugging



Complex interactions

Misconfigurations

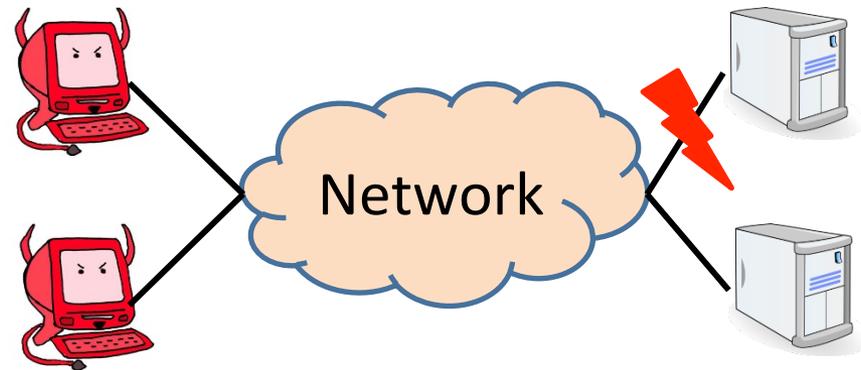
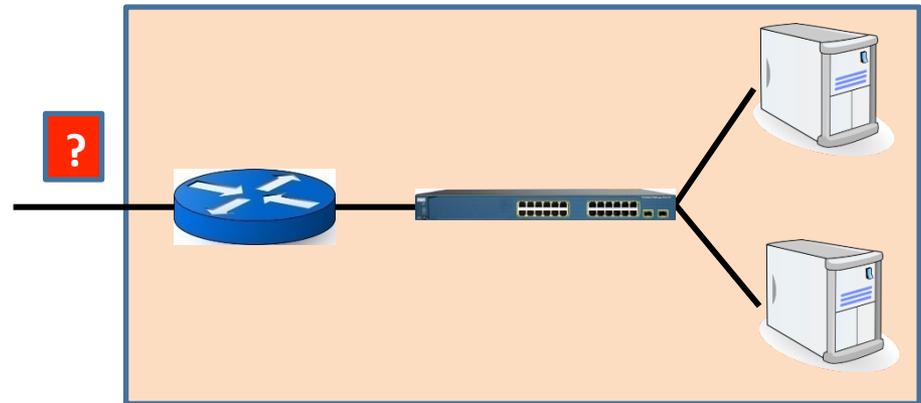
Unforeseen bugs

Difficult to test the entire network state space before deployment

http://groups.geni.net/geni/chrome/site/thumbnails/wiki/TangoGENI/OF-VLAN3715_1000.jpg

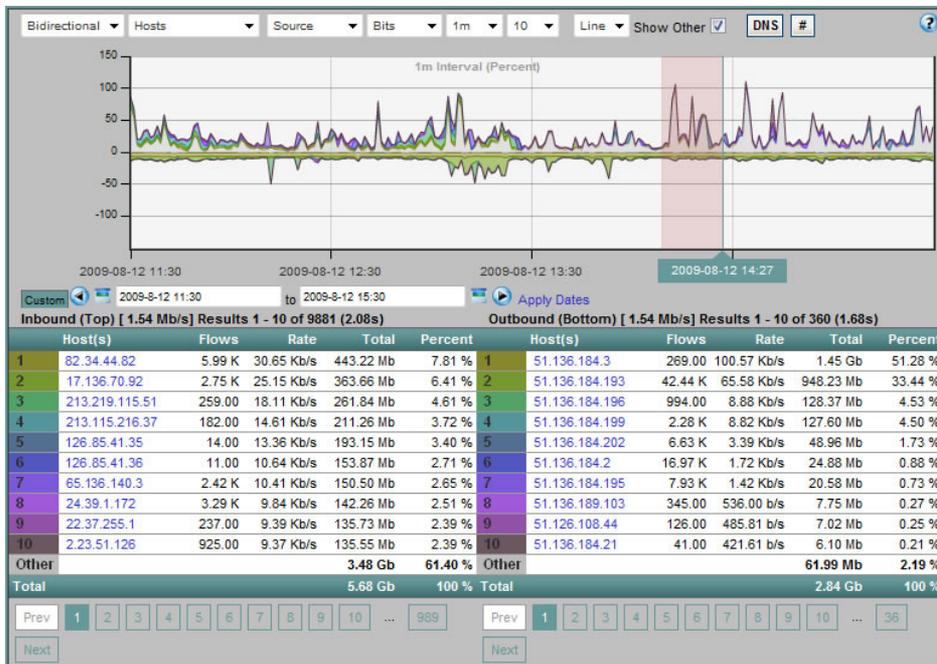
Effects of Network Errors

- Allow unauthorized packets to enter a secured zone in a network
- Make services and the infrastructure prone to attacks
- Make critical services unavailable
- Affect network performance



Network Debugging Techniques

Traffic/Flow Monitoring



Software using Cisco NetFlow

<http://snmp.co.uk/scrutinizer/>

Configuration Verification

```
hostname bgpdA
password zebra
!
router bgp 8000
  bgp router-id 10.1.4.2

! for the link between A and B
  neighbor 10.1.2.3 remote-as 8000
  neighbor 10.1.2.3 update-source lo0

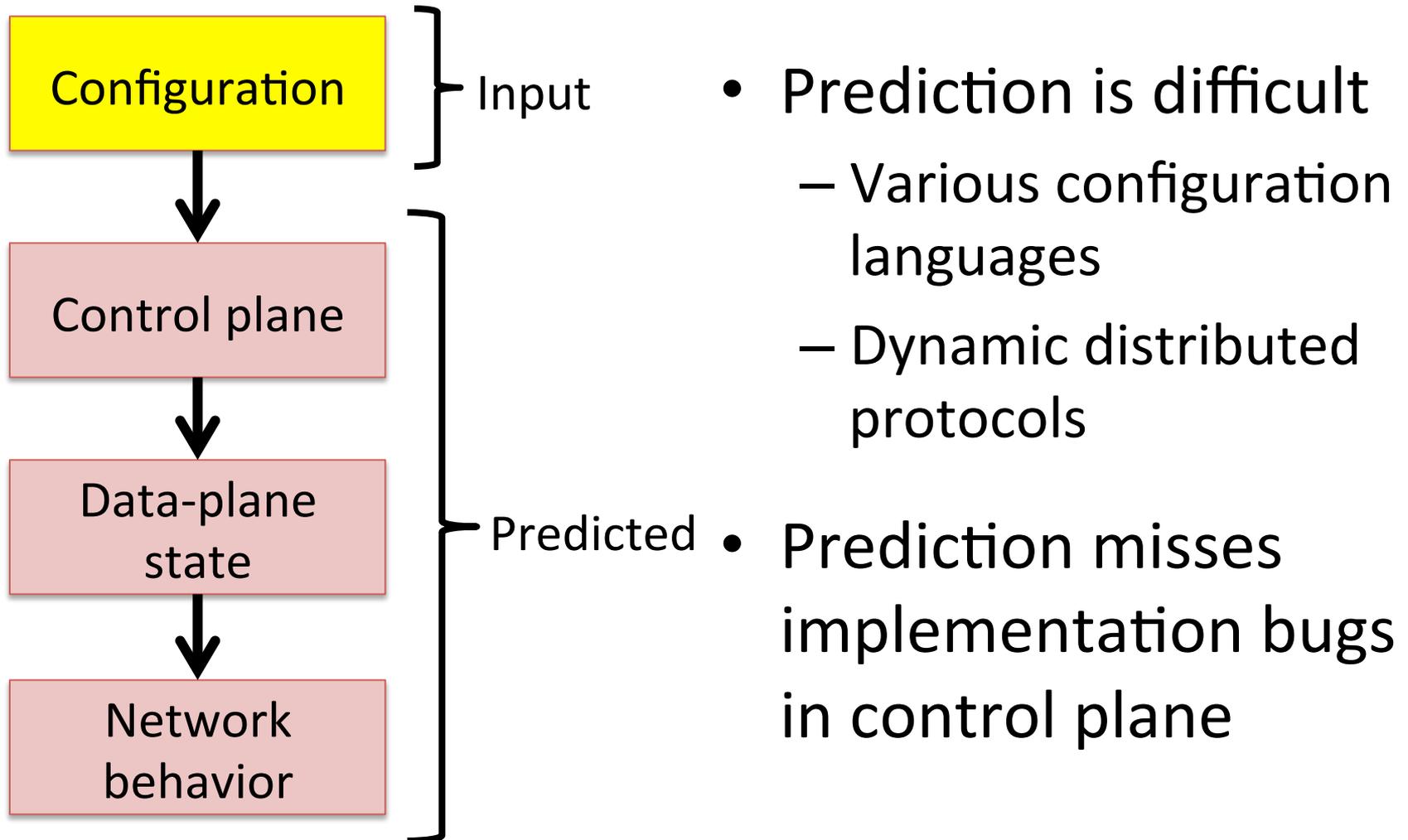
network 10.0.0.0/7

! for the link between A and C
  neighbor 10.1.3.3 remote-as 7000
  neighbor 10.1.3.3 ebgp-multi-hop
  neighbor 10.1.3.3 next-hop-self
  neighbor 10.1.3.3 route-map PP out

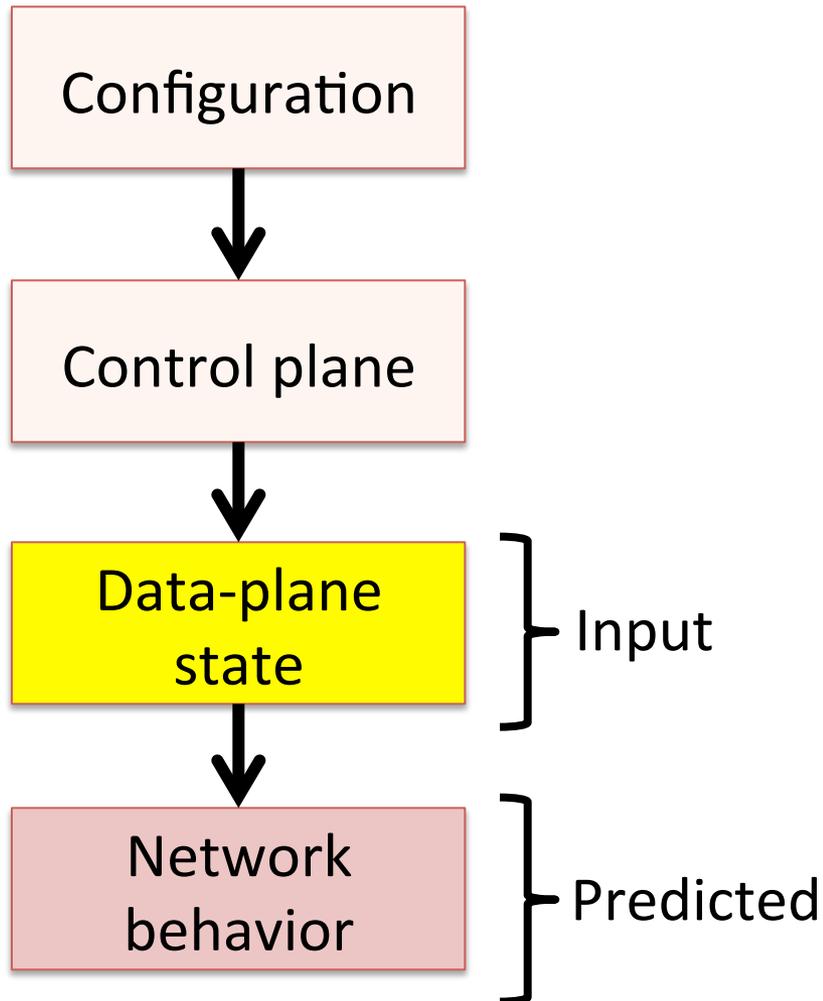
! for link between A and D
  neighbor 10.1.4.3 remote-as 6000
  neighbor 10.1.4.3 ebgp-multi-hop
  neighbor 10.1.4.3 next-hop-self
  neighbor 10.1.4.3 route-map TagD in

! route update filtering
  ip community-list 1 permit 8000:1000
!
```

Limitations of Configuration Verification



Our Approach: Data-plane Verification



- Less prediction
- Closer to actual network behavior
- Unified analysis for multiple control-plane protocols
- Can catch control-plane implementation bugs

Data Plane Verification in Action

- FlowChecker [[Al-Shaer et al., SafeConfig 2010](#)]
 - Uses BDD-based model checker
- Anteater [[Mai et al., SIGCOMM 2011](#)]
 - Uses SAT-based model checking
 - Revealed 23 real bugs in the UIUC campus network
- Header Space Analysis [[Kazemian et al., NSDI 2012](#)]
 - Uses set-based custom algorithm
 - Found multiple loops in the Stanford backbone network

Find problems
after they occur
and (potentially)
cause damage

Running time: Several seconds to a few hours

Can we run verification in real time?

Checking network-wide invariants in real time as the network evolves

Need to verify new updates at high speeds

Block dangerous changes

Provide immediate warning

Challenges in Real-Time Verification

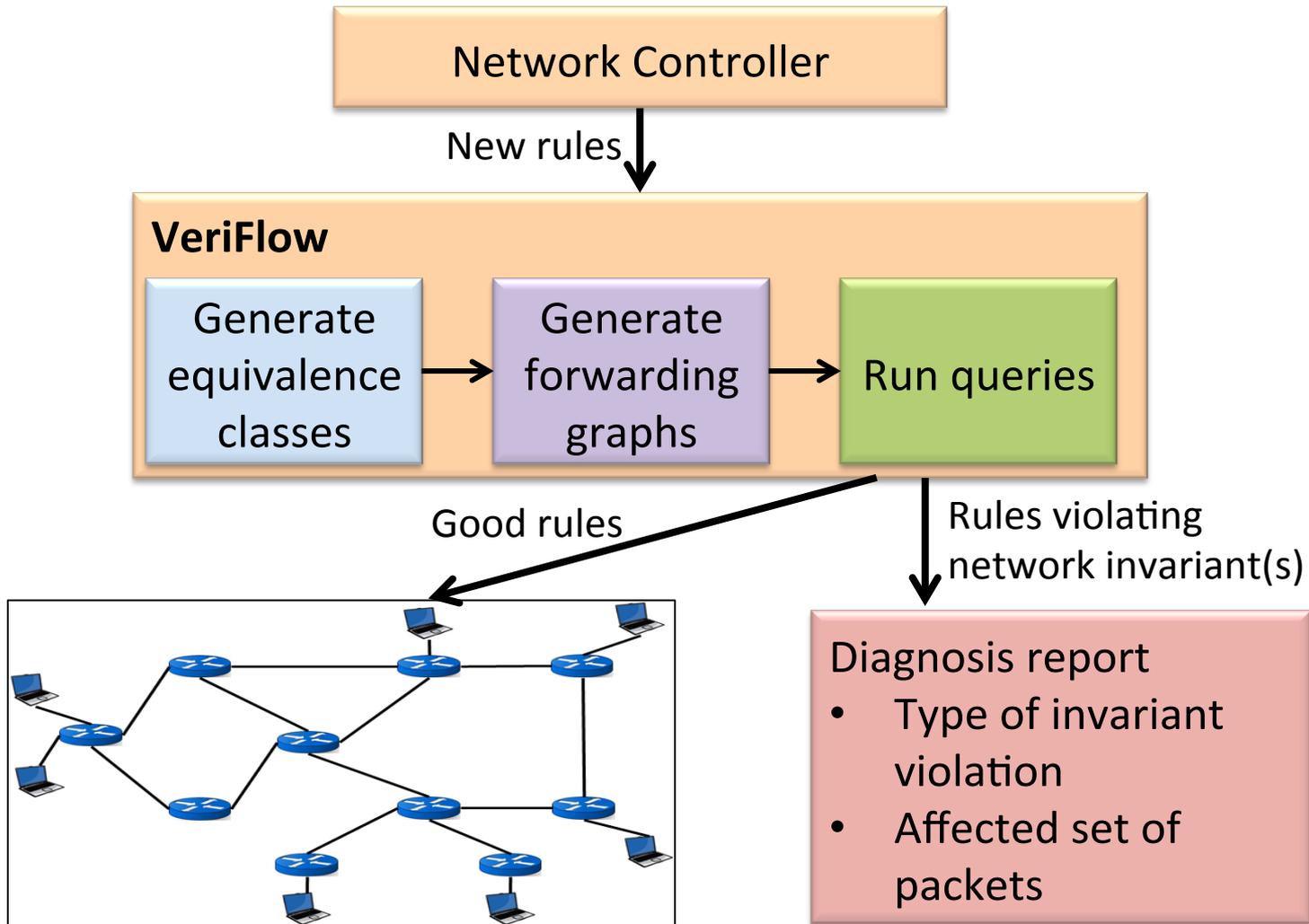
- Challenge 1: Obtaining real-time view of network
 - Solution: Utilize the **centralized** data-plane view available in an **SDN (Software-Defined Network)**
- Challenge 2: Verification speed
 - Solution: Off-the-shelf techniques?

No, too slow!

Our Tool: VeriFlow

- VeriFlow checks network-wide invariants in **real time** using data-plane state
 - Absence of routing loops and black holes, access control violations, etc.
- VeriFlow functions by
 - Monitoring **dynamic changes** in the network
 - Constructing a **model** of the **network behavior**
 - Using **custom algorithms** to automatically derive whether the network contains errors

VeriFlow Operation



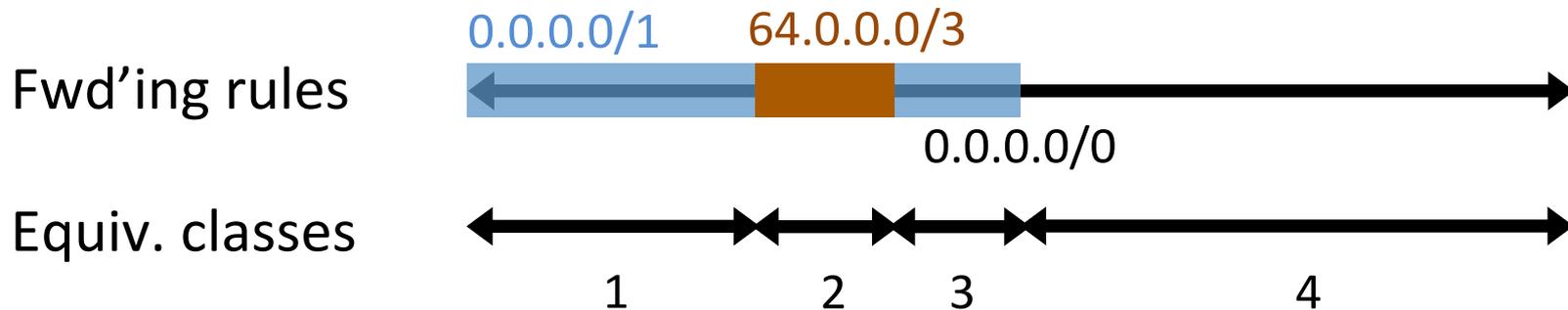
1. Limit the Search Space

VeriFlow

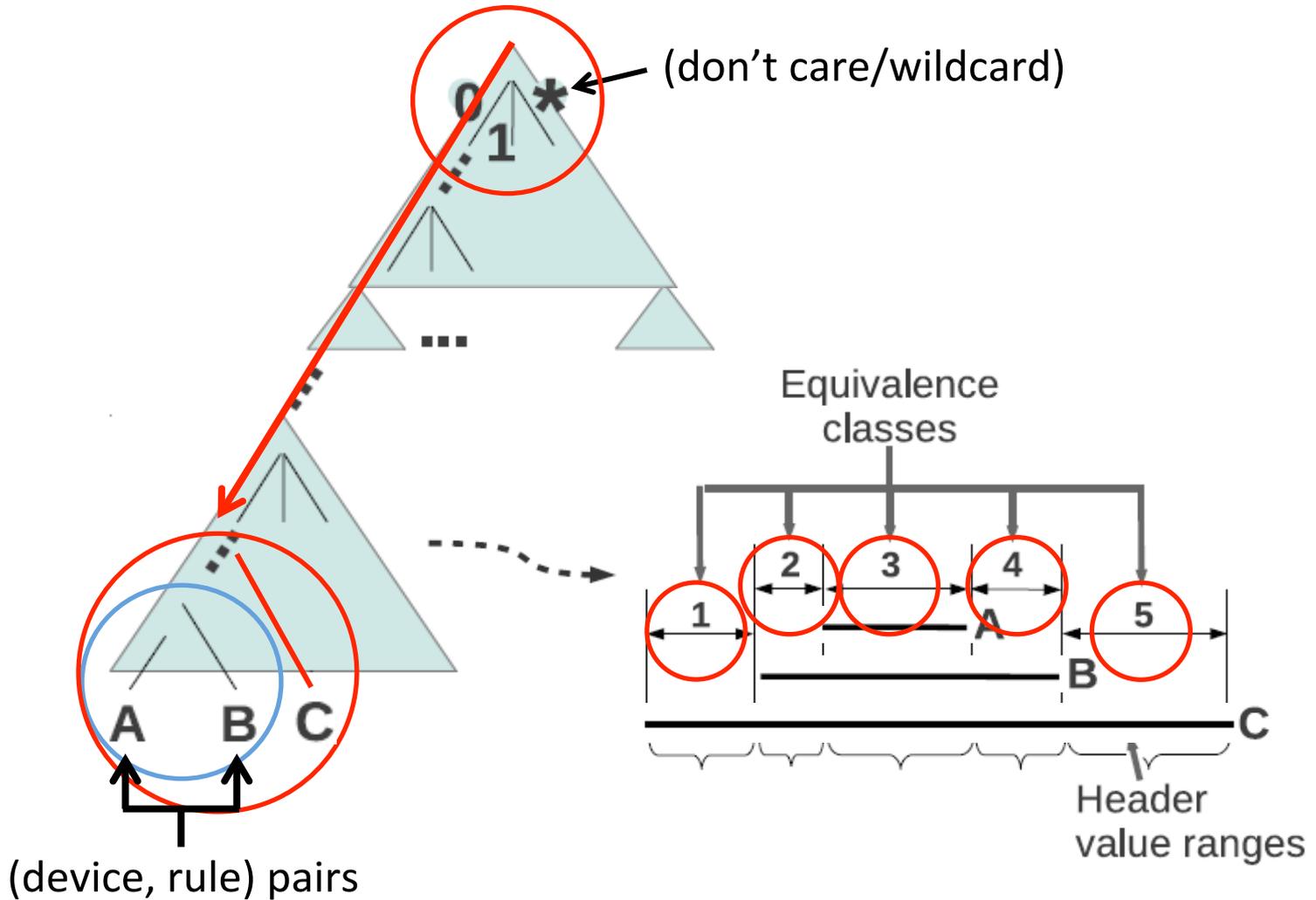
Generate
Equivalence
Classes

Updates

Equivalence class:
Packets experiencing
the same forwarding
actions throughout the
network.

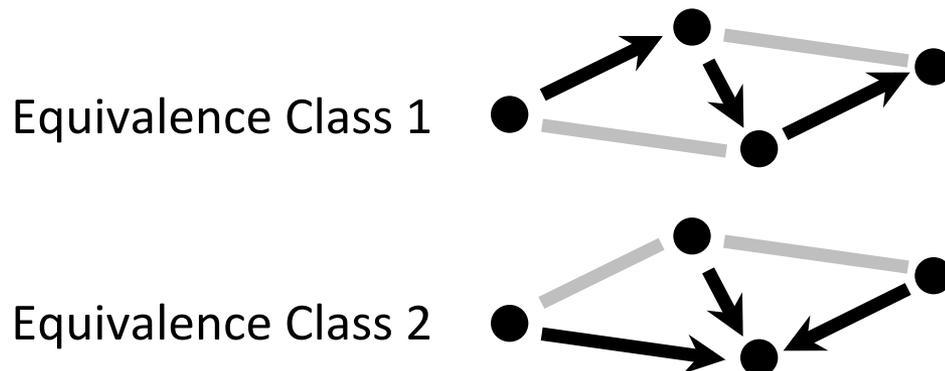
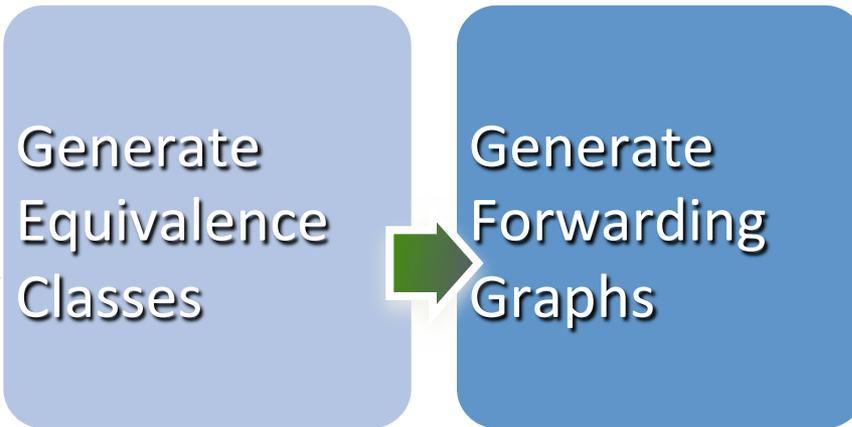


Computing Equivalence Classes



2. Represent Forwarding Behavior

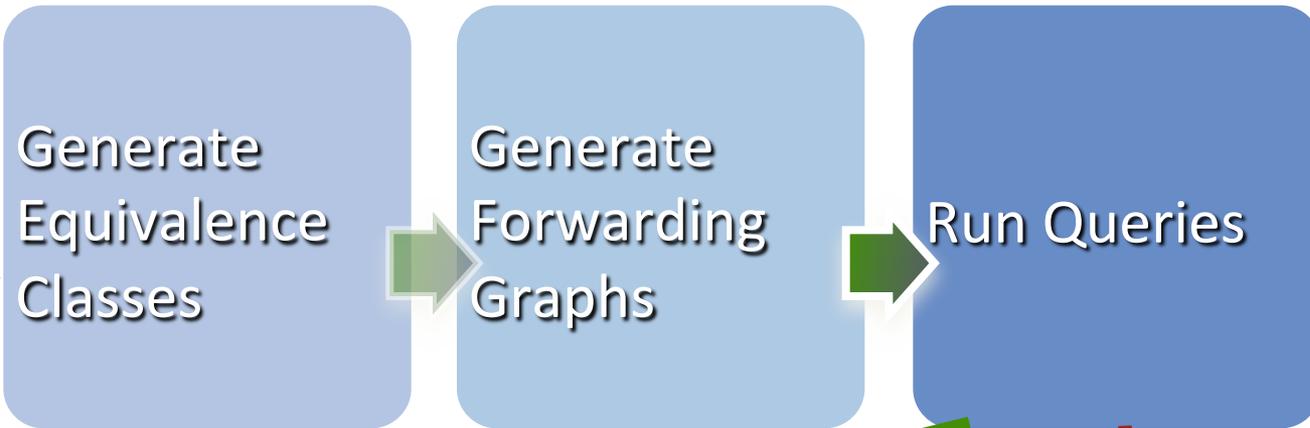
VeriFlow



All the info to answer queries!

3. Run Query to Check Invariants

VeriFlow

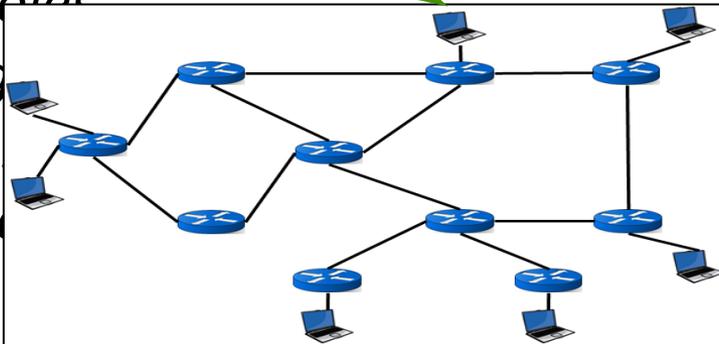


Updates

Good rules

Bad rules

Black holes
Routing
Isolation
Access



Diagnosis report

- Type of invariant violation
- Affected set of packets

API to write custom invariants

- VeriFlow provides a set of functions to write custom query algorithms
 - Gives access to the affected set of equivalence classes and their forwarding graphs
 - Verification becomes a standard graph traversal algorithm
- Can be used to
 - Check forwarding behavior of specific packet sets
 - Verify effects of potential changes

VeriFlow Claims

- VeriFlow achieves real-time verification
 - A layer between SDN controller and network devices
 - Handles multiple packet header fields efficiently
 - Runs queries within hundreds of microseconds
 - Exposes an API for writing custom invariants
- Future work
 - Handling packet transformations efficiently
 - Dealing with multiple controllers

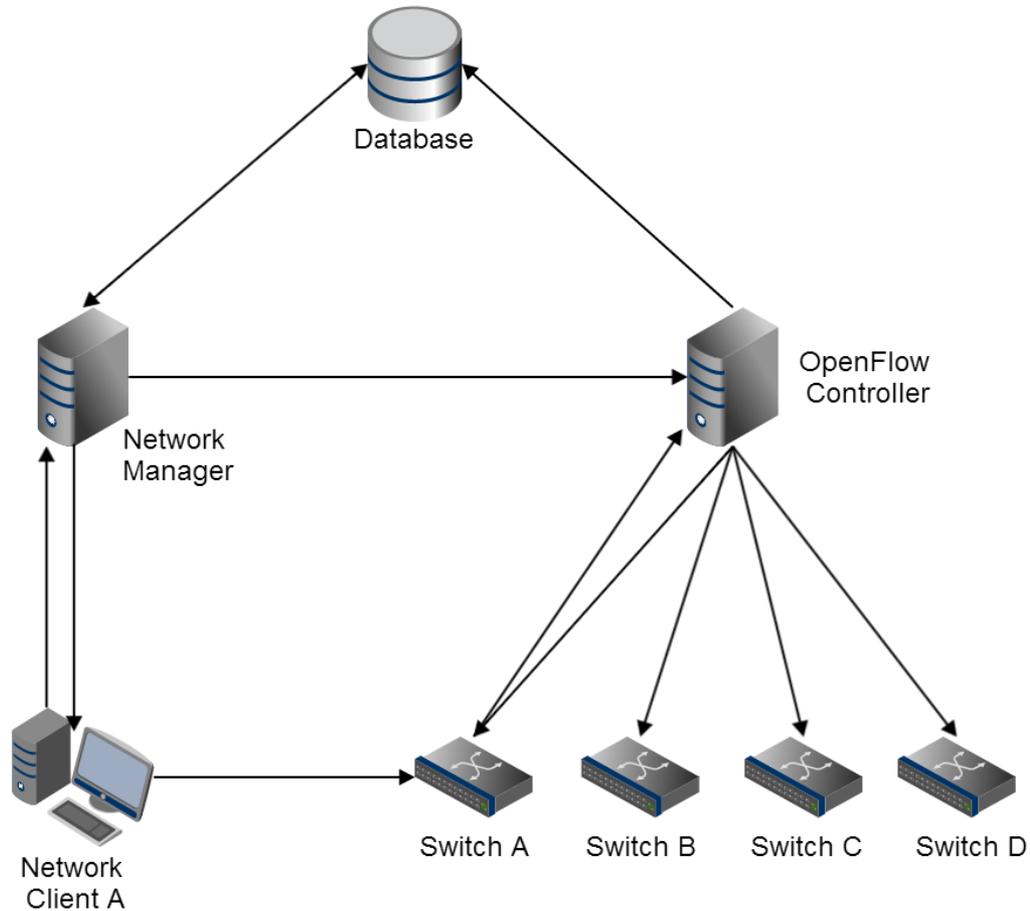
Potential Architecture for Decentralized Interdomain SDN Cooperation. RHC

- This proposed architecture describes a system that enables state updates across domains.
- The control planes within each domain remains autonomous of one another.
- However, enabling coordination among them allows for cooperative network control.

Within Each Domain

- Each domain managed by a Network Management application.
 - Maintains configuration and network state in database
 - Enforces policy and ensures consistency
- Network Manager issues high-level directives to an OpenFlow Controller application
 - OFC app translates directives into flow rules to be installed at each switch so as bring the network into compliance with the directive.

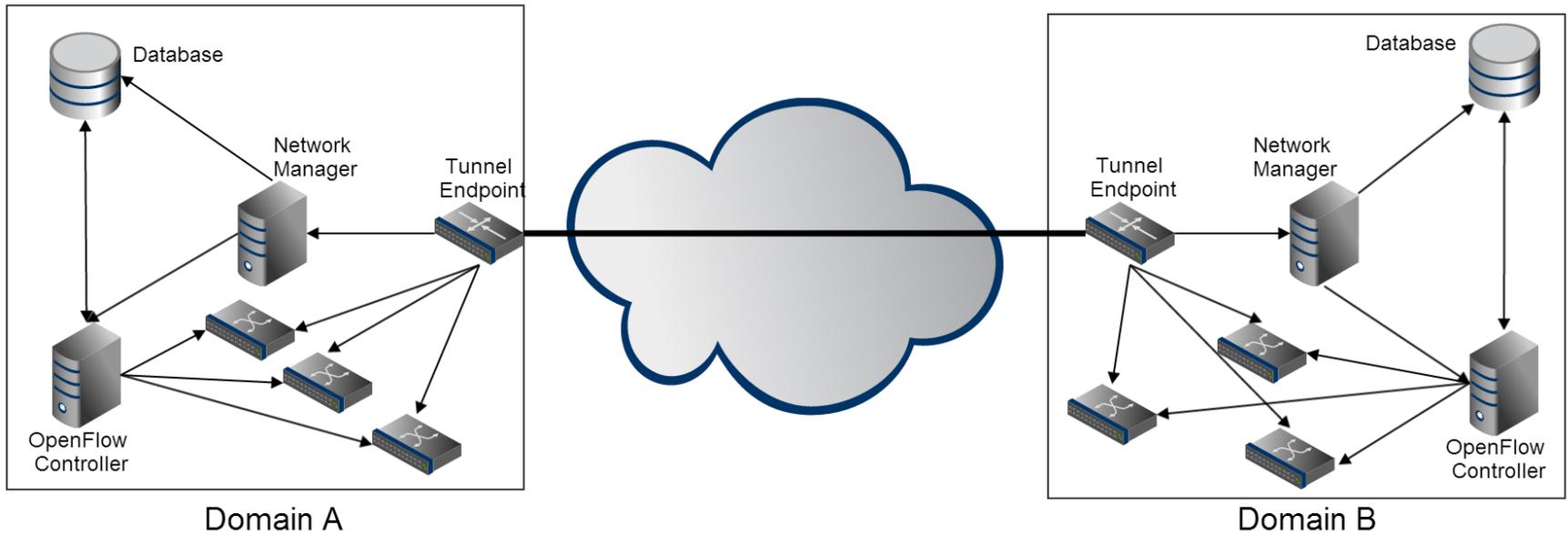
Within Each Domain



Interdomain SDN Cooperation

- Domains are connected via VPN, creating secure overlay
- Network Manager applications share network state information and state updates.
- Each domain's Network Manager retains autonomy, but can use state updates from remote domains to influence control decisions

Interdomain SDN Cooperation



Privacy, Confidentiality and Net Neutrality

- Examples of privacy compromises in networks
- Examples of confidentiality compromises in networks
- Greater control of networks impacts privacy and confidentiality mechanisms
- Demonstrations and proof of related invariants in SDN

SDN Summary

Lack of Security Metrics for SDN

- Infrastructure
- Clients and Servers
- Policies
- Experience

2013 DATA BREACH INVESTIGATIONS REPORT



69% of breaches were spotted by an external party — 9% were spotted by customers.



Social tactics — using email, phone calls and social networks to gain information on individuals — are often ignored, but contributed to 29% of attacks.



76% of network intrusions exploited weak or stolen credentials. Strict policies are required to reduce this easily preventable risk.

Breakout Session Questions

- Can we obtain a scalable real time view of SDN networks that allows for a practical security response? For example, can multidomain and multi-controllers be kept consistent and can we eliminate routing loops?
- Given experience from existing network attacks (example, compromised authorization, denial of service), what defenses does SDN make more effective, and what new defenses does it enable? What key attacks that currently occur need further research in SDN technologies?
- What new forms of attack might occur in an SDN network?
- What safeguards are required to provide privacy in a SDN network?
- How do we safeguard security for interdomain SDN networking?
- Has the SDN software ecosystem including systems, protocols and tools reached the maturity where SDN can be deployed without creating vulnerabilities? Is SDN solid enough to develop robust routing protocols?

Thanks