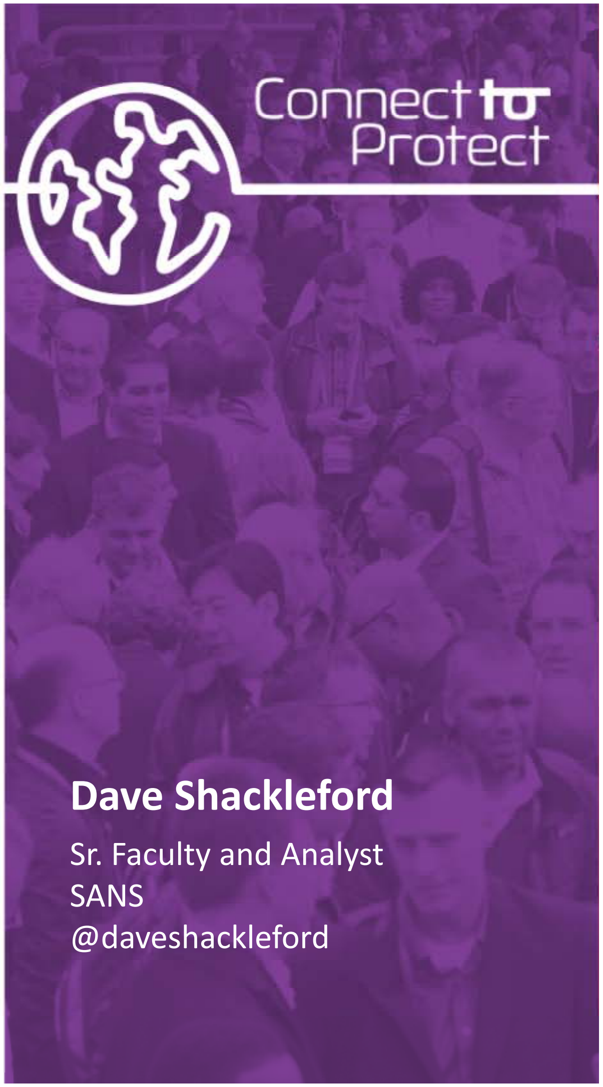
RSAConference2016

San Francisco | February 29 – March 4 | Moscone Center

SESSION ID: CSV-R03

Designing Virtual Network Security Architectures



Connect to Protect

Dave Shackleford

Sr. Faculty and Analyst
SANS
@daveshackleford

#RSAC

Introduction

- Much has been said about virtual networking and software-defined networking (SDN) in the past several years
  - Most of the conversation has been focused on operations
- There are major impacts to network security, however
- Major SDN tools and vendor products have emerged
- Architectural frameworks for virtual and software networking have emerged, as well
  - But where does security fit into all this?

NFV to SDN

- Network Functions Virtualization (NFV) decouples network functions from dedicated hardware devices
  - Network services (routers, firewalls, load balancers , etc.) can now be hosted on virtual machines
- SDN is an architectural model that offers network virtualization and programmability
  - SDN abstracts the network control plane from the data plane
  - Some definitions are less focused on decoupling the planes, and more on APIs and integration

Example Projects/Products

- OpenFlow is a specification for handling and processing network traffic flows in a software-defined manner
- OpenDaylight is a full implementation of SDN governed by the Linux Foundation
  - Includes a full-featured, open-source controller
  - Also supports OpenFlow and other SDN specifications
- Openstack Neutron is the SDN component of Openstack
- Commercial options from VMware (NSX) and Big Switch

Example Frameworks/Standards

- TOSCA - Topology and Orchestration Specification for Cloud Applications
- YANG: Modeling language for configuration and state data with Netconf
  - Netconf provides mechanisms to install, manipulate, and delete the configuration of network devices
- REST APIs are also common

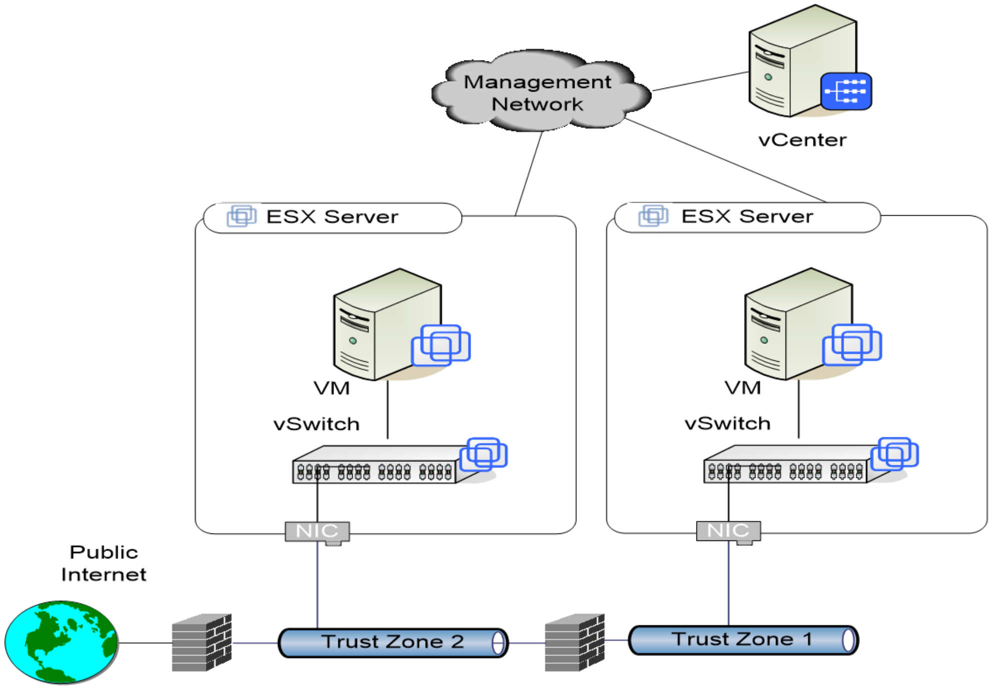
New Architectural Models

Virtual Networking

- The progression of virtual networking looks a bit like this:
  - Virtual switches (basic)
  - Virtual switches (distributed)
  - Parity with physical switches (Cisco Nexus 1000v, Open vSwitch)
  - NFV
  - SDN
- Architecture models have shifted, as well

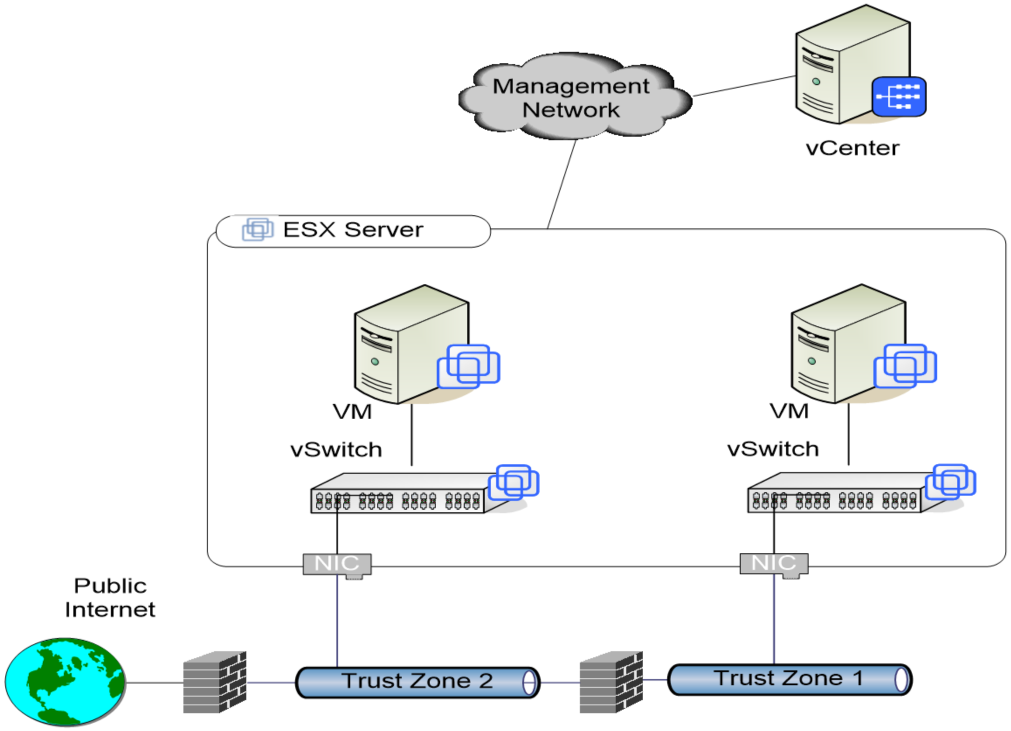
Old School: Separate Physical Trust Zones

- Systems are virtualized
- Network connections are still physically distinct
- Provides the most flexibility with existing network security tools



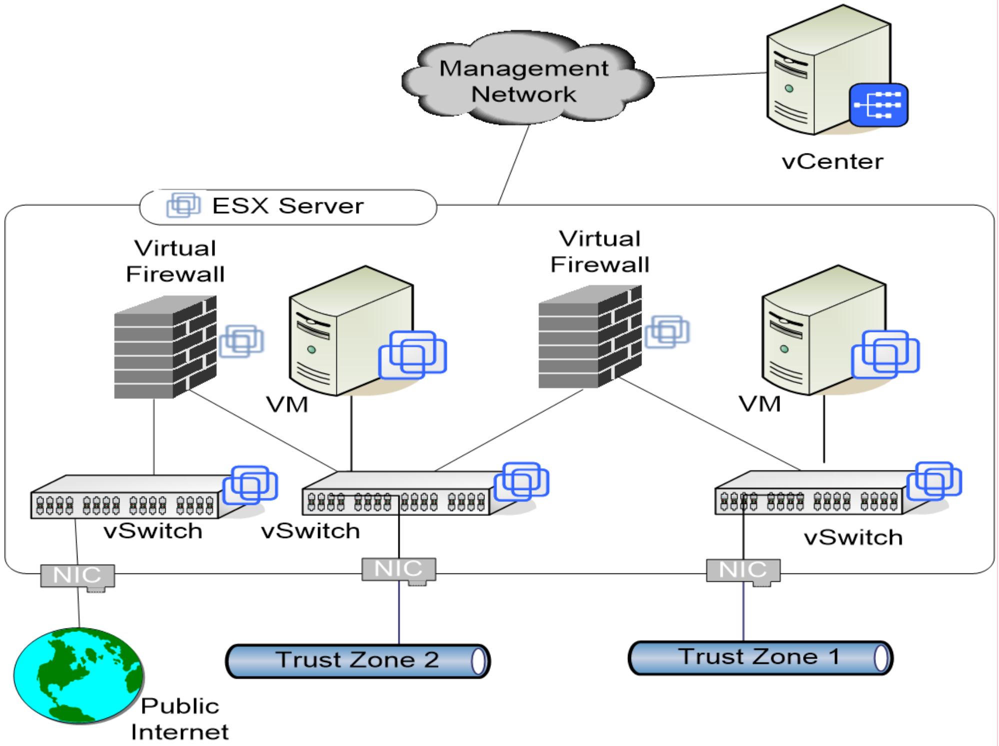
Consolidation: Virtually Separate Trust Zones

- Systems are virtualized
- Zones can be consolidated into one or more hypervisor hosts
- Network security devices and functions are still physically separate



More Consolidation…on to SDN?

- All systems are virtualized
  - Switches
  - Systems
  - Security devices and functions

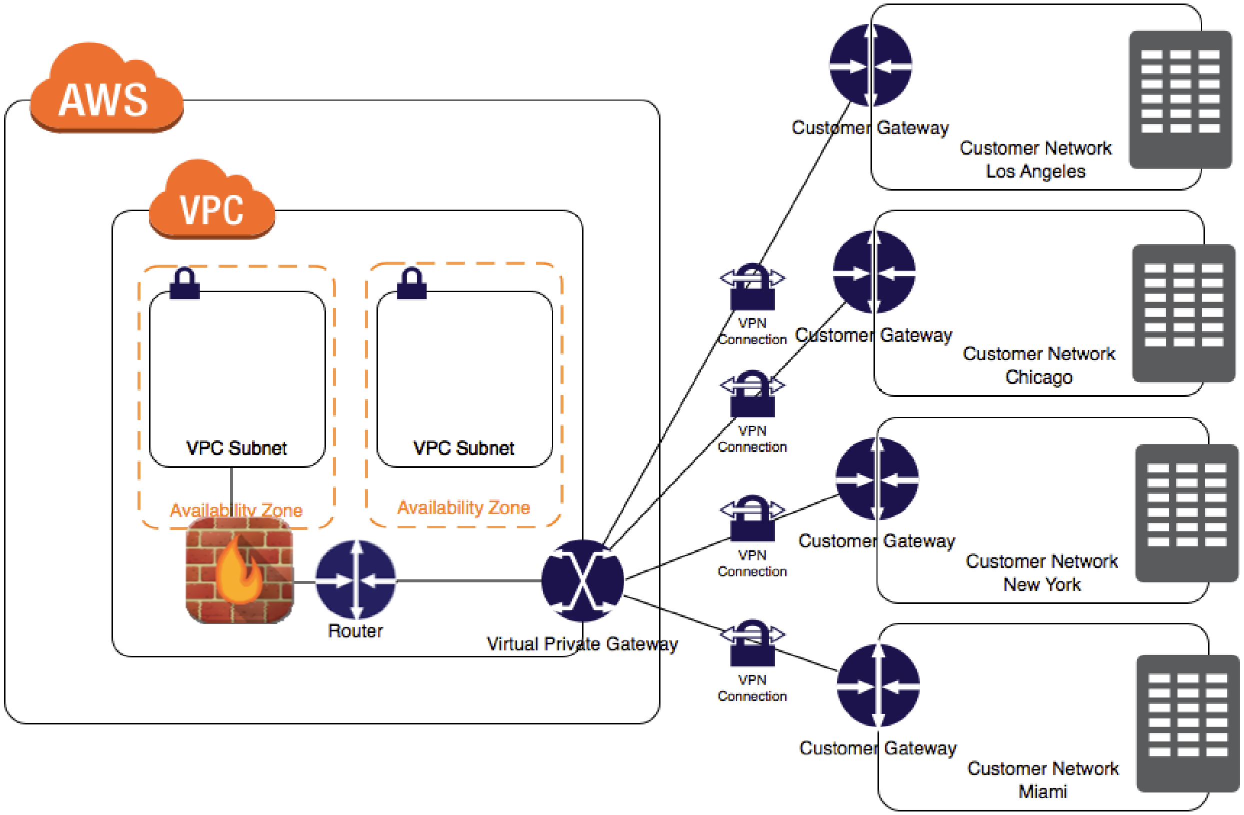


Progress Today

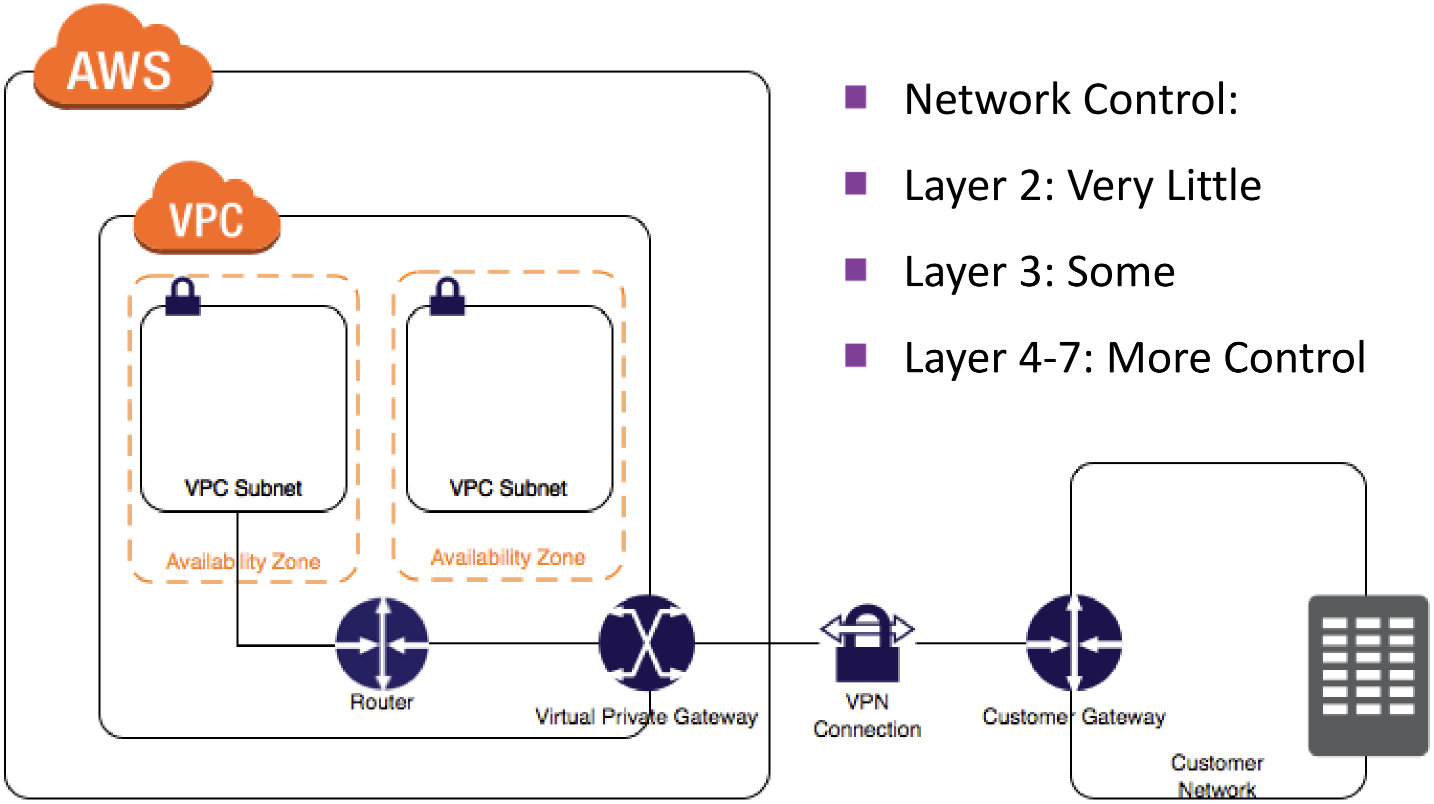
- Many network security controls have been successfully virtualized
  - Firewalls, Switches (traffic copy and flow export, ACLs, etc.), Routers, IDS/IPS, Load balancers, WAFs
- These all leverage the hypervisor in use, and still consolidate data and control planes (relative to function)
- Most public cloud consumers don’t have true SDN available…yet.

Hybrid Cloud Architecture

- Most hybrid cloud design uses:
  - NFV
  - Virtual appliances
  - VPN connections



Hybrid Cloud Architecture



- Network Control:
- Layer 2: Very Little
- Layer 3: Some
- Layer 4-7: More Control

Moving from physical -> virtual

- Evaluation criteria to consider:
  - Cost
  - Vendor viability
  - Native integration with hypervisor platforms
  - Management capabilities
  - Performance impacts and scalability
  - Architecture flexibility
  - Virtualization-specific features

Benefits and Drawbacks: NFV (and SDN)

PROs

- Rapid configuration control implementation
- New central control point for control plane aspects of enterprise networking
- Traffic shaping and QoS may be more flexible, with improved DoS and DDoS detection/prevention

CONs

- A new weak point to administer and audit
- Need to define policies and encryption controls for NFV/SDN
- Potential false positives for log management and SIEM in control traffic (and new log types)
- Availability!!

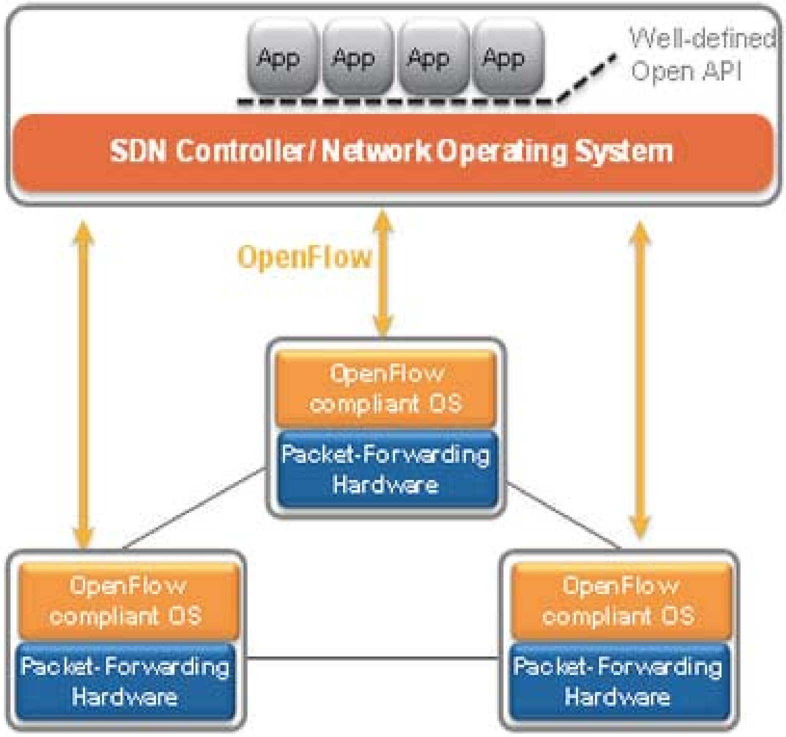
SDN: Reality versus Hype

SDN: Reality?

- First things first: SDN is real, it’s growing, and security needs to adapt.
- That said: SDN will not replace everything. Not soon, anyway.
- Abstraction of network functions to a virtualized model is becoming more mature all the time
  - This includes technology like VxLAN
- SDN protocols, frameworks, and controllers are maturing, too
- However, it’s not all “real” for many organizations yet

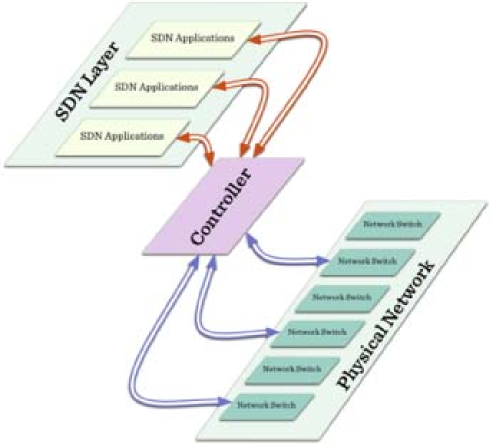
Reality: API-driven Networking

- The use of APIs to configure, control, and monitor networks exists and will grow
- Examples include OpenFlow, Netconf, OpenStack, etc.
- Some controllers are showing promise, too


App
App
App
App
Well-defined Open API
SDN Controller / Network Operating System
OpenFlow
OpenFlow compliant OS
Packet-Forwarding Hardware
OpenFlow compliant OS
Packet-Forwarding Hardware
OpenFlow compliant OS
Packet-Forwarding Hardware

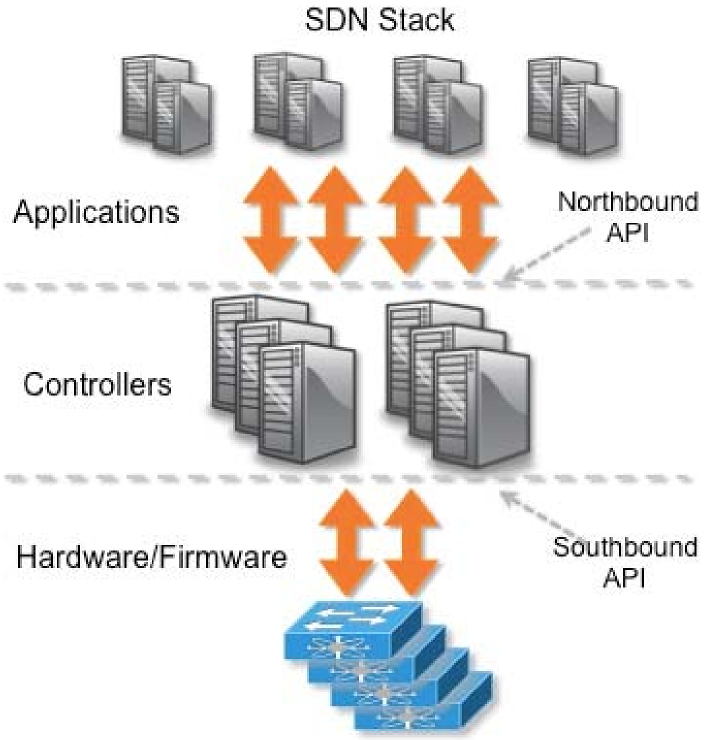
Hype: Programming it all…NOW.

- Shifting from hybrid physical+virtual networking functions and tools to a pure SDN architecture is highly impractical today for many
- More likely?
  - Some policy application
  - Some simple configuration
  - Monitoring
- The APIs are there…but turning the ship takes time.



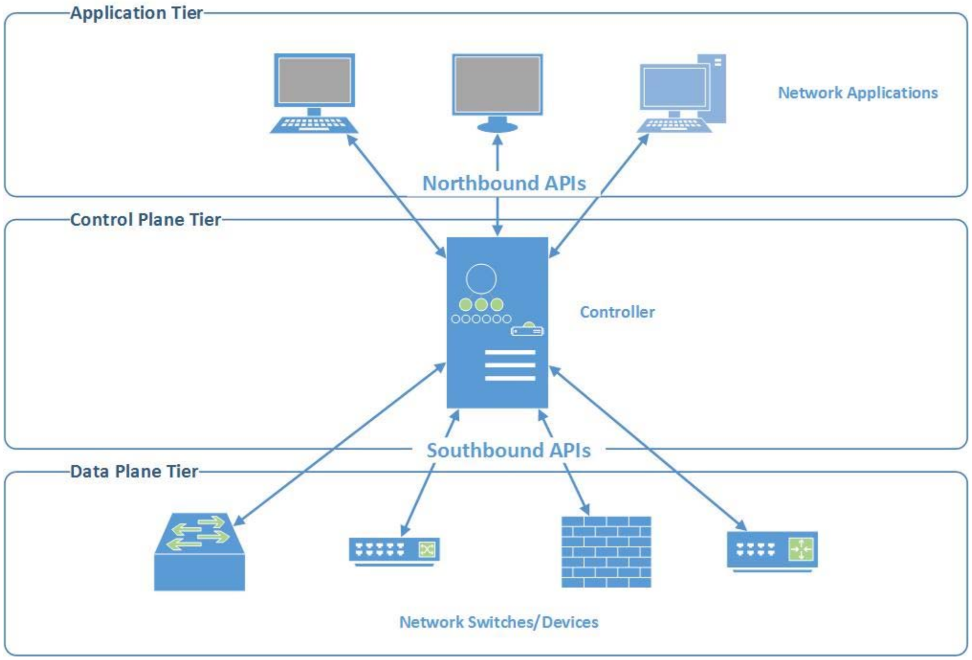
SDN Architecture

- The SDN architectural model leverages both northbound and southbound APIs
  - Northbound: Management and reporting tools
  - Southbound: Control, configuration, and monitoring commands



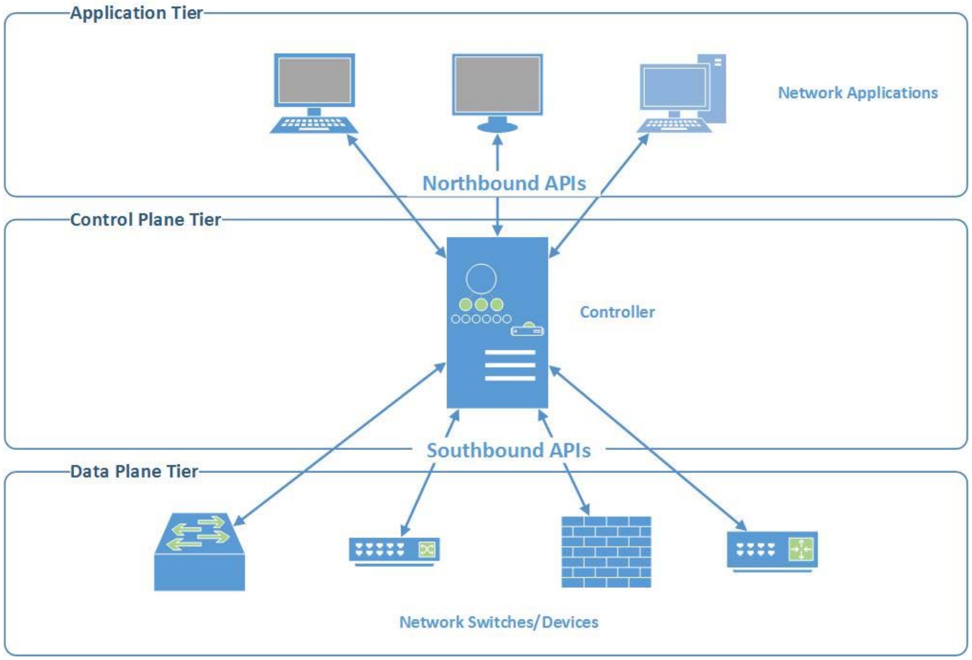
SDN: Switches

- Switches accept commands from SDN controllers
  - This is the data plane tier
- Switches are the “enforcement” point
  - Packet forwarding
  - Layers 2-7 ACLs
  - NAC



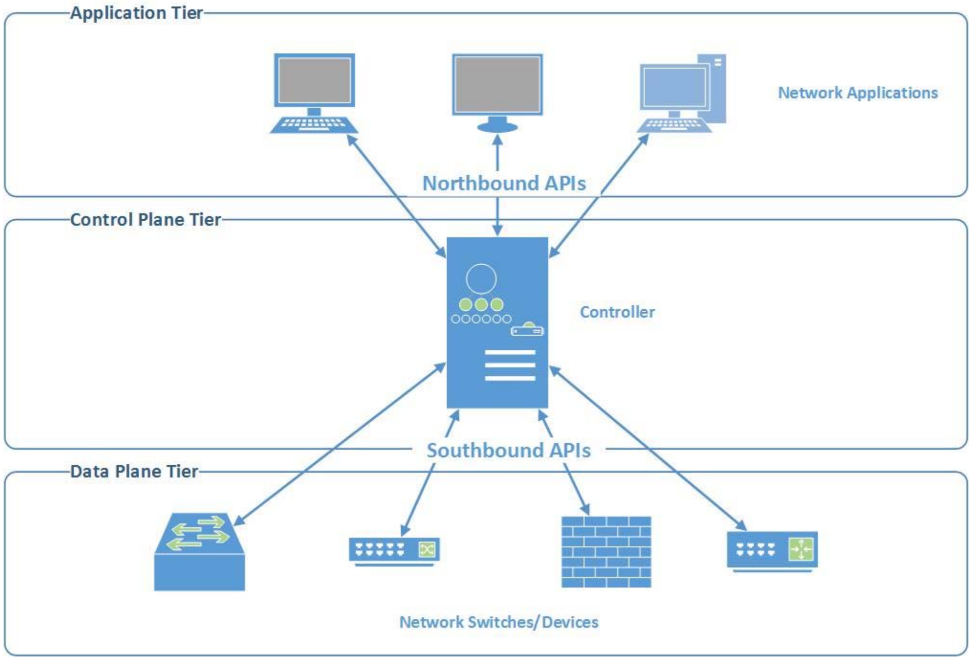
SDN: Controllers

- Controllers are the "brains" of SDN
  - Centralized
  - Programmable
  - Attackable
- Examples include:
  - Mininet
  - Floodlight
  - Cisco APIC
  - HP VAN SDN Controller
  - VMware NSX Controller



SDN: Integration and Control

- At the application tier, northbound APIs:
  - Allow monitoring of controllers and switches
  - Commands to be issued to the control plane
- Management tools from Cisco, HP, Juniper, VMware, BigSwitch, etc. all sit at the application tier
- Focus on role-based access and authentication/authorization



Security Changes with SDN

- Lots of security changes with SDN:
  - Security policy is defined and enforced from applications->controllers->hardware or virtual devices
  - Flow rules (policy) control when or if traffic goes through data plane devices
  - Security isn't enforced by physical topology anymore
  - Requires trust in SDN applications and controllers
  - Network and virtualization teams must collaborate with security teams closely

Network Security Programming and Automation

Automation+Orchestration Redux

- There are differences between classic orchestration and SDN automation
  - SDN != Orchestration
  - SDN != Automation
- SDN leverages APIs that can be used for coordinated automation, however
  - Anuta Networks NCX
  - Nuage Networks Virtualized Services Platform

TOSCA Examples

Node definition

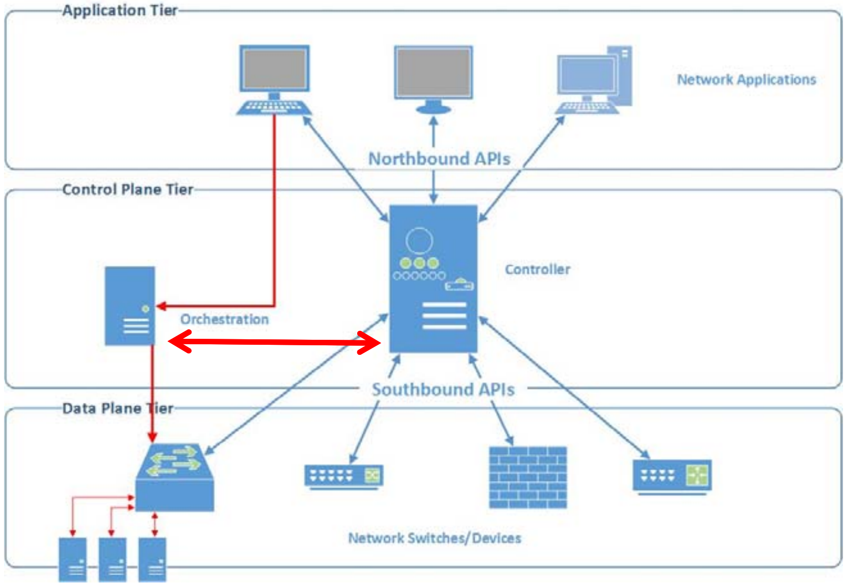
```
sans_vm:
  type: sans.openstack.nodes.Server
  properties:
    server: { get_input: server }
  relationships:
    type: sans.openstack.server_connected_to_floating_ip
    target: sans_ip
    type: sans.relationships.depends_on
    target: All_ports_open
```

Policy Statements

```
policy_node:
  type: policy_node_type
  relationships:
    target: sans_vm
    type: sans.relationships.depends_on
target: a_node
    type: sans.relationships.contained_in
  properties:
    nodes_to_monitor:
      sans_vm
      some_other_vm
```

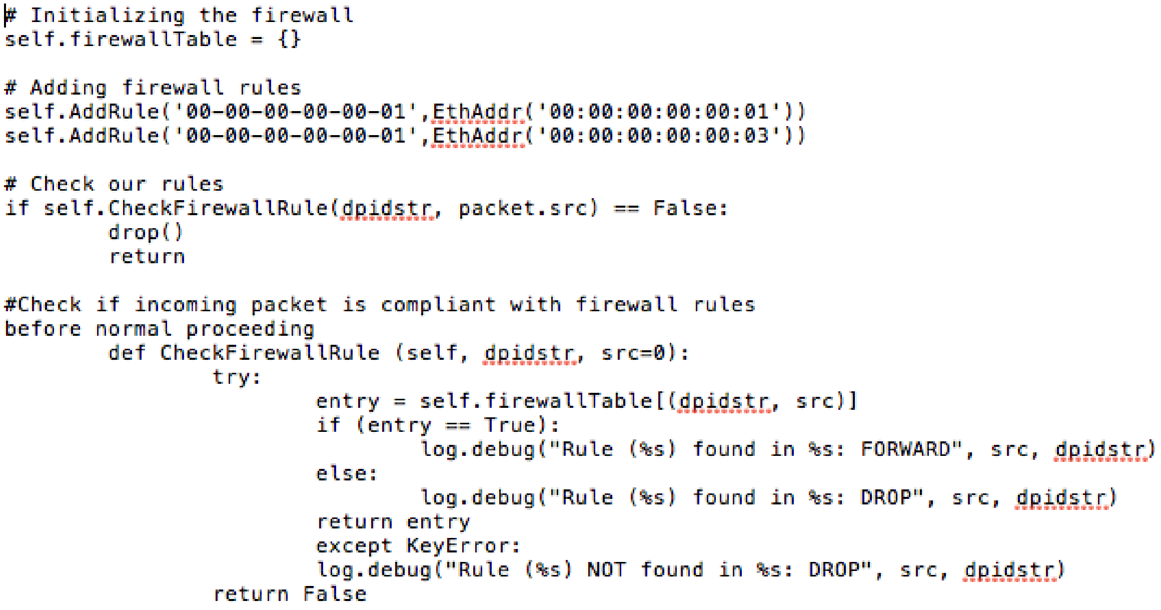
Programming Network Security

- Numerous languages and frameworks can be used to implement orchestration:
  - Ruby and Python
  - Chef and Puppet
  - Custom APIs and REST APIs
- Some will natively integrate with SDN Controllers



Example 1: Firewalls and Access Controls

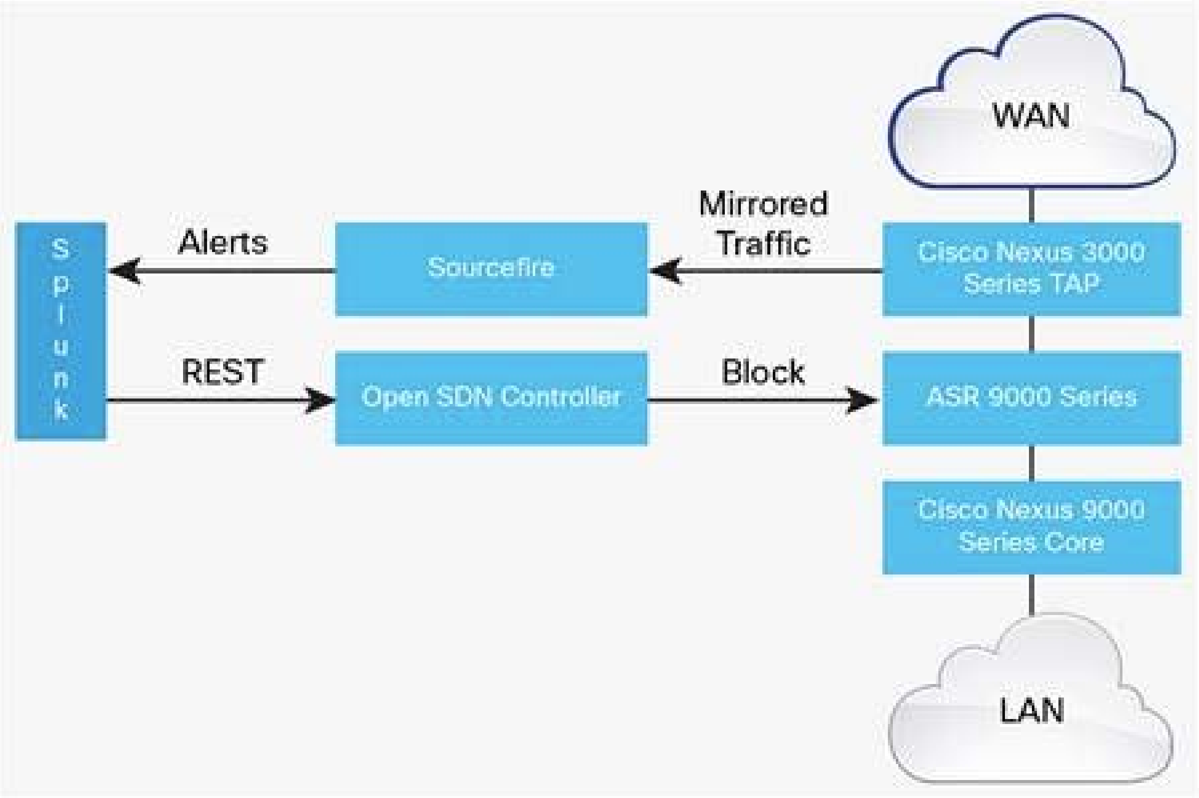
- Simple Python code for firewall implementation
- Central rules and policy can be defined at the controller
- Pushed to switches



```
# Initializing the firewall
self.firewallTable = {}

# Adding firewall rules
self.AddRule('00-00-00-00-00-01',EthAddr('00:00:00:00:00:01'))
self.AddRule('00-00-00-00-00-01',EthAddr('00:00:00:00:00:03'))

# Check our rules
if self.CheckFirewallRule(dpidstr, packet.src) == False:
        drop()
        return

#Check if incoming packet is compliant with firewall rules
before normal proceeding
        def CheckFirewallRule (self, dpidstr, src=0):
                try:
                        entry = self.firewallTable[(dpidstr, src)]
                        if (entry == True):
                                log.debug("Rule (%s) found in %s: FORWARD", src, dpidstr)
                        else:
                                log.debug("Rule (%s) found in %s: DROP", src, dpidstr)
                        return entry
                        except KeyError:
                        log.debug("Rule (%s) NOT found in %s: DROP", src, dpidstr)
                return False
```

Example 1: Firewalls and Access Controls

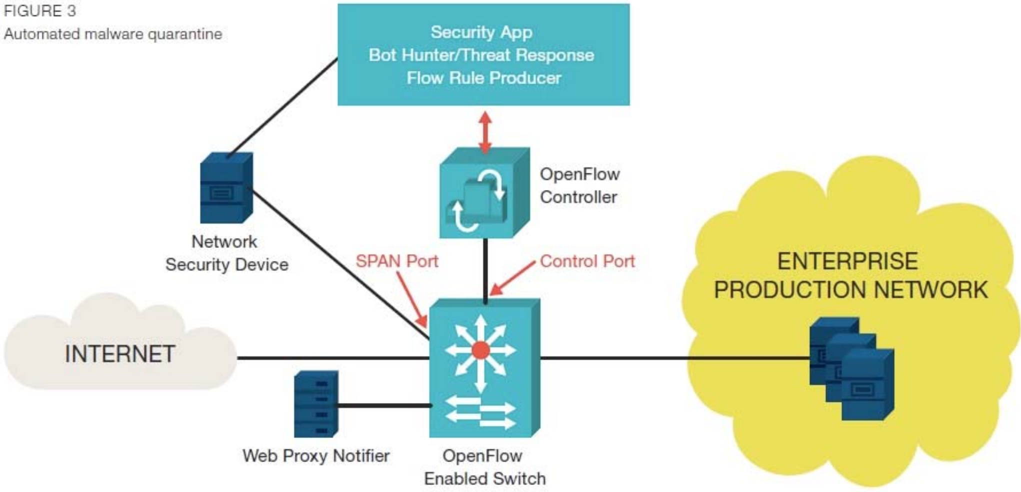
- Cisco Open SDN Controller accepts REST call from Splunk
- Certain events trigger null route block entry for attacker IP



Example 2: Quarantine and IR

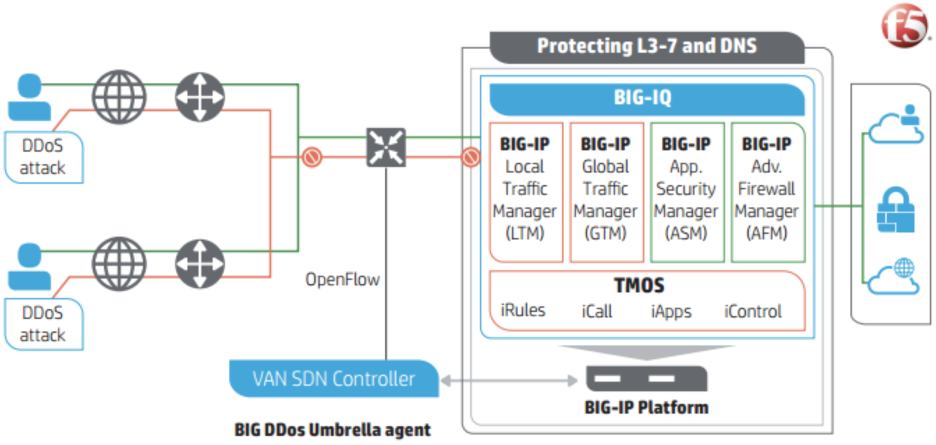
- Internal event at SIEM or other detection platform triggers SDN command to controller
- Controller sends a command to switch to change VLAN for VM or server

FIGURE 3
Automated malware quarantine



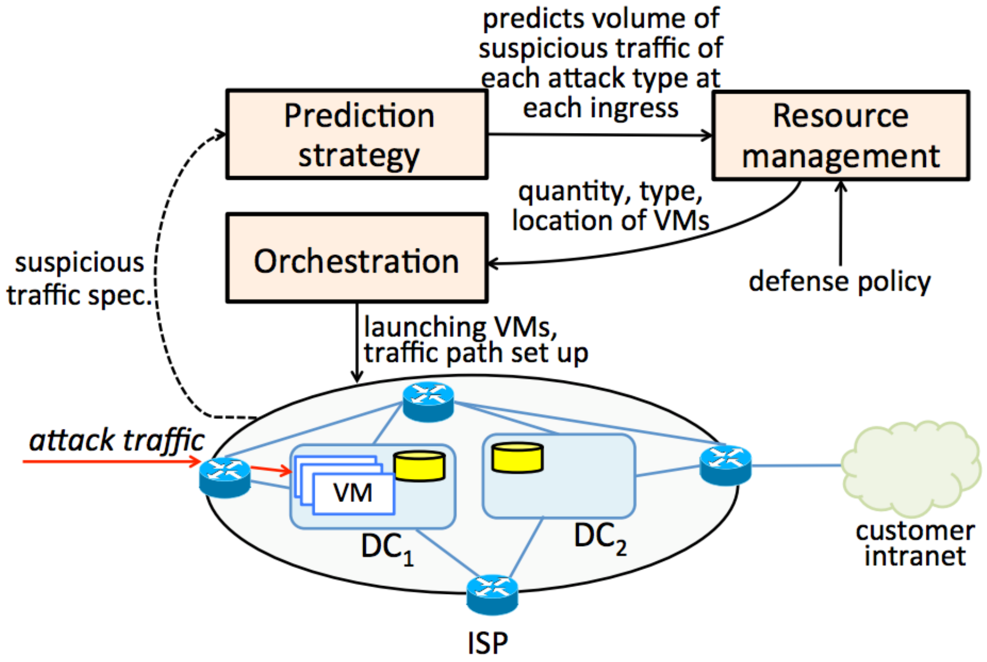
Example 3: DDoS Defense

- Packet attributes can be matched at gateway detection devices
- OpenFlow-enabled controllers can trigger rules in load balancing platforms
- HP and F5 example shown



Example 3: DDoS Defense (Bohatei)

- Bohatei is a DDoS defense system using SDN presented at USENIX 2015
- Uses packet identification, predictive modeling, and network orchestration


predicts volume of suspicious traffic of each attack type at each ingress
Prediction strategy
Resource management
quantity, type, location of VMs
Orchestration
defense policy
suspicious traffic spec.
launching VMs, traffic path set up
attack traffic
VM
DC1
DC2
customer intranet
ISP

Tools and Such

- There are many tools to experiment with SDN today, although security is usually “bolted on” by you
  - Mininet
  - OpenFlow and OpenDaylight
  - Floodlight
  - OpenStack
  - OpenContrail
  - FlowVisor
  - VMware NSX
  - Cisco APIC

Wrapping Up

Moving toward SDN and Security

- Next week you should:
  - Look at existing network vendors' capabilities and explore a lab setup
- In the first three months following this presentation you should:
  - Learn more about OpenFlow and related standards
  - Discuss internal use cases for SDN, and security specifically within SDN
- Within six months you should:
  - Align network update and architecture roadmaps with SDN capabilities and tools
  - Consider how automation and orchestration of network functions might work in your environment

Resources for Security Pros

- Great resources on SDN and (some) security:
  - http://searchsdn.techtarget.com/
  - https://www.sdxcentral.com/resources/security/security-challenges-sdn-software-defined-networks/
  - https://www.opennetworking.org/solution-brief-sdn-security-considerations-in-the-data-center
  - https://ngn.cs.colorado.edu/~coughlin/doc/a_survey_of_sdn_security_research.pdf