Abusing Software Defined Networks

Black hat Europe 2014, Amsterdam
Gregory Pickett, CISSP, GCIA, GPEN
Chicago, Illinois

gregory.pickett@hellfiresecurity.com
Overview

- What is it?
- Exploiting it!
- Fixing it!
- Moving Forward
- Wrapping Up
Modern Day Networks

- Vendor Dependent
- Difficult to scale
- Complex and Prone to Break
- Distributed and Often Inconsistent Configuration
- Uses inflexible and difficult to innovate protocols
- Unable to Consider Other Factors

... And Good Luck If You Want To Change It!
Enter . . . Software Defined Networking

- Separate the Control and Data Plane
  - Forwarding Decisions Made By a Controller
  - Switches and Routers Just Forward Packets

- Controllers
  - Programmed with the Intelligence
  - Full visibility of the Network
  - Can consider the totality of the network before making any decision
  - Enforce Granular Policy
Enter . . . Software Defined Networking

- Switches
  - Bare-Metal Only
  - Any Vendor . . . Hardware or Software
Solves Lots of Problems

- Less Expensive Hardware
- With BGP
  - Maintenance Dry-Out
  - Customer Egress Selection
  - Better BGP Security
  - Faster Convergence
  - Granular Peering at IXPs
Expands Our Capability

- Real-World Network Slicing of Flow Space
- Network and Server Load Balancing
- Security
  - Dynamic Access Control
  - Adaptive Traffic Monitoring
  - Attack Detection and Mitigation
Emerging Standards

Old and Busted
- SNMP
- BGP
- Netconf
- LISP
- PCEP

New Hotness
- OVSDB
- Openflow
Introducing Openflow

- Establishes Elements
  - Controller
  - Secure Channel
  - Forwarding Element
- Defines . . .
  - Forwarding Process
  - Messaging Format
Introducing Openflow

**Forwarding Process**
- Check Flow Table
- If Match Found, Execute Action
- If No Match, Send Packet to controller
- Update Flow Table

**Flow Tables**
- Match/Action Entries
- 12 fields available for matching
- Wildcard matching available
Introducing OpenFlow
Leading Platforms

**Proprietary**
- Cisco Application Policy Infrastructure Controller (APIC)
- Cisco Extensible Network Controller (XNC)
- HP Virtual Application Networks (VAN) SDN Controller
- IBM Programmable Network Controller

**Open-Source**
- Nox/Pox
- Ryu
- Floodlight
- OpenDaylight
Floodlight

- Open-Source Java Controller
- Primarily an Openflow-based controller
- Supports Openflow v1.0.0
- Fork from the Beacon Java Openflow controller
- Maintained by Big Switch Networks
Opendaylight

- Open-Source Java Controller
- Many southbound options including Openflow
- Supports Openflow v1.0.0 and v1.3.0
- Fork from the Beacon Java Openflow controller
- A Linux Foundation Collaborative Project
- Supported by Citrix, Red Hat, Ericsson, Hewlett Packard, Brocade, Cisco, Juniper, Microsoft, and IBM
So It’s Gonna Be All . . .

Not Exactly!
Protocol Weaknesses

- Encryption and Authentication via TLS
- More of a suggestion than a requirement though ...
  - Started Out Good
  - Heading Backwards
    - v1.0.0 over TLS
    - v1.4.0 over TCP or TLS
Protocol Weaknesses

**Controllers**
- Floodlight ... Nope
- Opendaylight ... Supported but not required

**Switches**
- Arista ... No
- Brocade ... Surprisingly, Yes
- Cisco ... Another, Yes
- Dell ... No
- Extreme ... Another, Yes
- HP ... No
Protocol Weaknesses

**Switches**

- Huawei ... No
- IBM ... No
- Juniper ... No
- NEC ... Another, Yes
- Netgear ... No
- Pronto ... Yes
- OVS ... No
Could Lead To . . .

- **Information Disclosure** through Interception
- **Modification through Man-in-the-Middle**
- And all sorts of **DoS Nastiness**!
DoS Nastiness

- Openflow
  - Centralization Entails Dependency
  - Dependency Can Be Exploited
  - How are vendors handling it?
- Floodlight
  - Explored by Solomon, Francis, and Eitan
  - Their Results ... Handling It Poorly
- Opendaylight
  - Unknown but worth investigating
  - It is Java for God Sake!
Tools

**of-switch.py**
- Impersonates an Openflow switch
- Utilizes Openflow v1.0.0

**of-flood.py**
- Floods an Openflow controller
- Disrupting the network and bringing it down
- Utilizes Openflow v1.0.0
Debug Ports

- No Encryption
- No Authentication
- Just Full Control of the Switch
- All Via “dpctl” command-line tool
- Not a problem yet . . .
- But Soon Will Be!
Controller Weaknesses

**Floodlight**
- No Encryption for Northbound HTTP API
- No Authentication for Northbound HTTP API

**Opendaylight**
- Encryption for Northbound HTTP API
  - Turned Off by Default
- Authentication for Northbound HTTP API
  - HTTP Basic Authentication
  - Default Password Weak
  - Strong Passwords Turned Off by Default
Could Lead To . . .

- **Information Disclosure** through Interception
  - Topology
  - Credentials
- **Information Disclosure through Unauthorized Access**
  - Topology
  - Targets
And . . .

+ Topology, Flow, and Message Modification through Unauthorized Access
  + Add Access
  + Remove Access
  + Hide Traffic
  + Change Traffic
Identifying Controllers and Switches

- Currently Listening on TCP Port 6633
- New Port Defined ... TCP Port 6653
- Hello's Exchanged
- Feature Request
  - Controller will send
  - Switch will not
Tools

**of-check.py**
- Identifies Openflow Services
- Reports on their Versions
- Compatible with any version of Openflow

**of-enum.py**
- Enumerates Openflow Endpoints
- Reports on their Type
- Compatible with any version of Openflow
of-enum.nse
- Enumerates Openflow Endpoints
- Reports on their Type
- Compatible with any version of Openflow
Demonstration
Some Attacks

- Small Local Area Network
  - One Admin Host
  - Two User Hosts
  - One Server
  - One IDS
- Attacker will ... 
  - Identify Targets
  - Enumerate ACLs
  - Find Sensors
Tool

**of-map.py**
- Downloads flows from an Openflow controller
- Uses the flows
  - To identify targets and target services
  - To build ACLs
  - To identify sensors
- Works with Floodlight and Opendaylight via JSON
Demonstration
And Some More Attacks . . .

- Small Local Area Network
  - One Admin Host
  - Two User Hosts
  - One Server
  - One IDS

- Attacker will . . .
  - Gain Access to the Server
  - Isolate the Administrator
  - Hide from the IDS
  - And Attack the Server
**Tool**

**of-access.py**

- Modifies flows on the network through the Openflow Controller
- Adds or Removes access for hosts
- Applies transformations to their network activity
- Hides activity from sensors
- Works with Floodlight and Opendaylight via JSON
Demonstration
And Now Some Pwnage . . .

Sorry Linux Foundation!
Zero-Day Exploit

- Opendaylight has other southbound APIs besides Openflow
  - No Encryption for Southbound Netconf API
  - No Authentication for Southbound Netconf API
- Just Connect and Exchange Messages
  - XML-RPC
  - Remember Java?
- Boom Goes Opendaylight
- And it runs as “Root”
Demonstration
If No Exploit...

- Service Not Available or They Fix It
- Not to Worry
- Password Guess the !!!!!!
  - Default Password Weak
  - Strong Passwords Turned Off
  - No Account Lockout
  - No SYSLOG Output
Attackers will ...

- Identify Targets
- Enumerate ACLs
- Find Sensors
- Gain Access to the Server
- Isolate the Administrator
- Hide from the IDS
- And Attack the Server
- And Pwn That Network Too!
Other Exploits Waiting to Be Found!

**Floodlight**
- Northbound HTTP API
- Southbound Openflow API

**Opendaylight**
- Northbound HTTP API
- Southbound Openflow API
- Southbound Netconf API (TCP, SSH)
- Southbound Netconf Debug Port
Other Exploits Waiting to Be Found!

- Opendaylight
  - JMX Access
  - OSGi Console
  - Lisp Flow Mapping
  - ODL Internal Clustering RPC
  - ODL Clustering
  - Java Debug Access
Available Solutions

- For Now
- For the Future
For Now

- Transport Layer Security
  - Feasible?
  - Realistic?
- Hardening ... Duh!
- VLAN ... It’s the Network Stupid!
- Code Review Anyone?
For the Future

- Denial of Service (SDN Architecture)
  - Network Partitioning
  - Controller Clustering
  - Static Flow Entries
- Modification (SDN Applications)
  - Traffic Counters
  - Respond to Abnormalities
- Verification (SDN Operations)
How Prevalent Is It Going To Be?

- Gartner: 10 critical IT trends for the next five years
- Major Networking Vendors Have Products or Products Planned for SDN
- InformationWeek 2013 Survey
  - 60% felt that SDN would be part of their network within 5 Years
  - 43% already have plans to put it in production
Reported

While Data Centers/Clouds are the Killer App for SDN
  - NIPPON EXPRESS
  - FIDELITY INVESTMENTS
  - VMWARE

Starting to see it moving toward the LAN
  - Caltech
  - Cern

And WAN
  - Google, NTT, and AT&T
How It Could Go Right

- Vendor Independence and ultimately lower cost
- Networks that match the application and the businesses needs not the other way around
- Faster Evolution of the Network
  - Production-Scale Simulation and Experimentation
  - Exchangeable Network Aspects
- Dynamic and Truly Active Defenses
How It Could Go Wrong

- Denial of Service
  - Peer Node
  - External Node
  - Selectively Dropping Traffic?
- MiTM
  - Entire Networks
  - Local Subnets or Hosts
- Shadow Operations
  - Darknets
  - Uber Admins
Making the Difference

+ Traditional Means of Securing Controllers Still Apply
+ Security Needs to Be Part of the Discussion
+ Until Now ... How SDN Can Help Security
+ But How Secure is SDN?
+ Analyses being Done
  + But By Outsiders
  + Traditional Approach and 2-D
+ Controller’s Need A Security Reference and Audit Capability
Final Thoughts

- SDN has the potential to turn the entire Internet into a cloud
- Benefit would be orders of magnitude above what we see now
- But there is hole in the middle of it that could easily be filled by the likes of the NSA . . . or worse yet, China
- Let’s Not Let That Happen
- And That Start’s Here
Toolkit

SDN-Toolkit v1.01 for Openflow Networks

- Discover, Identify, and Manipulate SDN-Based Networks
- Floodlight and Opendaylight support through Northbound HTTP-Based APIs
- Openflow v1.0.0 support through Southbound Openflow APIs
- Python-Based

SHA1 hash is 5de4f56de0ce24cc5b4fcd691ff4e7e910e0b80b
Updates can be found at http://www.hellfiresecurity.com/
Links

- http://www.sdncentral.com/
- https://www.opennetworking.org/
- http://www.projectfloodlight.org/
- http://www.opendaylight.org/
- https://www.coursera.org/course/sdn
- http://www.openflowhub.org/blog/blog/2012/12/03/sdn-use-case-multipath-tcp-at-caltech-and-cern/