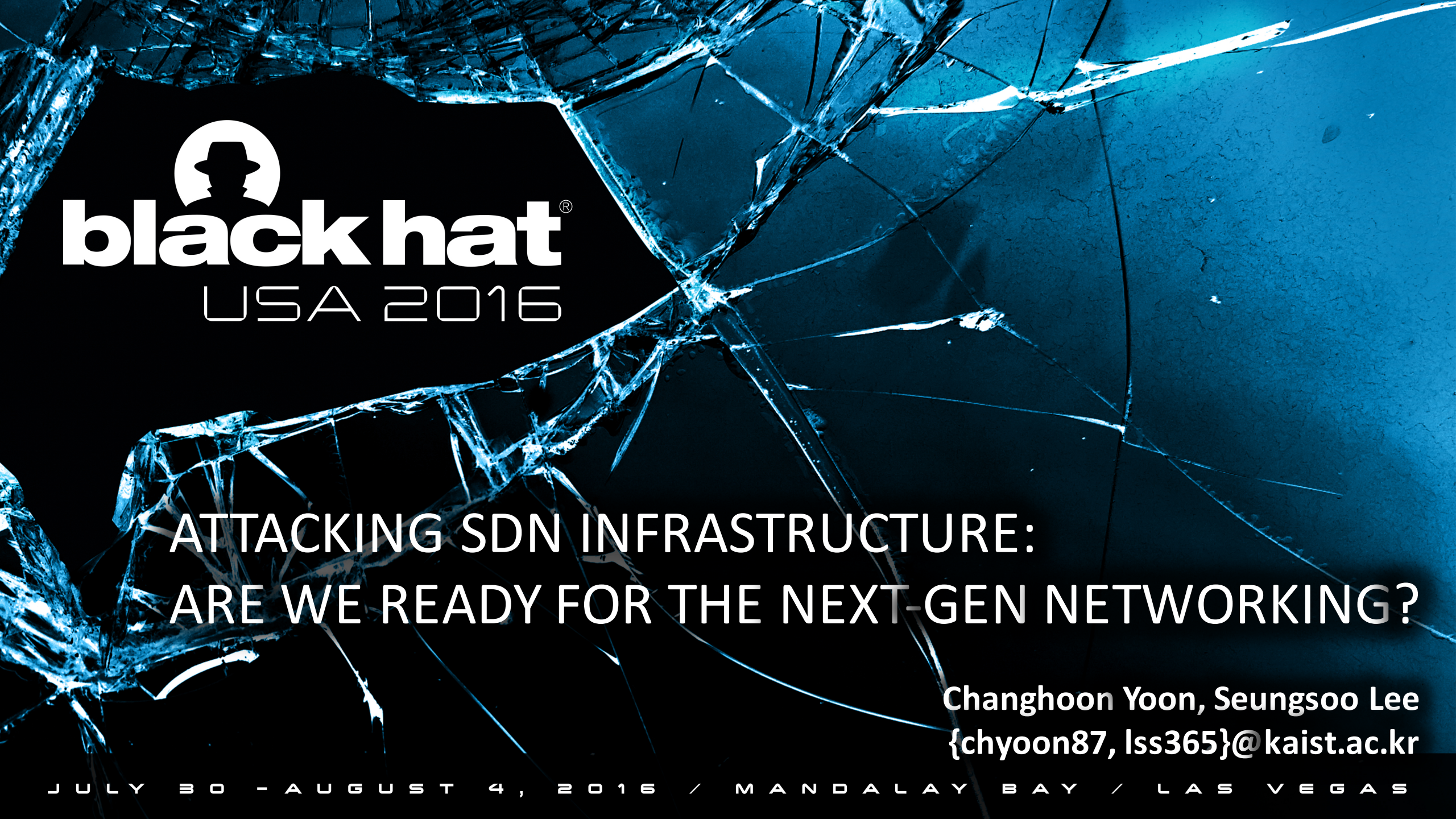


ATTACKING SDN INFRASTRUCTURE: ARE WE READY FOR THE NEXT-GEN NETWORKING?

Changhoon Yoon, Seungsoo Lee
{chyoon87, lss365}@kaist.ac.kr

JULY 30 - AUGUST 4, 2016 / MANDALAY BAY / LAS VEGAS

Contents

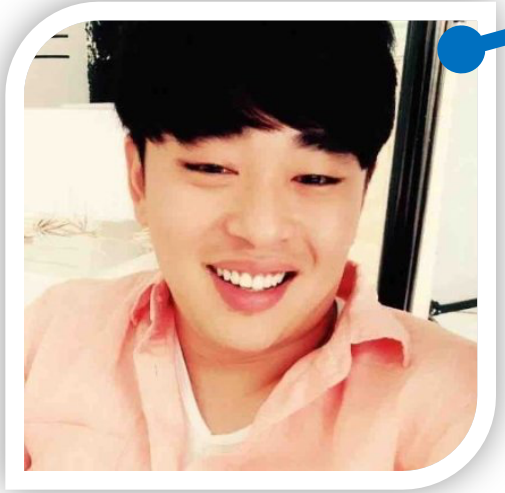
About us

NS2 Network and System Security Laboratory

http://nss.kaist.ac.kr

Seungwon Shin

- Assistant Professor of EE dept. at KAIST
- Research Associate of Open Networking Foundation (ONF)
- Leading Network and System Security Lab.



Changhoon Yoon

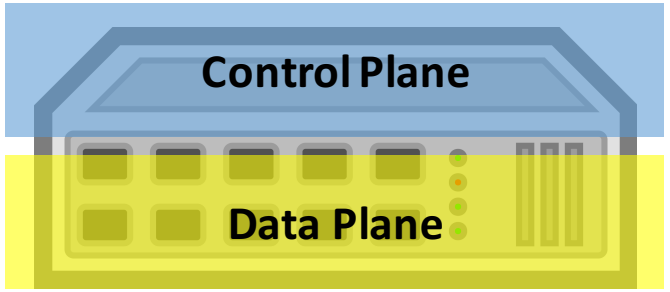
- PhD student at KAIST
- Project SM-ONOS

Seungsoo Lee

- PhD student at KAIST
- Project DELTA

Traditional Networking

- Too complicated
  - Control plane is implemented with complicated S/W and ASIC
  - Unstable, increased complexity in management
- Closed platform
  - Vendor specific
- Hard to modify (nearly impossible)
  - Hard to add new functionalities
  - Barrier to innovation

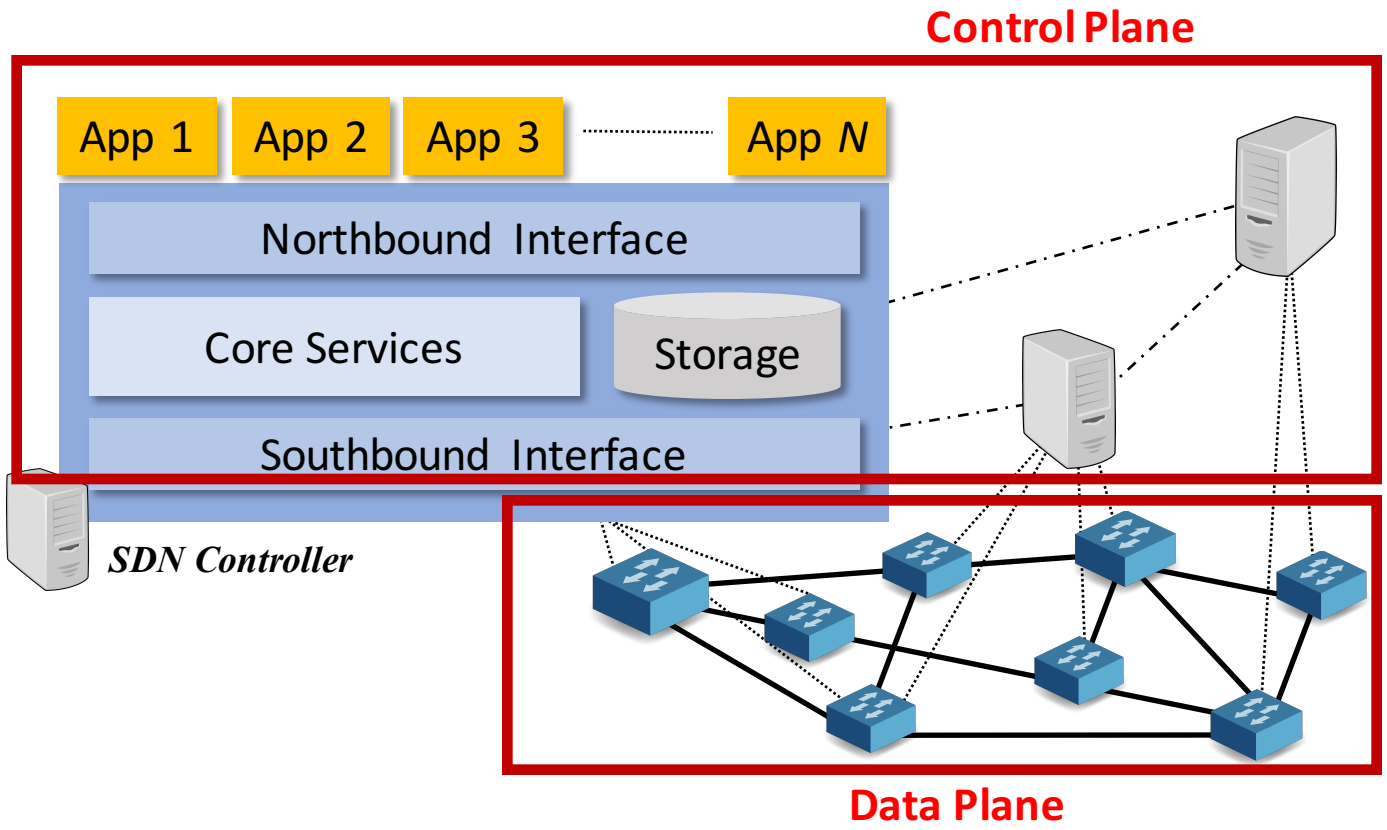


Legacy Network Device

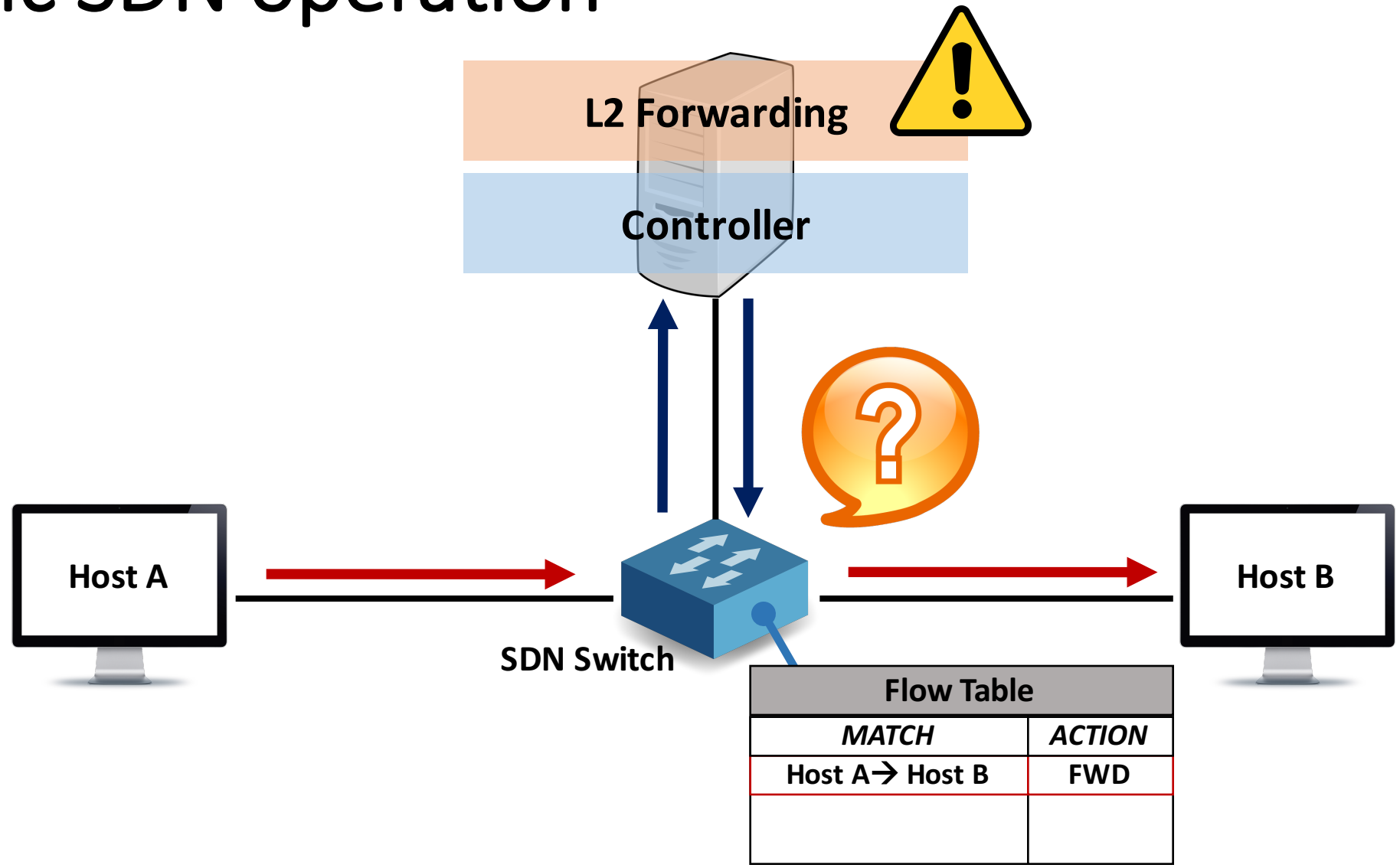
- New proposal: ***Software Defined Networking (SDN)***
  - Separate the control plane from the data plane

What is Software Defined Networking (SDN)?

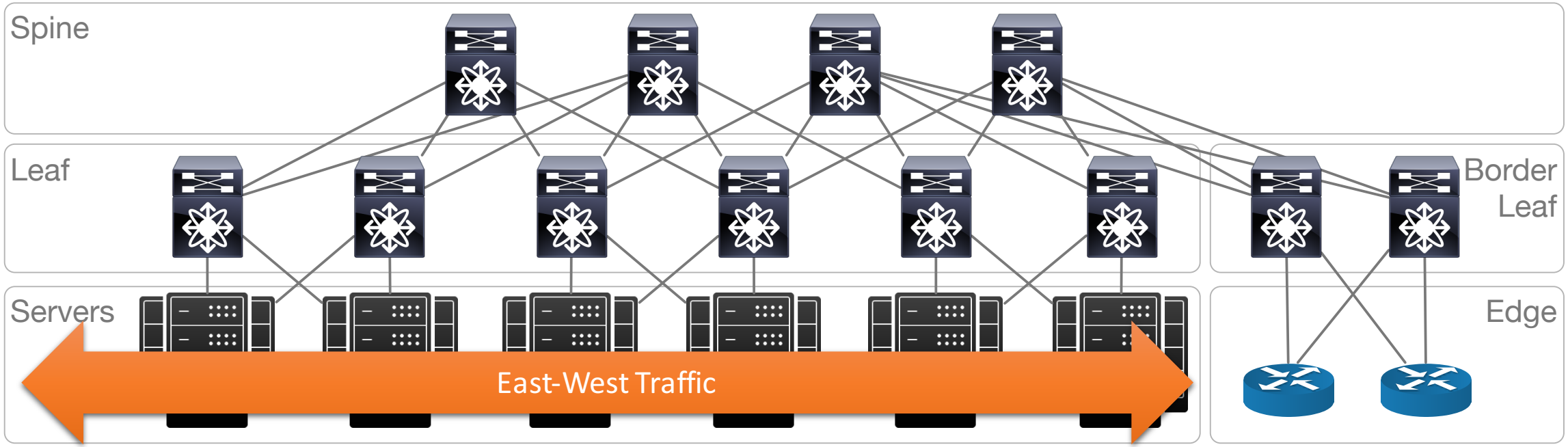
- Centralized network management
  - Via global network view
- Programmable network
  - Flexible and dynamic network control
  - useful, innovative SDN applications
- CAPEX, OPEX reduction
  - Commodity servers and switches



Basic SDN operation

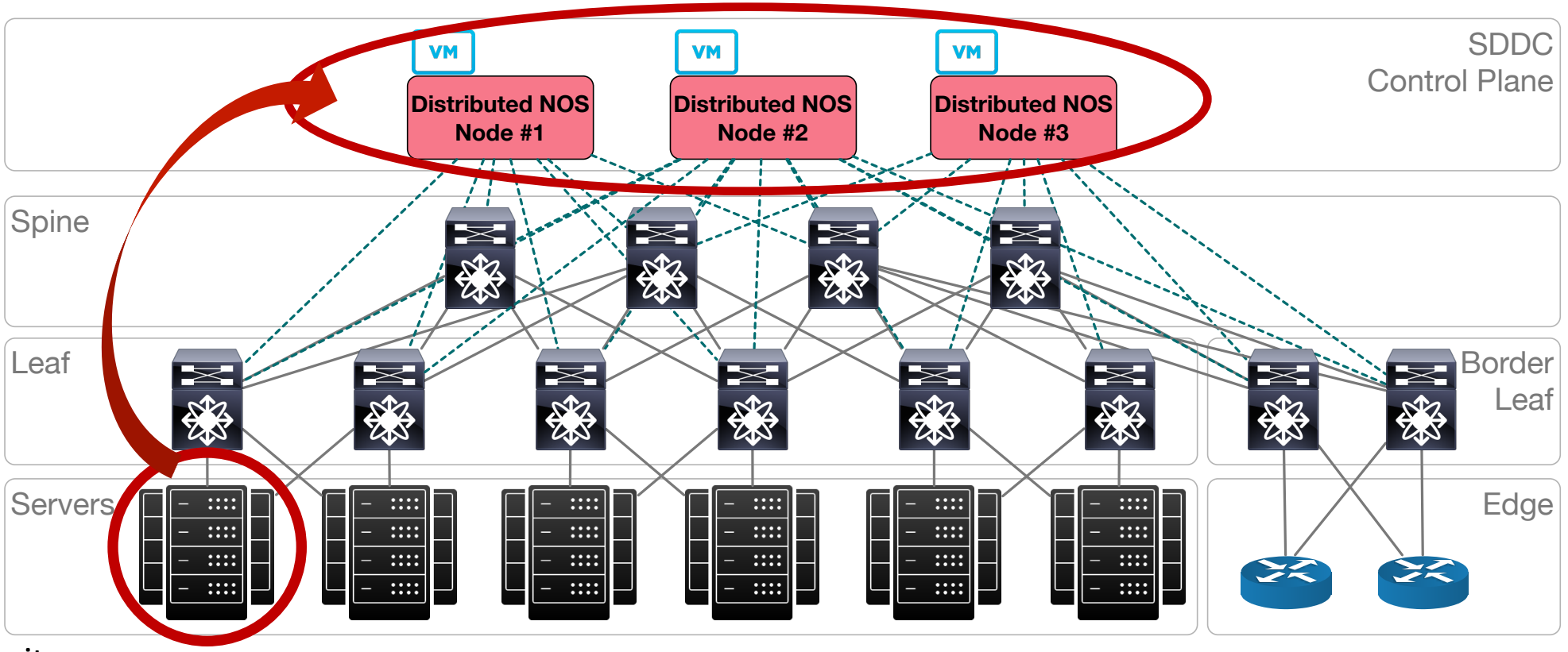


Data Center Network Design



- Today's Data Center involves a LOT of Virtual Machines (VMs)
- "Leaf-Spine" Design
  - Suitable for handling "East-West" traffic; low latency & bottlenecks
- Remaining challenges
  - Increased complexity – frequent VM migrations, a large number of links
  - Expensive to scale & maintain

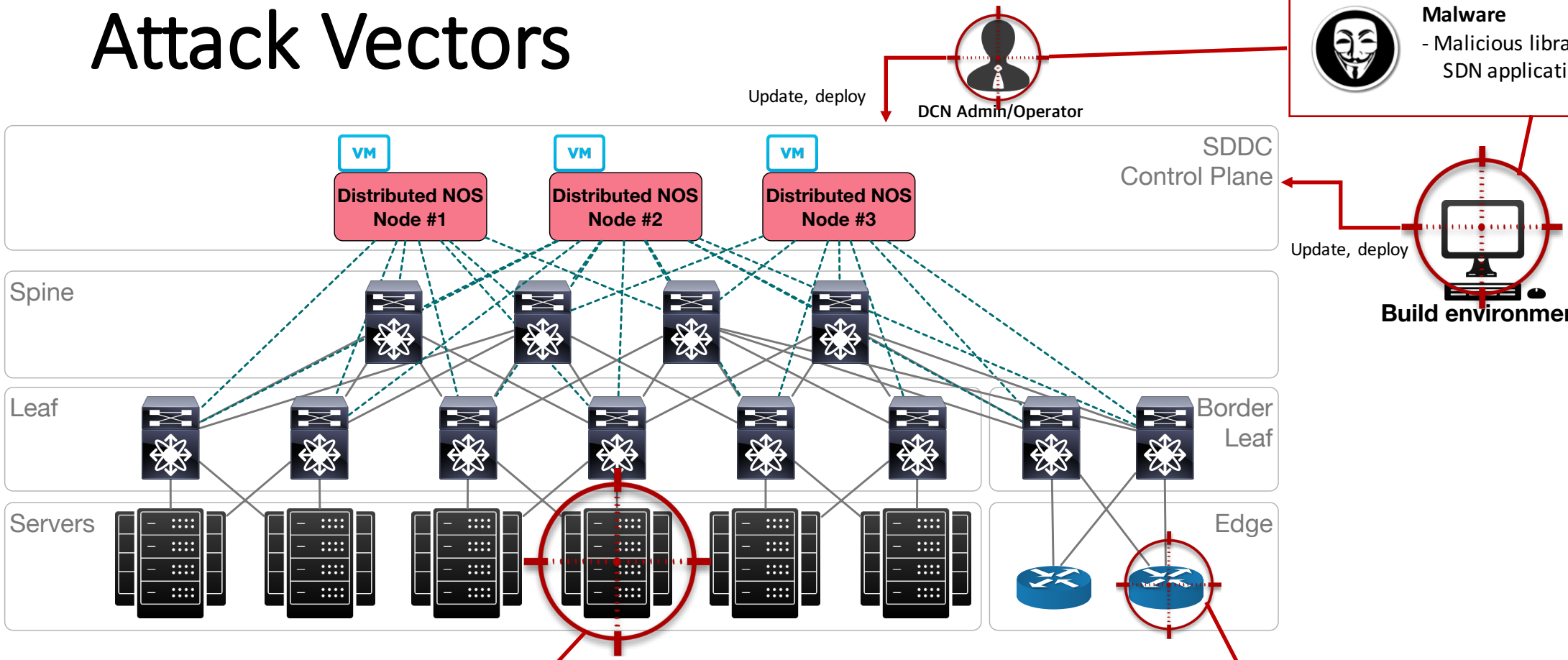
Software-Defined Data Center (SDDC)



- Low complexity
  - Global network view + Network programmability
- Low cost
  - Commodity servers & switches
  - Centralized & automated management

- Highly available & scalable control plane
  - Distributed SDN controller
    - VMs to host controller nodes

Attack Vectors



SDN Control Plane Components

- **Open Source SDN Controller (NOS) implementations**
  - Open Network Operating System (ONOS) & OpenDaylight (ODL)

  - Cutting-edge, distributed network operating systems (NOS)
  - Provide base design for commercial SDN controller products
    - Brocade SDN Controller [1] : ODL-based
  - Both are Maven projects
  - Both run on Karaf OSGi container

[1] http://www.brocade.com/en/products-services/software-networking/sdn-controllers-applications/sdn-controller.html

Attack Vector: Misconfiguration

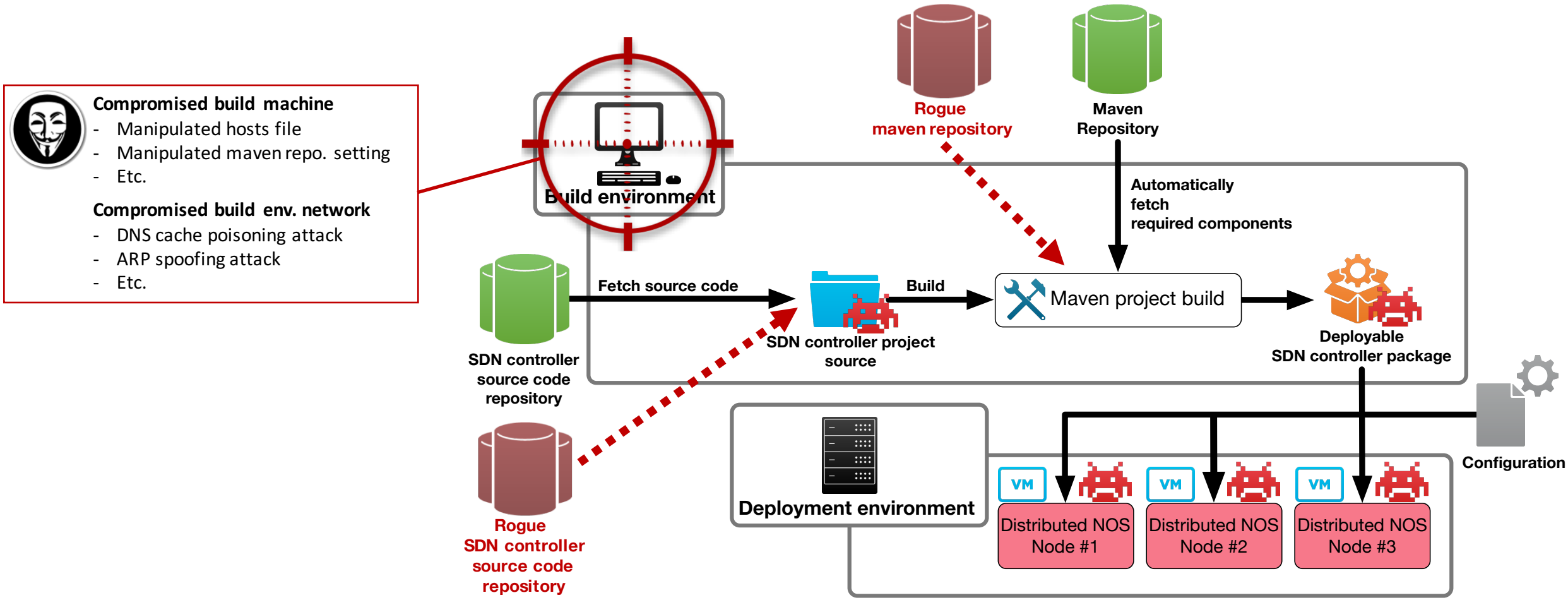
- **Remotely accessible interfaces**
  - Remote SSH to NOS host machines
  - Karaf container CLI
  - WebConsole, GUI
  - REST API
- **Defenses**
  - Follow the security guideline available here:
    - http://docs.opendaylight.org/en/latest/getting-started-guide/security_considerations.html
      - Changing default credentials
      - Properly configuring Firewall policies to block remote access

Attack Vector: Malware

- Malware infection at build-time
- Malware infection at runtime

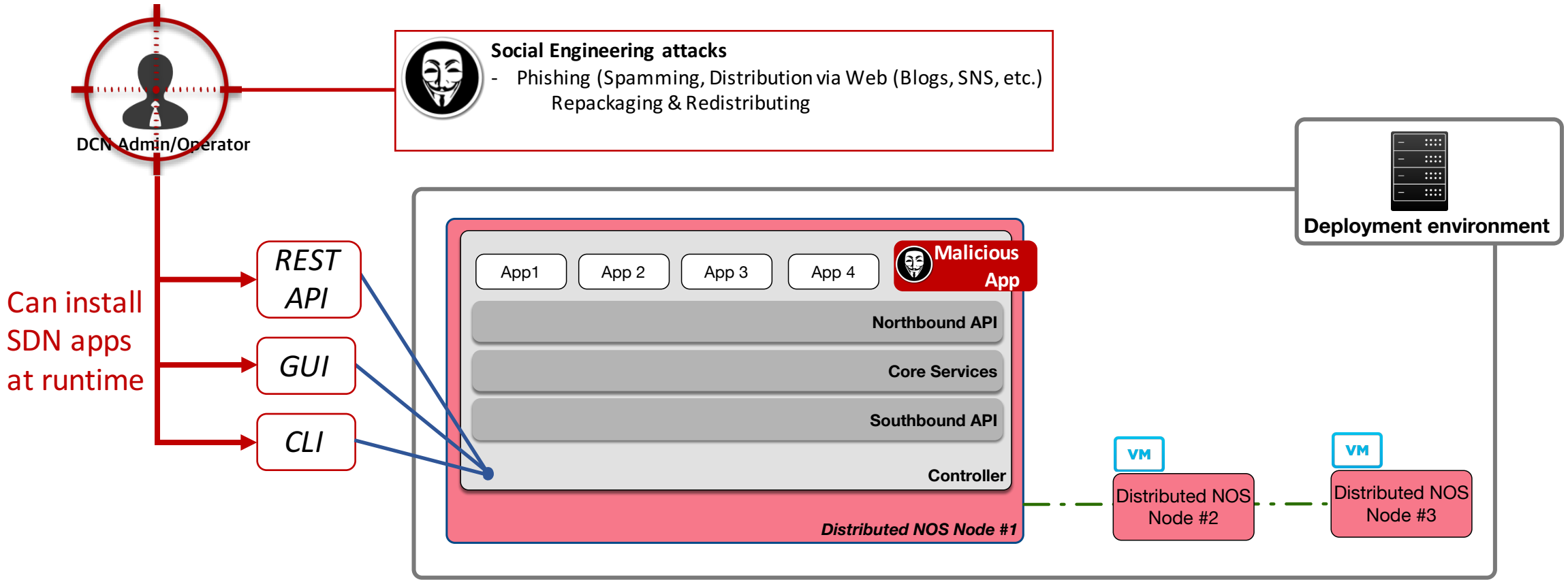
Attack Vector: Malware 1

- **Compromising SDN Control Plane at Build-time**



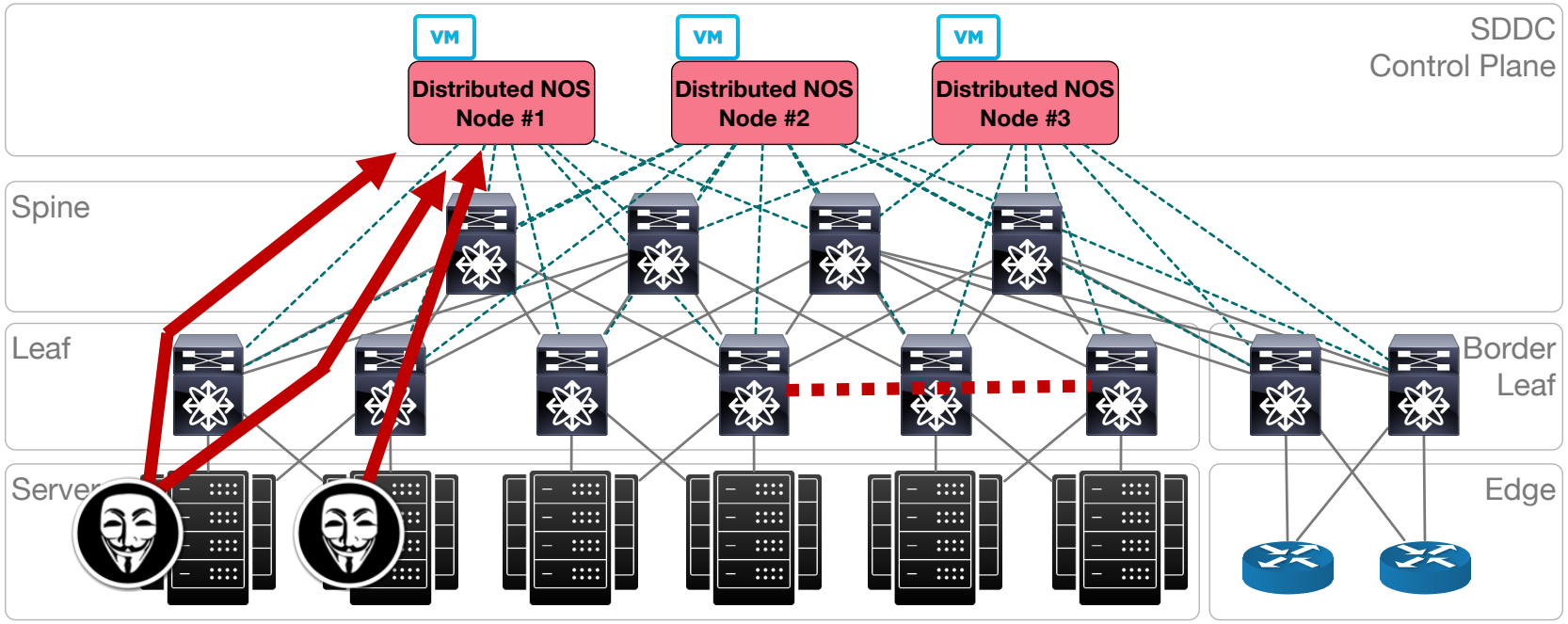
Attack Vector: Malware 2

- **Compromising SDN Control Plane at Runtime**



Attack Vectors: Insider (tenant) attacks

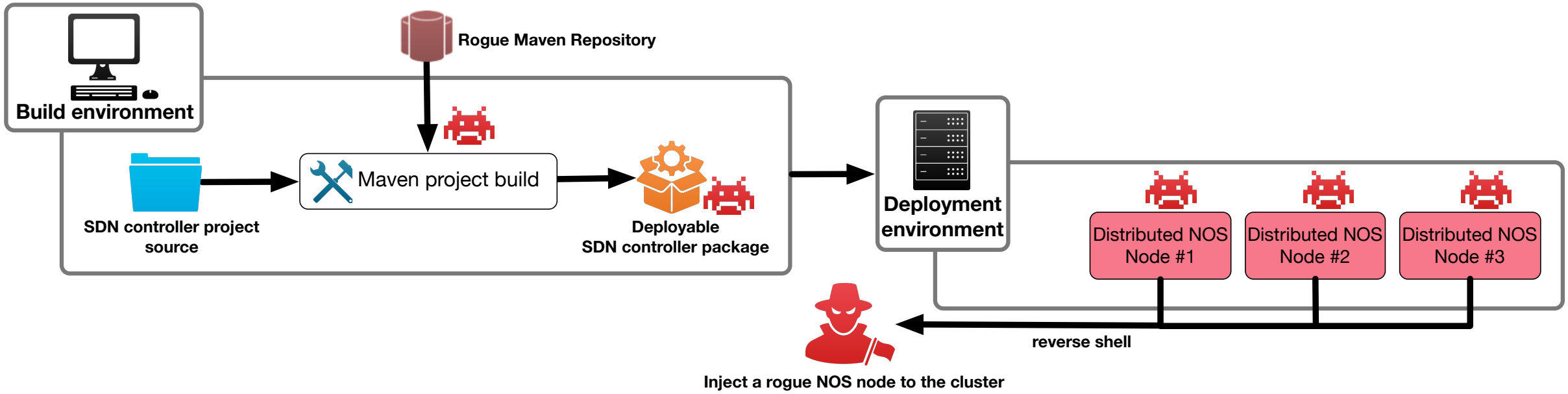
- **Malicious tenants**
  - Control plane saturation attack (DoS) against the NOS cluster [1]
  - Topology Poisoning [2]



[1] Shin, Seungwon, and Guofei Gu. "Attacking software-defined networks: A first feasibility study." Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking. ACM, 2013.
[2] Hong, Sungmin, et al. "Poisoning Network Visibility in Software-Defined Networks: New Attacks and Countermeasures." NDSS. 2015.

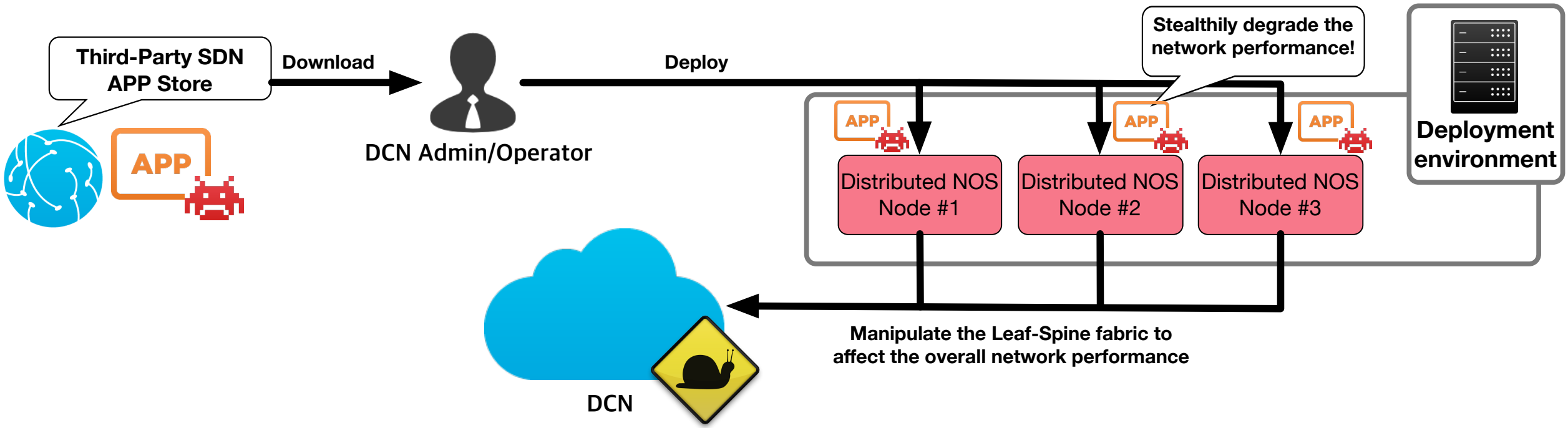
Attack Scenario 1

Compromising SDN control plane at build time to launch arbitrary SDN controller node injection attack



Attack Scenario 2

Compromising SDN control plane at runtime to launch stealth network performance attack



The Vulnerabilities (selected from the scenario)

1. No System Integrity Protection
2. No authentication of NOS cluster nodes
3. No application access control
4. Switch Device Firmware Abuse
5. Packet-IN Flooding
6. Control Message Manipulation
7. Eavesdrop
8. Internal Storage Manipulation
9. ….

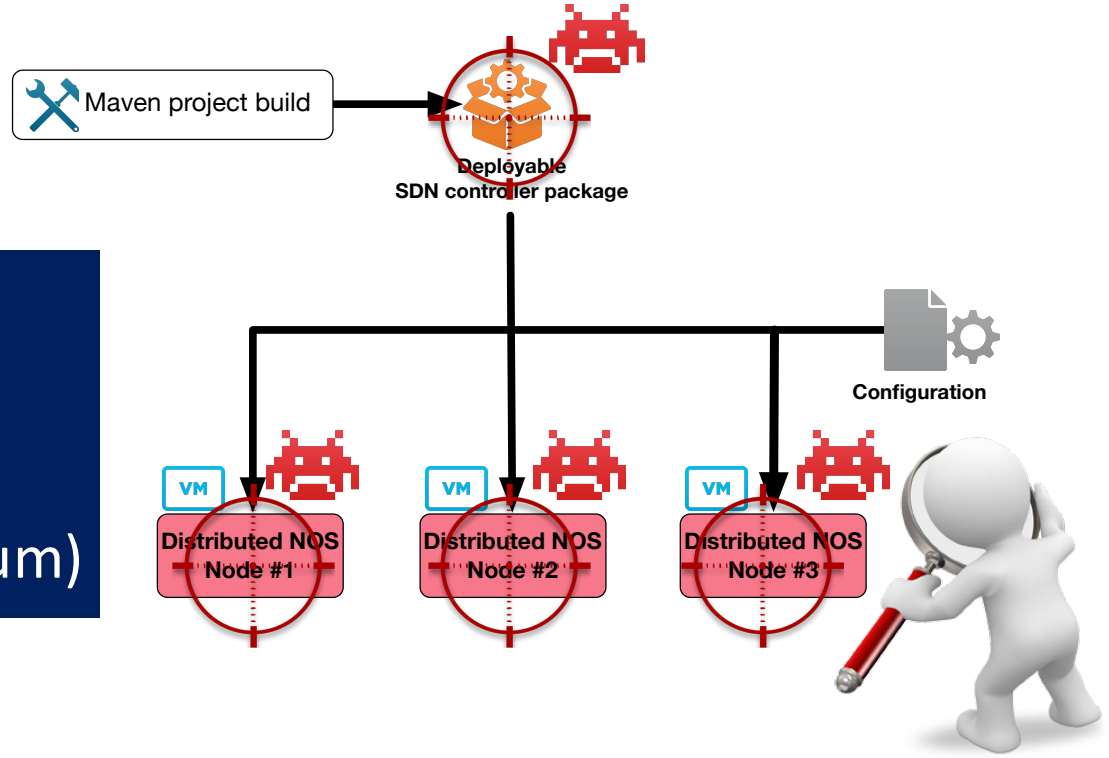
Want more?

Visit http://sdnsecurity.org !!

(will open in 09/2016)

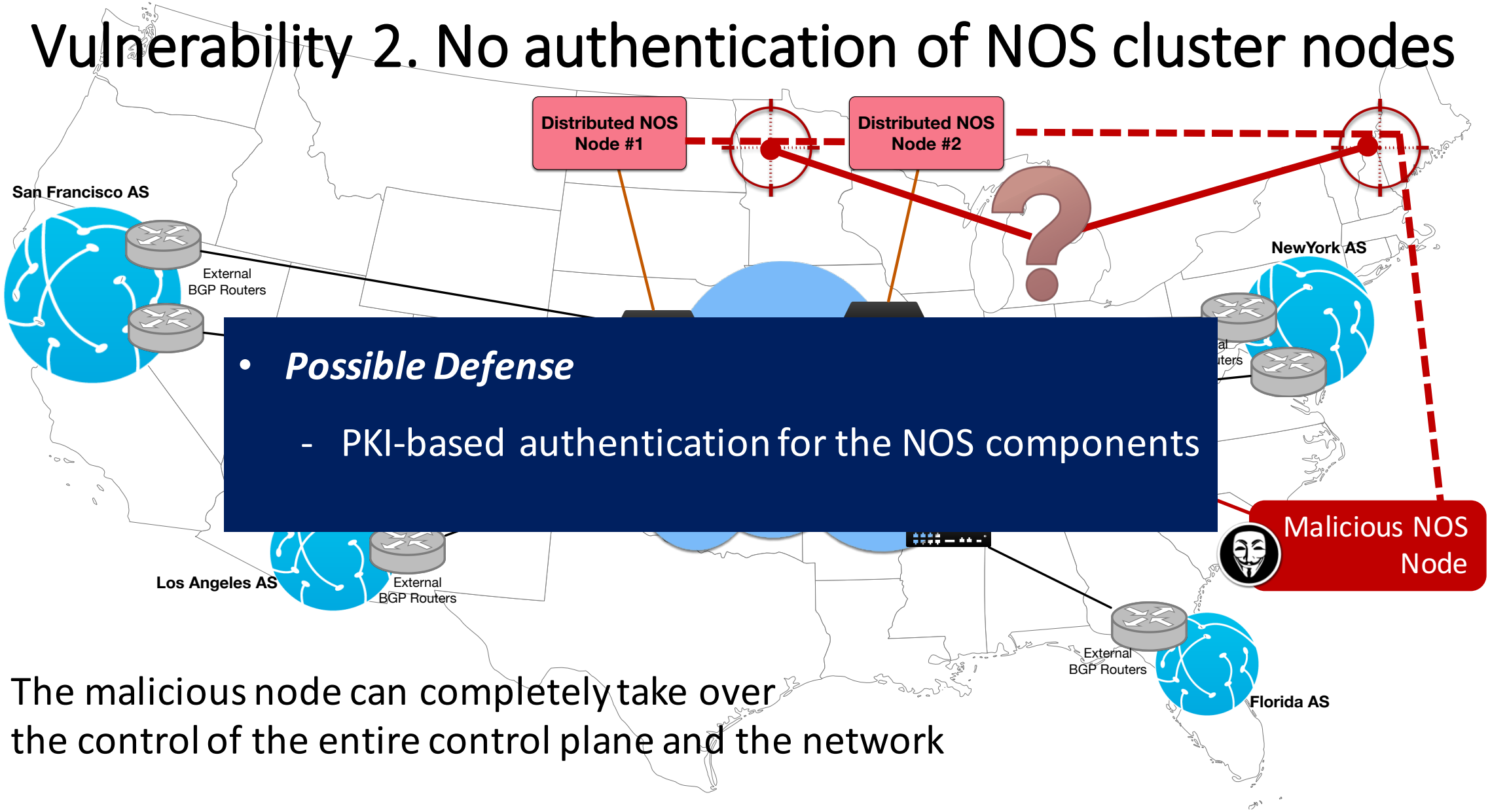
Vulnerability 1. No system integrity protection

- There is ***no system integrity protection*** for NOS components
  - Integrity of the CORE NOS components must be guaranteed



- ***Possible Defenses***
  - Code signing
  - Integrity protection mechanisms (e.g. checksum)

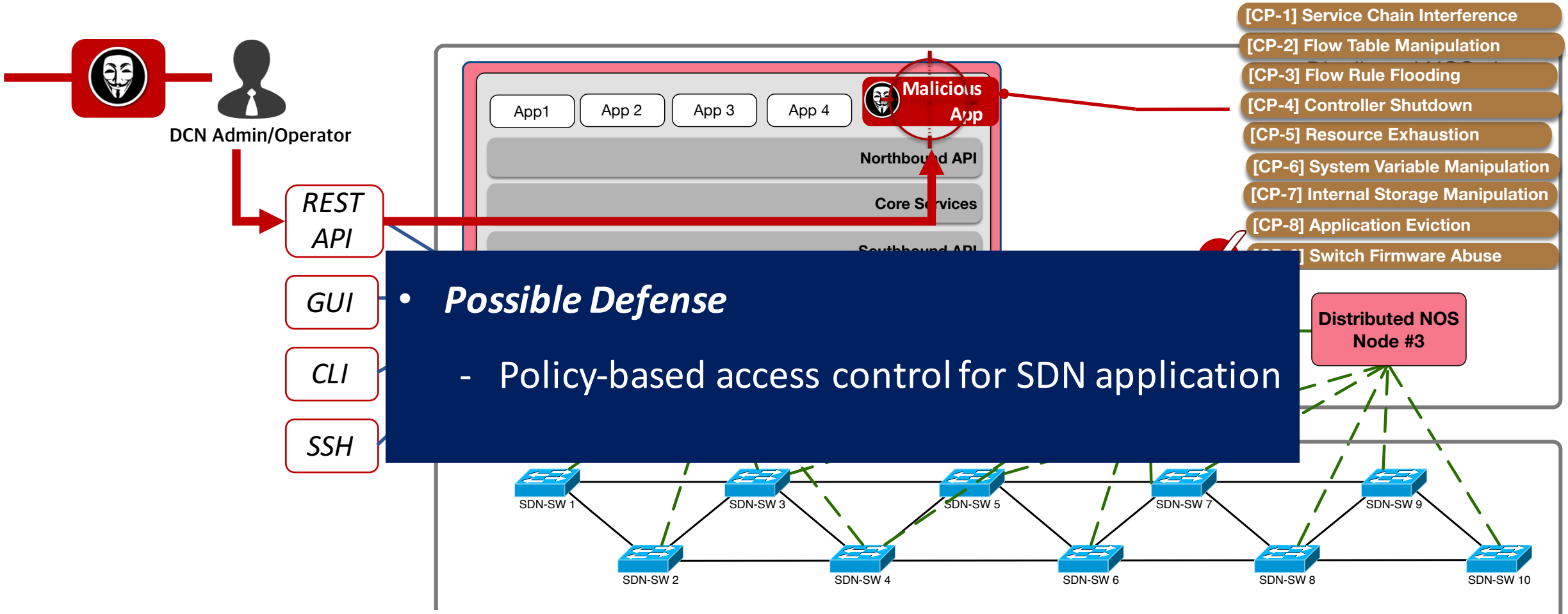
Vulnerability 2. No authentication of NOS cluster nodes



- ***Possible Defense***
  - PKI-based authentication for the NOS components

- The malicious node can completely take over the control of the entire control plane and the network

Vulnerability 3. No application access control

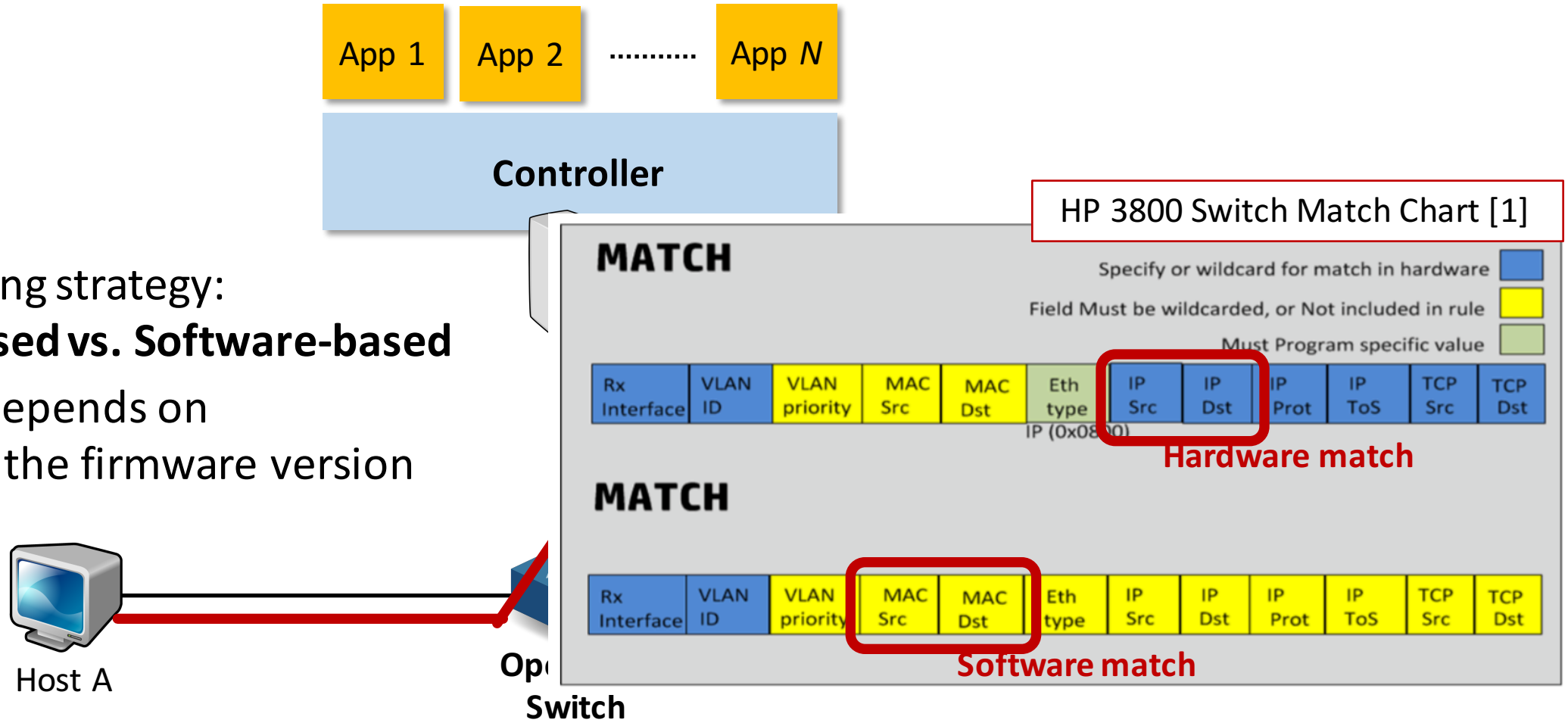


- ***Possible Defense***
  - Policy-based access control for SDN application

- SDN applications are granted very powerful authority; need to limit

Vulnerability 4. Switch device firmware abuse

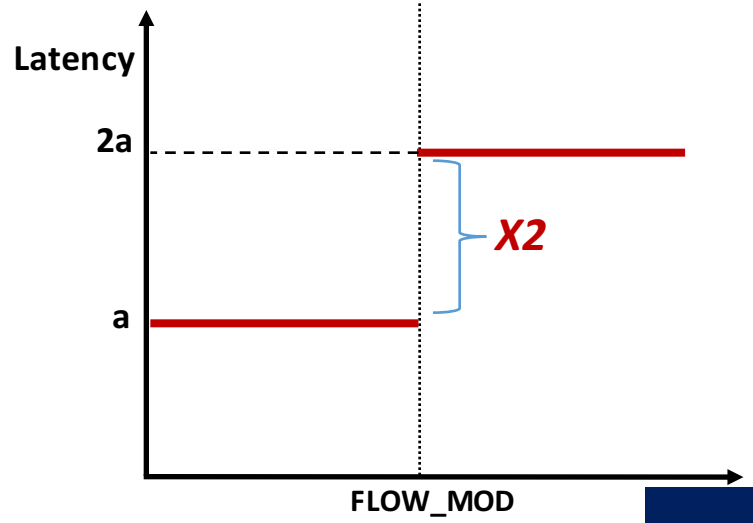
- Packet matching strategy:
  Hardware-based vs. Software-based
- The strategy depends on
  the vendor or the firmware version

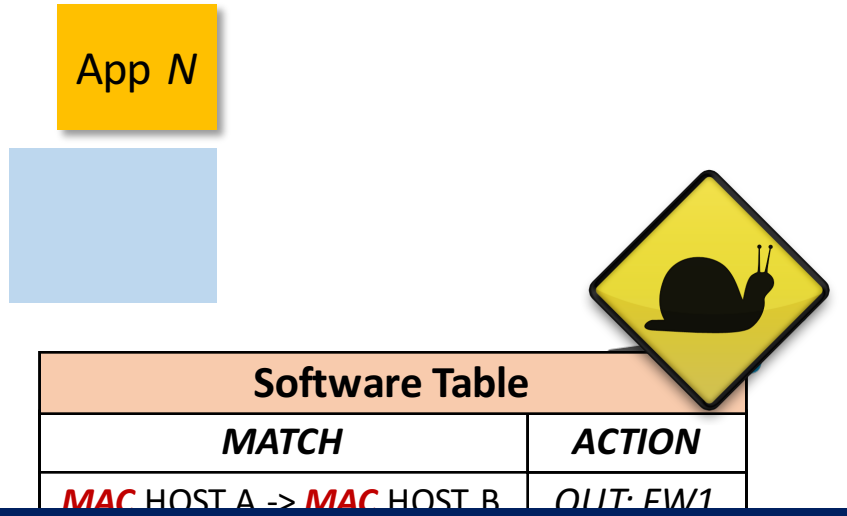


[1] HP. HP Switch Software OpenFlow Administrator's Guide K/KA/WB 15.14

Vulnerability 4. Switch device firmware abuse

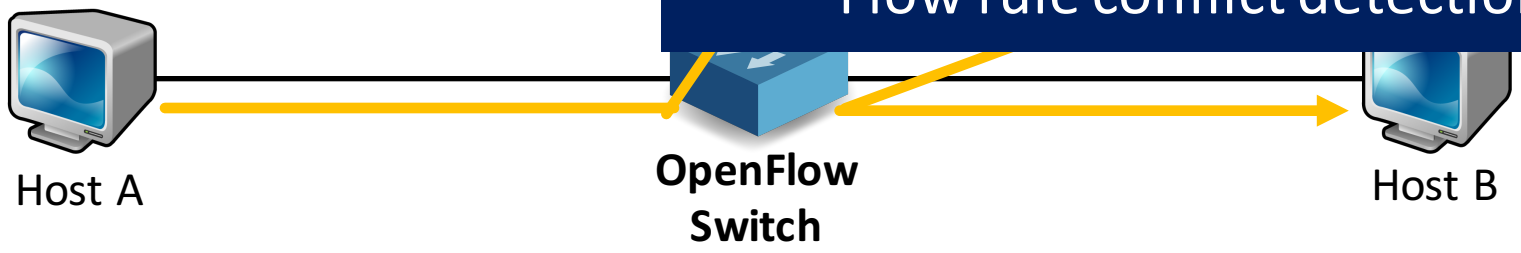
Network performance degradation





- Override IP matching flow rules with MAC matching flow rules!

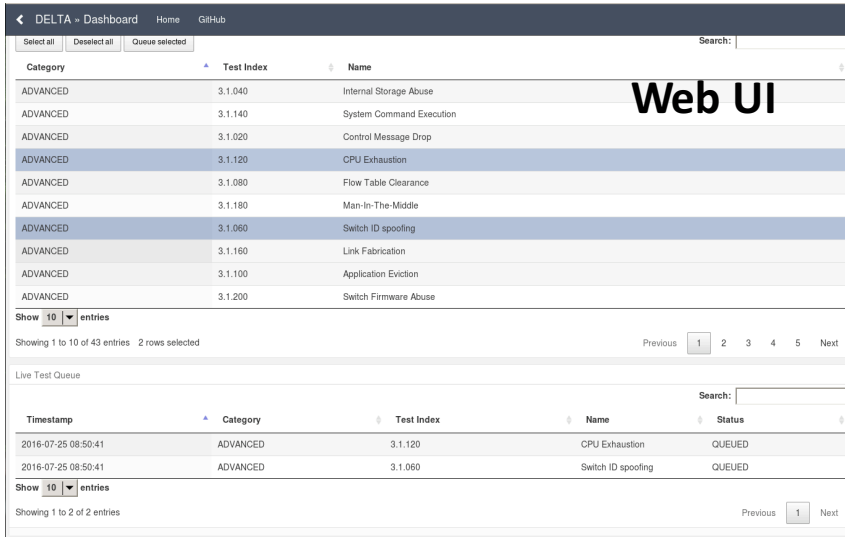
- ***Possible Defense***
  - Flow rule conflict detection & arbitration



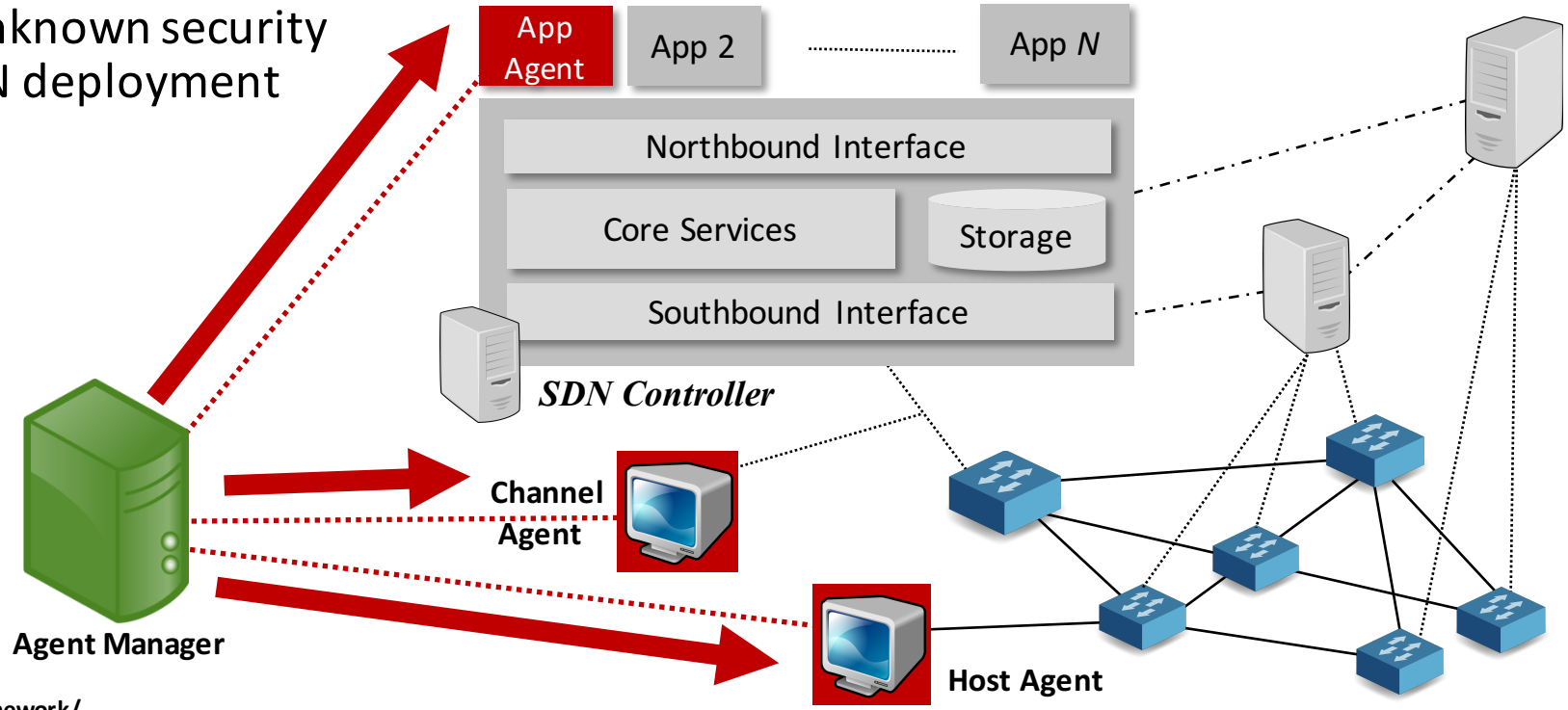
SDN Security Assessment: Project DELTA

- Delta (collaborate with ONF) is a new SDN security evaluation framework with two main functions [1]:
  1. Automatically instantiates known attack cases against SDN elements across diverse environments
  2. Assists in uncovering unknown security problems within an SDN deployment

OPEN NETWORKING FOUNDATION



Web UI



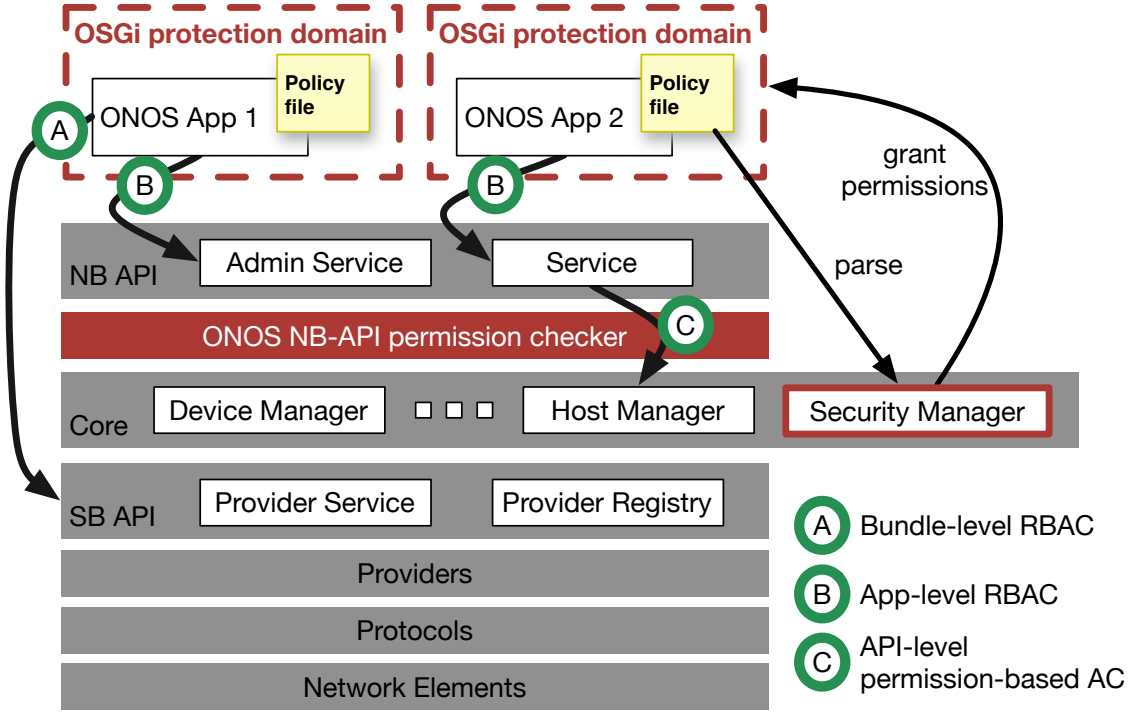
[1] http://opensourcesdn.org/projects/project-delta-sdn-security-evaluation-framework/

SDN Application security policy enforcement

- Security-Mode ONOS
  - Inspired by Mobile application security mechanisms
  - Constrains ONOS (SDN) applications' behavior
    - A security policy per app
  - Detects and blocks security policy violations at runtime

ONOS Open Network Operating System

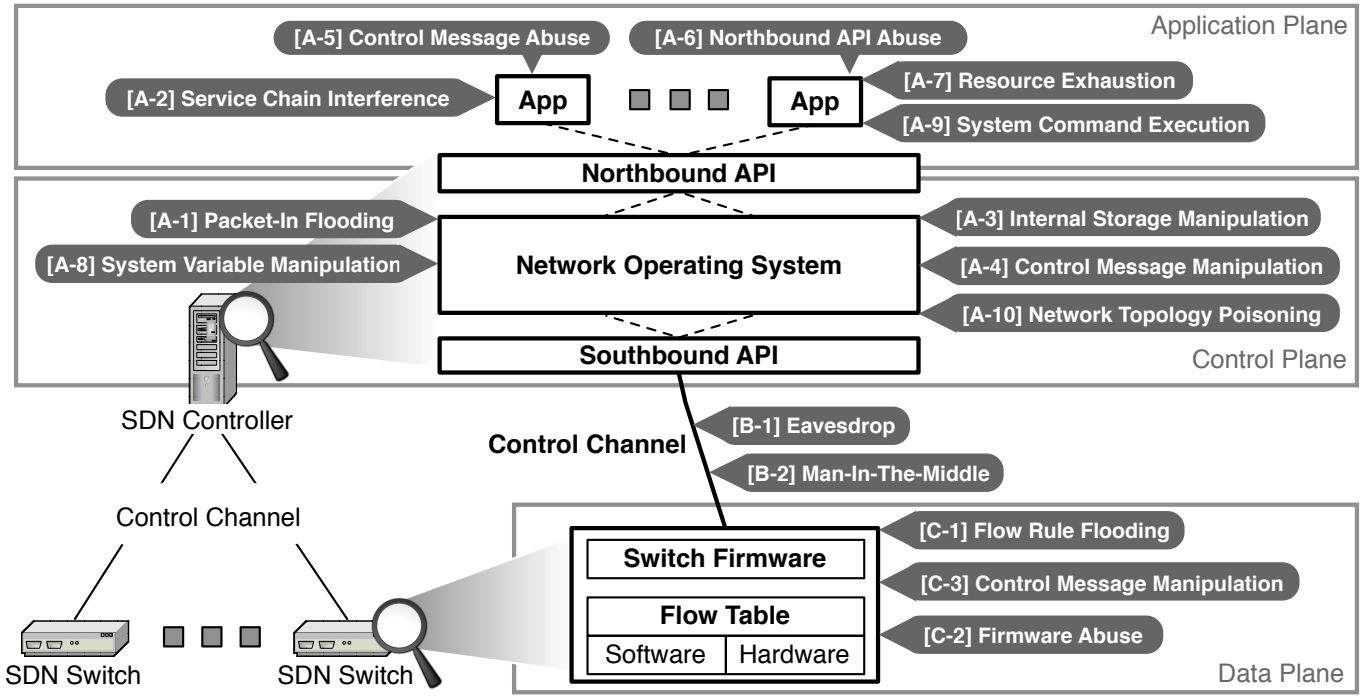
```
Application name: org.onosproject.attack
Application role: USER

Developer specified permissions:
        [APP PERMISSION] HOST_EVENT
        [APP PERMISSION] DEVICE_READ
        [APP PERMISSION] FLOWRULE_WRITE
        [APP PERMISSION] INTENT_READ
        [APP PERMISSION] INTENT_WRITE
        [CLI SERVICE] org.apache.karaf.shell.console.CompletableFunction(register)
        [CLI SERVICE] org.apache.karaf.shell.commands.CommandWithAction(register)
        [CLI SERVICE] org.apache.felix.service.command.Function(register)
        [CLI SERVICE] org.osgi.service.blueprint.container.BlueprintContainer(register)
        [Other SERVICE] org.onosproject.attack.Attack(get,register)
        [SB SERVICE] org.onosproject.net.link.LinkProviderRegistry(get,register)
        [CRITICAL PERMISSION] RuntimePermission exitVM.0 ()
```



SDNSecurity.org *(will open in 09/2016)*

- We try to discover SDN specific vulnerabilities and devote to systematizing and characterizing all related points.
- Currently, we have 8 on-going projects and 8 finished projects.

http://sdnsecurity.org



SDN Vulnerability Genome Project

Final remarks

- Are we ready for the next-gen networking?
  - **No**, not yet at least from a security point of view
- *A **LOT** of work still needs to be done to improve the security of SDN.*
- Your urgent attention is needed!

Thank you