Identifier Binding Attacks and Defenses in Software-Defined Networks

Samuel Jero[1], **William Koch**[2], **Richard Skowyra**[3], **Hamed Okhravi**[3], **Cristina Nita-Rotaru**[4], **and David Bigelow**[3]

[1]Purdue University, [2]Boston University,
[3]MIT Lincoln Laboratory, and [4]Northeastern University

USENIX Security 2017

DISTRIBUTION STATEMENT A. Approved for public release: distribution unlimited. This material is based upon work supported by the Department of Defense under Air Force Contract No. FA8721-05-C-0002 and/or FA8702-15-D-0001. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Department of Defense.© 2017 Massachusetts Institute of Technology. Delivered to the U.S. Government with Unlimited Rights, as defined in DFARS Part 252.227-7013 or 7014 (Feb 2014). Notwithstanding any copyright notice, U.S. Government rights in this work are defined by DFARS 252.227-7013 or DFARS 252.227-7014 as detailed above. Use of this work other than as specifically authorized by the U.S. Government may violate any copyrights that exist in this work.

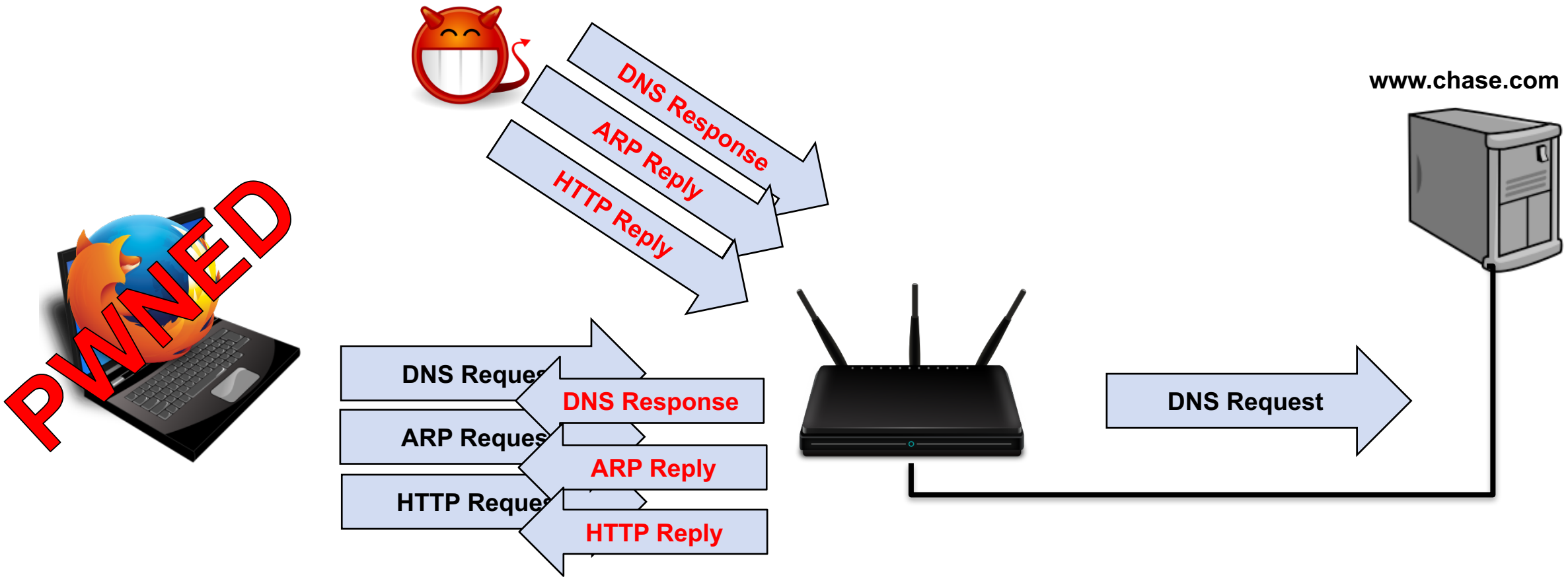
PURDUE UNIVERSITY

BOSTON UNIVERSITY

LINCOLN LABORATORY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Northeastern University

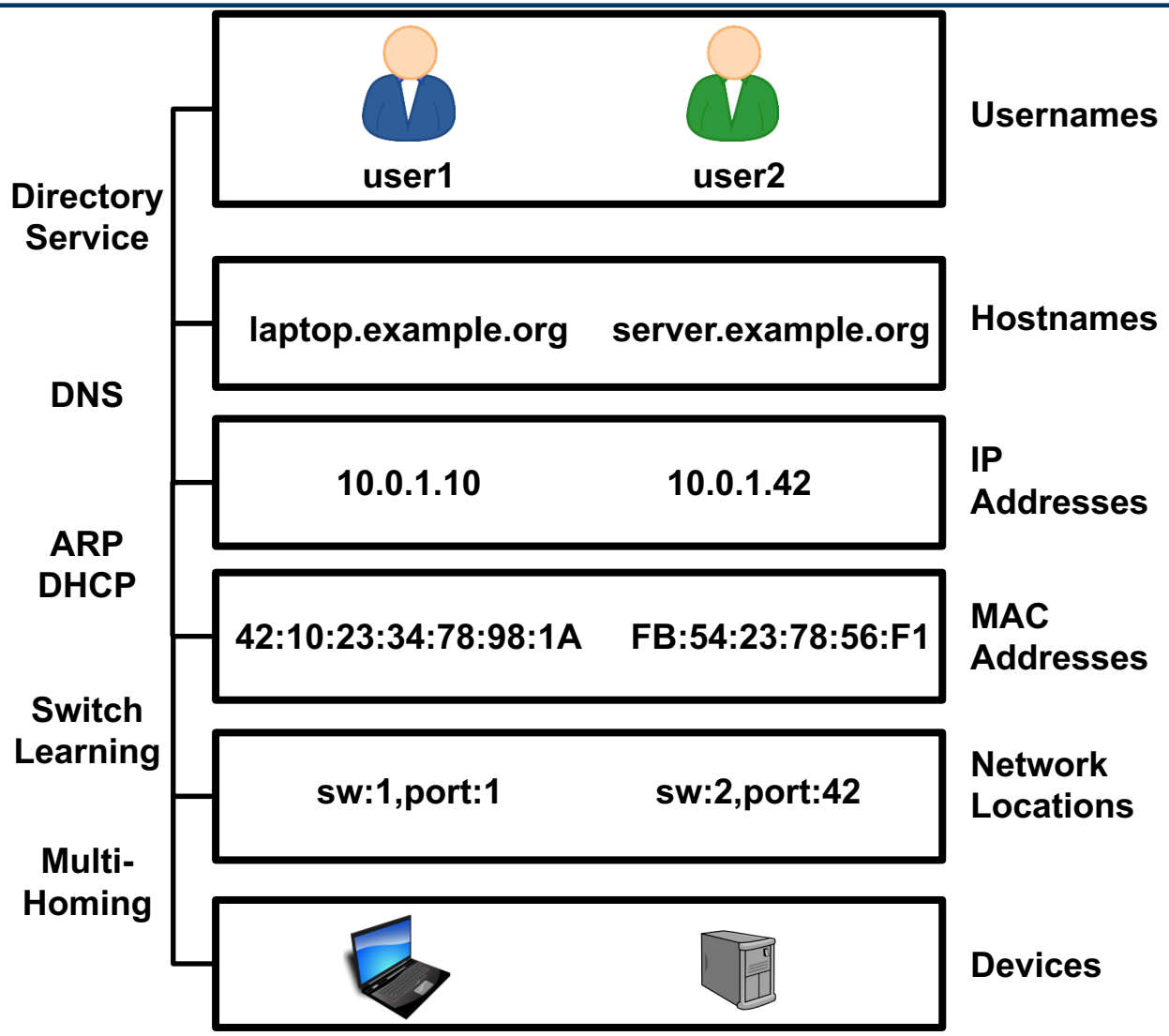
A Day in the Life Of Your Browser



Insecure identifier bindings make this possible

Network Identifiers and Their Bindings

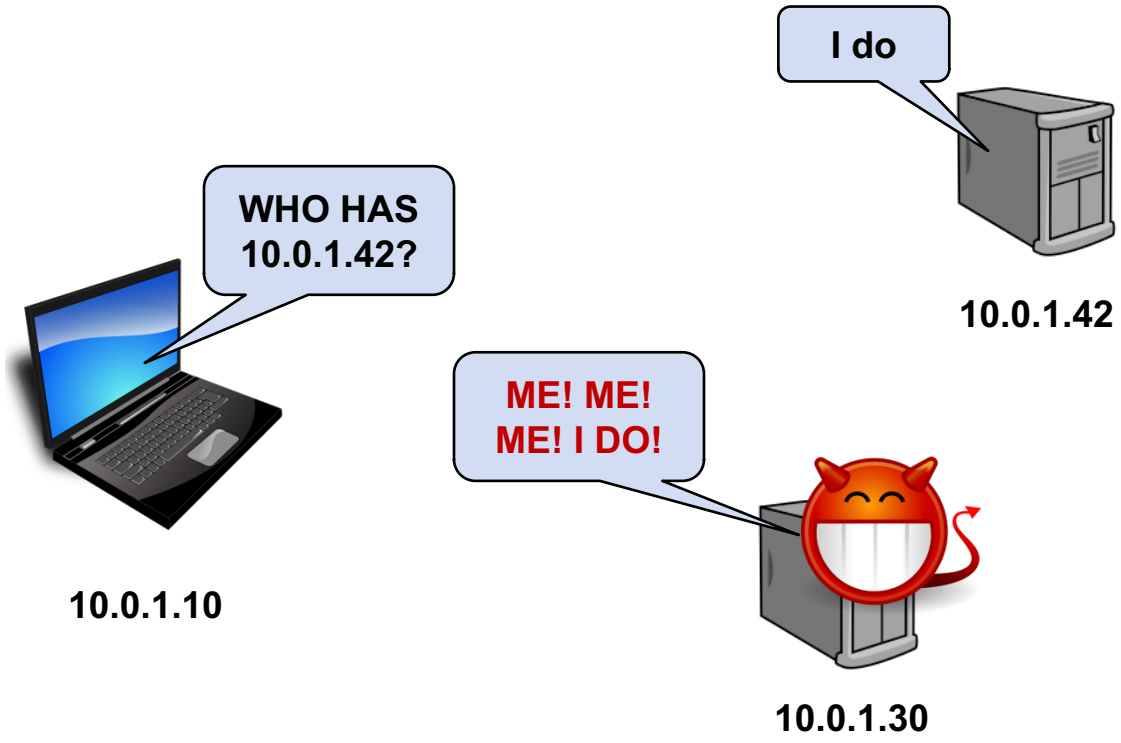
- Modern networks have many protocols layered on top of each other
- Network Identifier: An identifier for a device used at some layer of the network stack
  - Examples: IP addresses, MAC addresses, Hostnames
  - Used for forwarding as well as access control and authorization
- Devices need to bind identifiers from higher layers to lower layers
  - ARP, DHCP, DNS, Active Directory



Bindings Performed by Insecure Protocols

- No authentication
- Binding protocols often rely on simple broadcast queries
- No cross-layer checks
- No additional checks on binding updates
- Mutable Identifiers

Enables:
- Host Impersonation
- Man-In-The-Middle
- Privilege Escalation
- Denial of Service



ARP Poisoning

Existing Defenses

- **Port Security**
  - Limits number of MAC addresses per switch port
  - Heuristically blocks attackers spoofing MAC addresses
- **Cisco Dynamic ARP Inspection (DAI)**
  - Compares ARP responses with internal DHCP server records
  - Requires use of internal DHCP server
- **DHCP Snooping**
  - Trusted / Untrusted zones
  - DHCP packets from trusted zone used to setup bindings to filter untrusted zone packets
- **DNSSEC**
  - Prevents forged

Ad-hoc, limited to MAC layer, manual configuration

Limited to ARP, no protection for static IPs, manual configuration

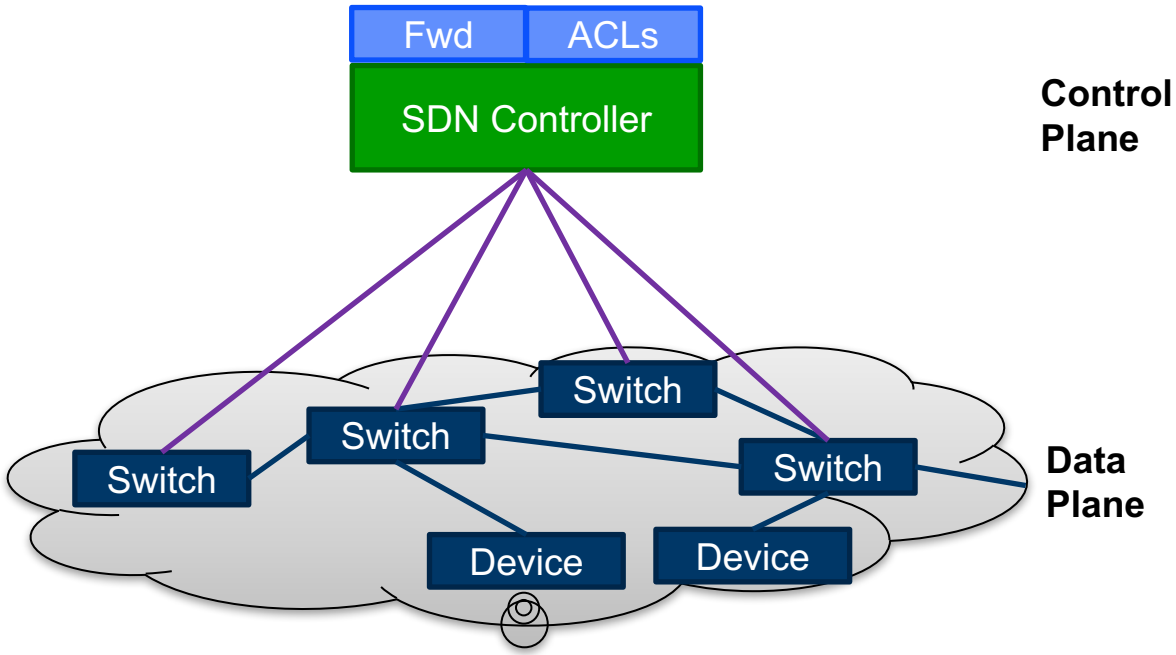
Limited to DHCP, no protection for static IPs, manual configuration

Limited to DNS, rogue servers can still exist, manual and complex configuration

These are ad-hoc solutions specific to particular identifiers and requiring manual configuration

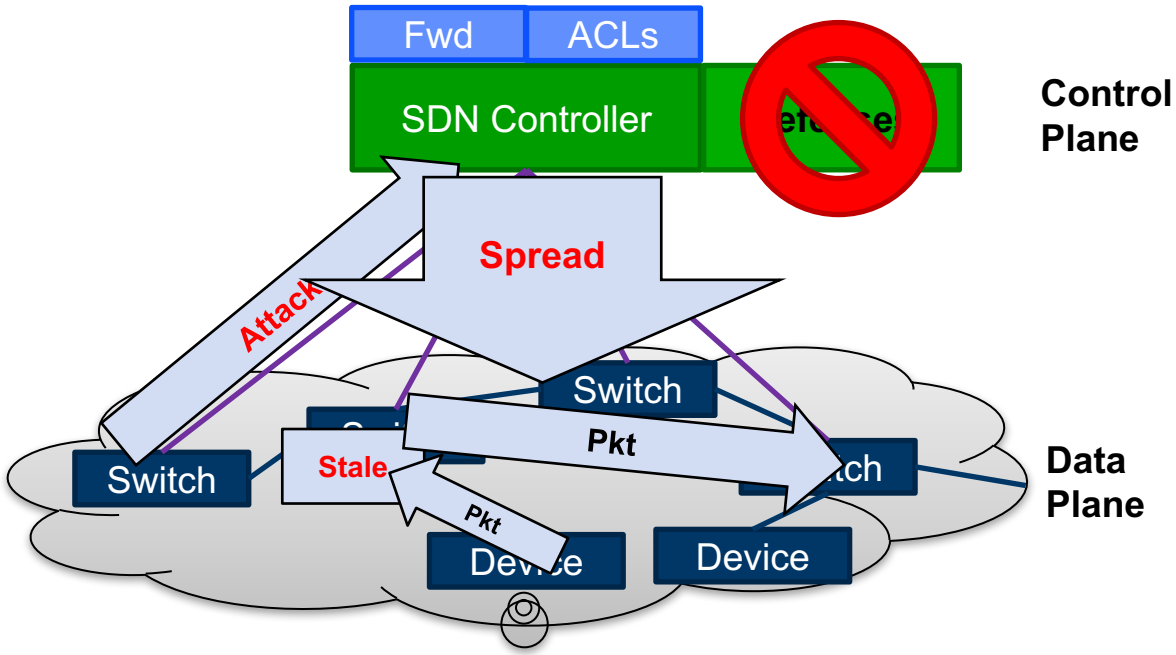
Identifier Binding in Software Defined Networks

- **Unified control plane**
  - Single binding table for entire network
- **Bare Metal Switches**
  - Switches do not include existing defenses against binding attacks
- **Delayed Flow Rule Consistency**
  - Temporary inconsistencies between controller and switches
  - Flow rules are not instantly removed from switches when controller state changes



Binding Attacks in Software Defined Networks

- **Unified control plane**
  - Attackers *anywhere* in the network can poison *any* binding
- **Bare Metal Switches**
  - Existing defenses must be implemented in the controller
  - Most controllers have not implement them
- **Delayed Flow Rule Consistency**
  - Attackers can cause a few packets to be forwarded with stale state



Binding Attacks Have Significantly Amplified Power In SDNs

Existing Software Defined Network Defenses

- **Ethane (SIGCOMM 07)**
  - **Fine grained acces**
  - **Leverages binding**
- **TopoGuard (NDSS 15)**
  - **Demonstrated attac**
  - **Defense based on**
- **SPHINX (NDSS 15)**
  - **Demonstrated attac**
  - **Defense by ensurin**

Vulnerable to MAC address spoofing, does not protect IP-hostname or hostname-user bindings

Vulnerable to MAC address spoofing, limited to MAC-location binding

Vulnerable to MAC address spoofing, limited to MAC-location binding

These solutions are focused only on specific identifiers, leaving other bindings vulnerable

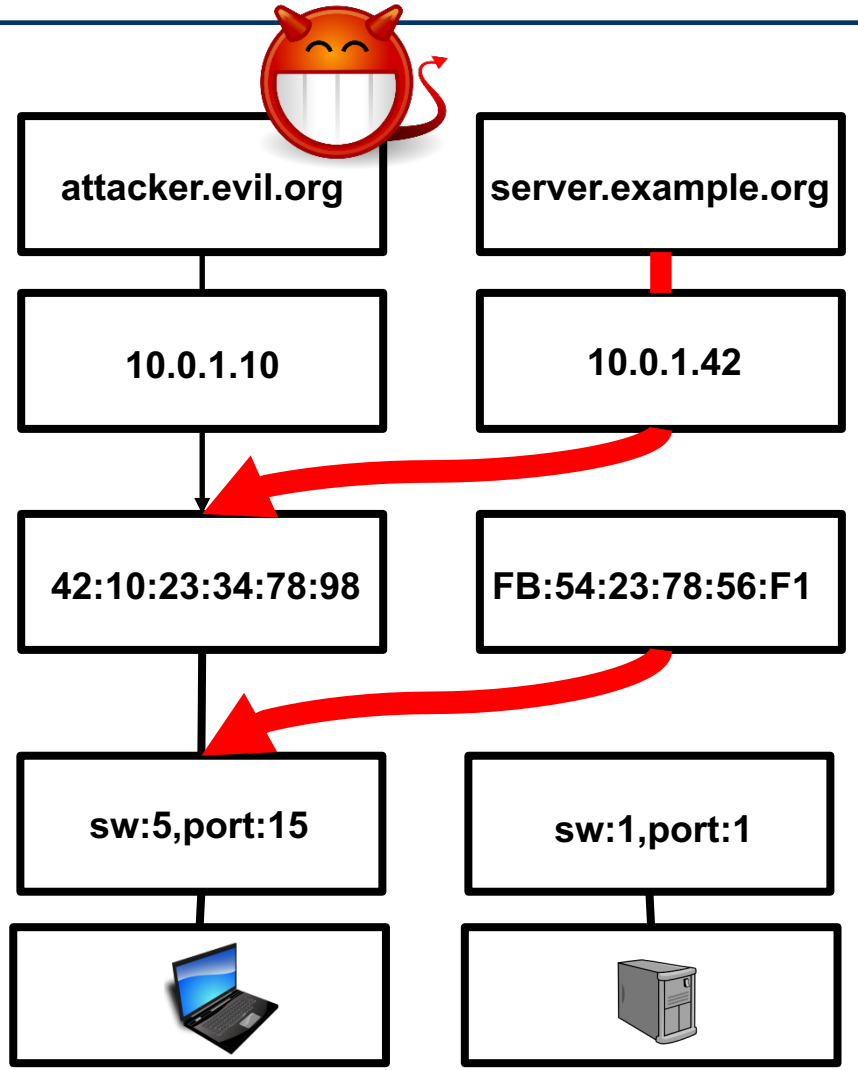
This Paper Focuses On

- **A new, more powerful binding attack**
  - **The Persona Hijacking Attack**
  - **Takeover of *ALL* Identifiers**
  - **Persistent**
- **A defense against *ALL* binding attacks**
  - **SecureBinder**
  - **Mediate Bindings**
  - **Root-of-trust**



Persona Hijacking

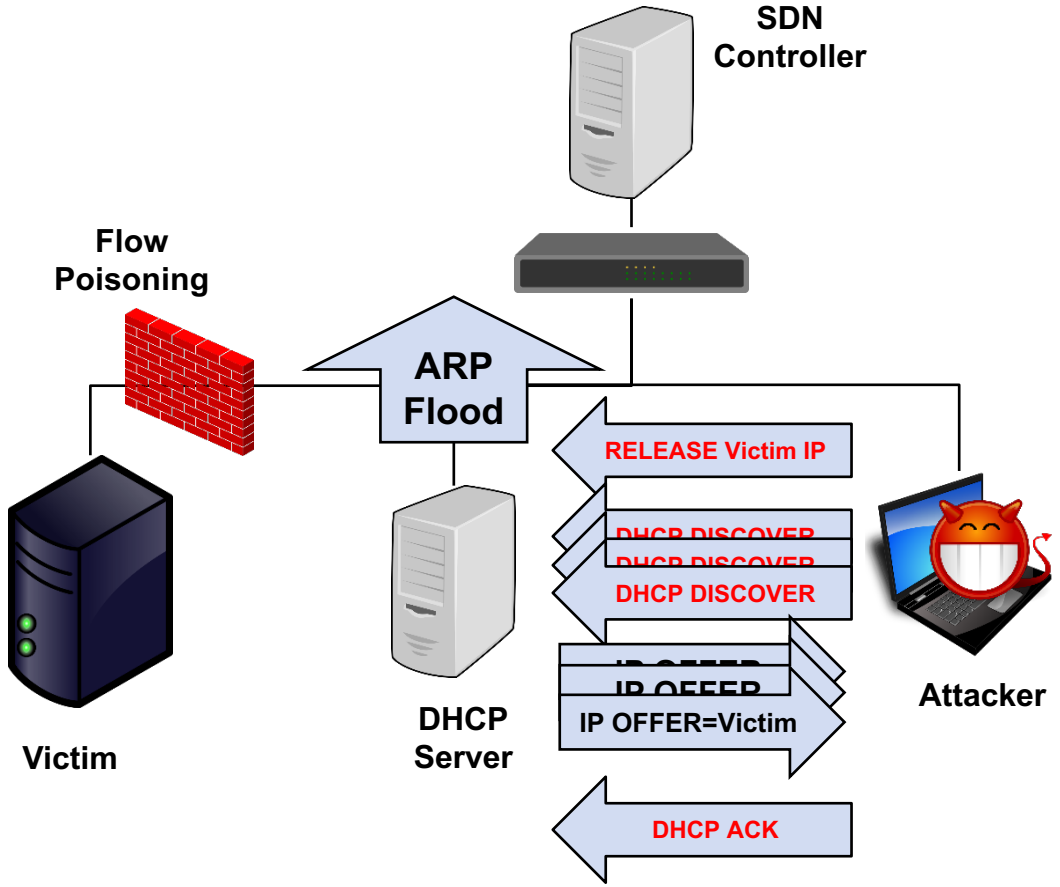
- Takeover of the Victim's IP address and DNS name
- Persists for hours or days
- Progressively breaks the MAC-Location, MAC-IP, and (possibly) IP-hostname bindings
- Attacker becomes the *owner-of-record* for the Victim's IP address
  - Victim appears to be malicious party
- DHCP server is co-opted to propagate this deception further into the network
- Operates in two phases:
  - IP Takeover – Leverage DHCP server to steal IP address and DNS name from Victim
  - Flow Poisoning – Leverage SDN to complete DHCP assignment



IP Takeover

Goal: Steal Victim's IP address and co-opt DHCP server to persist and propagate this deception

- Attacker sends forged DHCP_RELEASE for Victim
- DHCP Server releases Victim's address, but does not notify Victim
- Attacker sends DHCP_DISCOVER messages until Victim's address is offered
- Attacker sends DHCP_REQUEST and launches Flow Poisoning attack
- DHCP server checks offered IP with an ARP request. Victim's response is blocked by Flow Poisoning
- DHCP server completes handshake and Attacker becomes the owner-of-record of the Victim's IP address

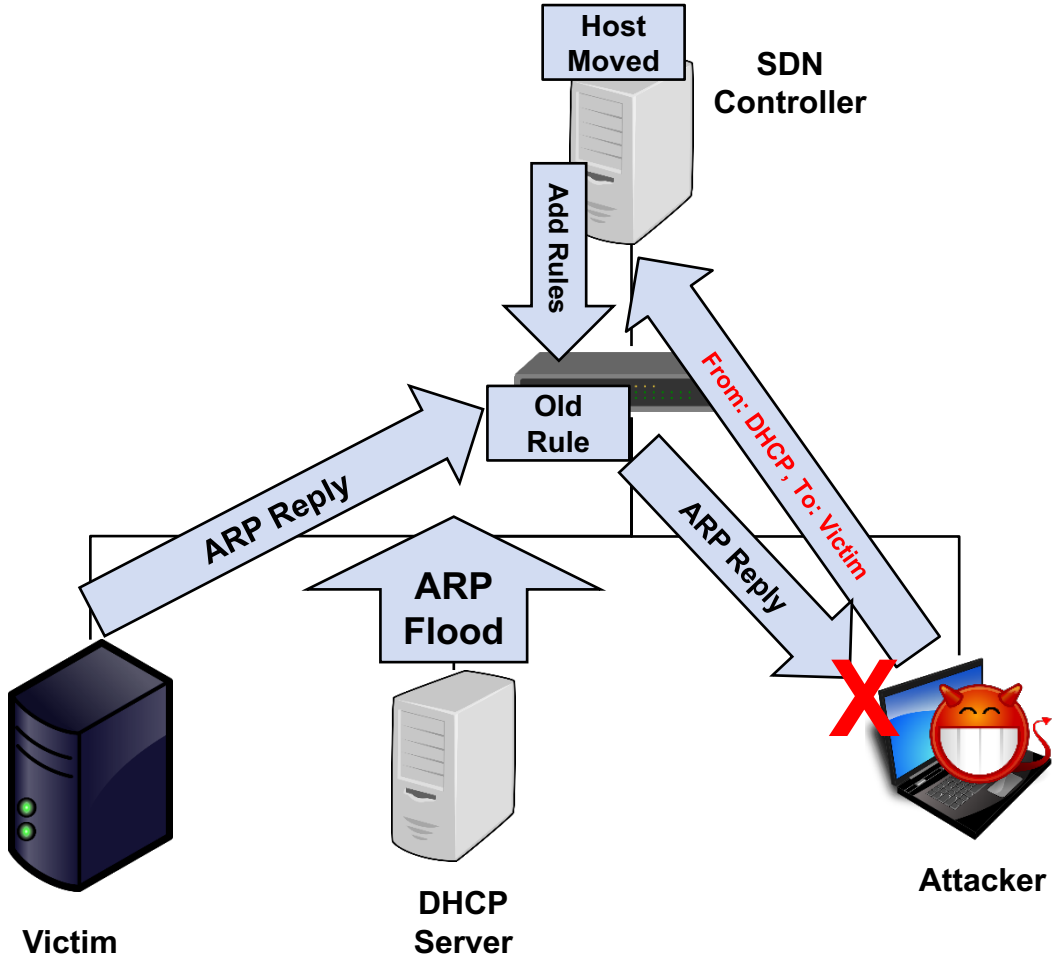


Attacker Owns Victim's IP Address

Flow Poisoning

Goal: Blackhole ARP response from Victim to DHCP server

- Attacker sends spoofed packets from the DHCP server to Victim
- SDN controller believes DHCP server has moved, inserts new flow rules
- DHCP server sends ARP broadcast to check for users of the Victim's IP address
- SDN controller discovers true location of DHCP server, but old flow rules are not removed
  - Removed asynchronously via separate thread or timeouts
- ARP reply from victim hits old flow rules and is sent to Attacker, not DHCP server
- DHCP server assigns Victim's IP address to Attacker



Persona Hijacking In The Wild

- We demonstrated Persona Hijacking against ONOS and Ryu
  - Emulated Mininet environment
  - Persona Hijacking succeeded against both
- Source code analysis suggests that POX and Floodlight are also vulnerable

Controller	Experimentally Vulnerable	Probably Vulnerable	Not Vulnerable
ONOS	✗		
Ryu	✗		
POX		✗	
Floodlight		✗	

Outline

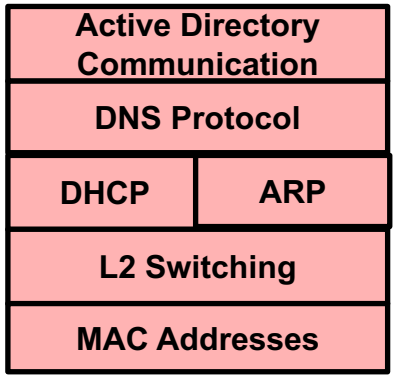
- Background
- SDN
- Persona Hijacking Attack
  - Takeover of *ALL* Identifiers
  - Persistent
- **SecureBinder: Designing A Defense**
  - **Mediate Bindings**
  - **Root-of-trust**
- **Evaluation**
- **Summary**



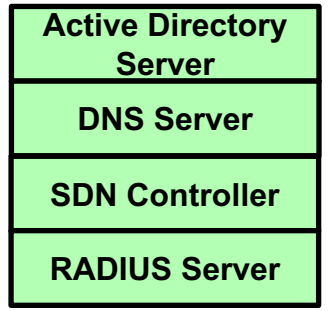
SecureBinder: Design

Goals:

- **Isolate identifier binding control traffic from the data-plane**
  - **Preventing attackers from observing and responding to queries**
  - **Mediate all bindings and answer queries**
- **Validate changes to existing bindings**
  - **Preventing attackers from poisoning these mappings and impersonating other hosts**
  - **Use a global network view to detect and resolve conflicts**
- **Validate binding control traffic**
  - **Preventing attacker from using forged lower-layer identifiers to change higher-layer bindings**
  - **Ensure that lower-layer bindings are valid before processing binding control traffic**
- **Protect readily-changed root identifiers (MAC Addresses)**
  - **Preventing attackers from impersonating known, but powered-off, devices**
  - **Use IEEE 802.1x to provide a root-of-trust for our network identifiers**



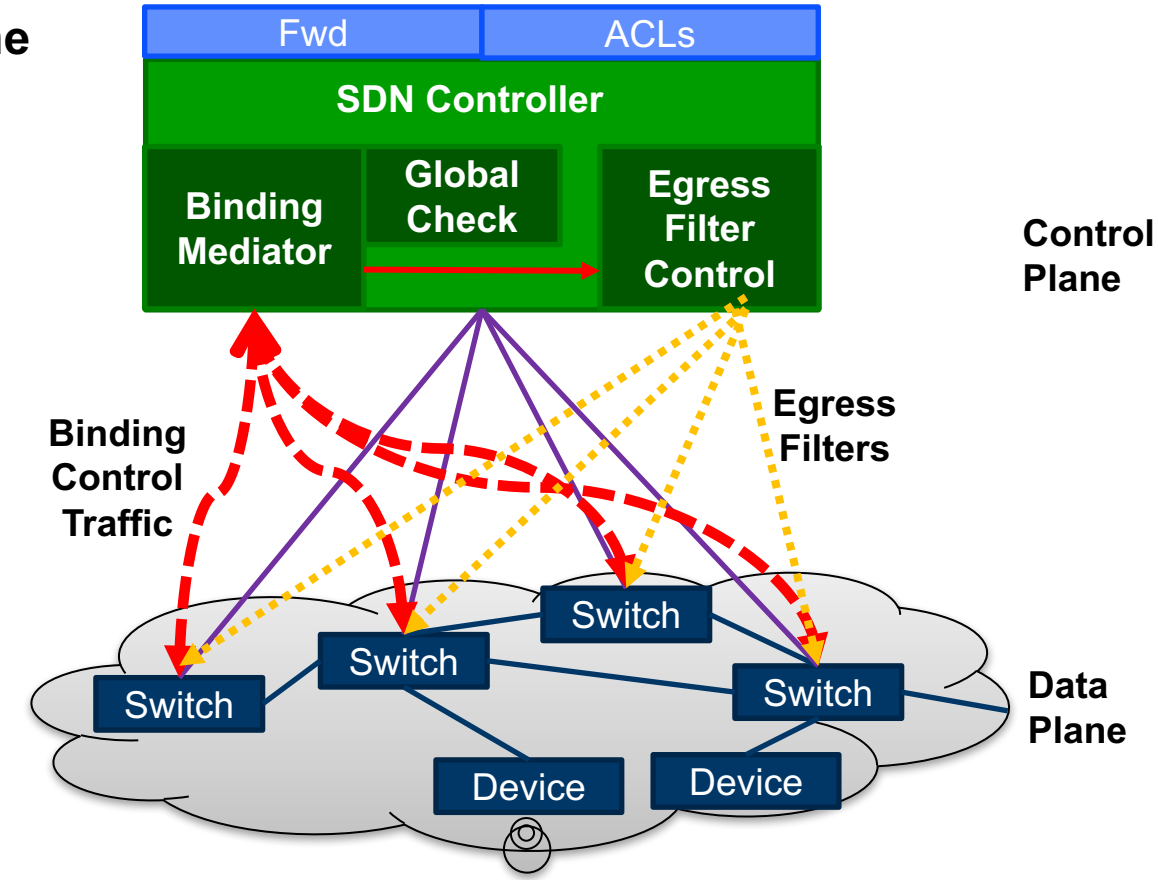
No trust required



Trusted

SecureBinder: Leveraging SDN

- **Separate control-plane and data-plane**
  - Send binding control traffic to the control-plane
  - Seamlessly interpose on all binding protocols and requests
    - ARP, DHCP, DNS, NETBIOS, Active Directory
  - Eliminates broadcast requests
- **Global view of the network**
  - Validate bindings by ensuring that identifiers are unique and binding requests come from expected locations
  - Validate all layers of binding control traffic
- **Programmatic control of the network**
  - Efficiently prevent all spoofed packets using PER PORT egress (outbound) filters
  - Painless for administrators



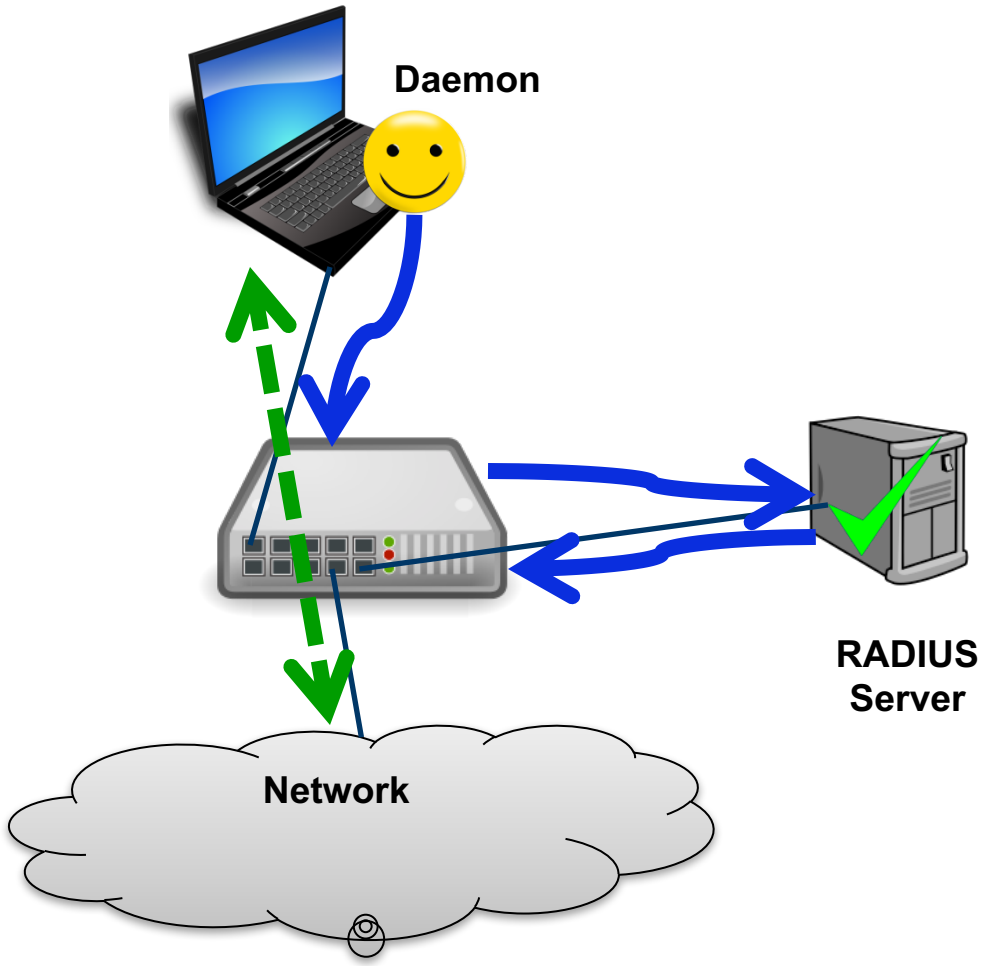
Need to Protect MAC Addresses

- **MAC addresses are easily modified**
  - **Attacker can trivially clone the MAC address of a victim device**
  - **Network cannot tell the difference between the legitimate device and a cloned MAC address**
- **MAC addresses are often equated with particular devices**
  - **i.e. The CEO's Laptop, My Desktop, the GIT server**
  - **Commonly used in or for Access Control**
- **A global network view does not help**
  - **Can ensure that MAC address is at exactly one place in the network**
  - **Cannot ensure it corresponds to the device we expect**
- **Need a root-of-trust**

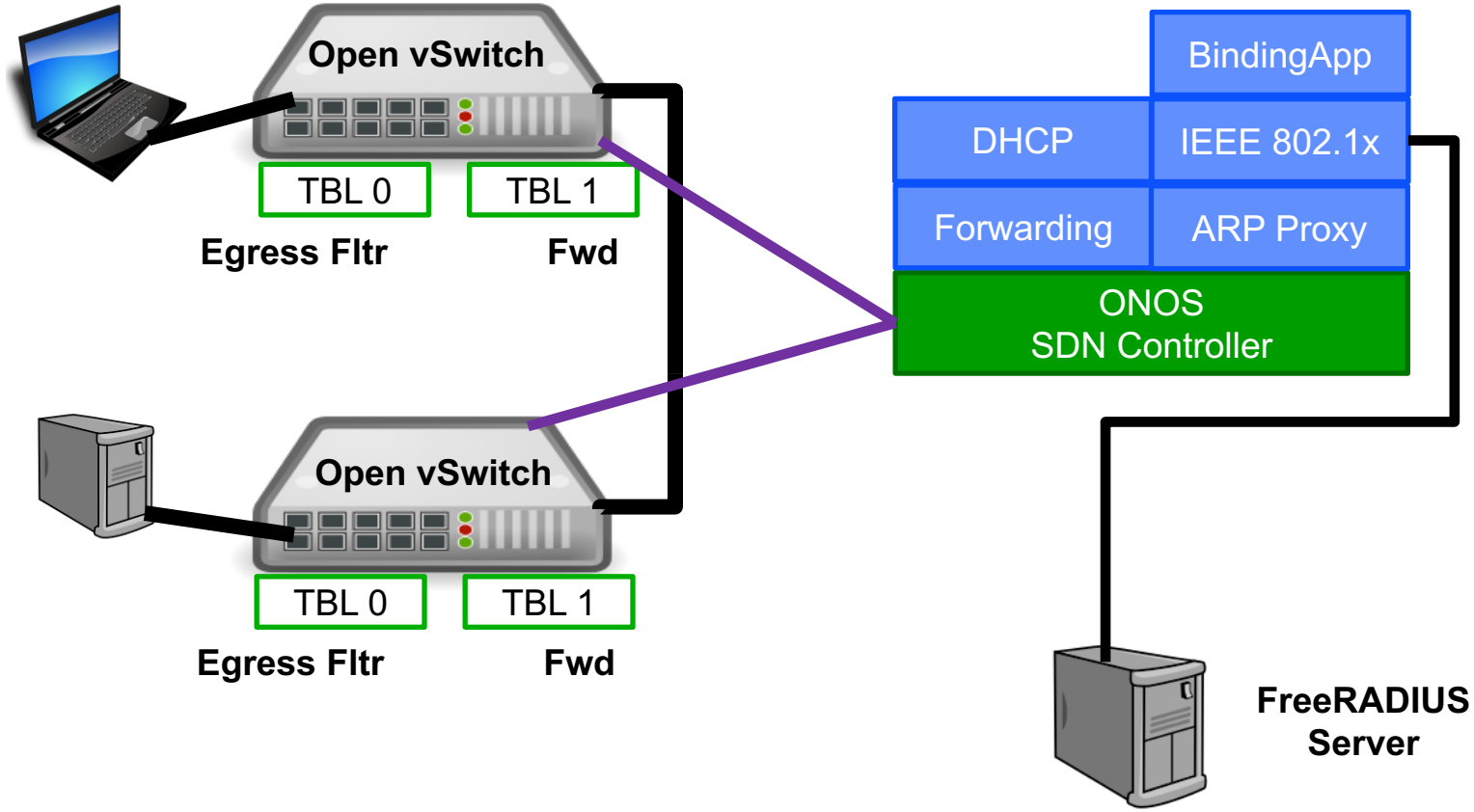
While not needed to prevent Persona Hijacking or ARP Poisoning, protecting MAC addresses is required to completely eliminate identifier binding attacks

An IEEE 802.1x Solution

- IEEE 802.1x provides cryptographic assurance that a host is authorized to access the network
  - Optional extension ensures that it has the MAC address we expect
- Supported by all major OSes and switches
  - Same technology as WPA2 Enterprise



Implementation



- **OF 1.3 Multiple Flow Tables**
  - **Table 0 is egress filtering and binding traffic separation**
  - **Table 1+ is forwarding, etc**
- **Based on the ONOS SDN controller**
  - **Version 1.5.1**
  - **Including Apps: Forwarding, ARP Proxy, DHCP**
- **New Apps**
  - **Binding Security App maintaining verified bindings**
  - **IEEE 802.1x app heavily modified**

Security Evaluation

Formal:

- Model checking analysis using SPIN
- Modeled ARP and DHCP
- 7 correctness invariants
- Found 6 attacks without SecureBinder
- No attacks found with SecureBinder

See the paper for more details

Experimental:

- Tested with Mininet, which provides an emulated SDN network environment
- Launched three identifier binding attacks against ONOS and SecureBinder

Attack	ONOS	SecureBinder
Persona Hijacking	✗	✓
ARP Spoofing	✗	✓
Host Location Hijacking	✗	✓

Attacks Stopped by ONOS and SecureBinder

Performance Evaluation

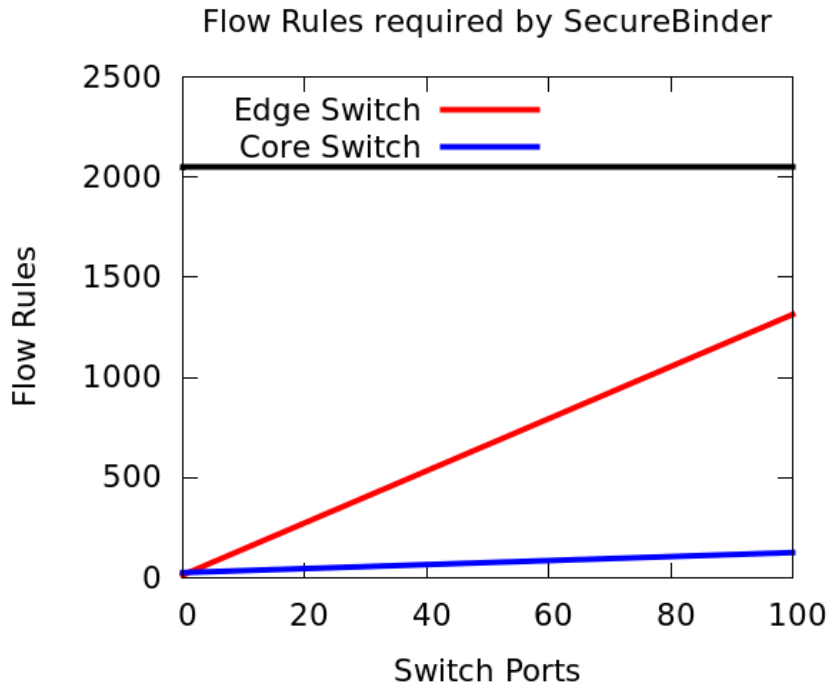
Latency and Controller Load:

- Host Join Latency: time from host connected to network to operational network
- New Flow Latency: time to start a new flow
  - Only the first packet will be impacted
- Approximate controller load based on number of pkt_in events

	ONOS	SecureBinder
Host Join Latency	505ms	3505ms (+3sec)
New Flow Latency	8ms	6ms (-2ms)
Pkt_in's (Load)	131	193 (+47%)

Flow Rules:

- Limited switch resource
- Typical switch has ~2-8K slots



Testing done with Mininet, which provides an emulated SDN network environment

Summary

- We demonstrated the power of identifier binding attacks in SDNs by developing a new attack called Persona Hijacking that progressively breaks multiple bindings to hijack a Victim's IP address and hostname persistently and co-opts the network infrastructure to propagate that deception
- We showed that this attack is effective against ONOS and Ryu
- We then developed a defense called SecureBinder that systematically and completely prevent all identifier binding attacks at multiple layers of the network stack by leveraging the programmatic control and global view of the network in SDN and a root-of-trust provided by IEEE 802.1x
- We showed that this defense is effective against 3 identifier binding attacks, including Persona Hijacking, and that its performance overhead is acceptable

Questions?

