VLAN Hopping, ARP Poisoning & Man-In-The-Middle Attacks in Virtualized Environments

Ronny Bull
Dr. Jeanna Matthews
&
Kaitlin Trumbull

(DEF CON 24)

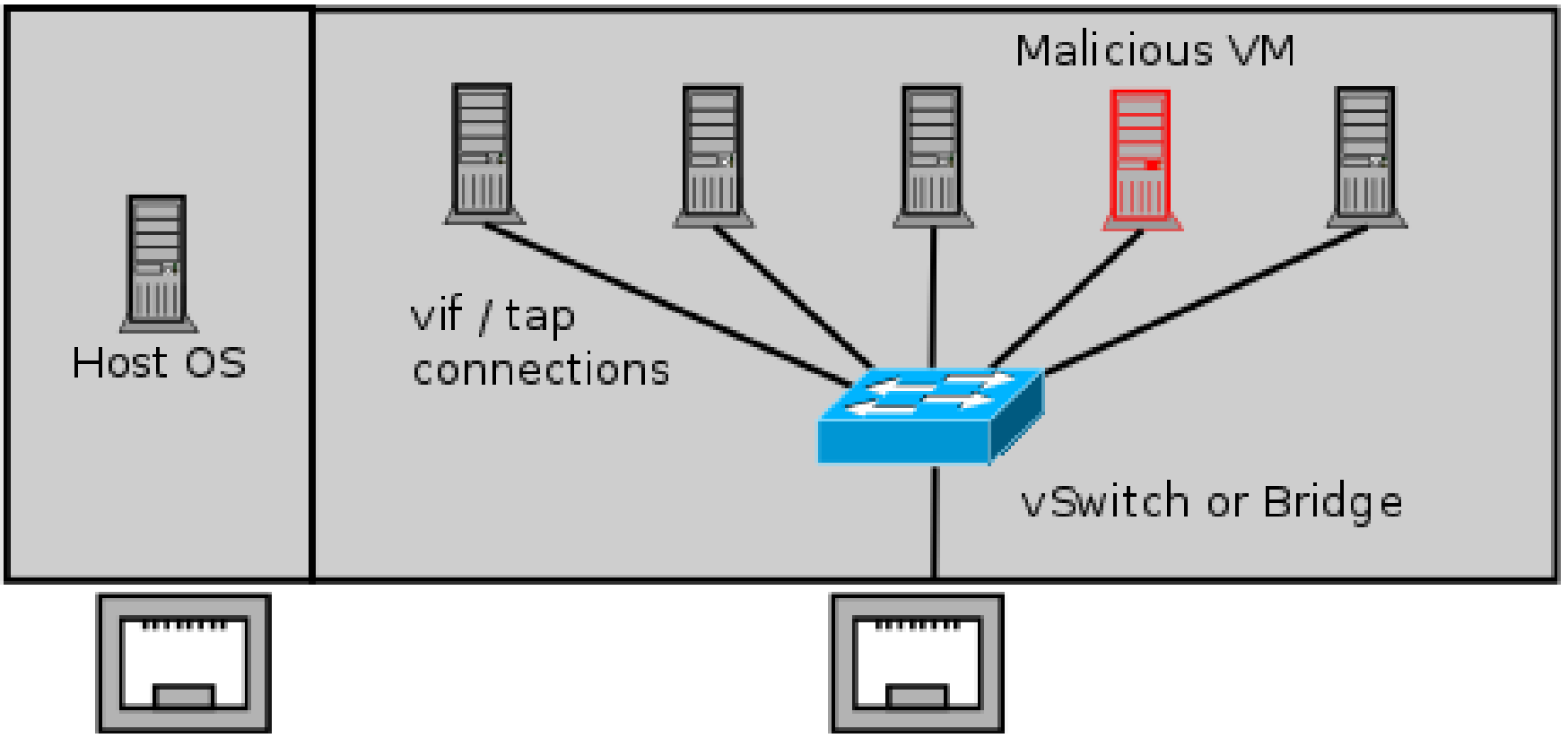
Road Map

- Context for the Problem of Layer 2 Network Security in Virtualized Environments
  - Virtualization, Multi-tenant environments, Cloud services
- Test platforms
  - Array of virtual networking implementations tested
- Specific attacks and results
  - MAC Flooding, DHCP Attacks *(previously discussed at DEF CON 23*
  - VLAN Hopping, ARP Poisoning *(this talk)*
- Conclusions

Key Question

- All client virtual machines hosted in a multi-tenant environment are essentially connected to a virtual version of a physical networking device. So do Layer 2 network attacks that typically work on physical devices apply to their virtualized counterparts?
- Important question to explore:
  - All cloud services that rely on virtualized environments could be vulnerable
  - This includes data centers hosting mission critical or sensitive data!
- Not the only class of attacks from co-located VMs
- Old lesson: vulnerable to those close to you

What If?



Bottom Line

- Our research ***proves*** that *virtualized network devices* **DO** have the potential to be exploited in the same manner as physical devices.
- In fact some of these environments allow the attack to leave the ***virtualized network*** and affect the ***physical networks*** that they are connected to!

Consequences

- So what if a malicious tenant successfully launches a Layer 2 network attack within a multi-tenant environment?
  - Capture all network traffic
  - Redirect traffic
  - Perform Man-in-the-Middle attacks
  - Denial of Service
  - Gain unauthorized access to restricted sub-networks
  - Affect performance

Test Scenarios & Results

- MAC Flooding Attack
  - Performance evaluation updates since our last talk
- VLAN Hopping
  - Attack Scenario Descriptions
  - Summary of Results
- ARP Poisoning
  - Man-In-The-Middle Attacks
  - Summary of Results

Old Test Environment

Built from what we could salvage

(RIP – you served us well!)

Old Hardware Specs

	Hardware Specs			
Platform	**CPU Type**	**Memory Size**	**Hard Disk**	**NICs**
OS Xen w/ Linux Bridging	Xeon 3040	4 GB	500 GB	2
OS Xen w/ Open vSwitch 1.11.0	Xeon 3040	4 GB	500 GB	2
OS Xen w/ Open vSwitch 2.0.0	Xeon 3040	4 GB	500 GB	2
Citrix XenServer 6.2	Xeon 3040	4 GB	500 GB	2
MS Server 2008 R2 w/Hyper-V	Xeon 5140	32 GB	145 GB	2
MS Hyper-V 2008 Free	Xeon 5140	32 GB	145 GB	2
VMware vSphere (ESXi) 5.5	Xeon E3-1240	24 GB	500 GB	2

(Full system specs are provided in the white paper on the DEF CON 23 CD, and are also availabe on the DEF CON Media Server)

New Environment

(After 30K of funding. Thanks Utica College!)

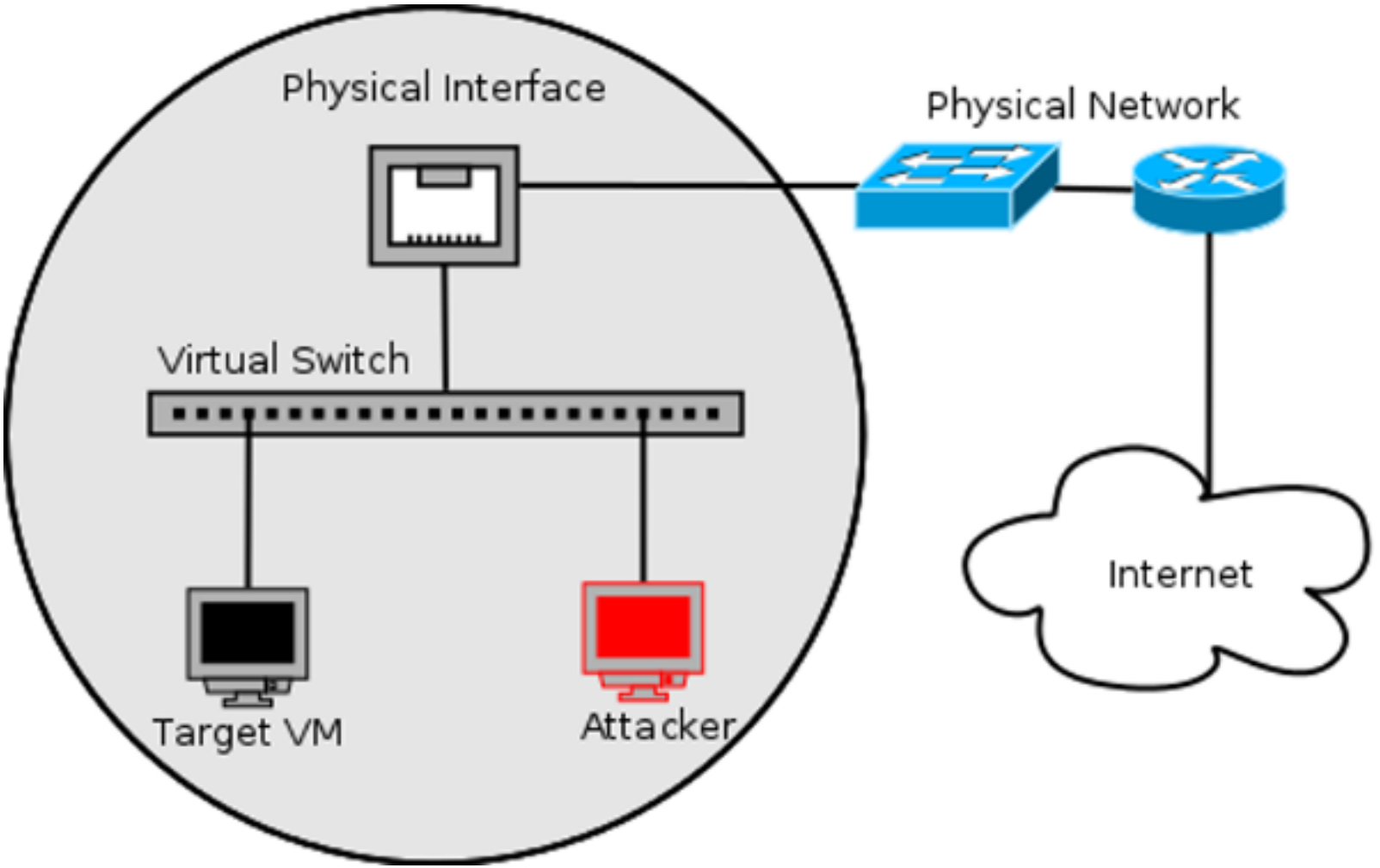
New Hardware Specs

Hypervisor Platform	Virtual Switch
Gentoo OS Xen 4.5.1	Linux 802.1d Bridging
Gentoo OS Xen 4.5.1	Open vSwitch 2.4.0
VMWare vSphere ESXi 6.0.0	Standard ESXi Virtual Switch
MS Server 2012 R2 DataCenter w/Hyper-V	Standard Hyper-V Virtual Switch
MS Server 2012 R2 DataCenter w/Hyper-V	Cisco Nexus 1000v 5.2(1)SM3(1.1a)
ProxMox 3.4 *(KVM)*	Linux 802.1d Bridging
Citrix XenServer 6.5.0	Open vSwitch 2.1.3
Kali 2.0 Standalone System	No virtual switch

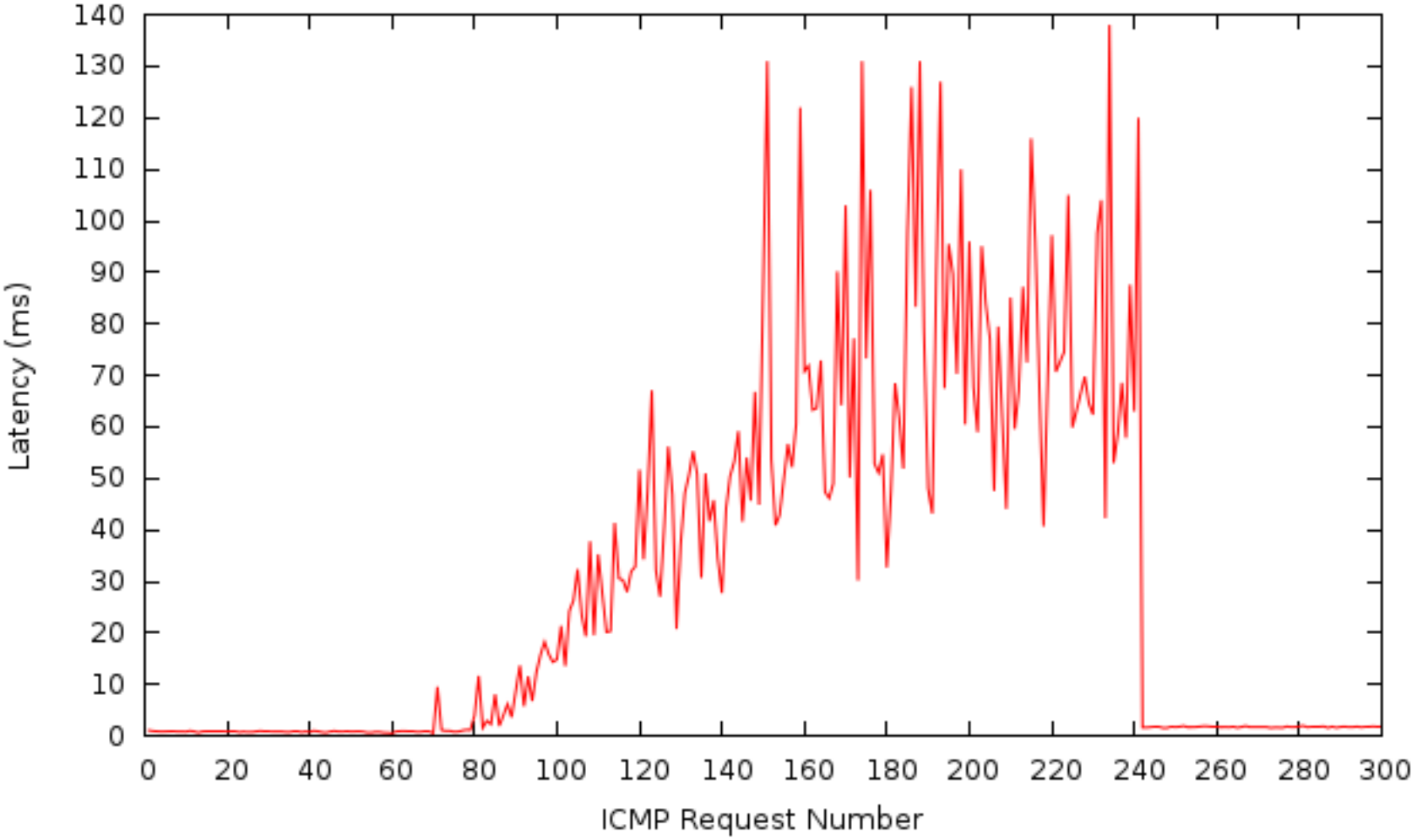
- *Identical Systems:*
  - *1U SuperMicro server system*
  - *CPU: Intel Xeon X3-1240V3 Quad Core w/ Hyper-Threading*
  - *RAM: 32GB*
  - *Hard Drive: 500GB WD Enterprise 7200RPM SATA*
  - *4 on-board Intel Gigabit network interface cards*

MAC Flooding Attack
- Performance Updates -

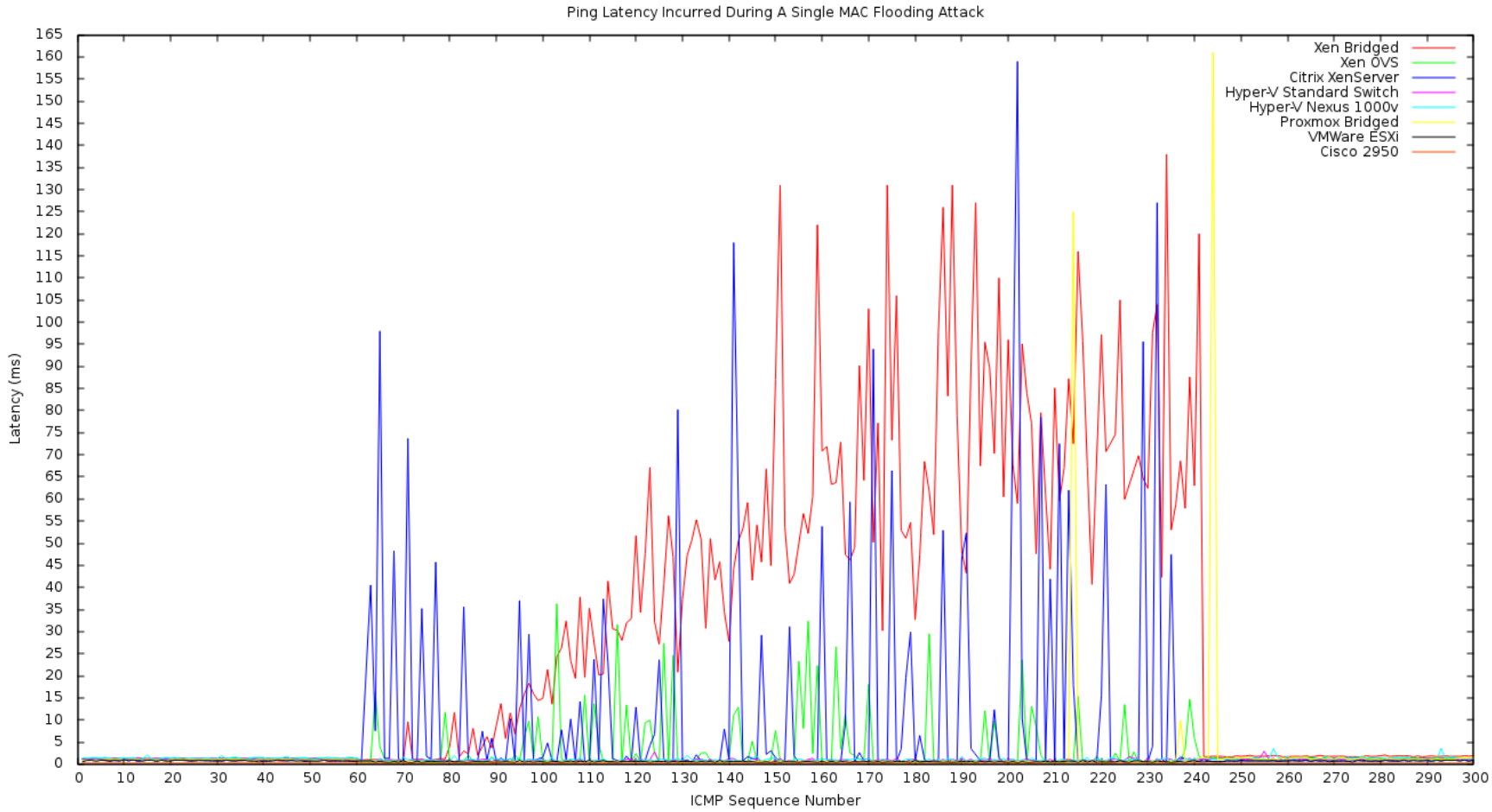
MAC Flooding Attacks
Network Diagram



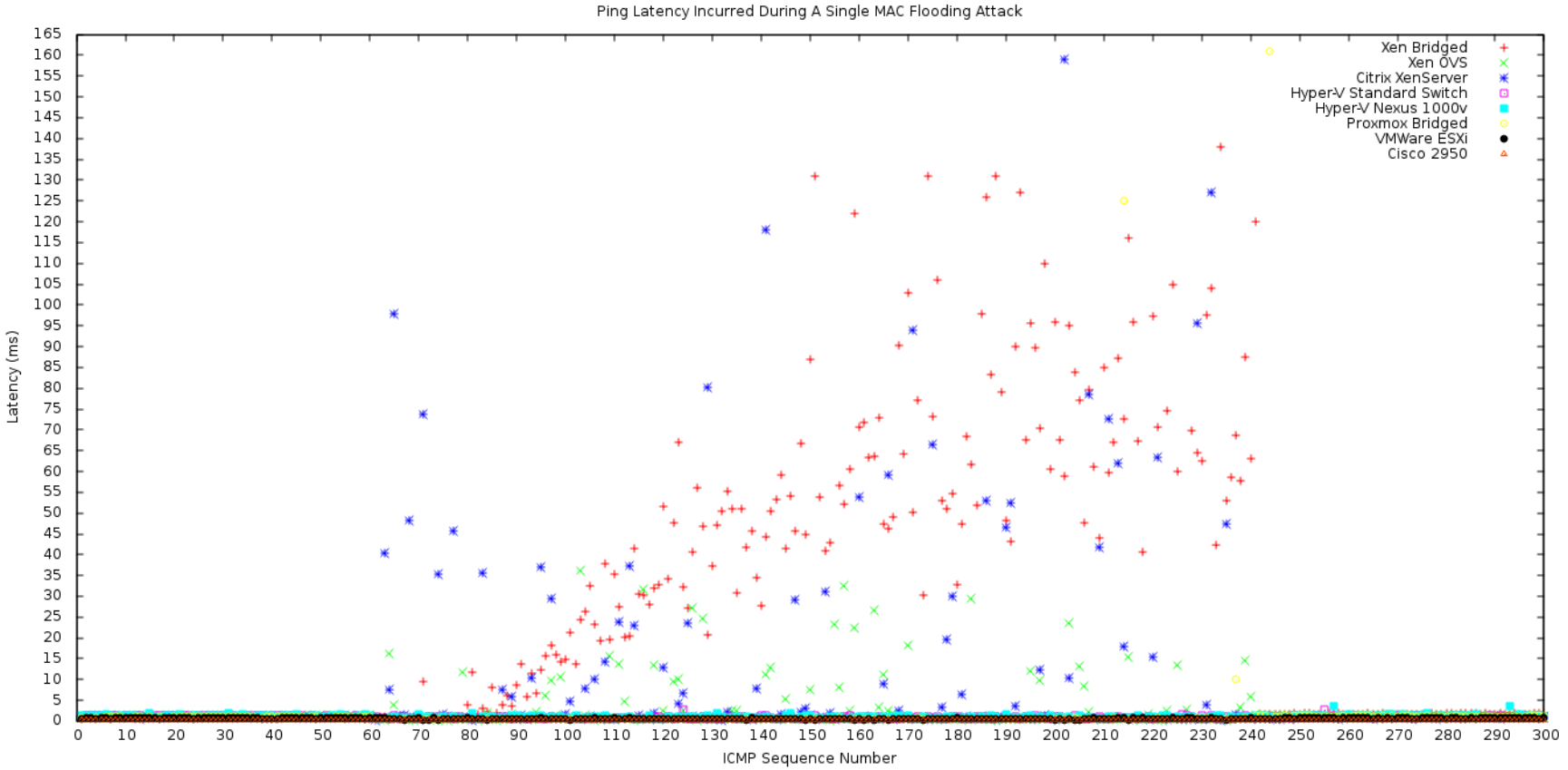
MAC Flooding
(*Network Performance Metrics)*
- Gentoo/Xen Bridged Interface -



MAC Flooding
(*Network Performance Metrics*)
- Every Platform Including Cisco 2950 Control -



MAC Flooding
(*Network Performance Metrics)*
- Every Platform Including Cisco 2950 Control -



Note: All Layer 2 vulnerabilities discussed were targeted towards the virtual networking devices not the hypervisors themselves

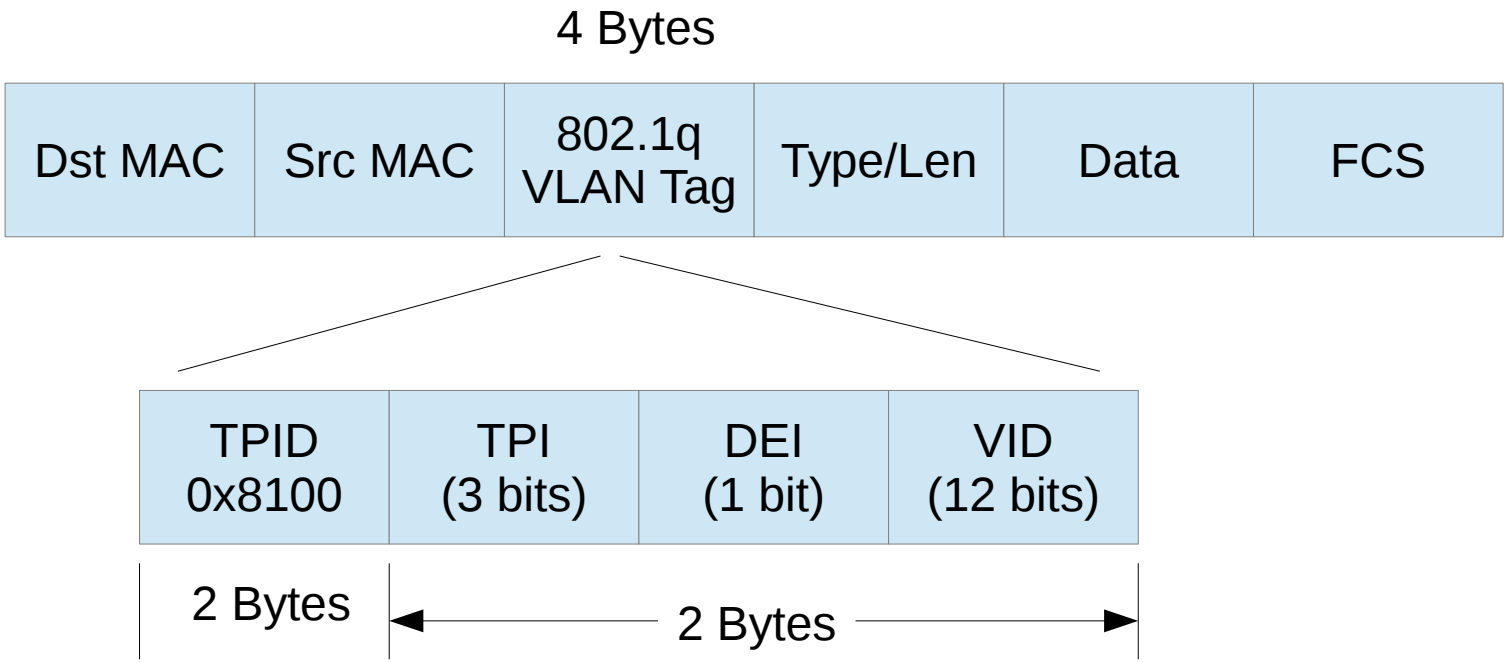
VLAN Hopping

VLAN Hopping Attacks

- Attack used to gain unauthorized access to another Virtual LAN on a packet switched network
- Attacker sends frames from one VLAN to another that would otherwise be inaccessible
- Two methods:
  - Switch Spoofing
    - Cisco proprietary
  - Double Tagging
    - Exploitation of 802.1Q standard

Virtual LAN Tag

- Ethernet frames are modified for VLAN traffic:
  - Addition of a 802.1q VLAN header
    - *32 bits of extra information wedged in*



Switch Spoofing

- CVE-2005-1942
  - http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2005-1942
  - *“Cisco switches that support 802.1x security allow remote attackers to bypass port security and gain access to the VLAN via spoofed Cisco Discovery Protocol (CDP) messages.”*

Switch Spoofing

- Cisco Discovery Protocol
  - Cisco proprietary Layer 2 protocol
  - Allows connected Cisco devices to share information
    - Operating system
    - IP address
    - Routing information
    - Duplex settings
    - VTP domain
    - VLAN information

Switch Spoofing

- CVE-1999-1129
  - http://www.cvedetails.com/cve/CVE-1999-1129/
  - *“Cisco Catalyst 2900 Virtual LAN (VLAN) switches allow remote attackers to inject 802.1q frames into another VLAN by forging the VLAN identifier in the trunking tag.”*

- And directly from Cisco:
  - *DTP: Dynamic Trunking protocol. "If a switch port were configured as DTP auto and were to receive a fake DTP packet, it might become a trunk port and it might start accepting traffic destined for any VLAN" (Cisco).*
    - *DTP Auto is the default setting on most Cisco switches!*

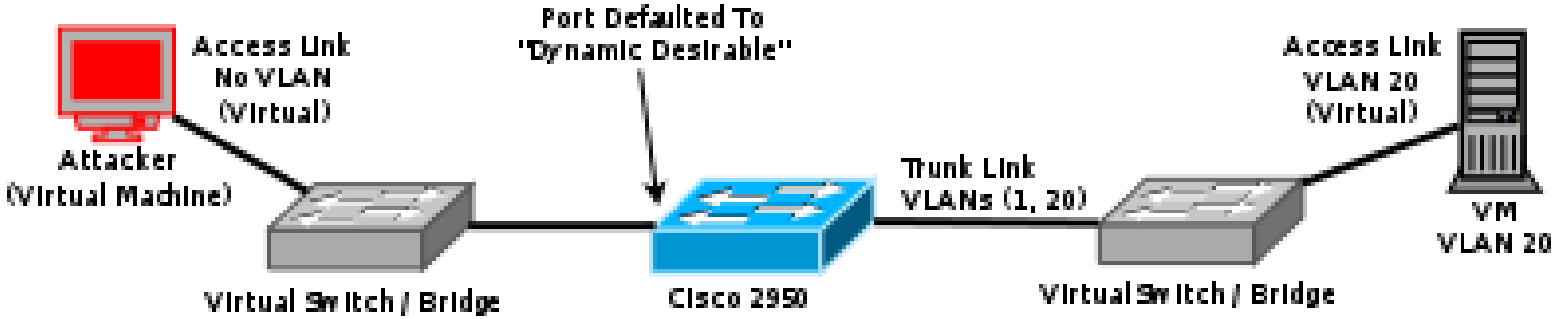
Switch Spoofing

- Dynamic Trunking Protocol
  - Cisco proprietary Layer 2 protocol
  - Allows automatic configuration of trunk ports on Cisco switches
    - Automatically configures VLAN trunking for all supported VLANs
  - Provides ability to negotiate the trunking method with neighbor devices
  - Pair this with CDP and your Cisco devices can pretty much configure themselves *(not very securely!)*

Switch Spoofing

- Consequences
  - Attacker's system has a trunk connection to the switch
    - Attacker can generate frames for any VLAN supported by the trunk connection
    - Attacker can communicate with any device on any of the associated VLANs
    - Two-way communication can occur between the attacker and a targeted node because the attacker can actually place themselves on the VLAN
    - Also allows attacker to eavesdrop on the traffic within a target VLAN

Switch Spoofing Demo (VMWare ESXi 6.0)



https://www.youtube.com/watch?v=mMGezerlg9c&feature=youtu.be&t=20s

Switch Spoofing Results

	Results of Attack	
Platform	**Negotiate Trunk Link**	**Unauthorized VLAN Access**
Physical Kali 2.0 Control System	✓	✓
OS Xen w/ Linux Bridging	✓	✓
OS Xen w/ Open vSwitch		
VMWare vSphere ESXi	✓	✓
MS Hyper-V Standard vSwitch		
MS Hyper-V Cisco Nexus 1000v		
Proxmox	✓	✓
Citrix XenServer		

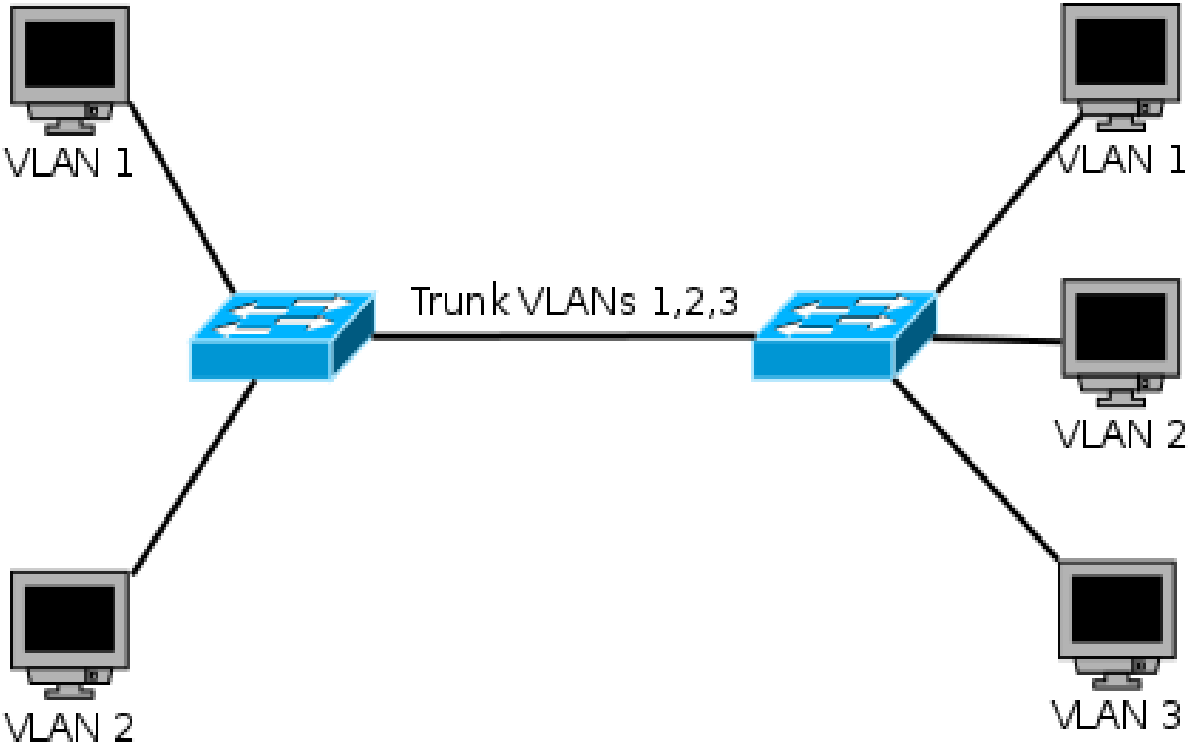
Switch Spoofing

- Mitigation
  - Disable unused switch ports
  - Disable CDP and DTP
    - Or use on an as need, per port basis!
  - Restrict the amount of trunk ports
    - Should only be configured when connecting devices require it *(ie. other switches)*
    - Limit VLAN access on trunk ports to only what the connected segments require
  - Configure all other ports as *access ports (no trunking)* with **no access** to the *native VLAN*

Double Tagging

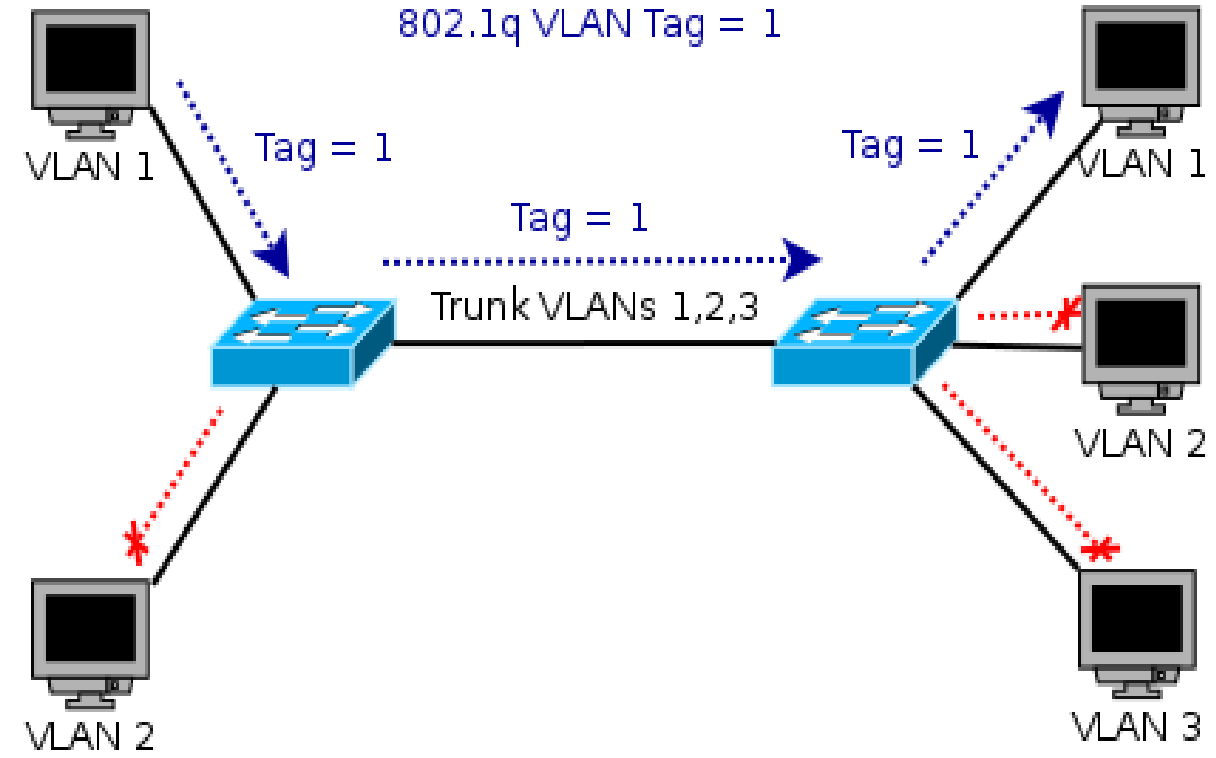
- CVE-2005-4440
  - http://www.cvedetails.com/cve/CVE-2005-4440/
  - "*The 802.1q VLAN protocol allows remote attackers to bypass network segmentation and spoof VLAN traffic via a message with two 802.1q tags, which causes the second tag to be redirected from a downstream switch after the first tag has been stripped.*"
  - A.K.A: *"Double-Tagging VLAN jumping attack"*

802.1Q Tagging



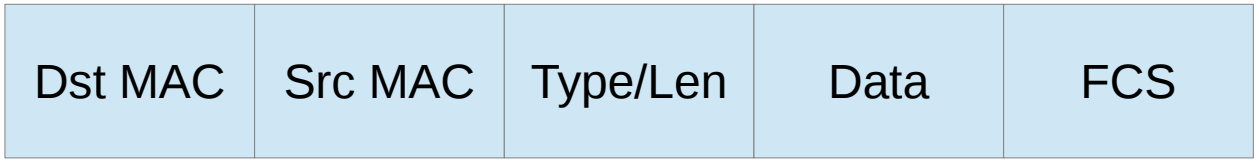
VLAN 1 - Native VLAN
VLANs 2,3 - Access VLANs

802.1Q Tagging



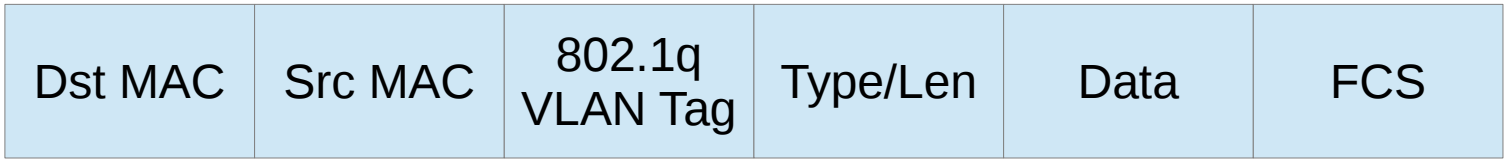
VLAN 1 - Native VLAN
VLANs 2,3 - Access VLANs

Double Tagging



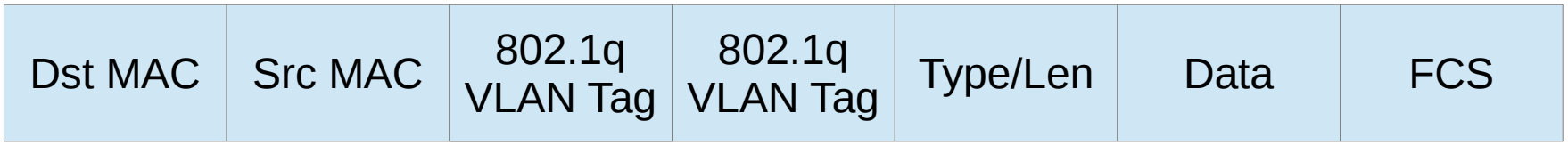
Standard 802.3 Ethernet Frame

4 Bytes



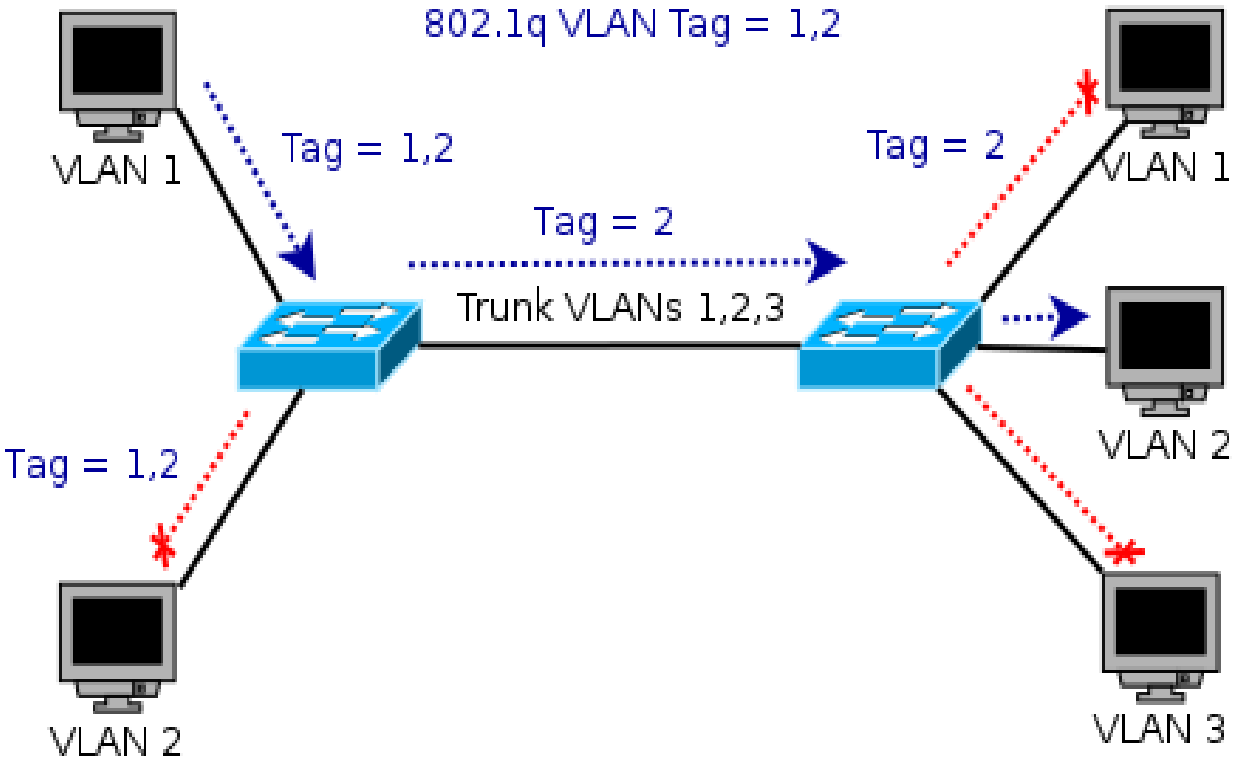
802.3 Ethernet Frame Tagged with 4 Byte 802.1q header

4 Bytes 4 Bytes



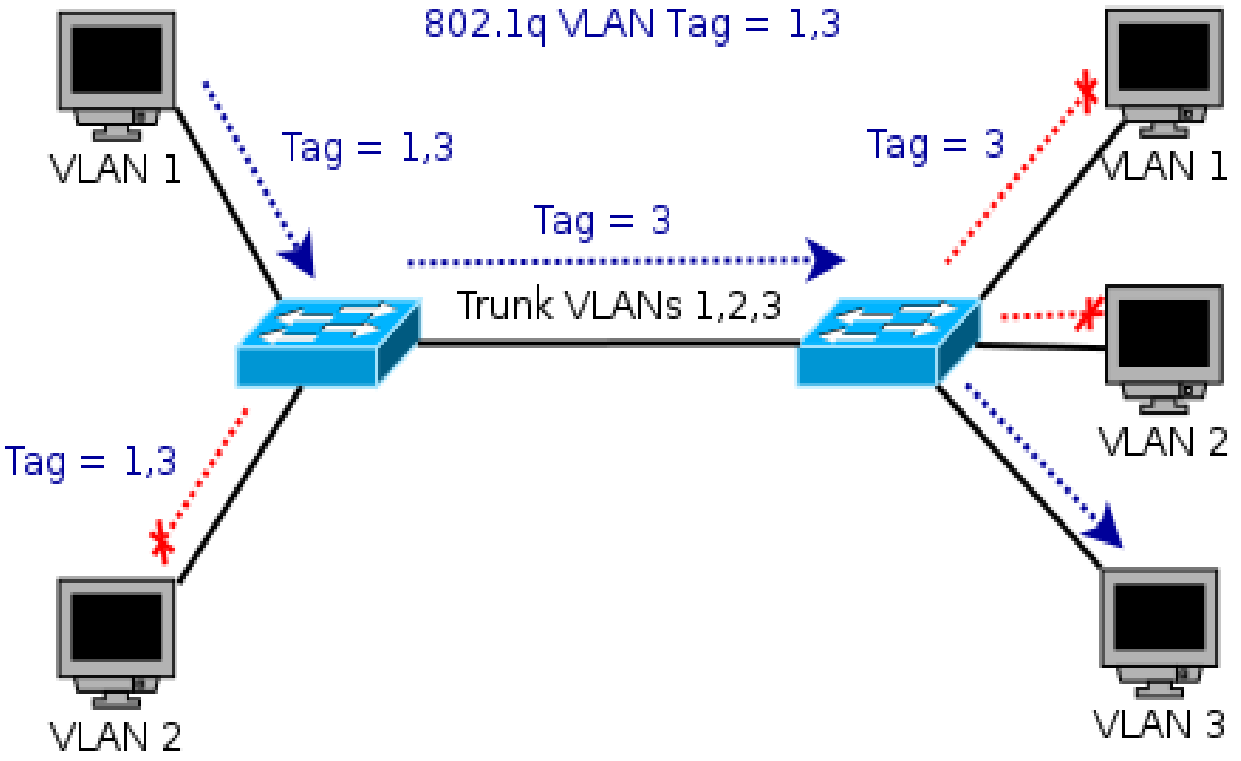
802.3 Ethernet Frame Tagged with multiple 4 Byte 802.1q headers

Double Tagging



VLAN 1 - Native VLAN
VLANs 2,3 - Access VLANs

Double Tagging

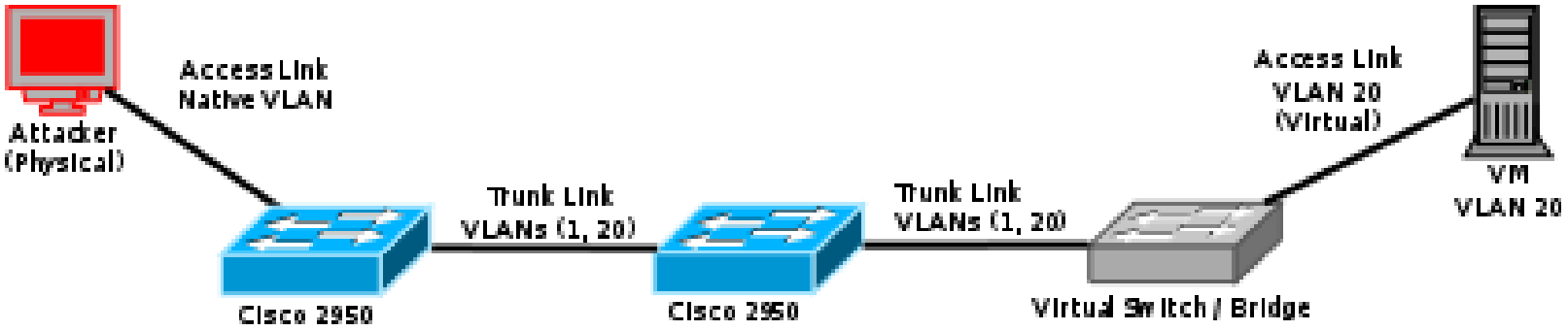


VLAN 1 - Native VLAN
VLANs 2,3 - Access VLANs

Double Tagging

- Consequences
  - Attacker can send packets to a target VLAN
  - Targeted system cannot respond back
    - Attacking system is on the native VLAN
    - Target is on an access VLAN isolated from the native VLAN broadcast domain
  - Not a good attack for eavesdropping
  - Excellent method for DoS attacks
  - Can be used as one way covert channels

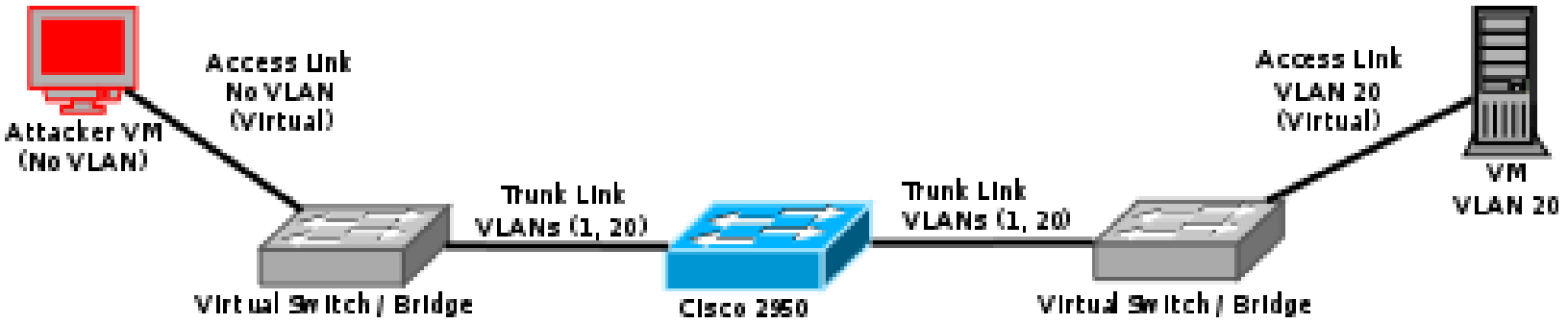
Double-Tagging Demo
(Two Physical Switches)



https://www.youtube.com/watch?v=V2Ht-GB4NbE&feature=youtu.be&t=45s

Physical Attacker, 2 Physical Cisco 2950 Switches, ProxMox Target

Double-Tagging Demo
(Two Virtual Switches w/ a Cisco 2950 in the Middle)



https://www.youtube.com/watch?v=jJDBJRoukIo&feature=youtu.be&t=45s

Attacker: XenServer VM
Target: ProxMox

Double-Tagging Demo
(One Physical Switch)

https://www.youtube.com/watch?v=np46KuXpk9c&feature=youtu.be&t=35s

Attacker: Physical Kali
Target: MS HyperV Guest via Cisco Nexus 1000v

Double Tagging Results

Platform	Results of Attack	
	Single Switch	Double Switch
OS Xen w/ Linux Bridging	✓	✓
OS Xen w/ Open vSwitch	✓	✓
VMWare vSphere ESXi	✓	✓
MS Hyper-V Standard vSwitch		
MS Hyper-V Cisco Nexus 1000v	✓	✓
Proxmox	✓	✓
Citrix XenServer	✓	✓

Platform	Results of Attack
	Virtual Switch
OS Xen w/ Linux Bridging	✓
OS Xen w/ Open vSwitch	✓
VMWare vSphere ESXi	
MS Hyper-V Standard vSwitch	
MS Hyper-V Cisco Nexus 1000v	
Proxmox	✓
Citrix XenServer	✓

Double Tagging

- Mitigation Techniques
  - Do not assign any hosts to VLAN 1 *(native VLAN)*
    - If necessary significantly limit access
    - Disable VLAN 1 on unnecessary ports
  - Change native VLAN on all trunk ports to something different than VLAN 1
  - Restrict access to switches by MAC address
    - Can spoof MAC addresses to get around this
  - Heart of this attack is having access to the native VLAN!
    - This is the default VLAN for all ports on a switch!

ARP Spoofing

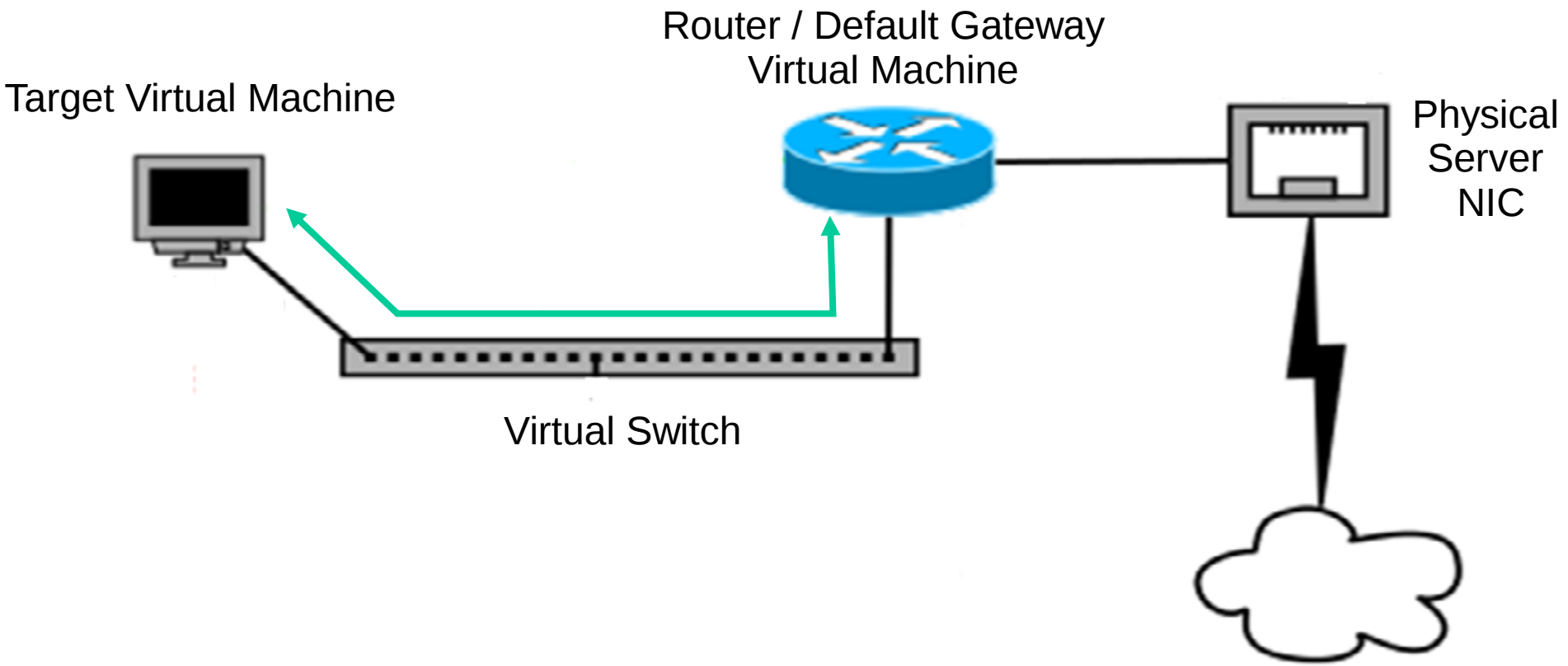
Address Resolution Protocol

- Layer 2 network protocol used to map physical MAC addresses to logical IP addresses within a broadcast domain
- Each system on the network maintains an 'ARP Cache'
  - Stores address translation information for 'discovered nodes' on the network
  - ARP caches will differ between inter-networked systems
    - not every node needs to communicate with every other node
  - Common entries that are generally seen in the 'ARP cache'
    - Default Gateway
    - Local DNS servers

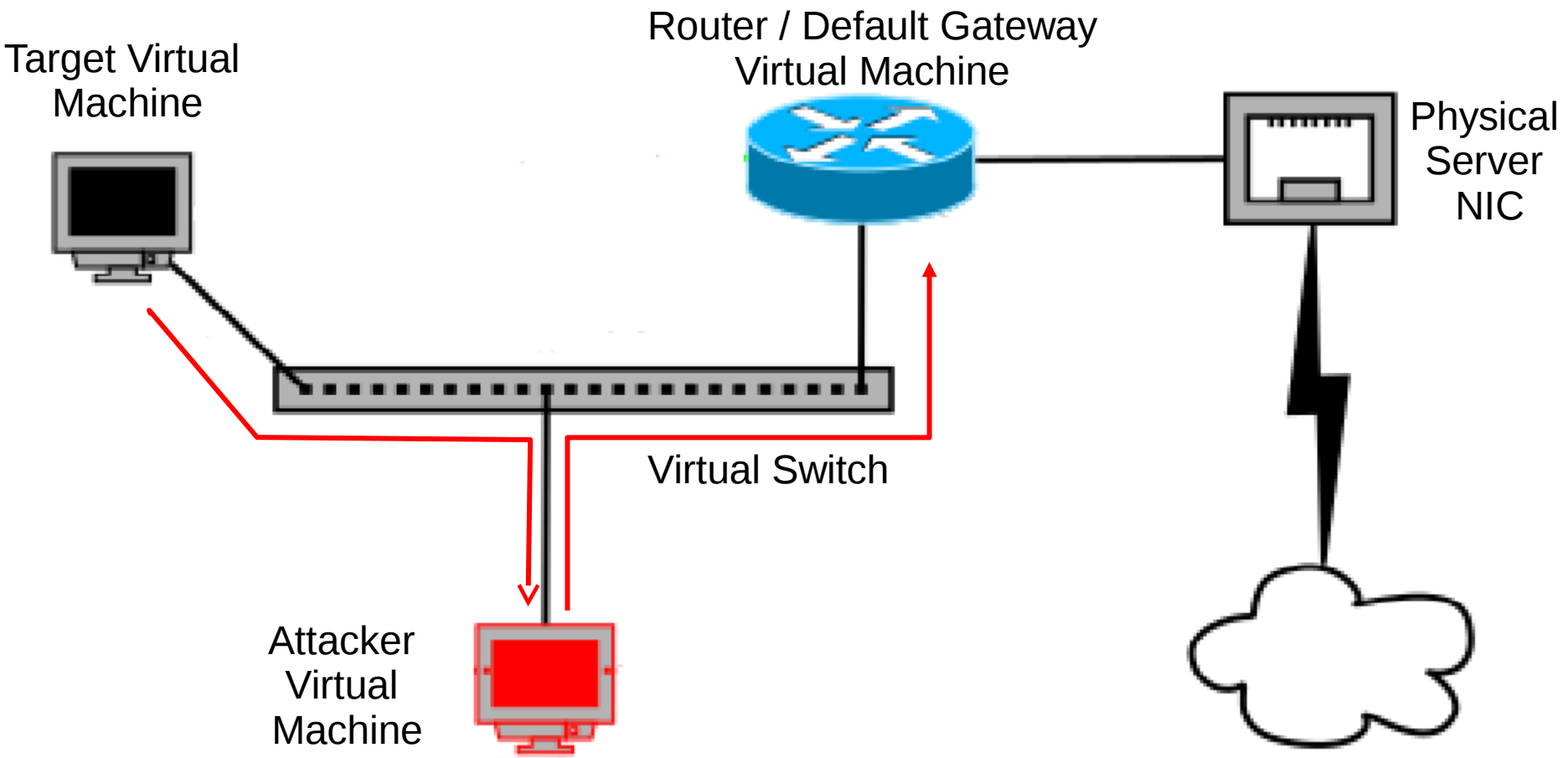
ARP Process

- Simple process to discover the Layer 3 address of another node within the Layer 2 broadcast domain
  - Initiating system sends a broadcast request to the entire Layer 2 network:
    - *Who has '192.168.1.10' tell '192.168.1.3'*
  - The node at '192.168.1.10' sees the broadcast and replies with its Layer 2 MAC address
    - *'192.168.1.10' is at 'ec:1b:d7:66:02:51'*
  - The initiating system then stores the translation of 'ec:1b:d7:66:02:51' to '192.168.1.10' in its ARP Cache so that it does not need to repeat the discovery process again

ARP Spoofing
Normal Traffic Flow



ARP Spoofing
Man-In-The-Middle Attack



ARP Spoofing
Man-In-The-Middle Attack Demo

https://www.youtube.com/watch?v=1h-pbTktCwI&feature=youtu.be&t=1m45s

Attacker: Physical Kali
Target: VMWare ESXi 6.0 VM

ARP Spoofing Results

	Results of Attack	
Platform	**Manipulate ARP Cache**	**Eavesdropping Allowed**
OS Xen w/ Linux Bridging	✓	✓
OS Xen w/ Open vSwitch	✓	✓
VMWare vSphere ESXi	✓	✓
MS Hyper-V Standard vSwitch	✓	✓
MS Hyper-V Cisco Nexus 1000v	✓	✓
Proxmox	✓	✓
Citrix XenServer	✓	✓

ARP Spoofing Mitigation

- Cisco switches can make use of DHCP snooping and Dynamic ARP inspection
  - Validate ARP requests to verify authenticity
  - Feature not supported on any virtual switches except the non-free version of the Cisco Nexus 1000v
- *arpwatch*
  - Linux utility developed at the *Lawrence Berkeley National Laboratory*
  - Runs as a service on a Linux system and monitors the network for changes in ARP activity

Conclusion: Virtual vs Physical?

- Results show that virtual networking devices can pose the same or even greater risks than their physical counterparts
- Which systems were vulnerable varied widely across the tests – no one “best” system
- Lack of sophisticated Layer 2 security controls similar to what is available on enterprise grade physical switches greatly increases the difficulty in securing virtual switches against these attacks

Bottom-line impact

- A single malicious virtual machine has the potential to sniff all traffic passing over a virtual switch
  - This can pass through the virtual switch and affect physically connected devices allowing traffic from other parts of the network to be sniffed as well!
- Significant threat to the confidentiality, integrity, and availability *(CIA)* of data passing over a network in a virtualized muli-tenant environment

What can users do?

- Educated users can question their hosting providers
  - Which virtual switch implementations being used?
  - To which attacks vulnerable?
- Audit the risk of workloads they run in the cloud or within multi-tenant virtualized environments
- Consider/request extra security measures – on their own and from hosting provider
    - Increased use of encryption
    - Service monitoring
    - Threat detection and alerting

Next steps for us

- Small team
  - Improvements this year but more we'd like to do
- Institute for apples-to-apples testing of virtualized environments
  - Looking for industrial partners to participate
- More testing in production environments
  - Leads from last year still to followup on
  - Bottleneck is need more students funded to do testing (good educational value :-))

- **Email:**
  - bullrl@clarkson.edu
  - jnm@clarkson.edu

- The white paper and slides are available on the DEFCON 24 CD. The white paper contains links to each of the demo videos used in this presentation.
- Links to all publications, presentations, and demo videos related to this research can also be found at *http://ronnybull.com*