Exploring Layer 2 Network Security in Virtualized Environments

Ronny L. Bull, Jeanna N. Matthews
Wallace H. Coulter School of Engineering
Clarkson University
Potsdam, NY 13699
Email: {bullrl, jnm}@clarkson.edu

Abstract—**Cloud service providers offer their customers the ability to deploy virtual machines in a multi-tenant environment. These virtual machines are typically connected to the physical network via a virtualized network configuration. This could be as simple as a bridged interface to each virtual machine or as complicated as a virtual switch providing more robust networking features such as VLANs, QoS, and monitoring. In this paper, we explore whether Layer 2 network attacks that work on physical switches apply to their virtualized counterparts by performing a systematic study across four major hypervisor environments - Open vSwitch, Citrix XenServer, Microsoft Hyper-V Server and VMware vSphere - in seven different virtual networking configurations. First, we use a malicious virtual machine to run a MAC flooding attack and evaluate the impact on co-resident VMs. We find that network performance is degraded on all platforms and that it is possible to eavesdrop on other client traffic passing over the same virtual network for Open vSwitch and Citrix XenServer. Second, we use a malicious virtual machine to run a rogue DHCP server and then run multiple DHCP attack scenarios. On all four platforms, co-resident VMs can be manipulated by providing them with incorrect or malicious network information.**

Keywords—Virtualization, Networking, Network Security, Cloud Security, Layer 2 Attacks.

I. Introduction

With the growing popularity of Internet-based cloud service providers, many businesses are turning to these services to host their mission critical data and applications. Cloud customers often deploy virtual machines to shared, remote, physical computing resources. Virtual machines running in cloud capacity are connected to the physical network via a virtualized network within the host environment. Typically, virtualized hosting environments will utilize either a bridged network interface or a virtualized switch such as Open vSwitch[1], [2] for Xen and KVM based environments, or the Cisco Nexus 1000V Series virtual switch for VMware vSphere environments[3]. These virtual switches are designed to emulate their physical counterparts. It is important for users of multi-tenant cloud services to understand how secure their network traffic is from other users of the same cloud services, especially given that VMs from many customers share the same physical resources. If another tenant can launch a Layer 2 network attack and capture all the network traffic flowing from and to their virtual machines, this poses a substantial security risk. By understanding which virtual switches are vulnerable to which attacks, users can evaluate the workloads they run in the cloud, consider additional security mechanisms such as increased encryption and/or increased monitoring and detection of Layer 2 attacks.

In this paper, we present the results of a systematic study to evaluate the effects of MAC flooding and DHCP attacks across four major hypervisor environments with seven different virtual network configurations. First, we provide some background information on the general network configuration options available to virtualized environments. We then introduce the test environment, and present our attack methodology using Media Access Control *(MAC)* and Dynamic Host Configuration Protocol *(DHCP)* attack scenarios. We conclude the paper by discussing related work and summarizing our results.

II. Network Configuration Options

There are two types of networking configurations that are typically used in virtualized environments; bridging and switching. In this section we describe both options and discuss how each one is applied within a virtualized network.

A. Bridging

Bridged mode is the simplest of configurations providing an interface dedicated to virtual machine use. A bridge connects two or more network segments at Layer 2 in order to extend a broadcast domain and separate each of the segments into their own individual collision domains[4]. A forwarding table[4], [5] is used to list the MAC addresses associated with devices located on each network segment connected to the bridge *(Figure 1)*. Requests are forwarded based upon contents of this table and the destination MAC address located in the Ethernet frame. A frame is forwarded across the bridge only if the MAC address in the destination block of the frame is reachable from a different segment attached to the bridge. Otherwise, the frame is directed to a destination address located on the same segment as the transmitting device or dropped.

In virtualized environments, guest machines utilize userspace virtual network interfaces that simulate a Layer 2 network device in order to connect to a virtual bridge. Typically, the virtual bridge is configured and bound to a physical interface on the host machine that is dedicated solely to virtual machine traffic.

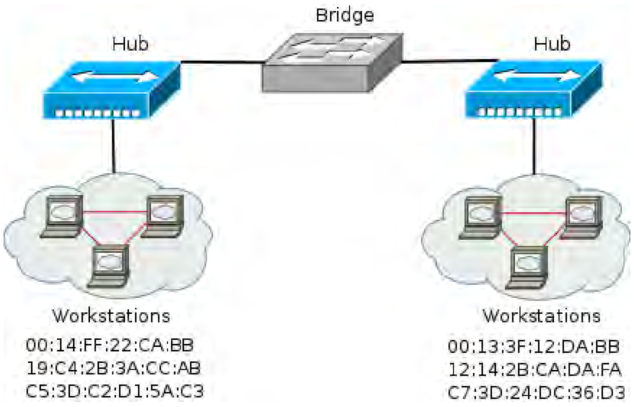

Fig. 1. A basic bridge using a forwarding table to pass requests between two network segments.

B. Switching

Physical switches have the capability of operating at Layer 2 or higher of the OSI model. Switches can be thought of as multi-port bridges[4] where each port of the switch is considered as its own isolated collision domain. Instead of a forwarding table, switches employ a CAM (content addressable memory) table[4] . Content addressable memory is specialized memory hardware located within a switch that allows for the retention of a dynamic table or buffer that is used to map MAC addresses of devices to the ports they are connected to *(Figure 2)*. This allows a switch to intelligently send traffic directly to any connected device without broadcasting frames to every port on the switch. The switch reads the frame header for the destination MAC address of the target device, matches the address against its CAM table, then forwards the frame to the correct device. The use of a CAM table and the separation of collision domains are key factors in preventing eavesdropping of network traffic between devices connected to the switch. However, a physical switch is an embedded device and has a finite amount of memory available to its CAM table, once it is used up the switch can no longer dynamically add to its buffer. If a MAC address is not found in the CAM table, a packet destined for it will be sent to all interfaces. The majority of physical switches in use today employ CAM chips that are capable of holding up to 32,000 addresses[4] which can easily be saturated by a single MAC flooding attack in a very short amount of time.

Virtual switches emulate their physical counterparts and are capable of providing features such as VLAN traffic separation, performance and traffic monitoring, as well as quality of service *(QoS)* solutions. Virtual machines are connected to a virtual switch by the way of virtual network interfaces *(VIF)* that are similar to the Layer 2 network devices used in conjunction with virtual bridges.

III. Test Environment

In this section, we provide details about the test environment that was created which consisted of seven server class systems all located on a test network isolated from local production networks to avoid impacting them. We deployed an optimized installation of Gentoo Linux and the Xen 4.3 hypervisor to three Dell PowerEdge 860 servers each equipped with a dual core Intel Xeon 3050 2.13GHz processor, 4 GB of memory, and a 500 GB hard drive. Each system contained dual Broadcom NetXtreme BCM5721 Gigabit Ethernet PCI Express network interface cards integrated into the motherboard. The first network interface was dedicated to the privileged control domain on each server for administrative functions, and the second configured to be utilized by guest virtual machines. Each sever's 500 GB hard disk was divided into four partitions; a 100MB ext3 /boot, a 10GB ext3 /, a 2GB swap, with the remainder allocated to LVM storage for virtual machine deployment.

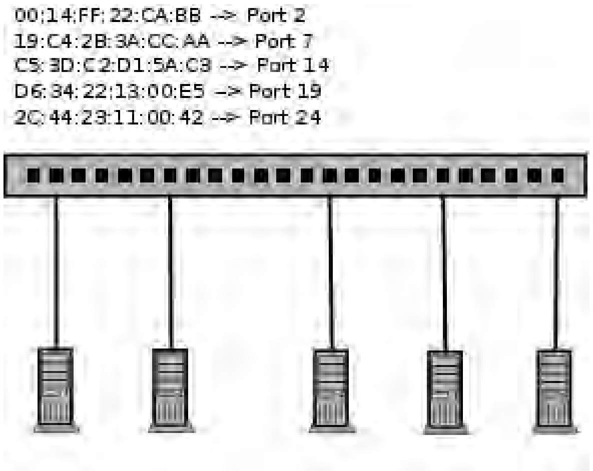
00;14:FF: 22:CA:BB --> Port 2
19:C4:2B:3A:CC:AA --> Port 7
C5:3D:C2:D1:5A:C3 --> Port 14
D6:34:22:13:00:E5 --> Port 19
2C:44:23:11:00:42 --> Port 24

Fig. 2. A switch and its CAM table.

Four additional servers were configured with enterprise level hypervisor solutions; Citrix XenServer 6.2, Microsoft Windows Server 2008 R2 with the Hyper-V hypervisor, Microsoft Hyper-V 2008 *(free edition)*, and VMware vSphere *(ESXi)* 5.5 *(free edition)*. The hardware utilized for the Citrix XenServer 6.2 system was identical to the three Gentoo systems, however the Microsoft Hyper-V and the VMware vSphere hypervisors were configured on systems with different hardware configurations due to a lack of additional Dell PowerEdge 860 systems. Both Microsoft Windows Server 2008 R2 along with the Hyper-V hypervisor as well as the free version of Hyper-V 2008 were installed to identical Dell PowerEdge 2950 server systems containing dual quad core Intel Xeon 5140 processors at 2.33GHz, 32GB of memory, and a 145GB SATA hard drive. VMware vSphere *(ESXi)* 5.5 *(free edition)* was deployed to a custom built server using a Supermicro X9SCL server motherboard, a quad core Intel Xeon E3-1240 processor at 3.30GHz, 24GB of memory, and a 500GB SATA hard drive. The Hyper-V and vShpere systems were each outfitted with two network adapters in order to provide separate dedicated interfaces for administrative purposes and virtual machine use. Though there are notably some variations in the hardware configurations summarized in *Table I*, it is important to note that these differences had no impact on the results of the experiments that were performed.

For the MAC flooding scenario, two virtual machines were deployed to each virtualization platform, one of which was setup as a malicious client attempting to eavesdrop on the traffic of other tenant virtual machines *(Figure 3)*. The Kali Linux security distribution[6] was selected due to the plethora

TABLE I. SUMMARY OF TEST ENVIRONMENT HARDWARE.

	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

of network security auditing tools that come pre-installed and configured. Two complete installations of Kali were installed to each server on 20GB LVM partitions as HVM guests. The systems were then allocated static IP addresses that positioned them on the same isolated subnet as the servers and were completely updated.

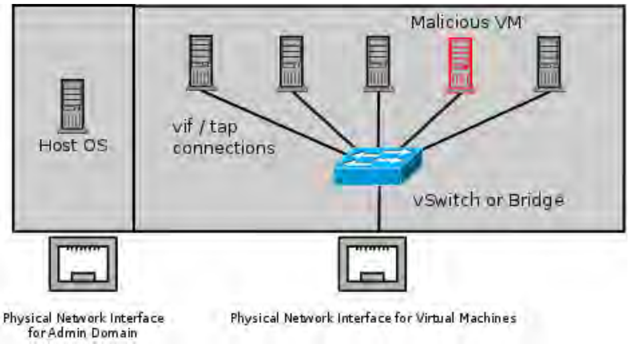
Malicious VM
Host OS
vif / tap connections
vSwitch or Bridge
Physical Network Interface for Admin Domain
Physical Network Interface for Virtual Machines

Fig. 3. A malicious virtual machine located on a multi-tenant virtual network.

The DHCP attack testing required a more elaborate setup. It was necessary to create four new virtual machines within each hypervisor platform in order to setup scenarios to conduct the experiments. Each new machine was created based upon a minimal installation of CentOS 6.5[7], and configured for a specific purpose *(Table II)*.

TABLE II. NEW VIRTUAL MACHINES ADDED TO EACH HYPERVISOR PLATFORM FOR LAYER 2 DHCP ATTACK TESTING.

Operating System	Completely Updated	System Purpose	Virtual Interfaces
CentOS 6.5	Yes	DHCP/DNS Server	1
CentOS 6.5	Yes	Simple Router	2
CentOS 6.5	Yes	HTTP Server	1
CentOS 6.5	No	Left Vulnerable to ShellShock	1

A virtual machine acting as a rogue DHCP server was setup and configured using DNSMasq[8] a lightweight DHCP and DNS server. It was also necessary to create a simple router using iptables[9] on a separate virtual machine in order to forward traffic between two broadcast domains using NAT and two network interfaces. A basic Apache[10] web server was setup on a third virtual machine to act as a malicious web server, and the final machine was configured as a minimal client that was left unpatched and vulnerable to shellshock[11].

IV. ATTACKS PERFORMED

Two Layer 2 networking attack categories were explored and thoroughly tested across all platforms; MAC flooding and DHCP attacks. Each attack simulation was performed identically on all platforms in order to analyze the differences between the environments when subjected to the different attack scenarios.

A. MAC Flooding

The most common Layer 2 Media Access Control attack is a MAC flooding attack in which the attacker generates many packets with random MAC addresses in an attempt to overflow the *(CAM)* buffer within a switch and thus force the switch into a mode in which it broadcasts packets on all interfaces. This happens because the legitimate MAC addresses are evicted from the CAM table in favor of the many random MAC addresses generated by the attacker. This is referred to as hub mode and when a switch is operating in hub mode, the inherent separation of collision domains is broken and all frames passing through the switch are forwarded to all connected devices. This allows for passive eavesdropping of all traffic passing through the device. MAC flooding can be mitigated by enforcing port security on physical switches which imposes a limit on the amount of MAC addresses that can send traffic to a specific port[12]. This feature is not implemented within the majority of the virtual switches available today rendering them vulnerable to MAC flooding attacks.

The program macof from the dsniff package[13] was used on a Kali virtual machine to perform a MAC flooding attack on the virtual network within each test environment. This type of attack when performed on a physical switch typically causes the CAM table on the switch to fill up forcing the device to go into a fail safe or hub mode which in turn causes all packets on the network to be broadcast to every node connected to the switch. Wireshark was used to determine if the attack was successful by monitoring the network for HTTP traffic which should not be intercept-able by other hosts on the virtual network.

All tests were conducted in the same manner. Each server had two Kali Linux virtual machines deployed on them. For testing purposes both virtual machines were brought online. On the first virtual machine *(Kali1)* macof was started up using the command:

```
macof -i eth0
```

and left to run. Then Wireshark was started on the same virtual machine and an HTTP filter was applied to only display sniffed HTTP traffic. The second Kali virtual machine *(Kali2)* was then used to surf the web. If the attack proved to be successful then the HTTP traffic from *Kali2* should be viewable in Wireshark on *Kali1*.

1) Bridged Interface: Running the attack within the bridged virtual network test environment resulted in a significant performance degradation that impacted the usability of the tenant virtual machines, essentially creating a denial of service *(DoS)* type of attack. This effect was observed as a large increase in latency when attempting to interact with any of the virtual machines on the system either through SSH or VNC. While the MAC flooding attack was occurring remote connections to the virtual machines became unstable due to the saturation of the virtual network with spoofed frames. This effect was quantified by using the ping utility on the second virtual machine to measure the transmission latency to a server located on the physical network while the attack was occurring *(Figure 4)*. The attack however did not result in the ability to sniff other virtual machine traffic passing over the interface. This most likely comes from the fact that the standard bridge interface is missing the CAM table that typically is found on switches mapping known MAC addresses to switch ports, an essential element of the attack.

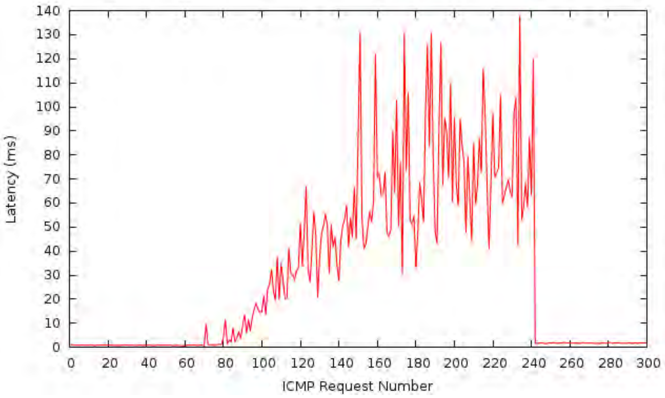

Fig. 4. Latency measured using the ping utility on a bridged virtual network during a MAC flooding attack. The attack was launched at ICMP request 61 and terminated at ICMP request 241.

2) Open vSwitch 1.11.0 Interface: When running the attack on the Open vSwitch 1.11.0 virtual network test environment not only was the same level of network performance degradation observed, but the attacking machine could also successfully sniff traffic from another tenant machine. *Figure 5* depicts the results of the successful attack and provides substance to the claim that virtual switches are vulnerable to some of the same Layer 2 attacks as physical switches.

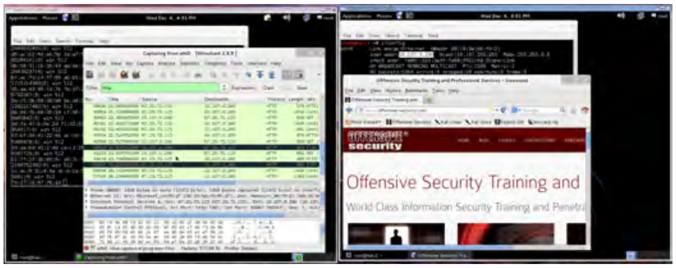
security
Offensive Security Training and
World Class Information Security Training and Penetr

Fig. 5. A malicious virtual machine running macof on an Open vSwitch virtual network and successfully sniffing HTTP traffic with Wireshark from another tenant virtual machine.

3) Open vSwitch 2.0.0 Interface: Running the attack on the latest version of Open vSwitch available at the time of this research revealed that the vulnerability still existed and had not been addressed. The system responded in the same way as the previous two attempts and the other tenant's HTTP traffic was view-able in Wireshark.

4) Citrix XenServer 6.2: Citrix XenServer 6.2 utilizes an older version of Open vSwitch (version 1.4.6) to provide virtual switching services to its client machines. When the MAC flooding test was attempted in the XenServer environment, it was also discovered that the flooding was able to escape the virtual environment which caused all upstream physical switches to go into hub mode as well. Not only did this allow the malicious virtual machine running Wireshark to sniff traffic from other tenant virtual machines, it also was able to eavesdrop on traffic from physical machines located within the same broadcast domain to which the physical Ethernet adapter was connected.

5) Microsoft Hyper-V Server 2008 R2: Testing under the Microsoft Hyper-V environment was performed both with and without the Windows Firewall service enabled to identify if there was any affect on the results. Both scenarios proved to be unsuccessful due to the fact that Microsoft Windows Server 2008 R2 provides some minimal protection for virtualized network traffic, this includes protection against MAC address spoofing[14].

Further testing was performed on the free version of Microsoft Hyper-V to see if the protection offered by Server 2008 R2 is also built into the bare metal product. As with the previous environment testing was performed both with and without the Windows Firewall service enabled. It was concluded that under both conditions the free version of Microsoft Hyper-V 2008 was also unaffected by the MAC flooding attack since it is built upon a minimal version of Microsoft Windows Server 2008 R2 entitled Server Core. The Core version of Microsoft Server 2008 R2 still provides the same level of network protection as the full version, but only allows for the installation of specific server roles to the operating system[15], in this case the Hyper-V hypervisor.

6) VMware vSphere (ESXi) 5.5 - free edition: All testing within the VMware vSphere environment was performed identically to the previous trials for completeness. Testing was performed on the free version of ESXi using the default virtual networking configuration. The results show that this particular configuration was not vulnerable to the MAC flooding attack in terms of a malicious user being able to eavesdrop on another tenant's network traffic. Due to the VMware end user license agreement[16] we are prevented from publishing any of the performance related results that were observed during the test.

7) Summary of MAC Flooding Results: It can clearly be seen from the results summarized in *Table III* that any virtualized network environment built upon the Open vSwitch virtual switch could be vulnerable to MAC flooding attacks, and has the potential to expose its client traffic to eavesdropping. Therefore, if a virtual machine is transmitting sensitive information over a virtual network that uses Open vSwitch precautions should be taken such as using encryption in order to ensure that the information in transit remains confidential.

TABLE III. MAC FLOODING ATTACK RESULTS ACROSS SEVEN TEST ENVIRONMENTS. ✓INDICATES THE PLATFORM WAS AFFECTED.

	Results of Attack	
Platform	**Eavesdropping Allowed**	**Impacted Performance**
OS Xen w/ Linux Bridging		✓
OS Xen w/ Open vSwitch 1.11.0	✓	✓
OS Xen w/ Open vSwitch 2.0.0	✓	✓
Citrix XenServer 6.2	✓	✓
MS Server 2008 R2 w/Hyper-V		✓
MS Hyper-V 2008 Free		✓
VMware vSphere (ESXi) 5.5		N/A

It should also be noted that in February of 2015 we notified the Open vSwitch security team of our discovery. They confirmed the vulnerability and immediately responded with a patch[17], [18] to resolve the issue. Since then the patch has been merged into every major branch of Open vSwitch from 2.0.0 on[19]. With that stated, it is important to recognize that at this time the current virtual switch implementation in Citrix XenServer has not been updated to a patched version of Open vSwitch. It is our recommendation that any environment running any version of Open vSwitch prior to the patched version of the 2.0.0 branch should be upgraded immediately, since both the vulnerability and exploitation technique have been made public.

B. DHCP Attacks

In order to perform a Layer 2 DHCP attack, an attacker must place a rogue DHCP server on a network in hopes that clients in the broadcast domain associate with it rather than the legitimate DHCP server. Once a client receives an IP address lease from a malicious DHCP server under an attacker's control, that client could also be seeded with the IP address of a poisoned DNS server, an incorrect default gateway, or be forced to run malicious code. This type of attack could also cause DoS situations where duplicate addressing occurs on the network causing the resources bound to those addresses to be inaccessible, or allow for the execution of man-in-the-middle attacks where traffic is first sent to an attacker and then onto the original destination. These attacks can be mitigated by enforcing static addressing, or by employing DHCP snooping on physical switches as well as DHCP server authorization within Active Directory environments.

Four different attack scenarios were duplicated across each of the seven test environments in order to evaluate the impact of these Layer 2 DHCP attacks. In the first scenario, the DNSMasq server was setup to pass option 100 to clients which was configured to leverage the shellshock exploit in order to remotely execute the echo command with root privileges on the target machine and place text into a file in /tmp. The following code was placed into the */etc/dnsmasq.conf* file on the DHCP server as a proof of concept to illustrate the vulnerability without damaging the client system.

```
dhcp-option-force=100,() { :; }; /bin/echo \\
'Testing shellshock vulnerability'>/tmp/shellshock_test
```

For the second scenario, the DNSMasq server was used to seed the minimal shellshock client with a poisoned DNS server through DHCP. Since DNSMasq also provides DNS server functionality the rogue DHCP server doubled as the poisoned DNS server that was passed to clients receiving addresses. The DNS server was setup to direct all traffic destined to www.gmail.com to be redirected to the malicious web server *(Figure 6)*. A command line web browser called *elinks*[20] was then used in the shellshock virtual machine to visit www.gmail.com in order to observe the effect.

Lastly, the DHCP server was configured to pass a bad default gateway address to clients that obtained their network configuration from it. First, it was set to pass 1.1.1.1 as the default gateway with the intention of causing a DoS attack for access of subnets outside of the existing broadcast domain. Second, the DHCP server was configured to point clients to the second virtual machine that was setup as a router to direct traffic to a malicious honeynet *(Figure 7)*. This in conjunction with a poisoned DNS server allows the attacker to direct traffic to malicious servers setup within the honeynet. In each case, the previously used web server was placed in the honeynet, and a DNS entry was setup to direct traffic to it through the rogue default gateway.

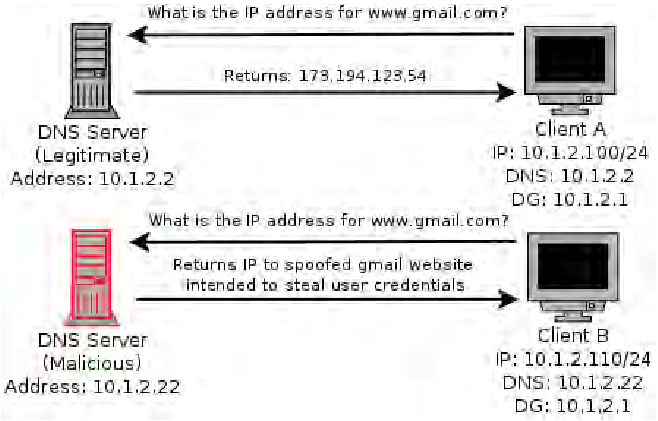

Fig. 6. Presence of a poisoned DNS server on a network whose address is provided to clients associated with a rogue DHCP server.

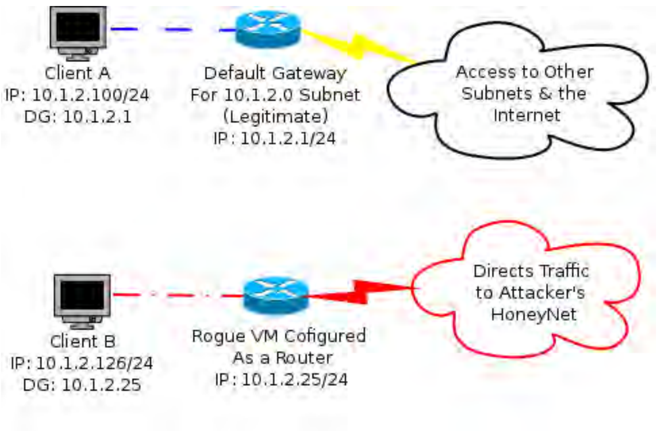

Fig. 7. Malicious virtual machine configured as a router on a network whose address is provided to clients as a default gateway when associated with a rogue DHCP server.

1) Summary of DHCP Attack Results: Table IV illustrates the results of all four DHCP attack scenarios that were run within each test environment. In all of the environments we tested, there was no protection provided against the attacks in their default configurations.

TABLE IV. DHCP ATTACK SCENARIO RESULTS ACROSS SEVEN TEST ENVIRONMENTS. ✓INDICATES A SUCCESSFUL ATTACK.

	Attack Scenarios			
Platform	**Shell Shock**	**Poisoned DNS**	**Invalid DG**	**Malicious DG**
OS Xen w/ Linux Bridging	✓	✓	✓	✓
OS Xen w/ Open vSwitch 1.11.0	✓	✓	✓	✓
OS Xen w/ Open vSwitch 2.0.0	✓	✓	✓	✓
Citrix XenServer 6.2	✓	✓	✓	✓
MS Server 2008 R2 w/Hyper-V	✓	✓	✓	✓
MS Hyper-V 2008 Free	✓	✓	✓	✓
VMware vSphere (ESXi) 5.5	✓	✓	✓	✓

V. Related Work

There has already been a substantial amount of work studying the vulnerability of physical networks to Layer 2 attacks [13], [21], [22], [23], but the impact on virtual networks has not received as much attention. This is beneficial in the fact that published research previously performed on physical networks can serve as a model for testing in virtual environments and comparisons can be made based upon the physical baselines. For instance, *Yeung et al.*[13] provide an overview of the most popular Layer 2 networking attacks as well as descriptions of the tools used to perform them. This work was very helpful in identifying possible attack vectors that could be emulated within a virtualized environment. *Altunbasak et al.*[21] also describe various attacks that can be performed on local and metropolitan area networks, as well as the authors' idea of adding a security tag to the Ethernet frame for additional protection. Cisco also published a white paper[22] regarding VLAN security in their Catalyst series of switches. The paper discloses testing that was performed on the switches in August of 2002 by an outside security research firm @stake which was acquired by Symantec in 2004. In the white paper, they discussed many of the same attacks that were mentioned by *Yeung et al.*[13], however the authors also went into detail about best practices and mitigation techniques that could be implemented on the physical switches in order to prevent the attacks from being successful.

VI. Future Work

Going forward, we intend to evaluate other Layer 2 networking attacks within these environments as well as develop mitigation techniques and hardening strategies that will contribute to an increased level of network security in virtualized environments. We also are especially interested in working with cloud service providers to assess the vulnerability of their platforms to these attacks. Understandably, it is unacceptable to run such experiments without the permission and cooperation of the cloud service provider. We hope that these results highlight that users should have the right to ask cloud service providers to document what additional defenses - either prevention or detection - if any they are providing to protect users from these types of attacks on their systems.

VII. Conclusion

This study demonstrates the degree to which virtual switches are vulnerable to Layer 2 network attacks. The Layer 2 vulnerabilities described in this paper are directed towards the virtual networking devices and not the hypervisor and without additional mitigation or preventive measures, could be performed on any host running a virtual switch including in a multi-tenant environment. Further study is necessary in order to perform a full Layer 2 security assessment on the state of virtual networking devices. The information could then be used to develop hardening and mitigation techniques focused on securing virtual networks against common Layer 2 networking threats. In their current state, virtual switches pose the same liability as their physical counterparts in terms of network security. One malicious virtual machine performing a MAC flooding attack against the virtual switch could be able to sniff all traffic passing over that virtual switch, potentially compromising the confidentiality, integrity, and availability of co-located clients.

References

[1] J. Pettit, J. Gross, B. Pfaff, M. Casado, and S. Crosby, "Virtual switching in an era of advanced edges," in *ITC 22 2nd Workshop on Data Center - Converged and Virtual Ethernet Switching (DC-CAVES)*, 2010.

[2] B. Pfaff, J. Pettit, T. Koponen, K. Amidon, M. Casado, and S. Shenker, "Extending networking into the virtualization layer," in *HotNets-VIII*, 2009.

[3] Cisco Systems, Inc. Cisco nexus 1000v series switches for vmware vsphere data sheet. [Online]. Available: http://www.cisco.com/en/US/prod/collateral/switches/ps9441/ps9902/data_sheet_c78-492971.html

[4] R. Seifert and J. Edwards, *The All-New Switch Book*. Indianapolis, Indiana: Wiley Publishing, Inc., 2008.

[5] LAN MAN Standards Committee, *IEEE Standard for Local and Metropolitan Area Networks: Media Access Control (MAC) Bridges*. New York, NY: The Institute of Electrical and Electronics Engineers, Inc., 2004.

[6] Kali Linux. The most advanced penetration testing distribution, ever. [Online]. Available: http://www.kali.org/

[7] CentOS. The centos project. [Online]. Available: http://www.centos.org

[8] thekellys.org. Dnsmasq - network services for small networks. [Online]. Available: http://www.thekelleys.org.uk/dnsmasq/doc.html

[9] P. N. Ayuso, P. McHardy, J. Kadlecsik, E. Leblond, and F. Westphal. The netfilter.org project. [Online]. Available: http://www.netfilter.org

[10] The Apache Software Foundation. The apache software foundation. [Online]. Available: http://www.apache.org

[11] National Vulnerability Database. Vulnerability summary for cve-2014-6271. [Online]. Available: http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2014-6271

[12] Cisco Systems, Inc. Catalyst 6500 release 12.2sx software configuration guide. [Online]. Available: http://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst6500/ios/12-2SX/configuration/guide/book/pref.html

[13] K.-H. Yeung, D. Fung, and K.-Y. Wong, "Tools for attacking layer 2 network infrastructure," in *IMECS '08 Proceedings of the International MultiConference of Engineers and Computer Scientists*, 2008, pp. 1143–1148.

[14] Microsoft. Hyper-v virtual switch overview. [Online]. Available: http://technet.microsoft.com/en-us/library/hh831823.aspx

[15] ——. What is server core? [Online]. Available: http://http://msdn.microsoft.com/en-us/library/dd184075.aspx

[16] VMware Inc. Vmware vsphere end user license agreement. [Online]. Available: http://www.vmware.com/download/eula/esxi50_eula.html

[17] B. Pfaff, R. Bull, and E. Jackson. mac-learning: Implement per-port mac learning fairness, openvswitch/ovs - github. [Online]. Available: https://github.com/openvswitch/ovs/commit/2577b9346b9b77feb94b34398b54b8f19fcff4bd

[18] B. Pfaff. [ovs-dev][patch] mac-learning: Implement per-port mac learning fairness. [Online]. Available: http://openvswitch.org/pipermail/dev/2015-February/051201.html

[19] ——. [ovs-dev][patch] mac-learning: Implement per-port mac learning fairness. [Online]. Available: http://openvswitch.org/pipermail/dev/2015-February/051228.html

[20] ELinks. Elinks full-featured text www browser. [Online]. Available: http://www.elinks.or.cz

[21] H. Altunbasak, S. Krasser, H. L. Owen, J. Grimminger, H.-P. Huth, and J. Sokol, "Securing layer 2 in local area networks," in *ICN'05 Proceedings of the 4th international conference on Networking - Volume Part II*, 2005, pp. 699–706.

[22] Cisco Systems, Inc. Vlan security white paper [cisco catalyst 6500 series switches]. [Online]. Available: http://www.cisco.com/en/US/products/hw/switches/ps708/products_white_paper09186a008013159f.shtml#wp39211

[23] K. Lauerman and J. King. Stp mitm attack and l2 mitigation techniques on the cisco catalyst 6500. [Online]. Available: http://www.cisco.com/c/en/us/products/collateral/switches/catalyst-6500-series-switches/white_paper_c11_605972.pdf/