ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 6, Special Issue 2, February 2017

National Conference on Recent Advancements in Engineering & Technology and Management 2017 (NCRAETM 2K17)

18th February 2017

Organized by

Sree Rama Engineering College (SRET), Tirupati- 517 507, India

Hub Attack Finding and Countermeasure Collection in Practical Set-Up Systems

Udayasree. C [1], Bharath kumar. B[2]

P.G. Student, Department of Computer Engineering, Sree Rama Engineering College, Tirupati, A.P, India[1]

Assistant Professor, Department of Computer Engineering, Sree Rama Engineering College, Tirupati, A.P, India[2]

ABSTRACT: We extensively review the literature on MITM to analyze and categories the scope of MITM attacks, considering both a reference model, such as the Open Systems Interconnection (OSI) model, as well as two specific widely used network technologies, i.e., GSM and UMTS. In particular, we classify MITM attacks based on several parameters, like location of an attacker in the network, nature of a communication channel, and impersonation techniques. Based on an impersonation techniques classification, we then provide execution steps for each MITM class. We survey existing countermeasures and discuss the comparison among them. Finally, based on our analysis, we propose a categorization of MITM prevention mechanisms, and we identify some possible directions for future research. we propose a multi-phase distributed vulnerability detection, measurement, and countermeasure selection mechanism called NETWORK INTRUSION, which is built on attack graph based analytical models and reconfigurable virtual network-based countermeasures. The proposed framework leverages Open Flow network programming APIs to build a monitor and control plane over distributed programmable virtual switches in order to significantly improve attack detection and mitigate attack consequences. The system and security evaluations demonstrate the efficiency and effectiveness of the proposed solution.

KEYWORDS: Network Security, Cloud Computing, Intrusion Detection, Attack Graph, Zombie Detection.

I. INTRODUCTION

The name Man-In-The-Middle is derived from the basketball scenario where two players intend to pass a ball to each other, while one player between them tries to seize it [1]. MITM attacks are sometimes referred to as bucket brigade attacks or fire brigade attacks [1]. Those names are derived from the fire brigade operation of dousing off the fire by passing buckets from one person to another, between the water source and the fire. MITM attack is also known as: Monkey-in-the middle attack, Session hijacking, TCP hijacking, TCP session hijacking [1]. A recent Cloud Security Alliance (CSA) survey shows that among all security issues, abuse and nefarious use of cloud computing is considered as the top security threat [1], in which attackers can exploit vulnerabilities in clouds and utilize cloud system resources to deploy attacks. In traditional data centers, where system administrators have full control over the host machines, vulnerabilities can be detected and patched by the system administrator in a centralized manner. However, patching known security holes in cloud data centers, where cloud users usually have the privilege to control software installed on their managed VMs, may not work effectively and can violate the *Service Level Agreement* (SLA). Furthermore, cloud users can install vulnerable software on their VMs, which essentially contributes to loopholes in cloud security. The challenge is to establish an effective vulnerability/attack detection and response system for accurately identifying attacks and minimizing the impact of security breach to cloud users. In [2], M. Armbrust *et al.* addressed that protecting ”Business continuity and services availability” from service outages is one of the top concerns in cloud computing systems. In a cloud system where the infras- tructure is shared by potentially millions of users, abuse and nefarious use of the shared infrastructure benefits attackers to exploit vulnerabilities of the cloud and use its resource to deploy attacks in more efficient ways [3]. Such attacks are more effective in the cloud environment since cloud users usually share computing resources, e.g., being connected through the same switch, sharing with the same data storage and file systems, even with potential attackers [4]. The similar setup for VMs in the cloud, e.g., virtualization techniques, VM OS, installed vulnerable software, networking, etc., attracts attackers to compromise multiple VMs.

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 6, Special Issue 2, February 2017

National Conference on Recent Advancements in Engineering & Technology and Management 2017 (NCRAETM 2K17)

18th February 2017

Organized by

Sree Rama Engineering College (SRET), Tirupati- 517 507, India

II. RELATED WORK

In this section, we present literatures of several highly related research areas to Network Intrusion, including: zombie detection and prevention, attack graph construction and security analysis, and software defined networks for attack countermeasures. The area of detecting malicious behavior has been well explored. The work by Duan *et al*. [6] focuses on the detection of compromised machines that have been recruited to serve as spam zombies. Their approach, SPOT, is based on sequentially scanning outgoing messages while employing a statistical method Sequential Probability Ratio Test (SPRT), to quickly determine whether or not a host has been compromised. Bot Hunter [7] detects compromised machines based on the fact that a thorough malware infection process has a number of well defined stages that allow correlating the intrusion alarms triggered by inbound traffic with resulting outgoing communication patterns. Bot Sniffer [8] exploits uniform spatial-temporal behavior characteristics of compromised machines to detect zombies by grouping flows according to server connections and searching for similar behavior in the flow.

III. NETWORK INTRUSION MODELS

In this section, we describe how to utilize attack graphs to model security threats and vulnerabilities in a virtual networked system, and propose a VM protection model based on virtual network reconfiguration approaches to prevent VMs from being exploited.

I. Threat Model

In our attack model, we assume that an attacker can be located either outside or inside of the virtual networking system. The attacker's primary goal is to exploit vulnerable VMs and compromise them as zombies. Our protection model focuses on virtual-network-based attack detection and reconfiguration solutions to improve the resiliency to zombie explorations. Our work does not involve host-based IDS and does not address how to handle encrypted traffic for attack detections.

II Attack Graph Model

An attack graph is a modeling tool to illustrate all possible multi-stage, multi-host attack paths that are crucial to understand threats and then to decide appropriate countermeasures [22]. In an attack graph, each node represents either precondition or consequence of an exploit. The actions are not necessarily an active attack since normal protocol interactions can also be used for attacks. Attack graph is helpful in identifying potential threats, possible attacks and known vulnerabilities in a cloud system.

III. VM Protection Model

The VM protection model of NETWORK INTRUSION consists of a VM profiler, a security indexer and a state monitor. We specify security index for all the VMs in the network depending upon various factors like connectivity, the number of vulnerabilities present and their impact scores. The impact score of a vulnerability, as defined by the CVSS guide [24], helps judge the confidentiality, integrity, and availability impact of the vulnerability being exploited. Connectivity metric of a VM is decided by evaluating incoming and outgoing connections.

III. SYSTEM DESIGN

In this section, we first present the system design overview of Network Intrusion and then detailed descriptions of its components.

A. SYSTEM DESIGN OVERVIEW

The proposed Network Intrusion framework is illustrated in figure 1. It shows the Network Intrusion framework within one cloud server cluster. Major components in this framework are distributed and light-weighted Network Intrusion-A on each physical cloud server, a network controller, a VM profiling server, and an attack analyzer. The latter three components are located in a centralized control center connected to software switches on each cloud server (i.e., virtual switches

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 6, Special Issue 2, February 2017

National Conference on Recent Advancements in Engineering & Technology and Management 2017 (NCRAETM 2K17)

18th February 2017

Organized by

Sree Rama Engineering College (SRET), Tirupati- 517 507, India

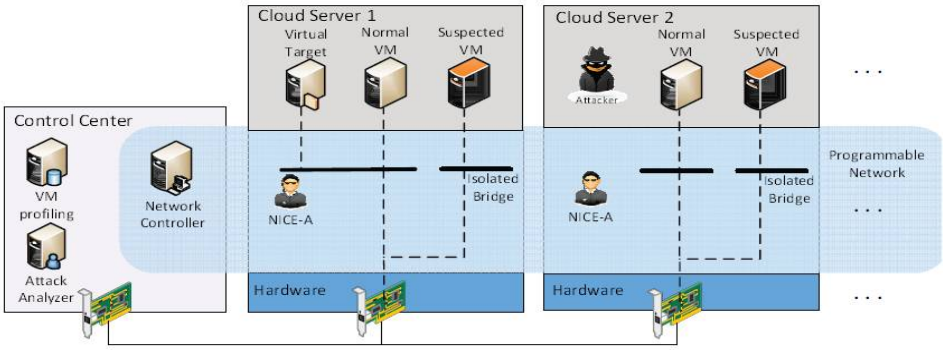

Fig. 1. NICE architecture within one cloud server cluster.

built on one or multiple Linux software bridges). Network Intrusion-A is a software agent implemented in each cloud server connected to the control center through a dedicated and isolated secure channel, which is separated from the normal data packets using Open Flow tunneling or VLAN approaches. The network controller is responsible for deploying attack countermeasures based on decisions made by the attack analyzer.

B. SYSTEM COMPONENTS

In this section we explain each component of Network Intrusion.

IV.II.I NETWORK INTRUSION-A The Network Intrusion-A is a Network-based Intrusion Detection System (NIDS) agent installed in either Dom0 or DomU in each cloud server. It scans the traffic going through Linux bridges that control all the traffic among VMs and in/out from the physical cloud servers. In our experiment, Snort is used to implement NETWORK INTRUSION -A in Dom0. It will sniff a mirroring port on each virtual bridge in the Open vSwitch. Each bridge forms an isolated subnet in the virtual network and connects to all related VMs. The traffic generated from the VMs on the mirrored software bridge will be mirrored to a specific port on a specific bridge using SPAN, RSPAN, or ERSPAN methods. The NETWORK INTRUSION-A sniffing rules have been custom defined to suite our needs.

IV.II.II VM Profiling

Virtual machines in the cloud can be profiled to get precise information about their state, services running, open ports, etc. One major factor that counts towards a VM profile is its connectivity with other VMs. Any VM that is connected to more number of machines is more crucial than the one connected to fewer VMs because the effect of compromise of a highly connected VM can cause more damage. Also required is the knowledge of services running on a VM so as to verify the authenticity of alerts pertaining to that VM. An attacker can use portscanning program to perform an intense examination of the network to look for open ports on any VM.

IV.II.III Attack Analyzer

The major functions of NETWORK INTRUSION system are performed by attack analyzer, which includes procedures such as attack graph construction and update, alert correlation and countermeasure selection. The process of constructing and utilizing the Scenario Attack Graph (*SAG*) consists of three phases: information gathering, attack graph construction, and potential exploit path analysis. With this information, attack paths can be modeled using SAG. Each node in the attack graph represents an exploit by the attacker. Each path from an initial node to a goal node represents a successful attack.

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 6, Special Issue 2, February 2017

National Conference on Recent Advancements in Engineering & Technology and Management 2017 (NCRAETM 2K17)

18th February 2017

Organized by

Sree Rama Engineering College (SRET), Tirupati- 517 507, India

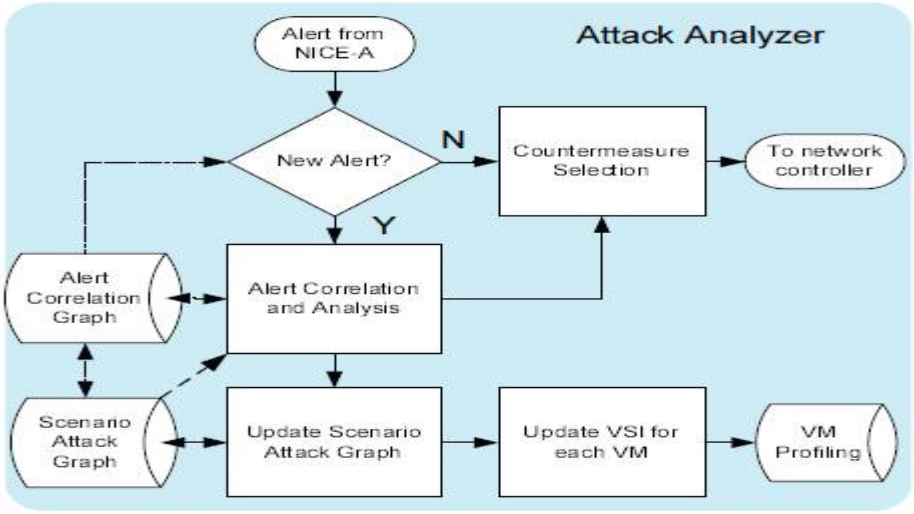

Fig. 2. Workflow of Attack Analyzer.

IV.II.IV Network Controller

The network controller is a key component to support the programmable networking capability to realize the virtual network reconfiguration feature based on Open- Flow protocol [20]. In NETWORK INTRUSION, within each cloud server there is a software switch, for example, Open vSwitch (OVS) [5], which is used as the edge switch for VMs to handle traffic in & out from VMs. The communication between cloud servers (i.e., physical servers) is handled by physical OpenFlow-capable Switch (OFS). In NETWORK INTRUSION, we integrated the control functions for both OVS and OFS into the network controller that allows the cloud system to set security/filtering rules in an integrated and comprehensive manner.

IV. NETWORK INTRUSION SECURITY MEASUREMENT, ATTACK MITIGATION AND COUNTERMEASURES

In this section, we present the methods for selecting the countermeasures for a given attack scenario. When vulnerabilities are discovered or some VMs are identified as suspicious, several countermeasures can be taken to restrict attackers' capabilities and it's important to differentiate between compromised and suspicious VMs. The countermeasure serves the purpose of 1) protecting the target VMs from being compromised; and 2) making attack behavior stand prominent so that the attackers' actions can be identified.

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 6, Special Issue 2, February 2017

National Conference on Recent Advancements in Engineering & Technology and Management 2017 (NCRAETM 2K17)

18th February 2017

Organized by

Sree Rama Engineering College (SRET), Tirupati- 517 507, India

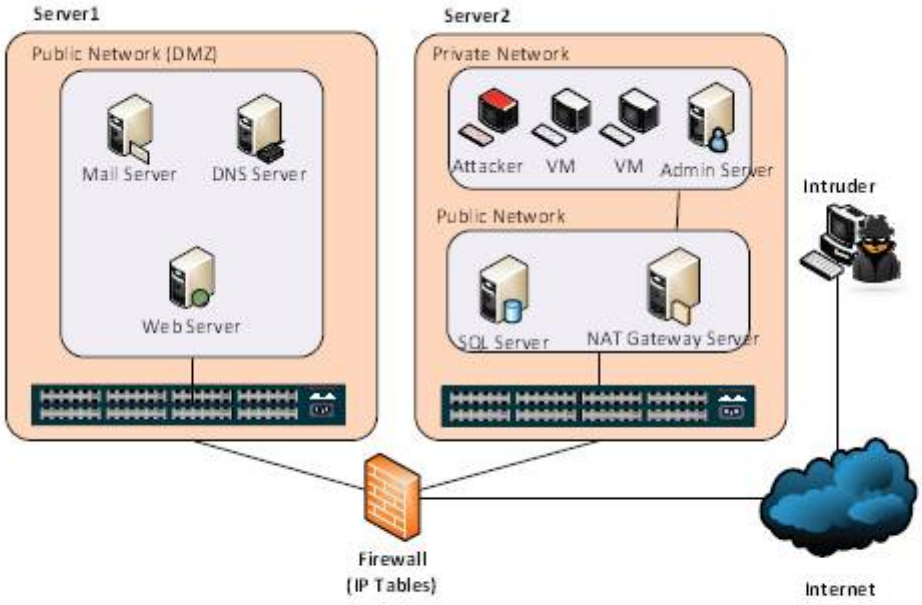

Fig. 3. Virtual network topology for security evaluation.

V. PERFORMANCE EVALUATION

In this section we present the performance evaluation of NETWORK INTRUSION. Our evaluation is conducted in two directions: the security performance, and the system computing and network reconfiguration overhead due to introduced security mechanism.

VI.I. Security Performance Analysis

To demonstrate the security performance of Network Intrusion, we created a virtual network testing environment consisting of all the presented components of Network Intrusion.

VI.I.II Attack Graph and Alert Correlation

The attack graph can be generated by utilizing network topology and the vulnerability information, and it is shown in Figure 4. As the attack progresses, the system generates various alerts that can be related to the nodes in the attack graph.

VI.I.III Countermeasure Selection

To illustrate how NETWORK INTRUSION works, let us consider for example, an alert is generated for node 16 (*vAlert* = 16) when the system detects LICQ Buffer overflow. After the alert is generated, the cumulative probability of node 16 becomes 1 because that attacker has already compromised that node. This triggers a change in cumulative probabilities of child nodes of node 16. Now the next step is to select the countermeasures from the pool of countermeasures *CM*. If the countermeasure *CM4: create filtering rules* is applied to node 5 and we assume that this countermeasure has effectiveness.

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 6, Special Issue 2, February 2017

National Conference on Recent Advancements in Engineering & Technology and Management 2017 (NCRAETM 2K17)

18th February 2017

Organized by

Sree Rama Engineering College (SRET), Tirupati- 517 507, India

VI.I.IV Experiment in Private Cloud Environment

Previously we presented an example where the attackers target is VM in the private network. For performance analysis and capacity test, we extended the configuration in figure 3 to create another test environment which includes 14 VMs across 3 cloud servers and configured each VM as a target node to create a dedicated SAG for each VM. These VMs consist of Windows (W) and Linux (L) machines in the private network 172.16.11.0/24, and contains more number of vulnerabilities related to their OSes and applications. We created penetration testing scripts with Metasploit framework [31] and Armitage [32] as attackers in our test environment. These scripts emulate attackers from different places in the internal and external sources, and launch diversity of attacks based on the vulnerabilities in each VM.

VI.I.V False Alarms

A cloud system with hundreds of nodes will have huge amount of alerts raised by Snort. Not all of these alerts can be relied upon, and an effective mechanism is needed to verify if such alerts need to be addressed. Since Snort can be programmed to generate alerts with CVE id, one approach that our work provides is to match if the alert is actually related to some vulnerability being exploited. If so, the existence of that vulnerability in SAG means that the alert is more likely to be a real attack. Thus, the false positive rate will be the joint probability of the correlated alerts, which will not increase the false positive rate compared to each individual false positive rate.

VI.II Network Intrusion System Performance

We evaluate system performance to provide guidance on how much traffic Network Intrusion can handle for one cloud server and use the evaluation metric to scale up to a large cloud system. In a real cloud system, traffic planning is needed to run Network Intrusion, which is beyond the scope of this paper. Due to the space limitation, we will investigate the research involving multiple cloud clusters in the future. To demonstrate the feasibility of our solution, comparative studies were conducted on several virtualization approaches. We evaluated Network Intrusion based on Dom 0 and Dom U implementations with mirroring-based and proxy-based attack detection agents (i.e., Network Intrusion-A). In mirror-based IDS scenario, we established two virtual networks in each cloud server: normal network and monitoring network.

VI. CONCLUSION

In this paper, we presented Network Intrusion, which is proposed to detect and mitigate collaborative attacks in the cloud virtual networking environment. Network Intrusion utilizes the attack graph model to conduct attack detection and prediction. The proposed solution investigates how to use the programmability of software switches based solutions to improve the detection accuracy and defeat victim exploitation phases of collaborative attacks. The system performance evaluation demonstrates the feasibility of Network Intrusion and shows that the proposed solution can significantly reduce the risk of the cloud system from being exploited and abused by internal and external attackers. Network Intrusion only investigates the network IDS approach to counter zombie explorative attacks. In order to improve the detection accuracy, host-based IDS solutions are needed to be incorporated and to cover the whole spectrum of IDS in the cloud system. This should be investigated in the future work. Additionally, as indicated in the paper, we will investigate the scalability of the proposed Network Intrusion solution by investigating the decentralized network control and attack analysis model based on current study.

REFERENCES

[1] Coud Sercurity Alliance, “Top threats to cloud computing v1.0,” https://cloudsecurityalliance.org/topthreats/csathreats.v1.0.pdf, March 2010.
[2] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, “A view of cloud computing,” *ACM Commun.*, vol. 53, no. 4, pp. 50–58, Apr. 2010.
[3] B. Joshi, A. Vijayan, and B. Joshi, “Securing cloud computing environment against DDoS attacks,” *IEEE Int’l Conf. Computer Communication and Informatics (ICCCI ’12)*, Jan. 2012.
[4] H. Takabi, J. B. Joshi, and G. Ahn, “Security and privacy challenges in cloud computing environments,” *IEEE Security & Privacy*, vol. 8, no. 6, pp. 24–31, Dec. 2010.
[5] “Open vSwitch project,” http://openvswitch.org, May 2012.

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 6, Special Issue 2, February 2017

National Conference on Recent Advancements in Engineering & Technology and Management 2017 (NCRAETM 2K17)

18th February 2017

Organized by

Sree Rama Engineering College (SRET), Tirupati- 517 507, India

[6] Z. Duan, P. Chen, F. Sanchez, Y. Dong, M. Stephenson, and J. Barker, "Detecting spam zombies by monitoring outgoing messages," *IEEE Trans. Dependable and Secure Computing*, vol. 9, no. 2, pp. 198–210, Apr. 2012. IEEE TRANSACTIONS ON DEPEDABLE AND SECURE COMPUTING

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. 14 [7] G. Gu, P. Porras, V. Yegneswaran, M. Fong, and W. Lee, "BotHunter: detecting malware infection through IDS-driven dialog correlation," *Proc. of 16th USENIX Security Symp. (SS '07)*, pp. 12:1–12:16, Aug. 2007.

[8] G. Gu, J. Zhang, and W. Lee, "BotSniffer: detecting botnet command and control channels in network traffic," *Proc. of 15th Ann.Network and Distributed Sytem Security Symp. (NDSS '08)*, Feb. 2008.

[9] O. Sheyner, J. Haines, S. Jha, R. Lippmann, and J. M. Wing, "Automated generation and analysis of attack graphs," *Proc. IEEE Symp. on Security and Privacy*, 2002, pp. 273–284.

[10] "NuSMV: A new symbolic model checker," http://afrodite.itc.it: 1024/□nusmv. Aug. 2012.

[11] S. H. Ahmadinejad, S. Jalili, and M. Abadi, "A hybrid model for correlating alerts of known and unknown attack scenarios and updating attack graphs," *Computer Networks*, vol. 55, no. 9, pp. 2221–2240, Jun. 2011.

[12] X. Ou, S. Govindavajhala, and A. W. Appel, "MulVAL: a logicbased network security analyzer," *Proc. of 14th USENIX Security Symp.*, pp. 113–128. 2005.

[13] R. Sadoddin and A. Ghorbani, "Alert correlation survey: framework and techniques," *Proc. ACM Int'l Conf. on Privacy, Security and Trust: Bridge the Gap Between PST Technologies and Business Services (PST '06)*, pp. 37:1–37:10. 2006.

[14] L. Wang, A. Liu, and S. Jajodia, "Using attack graphs for correlating, hypothesizing, and predicting intrusion alerts," *Computer Communications*, vol. 29, no. 15, pp. 2917–2933, Sep. 2006.

[15] S. Roschke, F. Cheng, and C. Meinel, "A new alert correlation algorithm based on attack graph," *Computational Intelligence in Security for Information Systems*, LNCS, vol. 6694, pp. 58–67. Springer, 2011.

[16] A. Roy, D. S. Kim, and K. Trivedi, "Scalable optimal countermeasure selection using implicit enumeration on attack countermeasure trees," *Proc. IEEE Int'l Conf. on Dependable Systems Networks (DSN '12)*, Jun. 2012.

[17] N. Poolsappasit, R. Dewri, and I. Ray, "Dynamic security risk management using bayesian attack graphs," *IEEE Trans. Dependable and Secure Computing*, vol. 9, no. 1, pp. 61–74, Feb. 2012.

[18] Open Networking Fundation, "Software-defined networking: The new norm for networks," *ONF White Paper*, Apr. 2012.

[19] "Openflow." http://www.openflow.org/wp/learnmore/, 2012.

[20] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: enabling innovation in campus networks," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, Mar. 2008.

[21] E. Keller, J. Szefer, J. Rexford, and R. B. Lee, "NoHype: virtualized cloud infrastructure without the virtualization," *Proc. of the 37th ACM ann. int'l symp. on Computer architecture (ISCA '10)*, pp. 350–361. Jun. 2010.

[22] X. Ou, W. F. Boyer, and M. A. McQueen, "A scalable approach to attack graph generation," *Proc. of the 13th ACM conf. on Computer and communications security (CCS '06)*, pp. 336–345. 2006.

[23] Mitre Corporation, "Common vulnerabilities and exposures, CVE," http://cve.mitre.org/. 2012.

[24] P. Mell, K. Scarfone, and S. Romanosky, "Common vulnerability scoring system (CVSS)," http://www.first.org/cvss/cvss-guide. html, May 2010.

[25] O. Database, "Open source vulnerability database (OVSDB)," http://osvdb.org/. 2012

[26] NIST, "National vulnerability database, NVD," http://nvd.nist. gov. 2012

[27] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, and S. Shenker, "NOX: towards an operating system for networks," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 3, pp. 105–110, Jul. 2008.

[28] X. Ou and A. Singhal, *Quantitative Security Risk Assessment of Enterprise Networks*. Springer, Nov. 2011.

[29] M. Frigault and L. Wang, "Measuring network security using bayesian Network-Based attack graphs," *Proc. IEEE 32nd ann. int'l conf. on Computer Software and Applications (COMPSAC '08)*, pp. 698–703. Aug. 2008.

[30] K. Kwon, S. Ahn, and J. Chung, "Network security management using ARP spoofing," *Proc. Int'l Conf. on Computational Scienceand Its Applications (ICCSA '04)*, LNCS, vol. 3043, pp. 142–149,Springer, 2004.

[31] "Metasploit," http://www.metasploit.com. 2012.

[32] "Armitage," http://www.fastandeasyhacking.com. 2012.

[33] M. Tupper and A. Zincir-Heywood, "VEA-bility security metric: A network security analysis tool," *Proc. IEEE Third Int'l Conf. on Availability, Reliability and Security (ARES '08)*, pp. 950–957, Mar. 2008.