UIDN: UNMASKING INFRINGEMENTS AND DRAFTING COUNTER-AGENTS IN VIRTUAL NETWORK SYSTEMS

Payaswini Y K
MTech-Student,
Atria Institute of Technology,
payaswini.tambakad@gmail.com

Tejashwini P S
MTech-Student,
Atria Institute of Technology,
tejashwini1007@gmail.com

Prof. Manjula M
Assistant Professor,
Atria Institute of Technology,
manjula.m82@gmail.com

ABSTRACT

Cloud security is one of most important issues that have attracted a lot of people in research and development field. Particularly, attackers can explore vulnerabilities of a cloud system and compromise virtual machines to deploy further large-scale Distributed Denial-of-Service (DDoS). To prevent vulnerable virtual machines from being compromised in the cloud, we propose multiphase distributed vulnerability detection, measurement, and counter-agent selection mechanism called UIDNS, which is built on attack graph-based analytical models and reconfigurable virtual network-based counter-agents. The proposed framework leverages OpenFlow network programming APIs to build a monitor and control plane over distributed programmable virtual switches to significantly improve attack detection and mitigate attack sequences. The system and security features demonstrate the efficiency and effectiveness of the proposed solution.

Keywords

Cloud computing, infringement detection, attack graph, zombie detection, network security.

1. INTRODUCTION

Recent studies have shown that users migrating to the cloud consider security as the most important factor. A recent Cloud Security Alliance (CSA) survey shows that among all security issues, abusive 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 form front their attacks. In classical data centers, where system administrators have full control over the host machines, susceptibility can be detected and rebuilt by the system administrator in a centralized manner. However, rebuilding known security holes in cloud data centers, where cloud users usually have the advantage to control software installed on their managed VMs, may not work effectively and can breach the service level agreement (SLA). Furthermore, cloud users can install vulnerable software on their VMs, which essentially contributes to threats in cloud security. The challenge is to establish a useful susceptibility/attack detection and response system for accurately identifying attacks and minimizing the impact of security breach to cloud users. In [2], 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 computing systems, where the infrastructure is shared by potentially millions of abusive users, the shared infrastructure benefits the attackers to exploit vulnerabilities of the cloud and use its resource to form front attacks in more efficient ways [3]. Such attacks are more effective in the cloud environment because cloud users usually share computing resources, e.g., being connected through the same network, sharing with the same file and media, even with potential

attackers [4]. The similar setup for Virtual Machines in the cloud, e.g., techniques of virtualization, Vitual Machine OS, installed susceptible software, networking, and so on, attracts assaulter to compromise multiple VMs. In this paper, we propose Unmasking infringements and drafting counter-measure in virtual network systems (UIDN) to establish a defence-in-depth infringement detection framework. For better attack detection, UIDNS incorporates attack graph analytical procedures into the infringement detection processes. We must note that the design of UIDNS does not intend to improve any of the existing infringement detection algorithms; indeed, UIDNS employs a reconfigurable virtual networking approach to detect and counter the attempts to compromise VMs, thus preventing malicious VMs. In general, UIDNS includes two main phases: 1) deploy a lightweight mirroring-based network infringement detection agent (UIDN-A) on each cloud server to capture and analyse cloud traffic. A UIDN-A periodically scans the virtual system vulnerabilities within a cloud server to establish Attack Graphs, and then based on the severity of identified vulnerability toward the collaborative attack goals, UIDNS will decide whether or not to put a VM in network supervision state. 2) Once a VM enters supervision state, Deep Packet Inspection (DPI) is applied, and/or virtual network reconfigurations can be deployed to the supervising VM to make the potential attack behaviours prominent. UIDN significantly advances the current network IDS/IPS solutions by employing programmable virtual networking approach that allows the system to construct a dynamic reconfigurable IDS system. By using software switching techniques, UIDNS constructs a mirroring-based traffic capturing framework to minimize the interference on users' traffic compared to traditional bump-in-the-wire (i.e., proxy-based) IDS/IPS. The programmable virtual networking architecture of UIDNS enables the cloud to establish inspection and quarantine modes for suspicious VMs according to their current vulnerability state in the current SAG. Based on the behavior of VMs in the SAG, UIDN can decide appropriate actions, for example, DPI or packet filtering, on the suspicious VMs. Using this approach, UIDN does not need to block traffic flows of a suspicious VM in its early attack stage. The contributions of UIDN are presented as follows:

- We devise UIDN, a new multiphase distributed network infringement detection and prevention framework in a virtual networking environment that captures and inspects suspicious cloud traffic without interrupting users' applications and cloud services.
- UIDN incorporates a software switching solution to quarantine and inspect suspicious VMs for further investigation and protection. Through arrangeable network approaches, UIDN can improve the attack detection probability and improve the resiliency to VM exploitation attack without interrupting existing normal cloud services. .
- A UIDN employs a novel attack graph approach for attack detection and prevention by correlating attack behaviour and also suggests effective counter-agents. A UIDN optimizes the implementation on cloud servers to minimize resource utilization. Our study shows that UIDNS consumes less computational overhead compared to proxy-based network infringement detection solutions.

2. RELATED WORK

In this section, we present literatures of several highly related research areas to UIDN, including: Zombie detection and infringement, attack graph construction and security study, and software defined networks for attack counter-agents. The area of detecting malicious behaviour has been well explored. The work by Duan [6] focuses on the detection of compromised machines that have been recruited to serve as spam zombies. Their

approach is based on sequentially scanning outgoing messages while employing a statistical method Sequential Probability Ratio Test (SPRT), to quickly determine whether a host has been compromised. BotHunter [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 infringement alarms triggered by inbound traffic with resulting outgoing communication patterns. BotSniffer [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. An attack graph is able to represent an array of exploits, called minute attacks that lead to a displeasing state, for example, a state where an attacker has obtained administrative access to a machine.

3. UIDN MODELS

This section describes how to utilize attack graphs to model security threats and susceptibilities in a virtual networked system, and propose a VM protection model based on virtual network reconfiguration approaches to prevent VMs from being exploited.

3.1 Hazard 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 intent is to exploit vulnerable VMs and compromise them as zombies. Our protection model focuses on network-based- virtual attack detection and reconfiguration solutions to improve the resiliency to zombie explorations. The paper does not involve the work related to host-based IDS and does not address how to handle encrypted traffic for attack detections. Our proposed solution can be employed in an Infrastructure-as-a-Service (IaaS) cloud networking system, and we assume that the Service Provider of the cloud is benign. Cloud service users are free to install whatever operating systems or applications they want, even if such action may introduce susceptibilities to their controlled VMs. We assume that the hypervisor is secure and free of any susceptibilities. The issue of a malicious tenant breaking out of DomU and gaining access to physical server have been studied in recent work.

3.2 Attack Graph Model

An attack graph is a modelling tool to illustrate all possible multistage, multihost attack paths that are acute to understand threats and then to decide appropriate counter-agents. In an attack graph, each node represents either precondition or consequence of an adventure. The actions are not necessarily an active attack because normal protocol interactions can also be used for attacks. Attack graph is helpful in identifying hidden threats, possible attacks, and known susceptibilities in a cloud system. Since the attack graph provides specifics of all known susceptibilities in the system and the connectivity information, we get a whole picture of existing security situation of the system, where we can predict the possible hazards and attacks by correlating detected events or activities. If an event is recognized as a hidden attack, we can apply specific counteragents to mitigate its impact or take actions to prevent it from contaminating the cloud system.

4. UIDN SYSTEM DESIGN

In this section, we first present the system design analysis of UIDN and then detailed descriptions of its components. The proposed UIDN structure is illustrated in Fig. 1. It shows the UIDN framework within one cloud server cluster. Major components in this structure are distributed and light-weighted UIDN-A on each physical cloud server, a network manager, profiling server, and an attack evaluator. The latter three components are located in a centralized control center connected to software switches on each cloud server (i.e., virtual switches built on one or multiple Linux software bridges).

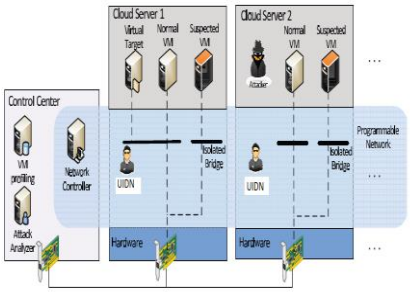

Figure 1: UIDN architecture with one cloud server cluster.

UIDN-A is a software agent implemented in each cloud server connected to the control center through a dedicated and isolated secure passage, which is separated from the normal data packets using OpenFlow tunnelling or VLAN approaches. The network manager is responsible for deploying attack counter-agents based on decisions made by the attack evaluator. In the following description, our terminologies are based on the XEN virtualization technology. UIDN-A is a network infringement detection engine that can be installed in either Dom0 or DomU of a XEN cloud server to capture and filter malicious traffic. infringement detection alerts are sent to control center when suspicious or anomalous traffic is detected. After receiving an alert, attack evaluator evaluates the severity of the alert based on the attack graph, decides what counter-agent approach to take, and then initiates it through the network manager. An attack graph is established according to the susceptibility information derived from both offline and real-time susceptibility searches. Offline searching can be done by running penetration tests and online real-time vulnerability scanning can be triggered by the network manager (e.g., when new ports are opened and identified by OFSs) or when new alerts are generated by the UIDN-A. Once new vulnerabilities are discovered or counter-agents are employed, the attack graph will be reconstructed.

5. SYSTEM COMPONENTS

In this section, we explain each component of UIDN.

5.1 UIDN-A

The UIDN-A is a Network-based Intrusion Detection System (NIDS) agent installed in either Dom0 or DomU in each cloud server. It searches the traffic going through Linux bridges that control all the traffic among VMs and in/out from the physical cloud servers. In our analysis, Snort is used to implement UIDN-A in Dom0. It will sniff a mirroring port on each virtual bridge in the Open vSwitch (OVS) [5]. Each bridge forms an isolated subnet in the virtual network and connects to all related Virtual Machines. The traffic generated from the Virtual Machines on the mirrored software bridge will be mirrored to a specific port on a specific bridge using SPAN, RSPAN, or ERSPAN methods. The UIDN-A sniffing rules have been custom defined to suite our needs. Dom0 in the Xen environment is an advantageous domain,that includes a virtual switch for traffic switching among VMs and network drivers for physical network interface of the cloud server. It is more efficient to scan the network traffic in Dom0 because all traffic in the cloud server needs go through it; however, our design is not dependent on the installed VM. We must note that the alert detection quality of UIDN-A depends on the implementation of UIDN-A, which uses Snort. We do not focus on the detection accuracy of Snort in this paper. Thus, there is no change in the individual alert detection's false alarm rate. However, the architectural design helps to reduce the false alarm rate.

5.2 Profiling Server

Virtual machines in the cloud can be profiled to get correct information about their services running, open ports, state and so on. One major factor that counts toward a profiling server is its connectivity with other Virtual Machines. Any Virtual Machine that is connected to more number of machines is more crucial than the one connected to fewer Virtual Machines

because the effect of compromise of a highly connected Virtual Machine can cause more harm. Also required is the knowledge of services running on a Virtual Machine so as to verify the authenticity of alerts pertaining to that Virtual Machine. An attacker can use port-scanning program to perform an intense examination of the network to look for open ports on any Virtual Machine. So information about any open ports on a Virtual Machine and the history of opened ports plays a significant role in determining how vulnerable the Virtual Machine is. All these factors combined will form the Virtual Machine profile. Virtual Machine profiles are maintained in a database and contain comprehensive information about infringements, alert, and traffic. The data comes from: Attack graph developer. While developing the attack graph, every detected susceptibility is added to its corresponding VM entry in the database. UIDN-A. The alert involving the VM will be recorded in the VM profile database. The traffic patterns involving the VM are based on five tuples (source MAC address, destination MAC address, source IP address, destination IP address, protocol). We can have traffic of the pattern, where packets radiate from a single IP and are delivered to multiple destination IP addresses, and the reverse is also true.

5.3 Attack Evaluator

The major functions of UIDN system are performed by attack evaluator, which includes procedures such as attack graph construction and update, alert correlation, and counter-agent selection. The process of constructing and utilizing the Attack Graph consists of three phases: gathering of information, attack graph construction, and potential exploit path analysis. With this information, attack trails can be modelled using Attack Graph. Each node in the attack graph represents an exploit by the attacker. Each trail from an initial node to a goal node represents a successful attack. In summary, UIDN attack graph is constructed based on the following information: Cloud system information is collected from the node controller (i.e., Dom0 in XenServer). The information includes the number of Virtual Machines in the cloud server, running services on each Virtual Machine, and Virtual Machine's Virtual Interface (VIF) information. Virtual network topography and configuration information is collected from the network manager, which includes virtual network topology, host connectivity, Virtual Machine connectivity, every MAC address, VM's IP address, port information, and traffic flow information. Susceptibility information is generated by both on demand susceptibility scanning (i.e., initiated by the network manager and UIDN-A) and regular penetration testing using the well-known vulnerability databases, such as Open Source susceptibility Database (OSVDB), Common Vulnerabilities and Exposures List (CVE), and NIST National susceptibility Database (NVD). The attack evaluator also handles alert correlation and analysis operations. This element has two major functions:

- Constructs ACG
- Provides threat information and appropriate counter-measure to network manager for virtual network reconfiguration. Fig. 2 shows the workflow in the attack evaluator component. After receiving an alert from UIDN-A, alert evaluator matches the alert in the ACG. Incase the alert already exists in the graph and it is a known attack (i.e., matching the attack signature), the attack evaluator performs counter selection procedure and then notifies network manager immediately to deploy counter-agent or mitigation actions. If the alert is new, attack evaluator will perform alert correlation and analysis. This algorithm correlates each new alert to a matching alert correlation set (i.e., in the same attack scenario). A selected counter-measure is applied by the network manager based on the severity of evaluation results. If the alert is a new susceptibility and is not present in the UIDN attack graph, the

attack evaluator adds it to attack graph and then reconstructs it.

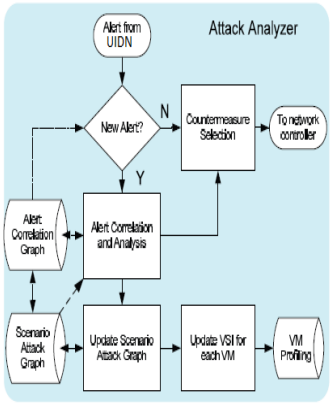

Figure 2: Workflow of Attack Evaluator

5.4 Network Manager

The network manager is a key component to support the programmable networking capability to realize the virtual network reconfiguration feature based on OpenFlow protocol. In UIDN, within each cloud server there is a software switch, which is used as the edge switch for evaluator to construct attack graphs. Through the cloud internal discovery program of studies that use DNS, DHCP, LLDP, and flow initiations, network manager is able to discover the network connectivity information from OVS and OFS. This data includes current data paths on each switch and detailed flow information associated with these paths, such as TCP/IP and MAC header. Network flow and topography change information will be automatically sent to the manager and then delivered to attack evaluator to reconstruct attack graphs. Yet another important function of the network manager is to assist the attack evaluator module. According to the OpenFlow protocol, when the controller receives the first packet of a flow, it holds the packet and analyzes the flow table for complying traffic policies. In UIDN, the network control also consults with the attack evaluator for the flow access control by setting up the filtering rules on the corresponding OVS and OFS. Once a traffic flow is admitted, the flow of packets are not handled by the network manager, but monitored by the UIDN-A. Network manager is also responsible for applying the counter-measure from attack evaluator. Based on VM Security Index (VSI) and severity of alert, counter-measures are selected by UIDN and executed by the network manager. If a severe alert is triggered and identifies some known attacks, or a VM is detected as susceptible, the network manager will block the VM immediately. An alert with average threat level is triggered by a suspicious compromised VM. Counter-measure in such case is to put the suspicious VM with exploited state into quarantine mode and redirect all its flows to UIDN-A DPI mode. An alert with a minor hindrance level can be generated due to the presence of a vulnerable VM. For this case, to interrupt the VM's normal traffic, doubtful traffic to/from the VM will be put into inspection mode, in which actions such as opposing its flow bandwidth and changing network configurations will be taken to force the attack exploration behaviour to stand out.

6. CONCLUSION

In this paper, we presented UIDN, which is proposed to identify and mitigate collaborative attacks in the cloud virtual networking environment. UIDN uses the attack graph model to conduct attack identification and prediction. The proposed solution investigates how to use the efficiency of software switches-based solutions to improve the detection accuracy and defeat victim exploitation phases of collaborative attacks. The system performance evaluation demonstrates the workability of UIDN 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. UIDN only utilizes the network IDS approach to counter zombie explorative attacks. 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. Additionally, we will

investigate the scalability of the proposed UIDN solution by investigating the decentralized network control and attack analysis model based on current study.

7. REFERENCES

[1]. Cloud Security Alliance, "Top Threats to Cloud ComputingV1.0,"https://cloudsecurityalliance. org/topthreats/csathreats. v1.0.pdf, Mar. 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 Comm., vol. 53, no. 4, pp. 50-58, Apr. 2010.

[3]. B. Joshi, A. Vijayan, and B. Joshi, "Securing Cloud Computing Environment Against DDoS Attacks," Proc. IEEE Int'l Conf.Computer Comm. 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 and Privacy, vol. 8, no. 6, pp. 24-31, Dec. 2010.

[5]. "Open vSwitch Project," http://openvswitch.org, May 2012.

[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.

[7] G. Gu, P. Porras, V. Yegneswaran, M. Fong, and W.Lee, "BotHunter: Detecting Malware Infection through IDS-driven Dialog Correlation," Proc. 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. 15th Ann. Network and Distributed Sytem Security Symp. (NDSS '08), Feb.2008.