2014 ASE BIGDATA/SOCIALCOM/CYBERSECURITY Conference, Stanford University, May 27-31, 2014

An Analysis of the Virtual Machine Migration Incurred Security Problems in the Cloud

Beaulah Navamani, Chuan Yue, Xiaobo Zhou, Edward Chow
University of Colorado Colorado Springs
Department of Computer Science
Email: {bnavaman, cyue, xzhou, cchow}@uccs.edu

Abstract

Cloud centric computing is a significant trend in computing. It is very appealing because it can offer a number of benefits such as cost efficiency, elasticity, scalability, and convenience to millions of organizations and end users. On the other hand, cloud computing creates many new security problems that should be properly addressed for its wide and successful adoption. In this paper, we analyze a new category of virtual machine (VM) migration incurred security problems. The migration of VM instances in the cloud is highly desirable and even inevitable for reasons such as load balancing and energy saving. We pinpoint that the migration of VM instances from one physical machine to another can weaken or even nullify the security protections provided by the intrusion prevention systems and intrusion detection systems to the original VM instances. We further analyze the root cause of this category of VM migration incurred security problems and discuss the potential approaches for addressing them.

Keywords: Cloud; Virtual Machine Instance; Migration; Security; Amazon EC2; Intrusion Prevention; Intrusion Detection

1 Introduction

Cloud centric computing is a significant trend in computing. While offering a number of benefits such as cost efficiency, elasticity, scalability, and convenience to millions of organizations and end users, cloud computing also creates many new security problems from both technical and non-technical perspectives such as unexpected side channels and covert channels, mutual auditability, and reputation fate-sharing in multi-tenant environments [1, 2, 3, 4, 5].

The benefits of cloud computing are, to a large extent, brought by its capability of provisioning virtual machine (VM) instances elastically to different physical machines (even with different geographical locations) and migrating the VM instances whenever necessary. Especially, researchers have intensively investigated different VM migration schemes to better achieve objectives such as fault management, load balancing, energy saving, and performance improvement [6, 7, 8, 9, 10, 11, 12].

However, the research on the VM migration related security problems is still very limited. Existing efforts mainly focus on the potential attacks that may occur during the live migration of VM instances, with a threat model in which attackers with high privileges such as administrator permissions can compromise the confidentiality, integrity, and availability of the lively migrated VM instances (see Section 2). Unfortunately, many practical and severe security problems that can be incurred by the VM migration have not been carefully investigated yet.

In this paper, we analyze a new category of VM migration incurred security problems. We pinpoint that the migration of VM instances from one physical machine to another can weaken or even nullify the security protections provided by the intrusion prevention systems and intrusion detection systems to the original VM instances. Especially, we exemplify some typical VM migration incurred security problems by using the very popular Amazon EC2 (Elastic Compute Cloud) cloud computing environment [13], the default Netfilter [14] Linux intrusion prevention (i.e., firewall) system, and the very popular Snort intrusion detection system [15]. VM migration incurred problems can cause severe consequences because they immediately provide many vectors for attackers to perform different types of malicious activities. We further analyze the root cause of this category of VM migration incurred security problems and discuss three potential approaches for addressing them.

The rest of the paper is structured as follows. Section 2 reviews the related work on VM migration security. Section 3 exemplifies and analyzes the typical VM migration incurred security problems. Section 4 discusses the potential approaches for addressing the VM migration incurred security problems. Section 5 concludes the paper.

2 Related Work

Existing research on the VM migration related security problems mainly focuses on the potential attacks that may occur during the live migration of VM instances.

In [16], Oberheide et al. empirically demonstrated that both Xen and VMware are vulnerable to practical man-in-the-middle attacks targeting their live migration functionality; these attacks can manipulate the memory of a VM as it traverses the network during a live migration. In [17], Santos et al. proposed a trusted cloud computing platform that allows users to verify the confidentiality and integrity of their data and computation; this platform can be used to secure the VM launch and live migration operations. In [18], Duncan et al. considered the potential for insider attackers such as cloud providers to harvest customer data while VMs are migrated, and asserted that digital-forensic techniques can be useful in detecting such insider attacks and may deter many potential malicious insiders from carrying out such attacks.

In [19], Danev et al. considered the problem of enabling secure migration of virtual TPM (Trusted Platform Module) [20] based VMs in private clouds, and proposed a secure migration protocol. In [21], Aslam et al. proposed a secure VM migration mechanism that uses TPM [20] capabilities to guarantee that a VM is migrated only to a trustworthy cloud platform. Another TPM based scheme is presented in [22], in which a role-based mechanism with remote attestation is used and the VM migration is controlled by specific policies that are protected in the sealed storage.

Our work is different from these related ones because we focus on investigating the problems that the migration of VM instances from one physical machine to another can weaken or even nullify the security protections provided by the intrusion prevention systems and intrusion detection systems to the original VM instances. This category of security problems can cause severe consequences but have not been carefully analyzed before.

3 Migration Incurred Security Problems

When VM instances are deployed into the cloud, they are often protected by different types of security systems such as intrusion prevention systems (IPSs) and intrusion detection systems (IDSs). Signature-based and anomaly-based security rules are widely used in IPSs and IDSs to detect malicious or suspicious activities and perform the protection actions such as raising alarms, dropping packets, and rejecting connections.

In the typical IPSs and IDSs, machines' identifiers such as IP addresses and hostnames are often used to define the security rules. For example, one IPS security rule may reject any incoming HTTP request sent to the IP address of the local machine, and one IDS security rule may raise an alarm if any outgoing TCP connection is initiated from the IP address of the local machine.

Without migration, the IPS and IDS security rules defined for a machine may work properly as designed and deployed. However, with migration, if the identifiers of machines are changed but the identifiers used in the security rules do not change accordingly, many rules could be weakened or even nullified and severe consequences could be incurred. As introduced in Section 1, in the cloud computing environments, the migration of VM instances is highly desirable for better achieving the objectives such as fault management, load balancing, energy saving, and performance improvement [6, 7, 8, 9, 10, 11, 12]. Indeed, the migration of VM instances is often even inevitable. For example, if we stop a VM instance and start it again, quite often the IP address, domain name, hostname, and MAC (media access control) address of the VM instance will be different from the previous ones. Therefore, this category of VM migration incurred security problems would be pervasive and should be comprehensively investigated.

3.1 Methodology

We perform an empirical study of the VM migration incurred security problems using the very popular Amazon EC2 (Elastic Compute Cloud) cloud computing environment [13]. We deploy Linux EC2 VM instances, use the default Linux firewall, Netfilter [14], as the IPS, and use the very popular Snort system [15] as the IDS. We define typical IPS and IDS security rules to secure our EC2 instances, migrate the VM instances by simply stopping and restarting them, measure the effectiveness of the IPS and IDS security rules before and after the VM migration, and analyze the measured results.

3.2 Amazon EC2 Service

Each Amazon EC2 instance is created based on an AMI (Amazon Machine Image) [23] that contains the operating system, applications, libraries, data, and configuration settings. Each EC2 instance is a virtual machine that can be stopped and restarted while retaining all the data. Typically, EC2 instances are directly connected to the Internet and are protected by their associated security groups, which define external firewalls with the rules enforced by the Amazon EC2 infrastructure [24]. To take the defense-in-depth approach and provide multiple layers of security controls, customers should further deploy host-level IPSs and IDSs directly on their EC2 instances [25].

When an Amazon EC2 instance is started, it is given a public (i.e., external) IP address, a public domain name, a private (i.e., internal) IP address (also the hostname in the form of the private IP address), and a private domain name. Within the Amazon EC2 cloud computing environment, both domain names are resolved to the private IP address; outside the Amazon EC2 cloud computing environment, the public domain name is mapped to the public IP address. Whenever an EC2 instance is stopped or terminated, these IP addresses and domain names are released back to the IP address and domain name pools, and new IP addresses and domain names will be assigned to the instance when it is restarted. A static Elastic IP (EIP) address can be allocated for a user's AWS (Amazon Web Services) account with extra charge, and can be associated with a running instance; however, stopping a classic EC2 instance also disassociates the EIP from it [26].

3.3 Experiments and Analysis

In our experiments, we follow the typical way of using the IP addresses as the identifiers to define the Netfilter [14] IPS rules and the Snort [15] IDS rules. However, domain name, hostname, and MAC address can also be used to define the Netfilter and Snort rules, so the similar security problems may also happen for those rules (see Section 3.3.3).

3.3.1 The Netfilter IPS (Firewall)

We configure the Netfilter firewall rules using the Linux *iptables* command line program [14]. Table 1 lists five experimental scenarios and the corresponding Netfilter firewall rules that we defined.

Netfilter Scenario 1: Internet Control Message Protocol (ICMP) is a core Internet protocol that transmits the network status and control information. For example, ICMP is used in the *ping* network administration utility. In this scenario, the Netfilter rule is deployed on the EC2 instance A to restrict incoming ICMP echo requests sent to the instance A by dropping the requests that are not sent from the instance B. This firewall rule protects the instance A from unsolicited ICMP packets that could be used in attacks such as OS finger printing, flooding, and denial of service (DoS) [27]. The default firewall rule in this scenario is accepting all the incoming traffic.

In our experiments, we first verified that this firewall rule works correctly for the original instances A and B. We then experimented with two different migration cases. In the first case, we stopped the instance A and restarted it. Because the IP address of the instance A is changed and the instance is migrated, this rule will no longer drop any ICMP echo request sent to the instance A. Therefore, the instance A becomes immediately vulnerable to many ICMP attacks such as OS finger printing, flooding, and DoS [27]. Oversized ICMP packets can also potentially crash or reboot the target system [27].

In the second case, we stopped the instance B and restarted it. Because the IP address of the instance B is changed and the instance is migrated, this rule will drop any ICMP echo request sent from the instance B to the instance A, thus denying the legitimate requests from the instance B and disrupting the normal interactions between the two instances.

Netfilter Scenario 2: User Datagram Protocol (UDP) is a core Internet protocol for hosts to efficiently send messages to other hosts on the IP network without establishing dedicated end to end connections. For example, UDP is widely used in the Internet domain name system (DNS). In this scenario, the Netfilter rule is deployed on the EC2 instance A to allow outgoing UDP requests sent from the instance A to some domain name server. The default firewall rule in this scenario is dropping all the outgoing traffic.

In our experiments, we first verified that this firewall rule works correctly for the original instance A. We then stopped the instance A and restarted it. Because the IP address of the instance A is changed and the instance is migrated, this rule will no longer accept the domain name resolution request sent from the instance A to the domain name server. Therefore, the IP addresses of the remote servers cannot be returned to the applications on the instance A, and many legitimate activities will be disrupted.

Netfilter Scenario 3: Transmission Control Protocol (TCP) is a connection oriented core Internet protocol that is widely used in different application layer protocols such as HTTP and SSH (Secure Shell). In this scenario, the Netfilter rule is deployed on the EC2 instance A to allow outgoing HTTP requests sent from the instance A. The default firewall rule in this scenario is also dropping all the outgoing traffic.

In our experiments, we first verified that this firewall rule works correctly for the original instance A. We then stopped the instance A and restarted it. Because the IP address of the instance A is changed and the instance is migrated, this rule will no longer accept any HTTP request sent from the instance A to remote Web services. Therefore, legitimate Web requests and interactions initiated from the instance A are disrupted.

Netfilter Scenario 4: In this scenario, the Netfilter rule is deployed on the EC2 instance A to restrict the incoming SSH connection requests sent to the instance A by dropping the requests that are not sent from the instance B. This firewall rule protects the instance A from remote login requests sent from any unauthorized host. The default firewall rule in this scenario is accepting all the incoming traffic.

In our experiments, we first verified that this firewall rule works correctly for the original instances A and B. We then experimented with two different migration cases. In the first case, we stopped the instance A and restarted it. Because the IP address of the instance A is changed and the instance is migrated, this rule will no longer drop any SSH request sent to the instance A. Therefore, the instance A becomes immediately vulnerable to remote login attacks such as online password dictionary and brute-force attacks.

In the second case, we stopped the instance B and restarted it. Because the IP address of the instance B is changed and the instance is migrated, this rule will drop any SSH request sent from the instance B to the instance A, thus denying the legitimate SSH connection requests from the instance B and disrupting the normal user activities.

Netfilter Scenario 5: In this scenario, the Netfilter rule is deployed on the EC2 instance A to accept the outgoing SMTP (Simple Mail Transfer Protocol) connection requests sent from the instance A. This firewall rule allows the instance A to transmit standard electronic mail messages across networks. The default firewall rule in this scenario is dropping all the outgoing traffic.

In our experiments, we first verified that this firewall rule works correctly for the original instances A. We then stopped the instance A and restarted it. Because the IP address of the instance A is changed and the instance is migrated, this rule will no longer accept any outgoing SMTP mail requests sent from the instance A. Therefore, for example, the instance A will not be able to communicate with the administrators for alert emails, and legitimate activities are disrupted.

3.3.2 The Snort IDS

Snort [15] can match the specified patterns in message headers and bodies to detect suspicious intrusion attempts. We installed the Snort IDS on our EC2 instances. Table 2 lists five experimental scenarios and the corresponding Snort IDS rules that we defined.

Snort Scenario 1: In this scenario, the Snort rule is deployed on the EC2 instance A to alert and block suspicious outgoing HTTP requests that are not sent from the instance A. This snort rule protects the instance A from being used

Table 1: Experimental Scenarios and Security Rules for the Netfilter IPS

Scenarios	**Netfilter Rules**
1. Restrict incoming ICMP echo requests sent to the instance A by dropping the requests that are not sent from the instance B	iptables -A INPUT -p icmp -icmp-type 8 -s !<IP-Address-of-instanceB> -d <IP-Address-of-instanceA> -m state –state NEW,ESTABLISHED,RELATED -j DROP
2. Allow outgoing UDP (domain name resolution in this case) requests sent from the instance A to some domain name server	iptables -A OUTPUT -p udp -s <IP-Address-of-instanceA> -d <Name Server> -dport 53 -m state –state ESTABLISHED,RELATED -j ACCEPT
3. Allow outgoing TCP (HTTP in this case) requests sent from the instance A	iptables -A OUTPUT -p tcp -s <IP-Address-of-instanceA> –dport 80 -m state –state NEW,ESTABLISHED -j ACCEPT
4. Restrict the incoming TCP (SSH connection in this case) requests sent to the instance A by dropping the requests that are not sent from the instance B	iptables -A INPUT -p tcp -d <IP-Address-of-instanceA> -s !<IP-Address-of-instanceB> –dport 22 -j DROP
5. Allow outgoing TCP (SMTP in this case) requests sent from the instance A	iptables -A OUTPUT -p tcp -s <IP-Address-of-instanceA> –dport 25 -j ACCEPT

Table 2: Experimental Scenarios and Security Rules for the Snort IDS

Scenarios	**Snort Rules**
1. Alert and block suspicious outgoing TCP (HTTP in this case) requests that are not sent from the instance A	alert tcp !<IP-Address-of-instanceA> any -> 80 (msg: Not from legitimate host, Connection blocked; sid:400;react:block;)
2. Alert ICMP messages sent from the instance B to the instance A	alert icmp <IP-Address-of-instanceB> any -> <IP-Address-of-instanceA> any (msg: "Server is Up and running"; sid: 121;)
3. Alert TCP (anonymous FTP in this case) requests sent from any host that is not instance B to the instance A	alert tcp !<IP-Address-of-instanceB> any -> <IP-Address-of-instanceA> 21 (msg: attempted anonymous ftp access; content: anonymous; offset: 5;)
4. Alert UDP (domain name resolution) responses sent from any unauthorized host to the instance A	alert udp !<Name-Server> 53 -> < IP-Address-of-instanceA > any (msg:"Non-Legitimate DNS";sid:254;)
5. Alert TCP (SSH connection in this case) requests sent from any host that is not instance B to the instance A	alert tcp !<IP-Address-of-instanceB> any -> <IP-Address-of-instanceA> 22(msg:"Unknown SSH connection detected";)

by malicious applications to send Web requests with spoofed source IP addresses and perform application layer DoS attacks.

In our experiments, we first verified that this Snort rule works correctly for the original instance A. We then stopped the instance A and restarted it. Because the IP address of the instance A is changed and the instance is migrated, this rule will alert and block all the legitimate HTTP requests sent from the instance A, thus disrupting the normal user and application activities.

Snort Scenario 2: In this scenario, the Snort rule is deployed on the EC2 instance A to alert ICMP messages sent from the instance B to the instance A. For one example, when an ICMP echo request is sent (e.g., using the *ping* utility) from the instance A to check the heartbeat of the instance B, an ICMP response message should be sent back to the instance A if the instance B is alive. Similarly, the instance B can also send ICMP echo requests to the instance A to check the heartbeat of the instance A. So the alert messages can be used for the heartbeat checking.

In our experiments, we first verified that this Snort rule works correctly for the original instances A and B. We then experimented with two different migration cases. In the first case, we stopped the instance A and restarted it. Because the IP address of the instance A is changed and the instance is migrated, this rule will no longer alert any ICMP message sent from the instance B. In the second case, we stopped the instance B and restarted it. Similarly, because the IP address of the instance B is changed and the instance is migrated, this rule will no longer alert any ICMP message sent from the instance B. Therefore, corresponding to the aforementioned example, the heartbeat checking function is disrupted and the two instances may not continue to perform their normal interactions with each other.

Snort Scenario 3: In this scenario, the Snort rule is deployed on the EC2 instance A to alert anonymous FTP requests sent from any host that is not instance B to the instance A. It will not only check the source and destination of the requests, but also check the content of the messages to match the username "anonymous".

In our experiments, we first verified that this Snort rule works correctly for the original instances A and B. We then experimented with two different migration cases. In the first case, we stopped the instance A and restarted it. Because the IP address of the instance A is changed and the instance is migrated, this rule will no longer alert any anonymous FTP request sent to the instance A, thus failing to detect suspicious FTP connection requests originated from any unauthorized host.

In the second case, we stopped the instance B and restarted it. Because the IP address of the instance B is changed and the instance is migrated, this rule will alert any legitimate anonymous FTP request sent from the instance B to the instance A. Therefore, many false alarms will be generated and the normal FTP interactions between the two instances may even be disrupted.

Snort Scenario 4: In this scenario, the Snort rule is deployed on the EC2 instance A to alert domain name resolution responses sent from any unauthorized host to the instance A. This rule can be used to capture anomalous DNS activities and protect the instance A from using the DNS resolution results generated by malicious hosts.

In our experiments, we first verified that this Snort rule works correctly for the original instance A. We then stopped the instance A and restarted it. Because the IP address of the instance A is changed and the instance is migrated, this rule will no longer alert any unauthorized DNS response sent to the instance A, thus providing the opportunities for attackers to use malicious DNS responses to manipulate the instance A and its applications.

Snort Scenario 5: In this scenario, the Snort rule is deployed on the EC2 instance A to alert SSH requests sent from any host that is not instance B to the instance A. It can be used to detect the suspicious remote login attempts from any unauthorized host.

In our experiments, we first verified that this Snort rule works correctly for the original instances A and B. We then experimented with two different migration cases. In the first case, we stopped the instance A and restarted it. Because the IP address of the instance A is changed and the instance is migrated, this rule will no longer alert any SSH request sent to the instance A, thus failing to detect suspicious SSH connections initiated from the unauthorized hosts.

In the second case, we stopped the instance B and restarted it. Because the IP address of the instance B is changed and the instance is migrated, this rule will alert any legitimate SSH connection request sent from the instance B to the instance A. Therefore, many false alarms will be generated and the normal SSH interactions between the two instances may even be disrupted.

3.3.3 Rules with Other Types of Identifiers

In the EC2 environment (see Section 3.2), we verified that when a domain name or a hostname (rather than an IP address) is used to define the Netfilter and Snort rules, the same VM migration incurred security problems as exemplified in the previous scenarios also exist. A MAC address can be used to specify the source of the incoming traffic in Netfilter rules and in the ARP spoof pre-processor for Snort. Therefore, in theory, some VM migration incurred security problems may still happen. However, because an incoming packet always carries the MAC address of the nearest router that delivers the packet, such MAC address based rules are rarely used, and we did not find a typical scenario to perform further verification.

4 Discussion

In the last section we used typical scenarios to exemplify that the migration of VM instances in the cloud can weaken or even nullify the security protections provided by the Netfilter IPS and the Snort IDS to the original VM instances. Note that the similar problems may also occur for other IPSs, IDSs, and application layer protection services as long as they use IP address, domain name, hostname, and MAC address as the identifiers to define their security rules. For example, Apache Web server configuration directives (particularly in the httpd.conf configuration file) can rely on a set of IP addresses or domain names. In a typical scenario of using the <directory> directive to control the website folder or file access based on the client IP address or hostname, we verified that the access control rule will be nullified if the client VM instance is migrated.

To address this type of VM migration incurred security problems, we advocate three potential approaches: adopting the best practices in configuring the security services, leveraging the advanced facilities provided in the cloud computing environments, and performing automatic security problem detection and correction.

4.1 Adopting the Best Practices

As we can see from the examples in the last section, using hard-coded IP addresses (or other types of identifiers) in the security rules is the main reason why those rules were weakened or even nullified after the VM migration. One best practice is to use environment variables that will dynamically change when VM instances are migrated, and avoid using hard-coded identifiers in security rules if possible.

For example, the following Linux shell script can extract the IP address of a host from the network configuration file and then assign the IP address value to the "EXTIP" environment variable. Linux text utilities *grep* and *sed* are used in this script for pattern matching and effective text extraction. Then, when we define the Netfilter and Snort security rules, we can directly use the "EXTIP" environment variable rather than any hard-coded IP address. Therefore, the current IP address of a VM instance will be dynamically updated to the security rules after a migration.

```
# Utilities
IFC='/sbin/ifconfig'
GREP='/bin/grep'
SED='/bin/sed'
#Interface
EXTIF=eth0
# Setting up the environment variables
EXTIP="`$IFC $EXTIF|$GREP addr:
|$SED's/.*addr:\([^ ]*\) .*/\1/'`"
```

4.2 Leveraging the Advanced Facilities

Some cloud computing environments provide specific configuration settings that allow VM instances to be always associated with certain static IP addresses and domain names. For example, Amazon provides the Virtual Private Cloud (VPC) environment [28] for customers to define a virtual network and deploy the VM instances. If we use VPC with regular IP addresses, the VM migration incurred security problems discussed in Section 3 will still occur. However, if we use the combination of VPC and Elastic IP (EIP) addresses, the static EIPs will still be assigned to the original VM instances (deployed in the VPC) after the instances are migrated. Therefore, carefully leveraging this and similar advanced facilities in the cloud computing environments, customers can avoid the VM migration incurred security problems discussed in Section 3.

4.3 Automatic Detection and Correction

The previous two approaches basically put the responsibilities on the customers who will deploy their systems and applications in the cloud. However, we cannot assume that customers will always follow the best practices or will always properly leverage the advanced facilities to prevent the potential VM migration incurred security problems. This is analogous to the situation that widespread security vulnerabilities in the software systems [29, 30, 31, 32, 33] can often be attributed to the failure of engineers to follow secure design, implementation, and deployment practices.

Therefore, we advocate the third approach, which is to perform automatic security problem detection and correction. We are currently developing such tools that can be used by VM instances to automatically (a) identify Netfilter and Snort security rules that contain hard-coded or fixed identifiers such as the IP addresses, (b) detect the occurrences of the VM migrations, and (c) update the new identifiers to the corresponding rules. Such tools will be integrated into the VM instances, migrated with the VM instances, and automatically executed (e.g., as the *cron* jobs in the Linux or Unix systems). With the help of such tools, customers will be freed from the burden, worry, and potential mistakes of addressing VM incurred security problems by themselves.

5 Conclusion

In this paper, we analyzed a new category of VM migration incurred security problems. We used typical scenarios to exemplify that the migration of VM instances in the cloud can weaken or even nullify the security protections provided by the intrusion prevention systems (such as the Netfilter Linux firewall) and the intrusion detection systems (such as Snort) to the original VM instances. We highlighted that these problems can cause severe consequences because they immediately provide many vectors for attackers to perform different types of malicious activities. We further analyzed the root cause of this category of VM migration incurred security problems and discussed three potential approaches for addressing them. In the future, we will focus on developing tools that can be used by VM instances to perform automatic security problem detection and correction, so that the VM incurred security problems can be automatically and reliably addressed for cloud computing customers.

6 Acknowledgement

The authors sincerely thank anonymous reviewers for their valuable suggestions and comments. This research was supported in part by the U.S. National Science Foundation under grant CNS-1359542.

References

[1] G. Anthes, “Security in the cloud,” *Commun. ACM*, vol. 53, no. 11, pp. 16–18, 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,” *Commun. ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[3] Y. Chen, V. Paxson, and R. H. Katz, “What’s new about cloud computing security?” EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2010-5, Jan 2010. [Online]. Available: http://www.eecs.berkeley.edu/Pubs/TechRpts/2010/EECS-2010-5.html

[4] T. Ristenpart, E. Tromer, H. Shacham, and S. Savage, “Hey, you, get off of my cloud: Exploring information leakage in third-party compute clouds,” in *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, 2009, pp. 199–212.

[5] H. Takabi, J. B. D. Joshi, and G.-J. Ahn, “Security and privacy challenges in cloud computing environments,” *IEEE Security and Privacy*, vol. 8, no. 6, pp. 24–31, 2010.

[6] J. Ahn, C. Kim, J. Han, Y.-R. Choi, and J. Huh, “Dynamic virtual machine scheduling in clouds for architectural shared resources,” in *Proceedings of the USENIX Workshop on Hot Topics in Cloud Computing (HotCloud)*, 2012.

[7] N. Bila, E. de Lara, M. Hiltunen, K. Joshi, and H. A. Lagar-Cavilla, “The case for energy-oriented partial desktop migration,” in *Proceedings of the USENIX Workshop on Hot Topics in Cloud Computing (HotCloud)*, 2010.

[8] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield, “Live migration of virtual machines,” in *Proceedings of the Symposium on Networked Systems Design & Implementation (NSDI)*, 2005, pp. 273–286.

[9] U. Deshpande, X. Wang, and K. Gopalan, "Live gang migration of virtual machines," in *Proceedings of the International Symposium on High Performance Distributed Computing (HPDC)*, 2011, pp. 135–146.

[10] S.-H. Lim, J.-S. Huh, Y. Kim, and C. R. Das., "Migration, assignment, and scheduling of jobs in virtualized environment," in *Proceedings of the USENIX Workshop on Hot Topics in Cloud Computing (HotCloud)*, 2011.

[11] F. Travostino, "Seamless live migration of virtual machines over the man/wan," in *Proceedings of the ACM/IEEE Conference on Supercomputing (SC)*, 2006.

[12] T. Wood, K. K. Ramakrishnan, P. Shenoy, and J. van der Merwe, "Cloudnet: Dynamic pooling of cloud resources by live wan migration of virtual machines," in *Proceedings of the ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments (VEE)*, 2011, pp. 121–132.

[13] "Amazon Elastic Compute Cloud (Amazon EC2)," http://aws.amazon.com/ec2/.

[14] "The netfilter.org project," http://www.netfilter.org/.

[15] M. Roesch, "Snort - lightweight intrusion detection for networks," in *Proceedings of the USENIX Large Installation System Administration (LISA) Conference*, 1999, pp. 229–238.

[16] J. Oberheide, E. Cooke, and F. Jahanian, "Empirical exploitation of live virtual machine migration," in *Proceedings of the BlackHat DC convention*, 2008.

[17] N. Santos, K. P. Gummadi, and R. Rodrigues, "Towards trusted cloud computing," in *Proceedings of the USENIX Workshop on Hot Topics in Cloud Computing (HotCloud)*, 2009.

[18] A. Duncan, S. Creese, M. Goldsmith, and J. S. Quinton, "Cloud computing: Insider attacks on virtual machines during migration," in *Proceedings of the IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2013, pp. 493–500.

[19] B. Danev, R. J. Masti, G. O. Karame, and S. Capkun, "Enabling secure vm-vtpm migration in private clouds," in *Proceedings of the Annual Computer Security Applications Conference (ACSAC)*, 2011, pp. 187–196.

[20] "Trusted Platform Module (TPM)," http://www.trustedcomputinggroup.org/developers/trusted_platform_module/.

[21] M. Aslam, C. Gehrmann, and M. Björkman, "Security and trust preserving vm migrations in public clouds," in *Proceedings of the IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2012, pp. 869–876.

[22] W. Wang, Y. Zhang, B. Lin, X. Wu, and K. Miao, "Secured and reliable vm migration in personal cloud," in *Proceedings of the International Conference on Computer Engineering and Technology (ICCET)*, 2010, pp. 705–709.

[23] "Amazon Machine Images (AMI)," http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AMIs.html.

[24] J. Varia, "Architecting for the cloud: Best practices," *Amazon Web Services*, 2010.

[25] C. Yue, W. Zhu, G. Williams, and E. Chow, "Using Amazon EC2 in Computer and Network Security Lab Exercises: Design, Results, and Analysis," in *Proceedings of the American Society for Engineering Education (ASEE) Annual Conference*, 2012.

[26] "Elastic IP Addresses (EIP)," http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/elastic-ip-addresses-eip.html.

[27] "ICMP Attacks Illustrated," https://www.sans.org/reading-room/whitepapers/threats/icmp-attacks-illustrated-477.

[28] "Amazon Virtual Private Cloud (Amazon VPC)," http://aws.amazon.com/vpc/.

[29] W. Du, K. Jayaraman, X. Tan, T. Luo, and S. Chapin, "Position paper: Why are there so many vulnerabilities in web applications?" in *Proceedings of the New Security Paradigms Workshop (NSPW)*, 2011.

[30] N. Provos, M. A. Rajab, and P. Mavrommatis, "Cybercrime 2.0: when the cloud turns dark," *Commun. ACM*, vol. 52, no. 4, pp. 42–47, 2009.

[31] "Security Response Publications — Symantec," http://www.symantec.com/security_response/publications/.

[32] C. Yue and H. Wang, "A measurement study of insecure javascript practices on the web," *ACM Transactions on the Web (TWEB)*, vol. 7, no. 2, pp. 7:1–7:39, 2013.

[33] R. Zhao, C. Yue, and K. Sun, "Vulnerability and risk analysis of two commercial browser and cloud based password managers," *ASE Science Journal*, vol. 1, no. 4, pp. 1–15, 2013.