HOW SECURE IS YOUR CLOUD: CLASSIFICATION OF SECURITY THREATS AND COUNTERMEASURES WITHIN CLOUD COMPUTING?

Sultan H. Almotiri[1], Mohammed A. Al Ghamdi[1], Atif Saeed[2], Muhammad Shahid[2], Khalid Masood[3], Arfan Ali Nagra[3], Muhammad Asif[3]
shmotiri@uqu.edu.sa, maeghamdi@uqu.edu.sa, asaeed@cuilahore.edu.pk, msbhatti@cuilahore.edu.pk, khalid.masood@lgu.edu.pk, arfanalinagra@lgu.edu.pk, drmuhammadasif@lgu.edu.pk

[1]Computer Science Department, Umm Al-Qura University, Makkah City, Saudi Arabia.
[2]Department of Computer Science, Comsats University Islamabad, Lahore Campus, Pakistan.
[3]Department of Computer Science, Lahore Garrison University, Lahore, Pakistan.

***Abstract*—Cloud computing is a contemporary cost-effective model in which the computing resources are dynamically scaled-up and scaled-down to customers, hosted within large- scale multi-tenant systems. These motivations can attract organizations to shift their sensitive data and critical infrastructure on cloud environments. But the cloud environments are facing a large number of challenges of misconfigurations, cyber-attacks, rootkits, malware instances etc. which manifest themselves as a serious threat to cloud environments. These threats noticeably decline the general trustworthiness, reliability and accessibility of the cloud. Security is the primary concern of a cloud service model. However, a number of significant challenges revealed that cloud environments are not as much secure as one could expect. cloud providers have implemented different security perimeters to mitigate these attacks, but these strategies are not impenetrable. A myriad of previous studies has demonstrated how cloud environment could be vulnerable to attacks through shared file systems, cache side-channels, or through compromising of hypervisor layer using rootkits. Thus, the threat of attacks is still possible because an attacker uses one VM to control or access other VMs on the same hypervisor. This paper presents the classification of security attacks across different cloud services. It also indicates attack types and risk levels associated with different cloud services. The attacks get more severe for lower layers where infrastructure and platform are involved. The intensity of these risk levels is also associated with security requirements of data encryption, multi-tenancy, data privacy, authentication and authorization for different cloud services.**

Key Words: Cloud Computing, Cross-VM attacks, Countermeasures, Virtual Machine, Virtualization.

1. Introduction

This research investigates the potential security threats and their countermeasures revealed within cloud systems. The main goal of this study is twofold: to review the existing threat model and to assess the state-of-the-art attacks in cross-VM settings along with their countermeasures. To place this work in context, this paper also offers a broader perspective on security issues in virtualization and their countermeasures.

Cloud computing security denotes to a broad set of strategies, technologies, and controls organized to protect data, applications, and the associated infrastructure of cloud computing [1]. It is the main concern of enterprises when shifting its critical information to geographically distributed cloud platforms and these platforms are directly not under the control of that organization. Additionally, traditional IT information system security procedures, security configurations, firewall rules can help in reducing the cloud attack surface. The main security principles that protect information assurance are confidentiality, integrity, availability, authentication, authorization, auditing, and accountability.

2. Attack Vectors in Cloud Computing

As discussed, cloud computing provides new characteristics to facilitate enterprises and individuals to deploy IT infrastructure. However, these new characteristics in turn strengthen already existing vulnerabilities and introduce new vulnerabilities. The Cloud Security Alliance (CSA) has recognized several well-known attacks that can compromise customers' computations and data in clouds [2]. These attacks have been categorized into different classification based on the attack vectors. Figure 1 illustrates the abstract architecture of a cloud system with the possible attack vectors. Here "x" indicates the possible attack vectors which are discussed as follows:

21. Potential Attack Vectors

- **Service interface:** To access the services in cloud computing, a customer first requires registering an account on the official website of the cloud provider. The customers need to login into their account to use the cloud services. At this point, an attacker can penetrate itself in a cloud system. As some cloud systems have not strong user identification and authentication process, this gives a strong possibility to an attacker to hijack customers' account and access or compromise the sensitive or private areas of cloud services. Secondly, customers interact with the cloud services by using some User Interfaces (UI) or Application Program Interfaces (APIs). If these interfaces are not configured properly or securely, an attacker can easily exploit the vulnerabilities to hack credentials or compromise the cloud services [3].
- **Networks:** Cloud computing allows customers and end-users to access its services through networks. If networks are not properly secured, attackers can steal sensitive data during transmission. Addition- ally, attackers can launch different attacks on net- works e.g., Denial-of-Service (DoS) that fully utilize the cloud system resources. As a result, such attacks prevent customers or end-users from accessing the data or applications [3].
- **Cloud Administrators:** The cloud administrators have been granted an admin role, a privileged role, that supports management of the cloud infrastructure and services. Non reliable or compromised cloud administrators can be the main source of high security risks to the customers in the following ways. First, an unfortunate mishap or accident can occur in a cloud system that leads to permanent data loss or leak of customers' sensitive data. Such mishap or accidents compromise of accidental data deletion by cloud administrators, or a physical disaster such as a fire in cloud data center or an earthquake. Second, an insider regardless has full access of the data stored in the cloud. He can easily misuse or exploit the sensitive information of customers stored in the clouds or compromise the key management of cloud [3].

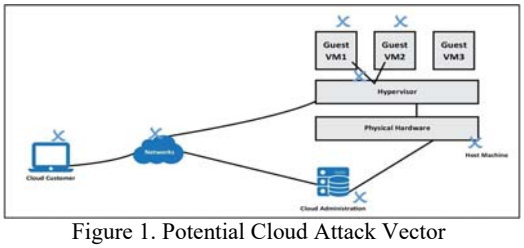

Figure 1. Potential Cloud Attack Vector

3. Vulnerabilities in Cloud Computing

Traditional security threats such as the vulnerabilities in networks and the related operating system attacks encountered in local networks and systems, are also applicable to the domain of cloud computing. Cloud providers introduce

new, advanced features in cloud computing, which can in turn lead to new security threats to both cloud providers and consumers. Firstly, consumers should trust cloud providers when using computational resources and storage devices. Security perimeters, resource allocations and the management of cloud services are not handled or managed by consumers, in fact these are strictly controlled and monitored by cloud providers. Consumers are highly dependent upon cloud providers for securing their sensitive data and computations. Secondly, cloud providers usually facilitate multi-tenancy infrastructure, which helps reduce its cost and maximize resources usage. Due to this feature, multiple VMs belonging to different untrusted consumers can be located on the same physical machine. This facility can effectively enhance whole system's resource utilization and can likewise reduce operational costs. However, this feature can also bring new vulnerabilities to a cloud system, due to sharing of the same hardware resources and storage devices. Thirdly, the cloud providers use virtualization technique to manage resources more efficiently. This extra software layer can make systems more complicated and can add new attack vectors. The aim of this research is to analyze the existing cloud vulnerabilities, and also to discuss countermeasures proposed in prior work for mitigating these cloud-based vulnerabilities. In general, the classification of these vulnerabilities and countermeasures are based on attack vectors.

3.1. Vulnerabilities in Virtualization

The infrastructure of cloud computing runs through the concept of virtualization, in which a single physical system is assigned to multiple users at the same time. In such situations, there are possibilities of exfiltration of data [4]. The introduction of a new layer in virtualization may create a single point of entry for attackers, if virtual machine monitor is compromised. Three types of vulnerabilities on virtualization can exist which are as follows:

- **OS level virtualization** Multiple guest OSs run on a host OS, that has the control and visibility of each guest OS. Within this type of configuration, by compromising the host OS, an attacker can obtain control of the entire guest operating systems running on the host OS [5].
- **Application-based virtualization** is layered above host OS. In this type of virtualization, each VM has its guest OS that is running different applications. Application-based virtualization also suffers from the same type of vulnerability as OS-based vulnerabilities [5].
- **Hypervisor or Virtual Machine Monitor (VMM)** Hypervisor is a piece of code embedded in host OS. Such a code may contain native errors. This code runs at a boot time of the host OS, to control multiple guest OSs. If the attacker is successful in compromising the hypervisor, then the entire controlled guest OSs can be compromised [5]. Well-known attacks on hypervisor are given below.

In [5], [6] VM escape techniques were defined as being where an attacker creates a program that executes a VM, whose purpose is to access the hypervisors' root privileges by breaking the isolation layer. Using VM escape, an attacker can access the host OS, bypassing the hypervisor layer and other VMs running on the same physical machine. Virtual machine sprawl is another challenge for cloud organizations. With virtual machine sprawl, the number of virtual machines running within a virtualized environment increases due to the creation of new virtual machines that are not necessary for business. Due to this, the new virtual machine will misuse the cloud infrastructure [7].

Virtual machine can run on cloud computing which can be accessed through the Internet. This indicates that their theft can take place remotely. Most hypervisors can store contents of the virtual disk for each VM as a file, which allows VMs to be copied and run from other physical machines. While this is a convenient feature, it is also a security threat. Attackers can copy the VM over the network, or to a portable storage media, and then access data on their own machine without physically stealing a hard drive [8]. Once attackers have direct access to the virtual disk, they then have an unlimited time to defeat all security mechanisms, such as passwords, by using offline attacks. The second security breach of the virtual disks discussed in [8], is how attackers could corrupt or externally-modify a file while the VM is offline. This means the integrity of an offline VM may be compromised if the host is not securely protected [9].

The hypervisor manages the resource allocation between the host and guests' machines. The ultimate goal of the at- tacker is to compromise the hypervisor, in order to access the host OS with the same privileges as that of the hypervisor [10].

3.2 Rootkit

Rootkit is a software or application-level tool that enables an unauthorized user to gain control of a computer system without being detected. In Virtualization, cross virtual machines can penetrate rootkits to other virtual machines, can crash hardware, or even can access sensitive information. Rootkits have multiple capabilities, they are able to not only hide malware, but can also conceal malware from analysis and detection processes utilized by defenders [11].

3.2.1 Rootkit in Hypervisor. The new guest OS in virtualization assumes it is running at the host OS, with the corresponding control over hardware and resource. However, in reality there is no concept of the host's existence. A compromised hypervisor can also be used to create a covert channel for executing unauthorized code into the system. Through this approach, an attacker is able to control any VM running on the host machine and can consequently manipulate system activities [5].

3.2.2 Hijacking the Hypervisor. If the rootkit can insert itself beneath the guest operating systems, it can control the entire system [12]. This is exactly what rootkits achieve through different modes of x86 modern architecture, as explained below. Figure 2 shows the x86 modern architecture, along with root-kit privileges level.

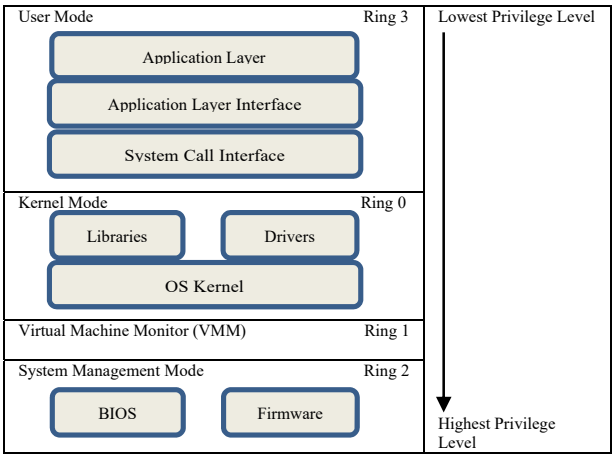

Figure 2. Privileges Level of Rootkit

- **The User-Mode Rootkit** resides in Ring 3, along with some other applications as user, rather than low- level system processes. These can help in achieving objectives by replacing a system's binary applications, or by over-writing a Dynamically Linked Library (DLL) [11]. **DLL Hooking and Injection** User-mode rootkits can exploit an API hooking by using DLL hooks. A well-known user mode rootkit is the Vanquish rootkit [13], which redirects Windows API calls to hide files, folders and registry entries [14]. This is accomplished by injecting a malicious DLL into a target process, acting as an intermediary for API calls to intercept requests for files, folders or registry entries, to filter them [11], [15].

- **Kernel-Mode Rootkit** These rootkits reside in Ring 0 which has the highest operating system privileges level by adding some code or modifying the portions of the core operating system, including both the kernel and associated device drivers.
- **System Management Mode Rootkits** These rootkits (SMMR) normally reside in Ring2. The purpose of their development is to support low-level strategy that is developed through BIOS and Firmware Rootkits.

4. Related Work

4.1. Cross-Channel Attack in Cloud:

The aforementioned attacks are general threats to a cloud model. Following are well-known Cross- Channel attacks in a cloud model that are close to our research.

4.1.1 Side Channel Attacks. Side-channel attacks are a class of physical attacks in which an adversary tries to exfiltrate the sensitive information of other virtual machines.
The use of virtualization can introduce new security vulnerabilities, such as using cross VM-Side channel attacks, to extract information from a target machine [16]. Some of the most well-known side channel attacks have been discussed as follows.
The most effective cross VM-attack is an access-driven attack that exploits shared micro-architectural modules like caches. In it, the attacker executes a program of code on the system, which performs cryptographic operations. This program executed by attacker monitors the use of cache to learn information about the key [17].
A time-driven side channel attack is possible when the total execution times of the cryptographic operations with a fixed key, is influenced by the value of key. The influence is exploited by an attacker who can calculate such timing, in order to statistically gather information about the key [18] [19].
A trace-driven attack is used to capture a profile of the cache activity. This states that the attacker can obtain access to a running profile in which cache activity is monitored, and then process it in order to extract the actual activity from the other profile content [18] [19].

4.1.2 Covert Channel. Covert channels are virtual communication channels between entities, which bypass the rules of communication between them. Within the virtualization context, these give the attackers new opportunities for communication, without being noticed by the VMM (Virtual Machine Monitor) security module [20]. Therefore, the evolution of security threats arises within the context of virtualization, and this need to be eliminated in order to obtain the full advantage of using this technology.
In [21], researchers demonstrated a covert channel between the virtual machines on the Xen hypervisor [22]. This was based on the fact that table that maps the machine address frames to the pseudo physical frames of the virtual machine can be read by any guest VM. Rutkowska [23] implemented a method of TCP/IP steganography called NUSHU, leaking sensitive data from a compromised system through network packets generated from it. Murdoch and Lewis [24] have addressed the various possibilities of covert channels, using TCP/IP header steganography. It has been observed that this may be applicable within the virtualized scenario. Murdoch and Lewis [24] described the header fields that make room for steganography and developed the novel 'Lathra' method for a covert channel, using TCP ISN (an initial sequence number). It also stated that an external warden, being a program or entity, which can watch and analyze data transfer between two systems or programs, cannot distinguish the ISN generated by a machine from a manipulated TCP header. The recovery of encoded message is only possible with the key used for generating ISN. This covert channel can be implemented between VMs and cannot be identified by the VMM [21].
It was hypothesized in [21] that a timing covert channel, which can be used to communicate covertly between VMs and an attacker, can be used to leak information from possible VMs. To send information covertly from a VM, TCP packets need to be sent at different time intervals. If a TCP packet is sent at a specific time interval, the receiver then recovers message as bit 1 else bit 0 [25].
A new network-based covert channel has been identified, using two sockets to transfer data covertly. This can be used by an attacker, in order to leak information from a VM [21].

4.1.3. Attacks on Images. An attacker can attack OS images, in order to obtain sensitive information from other guest OSs, and to crash the OS. Even if VMs are inactive, attackers can still manage to access them. This is because the backend data center is permanently activated, and an offline VM is not considered to be the powered-off home computer. Secondly, pre-built images need to be carefully scanned, in order to circumvent legacy-vulnerable applications or trapdoors. As an example, AWS pre-built images store builders SSH keys internally, meaning that all hosts using such types of images are accessible by publishers [6].
As a common practice in cloud computing, cloud providers can create a template image of an Operating System (OS), and then clone it to multiple machines. If there is some vulnerable VM template images, then it may spread over many systems. An attacker can rent one of these VMs and can accordingly analyze all important configurations, including administrative rights. Another key issue they have raised is that an image can even be taken from an untrustworthy source, which may provide back-door access to an attacker [26].
In [27], it was stated that the Guest VM can crash the Host OS, along with another guest VMs hosted by it. Modi et. al [5] detailed that sharing VM images in the cloud introduces security risks. The main concern of an image's owner is confidentiality, for instance the possibility of unauthorized accesses to the image. An image's user is concerned about safety, such as the potential of a malicious image corrupting or stealing the users' own private data. For example, instances running on Amazon's EC2 platform can be easily compromised through the performance of various attacks, such as signature- wrapping attacks, cross site scripting (XSS) attacks, and DoS attacks. Through such types of attack, attackers can create, modify and delete VM images, and can change administrative passwords and settings that are put into instances with EC2 for S3 access. Images of pre-configured virtual applications and machines may be tempered or misconfigured, before being uploaded [28].

4.1.4. Memory Attacks. An attacker can ex- filtrate data by attacking physical hardware, such as memory, storage units, and others. Different cases have been described in this section, showing how successfully attackers can engage different memory regions.
Once an attacker manages to co-reside on the same physical machine with a target, the next step is to exploit the hardware resources, and extract sensitive data through cross VM-attacks. It exploits hardware by using a technique for encoding information, thereby accessing the latencies of a shared L2 cache [18], [29].
The hostile VM first writes continuous blocks of memory, and then releases those blocks. Later the attacker will release those blocks, and the host VM will overwrite those blocks with its own instruction set. The way the target machine writes to those memory blocks, after the attacker has releasedthem, is an operational characteristic performed of the target. The attacking VM then tries to read back the same instruction set, checks for missing blocks of memory, and tries to replicate the possible instruction set [30].
An event channel is just like a signal channel used to inform communication parties about the occurrence of a new event. The writer tells reader about the data it just wrote. Then, once a reader finishes reading it, it deletes the data and tell the writer that a new space is ready for the next input through the event channel. Therefore, the event channel must be well protected, otherwise it will mess up the whole communication. Delivering the wrong information may cause the shared memory channel to go out-of-sync [6].
The grant table provides two types of grants between different VMs. One type is page-flipping, while the other is page-sharing. Per-packet page- flipping has too much performance overhead, due to the high frequency of hypercalls. Therefore, the new communications have dropped page-flipping but have preserved a page-sharing grant. The

Xen memory sharing mechanism is at a page granularity level. Shared pages are identified by an integer, which is known as a grant reference. The hypervisor keeps the grants information, passes the grant reference to the communication VMs and signals it via the event channel. The hypervisor will be the authority for authenticating communication parties. Under certain circumstances, the system may delegate communication parties to manage the grant table by themselves, for performance reasons [6], [31]. Ranjuth [21] has stated that if a VM is migrated to a new host, then the memory used by that VM will be recovered by the VMM. If a new VM is started on that VMM, there is the possibility that the memory used by the old VM will be allocated to the new one. This new VM may be an attacker. Therefore, the attacker can search all the memory pages for some specific information such as passwords, session keys, and other aspects about the first VM. When a VM is destroyed or shut down, the information from the virtual machine can then be leaked. After the destruction or shut down of a virtual ma- chine, its memory can be allocated to a new virtual machine which runs on the same VMM [21].

4.1.5. Row Hammer Attacks. Recent DRAM chips have huge capacity and a high density of memory cells. Therefore, a memory cell can suffer from disturbance errors due to electrical interference from neighboring cells. Especially when an adversary quickly and frequently accesses the DRAM with precise patterns, some data bits in the memory area, where the adversary has no access rights, can be flipped due to electrical interactions. Such DRAM hardware vulnerability is called a row hammer attack.
Xiao et al. [32] abused this memory loophole in order to attack a para-virtualized platform from a guest VM. In this attack, an adversary VM kept accessing selected data in the DRAM, in order to flip critical bits of this VM's page table entry. By doing so, the page table entry points to a fake page table, without being observed and checked by the hypervisor. The fake page table interprets the VM's virtual page to a physical page that does not have a connection with this VM. Consequently, the attacker VM can steal or tamper with the sensitive data of the co-located VMs.

4.1.6 Code Reuse. Code reuse exploits rely on code fragments (gadgets) located at predetermined memory addresses [23, 36, 39]. Code diversification and randomization techniques (colloquially known as fine- grained ASLR [105]) can thwart code-reuse attacks by perturbing executable code at the function [13, 64], basic block [38, 118], or instruction [57, 87] level, so that the exact location of gadgets becomes unpredictable [72].
Snow et al. introduced "just-in-time" ROP (JIT- ROP) [105], a technique for bypassing fine-grained ASLR for applications with embedded scripting sup- port. JIT-ROP is a staged attack: first, the attacker abuses a memory disclosure vulnerability to re- cursively read and disassemble code pages, effectively negating the properties of fine-grained ASLR (i.e., the exact code layout becomes known to the attacker); next, the ROP payload is constructed on- the-fly using gadgets collected during the first step.

4.1.7 Denial of Service. The Denial of Service attack (DoS) is a serious threat. Both the privileged host and the normal guest OS are under threat from this type of attack, due to poor authentication with current communication mechanisms [5], [6], [33]. DoS attacks make other hosts unable to perform actions in a timely way. Hardware sharing can also be exploited in order to conduct host-based DoS attacks. The adversary VM can generate contention regarding different types of shared resources, in order to degrade the victim VM's performance, or to increase its own performance. The most affected resource of focus is the CPU. Grunwald and Ghiasi [34] mentioned that it is possible to flush the shared processor pipeline, but this degrades the victim's performance. They achieved this by implementing de-normalized floating-point values, thereby creating an underflow so that the pipeline has to be flushed to handle the exceptional condition. Zhou et al. [35] exhibited a CPU resource attack, where an attacker VM can abuse the boost mechanism within the Xen credit scheduler, in order to increase its scheduling priority and to acquire more CPU resources than are paid for. Xu et al. [36] applied the DNS lookup method in finding out the victim VM's internal IP address, and consequently confirmed the co-location through two steps. The first step involved pre-filtering unlikely pairs of co-located VMs, by checking the /24 prefix in the internal IP addresses. If two VMs do not share the /24 prefix of the internal IP addresses, then they are not likely to be co-located. The second step is to use the bus locking covert channel to justify co-location, which involves building a bus locking covert channel between each pair of VMs. If two VMs can communicate with each other via this covert channel, then they are located on the same physical machine. Varadarajan et al. [37] also exploited the bus locking covert channels, as a means of evaluating the financial costs of co-location within different public clouds.

4.2. Inter VM-Communication, or Communication Between VMs and Hosts

Isolation is the main feature supported by virtualization. Such a feature, if not configured properly, can result in a potential threat to cloud infrastructure. Virtualization's isolation property ensures that applications executing one VM, do not interfere with the applications of another VM. It should be carefully observed that isolation means that breaking into one virtual machine should not allow for access to its co-located virtual machines within the same environment, and not even to its underlying host machine. In some VM technologies, the VM layer can keep a log of screen updates and keystrokes, through the virtual terminals that successively grant necessary authorization to the kernel of host operating systems. These captured logs are stored in the host, making it a possibility for hosts to observe these logs, even those of encrypted terminal connections inside the VMs. Some virtualization circumvents isolation. The basic idea behind this logic is to support applications considered for one operating system, to be executed on another operating system, without any modification. This solution abuses the security bearers within both of the operating systems. In such systems, where there is lack of isolation between the host and the VMs, they provide the virtual machines unlimited access rights for the utilization of underlying host's resources of file systems and networks.

5. Survey of the State of the Art

This section surveys the state-of-the-art related work in the area of cross-VM attacks, particularly cross- VM network channel attacks for data leakage or to escalate the privilege level of non-root user.

5.1 Vulnerabilities in Network channel

Cloud services are accessed through the network using standard protocols e.g., IP which is considered to be untrustworthy [26]. Internet Protocol (IP) is the method of transferring data from one machine to another. Each machine is assigned an IP address for communication. There are several vulnerabilities within Internet Protocols, which are discussed as follows. The use of same IP addresses by different users may at times lead to accessing different resources of other users [5], [4]. Address Resolution Protocol (ARP) is a protocol used to map an IP address to a corresponding physical machine address [38]. Within cross-VM Address Resolution Protocol (ARP) attacks [39], the attacking VM launches an ARP spoofing attack by forging an identical IP address within the target VM and sends an ARP packet to the virtual router. The virtual router updates the routing table when the spoofed ARP packet is received. As a result, any traffic directed to a target VM is instead sent to the attacking VM, which can then decide to either perform sniffing or modification. In bridge network configuration mode [39], the bridge acts as a virtual hub. All VMs share the virtual hub to communicate with the network. An attacking VM can sniff the virtual network by using a sniffing tool, such as Wireshark [40]. In the router network mode [39], a router plays a role of a virtual switch using a dedicated virtual interface to connect to each VM. Address Resolution Protocol (ARP) Poisoning [41] is also considered to be a well-known vulnerability for Internet protocols [42]. By using this vulnerability,

malicious VM can redirect all the inbound and outbound traffic of a co-located VM to the malicious VM, as ARP does not require Proof-of-Origin [5].

5.2 Return Oriented Programming

There are a number of different real-time attacks that exploit ROP systems. On the application layer, Adobe declared that a critical vulnerability had existed in the Adobe Flash Player 10.0.45.2, and its earlier versions [43]. These vulnerabilities also occur in Adobe Reader and Acrobat 9.x and are executed by applying ROP. As a result, this vulnerability bypass data execution prevention (DEP) [44], which is a security system implemented by Windows, thereby compromising the full system and making it possible for the attacker to take control of the victimized system [45], [46]. For the Windows operating system, the ROP-based rootkits have been used at the kernel layer. Upon execution, these rootkits can manage to hide malicious processes, files and network connections through windows. These rootkits are developed through ROP techniques, consequently circumventing kernel integrity protection systems such as SecVisor [47]. The ROP technique can be used to exploit the Apple iPhone, in which an unauthorized user installs applications, or leaks a customer's SMS database [48]. Table 1 presents already-existing cloud attacks.

5.3 Analysis

This section has surveyed cross-VM attacks to ascertain the most unknown research. [4], [39], [49]. In these papers, researchers have used different approaches such as ARP spoofing, sniffing the virtual network and ROP. The purpose of these attacks is to redirect the network traffic of compromised VM and to describe how an unprivileged VM using ROP technique can manage to modify the code of hypervisor through which it can escalate its privilege level. However, all these attack strategies have some limitations as the security perimeter of cloud computing blocks such attack settings by placing more layer of isolation between VMs or to stop executing arbitrary code. Key lessons learned for the overall approaches are described as- there are number of potential avenues of research channels which are remain unexplored. Researchers have neither suggested any indication for the redirection of network traffic by exploiting the network channel through impersonation and mirroring approach nor show any possibility for the privilege escalation by applying the conjunction of ROP and exploitation of the network channel. This research work focuses on the exploitation of network channel for a variety of following reasons: (i) It arguably has the highest potential to redirect the real time network traffic of a target VM and to escalate the privilege level of unprivileged VM. (ii) Prior work only redirects the network traffic and escalate the privilege level using traditional approaches such as ARP spoofing and 'ret' statement that can be easily countered in virtualization. The proposed approaches provide an in-depth analysis of the techniques involved and their effects, both in qualitative and quantitative terms, in various settings. (iii) The result of this proposed approach is so effective upon success; it can redirect the network traffic at hidden point as well as can control toolstack to manage other VMs. The anatomy of Cross-VM attacks is presented in Figure 3.

6. Countermeasures

6.1 Reducing Side/Covert Channel Leakage

There are several methods cloud providers can use to defend against side-channel cache attacks. The first method to inhibit cache sharing divides it into different regions for different VMs or applications. This is achievable through the use of software or hardware approaches. As for software, some researchers have misused the page coloring approach for cache partitioning [52], [53]. Kim et al. [54] designed the application 'STEALTHMEM', used it to partition the Last Level Cache (LLC). It offers each VM a number of covert pages, which cannot be changed in the cache. Liu et al. [55] abused the Intel Cache Allocation Technology to prevent the victim VM's sensitive data from being exchanged, by the attacking VM in the LLC. New hardware caches have been introduced to partition the caches by ways or sets in order to protect against side-channel attacks due to cache exclusions [56].

The second approach practices randomization in system design, so that attackers do not obtain any useful information. For software, system clock measurements are blurred in order to interrupt the attackers' monitoring [57], [58]. Zhang et al. [59] launched Duppel, which periodically executes cache purging during VM's executions, and adds noise to attacker's observations. For hardware, new caches have been designed to randomize memory-to-cache mappings, and cache fetching [60], [56], [61], [62].

6.2 Optimizing Resources

To alleviate DoS attacks caused by resource contention, the cloud provider can optimize resource usage between different domains and reduce performance interference. For memory contention, the approach divides memory resources between different domains, such as Intel Cache Allocation Technology [63]. For I/O contention, the cloud server can observe and manage the bandwidth of I/O traffic, in order to avoid resource exhaustion. It can also use Direct Device Assignment to physically eliminate I/O intrusion between each VM [64]. These solutions have been widely adopted by public cloud providers. For the power resource, Li et al. [65] proposed PAD as a safe data center for power attacks under the over-subscription setting. PAD creates a virtual battery pool, which activates load sharing and adjusts power utilization for each rack. It can sense and shear power spikes, in order to avoid power consumption failure.

Attack	Description	References
Co-location	Flooder DNS lookup Watermarking	[50] [36] [51]
Virtualization	OS level virtualization Application-based virtualization Hypervisor/VMM Rootkits	[5] [5] [5]
Side	Time-driven Access-driven Trace-driven	[19] [17] [18]
Covert Channel	TCP/IP Steganography TCP/IP Header Steganography Timing Channel	[23] [24] [21]
Images	Image Tempered	[28]
Memory	L2-cache Event Channel Grant Table	[18] [6] [31], [6]
DoS	The illegal use of resources	[33], [5], [6]
Network Channel	ARP Poisoning Sniffing Spoofing	[5], [41] [39] [39]
Row Hammer Attacks	DRAM Hardware Attack	[32]
Return-Oriented Programming (ROP)	ROP based rootkit	[45], [46], [47]

Table 1. Overview of Cloud Attacks.

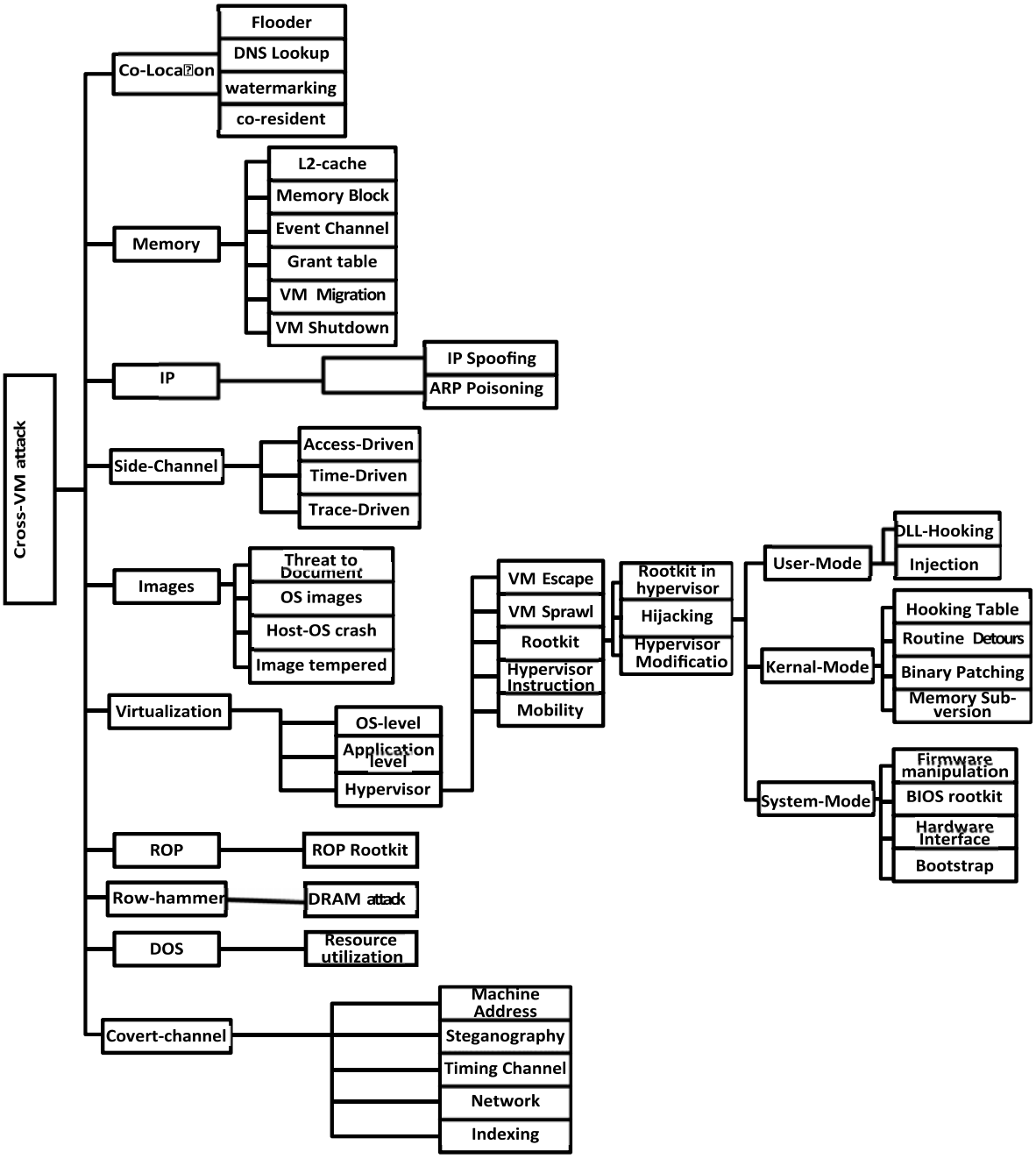

Figure 3. Anatomy of Cross-VM Attacks

6.3 Protection of VM images

It is important to safe VM images in the cloud ap- plication store and eliminate potential vulnerabilities of their publishers and the retrievers. Wei et al. [66] introduced an image management system, Mirage to deal with the security issues of VM images by different methods: (1) Access control: Mirage follows access permission of two types, check-out and check-in, to closely observe who is retrieving the images. (2) Image filters: Mirage uses special filters to delete sensitive information from the original images.

6.4 Network Defenses

Georgiev and Shmatikov [67] introduced 'CAPTCHAs' in the protocols to separate human users from malicious automated scanners.

6.5 Eliminating the Hypervisor's Vulnerabilities

Prior work has considered different methodologies for improving the security of hypervisors. The first approach develops new secure hypervisors. McCune et al. [68] presented TrustVisor, a tiny version of the hypervisor used to ensure the reliability of applications' sensitive data and codes. In this, a new secure guest mode has been introduced for running applications executed on x86 hardware architecture, supporting virtualization in order to impose strong memory isolation between the hypervisor, the host OS, and VM applications. TrustVisor also launched a software micro-TPM instance for each application, in order to perform integrity attestation. Vasudevan et al. [69] developed, configured and certified an open-source eXtensible and Modular Hypervisor Framework (XMHF). XMHF consists of a number of different XMHF main and small associate libraries. A hypervisor application can extend the XMHF core and the basic functionalities provided by this framework in order to execute preferred security features.

The second approach in this direction is to secure the integrity of hypervisors. Wang et al. [70] introduced Hyper Safe, a lightweight method for ensuring the integrity of Type-I hypervisors at runtime. HyperSafe applies the Write Protect (WP) bit to prevent hypervisor pages from being compromised by malicious applications. HyperSafe can also configure a target table consisting of all legitimate destinations for indirect control flow instructions, as a means of implementing the hypervisor program's control flow at runtime. Azab et al. [71] designed HyperSentry as a tool for calculating the integrity of hypervisor at runtime. A remote client who needs to validate the hypervisor can use the Intelligent Platform Management Interface (IPMI) to initiate the server into the System Management Mode (SMM), and to measure the hypervisor's code, data and CPU state. Another direction is to reduce the hypervisors' privileges and functionalities.

NoHype [72], [73] is designed to eradicate the hypervisor during the VM's runtime, thereby reducing the attack surface from the hypervisor. NoHype attains this by pre-allocating processor, cores and memory for each VM during VM creation, allocating virtualized I/O devices directly to VMs, in order to circumvent the need for a hypervisor to do I/O emulation. It also modifies the guest OS to cache host system configuration for later use. Butt et al. [74] proposed self-service cloud computing as a means of confining the host VM's privileges. It splits the administrative privileges between a system-wide VM, and per-client administrative VMs. The per- client administrative VMs are able to implement some privileged system tasks at their own VMs, while the system-wide VM cannot examine the code, data or computation of client VMs. In this case, security and privacy are conserved even if the host VM is compromised.

6.6 Defeating Row Hammer Attacks

Cross-VM row hammer attacks can be defended against through the use of hardware or software solutions. For hardware, Error-Correcting Code (ECC) memory can be used to ensure the correctness of one single-bit error, and to detect 2-bit errors. This makes row hammer attacks much harder [75]. For software, Brasser et al. [76] has suggested two solutions. The first solution is to extend the system bootloader to recognize exploited memory pages. Row hammer exploitation tools are executed offline, determining which memory pages could be tempered by row hammer attacks [77]. Then the boot loader indicates that these exploitable memory pages are inaccessible at boot-time, so that these pages will not be executed at runtime. The second solution is to extend the OS kernel, in order to impose strong isolation onto the physical memory of different system entities, such as user and kernel spaces. This ensures that memory between different entities is physically separated by at least one row, meaning that one entity cannot interfere with the memory of another. Irazoqui et al. [78] designed MASCAT, a static code analysis tool which can scan the application of binaries, and detect possible micro-architectural attacks, such as row hammer attacks. This tool applies the signature-based detection algorithm for searching binary files with implicit characteristics that micro-architectural attacks usually demonstrate in their design. In row hammer attacks, the attacker needs to continuously circumvent the cache, and access a fixed DRAM location. This is used as the signature of row hammer attacks.
The overview of all these attacks and their related countermeasures have been tabulated in Table 2.

6.7 Analysis

This section has surveyed the countermeasures of cross-VM attacks. In these studies, researchers have proposed different countermeasures such as dedicated VMs, assigning user rights and defeating ROP attacks. The purpose of these countermeasures are to block cross-VM, network channel and ROP attacks. However, all these countermeasures have some limitations as dedicated VMs on cloud computing are not appropriate for cloud providers because they save their operational cost by sharing the same hardware among multiple VMs. Similarly, assigning user rights to VMs and defeating ROP attacks can be circumvent by attacking VMs through privilege escalation. The lesson learned from the overall approaches is researchers have proposed several countermeasures, but there is very limited research in proposing the countermeasures of heterogeneous attack strategies.

Attack	Countermeasures	References
Co-location	Dedicated VMs	[79]
	Runtime VM migration	[80], [81]
	VM launch placement	[82]
Virtualization	Designing secure hypervisors	[68]
	NoHype	[72]
	Protecting hypervisor integrity	[71]
Side/Covert Channel	Sharing memory by dividing it into different regions	[52], [53], [54]
Images	Managing VM images	[66]
Memory	Scheduling adjustment to limit interruptions in memory	[83]
DoS	Optimizing Resources	[63], [64]
Network Channel	Client ID verification	[84]
	Assigning user rights	[84]
	CAPTCHA	[67]
Row Hammer Attacks	Error Correction Codes	[75]
	Memory Isolation	[76]
	Signature-based detection algorithm	[78]
Return-Oriented Programming (ROP)	Defeating ROP Attacks	[85]

Table 2. Attack and Their Related Countermeasures

7. Conclusion

This paper has provided a comprehensive study of state-of-the-art concepts of cloud computing, their essential characteristics, cloud computing models, security features, attack vectors and countermeasures. It has also been presented how they can be used to better study and quantify systems security in cloud model. The abstraction of the cloud system model has been presented and the concepts of a cloud system being composed of multiple nodes which interact with each other through different interfaces within the system environment have been introduced. Furthermore, the concept about virtualization has been discussed in detail and also its system support in terms of hardware and software. Cloud computing and their service models are also discussed in detail. The concept and definition of Cloud computing, as well as the key terminology and characteristics have been presented. Furthermore, the components of Cloud computing security including CIA, different services, as well as virtualization, and servers have been discussed in detail.

The concept of potential attack vectors in cloud model has been presented, and the critical need for empirical analysis and modeling of Cloud systems security discussed. A literature review of the current state-of-the-art of analyzing and characterizing Cloud attacks, including co-location attacks, side- channel attacks and network-channel attacks, has been presented and discussed in detail. Finally, cur- rent gaps in the state-of-the-art for these attacks as well as opportunities where security of cloud system model can be enhanced has been highlighted. The countermeasures solution has been presented of already existed attacks which includes optimizing resources, network defenses, reducing side-channel attacks and defense mechanism of row hammer at- tacks.

From the analysis, we further conclude that all of the above work suggests a strong need for further exploration of cross-VM attacks and associated channels. In contrast to these points, there is a need for research to consider a ROP perspective on cross- VM attacks. Hence further research is required to apply ROP and impersonating attacks allowing new scientific insights to be gained through examining results in novel ways.

Funding

This research work is supported by Data and Artificial Intelligence Scientific Chair at Umm Al-Qura University, Makkah City, Saudi Arabia, also it is supported by the PhD thesis [86].

References

[1] J. Somorovsky, M. Heiderich, M. Jensen, J. Schwenk, N. Gruschka, and L. Lo Iacono, "All your clouds are belong to us: security analysis of cloud management interfaces," in *Proceedings of the 3rd ACM workshop on Cloud computing security workshop*, pp. 3–14, ACM, 2011.

[2] T. T. W. Group *et al.*, "The treacherous 12: cloud comput- ing top threats in 2016," *Cloud Security Alliance*, 2016.

[3] R. Buyya, C. S. Yeo, and S. Venugopal, "Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities," in *High Performance Computing and Communications, 2008. HPCC'08. 10th IEEE International Conference on*, pp. 5–13, Ieee, 2008.

[4] V. Nirmala, R. Sivanandhan, and R. S. Lakshmi, "Data confidentiality and integrity verification using user au- thenticator scheme in cloud," in *Green High-Performance Computing (ICGHPC), 2013 IEEE International Confer- ence on*, pp. 1–5, IEEE, 2013.

[5] C. Modi, D. Patel, B. Borisaniya, A. Patel, and M. Rajara- jan, "A survey on security issues and solutions at different layers of cloud computing," *The journal of supercomput- ing*, vol. 63, no. 2, pp. 561–592, 2013.

[6] S. Zhang, "Deep-diving into an easily-overlooked threat: Inter-vm attacks," tech. rep., Technical Report). Manhat- tan, Kansas: Kansas State University, 2012.

[7] F. Sabahi, "Secure virtualization for cloud environment using hypervisor-based technology," *International Journal of Machine Learning and Computing*, vol. 2, no. 1, p. 39, 2012.

[8] T. Garfinkel and M. Rosenblum, "When virtual is harder than real: Security challenges in virtual machine-based computing environments.," in *HotOS*, 2005.

[9] D. Hyde, "A survey on the security of virtual machines," *Dept. of Comp. Science, Washington Univ. in St. Louis, Tech. Rep*, 2009.

[10] T. Ormandy, "An empirical study into the security expo- sure to hosts of hostile virtualized environments," 2007.

[11] J. S. Alexander, T. Dean, and S. Knight, "Spy vs. spy: Counter-intelligence methods for backtracking malicious intrusions," in *Proceedings of the 2011 Conference of the Center for Advanced Studies on Collaborative Research*, pp. 1–14, IBM Corp., 2011.

[12] I. Korkin and I. Nesterov, "Applying memory forensics to rootkit detection," *arXiv preprint arXiv:1506.04129*, 2015.

[13] Y.-M. Wang, D. Beck, B. Vo, R. Roussev, and C. Ver- bowski, "Detecting stealth software with strider ghost- buster," in *Dependable Systems and Networks, 2005. DSN 2005. Proceedings. International Conference on*, pp. 368– 377, IEEE, 2005.

[14] D. Molina, M. Zimmerman, G. Roberts, M. Eaddie, and G. Peterson, "Timely rootkit detection during live response," in *IFIP International Conference on Digital Forensics*, pp. 139–148, Springer, 2008.

[15] B. Blunden, *The Rootkit arsenal: Escape and evasion in the dark corners of the system*. Jones & Bartlett Publishers, 2012.

[16] S. Pearson and A. Benameur, "Privacy, security and trust issues arising from cloud computing," in *2010 IEEE Second International Conference on Cloud Computing Technology and Science*, pp. 693–702, IEEE, 2010.

[17] Y. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Cross-vm side channels and their use to extract private keys," in *Proceedings of the 2012 ACM conference on Computer and communications security*, pp. 305–316, ACM, 2012.

[18] 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 16th ACM conference on Computer and communi- cations security*, pp. 199–212, ACM, 2009.

[19] D. Page, "Defending against cache-based side-channel attacks," *Information Security Technical Report*, vol. 8, no. 1, pp. 30–44, 2003.

[20] Z. Wang and R. B. Lee, "New constructive approach to covert channel modeling and channel capacity estima- tion," in *International Conference on Information Secu- rity*, pp. 498–505, Springer, 2005.

[21] P. Ranjith, C. Priya, and K. Shalini, "On covert channels between virtual machines," *Journal in Computer Virology*, vol. 8, no. 3, pp. 85–97, 2012.

[22] https://xenproject.org/

[23] J. Rutkowska, "The Implementation of Passive Covert Channels in the Linux Kernel," Proc. Chaos Communication Congress, Dec. 2004.

[24] S. J. Murdoch and S. Lewis, "Embedding covert channels into tcp/ip," in *International Workshop on Information Hiding*, pp. 247–261, Springer, 2005.

[25] H.-C. Li, P.-H. Liang, J.-M. Yang, and S.-J. Chen, "Anal- ysis on cloud-based security vulnerability assessment," in *2010 IEEE 7th International Conference on E-Business Engineering*, pp. 490–494, IEEE, 2010.

[26] B. Grobauer, T. Walloschek, and E. Stocker, "Understand- ing cloud computing vulnerabilities," *IEEE Security & Privacy*, vol. 9, no. 2, pp. 50–57, 2011.

[27] H. Hlavacs, T. Treutner, J.-P. Gelas, L. Lefevre, and A.- C. Orgerie, "Energy consumption side-channel attack at virtual machines in a cloud," in *2011 IEEE Ninth Interna- tional Conference on Dependable, Autonomic and Secure Computing*, pp. 605–612, IEEE, 2011.

[28] Modi, C., Patel, D., Borisaniya, B., Patel A., and Rajarajan M. "A survey on security issues and solutions at different layers of Cloud computing". The Journal of Supercomputing, s63, 561–592 (2013).

[29] O. Acıiçmez, B. B. Brumley, and P. Grabher, "Newresults on instruction cache attacks," in *International Work- shop on*

Cryptographic Hardware and Embedded Systems, pp. 110–124, Springer, 2010.

[30] Z. Tari, "Security and privacy in cloud computing.," *IEEE Cloud Computing*, vol. 1, no. 1, pp. 54–57, 2014.

[31] J. Wang, K.-L. Wright, and K. Gopalan, "Xenloop: a transparent high performance inter-vm network loop- back," in *Proceedings of the 17th international symposium on High performance distributed computing*, pp. 109–118, ACM, 2008.

[32] Y. Xiao, X. Zhang, Y. Zhang, and R. Teodorescu, "One-bit flips, one cloud flops: Cross-vm row hammer attacks and privilege escalation," in *25th {USENIX} Security Symposium ({USENIX} Security 16)*, pp. 19–35, 2016.

[33] S. M. Hashemi and M. R. M. Ardakani, "Taxonomy of the security aspects of cloud computing systems-a survey," *networks*, vol. 2, p. 1Virtualization, 2012.

[34] D. Grunwald and S. Ghiasi, "Microarchitectural denial of service: Insuring microarchitectural fairness," in *35th Annual IEEE/ACM International Symposium on Microar- chitecture, 2002 (MICRO-35). Proceedings.*, pp. 409–418, IEEE, 2002.

[35] F. Zhou, M. Goel, P. Desnoyers, and R. Sundaram, "Scheduler vulnerabilities and coordinated attacks in cloud computing," *Journal of Computer Security*, vol. 21, no. 4, pp. 533–559, 2013.

[36] Z. Xu, H. Wang, and Z. Wu, "A measurement study on co-residence threat inside the cloud," in *24th USENIX Security Symposium (USENIX Security 15)*, pp. 929– 944, 2015.

[37] V. Varadarajan, Y. Zhang, T. Ristenpart, and M. Swift, "A placement vulnerability study in multi-tenant public clouds," in *24th {USENIX} Security Symposium ({USENIX} Security 15)*, pp. 913–928, 2015.

[38] Conrad E., Misenar S., and Feldman J. "Chapter 5 - Domain 4: Communication and Network Security (Designing and Protecting Network Security)", "CISSP Study Guide (Third Edition)", Syngress, 219 - 291, 2016.

[39] H. Wu, Y. Ding, C. Winer, and L. Yao, "Network security for virtual machine in cloud computing," in *Computer Sci- ences and Convergence Information Technology (ICCIT), 2010 5th International Conference on*, pp. 18–21, IEEE, 2010.

[40] https://www.wireshark.org/.

[41] H.-C. Li, P.-H. Liang, J.-M. Yang, and S.-J. Chen, "Anal- ysis on cloud-based security vulnerability assessment," in *e-Business Engineering (ICEBE), 2010 IEEE 7th Inter- national Conference on*, pp. 490–494, IEEE, 2010.

[42] http://www.cisco.com/c/en/us/products/ collateral/switches/catalyst-6500-series-switches/white paper c11 603839.html/, 1999. [Online; accessed1999].

[43] R. Mohandas, V. Thomas, and P. Ramagopal, "Malicious media files: Coming to a computer near you,"

[44] J.-s. Kim and J.-s. Moon, "Detecting code reuse attack using rnn," *Journal of Internet Computing and Services*, vol. 19, no. 3, pp. 15–23, 2018.

[45] Z. Xu, H. Wang, Z. Xu, and X. Wang, "Power attack: An increasing threat to data centers.," in *NDSS*, 2014.

[46] 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 CCS'09, pp. 199–212.

[47] A. Seshadri, M. Luk, N. Qu, and A. Perrig, "Secvisor: A tiny hypervisor to provide lifetime kernel code integrity for commodity oses," in *ACM SIGOPS Operating Systems Review*, vol. 41, pp. 335–350, ACM, 2007.

[48] J. Halliday, "Jailbreakme released for apple devices," *Avaliable: http://www. guardian. co. uk/technology/blog/2010/aug/02/jailbreak me-released- apple-devices-legal*, 2010.

[49] B. Ding, Y. Wu, Y. He, S. Tian, B. Guan, and G. Wu, "Return-oriented programming attack on the xen hyper- visor," in *Availability, Reliability and Security (ARES), 2012 Seventh International Conference on*, pp. 479–484, IEEE, 2012.

[50] A. Bates, B. Mood, J. Pletcher, H. Pruse, M. Valafar, and K. Butler, "On detecting co-resident cloud instances us- ing network flow watermarking techniques," *International Journal of Information Security*, vol. 13, no. 2, pp. 171– 189, 2014.

[51] R. Schick and C. Ruland, "Introduction of a new non- repudiation service to protect sensitive private data," *Ad- vances in Information and Communication Technologies*, pp. 71–76, 2012.

[52] H. Raj, R. Nathuji, A. Singh, and P. England, "Resource management for isolation enhanced cloud services," in *Proceedings of the 2009 ACM workshop on Cloud com- puting security*, pp. 77–84, ACM, 2009.

[53] J. Shi, X. Song, H. Chen, and B. Zang, "Limiting cache- based side-channel in multi-tenant cloud using dynamic page coloring," in *2011 IEEE/IFIP 41st International Conference on Dependable Systems and Networks Work- shops (DSN-W)*, pp. 194–199, IEEE, 2011.

[54] T. Kim, M. Peinado, and G. Mainar-Ruiz, "STEALTHMEM: System-level protection against cache-based side channel attacks in the cloud," in *Presented as part of the 21st {USENIX} Security Symposium ({USENIX} Security 12)*, pp. 189–204, 2012.

[55] F. Liu, Q. Ge, Y. Yarom, F. Mckeen, C. Rozas, G. Heiser, and R. B. Lee, "Catalyst: Defeating last-level cache side channel attacks in cloud computing," in *2016 IEEE in- ternational symposium on high performance computer architecture (HPCA)*, pp. 406–418, IEEE, 2016.

[56] Z. Wang and R. B. Lee, "New cache designs for thwart- ing software cache-based side channel attacks," *ACM SIGARCH Computer Architecture News*, vol. 35, no. 2, pp. 494–505, 2007.

[57] B. C. Vattikonda, S. Das, and H. Shacham, "Eliminating fine grained timers in xen," in *Proceedings of the 3rd ACM workshop on Cloud computing security workshop*, pp. 41–46, ACM, 2011.

[58] P. Li, D. Gao, and M. K. Reiter, "Stopwatch: a cloud architecture for timing channel mitigation," *ACM Trans- actions on Information and System Security (TISSEC)*, vol. 17, no. 2, p. 8, 2014.

[59] Y. Zhang and M. K. Reiter, "Düppel: retrofitting com- modity operating systems to mitigate cache side channels in the cloud," in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pp. 827–838, ACM, 2013.

[60] Z. Wang and R. B. Lee, "A novel cache architecture with enhanced performance and security," in *Proceedings of the 41st annual IEEE/ACM International Symposium on Microarchitecture*, pp. 83–93, IEEE Computer Society, 2008.

[61] F. Liu, H. Wu, K. Mai, and R. B. Lee, "Newcache: Secure cache architecture thwarting cache side-channel attacks," *IEEE Micro*, vol. 36, no. 5, pp. 8–16, 2016.

[62] F. Liu and R. B. Lee, "Random fill cache architecture," in *Proceedings of the 47th Annual IEEE/ACM Inter- national Symposium on Microarchitecture*, pp. 203–215, IEEE Computer Society, 2014.

[63] C. Intel, "Improving real-time performance by utilizing cache allocation technology," *Intel Corporation, April*, 2015.

[64] B.-A. Yassour, M. Ben-Yehuda, and O. Wasserman, "Di- rect device assignment for untrusted fully-virtualized vir- tual machines," 2008.

[65] C. Li, Z. Wang, X. Hou, H. Chen, X. Liang, and M. Guo, "Power attack defense: Securing battery-backed data centers," *ACM SIGARCH Computer Architecture News*, vol. 44, no. 3, pp. 493–505, 2016.

[66] J. Wei, X. Zhang, G. Ammons, V. Bala, and P. Ning, "Managing security of virtual machine images in a cloud environment," in *Proceedings of the 2009 ACM workshop on Cloud computing security*, pp. 91–96, ACM, 2009.

[67] M. Georgiev and V. Shmatikov, "Gone in six characters: Short urls considered harmful for cloud services," *arXiv preprint arXiv:1604.02734*, 2016.

[68] J. M. McCune, Y. Li, N. Qu, Z. Zhou, A. Datta, V. Gligor, and A. Perrig, "Trustvisor: Efficient tcb reduction and attestation," in *2010 IEEE Symposium on Security and Privacy*, pp. 143–158, IEEE, 2010.

[69] A. Vasudevan, S. Chaki, L. Jia, J. McCune, J. Newsome, and A. Datta, "Design, implementation and verification of an extensible and modular hypervisor framework," in *2013 IEEE Symposium on Security and Privacy*, pp. 430– 444, IEEE, 2013.

[70] Z. Wang and X. Jiang, "Hypersafe: A lightweight ap- proach to provide lifetime hypervisor control-flow in- tegrity," in *2010 IEEE Symposium*

on Security and Pri- vacy, pp. 380–395, IEEE, 2010.

[71] A. M. Azab, P. Ning, Z. Wang, X. Jiang, X. Zhang, and N. C. Skalsky, "Hypersentry: enabling stealthy in-context measurement of hypervisor integrity," in *Proceedings of the 17th ACM conference on Computer and communica- tions security*, pp. 38–49, ACM, 2010.

[72] E. Keller, J. Szefer, J. Rexford, and R. B. Lee, "Nohype: virtualized cloud infrastructure without the virtualization," in *ACM SIGARCH Computer Architecture News*, vol. 38, pp. 350–361, ACM, 2010.

[73] J. Szefer, E. Keller, R. B. Lee, and J. Rexford, "Eliminat- ing the hypervisor attack surface for a more secure cloud," in *Proceedings of the 18th ACM conference on Computer and communications security*, pp. 401–412, ACM, 2011.

[74] S. Butt, H. A. Lagar-Cavilla, A. Srivastava, and V. Gana- pathy, "Self-service cloud computing," in *Proceedings of the 2012 ACM conference on Computer and communica- tions security*, pp. 253–264, ACM, 2012.

[75] Y. Kim, R. Daly, J. Kim, C. Fallin, J. H. Lee, D. Lee, C. Wilkerson, K. Lai, and O. Mutlu, "Flipping bits in memory without accessing them: An experimental study of dram disturbance errors," in *ACM SIGARCH Computer Architecture News*, vol. 42, pp. 361–372, IEEE Press, 2014.

[76] F. Brasser, L. Davi, D. Gens, C. Liebchen, and A.-R. Sadeghi, "Can't touch this: Practical and generic software- only defenses against rowhammer attacks," *arXiv preprint arXiv:1611.08396*, 2016.

[77] D. Gruss, C. Maurice, and S. Mangard, "Rowhammer. js: A remote software-induced fault attack in javascript," in *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, pp. 300–321, Springer, 2016.

[78] G. Irazoqui, T. Eisenbarth, and B. Sunar, "Mascat: Stop- ping microarchitectural attacks before execution.," *IACR Cryptology ePrint Archive*, vol. 2016, p. 1196, 2016.

[79] Y. Azar, S. Kamara, I. Menache, M. Raykova, and F. B. Shepard, "Co-location-resistant clouds.," *CCSW*, vol. 14, pp. 9–20, 2014.

[80] M. Li, Y. Zhang, K. Bai, W. Zang, M. Yu, and X. He, "Improving cloud survivability through dependency based virtual machine placement.," in *SECRYPT*, pp. 321–326, 2012.

[81] S.-J. Moon, V. Sekar, and M. K. Reiter, "Nomad: Miti- gating arbitrary cloud side channels via provider-assisted migration," in *Proceedings of the 22nd acm sigsac conference on computer and communications security*, pp. 1595–1606, ACM, 2015.

[82] Y. Han, T. Alpcan, J. Chan, and C. Leckie, "Security games for virtual machine allocation in cloud comput- ing," in *International Conference on Decision and Game Theory for Security*, pp. 99–118, Springer, 2013.

[83] R. M. Yoo and H.-H. S. Lee, "Adaptive transaction scheduling for transactional memory systems," in *Pro- ceedings of the twentieth annual symposium on Par- allelism in algorithms and architectures*, pp. 169–178, ACM, 2008.

[84] M. Mulazzani, S. Schrittwieser, M. Leithner, M. Huber, and E. R. Weippl, "Dark clouds on the horizon: Using cloud storage as attack vector and online slack space.," in *USENIX security symposium*, pp. 65–76, San Francisco, CA, USA, 2011.

[85] K. Onarlioglu, L. Bilge, A. Lanzi, D. Balzarotti, and E. Kirda, "G-free: defeating return-oriented program- ming through gadget-less binaries," in *Proceedings of the 26th Annual Computer Security Applications Conference*, pp. 49–58, ACM, 2010.

[86] A. Saeed, Cross-VM network attacks & their countermea- sures within cloud computing environments. PhD thesis, Lancaster University, 2020.