Network and Hypervisor-Based Attacks in Cloud Computing Environments

Abstract: Cloud Computing (CC) has become one of the most transformative computing technologies and a key business avenue, following in the footsteps of main-frames, minicomputers, personal computers, the World Wide Web and smartphones. Its vital features have considerably reduced IT costs, contributing to its rapid adoption by businesses and governments worldwide. Despite the many technological and economic benefits that CC offers, at the same time, it poses complex security threats resulting from the use of virtualisation technology. Compromising the security of any component in the cloud virtual infrastructure will negatively affect the security of other elements and so impact the overall system security. Therefore, to create a practical understanding of such threats, this paper provides an analysis of common and underexplored network- and hypervisor-based attacks against CC systems from a technical viewpoint.

Keywords: Cyber security, Threat intelligence, Artificial intelligence, Machine learning, Cyber physical systems, Digital forensics, Big data

1 Introduction

Cloud Computing (CC), still an evolving paradigm, has become one of the most transformative computing technologies and a key business avenue, following in the footsteps of main-frames, minicomputers, personal computers, the World Wide Web and smartphones (Ruan, et al., 2011; Montasari, 2017a). CC is a shared collection of configurable networked resources (e.g., networks, servers, storage, applications and services) that can be reconfigured quickly with minimal effort (Mell & Grance, 2011). Its vital features have considerably reduced IT costs, contributing to its swift adoption by businesses and governments worldwide. As a result, CC has drastically transformed the way in which Information Technology (IT) services are created, delivered, accessed and managed (Hosseinian-Far, et al., 2018). Such a transformation, that offers many technological and economic benefits (Hosseinian-Far, et al., 2017), has produced substantial interest in both academia and industry. However, being still in its infancy, CC encounters many issues in strategy, capabilities, technical, organizational, and legal dimensions. The new concepts that cloud introduces such as multi-tenancy, resource sharing and outsourcing poses numerous security threats with devastating consequences. As a result, many organisations do not completely move their business IT infrastructure to the cloud mainly due to the fears of security threats. Some of these fears along with others are due to the issues such as processing of sensitive data outside organisations, shared data and ineffectiveness of encryption, etc. (Heiser & Nicolett, 2008; Montasari, 2017b; Montasari, et al., 2018a). Moreover, forensics in the cloud is also a challenging task (Jahankhani & Hosseinian-Far, 2015).

In view of its security requirements (confidentiality, integrity, availability, accountability, and privacy-preservability), new security policies, Digital Forensic models and protocols will need to be developed in order to mitigate these security challenges (Montasari, et al., 2015; Montasari, 2016a; Montasari, 2016b; Montasari, 2016c; Montasari, 2016d; Montasari, 2017c; Montasari, 2018; Montasari, et al., 2019a; Montasari, et al., 2019b; Montasari, et al., 2019c; Montasari, et al., 2019d). To this end, we identify and analyse both common and underexplored network and hypervisor-based attacks in CC from a technical viewpoint. Accordingly, in a follow-up study (currently under a review process), we will recommend emerging solutions with a view to mitigating such stacks. We will also provide insights into the future security perspectives in an attempt to generate a fresh perspective on developing more effective security solutions for cloud systems.

This study only focuses on analysing the technical aspects of cyber-security threats in cloud. To this end, this analysis emphasises the complexity, intensity, duration and distribution of the attacks, outlining the major challenges in safeguarding against each attack. Investigating other security aspects such as organisational, compliance, physical security of data centers, and the way in which an enterprise can meet regulatory requirements is outside the scope of this paper. Similarly, providing an exhaustive list of attack vectors is outside the scope of this study. The remainder of the paper is structured as follow. Section 2 and 3 provide a background to CC technology and cloud security respectively. Section 4 provides an analysis of network-based attacks, while Section 5 examines hypervisor-based attacks. In Section 6, a discussion is provided, and finally the paper is concluded in Section 7.

2 Background to Cloud Computing

National Institute of Standards and Technology (NIST) (Mell & Grance, 2011) define CC as: “a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.”

According to the NIST’s wider definition of CC, this model consists of (i) five important characteristics, (ii) three service models, and (iii) four deployment models as represented in Figure. 1. Sub-sections 2.1 to 2.3 briefly describe these elements as categorised in the NIST’s definition.

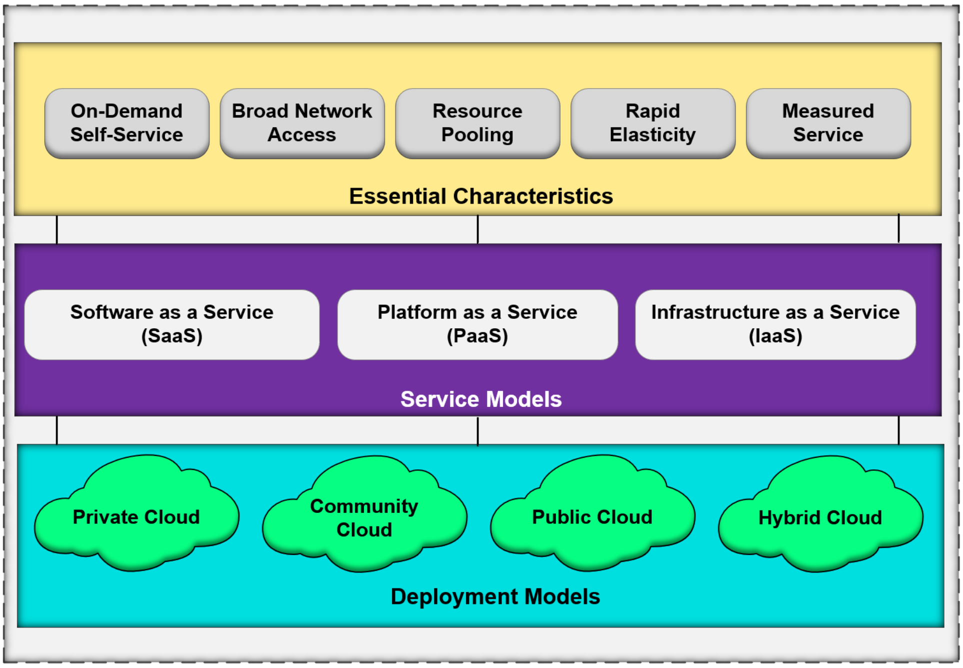

Figure 1: Graphical representation of Cloud Model based on NIST's definition.

2.1 Background to Cloud Computing

In On-Demand Self-Service, a customer will be able individually to provide computing capabilities, such as server time and network storage, as required without any need for human interaction with each cloud service provider (CSP). In Broad Network Access, capabilities are accessible over the network and via standard mechanisms which promote use by heterogeneous thin or thick client platforms. Resource Pooling involves the CSP's computing resources being pooled to serve multiple customers through the deployment of a multi-tenant model. In Rapid elasticity, capabilities can be elastically provided in order to scale rapidly outward and inward proportionate with demand. Measured Service involves cloud systems regulating and optimising resource use by taking advantage of a metering capability at some level of abstraction consistent with the type of service (Mell & Grance, 2011).

2.2 Service Models

As well as Private, Public, Community and Hybrid cloud environments (see Section 2.3) that host and store consumers' data, clouds are also separated into service categories deployed for various types of computing. Three are three main Cloud Computing Service Categories including: Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) as depicted in Figure 2. SaaS enables customers to use the CSP's applications running on a cloud infrastructure without having to download or install programs to engage with any transactions. They can access applications through client devices via a thin client interface, such as a web browser or a program interface. In this model, customers do not manage the underlying cloud infrastructure including network, servers, operating systems (OSs), storage or individual applications.

In PaaS, CSPs are responsible for providing consumers with hardware and software tools such as those required for application development over the Internet. Developers are not required to install in-house hardware and software to develop or run a new application. Furthermore, in this model, consumer-created or acquired applications (created using programming languages, libraries, services, and tools supported by the provider) are installed onto cloud. In this model, the CSP is responsible for managing the underlying cloud infrastructure such as network, servers, OSs, or storage. However, customers control the installed applications and probably configuration settings. IaaS is a self-service model for accessing, monitoring and managing remote data center infrastructures, such as hardware, storage, networking, and networking services. In SaaS, CSPs manage the underlying cloud infrastructure such as virtualisation, servers, hard drives, storage, and networking. However, customers are responsible for managing applications, data, runtime, middleware, and OSes.

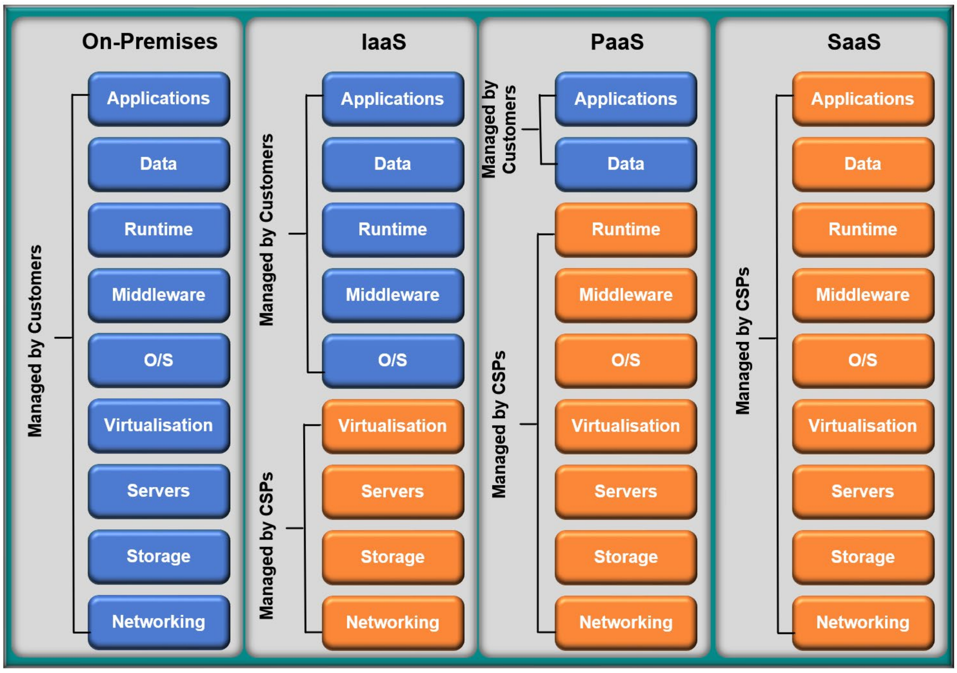

Figure 2: The three main Cloud Computing Service categories.

2.3 Deployment Models

In a Private Cloud, the cloud infrastructure is offered for exclusive deployment by a single organisation consisting of multiple customers. A Private Cloud could be owned, managed, and operated by the organisation, a third party or the combination of them. It could also exist on or off premises. In a Community Cloud, the cloud infrastructure is offered for exclusive deployment by a particular community of customers from organisations with similar concerns (such as security requirements). A Community Cloud could be owned, managed, and operated by one or more of the organisations in the community, a third party or the combination of both. It could also exist on or off premises. In a Public Cloud, the cloud infrastructure is offered for open deployment by the general public. A Public Cloud could be owned, managed and operated by a business, academic, government organisation or a combination of them. It could exist on the premises of the CSP. In a Hybrid Cloud, the cloud infrastructure is a formation of two or more separate cloud infrastructures that continue to remain unique entities. However, they are interconnected by standardised or proprietary technology that allows data and application portability (Mell & Grance, 2011).

2.4 Broad Divisions of Cloud Architecture

The wider divisions of cloud infrastructure include both front-end and back-end infrastructure as depicted in Figure 3. Back-end delivers the security of data for cloud users as well as the traffic control mechanism. It also offers the middleware that assists in connecting digital devices and their communication. The cloud technology architecture also includes front-end platforms entitled the cloud client that consists of servers, thin and fat client, tablets and mobile devices. The interaction is made via middleware, web-browser or virtual sessions. The CC architecture is an amalgamation of both services-oriented architecture and event-driven architecture. In consequence, cloud architecture comprises of all aspects of the CCE.

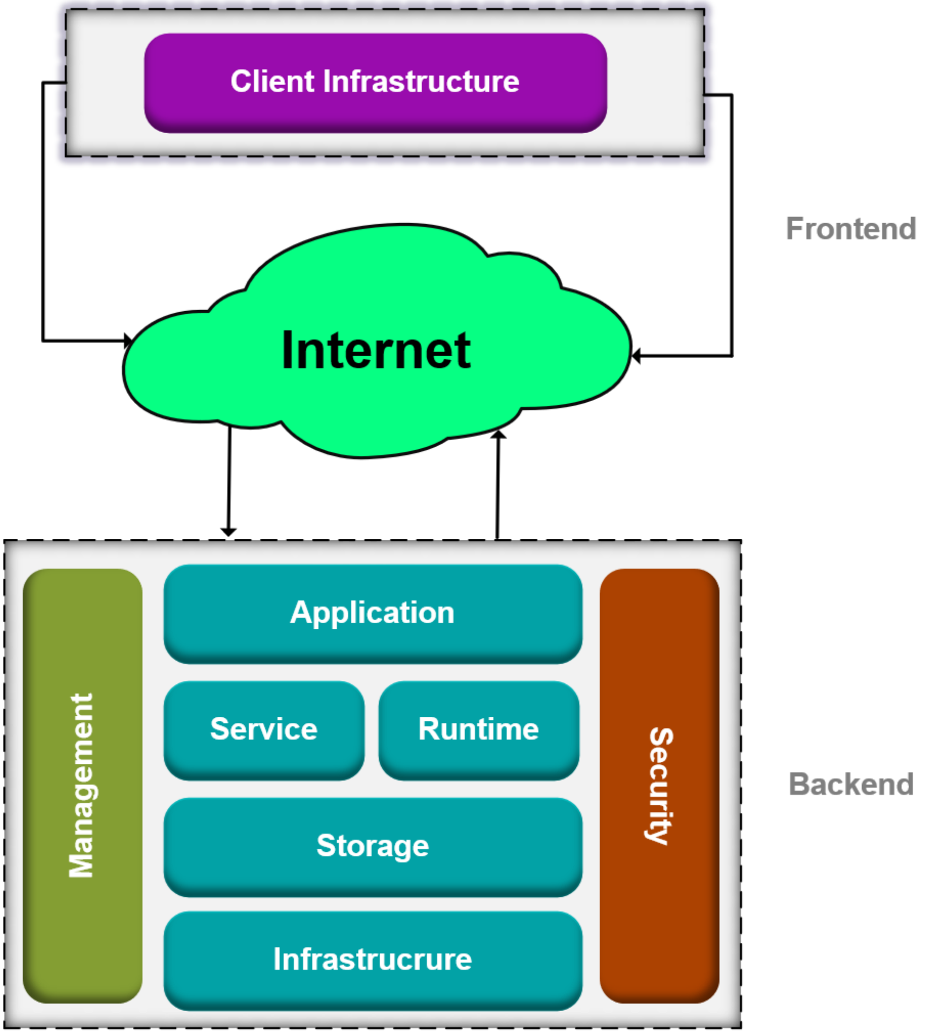

Figure 3: The three main Cloud Computing Service categories.

2.5 Cloud Security

Cloud security relates broadly to a set of measures adopted to safeguard cloud-based systems, digital assets, cloud data, applications and infrastructures (Jahankhani & Hosseinian-Far, 2015). These measures consist of policies; controls; procedures and technologies such as two-factor authorisation, deployment of virtual private networks (VPNs), security tokens and data encryption, etc. Such measures are put together to safeguard cloud data and customers' privacy, assist with regulatory compliance and implementing authentication rules for individual users and devices. Elements of security for CCEs, irrespective of a Public, Private or Hybrid Cloud, are to some extent similar to those for any On-Premise IT architecture. These include aspects such as unauthorised data exposure and leaks, weak access controls, vulnerability to attacks, and availability disruptions, that impact both traditional IT and CC systems alike. However, the fact that CCEs are highly interconnected renders them more vulnerable to security breaches and cyber-attacks (depending on the type of the attack). All cloud models are vulnerable to cyber threats, examples of which include data breaches, data loss, account hijacking, service traffic hijacking, insecure application program interfaces (APIs) and shared technology that can undermine cloud security. In consequence, many organisations are naturally concerned about moving their mission-critical systems to the cloud. Recent advances in machine learning, artificial intelligence and prediction (Farsi, et al., 2018) may offer suitable solutions to analyse relevant threats and secure systems.

In the following two sections, we shall identify and analyse both network-based and hypervisor-based attacks with a view to establishing a technical understanding of these types of security threats against CC systems. In particular, we will focus on those threats that are under-explored.

3 Network-Based Attacks

A network-based attack involves attacking applications running on a cloud system. These attacks can be passive in that information is monitored or active meaning that information is modified with the aim to steal, corrupt or destroy data or the network itself. The following sub sections provide an analysis of each attack vector followed by its associated countermeasure/s.

3.1 Denial and Distributed Denial of Service

The cloud infrastructure could easily become unavailable through a successful Denial of Service (DoS) attack, which in a CC setting can be more harmful than in a traditional computing context. This is due to the fact that when the workload for a particular service grows, the CC offers extra computational power to that service. On the one hand, the cloud system counter-attacks the impact of the attack. However, on the other hand, it inadvertently supports the perpetrators in their malicious activities by offering them more resources (Deshmukh & Devadkar, 2015; Coppolino, et al., 2018). A DoS attack is performed by utilising a single Internet connection to take advantage of a software vulnerability to flood the target with bogus requests. By carrying out a DoS attack, perpetrators can render a website unavailable and unresponsive to its legitimate users and disrupt the entire online user database. This adversary, who can be a real person, or a group of zombies controlled by him, can transmit many packets to the target with spoofed source IP addresses.

The Distributed Denial of Service Attack (DDoS) are the most threatening types of network-based attacks. They can be categorised into two groups based on the protocol level on which they focus including Network/Transport Level and Application Level Attacks. Compared to a DoS attack, a Distributed Denial of Service (DDoS) attack is often launched in an orchestrated manner from multiple connected devices that are distributed across the Internet. To carry out a DDoS attack, perpetrators first will need to employ an army of bots, called a botnet. In order to create a bot from a vulnerable computer, attackers will need to create specialised malware so that they can spread to more susceptible computers. The malware can then spread through compromised websites, e-mail attachments or organisations' networks. A user deceived into running the malware will inadvertently render his computing device into a bot and as a result, provide the attacker with an access point to his computer. Having been turned into a bot, the device connects to the attacker's control circuits that start sending requests from these centralised machines. These requests are comprised of directions for initiating an attack from bot malware against a specific target through selected attack methods. Examples of the most infamous botnets responsible for the DDoS attacks include: Srizbi, Kraken/Bobax, and Rustock).

Security threats posed by DoS and DDoS attacks are further aggravated in CCEs due to the computational power that cloud offers. This computational power enables perpetrators to compromise even more machines so as to attack larger number of systems. For instance, by utilising the cloud's on-demand self-service abilities, robust botnets can be rapidly developed. Adversaries could utilise malware-as-a-service (MaaS) to initiate more advanced DDoS attacks through heterogeneous thin or thick client platforms. DDoS attacks are growing larger and more advanced since they are capable of targeting specific applications such as DNS, HTTP or VoIP. Given computing capabilities of Broad Network Access, mobile devices such as smartphones and tablets are becoming an important platform for launching DDoS attacks. The increasing bandwidth, processing power and the absence of mobile devices security render them an important platform for attackers to undermine cloud resources.

3.2 Malware Injection Attacks

A Malware Injection attack in cloud involves hijacking a user's information. In Malware Injection attacks, perpetrators add an infected service implementation module to a SaaS or PaaS solution or a malicious VM instance to an IaaS solution. In a successful attack, the cloud system is deceived into redirecting the cloud user's requests to the attacker's module or instance, launching the execution of malicious code. In so doing, the perpetrator will be able to manipulate or hijack data or conduct eavesdropping. One of the ways to achieve this is by uploading a made-up image and deceiving the user into trusting that the image is part of his cloud environment. Once the malicious system or service is injected to the cloud environment, user requests will start forwarding to it, rendering the vulnerable code to execute. Cross-Site Scripting attacks and SQL Injection attacks are the most common forms of Malware Injection attacks.

3.3 Drive-by Attacks

This attack involves spreading malware through a malicious script that a perpetrator plants into HTTP or PHP code on a webpage of an insecure website. The malicious script can install malware directly to the user's machine or redirect him to a fake website controlled by the attacker. In contrast with other types of cyber-security attacks, a Drive-by attack does not require a victim to undertake any step that facilitates the attack. This is because it can simply exploit an application, OS or the web browser that contains security flaws. One of the methods to counter Drive-by attacks is to keep web browsers and OSs up-to-date and avoid websites that might contain malicious code. Also, unnecessary programs, applications or plug-ins should be removed.

3.4 Man-in-the-Middle Attacks

In a MitM attack, a perpetrator intercepts and saves old messages for transmission at a later stage, impersonating one of the participants. While there is currently no single technology or configuration to prevent all MitM attacks, encryption and digital certificates can be employed as an effective safeguard against such attacks. This ensures both the confidentiality and integrity of communications. However, encryption will not be an effective countermeasure in advanced cases in which MitM attacks are injected into the middle of communications.

3.5 Cross-Cloud Attacks

A Cross-Cloud Attack occurs when customers move their workloads into a public Cloud environment (such as Amazon Web Services or Microsoft Azure) and use a Virtual Private Network to switch between the public Cloud and the private Cloud. As a result, a perpetrator compromising one of the environments will be able to operate irrespective of the securitA Cross-Cloud attack occurs when customers move their workloads into a Public Cloud environment (such as Amazon Web Services or Microsoft Azure) and use a VPN to switch between the Public Cloud and the Private Cloud. As a result, a perpetrator compromising one of the environments will be able to operate irrespective of the security tools in operation. Once the perpetrator scans the environment, he will be able to use traditional vulnerabilities and exploits in order to gain an advantage in the Public Cloud. Such an attack can be detected in the public cloud even though its security measures are weaker than that in on-premise environments. Perpetrators have an advantage in switching between Public and Private Clouds facilitating a persistence in a target network.y tools in operation. Once the perpetrator scans the environment, he will be able to use traditional vulnerabilities and exploits in order to gain an advantage in the Public Cloud. Such an attack can be detected in the public cloud even though its security measures are weaker than that in on-premise environments. Perpetrators have an advantage in switching between public and private clouds facilitating a persistence in a target network.

3.6 Session Riding Attacks

A Session Riding attack (SRA), also known as Cross Site Request Forging Attack (CSRF), is a method to spoof requests on behalf of a genuine user. It enables attackers to spoof online transactions, change user details and withdraw funds. In an SRA Attack, a perpetrator employs third-party web resources to run scripts in the victim's web browser. A payload with malicious JavaScript will be injected into a website's database. Once the victim requests a page from the website, it will send the webpage containing the attacker's payload to the victim's browser, which will execute the malicious script. In a cookie-based session management, after a cookie is set by a web application, the browser will automatically attach it to every further request sent to the application.

For instance, when a user logs into an application, the application allocates a random and unique session token for the session and set it in the cookie. Subsequent communications by the user will include this cookie as the browser will attach the cookie automatically in the request header and send it to the application. If the application contains a form-based transaction for placing an order, the attacker can then easily draw off funds in such a situation. In order to safeguard against SRA attacks, data input must be sanitised in an HTTP request before retransmitting it back. Furthermore, data must be validated, filtered or escaped prior to retransition to the users.

3.7 Teardrop Attacks

By carrying out a Teardrop attack, perpetrators will be able to force the length and fragmentation offset fields in sequential Internet Protocol (IP) packets to overlap one another on the attacked host. The system attack will then attempt to reconstruct packets during the process; however, it fails to do so. Thus, the attacked host becomes confused and as a result crashes. To protect against Teardrop attacks, patches will need to be kept up-to-date. If this is not always possible, SMBv2 will need to be disabled, or ports 139 and 445 must be blocked.

3.8 Eavesdropping

In an Eavesdropping attack, the perpetrator intercepts network traffics in order to steal a victim's confidential information such as passwords, credit card numbers and other private information that the victim might send through the network. Eavesdropping attacks can be both passive and active. Passive Eavesdropping involves the attackers detecting the information by listening to the message transmission in the network. In contrast, in an active Eavesdropping, the attacker actively hijacks the information by impersonating a legitimate element and by sending queries to transmitters. Identifying passive Eavesdropping attacks are more important than detecting the active ones. This is due to the fact that active attacks will need perpetrators to learn about the legitimate element by performing passive eavesdropping in advance. Encryption methods can be used as a countermeasure to mitigate Eavesdropping attacks.

3.9 Smurf Attacks

Through a Smurf Attack, the attackers will be able to overwhelm a target network with traffic. This attack involves utilising IP spoofing and ICMP echo requests levied against broadcast IP addresses. Such ICMP requests stem from a spoofed victim address. For instance, if the intended victim address is 10.0.0.10, the attacker would spoof an ICMP echo request from 1Through a Smurf Attack, the attackers will be able to overwhelm a target network with traffic. This attack involves utilising IP spoofing and ICMP echo requests levied against broadcast IP addresses. Such ICMP requests stem from a spoofed victim address. For instance, if the intended victim address is 10.0.0.10, the attacker would spoof an ICMP echo request from 10.0.0.10 to the broadcast address 10.255.255.255. ICMP spoofing is a repeatable process that can be automated to create significant volume of network congestion.

In order to safeguard digital devices against Smurf attacks, one will need to disable IP-directed broadcasts at the routers so as to stop the ICMP echo broadcast requests at the network devices. Another countermeasure is to configure the end systems to prevent them from responding to ICMP packets from broadcast addresses.0.0.0.10 to the broadcast address 10.255.255.255. ICMP spoofing is a repeatable process that can be automated to create significant volume of network congestion. In order to safeguard digital devices against Smurf Attacks, one will need to disable IP-directed broadcasts at the routers so as to stop the ICMP echo broadcast requests at the network devices.

Another option would be to configure. Another countermeasure is to configure the end systems to prevent them from responding to ICMP packets from broadcast addresses (Melnick, 2018).

3.10 Account or Service Hijacking Attacks

An Account Hijacking attack within cloud environment occurs when an attacker steals or hijacks a customer's or an enterprise's security credentials and eavesdrops on activities and transactions. This type of attack is a common method in identity theft schemes, in which the adversary modifies data, inserts false information and redirects clients to illegitimate websites. The adversary could use a compromised email account or other credentials to impersonate the account owner. This attack at the enterprise level can have serious consequences. For instance, the organisation's integrity and reputations can be destroyed, and confidential data can be leaked or falsified resulting in substantial cost to the organisation or their customers.

3.11 TCP SYN Flood Attacks

In a TCP SYN Flood attack, a perpetrator exploits the buffer space during a Transmission Control Protocol (TCP) session initialisation handshake. His digital device floods the target system's small in-process queue with connection requests. However, the device will not respond when the target system answers those calls. As a result, the target system times out whilst awaiting the response from the perpetrator's device. This will lead to the system crashing or becoming ineffective when the connection queue fills up. In order to address this attack, one could place servers behind a firewall configured to halt inbound SYN packets. Another countermeasure would be to increase size of the connection queue and reduce the timeout on open connections.

3.12 Ping of Death Attacks

This attack involves using IP packets to ping a target system with an IP size over the maximum of 65,535 bytes. Considering that IP packets of this size are not permitted, perpetrators will need to fragment the IP packet. After the target system reassembles the packet, it will be subjected to buffer overflows, and as a result, it will crash. This type of attack can be prevented through a firewall that is able to examine fragmented IP packets for maximum size.

3.13 Secure Data Transmission

When transmitting data from clients to the cloud, data must be transmitted by using an encrypted secure communication channel like SSL/TLS to prevent attacks such as MitM, in which adversaries could intercept the communication and subsequently steal data (Lukan, 2014). Table below provides further examples of other threats associated with secure data transmission (Saripalli & Walters, 2010):

Table 1: Benefits of Tactical, Technical, Operational and Strategic CTI.

Threat	Description
Cross Site Scripting	Scripts are executed in a victim's browser to steal client sessions, destroy sites and present worms, etc.

Injection Flaws	Data sent by the client to a web application is not properly accepted. This will result in an inquisition on the server.
Malicious File Execution	Attackers will be able to run codes remotely, install a root kit, undermine the entire system and compromise the internal system, by means of SMB file wrappers for the PHP scripting language.
Insecure Cryptographic Storage	Those cloud services that do not deploy encryption techniques to ensure data transmission become vulnerable to attacks.

3.14 Insecure Application Programming Interfaces

Different cloud services are exposed by Application Programming Interfaces (APIs). Because the APIs are accessible from anywhere, adversaries could utilise them to undermine integrity and confidentiality of customers' data. An adversary who has acquired a token used by a customer to access the service via the API can utilise the same token to modify the customer's data.

4 Hypervisor-Based Attacks

Virtualised environments are usually implemented with the use of a hypervisor, which is a software layer that sits between a VM and the physical hardware. Hypervisors are often implemented as a software layer. They can also be implemented as code embedded in a system's firmware. There are two types of hypervisors. Type 1 hypervisors, also called "Bare Meta", are implemented directly on top of the system's hardware without any underlying OSs or other software. This type is the most common for the enterprise data centres. Instances of Type 1 hypervisors include VMware vSphere or Microsoft Hyper-V. Type 2 hypervisors, also called "Hosted Hypervisor", operate as a software layer on top of a host OS. Examples of Type 2 hypervisors consist of VMware Player and Parallels Desktop. Type 2 hypervisors are usually deployed on endpoints like PCs.

A hypervisor is aimed at running several VMs, each of which hosts an OS and applications concurrently on a single host computer, and providing abstraction among various guest VMs. Multi-tenancy in VM-based cloud infrastructures, along with the way in which physical resources are shared with guest VMs, can lead to new sources of threat. The gravest threat originates in the fact that malicious code can potentially leak out of the boundaries of its VM and affect the hypervisor or other guest VMs. Hypervisor is the most wanted runtime space since it contains Ring-1 privileges; therefore, commands can be executed from this space. The hypervisor can gain access to any resource in the host system (such as memory, peripherals, CPU state, etc). This denotes that it has the ability to access every guest VM's resources.

By exploiting vulnerabilities in a hypervisor, attackers could potentially find access to the physical host in which other adjacent VMs exist. CSPs of the IaaS model offer their services in a scalable manner in order to support multiple tenants sharing the infrastructure. Often, the underlying components such as Central Processing Unit (CPU) caches and Graphics Processing Units (GPUs) that comprise this infrastructure have not been designed to provide strong isolation properties for a multi-tenant architecture. To address this limitation, a virtualisation hypervisor mediates access between guest OSs and the physical compute resources. However, hypervisors, themselves, have displayed signs of vulnerabilities that have enabled the guest OSs to gain inappropriate levels of control or influence on the underlying platform (Alliance, Coud Sercurity, 2010; Montasari, et al., 2018b; Montasari, et al., 2019a; Montasari, et al., 2019e; Montasari, et al., 2019f; Montasari, et al., 2019g). The following sub sections describe various scenarios in which flaws in hypervisors can be exploited to perpetrate cyber-attacks in CCEs.

4.1 Hypervisor Flaws

A successful exploitation of a flaw present in a hypervisor could facilitate a cyber-attack that affects the three security properties of the hypervisors – confidentiality, integrity or availability – or one of its guest VMs. For instance, through such vulnerabilities, an adversary could potentially write to an arbitrary out-of-bounds memory location in the power management code of a hypervisor by fuzzing emulated I/O ports (Grance & Jansen, 2011). A DoS vulnerability is also likely to allow a guest VM to crash the host computer in addition to the other VMs being hosted. The attacker can also gain administrative control of guest VMs by using a MitM attack to change the codes used for authentication purposes. A compromised hypervisor can enable the adversary to attack each VM on a virtual host. This can lead to an upsurge in the resource usage of a VM that results in a DoS across the host or even among a group of servers. The security of a computer system is reliant on the quality of the underlying software kernel that controls the confinement and implementation of processes.

4.2 Virtual Machine Escape

A Virtual Machine Escape (VME) is associated with a vulnerability in the OS installed in a VM. The physical servers run multiple VMs on top of hypervisors. Adversaries could potentially exploit a hypervisor remotely by taking advantage of the vulnerability present in the hypervisor, itself. When such a weakness is exploited by the attackers, they will able to execute malicious code, escape the boundaries of the VM, gain access to the hypervisor and ultimately all the VMs running on it.

4.3 Rootkits

Rootkits are also potential means of hypervisor attack even though they are less common. A Rootkit in Hypervisor is a type of attack within could computing environments. In this attack, VM-based rootkits launch a hypervisor to undermine the existing host OS to a VM. The new guest OS will be under the impression that it is running as the host OS with the related control over the resources, whereas in reality this host does not exist. Hypervisor also forms a covert channel to inject unauthorised code into the system. This will enable the attacker to control the VM running on the host machine and to manipulate the operations on the system.

4.4 Malicious Virtual Machine

In an IaaS model, attackers could place a malicious VM co-resident to a victim’s VM to target cryptographic implementation in the system. This can facilitate the extraction of valuable information from the target VM. In order to ensure that the malicious VM has been placed next to the target VM, the attacker will need to employ various methods, including: Brute Forcing and Network Based Co-Residence Check, which are only specific to Elastic Compute Cloud (EC2). In Brute Forcing, the adversary initiates the VM and then repeatedly checks for the target in a zone by shutting down the VM that has been created in the wrong zone and repeating the process. Similarly, Network Based Co-Residence Check can be carried out via three main methods as described in Table 2.

Table 2: Benefits of Tactical, Technical, Operational and Strategic CTI.

Method	Description
Dom0 IP Address	This is specific Xen hypervisor and is considered to be the initial domain started. For instance, the Dom0 IP can be checked for the first hop on any route from the host. Therefore, this Dom0 IP can be distinguished from another instance, if the target is uncontrolled by conducting a TCP SYN probe and tracing the last hop.
Packet Round Trip Times	It reveals a pattern for VM's in a same host.
Closeness of Internal IP Address	Co-residency can be determined in how internal IP address is assigned to a group of VM's from a single box.

4.5 Insecure Cryptography

Cryptography algorithms often require random generation of numbers by utilising unpredictable sources of information. If numbers utilised in cryptographic algorithm are not truly random, flaws can be discovered easily, and as a result numbers can be brute forced. The VMs deployed on the cloud do not have adequate sources of entropy, and as a result they are vulnerable to attacks. In client computers, the main source of randomisation is the movement of user mouse and key presses. However, servers are typically running without user interaction, which denotes lower number of randomisation sources. Thus, the VMs must depend on the sources that they have available to them, which could lead to easily guessable numbers that do not provide much entropy in cryptographic algorithms.

4.5 Interrupt and Timer Mechanisms

A hypervisor will need to mimic the interrupt and timer mechanisms which the motherboard offers to a physical machine. These include the Programmable Interval Timer (PIT), the Advanced Programmable Interrupt Controller (APIC), and the Interrupt Request (IRQ) mechanisms (Perez-Botero, et al., 2013). For instance, the absence of authentication of data contained in the PIT-related data structures can result in a rogue VM creating a full host OS crash which is a grave DoS attack.

4.6 I/O and Networking

A hypervisor also mimics I/O and networking. The device imitation is carried out via separation of labour by having two kinds of device drivers including front-end drivers (operating in guest VMs) and back-end drivers. Front-end drivers offer the separation that the guest OS requires. However, those drivers cannot access physical hardware directly considering that the hypervisor will need to intercede user accesses to shared resources.

Thus, front-end drivers communicate with back-end drivers which have full access to the underlying hardware in order to accomplish the required operations. In turn, back-end drivers impose access policies and incorporate the actual devices. Device imitation is often performed in high-level languages such as C and C++. Therefore, data separation becomes richer but more dangerous when hijacked (Perez-Botero, et al., 2013). For instance, the authors in (Elhage, 2011) describe a bug that has been deployed to create the Virtunoid attack. According to the authors (Elhage, 2011), QEMU attempted to hot-unplug any device the programmers wished, irrespective of the device's support for hot-unplugging. Consequently, the absence of state clean-up by some virtual devices led to use-after-free opportunities in which data structures that were formerly being utilised by a hot-unplugged virtual device remained in memory and could be hijacked with executable code by an attacker (Perez-Botero, et al., 2013).

4.7 Hypercalls

Hypercalls are similar to system calls and specific to hypervisors. They deliver a procedural interface, by means of which guest VMs can request privileged actions from the hypervisor. Hypercalls can be employed to probe CPU activity, deal with hard disk partitions, and generate virtual interrupts. Hypercall susceptibilities can enable an adversary (who operates a guest VM) to gain heightened privileges over the host system's resources. According to the CVE-2009-3290 case (CVE Details., 2017), in the past, KVM would enable unprivileged (Ring 3) guest callers to issue MMU hypercalls. Because the MMU command structures need to be sent as an argument to those hypercalls by their physical address, they can only make sense when they are issued by a Ring 0 process. The Ring 3 callers can still send random addresses as arguments to the MMU hypercalls even if they do not have access to the physical address space. This will result in crashing the guest VM or reading or writing to kernel-space memory segments (Perez-Botero, et al., 2013).

4.8 VM Management

VM Management functionalities comprise the set of basic administrative operations that a hypervisor must support. The configuration of guest VMs is defined in relation to their allocated virtual devices, dedicated PCI devices, main memory quotas, virtual CPU topologies and priorities, etc. (Perez-Botero, et al., 2013). The hypervisor will then need to be able to start, pause and stop VMs that are true to the configurations acknowledged by the CSP. These tasks are started by Xen's Dom0 and KVM's libvirt toolkit. Kernel images must be decompressed into memory and interpreted by the management domain when booting up a VM. A hypervisor's bootloader for paravirtualised images can use Python exec() statements to process the custom kernel's user-defined configuration file. This will result in the probability of executing random python code inside Dom0. By modifying the configuration file, an attacker can deceive Dom0 into issuing a command that would activate the destruction of another co-hosted domain.

4.9 Remote Management Software

Remote Management Software are web applications that run as a background process and are not necessary for the implementation of the virtualised environment. Their objective is generally to make easier the hypervisor's administration via user-friendly web interfaces and network-facing virtual consoles. Attackers can take advantage of the vulnerabilities in these applications from anywhere. Such exploitation can result in full control over the virtualised environment. For instance, according to the CVE-2008-3253 (CVE Details., 2017), a Cross-Site Scripting Attack on a remote administration console can reveal all of a hypervisor's VM Management operations to a remote attacker who has previously stolen a user's authentication cookies.

4.10 Hypervisor Add-ons

Some hypervisors, such as Xen and KVM, have modular designs that facilitate extensions to their operations (Hypervisor Add-ons). Hypervisor Add-ons increase the probability of hypervisor vulnerabilities that are already present because they add to the size of the hypervisor's codebase. For instance, a heap overflow (a type of buffer overflow that occurs in the heap data area) opportunity in one of the hypervisor's optional security modules can lead to a leak from an unprivileged domain directly to the hypervisor.

4.11 Sources of Trigger and Target of Attack

A hypervisor susceptibility reveals itself inside a module's code; however, it can be activated from a variety of runtime spaces and has the potential to target one or more of such runtime spaces. This includes: (i) Network, (ii) Guest VM's User-Space, (iii) Guest VM's Kernel-Space, (iv) Dom0/Host OS, and (v) Hypervisor (Perez-Botero, et al., 2013). The source of trigger and target of attack are of great significance in evaluating a susceptibility's ease of exploitability and effect. The source of trigger can be established through the comparison of the limitations of each of the runtime spaces with the implementation rights needed to regenerate the susceptibility.

Network is the runtime space that is relatively easy to achieve. A remote adversary can launch an attack on hypervisor and its guest VMs if it resides in a subnet where the machine operating the hypervisor is accessible. In Guest VM's User-Space, codes can be run from a guest VM's Ring 3 even though some performance will be restricted by the OS or the hypervisor. However, it is not difficult to render user-space code to be executed. Therefore, any loophole from Ring 3 is of great importance to an adversary. For instance, an attacker can employ the CPUID x86 instruction to launch an attack from a guest VM's Ring 3. In Guest VM's Kernel-Space, inserting malicious OS-level (Ring 0) code necessitates undermining the OS security (Sgandurra & Lupu, 2016). In IaaS cloud models, tenants have the ability to rent VMs and execute their OS of choice, which might be malicious. Therefore, an adversary can launch an attack from a guest VM's Kernel-Space because it needs control over the paravirtualised front-end driver.

5 Discussion

Cyber threats are constantly evolving and becoming more sophisticated, and CC is not an exception to this phenomenon. As a result, the security of cloud systems and cloud data is increasingly becoming essential as more organisations place their digital assets, data centers, business processes and more in the cloud. This highlights the inevitability for implementation of robust cloud security measures for those organisations with a cloud presence (or those transitioning to cloud). It is imperative that the appropriate security provisions are designed and implemented irrespective of deploying a native cloud, hybrid or on-premise environment. These security measures must be able to deliver all the functionality of traditional IT security and simultaneously enable organisations to harness the many benefits of CC whilst remaining secure. In addition, such measures must ensure that data privacy and regulatory compliance are met.

Ensuring a robust and effective cloud security can only be realised through an organisational culture of security and complete security solutions. Such comprehensive security solutions must be able to safeguard cloud apps and cloud data; preclude unauthorised access, data breaches, compromised accounts and other threats; and allow users to configure security policies on a per-device basis. Furthermore, in order to satisfy consumers' security concerns, CSPs will need to follow certain regulatory requirements for storing sensitive data such as credit card numbers and health information. The manner in which cloud security should be implemented will be unique to the individual CSP or the cloud security solutions in place. However, implementation of cloud security processes should be a joint responsibility between the customer and the CSP.

6 Conclusion

This paper identified and analysed both common and underexplored cyber-security attacks carried out within CCEs. To mitigate the attacks described in this paper, encryption must be used as the primary defence measure to keep cloud data secure. Encryption techniques must include complex and robust algorithms so as to be able to hide cloud-protected data. While it is possible to decrypt encryption, itself, doing so would be challenging considering the fact that such an undertaking would require a significant amount of computing process power and time, human skills and appropriate Digital Forensic tools. Intrusion Detection Systems (IDS) must also be used as the next line of defence to address such threats. A fully managed IDS could detect and alert the malicious use of cloud services by intruders. Such an IDS must be able to offer network monitoring and notifies the security teams of the abnormal network activities.

Furthermore, from our analysis, it can be deduced that the security fundamentals remain the same for each specific attack. These include: keeping the systems and anti-virus databases up-to-date, training employees and customers, configuring firewalls to whitelist only the specific ports, creating robust passwords, employing a least-privilege model in the IT environment, creating regular backups, and continuously auditing the IT systems for abnormal network activities. Furthermore, software versions, constant software security updates, monitoring networks with IDS/IPS systems, log monitoring, integrating SIEM into the network, security practices, vulnerability profiles, intrusion attempts, and security design, etc. are some of the main aspects for estimating an enterprise's security stance.

References

Alliance, Coud Sercurity, 2010. *Top threats to cloud computing v1. 0.* s.l.:White Paper.

Coppolino, L., D'Antonio, S., Mazzeo, G. & Romano, L., 2017. Cloud security: Emerging threats and current solutions. *Computers & Electrical Engineering,* Volume 59, pp. 126-140.

CVE Details., 2017. *Vulnerability Details : CVE-2017-12188..* [Online]
Available at: https://www.cvedetails.com/cve/CVE-2017-12188/
[Accessed May 2020].

Deshmukh, R. V. & Devadkar, K. K., 2015. Understanding DDoS attack \& its effect in cloud environment. *Procedia Computer Science,* Volume 49, pp. 202-210.

Elhage, N., 2011. *Virtunoid: Breaking out of KVM.* s.l.:Black Hat USA.

Farsi, M. et al., 2018. Crime data mining, threat analysis and prediction. In: H. Jahankhani, ed. *Cyber Criminology.* London: Springer, pp. 183-202.

Grance, T. & Jansen, W., 2011. *Guidelines on Security and Privacy in Public Cloud Computing,* s.l.: NIST.

Heiser, J. & Nicolett, M., 2008. Assessing the security risks of cloud computing. *Gartner Reprt,* Volume 27, pp. 29-52.

Hosseinian-Far, A., Ramachandran, M. & Sarwar, D., 2017. *Strategic engineering for Cloud computing and big data analytics.* 1 ed. London: Springer.

Hosseinian-Far, A., Ramachandran, M. & Slack, C., 2018. Emerging Trends in Cloud Computing, Big Data, Fog Computing, IoT and Smart Living. In: M. Dastbaz, H. Arabnia & B. Akhgar, eds. *Technology for Smart Futures.* London: Springer, pp. 29-40.

Jahankhani, H. & Hosseinian-Far, A., 2015. Challenges of Cloud Forensics. *International Workshop on Enterprise Security*, pp. 1-18.

Lukan, D., 2014. *Cloud Forensics: An Overview.* [Online]
Available at: https://resources.infosecinstitute.com/overview-cloud-forensics/
[Accessed May 2020].

Mell, P. & Grance, T., 2011. *The NIST definition of cloud computing..* s.l.:Computer Security Division, Information Technology Laboratory, National Institute of Justice.

Melnick, J., 2018. *Top 10 most common types of cyber attacks.* s.l.:Netwrix Blog.

Montasari, R., 2016a. Review and assessment of the existing digital forensic investigation process models. *International Journal of Computer Applications,* 147(7), pp. 41-49.

Montasari, R., 2016b. Formal two stage triage process model (FTSTPM) for digital forensic practice. *Int. J. Comput. Sci. Secur,* Volume 10, pp. 69-87.

Montasari, R., 2016c. A comprehensive digital forensic investigation process model. *International Journal of Electronic Security and Digital Forensics,* 8(4), pp. 285-302.

Montasari, R., 2016d. An ad hoc detailed review of digital forensic investigation process models. *International Journal of Electronic Security and Digital Forensics,* 8(3), pp. 205-223.

Montasari, R., 2017a. A standardised data acquisition process model for digital forensic investigations. *International Journal of Information and Computer Security,* 9(3), pp. 229-249.

Montasari, R., 2017b. An Overview of Cloud Forensics Strategy: Capabilities, Challenges, and Opportunities. In: *Strategic Engineering for Cloud Computing and Big Data Analytics.* London: Springer, pp. 189-205.

Montasari, R., 2017c. Digital Evidence: Disclosure and Admissibility in the United Kingdom Jurisdiction. *International Conference on Global Security, Safety, and Sustainability*, pp. 42-52.

Montasari, R., 2018. Testing the comprehensive digital forensic investigation process model (the CDFIPM). In: M. Dastbaz, H. Arabnia & B. Akhgar, eds. *Technology for Smart Futures.* London: Springer, pp. 303-327.

Montasari, R., Hill, R. & Carpenter, V., 2019d. A road map for digital forensics research: a novel approach for establishing the design science research process in digital forensics. *International Journal of Electronic Security and Digital Forensics,* 11(2), pp. 194-224.

Montasari, R., Hill, R., Carpenter, V. & Hosseinian-Far, A., 2019b. Evaluation of the Standardised Digital Forensic Investigation Process Model (ESDFIPM). In: H. Jahankhani, ed. *Cyber Security Practitioners' Guide.* s.l.:World Scientific.

Montasari, R., Hill, R., Carpenter, V. & Hosseinian-Far, A., 2019c. The Standardised Digital Forensic Investigation Process Model (SDFIPM). In: H. Jahankhani, et al. eds. *Blockchain and Clinical Trial.* London: Springer, pp. 169-209.

Montasari, R., Hill, R., Carpenter, V. & Montaseri, F., 2019a. Digital Forensic Investigation of Social Media, Acquisition and Analysis of Digital Evidence. *International Journal of Strategic Engineering (IJoSE),* 2(1), pp. 52-60.

Montasari, R., Hill, R., Hosseinian-Far, A. & Montaseri, F., 2019g. Countermeasures for timing-based side-channel attacks against shared, modern computing hardware. *International Journal of Electronic Security and Digital Forensics,* 11(3), pp. 294-320.

Montasari, R. et al., 2019f. Internet of Things Devices: Digital Forensic Process and Data Reduction. *International Journal of Electronic Security and Digital Forensics.*

Montasari, R. et al., 2019e. Hardware-Based Cyber Threats: Attack Vectors and Defence Techniques. *International Journal of Electronic Security and Digital Forensics.*

Montasari, R., Hosseinian-Far, A. & Hill, R., 2018a. Policies, innovative self-adaptive techniques and understanding psychology of cybersecurity to counter adversarial attacks in network and cyber environments. In: H. Jahankhani, ed. *Cyber Criminology.* London: Springer, pp. 71-93.

Montasari, R. et al., 2018b. Are Timing-Based Side-Channel Attacks Feasible in Shared, Modern Computing Hardware?. *International Journal of Organizational and Collective Intelligence (IJOCI),* 8(2), pp. 32-59.

Montasari, R., Peltola, P. & Evans, D., 2015. Integrated computer forensics investigation process model (ICFIPM) for computer crime investigations. *International Conference on Global Security, Safety, and Sustainability*, pp. 83-95.

Perez-Botero, D., Szefer, J. & Lee, R. B., 2013. Characterizing hypervisor vulnerabilities in cloud computing servers. *Proceedings of the Workshop on Security in Cloud Computing (SCC)*, May.

Ruan, K., Carthy, J., Kechadi, T. & Crosbie, M., 2011. Cloud Forensics. In: *IFIP International Conference on Digital Forensics.* Berlin: Springer, pp. 35-46.

Saripalli, P. & Walters, B., 2010. Quirc: A quantitative impact and risk assessment framework for cloud security. *2010 IEEE 3rd international conference on cloud computing*, pp. 280-288.

Sgandurra, D. & Lupu, E., 2016. Evolution of attacks, threat models, and solutions for virtualized systems. *ACM Computing Surveys (CSUR),* 48(3), pp. 1-38.