Hindawi
Security and Communication Networks
Volume 2018, Article ID 1681908, 20 pages
https://doi.org/10.1155/2018/1681908

Research Article

Securing Cloud Hypervisors: A Survey of the Threats, Vulnerabilities, and Countermeasures

John Patrick Barrowclough[1] and Rameez Asif[1,2]

[1] *Centre for Distributed Computing, Networks, and Security, School of Computing, Edinburgh Napier University, Edinburgh EH10 5DT, UK*
[2] *The Cyber Academy, Edinburgh Napier University, Edinburgh EH10 5DT, UK*

Correspondence should be addressed to Rameez Asif; r.asif@napier.ac.uk

Received 8 December 2017; Revised 19 April 2018; Accepted 10 May 2018; Published 11 June 2018

Academic Editor: Sherali Zeadally

The exponential rise of the cloud computing paradigm has led to the cybersecurity concerns, taking into account the fact that the resources are shared and mediated by a 'hypervisor' that may be attacked and user data can be compromised or hacked. In order to better define these threats to which a cloud hypervisor is exposed, we conducted an in-depth analysis and highlighted the security concerns of the cloud. We basically focused on the two particular issues, i.e., (a) data breaches and (b) weak authentication. For in-depth analysis, we have successfully demonstrated a fully functional private cloud infrastructure running on CloudStack for the software management and orchestrated a valid hack. We analyzed the popular open-source hypervisors, followed by an extensive study of the vulnerability reports associated with them. Based on our findings, we propose the characterization and countermeasures of hypervisor's vulnerabilities. These investigations can be used to understand the potential attack paths on cloud computing and Cloud-of-Things (CoT) applications and identify the vulnerabilities that enabled them.

1. Literature Overview

Over the past few years, demand for access to data for ever-increasing online users has grown exponentially, with the traditional data centre model not being able to cope with the access from anywhere and any device [1]. This changing world has forced the need to create a new way of supporting these demands; the cloud. This environment creates a model enabling ubiquitous, ondemand services with the advantages of rapid deployment and revenue savings [2]. Small businesses are embracing cloud technology because it allows them to use enterprise infrastructure only previously afforded by larger companies [3]. Although there is no universal definition of cloud computing [4], most authors seem to have adopted the National Institute of Standards and Technology (NIST) definition of three service models (service, platform, and infrastructure) and four deployment models (private, community, public, and hybrid) [5, 6].

1.1. Cloud Delivery Models. As more users move their services over to the cloud, the cloud providers are increasing their service offerings and the concept of an on demand, pay-as-you-go service is something the providers are pushing. This has been emphasized in the "as a service" referencing by authors [4]. The most prominent services are as follows and are depicted in Figure 1:

1.1.1. Infrastructure as A Service (IaaS). You can consider this as a customer using someone else's hardware (network, storage, and virtual machines) located in their remote data centre, with the provision and hardware maintenance of this service managed by the cloud provider. This service allows the consumer full control over the operating system, network, and storage to deploy their own security policies and applications [6].

1.1.2. Platform as a Service (PaaS). This is one layer above IaaS, where the consumer rents a fully working and supported

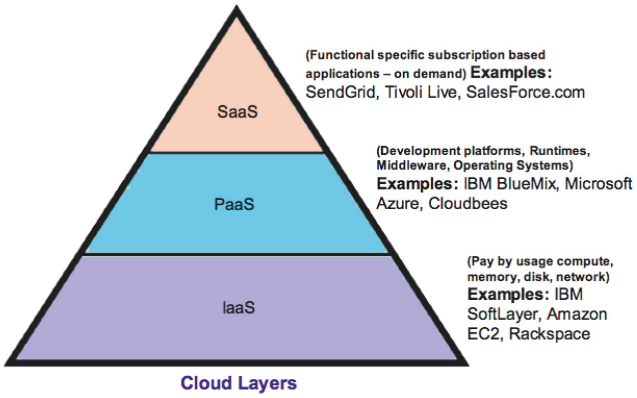

SaaS
PaaS
IaaS
(Functional specific subscription based applications – on demand) Examples: SendGrid, Tivoli Live, SalesForce.com
(Development platforms, Runtimes, Middleware, Operating Systems) Examples: IBM BlueMix, Microsoft Azure, Cloudbees
(Pay by usage compute, memory, disk, network) Examples: IBM SoftLayer, Amazon EC2, Rackspace
Cloud Layers

Figure 1: Cloud delivery models with respect to the services (source: ADDI Summit Dec, 2009).

environment; this could be an operating system, storage, or both. Some consider this as a development environment for the consumer, as it allows for application development without worrying about the underlying infrastructure [5]. However, this is a bit restrictive and there is nothing stopping this environment being used to develop, test, and deploy custom applications [7].

1.1.3. Software as a Service (SaaS). SaaS is the top-level service in the NIST cloud delivery model [1, 2]. You could consider this a fully outsourced service where the consumer buys an application as a pay-as-you-go service and does not expect to perform any support or maintenance for this application [5]. Example applications are Gmail and Dropbox [7].

1.2. Cloud Deployment Models. Based on the requirements and the services provided by the companies to the subscribers, cloud computing can be deployed in an organization through several deployment models. The following are the most common types of cloud deployments: public cloud, private cloud, community cloud, and hybrid cloud. They can be summarized as in Figure 2.

1.2.1. Private Cloud. A private cloud model is where a single consumer wants to take advantage of the clouds convenient on demand services, offered by the cloud providers. This may be for cost savings or just to take advantage of the ubiquitous and elasticity of the cloud [8, 9]. Customers who adopt a private cloud setup will feel they have more control over the security within this cloud and could force the cloud provider to implement specific custom requirements [9], a configuration which could be an ideal fit for a bank or government agency.

1.2.2. Public Cloud. As the name suggests, this is a model where the cloud provider offers public services on a pay-as-you-go basis [9]. These resources are shared with all consumers and because of this security is more challenging compared to the private cloud [6, 8]. However, this model does offer some advantages over the private cloud due to the economies of scale and its ability to offer short-term usage [9].

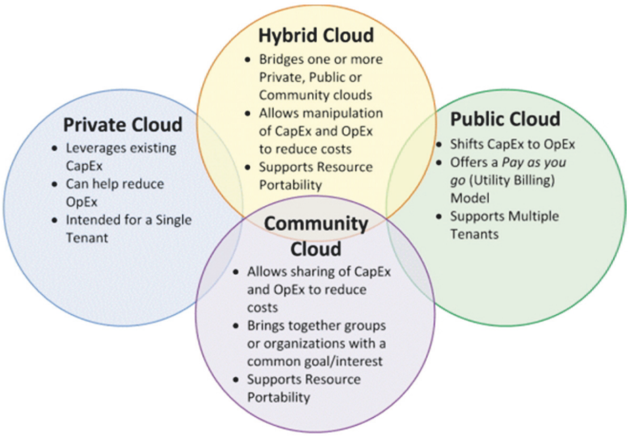

Hybrid Cloud
• Bridges one or more Private, Public or Community clouds
• Allows manipulation of CapEx and OpEx to reduce costs
• Supports Resource Portability
Private Cloud
• Leverages existing CapEx
• Can help reduce OpEx
• Intended for a Single Tenant
Public Cloud
• Shifts CapEx to OpEx
• Offers a Pay as you go (Utility Billing) Model
• Supports Multiple Tenants
Community Cloud
• Allows sharing of CapEx and OpEx to reduce costs
• Brings together groups or organizations with a common goal/interest
• Supports Resource Portability

Figure 2: Cloud deployment models with respect to the services. [Opex: operational expenditure is an ongoing cost for running a product, business, or system; Capex: capital expenditure is the cost of developing or providing nonconsumable parts for the product or system.] (source: NIST).

1.2.3. Community Cloud. Community Cloud is an infrastructure employed by several organizations and supports a specific community that shares common requirements, such as security or legal compliance policies. The community cloud model can provide greater cost savings than the private cloud while offering some of its security features. It may be managed by the organizations or a third party and may exist on premise or off premise [6].

1.2.4. Hybrid Cloud. A hybrid cloud is a mixture of public and private, a consumer reluctant to trust the public cloud model due to its general nature might feel more comfortable sharing a cloud with a group of similar organizations [8]. By grouping similar like-minded organizations custom security and data standards could be applied across the cloud [1, 7–9]

1.3. Virtualization. One of the main technologies enabling cloud computing to thrive is virtualization. A fundamental part of the virtualized environment is the hypervisor or virtual machine monitor (VMM) [5]. This reduced footprint operating system manages the physical platform and local resources and is responsible for the separation of resources for the guest systems running on this physical platform [10–12]; The hypervisor or VMM (virtual machine monitor) is the software which is responsible for managing the physical server resources (CPU, memory, and storage) [10, 12]; it is the management layer between the physical hardware and the virtual machines running above. The hypervisor controls the resource allocation to the virtual machines (VMs); these physical machines can be grouped together to form a large visualized infrastructure, expanding their capability to load balance or moving VMs between physical servers without any service downtime [11, 12]. It is this ability to share

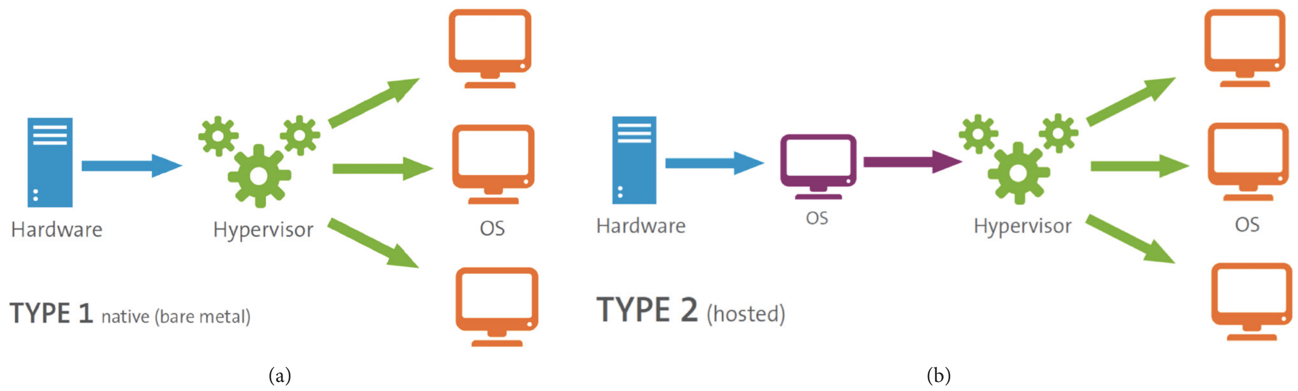


FIGURE 3: (a) Type 1 and (b) Type 2 hypervisors. [13].

resources and provide ondemand services that has enabled cloud service providers to offer their service available today.

There are two types of hypervisors:

(1) **Type 1**: It has a lightweight operating system that runs directly on the physical hardware [14]; sometimes these are referred to as bare-metal hypervisors [15]. By running directly on the hardware, they have full control over the underlying resources [16]. They tend to have a smaller footprint as shown in Figure 3, which reduces the vulnerabilities because this is generally proportional to the code size and because of this they are seen as having enhanced security compared to Type 2 hypervisors [11, 12, 14–16]. The example in Figure 3 shows a XenServer hypervisor, which is a Type 1; other similar types of hypervisors are VMware ESX and Hyper-V.

(2) **Type 2**: It runs on top of an existing operating system (OS), as shown in Figure 3 (image source: www.flexiant.com/2014/02/05/what-does-a-hypervisor-do/), and relies on that OS to monitor requests from the guest machines and send the requests to the appropriate application program interface (API) function. This type is sometimes referred to as a hosted platform [16]; it runs on a fully functioning operating system. The example in Figure 4 shows the KVM hypervisor, which is a type 2; other similar hypervisors are VMware Workstation, Microsoft Virtual PC, and Oracle Virtual box.

1.4. Vulnerabilities in Cloud Computing. Cloud security is a growing concern because the underlying concept is based on sharing hypervisor platforms, placing the security of the clients data on the hypervisors ability to separate resources from a multitenanted system and trusting the providers with administration privileges to their systems [13]. Compromising this hypervisor with a malware attack or gaining root permission for an attacker would allow full access to the shared memory of the physical machine and therefore the content of all the guest virtual machines (VMs) running on this physical platform [5]. This foothold in the visualized environment could be extended to target the shared data storage, further compromising the confidentiality, integrity, and availability of the entire infrastructure [3–5]. Insider threats are a real possibility and nothing new in the computer world but with the cloud the risks are greater due to the unprecedented amount of information and multiclients data [17].

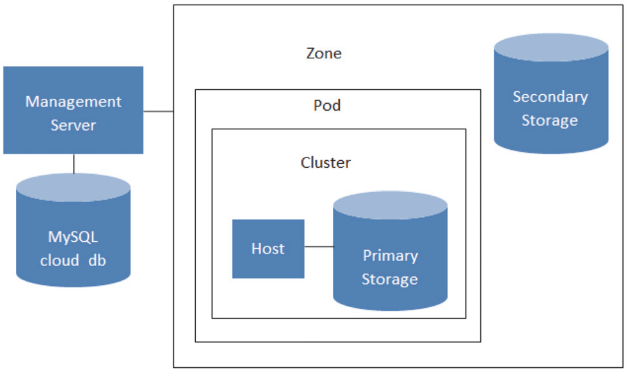


FIGURE 4: Schematic diagram of the basic CloudStack based lab implementation.

The Cloud Security Alliance (CSA) affectionately calls its top cloud threats in 2016 the treacherous twelve, 12 security concerns associated with cloud technology [18]. These are as follows:

(1) Data loss
(2) Weak identity, credentials, and access management
(3) Insecure APIs
(4) System and application vulnerabilities
(5) Account hacking
(6) Malicious insiders
(7) Advanced persistent threat (APTs)
(8) Permanent data loss
(9) Insufficient due diligence
(10) Abuse and nefarious use of cloud services
(11) DOS attacks
(12) Shared technology or shared dangers

In this paper, we have specifically implemented and evaluated in detail (Section 2) the two important security concerns in private cloud computing scenario, i.e., (a) data loss and (b) weak identity, credentials, and access management.

Data Loss. Cloud providers are enticing customers to their infrastructure with ondemand processing capabilities and large cheap storage solutions. This has seen a huge increase of firms moving or planning to move their data to the cloud [19]; some familiar companies are already using the cloud, for example, Netflix and Dropbox [20]. Data breaches are not a new security issue. Over the past few years the computer industry has seen some major data breaches; eight of the largest health-care trusts in the USA experienced data breaches in 2015, with over 100 million health records compromised [20]. These types of records tend to hold valuable data that rarely changes, e.g., date of birth or next of kin. As more firms move their sensitive data to the cloud it will become an attractive target for hackers.

(1) **Data storage and backup**: historically when a user backed up a server, the backup media would be stored in a safe place onsite and it would be noticeable if it went missing. In a cloud environment we now have to trust a third party with the process of securing this data and maintaining its safe protection [5]. Some suggest that it is this lack of trust that is impeding the growth of moving sensitive data to the cloud. Data breaches are not always intentional; data loss can be caused by a physical disk failure or power failure [1–5], in either case it all comes down to trusting the cloud provider to have the relevant process and procedures to secure and protect the customers data.

Virtualized machines are easier to create copies of, than that of a physical machine; this could be as simple as copying the VM container file or snapshots to removable storage [11]. Unscrupulous cloud administrators could copy these images and then use password-cracking tools to obtain the customers administrator details [20]. This could lead to further breaches by using these credentials to target the running VM [21, 22]. To reduce the chance of data loss from simply copying the image or snapshot, the storage needs to be protected against unauthorized access [7]. This could be done by encrypting data using cryptography techniques with the end user managing their own crypto keys, which would stop an insider trying to access the data [8]. File encryption is a feature on the majority of cloud providers but it is an optional setting and not the default; therefore if the storage was unencrypted and is no longer required, the cloud provider could reassign this area of storage to a new tenant. If this new tenant were malicious, they could use file recovery techniques to retrieve the previous tenants files [21]. To protect against these data recovery techniques, proper media sanitization or data encryption must be enforced [8, 23].

(2) **VM migration**: virtualization software provides the ability to move a running VM from one physical server to another. This feature enhances disaster recovery procedures as the running VMs could be moved, due to physical problems, or just to allow for maintenance on the underlying physical server. However, if this network is inadequately protected when the VM is migrated over the network, it is at risk of a man-in-the-middle attack, using tools like WireShark or XENsploit [11, 21, 24]. In addition, if the VM is moved to a different part of the network, its security policies must be applied to the security devices in that area, such as IDS and firewalls, if not the VM could be left vulnerable [24].

Weak Identity, Credentials, and Access Management. Traditionally web applications run on isolated networks commonly referred to as demilitarized zones (DMZ). These Internet-facing servers authenticate the remote users via accessing internal authentication servers using directory services (DS) or Lightweight Directory Access Protocol (LDAP) [25] to the cloud cannot rely on this method of authentication; it requires a scalable "identity access management system," which uses multifactor authentication. Author [26] suggests using federated identity management (FIM) to implement the management of identities (IDM). This security model creates a trust relationship between an identity provider (IDPs) and service providers (SPs), enabling remote organizations to trust their login credentials to cross security domains [27].

In a cloud environment, single sign-on (SSO) process allows the customers to access cloud based applications without having to log into the cloud environment. This login process is often described as single sign-on (SSO), accessing resources on a second security domain by using the trust relationship between the central identity provider and the cloud provider [27]. Combining digital signatures and single sign-on (SSO) allows the strongest authentication process suitable for a cloud environment to take place; the private key can be used to authenticate users or devices to guarantee mobility and therefore secure the elasticity in the cloud.

(1) **Encryption and key management (EKM)**: the protection of data in the cloud is now a joint responsibility between the cloud provider and customer; the chosen method to secure this data is to use encryption with public and private keys. The data must be protected 'at rest' on storage or backup media and while 'in transit' on the local network or Internet [14]. Once the data is encrypted, there must be processes in place to protect and manage the cryptographic keys; only authorized personnel should have access to these keys and good key rotation must be enforced [28]. If the keys are stolen, having a good key rotation policy will reduce the effective breach time by removing the compromised key from circulation quickly.

To summarize, it is inevitable that cloud computing will continue to grow and if consumers are encouraged to develop comprehensive security policies, they will benefit from the reduced cost of ownership and increased mobility of their applications [29]. The cloud providers are already offering encryption and key management to help protect customer data, along with the traditional network protection devices such as firewalls and intrusion detection systems (IDS) [30]. However, as more customers migrate their applications and data to the cloud, this will only increase a hacker's motivation to target this new frontier. The potential rewards of compromising a single vulnerability and gaining access to multiple customer's applications and data will be too difficult to resist.

Literature reviewed for this research explains the cloud services and deployment models and how they interact with the underpinning visualization technology [29, 30]. It describes the virtualization features, for example, live migration and quick servers deployment, which are some of the reasons why cloud computing is possible. Cloud has added a management layer, capable of separating and billing tenants, therefore allowing the cloud service providers to create an on demand service for their customers. This survey paper highlights the security concerns of the cloud and focuses on two particular issues, data breaches and weak authentication. For in-depth analysis, a small private cloud environment was built running CloudStack for the management software. Utilizing the designed cloud environment six vulnerabilities were attempted, three data breaches and a further three authentication exploits. These exercises demonstrated that traditional forensic and authentication exploits are still valid security concerns for the cloud providers.

2. Designing the Private Cloud Environment

In this section we shall design a private cloud environment to analyze the vulnerabilities described in Section 1 but within a safe and controlled environment. It would have been possible to demonstrate some of these vulnerabilities using just a pair of hypervisors but that would not be a fair comparison of a fully featured cloud environment. Using the stand-alone hypervisors would have omitted the management layer that cloud computing provides; this layer is responsible for the provisioning, multitenant separation, security controls, and of course the billing features of any on demand service. While designing the private cloud environment for this paper, there are a number of restrictions considered for the software selection process; these are as follows:

(1) The cloud management software must be open source and easy to install and manageable. This paper analyzes the concepts behind these vulnerabilities but does not focus on the design concepts of a corporate cloud.

(2) The visualization software must be capable of running on small and medium enterprises (SME) servers and yet be as close to the hypervisors running in any of large cloud environments, like Amazon web services (AWS) and RackSpace.

(3) The solution must be capable of supporting network-attached storage (NAS) technology for the shared storage, required to perform the moving of VMs between physical servers; this was due to cost constraints of the high-end storage solutions on the market.

(4) Virtual local area network (VLAN) technology would be used to segregate network traffic rather than software defined networking (SDN) solutions that are too expensive for a small lab environment.

Virtualized data centres have been around for a number of years especially for small educational or midrange industrial setup. This has introduced the ability for multitenant and billing of services, allowing companies like Amazon and RackSpace to build huge shared data centres for small to enterprise-sized companies to use on a pay-as-you-go basis. Reviewing the available cloud management suites in the open-source arena has identified two management platforms capable of creating a private cloud; these are OpenStack and CloudStack. Both of these offer the required features to test vulnerabilities within this paper; however, CloudStack has a simpler installation and one central console to manage the environment, whereas OpenStack is complicated to install and has multiple management interfaces. Therefore, this private cloud environment will use the CloudStack management suite. The CloudStack management software supports all the major hypervisors; reviewing hypervisors in the literary review identified that the Xen-Server hypervisor is used in Amazons Ec2 cloud environment and, as they are the biggest cloud provider, this hypervisor has been selected for this lab environment [28].

2.1. CloudStack. The lab environment will deploy a basic setup of the CloudStack product; this setup will simulate a cloud environment capable of proving the vulnerabilities in this paper. The CloudStack suite has a hierarchical management structure, allowing the product to manage thousands of physical hypervisors, across multiple time zones and locations. Figure 4 shows the basic view of this hierarchy with the remaining part of this section giving a detailed overview of each block.

2.1.1. CloudStack Management Server. CloudStack management server sits on top of all the zones in the cloud; it provides the management and orchestration layers of the private cloud environment, responsible for the automation of tasks initiated by customer's requests from the self-service interface. If a user requests a new VM, the orchestration layer will build up a series of tasks and scripts to provision the VM and storage and assign the domain name service (DNS) name, Internet protocol (IP) addresses, and if required any firewall rules and intrusion detection system (IDS) requirements [31]. This basic CloudStack deployment has a single Unix server and MySQL database to store the management state of the cloud. In an enterprise cloud, these would be highly available servers and the database would be mirrored to a separate location for resilience; none of this is required for this basic lab environment. The management interface for both users and administrators is through a web application, shown in Figure 5. This easy to use interface allows the administrators to build the cloud without needing to use the individual tools of each of the hypervisors manufacturers, providing a standard interface regardless of which hypervisor is chosen for the underlying virtualized environment. The same console

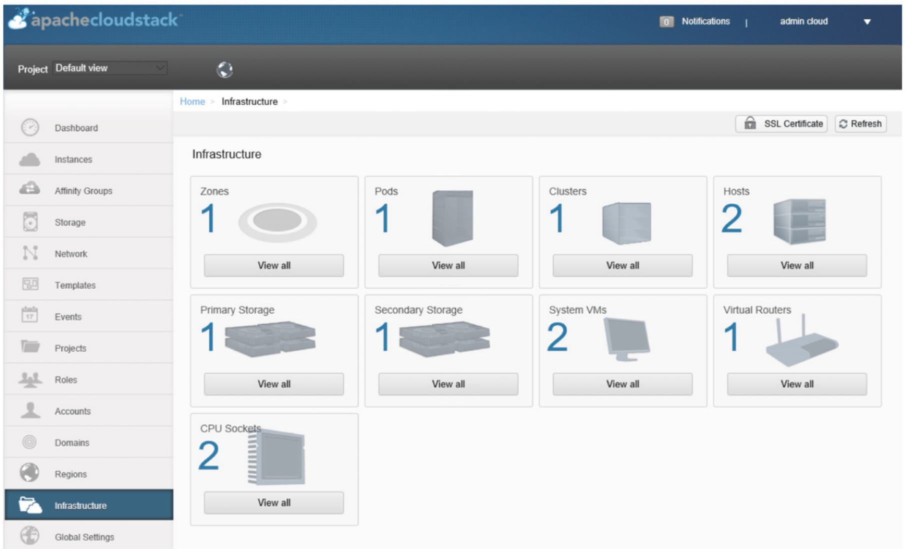


Figure 5: Infrastructure overview of the private cloud environment.

is also the interface for users of the cloud, so that they can provision resources and perform management tasks on their virtual hosts, like snapshots or starting and stopping the VMs.

2.1.2. Zone. The highest part of the hierarchy is the zone; it typically represents a full data centre or geographical region where the equipment is located. Physical hosts in one zone cannot communicate with hosts in others zones by default. When configuring CloudStack for the first time the zone has two options, basic or advanced configuration. A basic setup has been chosen for the lab environment as shown in Figure 6(a).

2.1.3. Pods. Pods further divide the zones into smaller manageable entities; if the zone is an entire data centre, then a pod could be one floor of that data centre or one rack in a smaller zone. All hosts and storages within a pod are provisioned within the same network address range or subnet as shown in Figure 6(b). When a customer chooses to build a new VM, their only choice is the zone they would like their server to be located in; they have no say as to which pod or cluster it will be built in.

2.1.4. Clusters. This is the smallest entity where physical hypervisors are located; a unique cluster can only have the same deployment of hypervisors as shown in Figure 6(c). For example, they must be either all XenServers or all VMware servers and you cannot have both XenServers and VMware servers in the same cluster. Hypervisors within a cluster are accessible to other hypervisors in the same cluster, with the VM migration function supported at this level. This is where the virtual machine can be moved between any hypervisor in the cluster using the live migration feature.

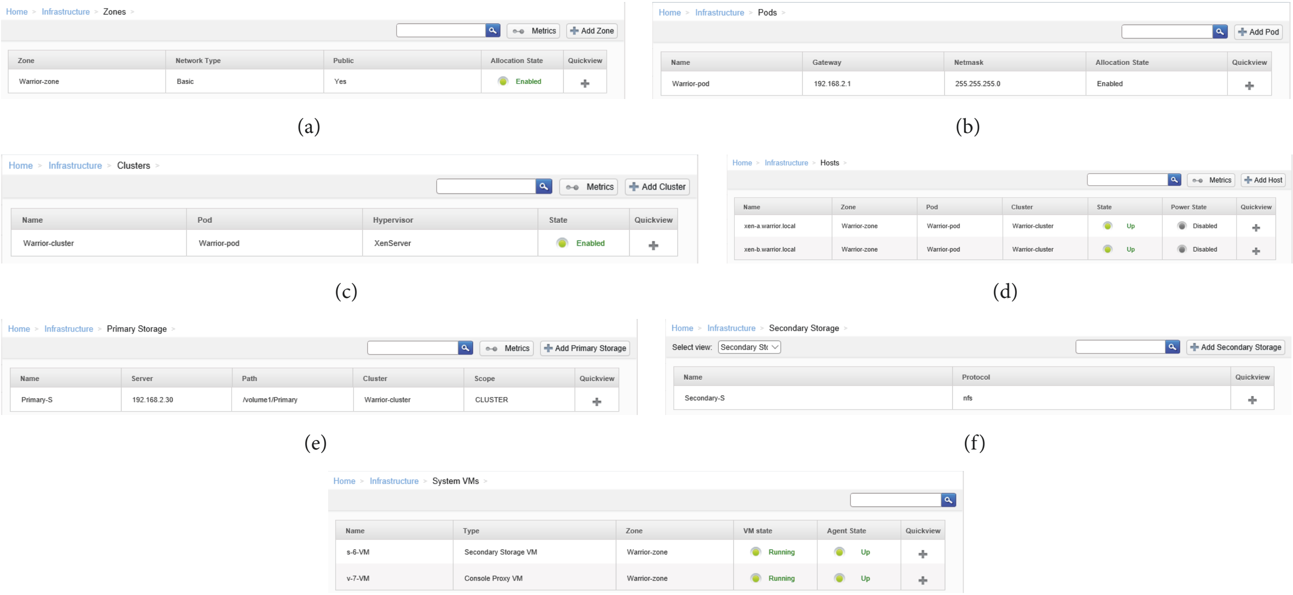
2.1.5. Hypervisor Hosts. Citrix XenServer is the leading open-source virtualization hypervisor used to power some of the largest cloud environments in the world, including Amazon Ec2. It is available in two editions, standard and enterprise. For the purpose of this paper, the standard edition provides all the features required to build the cloud-virtualized environment. Two XenServers with the specification in Table 1 underpin the virtualized environment as shown in Figure 6(d).

Table 1: Hardware specification of each hypervisors.

CPU	Intel Xeon CPU-E3
Memory	PC3-24000-24Gb
HDD	512 GB Serial ATA III
NIC	Intel 219-Gigabit Ethernet

2.1.6. Primary Storage. Each cluster must have at least one primary storage device; this storage is where the cloud will store the guest machines virtual disk and it must be accessible from all hypervisors in the cluster. For an enterprise cloud, this would be provisioned on fast I/O storage but, for the lab environment, it used a slower NFS (network files system) storage device. With all the guest operating system disks and data disks being stored on this NFS volume, it becomes a prime target. If the data is not sufficiently protected, using encryption techniques, anyone gaining access to this data, when it is 'at rest' could expose the data. This storage was targeted in the vulnerability section of this paper because it is a place where data is 'at rest' and potentially vulnerable. The brief settings for the private cloud environment are given in Figure 6(e).

2.1.7. Secondary Storage. Each zone must have at least one secondary storage and all pods and hypervisors must have access to this storage. The brief settings for the private cloud


Home > Infrastructure > Zones >
Metrics
Add Zone

Zone	Network Type	Public	Allocation State	Quickview
Warrior-zone	Basic	Yes	● Enabled	+

(a)

Home > Infrastructure > Pods >
Add Pod

Name	Gateway	Netmask	Allocation State	Quickview
Warrior-pod	192.168.2.1	255.255.255.0	Enabled	+

(b)

Home > Infrastructure > Clusters >
Metrics
Add Cluster

Name	Pod	Hypervisor	State	Quickview
Warrior-cluster	Warrior-pod	XenServer	● Enabled	+

(c)

Home > Infrastructure > Hosts >
Metrics
Add Host

Name	Zone	Pod	Cluster	State	Power State	Quickview
xen-a.warrior.local	Warrior-zone	Warrior-pod	Warrior-cluster	● Up	● Disabled	+
xen-b.warrior.local	Warrior-zone	Warrior-pod	Warrior-cluster	● Up	● Disabled	+

(d)

Name	Server	Path	Cluster	Scope	Quickview
Primary-S	192.168.2.30	/volume1/Primary	Warrior-cluster	CLUSTER	+

(e)

Select view: Secondary Stc

Name	Protocol	Quickview
Secondary-S	nfs	+

(f)

Name	Type	Zone	VM state	Agent State	Quickview
s-6-VM	Secondary Storage VM	Warrior-zone	● Running	● Up	+
v-7-VM	Console Proxy VM	Warrior-zone	● Running	● Up	+

(g)

Figure 6: Private cloud environment settings for (a) zone, (b) pods, (c) clusters, (d) Hypervisor, (e) primary storage, (f) secondary storage, and (g) system VMs.

environment are given in Figure 6(f). One surprising feature of this storage is that it only supports NFS (Network files system) to map connections. The purpose of this storage is to store templates, ISO build images, and snapshots of the running VMs.

2.1.8. System VMs. A number of system virtual machines manage the orchestration layer of the CloudStack environment; these virtual machines perform all system tasks within the cloud. They are built from a standard system template with each having specific roles to play in the managing of resources. Figure 13 shows two system VMs one to manage the storage allocation and the other proxy to manage user requests. The brief settings for the private cloud environment are given in Figure 6(f).

3. Implementing the Vulnerabilities

As suggested in the previous Sections 1 and 2, this paper will focus on just two kinds of vulnerabilities: (a) data loss and (b) weak identity, xredentials, and access management. For each of the vulnerabilities, a simulated attack on the cloud lab environment will be attempted to demonstrate the exploitation of the various artefacts.

3.1. Data Loss Evaluation. In the following evaluation, we have investigated the following three data breaches:

(1) Using forensic data recovery tools to access a previous customer's data

(2) Accessing the virtual machines disk and loading this to an unsecure hypervisor

(3) Capturing network traffic as a virtual machine between hypervisors using XenMotion.

3.1.1. Forensic Data Recovery. To enable the cloud environment to offer its ondemand, pay-as-you-go service, it must have a shared storage environment that allows users to lease and use storage but also to release that storage when the environment is no longer required. With this being a shared storage environment, if the cloud environment does not sanitize the previously used storage before releasing it to the next tenant, it may become possible to utilize forensic tools to access data that the previous tenant stored. If this is possible it could provide a data loss of valuable assets [8, 23]. The cloud lab environment was configured with two hypervisors running XenServer 7.0 as shown in Figure 7; these two physical servers had network access to the shared primary volume where CloudStack holds the virtual machines virtual hard drive (VHD) files.

The principle behind this vulnerability is that if a malicious tenant was allocated storage on the same device where a previous tenant had been but had since left the cloud, would it be possible to use standard disk carving forensic tools to recover the previous tenant's data? There is a wealth of data forensic tools available, both commercial and open source but we have used the following tools for our investigations:

(1) **Scalpel**: an open-source program to recover deleted files

(2) **PhotoRec**: an open-source multifunctional program that can recover deleted files.

3.1.2. Virtual Hard Disk (VHD) Exploit. VHD is an image file that contains the file structure of a hard drive; its structure

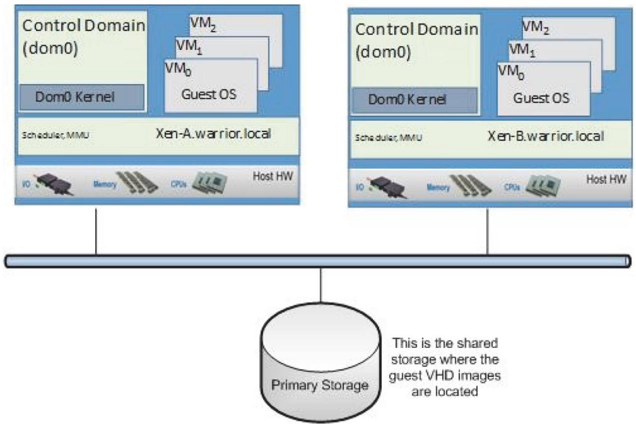


Figure 7: High-level view of storage and XenServers.

can replicate any part of a physical hard drive. The file can contain partitions that can hold directories and files and can have a bootable image. VHD images allow multiple operating systems to share the same physical machine and have introduced features like snapshots and guest migrations between physical hosts to provide additional resilience. However, the features that have made the VHD image file so popular have also brought with it vulnerabilities. Anyone with access to make a copy of the VHD has the ability to steal an entire machine; malicious insiders could remove this portable system image and load it onto an entirely different host. This could be done without the owner even knowing they had been compromised. Before copying any individual VHD images, the underlying format type of the image must be understood. VHD supports three kinds of implementation and they have an impact on how the image is transferred between hosts that are

(1) **Fixed hard disk image**: the simplest format, the VHD image, allocates its full storage allowance when created; for example, if a 20GB drive was required the VHD image would be created as a 20GB file [32].

(2) **Dynamic hard disk image**: as the name suggests, a dynamic VHD image is built to accommodate the data at the time of creation; therefore if a 20GB drive was requested but only 10Gb of data was initially loaded, a dynamic VHD file would allocate just over 10Gb and grow as more data was added [32].

(3) **Differencing hard disk image**: the differencing VHD image relies on a master or parent image and is only the difference from that image; a parent image could be associated with multiple differencing images [32].

In the cloud lab, the guest machine's image files are stored on the central primary storage device; gaining read-only access to this storage would allow the guest operating system to be removed from the cloud environment and examined offline to further compromise the image. For example, cracking the user's password would allow the malicious insider to go back to the live server and log in with valid user credentials.

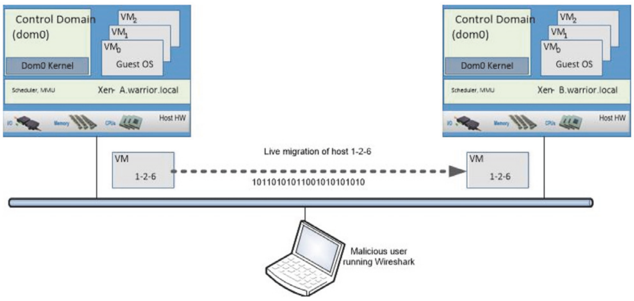


Figure 8: Man-in-the-middle attack using WireShark.

3.1.3. Virtual Machine Migration Exploit. With the growing demand for greater service availability, the ability to migrate a live virtual machine from one physical server to another has huge benefits; it can help move machines from a hypervisor to facilitate maintenance to that physical machine. Another advantage of moving live machines is when a local data centre is experiencing a serious failure or local disaster; in this scenario all VMs could be migrated to a second data centre, outside this particular area and away from the potential problem. This is used extensively in company's business contingency plans. A new concept of how to save energy costs in large data centres is virtualization power management; this uses the live migration features to consolidate virtual machines onto fewer hypervisors when peak demand drops [33]. Consolidating these VMs significantly reduces power consumption, which in turn has a financial benefit to the cloud providers. This exercise used the man-in-the-middle approach, where the attacker is able to listen and steal data as it passes between two end-points [23]. In the case of this test, two physical hypervisors transferred the control of a VM from one hypervisor to the other. Scanning the network via laptop running WireShark, as shown in Figure 8, is to verify if this intruder could capture anything of interest from the network trace.

3.2. Weak Identity, Credentials, and Access Management Evaluation. Traditionally data integrity and its confidentiality were the responsibility of the company who owned the assets. However, in a cloud environment there is a large number of unrelated organizations running on shared resources and managed by the cloud server provider's personnel. The customer might trust the cloud service provider (CSP) but it is very unlikely they would trust other customers on the shared platform. This is one of the concerns customers have when they move their digital assets to the cloud and a reason why some are reluctant to adopt this technology [8]. Weak passwords and poor authentication have been some of the reasons behind data breaches, the cloud service provider must have secure methods in place to audit and manage this safely. No one individual or area should have full control of a customer's assets; permissions should be divided into job roles and, as personnel move departments or leave the service provider, those roles should be removed. This exercise attempted to compromise the administrator account of the

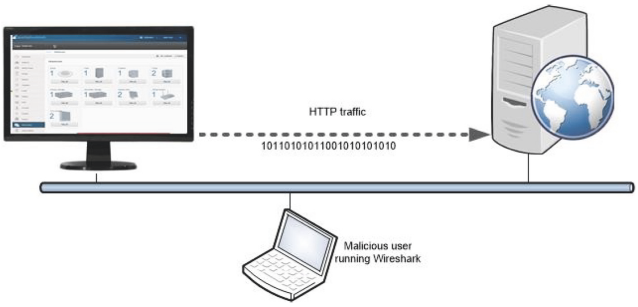


Figure 9: HTTP capture and compromise.

lab environment; this user has full control over the resources, which means that once compromised the malicious user can access customer data or destroy the machine.

In the following evaluation, we have investigated the following three methods:

(1) Man-in-middle attack using WireShark: the network data was captured while the administrator was logging into the cloud.

(2) Man-in-middle attack using Burp-suite proxy: this targeted stealing the php session ID.

(3) Brute force password crack, to identify if the CloudStack interface disables access after a number of login attempts.

3.2.1. Network Capture to Compromise Accounts. During the installation of CloudStack, it was noted that the web-interface was configured to use HTTP (Hyper Text Transfer protocol) as its default setting as in Figure 9. This protocol was developed in 1990 and is used to send and receive data between the client's browser and the web-server. At that time, this protocol revolutionized the Internet, allowing a business to interact with their clients 24 hours a day, 7 days a week. However, as the Internet has grown and financial transactions increased, this protocol has proved to be insecure because it transfers all packets in clear text and is frequently targeted by hackers. In the lab environment, the network switch was configured to SPAN (switched port analyzer) all traffic from the client PC and web-server to the malicious laptop. This laptop then ran the WireShark network analyzer to capture all traffic between these end-points.

3.2.2. Session Hacking. The HTTP protocol design was to be connectionless; this means that you can log into a website and do other tasks and, as long as the idle timer of the site has not expired, you pick off exactly where you left off. When it comes to authenticating with a server, instead of having to login for every page, developers have created cookies to allow the application to track the session throughout its connection. When you authenticate with a server it sends a cookie to the client as proof this was successful and, each time the client browser requests information from the server, it supplies this cookie information. The problem arises when the communication between the client and server is compromised; as previously stated when using HTTP, anything sent between client and server is in clear text and therefore can be compromised. If the cookie information, which includes the authenticated session id, can be captured by some means, a second session providing the correct cookie information can easily impersonate the first initial connection, stealing their session. The exercise in next section will demonstrate that an intercepting proxy server is capable of seeing and therefore stealing this information. If a malicious user were to obtain this information they could easily jump into someone else web session.

3.2.3. Brute Force Password Attack on Administrator Account. The following common vulnerabilities in web applications can lead to an intruder being able to crack the users password [34]:

(1) Allowing unlimited attempts to login to the password

(2) Automatic reset of account after a period of time, allowing the user to retry

(3) Allowing simultaneous login attempts

(4) Using HTTP either during the login process or for the entire session.

Traditionally, HTTP was used as the default protocol and the username and password were seen using WireShark. However, in order to expose this vulnerability, this required a physical connection to the local network and WireShark, to gather the network packets. Further examination of the CloudStack application highlighted that failed login attempts do not lock out the user account; this is very poor practice. In general, accounts should be locked if there has been more than five failed login attempts; without this limit, malicious users could try an endless amount of times to guess the users password. Knowing that the user accounts are not locked after multiple failed attempts opens the door to using one of the many brute force password hacking tools. One of the best is THC-Hydra, a very efficient password cracker supporting 50 different protocols, including web-forms. The exercise in Section 5 attempted to prove this vulnerability; it used Burp-suite to capture the parameters of the web application and armed with the information THC-Hydra sent a list of passwords to the web-interface.

4. Analysis and Discussions

As a first step, we investigated the data loss to demonstrate that it is possible to compromise the security of a cloud environment.

4.1. Forensic Data Recovery. The CloudStack management suite requires a shared storage environment to store the various guest images, template, and ISO images. This storage is split into primary and secondary volumes; the primary volume is where the guest VHD image files are stored and, therefore, where the customer's sensitive and personal data is held. The forensic recover was performed against this primary

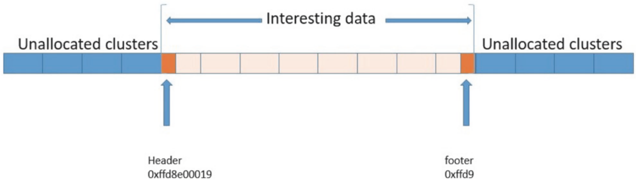


Figure 10: Signature of the file under investigation.

volume. This exercise will utilize forensic data carving tools, to recover data a previous tenant has left behind when they deleted their virtual machine. The lab environment was built using brand new storage so there was no previous data to recover. Therefore, the first stage was to create 20 new virtual machines each with a 50Gb disk, each volume to hold the operating system, and a large amount of jpg files to fill the partition. Once the storage was full, all the VMs were then deleted, freeing up the storage and leaving old data that potentially could be recovered. The deleted files no longer had any security controls stopping access, so forensic tools should be able to recover and access these files without any security tools detecting the data breach. The next part was to create another VM containing a 50Gb operating system volume but also a 500Gb data volume, which was entirely empty. This second volume was half the size of the primary storage and therefore was reusing storage where deleted VMs would have been allocated. This forensic technique is to extract files based on their content and not the metadata of the file. The Scalpel tool looks for files headers and footers; once a file signature is located, it carves out all the data between these points and creates a new file with this data. Figure 10 shows the headers and footer of a jpg file; the header is identified by the hex value *ffd8e00019* and the footer is identified by the *ffd9*. If this was a valid jpg file, the contents between these two markers could be carved out to recreate the deleted file.

Scalpel was run on the empty data volume but it failed to locate any jpg files on the storage under test; this was a strange result given the amount of jpg files that had been copied to the primary storage volume. With Scalpel, as in Figure 11, failing to locate any jpg files the evaluation moved onto the second file carving tool named as PhotoRec. This tool is another open-source application which as the name suggests specializes in recovering lost pictures. Unlike Scalpel, PhotoRec does need to know what the previous file format was, for example, NTFS or Ext4. This posed a very interesting query, the virtual machines were all Linux operating systems with Ext4 formatted partitions but these partitions were encapsulated within a virtual hard drive (VHD). Running PhotoRec on the data partition failed to find any files; this was because it was looking for an Ext4 file system and one did not exist in the standard form; it was hidden within the VHD image.

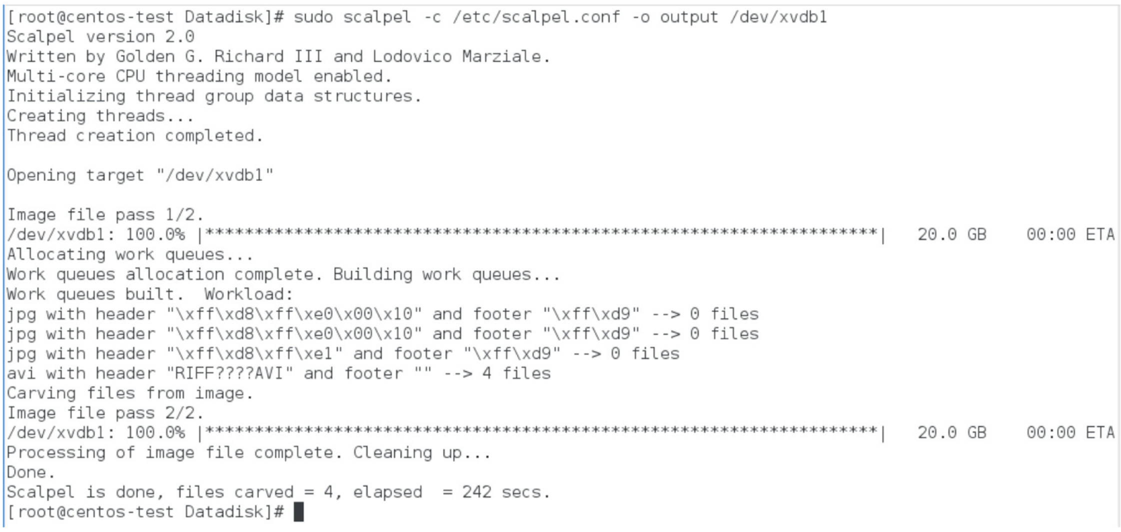
Both of these tools look for file signature to locate the deleted files and carve out the data; further investigation was needed to find out why both these tools had failed. The next step was to use the Hexedit application as in Figure 12; this tool is capable of viewing the hex contents of a disk partition. Running this on the 500Gb data volume revealed that the whole disk had been overwritten with zeros, which is why none of the data carving tools could find any deleted data.

The virtual hard drive format was a creation of Microsoft and has been licensed to the major virtualization companies and adopted as a standard file format for visualized machines. Researching information on this file format uncovered that creating fixed-sized VHDs requires a fair amount of time because every sector of the allocated file has to be written with zeros to ensure no data is "pirated" from a previously existing file [35]. Both [23, 36] had suggested in their studies that if data was deleted from a previous tenant and the cloud provider did not have a rigorous data cleansing process, it would be possible to retrieve that data. This was not proved to be correct in this study, because Microsoft designed the VHD image format understanding that this could have been a problem when storage was reused. If other virtual disk formats were used, it would be worth repeating this exercise to make sure they too follow this very good design practice.

4.2. Virtual Disk Exploit. If virtualization is the underpinning technology of the cloud, then the virtual disk is the underlying success of virtualization. The virtual disk is a file, which can be partitioned and formatted just as a physical hard drive; it is transportable which allows it to be migrated from one physical host to another. This portability also means that if someone were to gain access to the virtual image file, they would be able to steal the entire system, including passwords, security policies, and data. In a cloud environment, if a malicious insider could gain access to the primary storage where the VHD files are stored and take copies of these images, they would, effectively, steal the servers in their entirety. To demonstrate this vulnerability, a new virtual machine was created on the CloudStack lab; this had a customer host name of test-VHD and a cloud internal name of i-2-6-VM (Figure 13). The reason for this internal name is that the cloud must have unique names for each VM and storage volume across the entire cloud, the only way to guarantee this is for the cloud software to generate these unique names.

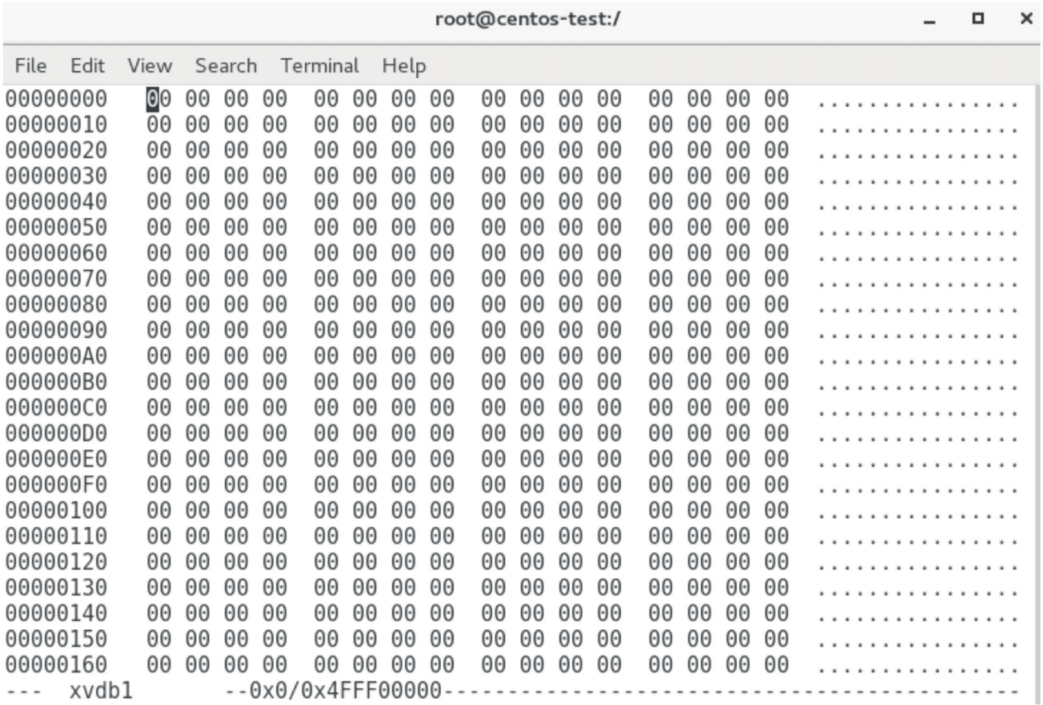
As in Figure 13, the view within CloudStack of the newly created host shows both the customer chosen name and the generated internal name. To simulate how this vulnerability could take place, a third XenServer was added to the network; however, this hypervisor was outside the control of the CloudStack environment. Figure 14 shows Xen-a & Xen-b hypervisors belonging to a pool called Warrior; these hypervisors were previously shown in Figure 10 as the physical hosts of the CloudStack cloud. The Warrior pool shows a number of running VMs on both these hosts, 'i-2-6' is the virtual machine targeted by this investigation.

Logging onto the new hypervisor Xen-C and mounting the primary storage device allow the intruder to view all the virtual disks of the running VM's; these files are shown in Figure 15. However, looking at the directory listing, it is not clear which VHD belongs to the target host. In a real cloud environment this storage would contain in excess thousands of VHD images; in this targeted attack, more investigation is required to identify the correct VHD to hack.



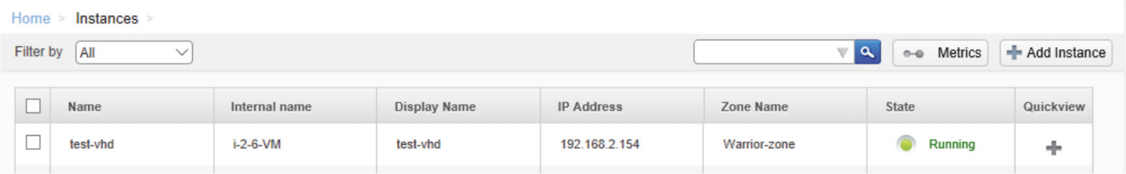
```
[root@centos-test Datadisk]# sudo scalpel -c /etc/scalpel.conf -o output /dev/xvdb1
Scalpel version 2.0
Written by Golden G. Richard III and Lodovico Marziale.
Multi-core CPU threading model enabled.
Initializing thread group data structures.
Creating threads...
Thread creation completed.

Opening target "/dev/xvdb1"

Image file pass 1/2.
/dev/xvdb1: 100.0% |********************************************************************|   20.0 GB    00:00 ETA
Allocating work queues...
Work queues allocation complete. Building work queues...
Work queues built.  Workload:
jpg with header "\xff\xd8\xff\xe0\x00\x10" and footer "\xff\xd9" --> 0 files
jpg with header "\xff\xd8\xff\xe0\x00\x10" and footer "\xff\xd9" --> 0 files
jpg with header "\xff\xd8\xff\xe1" and footer "\xff\xd9" --> 0 files
avi with header "RIFF????AVI" and footer "" --> 4 files
Carving files from image.
Image file pass 2/2.
/dev/xvdb1: 100.0% |********************************************************************|   20.0 GB    00:00 ETA
Processing of image file complete. Cleaning up...
Done.
Scalpel is done, files carved = 4, elapsed  = 242 secs.
[root@centos-test Datadisk]#
```

Figure 11: Output from the Scalpel tool.



```
00000000   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  ................
00000010   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  ................
00000020   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  ................
00000030   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  ................
00000040   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  ................
00000050   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  ................
00000060   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  ................
00000070   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  ................
00000080   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  ................
00000090   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  ................
000000A0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  ................
000000B0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  ................
000000C0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  ................
000000D0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  ................
000000E0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  ................
000000F0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  ................
00000100   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  ................
00000110   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  ................
00000120   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  ................
00000130   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  ................
00000140   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  ................
00000150   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  ................
00000160   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  ................
---  xvdb1       --0x0/0x4FFF00000-----------------------------------------
```

Figure 12: Output from the Hexedit tool.



□	Name	Internal name	Display Name	IP Address	Zone Name	State	Quickview
□	test-vhd	i-2-6-VM	test-vhd	192.168.2.154	Warrior-zone	● Running	+

Figure 13: New target VM.

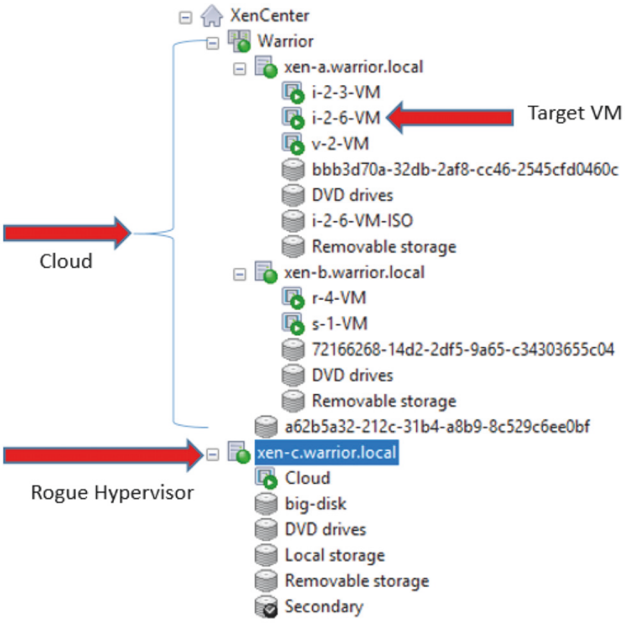
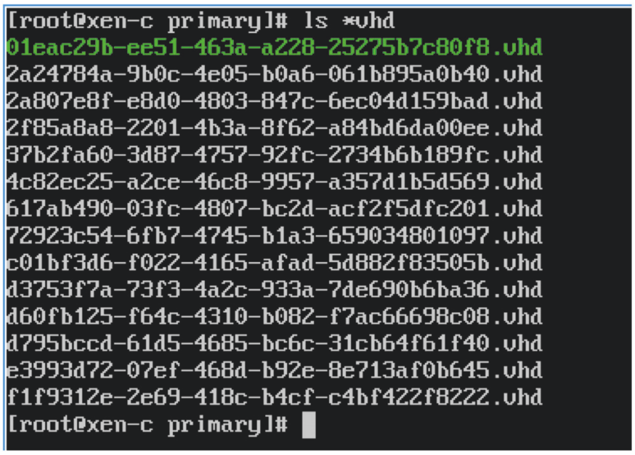


Figure 14: XenCentre view and settings.



```
[root@xen-c primary]# ls *vhd
01eac29b-ee51-463a-a228-25275b7c80f8.vhd
2a24784a-9b0c-4e05-b0a6-061b895a0b40.vhd
2a807e8f-e8d0-4803-847c-6ec04d159bad.vhd
2f85a8a8-2201-4b3a-8f62-a84bd6da00ee.vhd
37b2fa60-3d87-4757-92fc-2734b6b189fc.vhd
4c82ec25-a2ce-46c8-9957-a357d1b5d569.vhd
617ab490-03fc-4807-bc2d-acf2f5dfc201.vhd
72923c54-6fb7-4745-b1a3-659034801097.vhd
c01bf3d6-f022-4165-afad-5d882f83505b.vhd
d3753f7a-73f3-4a2c-933a-7de690b6ba36.vhd
d60fb125-f64c-4310-b082-f7ac66698c08.vhd
d795bccd-61d5-4685-bc6c-31cb64f61f40.vhd
e3993d72-07ef-468d-b92e-8e713af0b645.vhd
f1f9312e-2e69-418c-b4cf-c4bf422f8222.vhd
[root@xen-c primary]# █
```

Figure 15: View of the primary storage containing all the virtual hard drives.

Using the cloud interface and examining the volume further shows a volume id of *32098385-7aa9-45cd-9caa-9f5dbd140fa1*; comparing this volume ID in Figure 16 to the VHD files in Figure 15 does not show a match, suggesting that this information is hidden from even the privileged cloud user.

It would be impossible for CloudStack to function without it having some mechanism to associate the VM with its storage device. Therefore, this information is not available in the graphical interface; it must reside within the cloud database. The CloudStack database structure is not too complicated and, by gaining access to the database command line interface, the following MySQL commands soon identified which tables were required to find the VHD image associated with the target host. The "show" table command, in Figure 17, identified two tables of interest; these were the volume and instance tables. A further MySQL query identified the structure of both of these tables.

Armed with the database and table structure, a further and more specific MySQL query was run on the database to map the UUID value from Figure 16 to the actual VHD file listed in Figure 15.

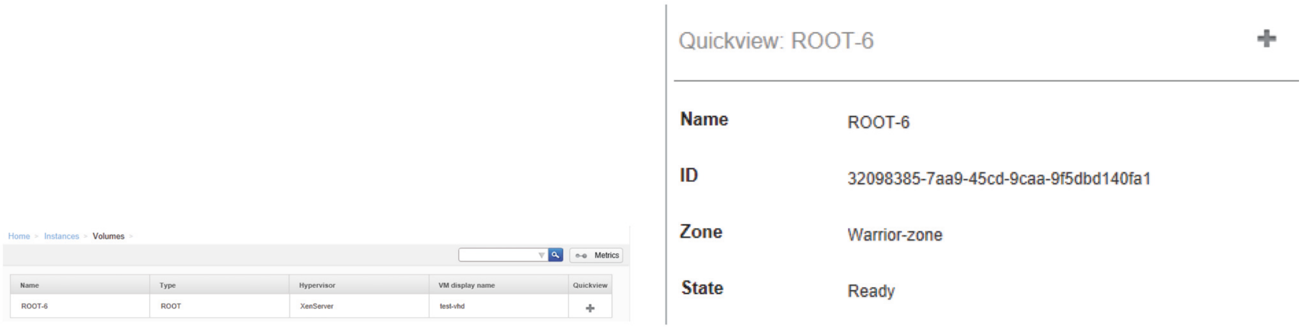
To avoid detection and corrupting the customers image, the first stage is to take a copy of this file to an area on our rogue hypervisor, where we can run a series of commands to determine its format. With the target VHD image identified in Figure 18, listed in the path column, the next stage is to identify what kind of implementation was selected when the VHD image was deployed. Section 3.1.2 listed three possible formats to the VHD image and depending on which format, it could be possible that more than just one image is needed to steal the entire server.

Because the VHD image is a file, a number of unsupported utilities have been developed that allow these files to be viewed and even updated without the involvement of a hypervisor. One such utility is 'VHD-util'; it is installed as standard on XenServer 7.0; this tool will be used to read the file format of our VHD image. As highlighted in previous section, this means that it must have a parent VHD file and this will be required to be able to copy a valid image of the target host.

Looking back at Figure 15, this parent image is listed; however it could be the master for a number of the VM's running in the cloud. Therefore, before running commands against it, a copy was taken and stored on Xen-C; this allowed the image to be examined without compromising any of the running guests. With both images in a safe area on Xen-C, 'VHD-util' was used to scan both images to make sure there were no other files in the chain making up the complete server.

Figure 19 confirms that the first image identified has a parent image, confirming it is using the differencing file format and its parent does not have any more images in the chain. At this point, either of the disks could be loaded onto the rogue XenServer (Figure 20); to be able to do this they needed to be merged together to create one complete image file. The same utility 'vhd-util provides this feature, to coalesce the two images. With the image now merged into one VDH file, we can use the standard import utility on the rogue XenServer to load up the stolen guest.

With a copy of the virtual machine running on a compromised hypervisor [36], the malicious insider has a number of options. They can simply blank out the 'root' password by booting into maintenance mode and following standard procedures listed on the Internet helping administrators who have forgotten their root password. Once they have root access to the customers host, they could simply access the data on the server. But they could decide it would be more rewarding to crack all the user's passwords on the system by using the tools like John-the-Ripper password cracker. This would allow the insider to login to the live machine undetected, as they would have a legitimate username and password; this may then allow further access into the customer's network. Protecting against this kind



Home > Instances > Volumes >

Name	Type	Hypervisor	VM display name	Quickview
ROOT-6	ROOT	XenServer	test-vhd	+

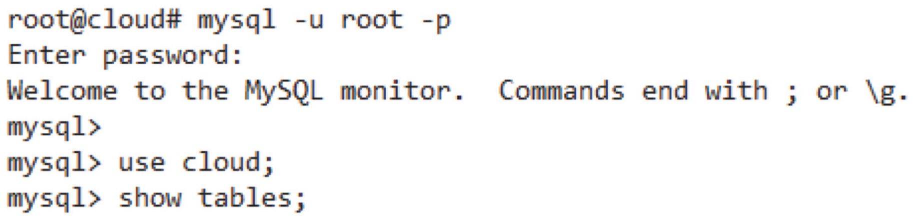
Quickview: ROOT-6

Name	ROOT-6
ID	32098385-7aa9-45cd-9caa-9f5dbd140fa1
Zone	Warrior-zone
State	Ready

(a)

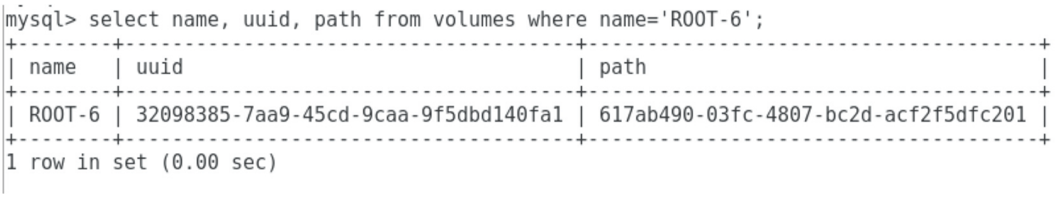
(b)

Figure 16: (a) and (b) A detailed view of the ROOT-6 volume.



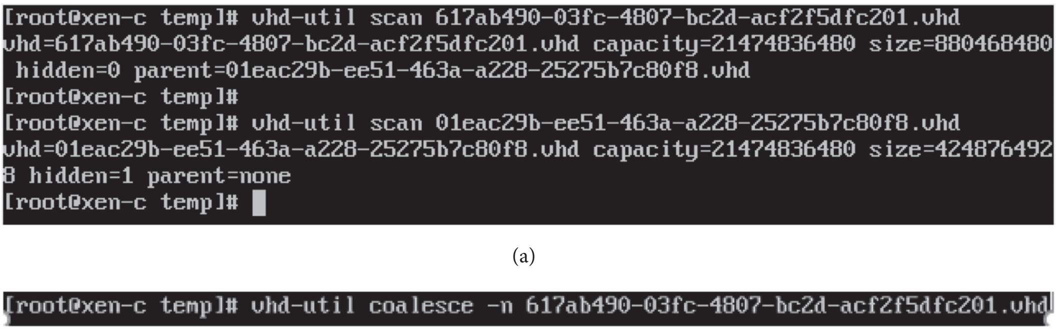
```
root@cloud# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
mysql>
mysql> use cloud;
mysql> show tables;
```

Figure 17: The MySQL commands for showing tables of interest.



```
mysql> select name, uuid, path from volumes where name='ROOT-6';
+--------+--------------------------------------+--------------------------------------+
| name   | uuid                                 | path                                 |
+--------+--------------------------------------+--------------------------------------+
| ROOT-6 | 32098385-7aa9-45cd-9caa-9f5dbd140fa1 | 617ab490-03fc-4807-bc2d-acf2f5dfc201 |
+--------+--------------------------------------+--------------------------------------+
1 row in set (0.00 sec)
```

Figure 18: Mapping UUID to VHD image.



```
[root@xen-c temp]# vhd-util scan 617ab490-03fc-4807-bc2d-acf2f5dfc201.vhd
vhd=617ab490-03fc-4807-bc2d-acf2f5dfc201.vhd capacity=21474836480 size=880468480
 hidden=0 parent=01eac29b-ee51-463a-a228-25275b7c80f8.vhd
[root@xen-c temp]#
[root@xen-c temp]# vhd-util scan 01eac29b-ee51-463a-a228-25275b7c80f8.vhd
vhd=01eac29b-ee51-463a-a228-25275b7c80f8.vhd capacity=21474836480 size=424876492
8 hidden=1 parent=none
[root@xen-c temp]#
```

(a)

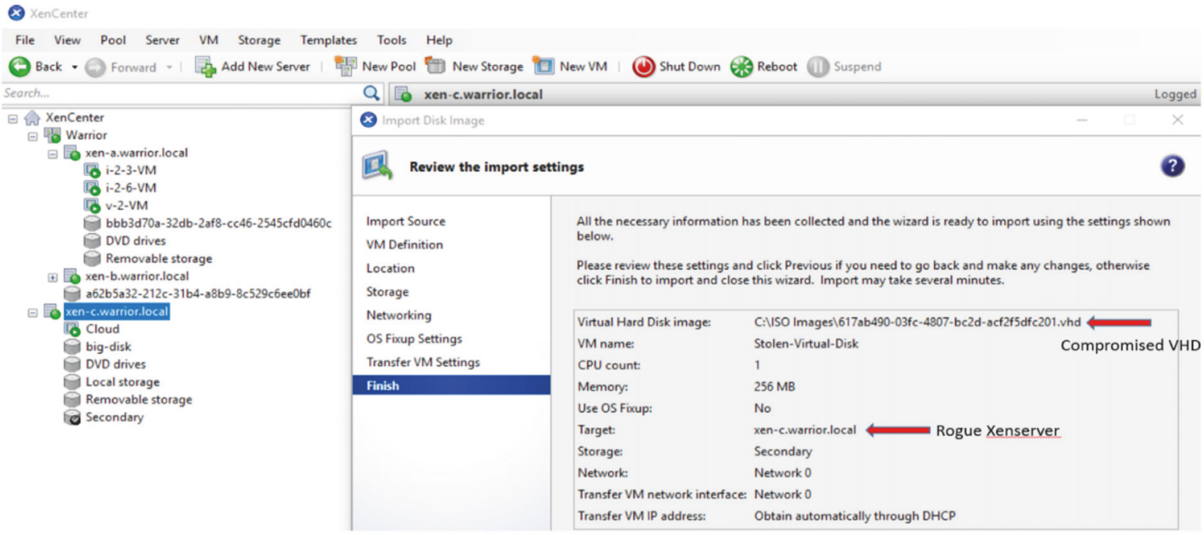
```
[root@xen-c temp]# vhd-util coalesce -n 617ab490-03fc-4807-bc2d-acf2f5dfc201.vhd
```

(b)

Figure 19: (a) Screenshot of the scan of vhd-util and (b) coalesce both images into one image.

of attack is easy if the data ‘at rest’ was encrypted. For example, if the VHD images held on the primary storage were encrypted, copying these images to an alternative hypervisor would be useless without the encryption key; even the tools used in this exercise would not be able to compromise the encryption.

4.3. Virtual Machine Live Migration Exploit. Live migration is the process of moving a running virtual machine (VM) from one hypervisor to another; this process has three phases: precopy, stop-copy, and resume phase [37]. This exploit was interested in the precopy phase; this is the phase which sends the content of the VM over the network to the target



Review the import settings

All the necessary information has been collected and the wizard is ready to import using the settings shown below.

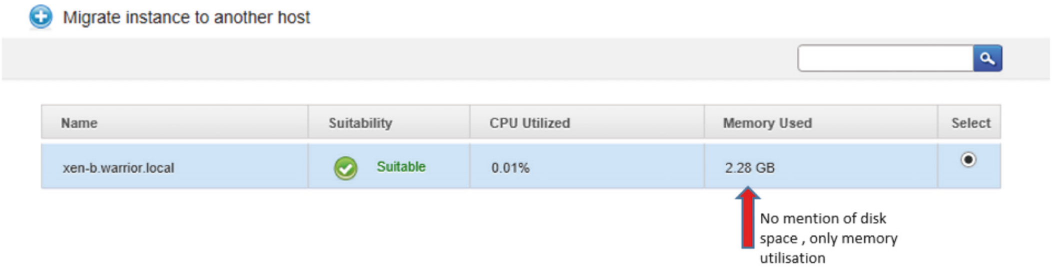
Please review these settings and click Previous if you need to go back and make any changes, otherwise click Finish to import and close this wizard. Import may take several minutes.

Virtual Hard Disk image:	C:\ISO Images\617ab490-03fc-4807-bc2d-acf2f5dfc201.vhd
VM name:	Stolen-Virtual-Disk
CPU count:	1
Memory:	256 MB
Use OS Fixup:	No
Target:	xen-c.warrior.local
Storage:	Secondary
Network:	Network 0
Transfer VM network interface:	Network 0
Transfer VM IP address:	Obtain automatically through DHCP

Compromised VHD

Rogue Xenserver

Figure 20: Compromised VHD file.



Name	Suitability	CPU Utilized	Memory Used	Select
xen-b.warrior.local	✓ Suitable	0.01%	2.28 GB	●

No mention of disk space , only memory utilisation

Figure 21: Migrate instance.

hypervisor. When starting this paper, the understanding of this process was that the full disk contents of the VM would be transferred over the network. On reflection, this would never have been the case, as it would take far too long to migrate a virtual machine, if all the data had to be transferred. CloudStack uses its primary storage to store the virtual machine images; all hypervisors in the same pod require access to this shared storage. When a machine is migrated from one hypervisor to another, the only data transferred is control data, making for a quick and smooth transfer. To verify the theory that only the VM's memory and storage control transferred over the network, an initial test was performed to move the VM between the two hypervisors.

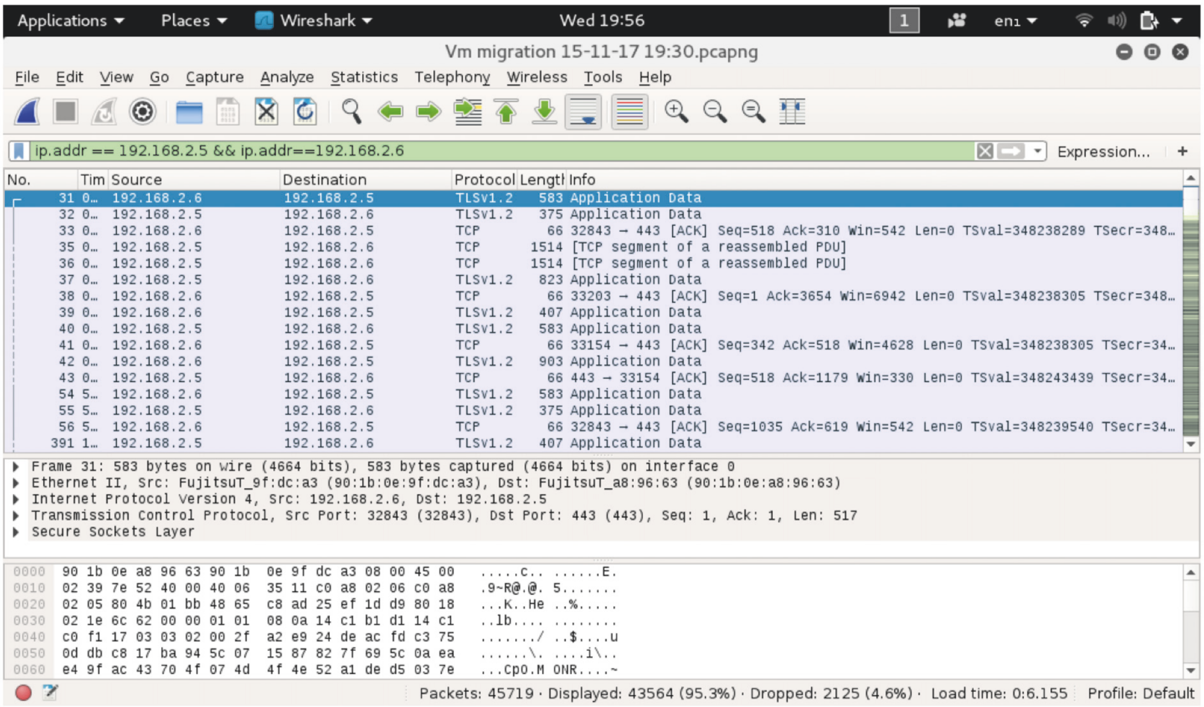
Figure 21 makes no mention of storage and only indicates the memory size, which does suggest that [37] was correct stating that the precopy only transfers memory. If further proof were required, the full migration of the virtual machine (VM) took just 15 seconds, confirming that the storage data could not have transferred over the network in that timescale. This contradicts [8] who suggest a VM's data is exposed to the network during this phase and could lead to privacy and data integrity issues.

The original 'man-in-the-middle' attack, had hoped to capture files using WireShark but, as stated, the file systems are not transferred over the network. Therefore, to complete the exploit and to identify if any other interesting traffic is exposed during the live migration, the network switch was configured to enable port mirroring, or SPAN (Switched Port Analyzer) which is a way of monitoring traffic on a switched network. Three network ports were mirrored (Xen-A, Xen-B, and the primary storage) to the port where the WireShark laptop was connected.

The output from Figure 22 is what is expected from a corporate product like XenServer; all the communication between the two hosts are encrypted, so although it may be possible to disrupt this transfer it is certainly not a trivial task to intercept data. On a real cloud environment, it would be harder to install and configure a network analyzer, so this test failed to capture private data during a VM migration.

The following analysis demonstrates a number of methods to compromise user accounts or browser sessions to gain full control over the cloud environment.

4.4. Network Capture to Compromise Accounts. As stated in the previous section, the HTPP protocol is very insecure and should never be used for sites containing sensitive data. Surprisingly, the default installation of CloudStack configures the Apache web-server to use HTTP and with the administrator using this web application to perform live migrations; any


Applications
Places
Wireshark
Wed 19:56
1
en1
Vm migration 15-11-17 19:30.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Expression...
+

```
ip.addr == 192.168.2.5 && ip.addr==192.168.2.6
```

```
No.    Tim Source          Destination     Protocol Lengtl Info
   31 0… 192.168.2.6     192.168.2.5     TLSv1.2   583 Application Data
   32 0… 192.168.2.5     192.168.2.6     TLSv1.2   375 Application Data
   33 0… 192.168.2.6     192.168.2.5     TCP        66 32843 → 443 [ACK] Seq=518 Ack=310 Win=542 Len=0 TSval=348238289 TSecr=348…
   35 0… 192.168.2.5     192.168.2.6     TCP      1514 [TCP segment of a reassembled PDU]
   36 0… 192.168.2.5     192.168.2.6     TCP      1514 [TCP segment of a reassembled PDU]
   37 0… 192.168.2.5     192.168.2.6     TLSv1.2   823 Application Data
   38 0… 192.168.2.6     192.168.2.5     TCP        66 33203 → 443 [ACK] Seq=1 Ack=3654 Win=6942 Len=0 TSval=348238305 TSecr=348…
   39 0… 192.168.2.6     192.168.2.5     TLSv1.2   407 Application Data
   40 0… 192.168.2.5     192.168.2.6     TLSv1.2   583 Application Data
   41 0… 192.168.2.6     192.168.2.5     TCP        66 33154 → 443 [ACK] Seq=342 Ack=518 Win=4628 Len=0 TSval=348238305 TSecr=34…
   42 0… 192.168.2.6     192.168.2.5     TLSv1.2   903 Application Data
   43 0… 192.168.2.5     192.168.2.6     TCP        66 443 → 33154 [ACK] Seq=518 Ack=1179 Win=330 Len=0 TSval=348243439 TSecr=34…
   54 5… 192.168.2.6     192.168.2.5     TLSv1.2   583 Application Data
   55 5… 192.168.2.5     192.168.2.6     TLSv1.2   375 Application Data
   56 5… 192.168.2.6     192.168.2.5     TCP        66 32843 → 443 [ACK] Seq=1035 Ack=619 Win=542 Len=0 TSval=348239540 TSecr=34…
  391 1… 192.168.2.5     192.168.2.6     TLSv1.2   407 Application Data
```

```
▶ Frame 31: 583 bytes on wire (4664 bits), 583 bytes captured (4664 bits) on interface 0
▶ Ethernet II, Src: FujitsuT_9f:dc:a3 (90:1b:0e:9f:dc:a3), Dst: FujitsuT_a8:96:63 (90:1b:0e:a8:96:63)
▶ Internet Protocol Version 4, Src: 192.168.2.6, Dst: 192.168.2.5
▶ Transmission Control Protocol, Src Port: 32843 (32843), Dst Port: 443 (443), Seq: 1, Ack: 1, Len: 517
▶ Secure Sockets Layer
```

```
0000  90 1b 0e a8 96 63 90 1b  0e 9f dc a3 08 00 45 00   .....c.. ......E.
0010  02 39 7e 52 40 00 40 06  35 11 c0 a8 02 06 c0 a8   .9~R@.@. 5.......
0020  02 05 80 4b 01 bb 48 65  c8 ad 25 ef 1d d9 80 18   ...K..He ..%.....
0030  02 1e 6c 62 00 00 01 01  08 0a 14 c1 b1 d1 14 c1   ..lb.... ........
0040  c0 f1 17 03 03 02 00 2f  a2 e9 24 de ac fd c3 75   ......./ ..$....u
0050  0d db c8 17 ba 94 5c 07  15 87 82 7f 69 5c 0a ea   ......\. ....i\..
0060  e4 9f ac 43 70 4f 07 4d  4f 4e 52 a1 de d5 03 7e   ...CpO.M ONR....~
```

```
Packets: 45719 · Displayed: 43564 (95.3%) · Dropped: 2125 (4.6%) · Load time: 0:6.155   Profile: Default
```

Figure 22: WireShark capture.

malicious user capable of capturing the communication from the client browser to the web-server would have the ability to obtain the full permissions of this privileged user. This exercise installed a Kali laptop on the network, to simplify the evaluation the network switch which was configured to SPAN all traffic from the web-server to the Kali network port. With WireShark running, a full network trace was captured, while the administrator logged into the management interface.

Figure 23 shows the WireShark filtering parameters used to display the username and password as it was transferred over the network in clear text. Most hackers have a good understanding of network protocols; however if diving through a WireShark output is too difficult, Netresec have developed NetworkMiner a tool that analyzes and formats the output of WireShark file for you. Running NetworkMiner and opening the capture pcap file display the interesting data in tabular format; Figure 24 clearly shows the cloud username and password.

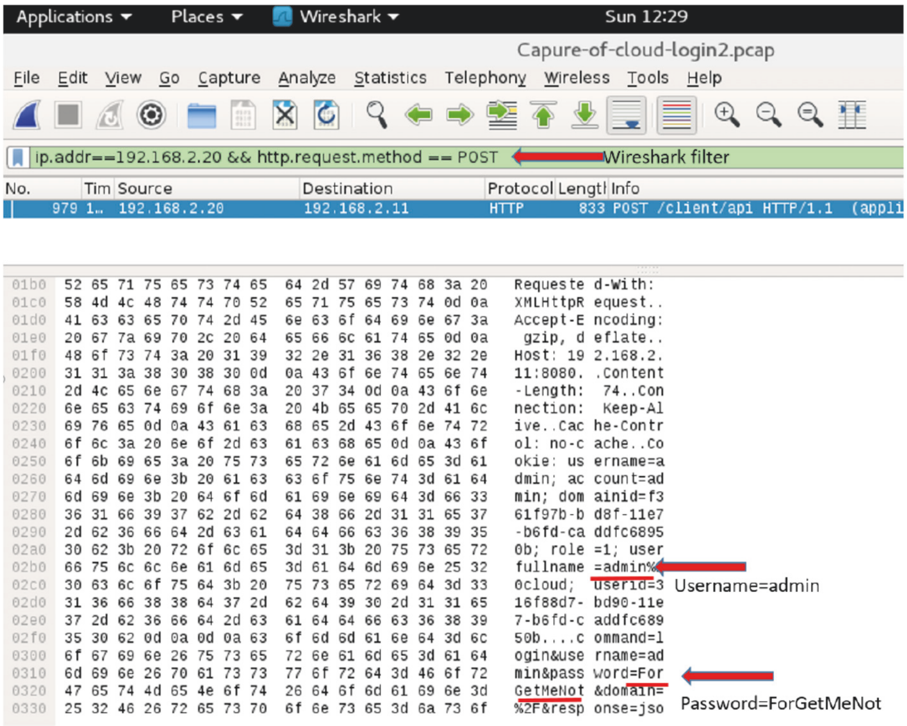
This type of attack could have been prevented if the CloudStack application was configured to use HTTPS for all web traffic; this is a secure and encrypted communication channel. Therefore, if the traffic between the client browser and web-server were encrypted using HTTP, WireShark would not have been able to compromise this password.

4.5. Session Hacking. The following demonstration depicts that if the communication channel between the client and server is insecure, it is possible for a third party to hack the users web session. The default settings for the CloudStack web-server are HTTP and have already been demonstrated as being insecure in previous investigation. This investigation required a proxy server to intercept HTTP traffic between the client and server and then a cookie editor to be able to insert the stolen credentials onto a different user's session. For the proxy server, the Burp-suite application was chosen; this product has a number of features that can be used to evaluate security, one of these being an intercepting proxy server. With the proxy server configured and set to intercept all web traffic, the clients web browser was pointed to the proxy server. With the proxy server having the ability to view all HTTP traffic between client and server, by selecting the parameter tab Figure 25, all the cookie information of the administrator session is displayed. This information does not change for the duration the administrator is connected.

On a separate machine, an unprivileged user was logged into the cloud; this generated a similar cookie but the user had no privileges. Using the browser addon "EditThisCookie" the cookie values from the privileged user in Figure 25 were copied to the unprivileged user in Figure 26. Once the browser was refreshed, the user changed from an unprivileged user to a privileged one; at this point the original administrator was logged out of the system.

Like the previous exercise, this exploit was possible because the traffic between the client's browser and the web-server were insecure because they used HTTP. If the web application used HTTPS and the communication channel is encrypted and this exploit would be a lot harder to perform.

4.6. Brute Force Password Exploit. THC-Hydra application was chosen to perform the brute force password attack; however to build the correct command combination to perform this exploit required a number of values:


Applications
Places
Wireshark
Sun 12:29
Capure-of-cloud-login2.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
ip.addr==192.168.2.20 && http.request.method == POST
```

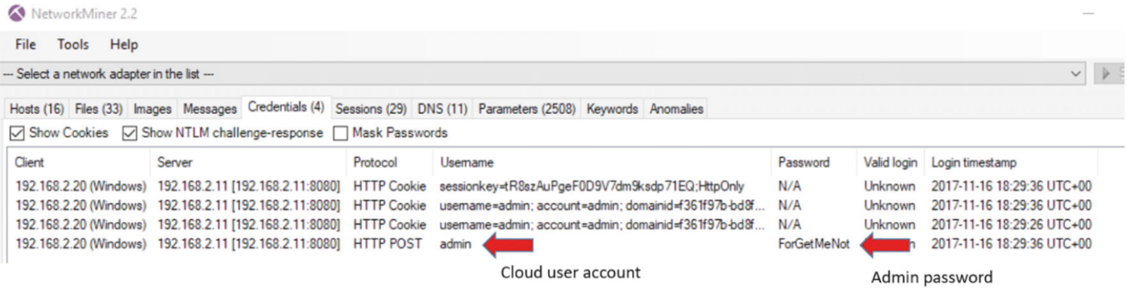
Wireshark filter

```
No.    Tim Source          Destination      Protocol Lengtl Info
  979 1… 192.168.2.20     192.168.2.11     HTTP        833 POST /client/api HTTP/1.1  (appli

01b0  52 65 71 75 65 73 74 65  64 2d 57 69 74 68 3a 20   Requeste d-With:
01c0  58 4d 4c 48 74 74 70 52  65 71 75 65 73 74 0d 0a   XMLHttpR equest..
01d0  41 63 63 65 70 74 2d 45  6e 63 6f 64 69 6e 67 3a   Accept-E ncoding:
01e0  20 67 7a 69 70 2c 20 64  65 66 6c 61 74 65 0d 0a    gzip, d eflate..
01f0  48 6f 73 74 3a 20 31 39  32 2e 31 36 38 2e 32 2e   Host: 19 2.168.2.
0200  31 31 3a 38 30 38 30 0d  0a 43 6f 6e 74 65 6e 74   11:8080. .Content
0210  2d 4c 65 6e 67 74 68 3a  20 37 34 0d 0a 43 6f 6e   -Length:  74..Con
0220  6e 65 63 74 69 6f 6e 3a  20 4b 65 65 70 2d 41 6c   nection:  Keep-Al
0230  69 76 65 0d 0a 43 61 63  68 65 2d 43 6f 6e 74 72   ive..Cac he-Contr
0240  6f 6c 3a 20 6e 6f 2d 63  61 63 68 65 0d 0a 43 6f   ol: no-c ache..Co
0250  6f 6b 69 65 3a 20 75 73  65 72 6e 61 6d 65 3d 61   okie: us ername=a
0260  64 6d 69 6e 3b 20 61 63  63 6f 75 6e 74 3d 61 64   dmin; ac count=ad
0270  6d 69 6e 3b 20 64 6f 6d  61 69 6e 69 64 3d 66 33   min; dom ainid=f3
0280  36 31 66 39 37 62 2d 62  64 38 66 2d 31 31 65 37   61f97b-b d8f-11e7
0290  2d 62 36 66 64 2d 63 61  64 64 66 63 36 38 39 35   -b6fd-ca ddfc6895
02a0  30 62 3b 20 72 6f 6c 65  3d 31 3b 20 75 73 65 72   0b; role =1; user
02b0  66 75 6c 6c 6e 61 6d 65  3d 61 64 6d 69 6e 25 32   fullname =admin%2
02c0  30 63 6c 6f 75 64 3b 20  75 73 65 72 69 64 3d 33   0cloud;  userid=3
02d0  31 36 66 38 38 64 37 2d  62 64 39 30 2d 31 31 65   16f88d7- bd90-11e
02e0  37 2d 62 36 66 64 2d 63  61 64 64 66 63 36 38 39   7-b6fd-c addfc689
02f0  35 30 62 0d 0a 0d 0a 63  6f 6d 6d 61 6e 64 3d 6c   50b....c ommand=l
0300  6f 67 69 6e 26 75 73 65  72 6e 61 6d 65 3d 61 64   ogin&use rname=ad
0310  6d 69 6e 26 70 61 73 73  77 6f 72 64 3d 46 6f 72   min&pass word=For
0320  47 65 74 4d 65 4e 6f 74  26 64 6f 6d 61 69 6e 3d   GetMeNot &domain=
0330  25 32 46 26 72 65 73 70  6f 6e 73 65 3d 6a 73 6f   %2F&resp onse=jso
```

Username=admin

Password=ForGetMeNot

Figure 23: WireShark capture of HTTP login.


NetworkMiner 2.2
File Tools Help
--- Select a network adapter in the list ---
Hosts (16) Files (33) Images Messages Credentials (4) Sessions (29) DNS (11) Parameters (2508) Keywords Anomalies

☑ Show Cookies ☑ Show NTLM challenge-response □ Mask Passwords

Client	Server	Protocol	Username	Password	Valid login	Login timestamp
192.168.2.20 (Windows)	192.168.2.11 [192.168.2.11:8080]	HTTP Cookie	sessionkey=tR8szAuPgeF0D9V7dm9ksdp71EQ;HttpOnly	N/A	Unknown	2017-11-16 18:29:36 UTC+00
192.168.2.20 (Windows)	192.168.2.11 [192.168.2.11:8080]	HTTP Cookie	username=admin; account=admin; domainid=f361f97b-bd8f...	N/A	Unknown	2017-11-16 18:29:36 UTC+00
192.168.2.20 (Windows)	192.168.2.11 [192.168.2.11:8080]	HTTP Cookie	username=admin; account=admin; domainid=f361f97b-bd8f...	N/A	Unknown	2017-11-16 18:29:26 UTC+00
192.168.2.20 (Windows)	192.168.2.11 [192.168.2.11:8080]	HTTP POST	admin	ForGetMeNot	n	2017-11-16 18:29:36 UTC+00

Cloud user account

Admin password

Figure 24: NetworkMiner packet analyzer.

(1) URL of the web application

(2) The type of protocol the login screen understands

(3) The username or a list of usernames to use

(4) A list of passwords to attempt

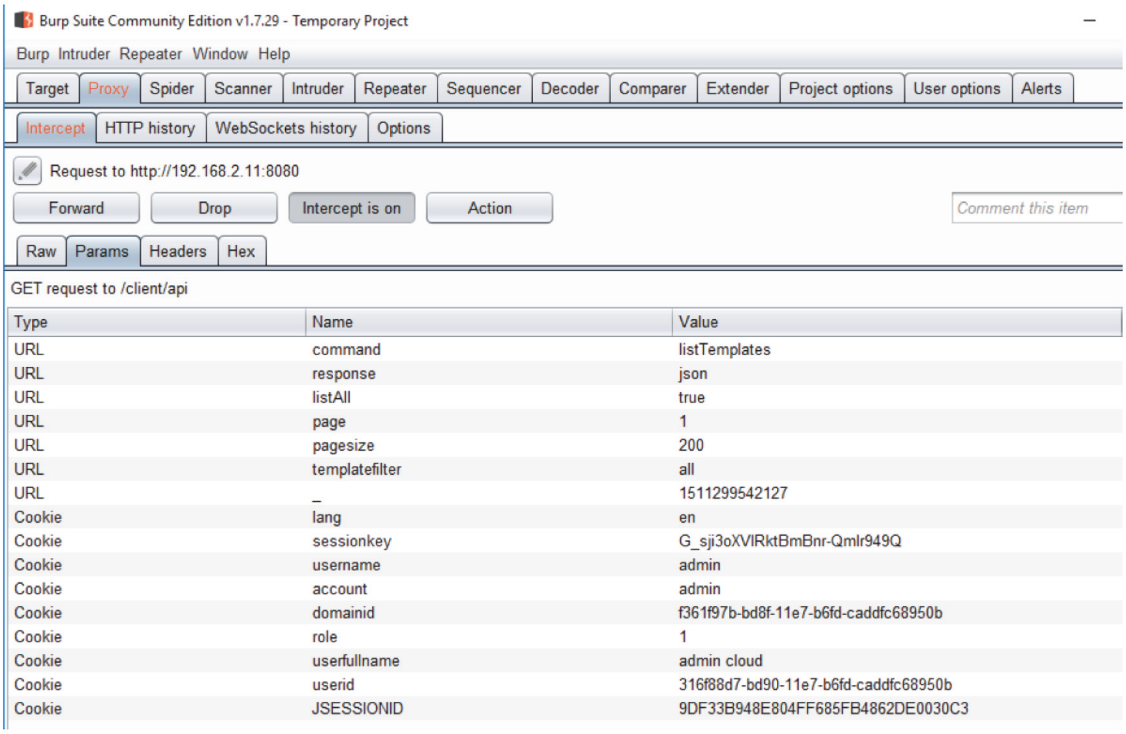
(5) HTTP error message returned when the username or password are incorrect.

Only two points that were unknown were the type of protocol (1) and the error message passed back via HTTP. To capture these values, a proxy server capable of intercepting web traffic and displaying its content was required. Burp-suite is a security testing application, which has an entire testing suite, included within this is a proxy server capable of displaying the contents within the HTTP packets. Using a Kali laptop, the Burp-suite application was configured and set to intercept all web traffic and the local browser pointing to this new proxy server with settings at *127.0.0.1:8080* and removing the bypass proxy field from *127.0.0.1*.

To capture both these missing parameters, the browser was configured to use the proxy and an unsuccessful attempt to login to the cloud console was performed. The purpose of this failed attempt was to capture both the web-form and its associated error string for a failed login. Figure 27 shows the output captured by the proxy server; it unexpectedly showed the web application using javascript not PHP and the error string returned by a JavaScript Object Notation (Json), which cannot be displayed by the proxy server.

A new piece of information captured by the proxy is that the authentication information is passed using a HTTP POST packet. For a first attempt, THC-Hydra was configured to send http-post-form request with the following parameters options:

(1) -V verbose mode

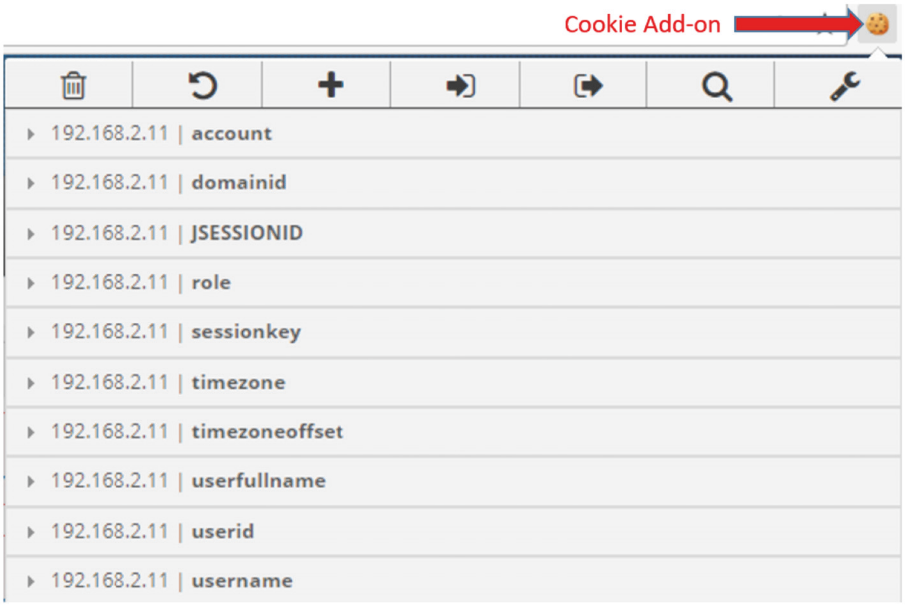


Request to http://192.168.2.11:8080

GET request to /client/api

Type	Name	Value
URL	command	listTemplates
URL	response	json
URL	listAll	true
URL	page	1
URL	pagesize	200
URL	templatefilter	all
URL	_	1511299542127
Cookie	lang	en
Cookie	sessionkey	G_sji3oXVIRktBmBnr-Qmlr949Q
Cookie	username	admin
Cookie	account	admin
Cookie	domainid	f361f97b-bd8f-11e7-b6fd-caddfc68950b
Cookie	role	1
Cookie	userfullname	admin cloud
Cookie	userid	316f88d7-bd90-11e7-b6fd-caddfc68950b
Cookie	JSESSIONID	9DF33B948E804FF685FB4862DE0030C3

Figure 25: Burp-suite view of cookies.



192.168.2.11 | account

192.168.2.11 | domainid

192.168.2.11 | JSESSIONID

192.168.2.11 | role

192.168.2.11 | sessionkey

192.168.2.11 | timezone

192.168.2.11 | timezoneoffset

192.168.2.11 | userfullname

192.168.2.11 | userid

192.168.2.11 | username

Figure 26: Screenshot of the cookie editor.

(2) -l lowercase is used to supply the users name instead of a list of names

(3) -P uppercase is used to supply a list of password “passwords.txt”

(4) -s is the port the service is running, e.g., 8080

(5) 192.168.2.11 is the Apache server IP address

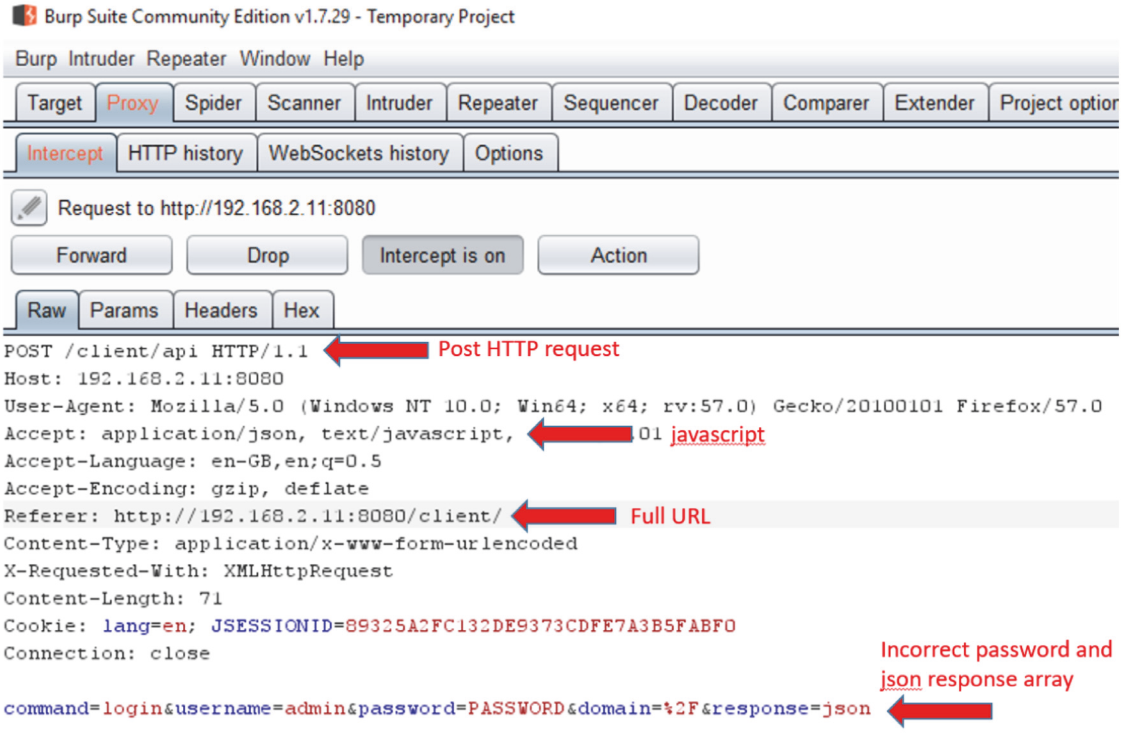
(6) http-post-form is the type of HTTP packet the data will be sent in

(7) Client is the first web-page

(8) Username is either a single name or list

(9) Password is a file containing a list of passwords

(10) F is the failed return text.


Burp Suite Community Edition v1.7.29 - Temporary Project
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project optior
Intercept HTTP history WebSockets history Options
Request to http://192.168.2.11:8080
Forward
Drop
Intercept is on
Action
Raw Params Headers Hex

```
POST /client/api HTTP/1.1
Host: 192.168.2.11:8080
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:57.0) Gecko/20100101 Firefox/57.0
Accept: application/json, text/javascript, 01
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.2.11:8080/client/
Content-Type: application/x-www-form-urlencoded
X-Requested-With: XMLHttpRequest
Content-Length: 71
Cookie: lang=en; JSESSIONID=89325A2FC132DE9373CDFE7A3B5FABF0
Connection: close

command=login&username=admin&password=PASSWORD&domain=%2F&response=json
```

Post HTTP request

javascript

Full URL

Incorrect password and json response array

Figure 27: Failed login attempt.

The commands "password.txt" shown in Figure 27 only tried five passwords that were incorrect and one correct, yet THC-Hydra reported all six passwords were right, quite impossible. After trying an endless amount of commands, it was apparent that something was protecting this site from a brute force attempt, even though the account was not disabled with these repeated attempts. Although the next steps would not be possible for a remote attacker, it was required to understand how the site was blocking the brute force attempts. For a successful and unsuccessful login, WireShark captured the network traffic between the browser and web-server; on both traces, it became apparent that each connection had a unique jsessionid. For the successful login attempt, this jsessionid changed to one that was retained for the full session. Running WireShark while the THC-Hydra was performing a brute force attack captured the reason why each variation of the hydra commands had failed; they all failed due to HTTP 1.1 401 unauthorized. This was because it was not possible to supply a correct jsessionid for each password attempt. Although the CloudStack did not lock the account, which is very poor practice, it did have measures in place to stop a brute force attack.

5. Conclusion

To summarize, we have reviewed the security vulnerabilities of cloud hypervisors in detail by considering the threats and countermeasures. We have focused on the security concerns with this emerging cloud technologies, in particular on data breaches and weak authentication. This paper discussed the various delivery and deployment models and the relationship between the virtualization technology and the cloud management layer; the remainder of the paper focused on the current threat landscape of cloud and virtualization technology, highlighting the major areas of concern. Moreover, we have successfully designed and optimized a small private cloud environment to analyze the weaknesses documented in the literature. We created a number of practical security exploits based on the cloud design that demonstrate the tools and techniques to compromise the CloudStack security. We found that, in case of cloud provider did not correctly sanitize the cloud storage after the storage is released by the previous user, if a virtual hard disk (VHD) is created, its contents are overwritten with zeros, meaning that any forensic tool would not be able to recover signatures of files from the previous tenant. For the specific case of visualization, if all data 'at rest' is encrypted and the data owner managed the key security and rotation, no one stealing this VHD would be able to access the data without the encryption key, not even the cloud provider. Furthermore, for the unencrypted protocol like HTTP, the cookies from the administrator's browser can be captured by using an intercepting proxy server. In a second session, the cookie was imported allowing this unauthorized person access to the privileged cloud account and content, while upgrading this protocol to HTTPS would compensate this vulnerability to some extent. These investigations can be used to understand potential attack paths and identify the vulnerabilities that enabled them.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Acknowledgments

The authors would like to acknowledge the project (Grant no. 830965) research funding from Edinburgh Napier University, UK, for Project STRENGTH (Scalable, Tuneable, Resilient and Encrypted Next Generation Transmission Hub).

References

[1] S. Singh, Y. Jeong, and J. H. Park, "A survey on cloud computing security: Issues, threats, and solutions," *Journal of Network and Computer Applications*, vol. 75, pp. 200–222, 2016.

[2] NIST, *Final Version of NIST Cloud Computing Definition Published*, 2011.

[3] L. Turnbull and J. Shropshire, "Breakpoints: An analysis of potential hypervisor attack vectors," in *Proceedings of the IEEE SoutheastCon 2013: Moving America into the Future*, 2013.

[4] S. Manavi and S. Mohammadalian, "Secure model for virtualization layer in cloud infrastructure," *International Journal of Cybersecurity and Digital Forensics*, vol. 1, no. 1, pp. 32–40, 2012.

[5] L. Almutair and H. Zaghloul, in *Proceedings of the The Third International Conference on Digital Information Processing and Communications (ICDIPC '13)*, pp. 676–686, UAE, 2013.

[6] D. Zissis and D. Lekkas, "Addressing cloud computing security issues," *Future Generation Computer Systems*, vol. 28, no. 3, pp. 583–592, 2012.

[7] Kamesh and N. Sakthi Priya, "Security enhancement of authenticated RFID generation," *International Journal of Applied Engineering Research*, vol. 9, no. 22, pp. 5968–5974, 2014.

[8] M. Ali, S. U. Khan, and A. V. Vasilakos, "Security in cloud computing: opportunities and challenges," *Information Sciences*, vol. 305, pp. 357–383, 2015.

[9] O. Harfoushi, B. Alfawwaz, N. a. Ghatasheh et al., "Data security issues and challenges in cloud computing: A conceptual analysis and review," *Communications and Network*, vol. 6, no. 1, pp. 15–21, 2014.

[10] N. Arya, M. Gidwani, and S. K. Gupta, "Hypervisor security - A major concern," *International Journal of Information and Computation Technology*, vol. 3, no. 6, pp. 533–538, 2013.

[11] A. Cleeff, W. Pieters, and R. Wieringa, "Security implications of virtualization: A literature study," in *Proceedings of the 12th IEEE International Conference on Computational Science and Engineering (CSE '09)*, vol. 3, pp. 353–358, 2009.

[12] S. Jin, J. Seol, J. Huh, and S. Maeng, "Hardware-assisted secure resource accounting under a vulnerable hypervisor," *ACM SIGPLAN Notices*, vol. 50, no. 7, pp. 201–213, 2015.

[13] R. L. Mitchell, *The Scary Side of Visualization*, Computer World, 2010.

[14] S. N. Brohi, M. A. Bamiah, M. N. Brohi, and R. Kamran, "Identifying and analyzing security threats to virtualized cloud computing infrastructures," in *Proceedings of the 2012 International Conference on Cloud Computing Technologies, Applications and Management, ICCCTAM 2012*, pp. 151–155, are, December 2012.

[15] J. Shropshire, "Analysis of monolithic and microkernel architectures: Towards secure hypervisor design," in *Proceedings of the 47th Hawaii International Conference on System Sciences, HICSS 2014*, pp. 5008–5017, usa, January 2014.

[16] G. Pek, L. Buttyan, and B. Bencsath, "A survey of security issues in hardware virtualization," *ACM Computing Surveys*, vol. 45, no. 3, article no. 40, pp. 1–34, 2013.

[17] L. Coppolino, S. D Antonio, G. Mazzeo, and L. Romano, "Cloud security: Emerging threats and current solutions," *Computers and Electrical Engineering*, pp. 1–15, 2015.

[18] T. Threats and W. Group, *Cloud Computing Top Threats in 2016 The Treacherous 12*, 2016.

[19] C. Skinner, "The most innovative banks of 2015," *The Finanser*, pp. 1–11, 2015.

[20] A. Ukil, D. Jana, and A. De. Sarkar, "A Security Framework in Cloud Computing," *International Journal of Network Security & Its Applications*, vol. 5, no. 5, pp. 11–24, 2013.

[21] L. Adhianto, S. Banerjee, M. Fagan et al., "HPCTOOLKIT: Tools for performance analysis of optimized parallel programs," *Concurrency and Computation: Practice and Experience*, vol. 22, no. 6, pp. 685–701, 2010.

[22] W. Dawoud, I. Takouna, and C. Meinel, "Infrastructure as a service security: challenges and solutions," in *Proceedings of the 7th International Conference on Informatics and Systems*, pp. 1–8, March 2010.

[23] A. Singh and K. Chatterjee, "Cloud security issues and challenges: A survey," *Journal of Network and Computer Applications*, vol. 79, pp. 88–115, 2017.

[24] S. K. Majhi and S. K. Dhal, "A Study on Security Vulnerability on Cloud Platforms," in *Proceedings of the 1st International Conference on Information Security and Privacy 2015*, pp. 55–60, ind, December 2015.

[25] Alliance Cloud Security, *Security Guidance for Critical Areas of Focus in Cloud Computing*, Cloud Security Alliance, 3 edition, 2011.

[26] I. M. Khalil, A. Khreishah, and M. Azeem, "Cloud computing security: A survey," *Computers*, vol. 3, pp. 1–35, 2014.

[27] A. A. Malik, H. Anwar, and M. A. Shibli, "Federated identity management (FIM): Challenges and opportunities," in *Proceedings of the Conference on Information Assurance and Cyber Security, (CIACS '15)*, vol. 1, pp. 75–82, 2016.

[28] S.-K. Kim, S.-Y. Ma, and J. Moon, "A novel secure architecture of the virtualized server system," *The Journal of Supercomputing*, vol. 72, no. 1, pp. 24–37, 2016.

[29] S. Subashini and V. Kavitha, "A survey on security issues in service delivery models of cloud computing," *Journal of Network and Computer Applications*, vol. 34, no. 1, pp. 1–11, 2011.

[30] I. Khalil, A. Khreishah, and M. Azeem, "Cloud Computing Security: A Survey," *The Computer Journal* , vol. 3, no. 1, pp. 1–35, 2014.

[31] N. Sabharwal and R. Shankar, *Apache Cloudstack Cloud Computing*, Packet Publishing Ltd., 2013.

[32] Microsoft, *Virtual Hard Disk Image Format Specifications*, 2006, Virtual Hard Disk Image Format Specifications.

[33] X. Wang, Z. Du, Y. Chen, and M. Yang, "A green-aware virtual machine migration strategy for sustainable data-center powered by renewable energy," *Simulation Modelling Practice and Theory*, vol. 58, pp. 13-14, 2015.

[34] K. Beaver, *Cracking Passwords The Web Application Way*, 2007.

[35] T. Cerling, J. Buller, C. Enstall, and R. Ruiz, *Mastering Microsoft Virtualization Imprint*, Wiley Publishing, Inc., 2009.

[36] M. A. Khan, “A survey of security issues for cloud computing,” *Journal of Network and Computer Applications*, vol. 71, pp. 11–29, 2016.

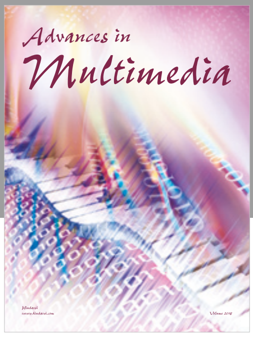
[37] F. Zhang and H. Chen, “Security-preserving live migration of virtual machines in the cloud,” *Journal of Network and Systems Management*, vol. 21, no. 4, pp. 562–587, 2013.

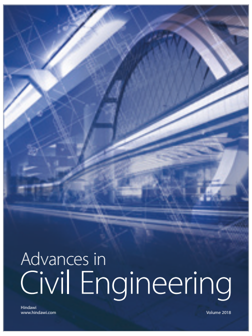
Journal of
Engineering

The Scientific
World Journal

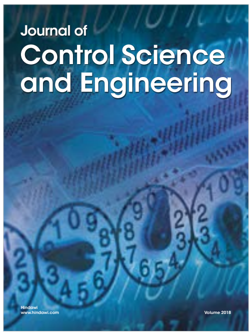
International Journal of
Rotating
Machinery

Journal of
Sensors

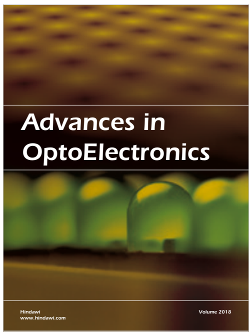

Advances in
Multimedia


Advances in
Civil Engineering

Journal of
Robotics

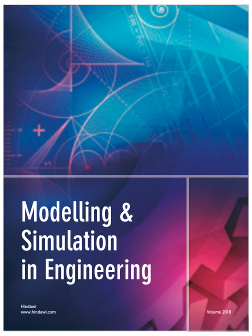

Journal of
Control Science
and Engineering

Journal of
Electrical and Computer
Engineering

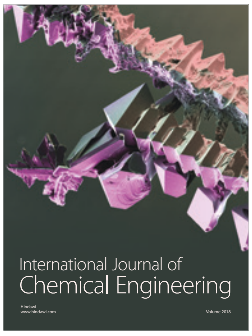

Advances in
OptoElectronics

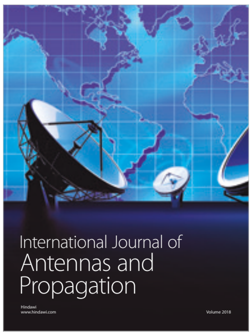
VLSI Design

International Journal of
Navigation and
Observation

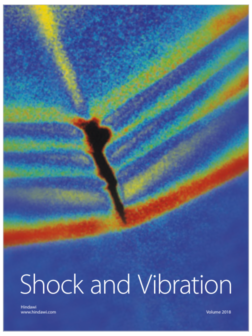

Modelling &
Simulation
in Engineering

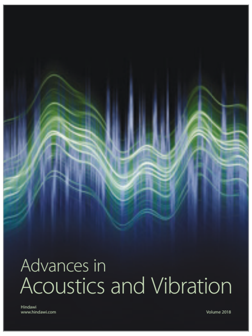
International Journal of
Aerospace
Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and
Propagation

Active and Passive
Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration