

Authenticated control to VMs in Xen cloud

CS526 Semester Project

By

Abdulrhman Alsahafi

&

Ahmed Alharthi

Director

Prof. Edward Chow

Abstract

The Xen Cloud is an open source virtualization system and cloud computing platform. Xen Cloud is based on Xen Hypervisor that supports a range of guest operating systems such as Windows and Linux, network, and storage. Indeed, it addresses user requirements for security, availability, performance and isolation. By combining the isolation and multi-tenancy capabilities of the Xen hypervisor with enhanced security, storage and network virtualization technologies to offer a rich set of virtual infrastructure cloud services. Consequently, Xen API projects are needed to configure, manage and control Xen Cloud Platform (XCP) including the virtual machines. In fact, there are a number of management tools that use Xen through Xen API as part of XCP; however, for educational purpose, we need a XEN API to control XPC and VMs in a way that is easy and secure. This idea requires that the API can be accessible by only authenticated users who do not have an account on the host itself. Our objective in this project is to implement a command line-based XenAPI that implies an authenticated procedure that ensures the consistency, availability, and integrity to XCP and VMs.

1. Related works:

These are a number of management tools that use Xen through Xen API, which is part of XCP.

- **OpenXenManager**: OpenXenManager is an open source clone of Citrix's [XenServer XenCenter](#) and manages both XCP and Citrix's [XenServer](#).
- **Xen Cloud Control System (XCCS)**: XCCS is a lightweight front end package for the excellent Xen Cloud Platform cloud computing system. XCCS is totally web based so any computer or smart phone with a web browser can be used with it!
- **Xen Orchestra**: Simple web management interface to Xen infrastructure
- **Xen Web Manager**: Open source project to manage XCP and [XenServer](#) virtual machines with web (similar to [XenCenter](#)). It is web-based and shares [OpenXenManager](#)'s codebase
- **Zentific**: Zentific is a web-based management interface for the effective control of virtual machines running upon the Xen(R) hypervisor. Designed to maintain a high level of reliability, security, scalability, and ease of use, the interface serves both administrators and end-users of all proficiency levels. In the spirit of the open-source projects which have made Zentific possible, the application is itself open to the community.

2. Background

3.1. Introduction to Virtualization

Virtualization in computing is the creation of a virtual (rather than actual) version of something, such as a hardware platform, operating system, storage device, or network resources. IT community define virtualization as “framework or methodology of dividing the resources of a computer into multiple execution environments, by applying one or more concepts or technologies such as hardware and software partitioning, time-sharing, partial or complete machine simulation, emulation, quality of service, and many others.”

IT point of view, virtualization can help the industry overall by providing virtual environments that help clients efficiently who are pay for it. The main goal of virtualization is use overall hardware-resources efficiently in order to reduce cost and improve scalability, all that in environment that can support virtualization machines which has to be partitioned so that multiple operating systems can concurrently executed. [1]

3.2. Why is Virtualization Useful?

A number of reasons were found out that do make virtualization very useful. Sources include reviews mainly from kernelthread.com [2] and Technology Trends [3].

- With only few machines the work load of several underutilized servers can be consolidated, benefiting in terms of monitorial savings of hardware, management and hardware.
- Secure Computing platforms are provided. Harmful applications could be separated from the OS and processed in a different ring or a container depending on the type of virtualization technique being used. It also enables existing operating systems to run on shared memory multiprocessors.
- Resource limits can be set that give advantage of setting execution environment with allocated time slices, memory and other resources.

- Simulated networks of independent computers can be created as virtualization can project an illusion of hardware or hardware configuration that is not present.
- Some old files require formats that are not currently used but with virtualization different operating systems can be run even different versions.
- Virtual machine monitors can debug operating systems without losing productivity.
- Faults and errors can be contained as virtual machines can isolate what they run. Entire Research work can also be isolated, saved separately and later loaded to run.
- With hypervisor working virtualization can be made effective to provide binary compatibility.
- Server utilization is being increased with virtualization as multiple applications can be run without each intervening with another.
- Multiple virtual computers can be hosted on a single server as each application runs as if it was on a real server and on its own operating system.

3.3. Types of Virtualization

3.3.1 Paravirtualization

Paravirtualization allows multiple operating systems to run on hardware at the same time by making more efficient use of system resources, such as processors and memory, through effective resource sharing. Paravirtualization requires an operating system which has to be modified to run with paravirtualization in order to make it cooperate with the host more efficiently which is actually main disadvantage in this type. This means guest operating systems are aware that they are executing on a VM. However, paravirtualization's efficiencies can also lead to better scaling for many facts, such as it requires 10% processor utilization per guest, it has higher throughput, lower latency and reduced CPU utilization. Xen is an example of paravirtualization environment.

3.3.2 Full Virtualization

Full virtualization provides complete abstraction of the underlying physical system and creates a complete virtual system in which the guest operating system can execute. Unlike paravirtualization, no modification is required in the guest operating system or application; the guest OS or application is not even aware that it is running within a virtualized environment. The guest OS executes on the VM just as they would on a physical system. This can be advantageous because it enables complete decoupling of the software from the hardware. As a result, full virtualization can streamline migration of applications and workloads between different physical systems. Full virtualization provides complete isolation of different virtualized applications, which makes this approach highly secure by design. Microsoft Virtual Server and VMware ESX Server are examples of full virtualization.

3.3.3. Kernel Based Virtual Machine (KVM)

KVM is a full virtualization solution for Linux on x86 hardware containing virtualization extensions (Intel VT or AMD-V). It consists of a loadable kernel module, `kvm.ko`, that provides the core virtualization infrastructure and a processor specific module, `kvm-intel.ko` or `kvm-amd.ko`. KVM also requires a modified QEMU although work is underway to get the required changes upstream. *QEMU* stands for "Quick EMUlator" and is a processor emulator that relies on dynamic binary translation to achieve a reasonable speed while being easy to port to new host CPU architectures. Indeed, using KVM, can run multiple virtual machines running unmodified Linux or Windows images. Each virtual machine has its own virtualized hardware, a network card, disk, graphics adapter, etc. . As result, adding KVM virtualization make machines benefit from Linux kernel environment

efficiently.

4. Xen

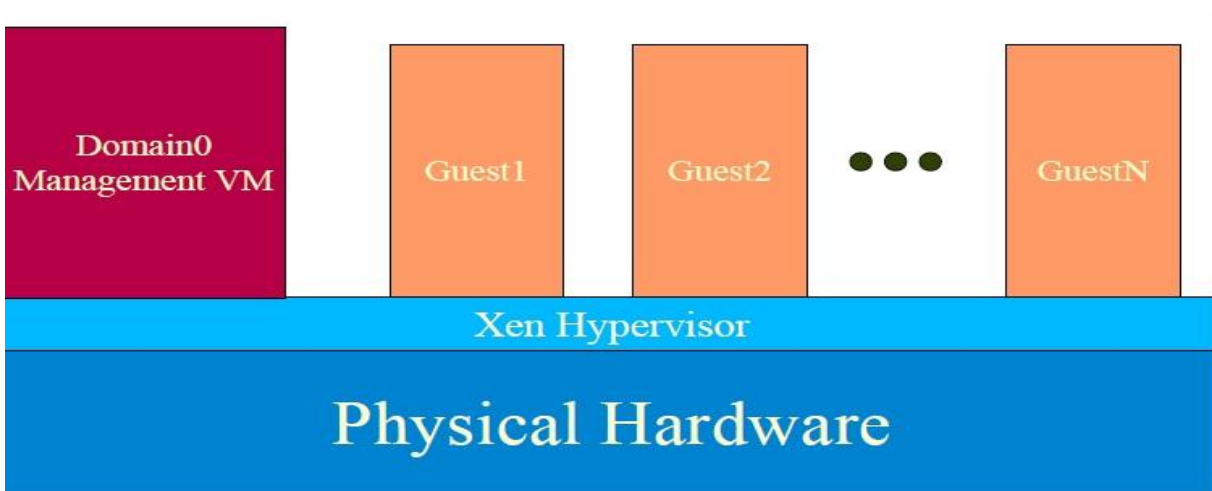
4.1. History:

The Xen hypervisor was first created by Keir Fraser and Ian Pratt as part of the Xenoserver research project at Cambridge University in the late 1990s. During the years research has not sopped until the first Xen 1.0 was officially released followed a short time later by Xen 2.0. At the same time, Ian Pratt and several other technology leaders founded XenSource, Inc. to convert the Xen hypervisor from a research tool into a competitive product for enterprise computing. As part of the corporate strategy, the Xen hypervisor remained an open source solution. The product has succeed and has been applied in most cloud computing technology such as, Amazon and Google cooperators. [5]

4.2. What is Xen?

Xen is an open-source paravirtualization technology that provides a platform for running multiple operating systems in parallel on one physical hardware resource, while providing close to native performance. The current version of Xen also supports the new generation of AMD Pacifica and Intel VT-x chipsets and can run an OS on these chips without any modifications by using a version of the hypervisor called the *Hardware Virtual Machine* (HVM). HVM mediates between the guest operating system and the hardware and passes on the calls made by the guest to the physical hardware. So user can run Microsoft Windows on these chips using Xen. [5]

5. Xen Hypervisor architecture



The Xen Hypervisor is the major abstraction level of software that located on the hardware under any operation systems. It is a small hypervisor kernel that deals with virtualizing the CPU, memory and critical I/O resources such as the Interrupt controller. In other words, hypervisor has two main jobs, managing CPU scheduling and memory partitioning of many virtual machines which running on the hardware device. In addition, the hypervisor has no knowledge about the surrounding environment such as, networking, storage devices, videos, or any other I/O sources. Figure (1) shows the general architecture and the main job of Xen hypervisor.

5.1.Domain 0 guest

The term “Domain0” refers to a special domain that provides the management environment. This may be run either in graphical or in command line mode. Domain 0, which is the primary domain in Xen systems is a modified linux kenral (Para-virtualized Linux) running on the hypervisor supported by direct hardware access and guest management responsibilities. It can be any other operation system except Windows. Moreover, domain0 has control panel application that controls and manages the sharing of the processor, memory network and disk devices so Xen virtualization environments require Domain 0 to be running before any other virtual machines can be started.

5.2.Multiple Domain U

Unprivileged domain guest. Typically, they have no direct access to physical hardware on the machine as a Domain0 guest does. Dom Us are launched and controlled by the Domain 0 and independently operate on the system. Furthermore, the Dom Us can be run with a special modified operating system referred to as parvirtualuizion or un-modified operating systems. Domain us has two major parts:

5.2.1. Domain U PV (Paravirtualized):

which is Paravirtualiztion domain guest runs on Xen hypervisor. They are a modified operation systems such as, UNIX, Solaris, etc.

5.2.2. Domain U HVM (Hardware Virtualization Machine):

which is a fully virtualiztion domain guest runs on Xen hypervisor but they use un-modified operation system, windows for example. (6)

Note: Domain U PV guest is aware that it does not have direct physical access to the hardware. Also, it is aware that there are other virtual machines running side by side on the same machine. Domain HVM guest works oppositely, it is not aware that it is sharing resources such as process time and there are other machines present.

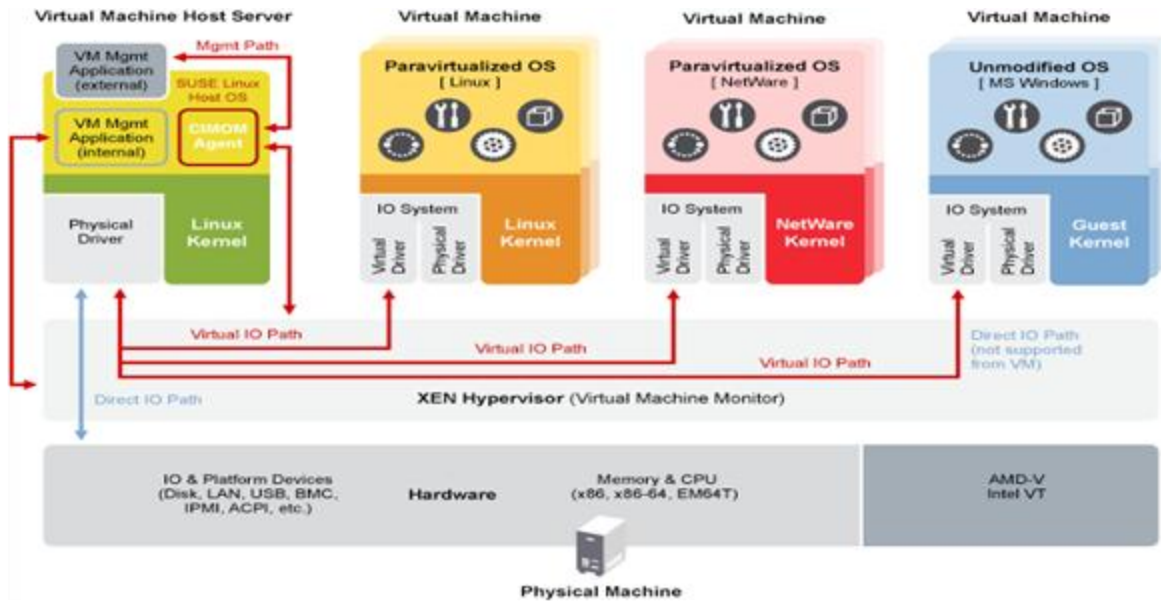


Figure 2 shows how Xen actually handling virtual environment.

VM: Virtual Machine can run applications on its own operating system. **VMM:** Virtual Machine Monitor provides the ability to run virtual machine.

6. Citrix XenServer:

Understanding point: Before we start to discuss Citrix XenServer, we need to understand first the differences between Xen and XenServer and Xen Cloud Platform (XCP). First, Xen: the first product of xen family, open source and completely free, which have many product built on it, such as Citrix XenServer. Xen is different from its family member; it is only hypervisor and not virtualization product (it has nothing, no management console, no other features). Second, XenServer: the first commercial virtualization product which built on Xen hypervisor. XenServer is not completely free. XenServer is a full suite server virtualization which has an all management tools. Comparing with Xen, XenServer does not require a perfect Knowledge of Dom0 environment. Last, Xen Cloud Platform (XCP) is a full virtualization suite based on Xen. It uses real Xen and combines in all of the other components needed to be a full suite just like XenServer. It is an open source and derived from the XenServer.

- ❖ **Citrix XenServer:** is complete server Virtualization platform for managing, integrating and automate a virtual Servers and datacenters that built on top of powerful xen hypervisor. Xen server was design to host and manage different guest's virtual servers simultaneously such as Linux and Windows. In addition, using leveraging resource clustering technology and industry-standard shared storage architectures which is created by XenSource, XenServer allow combining multiple XenServer into a powerful Resource Pool that share the storage, CPU, memory, and networks resource. Furthermore, XenServer simply the management to the IT managers through allowing them to create multiple cluster of resource pool and managing them from a single XenCenter management console. Thus, XenServer reduce the complexity and simplifying the adapt with utility of data center. XenServer is also high available compute cluster that offers no-downtime maintenance through allowing VMs to move from cluster to another while they are running.

Firstly, the power of XenServers that allows the IT administrator to assign up to 16 X86 servers in a single resource pool, And managing them from One xen center management console. In resource pool the collection of servers are strongly combined to each other's, and their resources are virtualized to host numbers of virtual Machines. Each Server in a resource pool proctors its peers to check their configuration state and availability, also xenServer management state is re-duplicated across servers in cluster. This can prevent system failure through replacing the failed node by any node in the cluster. In addition, XenServer can protect the the workload of a cluster from server failure, through the unique XenServer clustering architecture, the combination of shared storage, Xen virtualization, and re-duplicated state management between server in cluster. Secondly, every virtual machine that assigned to a resource pool gets mapped into the physical resource automatically. However, the IT administrators still have the control privileges to the resource assignment and full visibility into each server in the pool and each Virtual machine that server hosted. XenServer simplify the management to the IT administrator, the administrator just need to assign virtual machine or a group of virtual machine to a resource pool, and XenServer do the rest including the assignment of physical resource, and implement the administrator policies to restart the virtual machines. According above information we can conclude that XenServer Has been extended the power of Xen virtualization to server cluster via providing different features supports variety kind of cluster needs[7].

6.1.XenServer Functions:

Starting to understand the hypervisor functionality, which is to separate both layers of the operating systems and the applications from the hardware, we try here to give brief explanation of most important XenServer functions of the management layer:

First, Embedded Management Console [8]: Citrix offers a built in graphical interface with a management functions including the capacity planning tools and performance monitoring. To compare this functionality with Hyper-V and VMware, you have to buy additional tools to do such as functionality.

Second, Xen Motion [8] : If XenServer A ,which hosted in Resource pool, needs a maintenance, Xen Motion Copies Virtual Machines that hosted in server A to another available server in the same Pool while they are

running. XenServer A then can go under maintenance, and Virtual Machines copied back after that. This functionality, which is part of Citrix maintenance mode, offers Zero downtime for XenServer A. This functionality is also called migration

Third, High Availability: XenServer is a high available [8] compute cluster that's appears if a host server fails, the virtual machines automatically restart on another server within the pool. Moreover, the citrix management is not one server responsibility, its pool of server and that appears if management server fails, another server will take over management.

Fourth, Open APIs for storage management [8]: Compared with VMware Xen doesn't have its own file systems. However, Xen server is the unique in virtualization market that have built-in support for various kind of drives such as IDE, SATA, SCSI, and SAN storage drives. XenServer can manage all forms of storage in any server located in Resource Pool through offering an open API to integrate with different kind of storage management technology.

Fifth, Tagging and searching [8]: This functionality allows the administrator to add a metadata to identify groups. This is useful for a large number of machines that required same application update for example, IT administrator can search for others machine tagged this application, then push out the update requested.

Last but not least, Dynamic workload management [8]: for both physical and virtual Servers, and according to resource availability and others reasons, the workload Streams can be managed by XenServer. The main advantage is this done for both physical and virtual machines, and it works for both the new users and when the server fails and the machines need to be reassigned.

❖ XenServer Editions:

Platinum: Data protection and resiliency for enterprise-wide virtual environments.

Enterprise: Essential integration and optimization capabilities for production deployments of virtual machines.

Enterprise: Key high availability and advanced management tools

Free Edition: delivers uncompromised performance, scale, and flexibility at no cost

The table 1 below shows the commercials XenServer aditions and the features that combined in each one.

<i>XenServer Features by Edition</i>	Advance d	Enterpris e	Platinu m
XenServer Hypervisor	✓*	✓*	✓*
Resilient distributed management architecture	✓*	✓*	✓*
VM disk snapshot and revert	✓*	✓*	✓*
XenCenter management console	✓*	✓*	✓*
XenConvert	✓*	✓*	✓*
XenMotion® live migration	✓*	✓*	✓*
Automated VM protection and recover	✓*	✓*	✓*
Distributed virtual switching	✓*	✓*	✓*
Heterogeneous pools	✓*	✓*	✓*
High availability	✓*	✓*	✓*
Memory optimization	✓*	✓*	✓*
Performance reporting and alerting	✓*	✓*	✓*
Dynamic workload balancing		✓*	✓*
GPU pass-thru		✓*	✓*
Host power management		✓*	✓*
IntelliCache for optimized XenDesktop deployments		✓*	✓*
Live memory snapshots and revert		✓*	✓*
Provisioning Services (virtual)		✓*	✓*

Role-based administration		✓*	✓*
StorageLink™		✓*	✓*
Web management console with delegated admin		✓*	✓*
Provisioning services (physical)			✓*
Site Recovery			✓*

(Table1[Citrix site])

7. Xen Cloud Platform(XCP):

XCP is an open source project for server virtualization and cloud computing platform that offer near bare-metal with a full server virtualization. XCP is built on Xen hypervisor and companies all needed tools to be a full suite such as XenServer. Since XCP was the open source version of xenServer, it was design to work with the most XenServer APIs. In addition, XCP is also support a range of guest operating systems such as linux and Windows. Network, storage, management tools and XCP Appliances are also supported.

7.1. Appliances:

There are many xen Appliances support different xen systems. These appliances are also called Xen management APIs or XAPI, here we are going to discuss briefly some of these Appliances giving brief explanation for each one.

1) Command line Interface:

XAPI is including a strong Command line interface (xe). The xe CLI need to be installed in another Linux box, and the Xe command may be controlled locally in Dom0, or by remote host. This makes dealing with XenServer or XCP more efficiently. CLI command can be spilt into two groups; low level command and which dealing with listing and controlling parameter of API objects, and higher level command which interaction with the host and VMs in higher level.

The low level command can be listed:

- Class-list
- Class-param-set
- Class-param-get
- Class-param-add
- Class-param-clear
- Class-param-remove

The classes can be any thing that deal with VMs and Host in low level which may be one of these class:

- VM
- Host
- Pool
- Network
- Console

Several xe command must take a parameter when they are operating, and most of the parameter takes one value such as “name=label” example: “xe vm-param-start”. Other type of parameter is multi-parameter, takes two or more parameters and list of values.

Example: `xe vm-param-set uuid=VM uuid other-config:foo=baa`

2) Citrix XenCenter: is a graphical user interface for managing XenServer and Xen Cloud Platform (XCP) and the virtual machines hosted by these servers. It is support Variety features include:

- Searching, sorting, and grouping folder and custom files using tags functionality.
- Dynamic memory management.
- Complete resource pool management.
- High availability configuration.
- VM snapshot management.
- Full memory snapshots and VM rollback.
- Workload balancing, for pool-wide load balancing recommendations.
- Workload Reports, giving performance views over time and across the datacenter.
- Automated power management, to turn off idle servers.
- Performance metrics display.
- Long term metrics gathering and analysis.

7.2. Xen Tools:

Xen tools can be define simply as collection of a small Perl script that allow or help Xen users to do different functionality such as creating new guest xen domain, and deleting xen domain which the user had created before. Here we are going to briefly discuss one of these tools:

- **xen-create-image :**

This simple script allows you to create new xen guest domain easily. The new guest domain will be configured with networking and having open ssh installed upon it, also most of its basic files will setup correctly without specifying all options because they are already specified in the configuration files. The new image will be given two volumes, which are one image for system root disk, and one for system swap device.

Example1: there are deferent scenario depend on the host file, in this example we try to consternate on LVM logical volume. Using LVM volumes is an efficient process it just require the host has LVM setup, and specify `lvm=volume-group` using command line.

The command is like this:

```
xen-create-image --hostname=test.my.flat \  
--ip=192.168.1.1 \  
--lvm=vol1 \  
--dist=sarge.
```

When this command is finished, you are able to see your new instance, just type this command “`lvsdisplay -c`” .The output will have two volumes mentioned before one for system root disk and one for system swap device as the shows below:

```
# lvsdisplay -C LV VG Attr LSize Origin Snap% Move Log Copy% test.my.flat-  
disk mine -wi-ao 4.00G test.my.flat-swap mine -wi-ao 20.00M
```

8. The Proposed Approach:

We propose a simple command line based Xen controller that allows users to manage Xen hypervisor in secure manner. That is, instead of assigning users with many accounts on Xen system, we just assign an application with account and make it behave as a user based on the actual users' commands. To simplify the approach, let us assume that a user wants to list all virtual machines on Xen. Typically, this user must use Linux Shell Terminal or SSH connection to execute Xen command such as `[xm vm-list]`. However, allowing untrusted users to connect to Xen system is sometimes considered as a system threat. Our proposal is to have an approach that behaves like Linux Shell Terminal and executes only and only Xen commands. However, the approach should be secure enough to be used by even untrusted users. Our target is those who want to practice Xen commands through this command line based interface. In addition, it is aimed to help those professional users who are willing to control their Xen based cloud platforms through our approach anytime and anywhere without carrying assistant instruments. For example, if the system administrator is traveling and he indeed wants to control his system through public computers, he can remotely access a web-based interface and do his business. The approach hides the other requirements and controls the access and the type of commands that can be used.

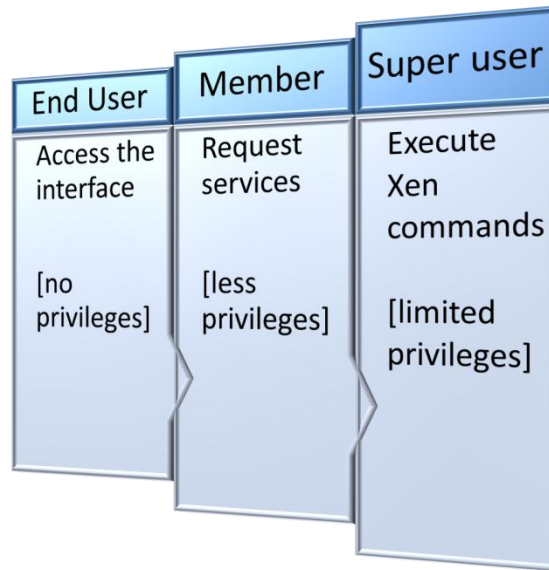
8.1. Model Design:

In principle, the design consists of levels of users and components. Each user and component function based on some roles specified in the design. Figure 1 below shows three different levels of system users:

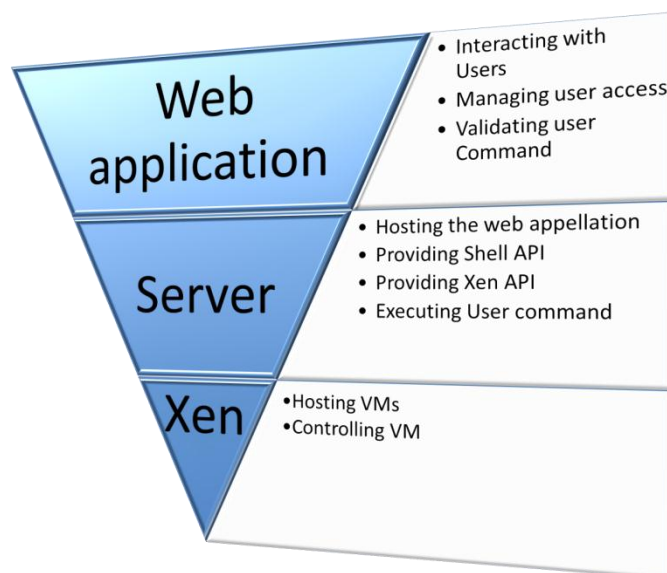
End user: This term means the individual who uses can access this approach, but cannot use it without permission. Usually end user needs to be second level to benefit from the approach because end user has no privileges completely. We intend to have this kind of user level in case there be public information can be deployed to any user access the approach.

Member: A member is a person who belongs to this approach and has been approved to use and benefit from the approach. There are different levels of privileges applied to this level of users depending on the designer. For example, some users can be admin who is capable to control other members and assign them with appropriate privileges

Superuser: it is the highest level of users which is the conventional name of the user who has all privileges or permissions to run low level commands. In this approach, superuser is the user that Xenhypervisor communicates with.



These three users exist in different levels of components which each have limited access control. Figure 2 illustrate the three main components in hierarchical manner.



Web application: an application that is accessed over the Internet which is usually coded in a browser-supported language. In our approach we use PHP and HTML languages. The reason of choosing web application for our approach is due to the ubiquity of web browsers, the ability to update and maintain it without affecting the users, and the capability of allowing many users to access it simultaneously.

Webserver: a web server refers to either the physical web server or virtual web server that helps to deliver web content through the Internet. This component is used to host and run the web application in order users can access it. In our approach, we use virtual machine with web server installed to host and run the implementation of this approach.

Xen hypervisor: A thin layer known between the server's hardware and virtual machines. In our approach communicates with Xen cloud platform through an integrated API named xapi using the command *xe*.

8.2. Implementation:

1. Test Bed:

In real world, we implemented the approach based on the model shown in Figure 0 which consists of three main entities:

Xen hypervisor-Based Server: We chose Xen Cloud Platform (XCP) 1.5 Beta and XenServer (XS) 6.0 to be the targeted Xen hypervisor-Based virtualization servers. Both XCP and XS have Xen API which is a management interface for remotely configuring and controlling virtualized guests running on a XCP or XS host. XAPI executes `xe` command for managing the host, guest, storages, and so on.

Linux-Based Virtual Machine: We needed a Linux operating system to install XAPI on it so that the `xe` commands can be executed by users through the web interface to manage Xen hosts and guests remotely. XAPI uses *openssl* in its communication with other Xen hosts. In our implementation we used it to take advantage of this security feature. Figure 0020 show `xe` command works on a Linux VM after installing the *xapi* binary package successfully on it.

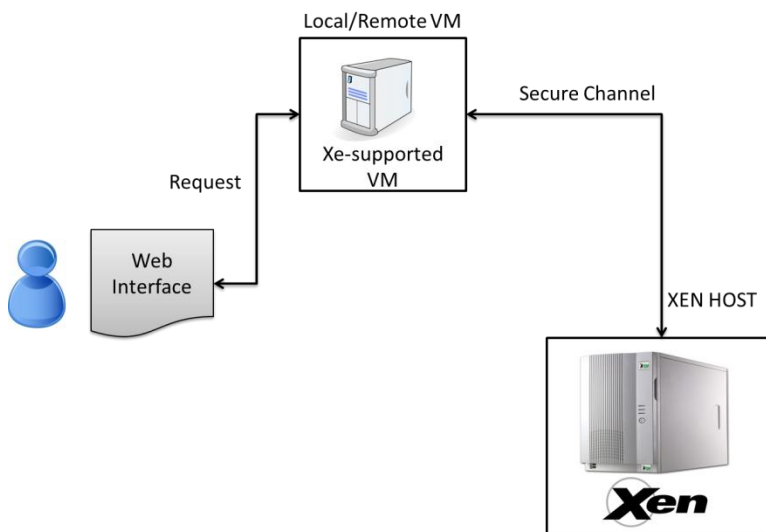


```
xcp@dhcpc7:~/home/xcp
File Edit View Search Terminal Help
[xcp@dhcpc7 ~]$ su
Password:
[root@dhcpc7 xcp]# xe -s 10.0.0.8 -u root -pw 123456 vm-list
uuid ( RO)          : 1f0d57a4-aa95-4162-8670-4f7d6429106c
  name-label ( RW)  : Control domain on host: XS-IBM
  power-state ( RO) : running

uuid ( RO)          : 2171d2fb-7109-9d9f-5739-1e9e2a22dbe7
  name-label ( RW)  : DemoVM
  power-state ( RO) : halted

[root@dhcpc7 xcp]#
```

PHP Web Application: Interactive web application needs one or more web programming language. We chose PHP since it is a powerful server side web language and easy to use.



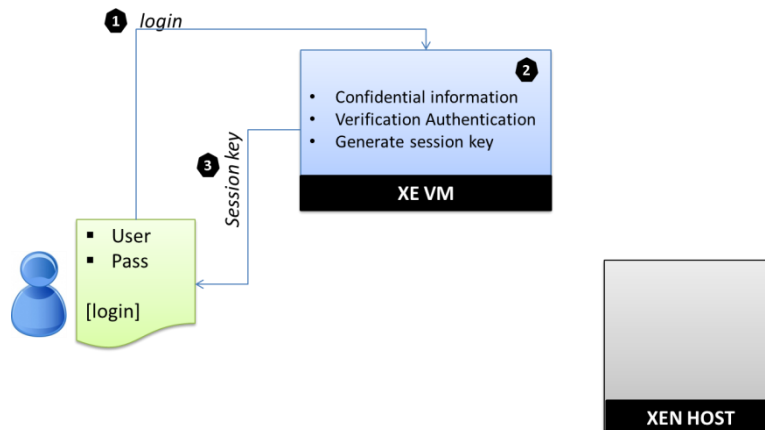
2. Process Phases:

1) Authentication:

End user must become a member which the second level of the user levels. That is, member needs to identify himself by providing his credentials such as username and passcode. The protocol used in this phase basically is:

- Member → Web App: Username& Password
- Web App → Member: Session key

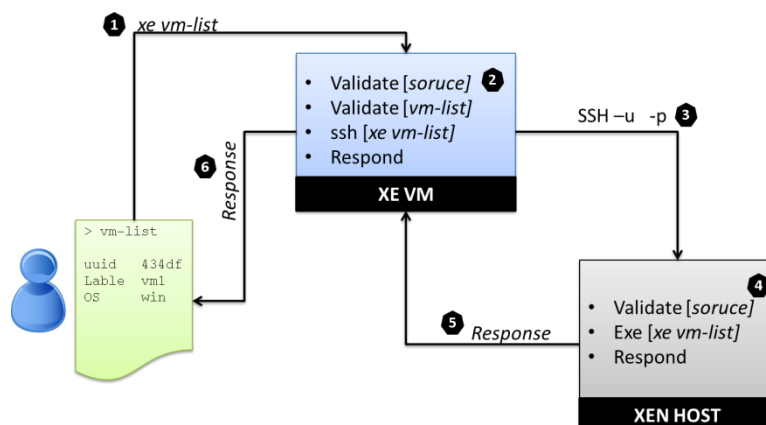
Figure 232 shows how the first phase is executed.



3. Input validation:

When a member sends a request, it must be validate so that the command is not malicious and comes from valid member. Before passing the request to Xen host, it is combined with the prefix (`xe -u superuser -pw password -s servername`). This prefix guarantees that a member can only execute xe commands. Then the request is transmitted to Xen host and executed. After that, the member receives the response coming from the Xen host. Thus, the protocol which is applied to this phase is:

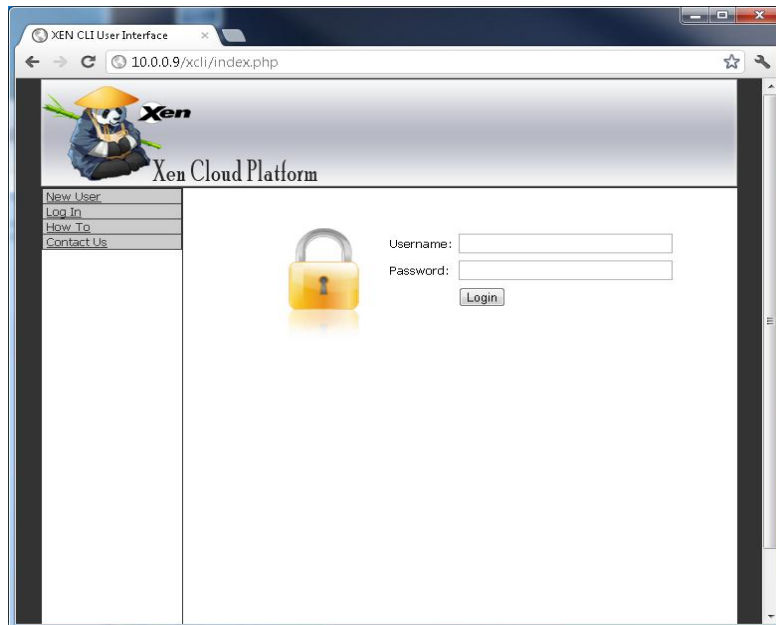
- Member → Web App: command
- Web App → Xen Host: valid xe-command
- Xen Host → Web App → Member: result



4. Test and Analysis:

Authenticating the Member:

End user must login to the system in order to lunch requests to control Xen host and VMs. Figure 32434 show the login page indicating that the user must enter his correct credentials.



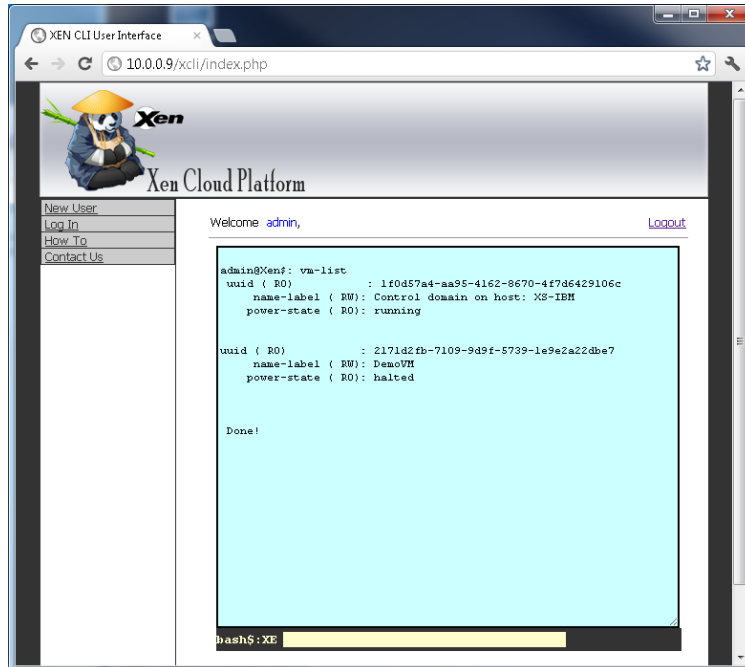
Establishing the session:

If the system is able to identify the user, a session will be created. During the session the member can lunch only *xe* commands.



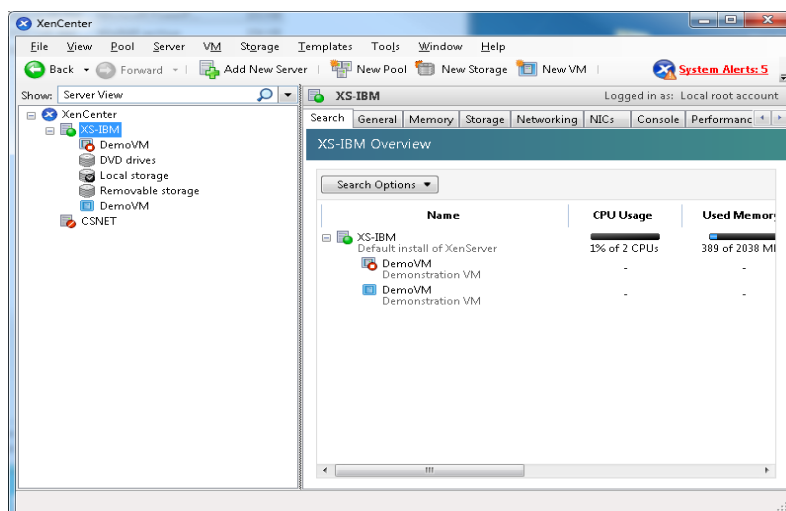
Executing xe commands:

When a *xe* command is lunched, it is validated and passed to the targeted Xen host. After getting a response from the Xen host, the results are displayed to the member. As Figure 0025202 indicates, Admin lunched [*vm-list*] command and the results are being displayed to him/her.



Controlling VMs:

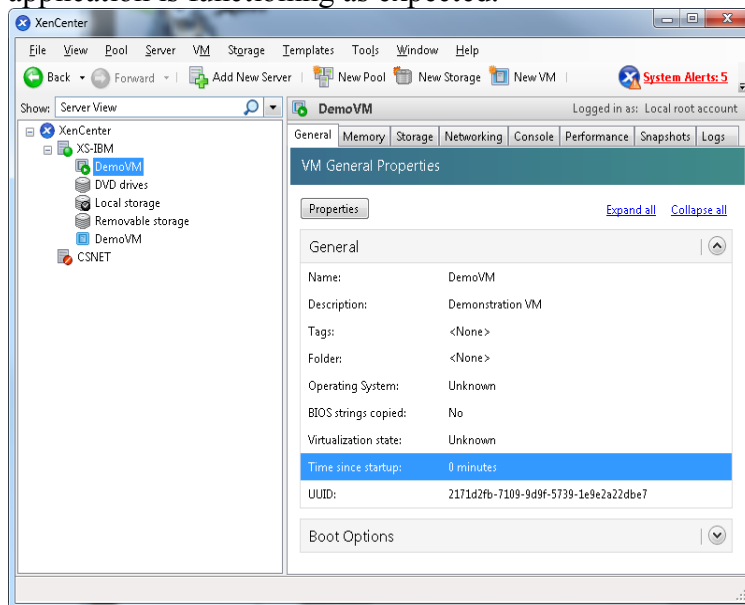
In this example, we tried to demonstrate how the member Admin is able to bring up a virtual machine and stop it. Figure 15855 shows how the virtual machine [DemoVM] is stopped.



Now, Admin wants to start up the virtual machine by lunched the request [*vm-start vm-DemoVM*]. After Admin lunched the request, he gets successful response as shown in figure 21212.



Figure 303 exhibit that the virtual machine is being stated up. That is, the web application is functioning as expected.



9. Lesson learned:

To be more effective in our approach, we have to maintain the approach API to be compatible with any new API's updates. Moreover, as in our goals, we want to avoid an unwanted command, and therefore the approach command validation must be strong and support two parameters validation.

10. References:

- [1] www.xen.org
- [2] www.kernelthread.com
- [3] www.TechnologyTrends.co,
- [4] <http://www.wikipedia.org/>
- [5] <http://uod.ac/it/wp-content/uploads/2011/04/Xen-Virtualization-APractical-Handbook.pdf>
- [6] <http://uod.ac/it/wp-content/uploads/2011/04/Xen-Virtualization-APractical-Handbook.pdf>
- [7] Citrix Documentation
<http://docs.vmd.citrix.com/XenServer/4.0.1/installation/ch01s03.html>
- [8] What Citrix XenServer add to Xen Virtualization
- [9] Citrix Community: <http://community.citrix.com/display/xs/XenCenter>
- [10] Linux ManPage: <http://man.cx/xen-create-image%28%29>