

Xen Cloud Platform 0.1 User Security

0.1

Published October 2009

1.0 Edition

Xen Cloud Platform 0.1 User Security

Published October 2009

Copyright © 2009

Xen, XenSource, XenEnterprise, Xen Cloud Platform, XenExpress and logos are either registered trademarks or trademarks of Citrix Systems, Inc. in the United States and/or other countries. Other company or product names are for informational purposes only and may be trademarks of their respective owners.

Contents

1. Introduction	5
Architecture	5
2. Host security	7
Networking Configuration	7
Administration Network	7
Virtual Machine Network	8
Storage Configuration	9
Raw storage	9
Converting storage to VHD mode after upgrading	10
Accessing Xen Cloud Platform Hosts	10
Verifying Host Identity	10
Replacing the SSL certificate of a host	10
Updating the Xen Cloud Platform Host	11
Hotfix Format	11
Updating using the CLI	11
3. Guest Security	13
Hypervisor Protection	13
Guest Communication	13
Guest Consoles	14
Operating System Recommendations	14
Microsoft Windows	14
CentOS 4	14
CentOS 5	15
RHEL 5	15

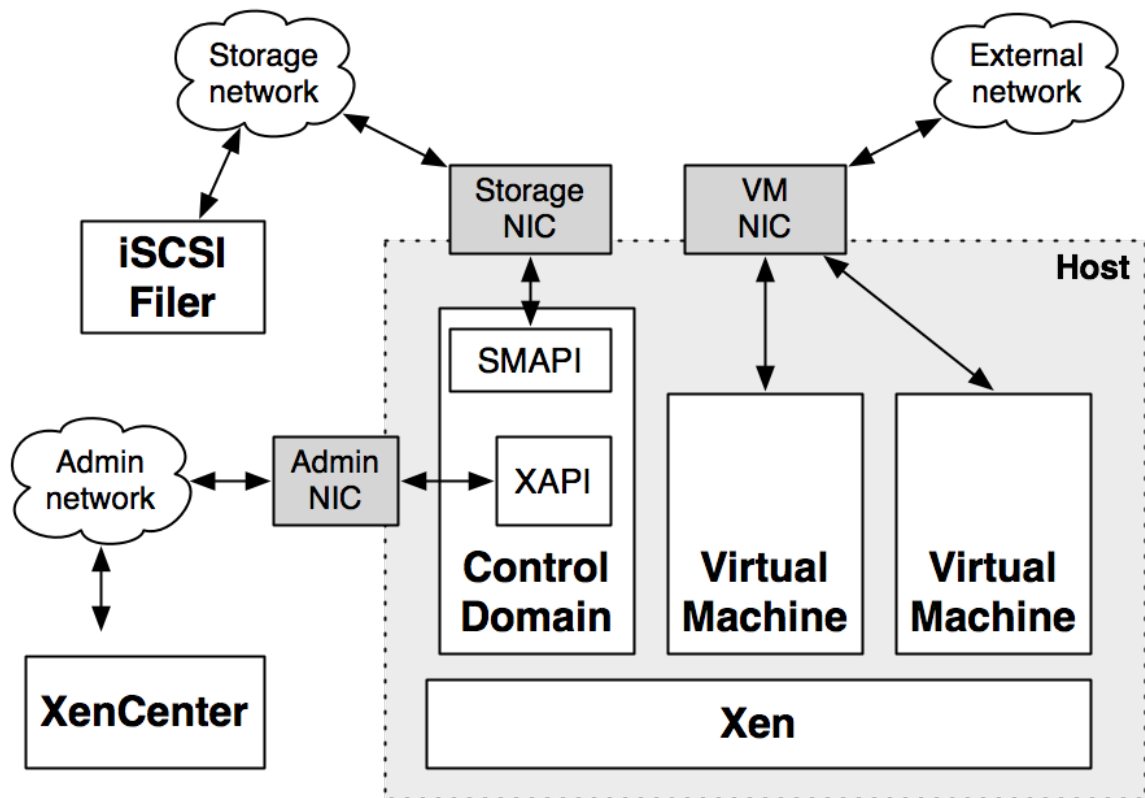
Chapter 1. Introduction

Xen Cloud Platform is a server virtualization platform that can run multiple operating systems simultaneously. This consolidation both reduces the operational costs and increases the agility associated with running IT infrastructure. Although Xen Cloud Platform aims to be secure by default and requires little out-of-the-box configuration with respect to security, you should be aware of how it works and how to maintain a secure installation. This document provides an outline of the security architecture of Xen Cloud Platform, and pointers to maintenance procedures and best practices for keeping your Xen Cloud Platform deployment secure and reliable.

Architecture

Xen Cloud Platform consists of several different components:

- The *Xen hypervisor* is the first software to load when Xen Cloud Platform boots. It runs in 64-bit mode and virtualizes the CPUs, interrupts and host memory. Xen is a very thin layer of software which does not have any device drivers beyond the serial port, and consists of around 150,000 lines of code.
- The *control domain* boots next, which is a 32-bit Linux-based embedded distribution. The control domain is a normal Xen Cloud Platform VM that has additional privileges granted to it which allows it to control host hardware devices and also create further guest domains.
- The *XAPI management stack* runs inside the control domain and manages all resources required for running guest domains. It consists of a distributed database and control software which listens on the administration interface for XenAPI clients that issue control instructions.
- The *Storage Manager* (or SMAPI) runs inside the control domain and provides a consistent interface to a variety of storage backends, such as Fibre Channel, iSCSI, file-based VHD disks, or local storage.
- Xen Cloud Platform can run multiple *Virtual Machines* on the same host, each of which provides entirely isolated computation, storage and networking to the operating system running inside of it.
- Finally, multiple Xen Cloud Platform hosts can be aggregated into a *resource pool* which acts as a single unit of administration across a cluster of machines.



These concepts are illustrated in the above diagram which shows the components in a single physical host. The following chapters describe:

- how to secure the Xen Cloud Platform host itself by ensuring that the control domain is correctly configured and patched
- best practices for storage configuration
- best practices for securing VMs
- XenCenter represents any third party management tool

Chapter 2. Host security

The Xen Cloud Platform control domain provides administrative access to a resource pool. If access to this control interface is compromised, attackers could control virtual machines and the storage and networking layers.

Networking Configuration

The control domain is initially configured during installation (see the section called “Installing the Xen Cloud Platform host” in *Xen Cloud Platform Installation Guide*) and subsequently by using the CLI. There are three distinct types of networks which may be configured in the control domain: the administration network, the storage network, and the VM network.

The three networks can also run on the same physical NIC, and are initially set up like this after installation. For maximum security, you can assign a set of NICs for dedicated use for administration, storage and VM traffic. These dedicated NICs can also be paired up as NIC bonds to ensure maximum resilience against physical component failure. See Bonding two NICs together in *Xen Cloud Platform Administrator's Guide* for more information.

Administration Network

The administration network is used for the following functions:

- control requests from the XenAPI
- Virtual Machine live relocation (or *XenMotion*)
- VM import, VM export, hotfix application, and resource pool metadata backups.
- sending e-mail alerts
- intra-resource pool communication

The XAPI tool-stack listens on port 80 (plain-text) and port 443 (SSL encrypted) for XenAPI requests. To be sure that all traffic is encrypted, add firewall rules to the administration router to block external requests from port 80.

The XAPI tool-stack itself is written in a high-level, statically type-safe language known as Objective Caml (or *OCaml*). This guarantees that it is free from low-level memory corruption issues such as buffer overflows or integer overflows, making it much more robust against malicious attacks over the administration network. The SSL layer uses the popular `stunnel` package to provide industry-standard SSLv3 encryption.

VM live relocation involves transferring the memory image of the VM while it is still running. Since a high-performance transfer will minimize the performance impact on the running VM, and live relocation is only supported between machines on a local network, this transfer occurs in plain-text over port 80. If you configure XenMotion across WAN links you will need to use IP-level security (for example, IPsec) to encrypt the memory image.

It is possible to not bind an IP address to the administration network interface, which will mean that none of the administration functions will work from outside of the local console

on the Xen Cloud Platform host. Be aware that in this configuration you will not be able to create resource pools, import/export VMs, or otherwise take advantage of features such as e-mail alerting.

Resource Pools

When using Xen Cloud Platform, multiple hosts can be clustered together into *resource pools*. These resource pools are assigned a *pool master* which controls all the other hosts. All communication between resource pools is done over SSL, and hosts authenticate themselves to each other using a randomly generated symmetric key that is created at the time of pool creation.

A common way to isolate this administration traffic is to use Ethernet VLANs to segregate it from other non-administration traffic. You can configure your routers to tag all traffic from the administration NICs with an administration VLAN tag. This VLAN can also be used for other appliance control traffic in your server farm, such as Citrix Provisioning Server or Citrix NetScaler.

Storage Network

If you are using IP-based storage backends, such as an NFS or iSCSI SR, the storage traffic also flows through the control domain. Storage traffic is not encrypted, and so it is important to isolate this traffic from other administration traffic for both reliability and security.

Dedicated storage NICs require an IP address that must be on a different IP subnet to the main administration interface.

In the case of iSCSI traffic, the software OpeniSCSI initiator is used to connect to iSCSI targets. This initiator fully supports CHAP to authenticate to the remote target, and you should configure this on your iSCSI filer if other third-parties should also be able to connect to the filer and access the VM data.

The file-based NFS storage repository mounts the specified NFS repository on the control domain. Each SR is represented as a directory on the remote NFS server, with each virtual disk being a file in that directory stored in the VHD file format. VHD is not an encrypted file format, and so you should ensure that only the Xen Cloud Platform hosts and authorized administrators can mount the filesystem. For further security, ensure that the storage interfaces of the remote filer are not visible outside of the dedicated storage network.

Virtual Machine Network

The VM network does not require an IP address in the control domain. Instead, VM network packets are *bridged* at the Ethernet layer over the host NIC assigned to the virtual network interface in the VM. This bridge acts as an Ethernet switch, ensuring traffic from VMs are isolated from each other at Layer 2.

A common configuration is to have different virtual network interfaces inside a VM for "front end" traffic (for example, a web server) and "back end" traffic (for example, a database). This traffic is best isolated by using VLANs, which will tag the Ethernet traffic separately

but still go over the same physical NIC on the host (see the section called “VLANs” in *Xen Cloud Platform Administrator’s Guide*).

Some specialized VMs need to see all the traffic on their network segment, such as Intrusion Detection Systems. It is possible to switch a virtual or physical interface into promiscuous mode by toggling a parameter using the XE command-line interface (see the section called “Miscellaneous settings” in *Xen Cloud Platform Software Development Kit Guide*).

Warning

Do not enable the promiscuous flag without good reason, as other VMs network traffic can be accessed by the VM in promiscuous mode.

Storage Configuration

Storage on the control domain is automatically configured during installation and subsequently by using the CLI. Xen Cloud Platform supports several types of storage repositories (SRs) and virtual disk image (VDI) types (see Chapter 3, *Storage in Xen Cloud Platform Administrator’s Guide* for a complete list of SR and VDI types).

Many of the SR-types available in Xen Cloud Platform 0.1 are designed so that, by default, data stored in a deleted VDI won’t be available in a new VDI, even when they occupy the same physical space on the underlying storage media. The following SR-types are designed like that:

- Citrix Storage Link Gateway (*cs/g*)
- EqualLogic Adaptor (*equal*)
- LVM over FC (*lvmohba*)
- LVM over iSCSI (*lvmoiscsi*)
- Local EXT3 VHD (*ext*)
- Local LVM (*lvm*) - default SR type for local storage.
- NFS VHD (*nfs*)
- NetApp Adaptor (*netapp*)

Raw storage

Some of the SR-types can be used in a mode where the VM has direct access to the underlying storage media. This is referred to as *raw mode*, as opposed to the default mode which is called *VHD mode*.

Please refer to the section called “sr-create” in *Xen Cloud Platform Administrator’s Guide* and the section called “vdi-create” in *Xen Cloud Platform Administrator’s Guide* for details on creating and using SRs in this mode. When using raw mode Xen.org recommends that you treat VDIs similarly to physical hard disks in respect to re-use and decommissioning of the space on the underlying media. If organizational rules mandate that physical hard disks must be securely deleted before re-use or disposal then this rule should be extended to also include VDIs on SRs used in raw mode.

Warning

Raw mode was the default mode for LVM-based SRs in all versions of Xen Cloud Platform prior to 0.1. The [section Converting storage when upgrading](#) below provides details on how to modify SRs and VDIs on an upgraded version of Xen Cloud Platform.

Converting storage to VHD mode after upgrading

Details on how to convert existing SRs from raw to VHD mode is covered in the section called “Storage Repository Types” in *Xen Cloud Platform Administrator's Guide*. After converting an SR it is necessary to convert the VDIs individually. Converting an old VDI can be done by creating a new VDI, making sure the type is VHD, and then, from inside a VM, copying the contents of the old VDI over to it.

Accessing Xen Cloud Platform Hosts

Each Xen Cloud Platform host generates a set of random cryptographic keys when it is installed. Each key is split into two portions: a *public* key which is displayed to clients, and a *private* key which is only known to the host and can be used to prove that it is the owner of the public key.

The two classes of keys generated by a Xen Cloud Platform host are for the Secure Shell (SSH) protocol, and for the XenAPI SSL network service. SSH is only used for advanced configuration of the control domain, and should not be needed in normal use. The SSL communication is used much more often.

The main use of these keys is to confirm that you are connecting to the correct host. Although the SSL communication is encrypted, you also need to ensure that the host you are communicating with is actually the one you think you are connected to.

Verifying Host Identity

Replacing the SSL certificate of a host

Xen.org recommends that you replace the default certificates on hosts by certificates adhering to the standards of the organization where Xen Cloud Platform is deployed. The replacement file must contain the certificate and private key in PEM format.

To install the new certificate, first move the current SSL certificate:

```
mv /etc/xensource/xapi-ssl.pem /etc/xensource/xapi-ssl.pem_orig
```

Then copy the new certificate into its place:

```
cp <cert>.pem /etc/xensource/xapi-ssl.pem
```

Optionally the certificate file may also contain Diffie-Hellman parameters. These can be generated using OpenSSL:

```
openssl dhparam 512 >> /etc/xensource/xapi-ssl.pem
```

After this reboot the Xen Cloud Platform host.

Updating the Xen Cloud Platform Host

Xen Cloud Platform hosts are updated by applying *hotfixes*. These hotfixes are cryptographically signed code bundles which update portions of the control domain. The control domain is an embedded Linux distribution based on the CentOS 5.1 distribution, with many customizations and modified packages.

Although the control domain ships with the `yum` update mechanism, it is disabled by default and should not be enabled. Using upstream CentOS 5 packages directly will conflict in many cases, and result in a non-functional product. The sole supported update mechanism for Xen Cloud Platform is through hotfixes issued by Xen.org.

Hotfix Format

Xen Cloud Platform hotfixes are cryptographically signed by Xen.org before being issued. When they are uploaded to a Xen Cloud Platform host, this signature is checked and any invalid hotfixes are immediately rejected. Any third-party modifications to a hotfix will result in the signature being invalidated, thus ensuring that malware cannot be introduced through a hotfix.

The hotfix also includes the following metadata embedded in it:

1. A unique UUID which identifies the hotfix.
2. A pre-check function which ensures that the hotfix is relevant to the Xen Cloud Platform host it has been uploaded to. This function commonly checks the version number and checksums of important files to ensure that it will apply cleanly.
3. Post-install *guidance* which represents actions which need to be taken to activate the hotfix after it has been applied, if any. These include host rebooting, restarting some guests (for example, all Windows guests), or restarting the management tool-stack.

The embedded edition of Xen Cloud Platform has a different update mechanism from the retail edition. It consists of two parts: a digital signature and a replacement filesystem image for the host. Although this generally makes the OEM hotfixes larger than retail hotfixes, it also has some advantages. When an OEM hotfix is applied, it is unpacked into a second partition, and so OEM hotfix can be reverted by switching back to the backup partition if required.

Updating using the CLI

The XE command-line interface also provides full support for applying hotfixes, which is a good way to integrate Xen Cloud Platform hotfix management with any existing configuration management software you may be using for other infrastructure.

The hotfix must be installed on all Xen.org Xen Cloud Platform Hosts hosts, including each host in a resource pool. For more details, please refer to the section called “Applying updates using the CLI” in *Xen Cloud Platform Installation Guide*.

Chapter 3. Guest Security

The primary rule for guest operating systems running within Xen Cloud Platform is to ensure that you follow normal security practice on those guests. Install virus scanners within Windows, activate packet filters and firewalls, and keep your packages up-to-date in Linux distributions.

This chapter describes the protection that Xen Cloud Platform uses to isolate VMs from each other.

Hypervisor Protection

Xen Cloud Platform runs as a 64-bit hypervisor that runs guests in two modes: para-virtualized and hardware assisted. In both modes, the hypervisor provides strong isolation against CPU instructions running in one guest affecting the state, including memory, of another guest.

Paravirtual (or "PV") guests run kernels which have been specially modified to be Xen Cloud Platform-aware. Xen Cloud Platform provides a *hypercall* interface which is used by PV guests to communicate with the hypervisor. In a 32-bit kernel, the Xen hypervisor is permanently mapped into the top of the address space, and runs in ring-0. The guest kernel runs in ring-1 and is modified to make Xen hypercalls to modify its page tables. Xen Cloud Platform validates all page table updates to ensure that guests cannot see each other's memory pages. The user-space runs in ring-3 as normal, and does not need any modification.

Hardware assisted mode (or *HVM*) uses the Intel VT or AMD-V hardware extensions to offload the effort of virtualization onto the hardware. Guests in HVM mode see a complete physical machine, and the hypervisor is not actually present within the virtual memory area. Instructions which normally require privileged-level access are virtualized by the hardware, and the hypervisor uses new instructions (for example `VMENTER`) to switch in and out of the VM.

HVM mode does not automatically provide block and network devices; instead, these are initially emulated by a `qemu-dm` helper process which runs in the control domain. To ensure that bugs in this program do not compromise the control domain, Xen Cloud Platform runs each `qemu-dm` inside a Linux chroot with a unique unprivileged process ID. Later on in the boot process, the high-performance paravirtual drivers are activated which switch away from the emulated devices to high-speed virtual channels which use a similar mechanism to para-virtualized guests to communicate with the outside world with minimal overhead.

Guest Communication

Guests communicate control information and flags to and from the control domain through a tree known as *Xenstore*. The Xenstore process runs in the control domain and has a comprehensive Access Control List (ACL) mechanism which limits read, write and access permissions to various portions of the tree. When VMs are started, the control domain writes some control values into Xenstore which are read-only to the guest, and the guest can write into a sub-tree of its namespace in order to communicate information back to the control

domain (for example, its IP address, or performance metrics). Access control is used to make sure that the guest's sub-tree isn't readable by other guests.

Xen Cloud Platform uses a specially hardened version of Xenstore which has several improvements over the open-source version. Guests are rate-limited in terms of the number of writes to their tree, the control domain is scheduled in preference, and handles to Xenstore can be created with more restrictive permissions. See the section called "Security enhancements" in *Xen Cloud Platform Software Development Kit Guide* for more information on these improvements.

Guest Consoles

Access to VM consoles is provided over the VNC protocol. The exact mechanism varies depending on whether the guest is running in PV or HVM mode (see the section called "VM console forwarding" in *Xen Cloud Platform Software Development Kit Guide* for details). In the case of PV guests, a `vncterm` process in the control domain makes the text console available over VNC for rendering. The `vncterm` process runs in a Linux chroot with a unique process ID to ensure that bugs in the console emulation do not result in a compromise of the control domain.

For all guests, the VNC server is exposed over a TCP port bound to the `localhost` interface in the control domain. The XAPI toolstack then exposes this interface over the XenAPI securely over SSL, ensuring only authorized users can connect to the guest. Note that any user logged into the control domain who can access the `localhost` interface will be able to access all the VM consoles, so be extremely careful about granting shell access to the control domain to untrusted users.

Operating System Recommendations

This section covers security recommendations specific to a particular guest operating system.

Microsoft Windows

Xen Cloud Platform includes a set of WHQL-certified disk and network drivers for all supported versions of Windows. Ensure that you have the most up-to-date drivers installed in the guest.

As always, ensure that you also configure Windows Update to regularly download critical security updates from Microsoft to prevent conventional malware from corrupting your guest.

CentOS 4

Run the following commands as root in the guest:

```
yum update
reboot
```

Follow instructions on the screen regarding accepting RPM keys as necessary.

CentOS 5

Run the following commands as root in the guest:

```
yum update  
reboot
```

Follow instructions on the screen regarding accepting RPM keys as necessary.

RHEL 5

Run the following commands as root in the guest:

```
yum update  
reboot
```

Follow instructions on the screen regarding accepting RPM keys as necessary.