Common Criteria Security Target

For

Citrix XenServer 5.6 Platinum Edition

Version 1-0   30 July 2010

Summary of Amendments

Version 1-0 30 July 2010

First public version.

0. Preface

0.1 Objectives of Document

This document presents the Common Criteria (CC) Security Target (ST) to express the security and evaluation requirements for the Citrix XenServer ® 5.6 Platinum Edition product.

The product is designed and manufactured by Citrix Systems Inc. (http://www.citrix.com/).

The Sponsor and Developer for the EAL2 (augmented with ALC_FLR.2) evaluation is Citrix Systems Inc.

0.2 Scope of Document

The scope of the Security Target within the development and evaluation process is described in the Common Criteria for Information Technology Security Evaluation [CC]. In particular, a Security Target defines the IT security requirements of an identified TOE and specifies the functional and assurance security measures offered by that TOE to meet stated requirements [CC1, Section C.1].

Security Functional Requirements (SFRs), as defined in [CC2], are the basis for the TOE IT security functional requirements expressed in this Security Target. These requirements describe the desired security behaviour expected of a TOE and are intended to meet the security objectives as stated in this Security Target. Security Functional Requirements express security requirements intended to counter threats in the assumed operating environment of the TOE, and cover any identified organisational security policies and assumptions.

0.3 Intended Readership

The target audience of this ST are consumers, developers, evaluators and certifiers of the TOE, additional information can be found in [CC1, Section 6.2].

0.4 Related Documents

Common Criteria[1]

[CC1] Common Criteria for Information Technology Security Evaluation,
Part 1: Introduction and General Model,
CCMB-2009-07-001, Version 3.1 Revision 3, July 2009.

[1] For details see http://www.commoncriteriaportal.org/

[CC2] Common Criteria for Information Technology Security Evaluation,
Part 2: Security Functional Components,
CCMB-2009-07-002, Version 3.1 Revision 3, July 2009.

[CC3] Common Criteria for Information Technology Security Evaluation,
Part 3: Security Assurance Components,
CCMB-2009-07-003, Version 3.1 Revision 3, July 2009.

[CEM] Common Methodology for Information Technology Security Evaluation,
Evaluation Methodology,
CCMB-2009-07-004, Version 3.1, Revision 3, July 2009.

Developer documentation

[CCAG] "Common Criteria Administrator's Guide for Citrix XenServer 5.6, Platinum Edition", 1.0 Edition, August 2010

[CCECG] "Common Criteria Evaluated Configuration Guide for Citrix XenServer 5.6, Platinum Edition", 1.0 Edition, August 2010

0.5 Abbreviations

Acronym	Meaning
EPT	Extended Page Tables
NIC	Network Interface Card
NTP	Network Time Protocol
OS	Operating System
OSP	Organisational Security Policy
PAM	Pluggable Authentication Modules
SAR	Security Assurance Requirement
SFR	Security Functional Requirement
SSL	Secure Sockets Layer
ST	Security Target
TLS	Transport Layer Security
TOE	Target of Evaluation
TSF	TOE Security Functionality
VM	Virtual Machine

See [CC1] for other Common Criteria abbreviations.

0.6 Glossary

Term	Meaning
Assurance	Grounds for confidence that a TOE meets the SFRs [CC1].
dom0	See Domain 0.
domU	See Domain U.
Domain	A running instance of a virtual machine. (In most parts of this Security Target the terms 'domain' and 'virtual machine' can be used interchangeably.)
Domain 0	A special-purpose domain (based on a Linux kernel) that exists in a single instance on each XenServer host. Domain 0 is the only privileged domain (meaning that it can use privileged hypervisor calls, for example to map physical memory into and out of domains) on a XenServer host, and is thus the only domain that can control access to physical input/output resources directly and access the content of other domains (i.e. Domain U). In contrast to the HVM domains in which HVM Guests run, which are not aware that they are running on a virtualised platform, dom0 is necessarily a 'PV domain' (cf. PV Guest) which is aware of the virtualised environment.
Domain U	The collection of domains other than Domain 0. Each of these domains is either an HVM Guest or PV Guest, and is a domain in which a guest operating system has been (or will be) installed. (Only HVM Guests are included in the evaluated configuration under this Security Target.)
Domain U Guest	An HVM Guest or PV Guest. (Only HVM Guests are included in the evaluated configuration under this Security Target.)
Guest Operating System (Guest OS)	An operating system that has been installed in a Guest Domain. (Windows is the only Guest OS included in the evaluated configuration under this Security Target.)
Guest OS User	A user of a Guest OS, including both ordinary users and administrators of the Guest OS.
Host	An installation of XenServer on a dedicated server.
HVM Guest	A member of domU in which an unmodified Guest OS can be installed and run. This is contrasted with a PV Guest. (Only HVM Guests are included in the evaluated configuration under this Security Target.)
Hypercall	Synchronous calls made from a domain to the hypervisor. Any domain may make calls to the hypervisor, but only dom0 can make privileged calls, such as those that cause memory (including memory representing physical resources) to be mapped into or out of domains.

Term	Meaning
Hypervisor	An abstraction layer implementing a set of software calls (hypercalls) that can be made by domains, and providing an asynchronous event-based interface for communication from the hypervisor to domains. The hypervisor controls the scheduling of the CPU and the partitioning of memory between virtual machines, but has no knowledge of the actual physical devices on the host (when the devices are used, this knowledge is provided by device drivers running in dom0).
License Server	A server that validates licenses for XenServer.
NFS	A protocol developed by Sun Microsystems, and defined in RFC 1094, which allows a computer to access files over a network as if they were on its local disks.
Pluggable Authentication Module (PAM)	A library used to provide a common authentication service to Linux programs.
Pool	A group of hosts in which one host takes the role of master and the others are slaves. Storage and configuration metadata are shared across the pool. The master can decide which hosts to start VMs on.
PV Drivers	Drivers that replace default drivers in an HVM Guest, in order to accelerate storage and network data paths. These are treated as part of the Guest OS, use unprivileged XenServer interfaces, and are not involved in implementing XenServer security functions.
PV Guest	A member of domU in which a modified Guest OS can be installed and run: the modifications make the Guest OS aware that it is in a virtualised environment in which other virtual machines are running on the same host, and in which it does not have direct access to the physical networking and storage resources. (PV Guests are not included in the evaluated configuration under this Security Target.)
Secure Sockets Layer	An open, non-proprietary protocol that provides data encryption, server authentication, message integrity and optional client authentication for a TCP/IP connection.
Target of Evaluation	A set of software, firmware and/or hardware possibly accompanied by guidance. [CC1]
TOE Security Functionality	A set consisting of all hardware, software, and firmware of the TOE that must be relied upon for the correct enforcement of the SFRs. [CC1]
Transport Layer Security	The latest, standardised, version of SSL, providing server authentication, data stream encryption and message integrity checks.
VHD	A file format containing the complete contents and structure representing a virtual Hard Disk Drive

Term	Meaning
Virtual Machine	An abstraction of a real hardware machine that creates an environment in which software (typically an operating system) that would otherwise run directly on hardware as the only software to be executing can be run with the illusion of exclusive access to a set of physical resources. In XenServer a virtual machine is characterised by a defined set of resources (e.g. memory and storage capacities and available network connections). A virtual machine that has been allocated real resources and in which processes are running is a Domain.
VM Data	The ‘VM data’ of a particular VM comprises all data stored in host memory that is mapped into that particular VM (or domain).
XenAPI	The API for managing XenServer installations, i.e. for remotely configuring and controlling domains running on hosts in a XenServer pool.
XML-RPC	A protocol for sending Remote Procedure Calls (RPC) formatted as XML. (See www.xmlrpc.com)

See [CC1] for other Common Criteria abbreviations and terminology.

Contents

Figures / Tables

1. ST Introduction

1.1 ST and TOE Reference Identification

TOE Reference:	Citrix XenServer 5.6 Platinum Edition
ST Reference:	CIN3-ST-0001
ST Version:	1-0
ST Date:	30 July 2010
Assurance Level:	EAL2 augmented with ALC_FLR.2 Flaw Reporting Procedures
ST Author:	SiVenture

1.2 TOE Overview

1.2.1 Usage and major features of the TOE

The TOE is Citrix XenServer 5.6 Platinum Edition installed (abbreviated in this document to "XenServer").

XenServer is a server virtualisation product that runs directly on server hardware and establishes an environment comprising a number of virtual machines (or "domains"), each configured to operate with a set of virtual CPU, memory, storage, and network resources (see Figure 1 in section 1.3). In this way, a single physical server can present a number of separate logical servers, with each server acting as though its resources were independent and running applications on a typical Windows operating system[2]. XenServer maps and schedules the virtual resources onto the physical resources of the server hardware, and thereby provides a number of potential advantages including increased utilisation of the physical server resources.

The structure and operation of the TOE is described in more detail in section 1.3.

1.2.2 Required non-TOE hardware and software

The TOE is installed on one or more dedicated x86 servers with the following characteristics:

[2] XenServer supports installation and operation of a variety of Windows and Linux guest operating systems (see section 1.3 for more explanation of guest operating systems), but only Windows installations are supported by the evaluated configuration in this Security Target.

- Servers each contain more than one CPU core[3]
- Processor type: 64-bit Intel-VT with EPT
- At least 3 NICs per host

The TOE is required to be connected to the following non-TOE components:

- Storage: VHD on NFS
- Citrix License Server
- NTP server.

A XenServer installation will also have a guest operating system installed in each Domain U VM[4], and these guest operating systems are not included in the scope of the TOE. The evaluated configuration applies only to HVM guests, which run various versions of Windows as the guest operating system. After initial installation, each guest operating system image is modified by installing paravirtualised device drivers known as the Citrix Tools for Virtual Machines (these are also known as “PV drivers”, and are discussed further in section 1.3). These drivers, which improve the performance of the guest operating system, are also outside the scope of the TOE.

1.3 TOE Description

XenServer is a server virtualisation product that runs directly on server hardware and establishes an environment comprising a number of virtual machines (or “domains”), each configured to operate with a set of virtual CPU, memory, storage, and network resources (see Figure 1).

[3] Where only one CPU core is available then different code paths are used in the TOE, and these were not tested in the evaluated configuration.

[4] See section 1.3 for a description of the TOE which explains Domain U and other terms used in this paragraph.

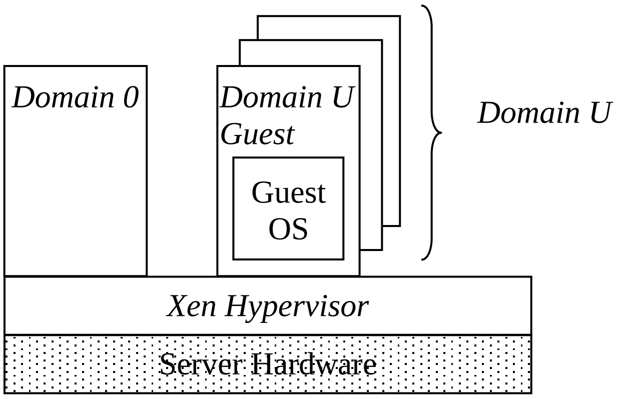

Figure 1: Illustration of XenServer components

One of these domains, named "Domain 0" or "dom0", has special status and is in effect the part of the TOE which controls the possible accesses from all other domains to physical resources[5]. In each of the other domains (referred to collectively as "Domain U" or "domU", or individually as a "Domain U Guest") an operating system such as Windows is installed, and the domain will then behave as a separate server. In the XenServer TOE the guest operating system is not aware that it is running inside a virtual machine, it simply accesses resources that it believes to be physical resources but which are actually virtual resources defined in its domU guest[6].

The Xen Hypervisor provides a basic abstraction layer on top of the hardware. It is responsible for CPU scheduling, and access by a domain to the memory that has been allocated to it. Although domU guests access the Hypervisor, in XenServer only dom0 can execute the privileged hypervisor commands that map domain memory (from virtual to physical) in order to enable access to physical resources[7].

It was stated above that in the XenServer TOE the guest operating system is not aware that it is running inside a virtual machine, and this situation deserves some more detailed clarification. For XenServer in general, a domU guest is either a PV guest or an HVM guest. A PV guest is aware that it is running as a virtualised environment with other virtual machines present, and with no direct access to hardware. Creating a PV guest therefore involves making modifications to the guest operating system itself to enable the OS to run with this awareness. These modifications are affected via virtualization-aware disk and

[5] Memory is accessed directly by domU, but only using tables set up by dom0 (using privileged hypervisor calls) that control which memory can be used by domU.

[6] The virtual resources include storage, network and display (and some others). The CPU is shared, so that instructions in domU are scheduled by the Xen hypervisor, and then run directly (as native instructions) on the CPU.

[7] Note that in the wider Xen community domains other than dom0 can be privileged. However, in XenServer dom0 is the only privileged domain.

network device drivers that are installed within the guest OS. These drivers communicate with services in dom0 that read/write disk blocks and send/receive network packets on behalf of the guest. In effect the domU guest provides one ‘half’ of a driver for the storage or network resource, which is then linked to its other ‘half’ in dom0.

An HVM guest, by contrast, runs without modification of the OS, and the guest OS is not aware of its virtualised environment. Although an HVM Guest could run completely unmodified in this way, in practice some of the default device drivers are replaced with paravirtualised drivers (PV drivers) to improve performance. The PV drivers use unprivileged XenServer interfaces to accelerate the storage and network data paths, and are not involved in implementing any of the XenServer security functions.

XenServer supports both PV and HVM guests, but only HVM guests are included in the TOE; no trust is placed in the PV drivers (known as the Citrix Tools for Virtual Machines) and hence they are not included in the scope of the TOE but are present in the evaluated configuration.

In a virtualised situation such as this, it is possible that different domU guests will run software and deal with data that has different security requirements. Although the specific guest security requirements will be defined (and implemented) by the Guest OS, application and end-users, it is important that XenServer supports situations in which the security requirements of its domains are different. The fundamental property defined in this Security Target is therefore the separation of resources between domains, such that the processing in any one domain is protected from unauthorised access by any other domain. The security of software running in a domU guest remains the responsibility of the user and/or administrator of the guest (e.g. to maintain appropriate patch states for software, and virus protection within the domain).

A physical server with XenServer installed is referred to as a “host”, and a number of hosts may be logically linked together to create a “pool”, which enables them to benefit from shared storage (hence enabling a requirement for a new VM to be satisfied by any of the hosts in the pool). A pool is structured so that one of the hosts is the master (which maintains data about the pool and establishes any required communication paths between hosts) and the others are slaves. However, if the master is lost then it is possible for any of the slaves to become a replacement master.

The interfaces operated by XenServer hosts are illustrated in Figure 2. Note that the physical protection boundary in the diagram represents the parts of the TOE, and its connected storage, that must be protected by physical and procedural security to prevent unauthorised access (cf. OE.Secure_Resource in section 4.2.1).

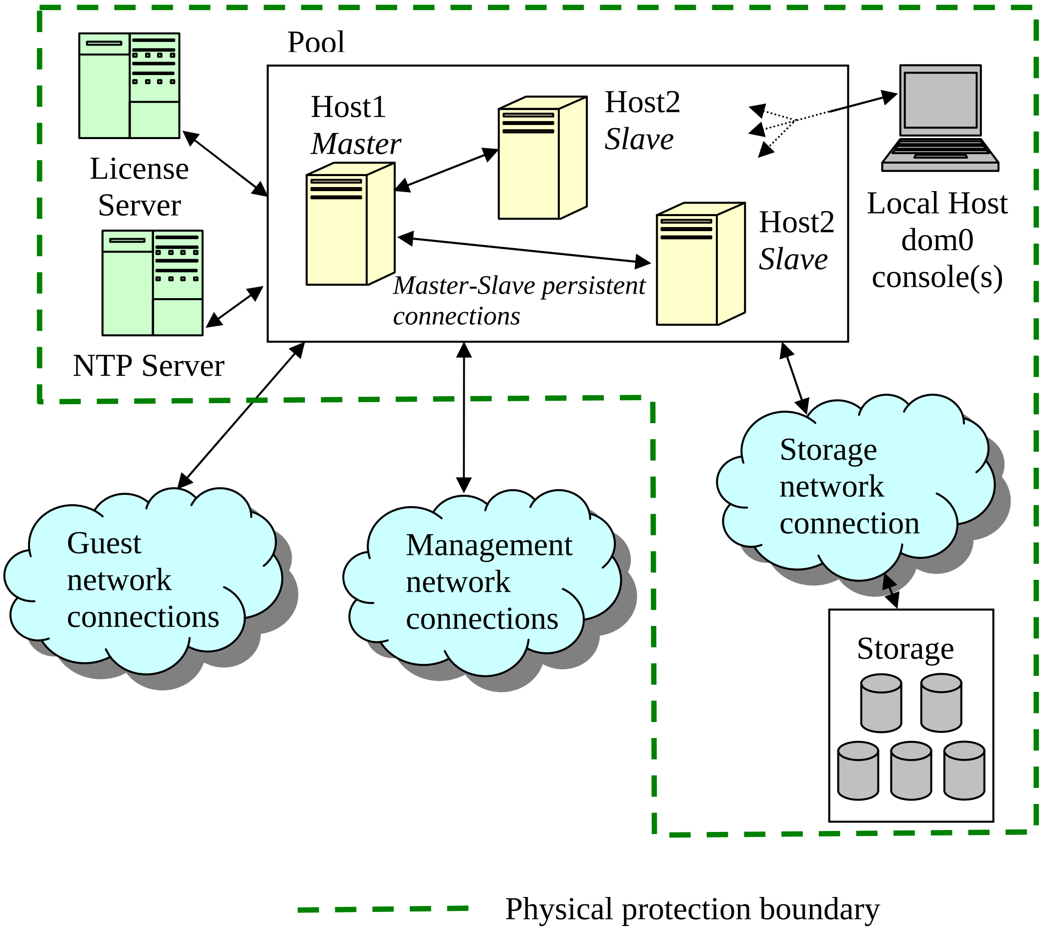
Pool
Host1
Master
Host2
Slave
Host2
Slave
Master-Slave persistent connections
License Server
NTP Server
Local Host dom0 console(s)
Guest network connections
Management network connections
Storage network connection
Storage
Physical protection boundary

Figure 2: XenServer Interfaces

The connections, and their basic protection measures, are as follows:

- Master-Slave persistent connections provide for communication about the pool and its state between members of the pool. While this connection is separately identified on functional grounds, its traffic travels over the management network (see below).

  The confidentiality and integrity of master-slave database traffic is protected by the use of TLS for these connections. Authentication is based on use of a secret shared between the hosts in the pool.

- Management network connections carry traffic relating to the management (configuration and control) of hosts, using a specific set of commands sent using XML-RPC over a specific application programming interface called XenAPI, or using one of a variety of "bulk data transfer services" and "interactive services" (these services, which include local console access and VM console access, are session-based and use the HTTP protocol). Communication with the License Server also takes place over this network. The management network uses a dedicated NIC on each host.

The confidentiality and integrity of management network traffic (other than the License Server and NTP server traffic discussed in section 1.4.2) is protected by the use of TLS for these connections – this is necessary because the general management activities can be carried out from remote terminals. Authentication is based on session credentials (i.e. a username/password combination is used to establish a session, with the credentials being checked by the PAM in dom0 on the relevant host) for XenAPI and bulk data transfer/interactive services.

- Storage connections provide a route between dom0 on a host and the physical storage devices available to the pool[8]. This connection therefore deals with both TSF data and user data stored and retrieved from the guest OS.

  The confidentiality and integrity of storage traffic is achieved by physical protection of the connections. The storage network is not accessible via the management or guest networks.

- Guest network connections are not used by dom0[9], but represent the networking resource available for use by each guest OS and its applications.

  As a general network resource, the guest network connection is not protected by the TOE. Any protection requirements will be based on the requirements of a guest OS and its applications, and are therefore the responsibility of the guest to provide.

These connections use a dedicated NIC in each host for each of the management and storage connections[10]. One or more additional NICs may be allocated on a host to provide the guest network connection.

1.3.1 Evaluated Configuration

The evaluated configuration of the TOE assumes the use of XenServer features indicated in the list below. ‘Base Product Features’ are options within the basic XenServer product which can be configured on or off, and which therefore need to be appropriately configured to achieve the evaluated configuration. ‘Separately Installed Features’ relate to items of software that are separately installed, and hence the list indicates whether or not the relevant item should be installed to achieve the evaluated configuration. Further details on installing the TOE and achieving the evaluated configuration are given in [CCECG].

[8] XenServer VMs can also make use of local storage, but the TOE excludes such use (the ability to use local storage is one part of the configuration data for a VM).

[9] In fact dom0 is responsible for switching guest network packets at level 2 to route them to guests, but dom0 does not use the guest network for its own communications.

[10] The NIC for the management network is defined when XenServer is installed, and the NIC for storage is part of the configuration data for a host.

Feature	Included in Evaluated Configuration?
Base Product Features	
Native 64-bit Xen hypervisor	Yes
Windows guests	Yes
Linux guests	No
XenMotion live migration	No
Multi-server management	Yes
Active Directory integration	No
Enhanced Snapshot	No
Live Memory Checkpoint / Rollback	No
Dynamic Memory Control (Ballooning)	No
High availability	No
Role Based Administration	No
SNMP[11]	No
Separately Installed Features	
XenCenter management console[12]	Yes
P2V/V2V Conversions & OVF Support	No
Workflow Studio orchestration	No
Provisioning services – virtual servers	No
Integrated StorageLink – Advanced Storage Mgmt.	No
Dynamic Workload balancing	No
Provisioning services – physical servers	No
Lab management	No
Stage Management	No

It is also noted that the following aspects of operation are out of scope of the TOE:

[11] In the evaluated configuration SNMP is configured off, and is further prevented by firewall rules used by dom0 when routing network packets.

[12] The XenCenter management console is not included in the TOE (it does not implement any security functions, nor is it necessary for their operation), but is present in the evaluated configuration.

- XenServer domains running PV guests
- Guest operating systems (i.e. the operating systems that run inside the virtual machines (other than dom0) created by XenServer)[13,14]
- Protection of confidentiality, integrity and authenticity of data transmitted over the guest network (as noted in the description of Figure 2, this is the responsibility of applications and/or the guest operating system running in the domain.)

The following aspects are part of establishing the evaluated configuration (see [CCECG]):

- The TOE must be connected to a License server with a Citrix Essentials (Enterprise edition) license.
- DomU virtual machines are configured not to use local storage or other local devices (printers, CD-ROM drive, etc.)[15]
- The storage connection (NAS) is physically isolated from other networks (management network and guest network)
- Servers are configured to use a separate, dedicated NIC for management traffic (i.e. for XenServer administrative operations, such as use of XenAPI) and for storage traffic

(Also see section 1.2.2 for a list of non-TOE hardware and software that is required to operate the TOE.)

1.4 TOE Boundaries

1.4.1 Physical Boundary

The physical boundary of the TOE is that of the server hardware on which the TOE is installed.

[13] As noted in section 1.3, when installing Windows as a guest operating system, there is a step required to install paravirtualised device drivers (PV drivers). The PV drivers use unprivileged XenServer interfaces to accelerate the storage and network data paths. Furthermore, the PV drivers are not involved in implementing any of the XenServer security functions. Hence we do not place any trust in the PV drivers; they are not part of the TOE and are ignored for the purposes of this Security Target.

[14] Note that the evaluated configuration assumes the use of HVM but not PV guests, hence only a guest OS that is available as an HVM guest is supported by the evaluated configuration. The difference between HVM and PV guests arises because of the different ways in which they execute on the XenServer platform.

[15] This assumption relates to the use of local storage only by domU. Dom0 boots from local storage, and continues to use this local storage for the dom0 filesystem.

1.4.2 Logical Boundary

The logical boundary of the TOE is shown in Figure 3.

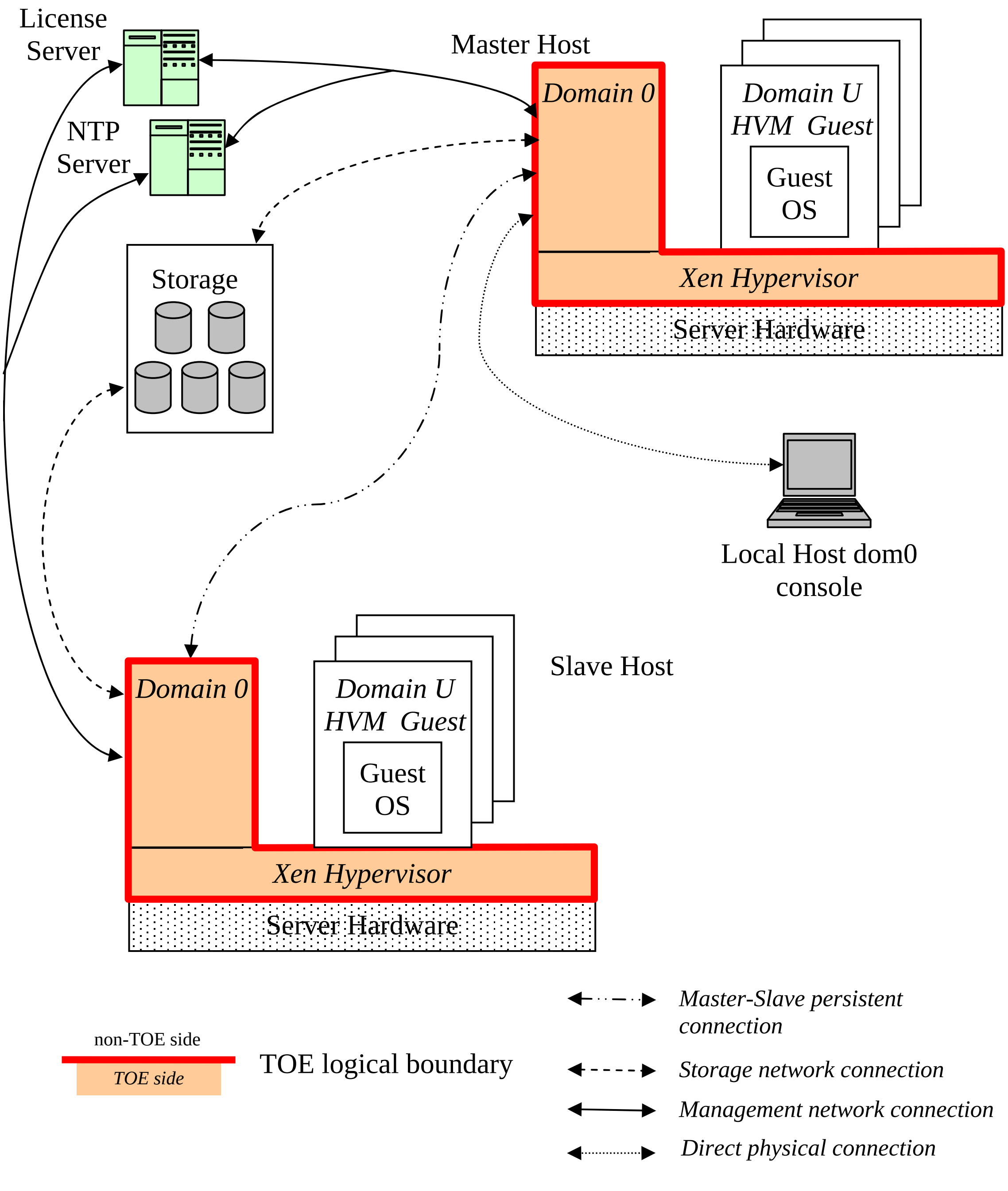

Figure 3: Illustration of TOE components and logical boundary

As shown above, the TOE includes the Xen Hypervisor and Domain 0. Guest operating systems (along with their PV drivers), and Domain U[16] are excluded from the TOE.

The protection of data on the various connections is described in section 1.3 (as noted in section 1.3, the connections to the License Server and NTP server are made over the management network but are not protected by TLS. The License Server and NTP server are assumed to be kept within a secure physical environment but are not responsible for implementing any of the TOE security, nor do they provide any operations that would threaten the Security Functional Requirements).

[16] Domain U is a mapping set up by Domain 0, but after it has been set up it acts simply as TSF data: it does not execute processes, and is not represented by subjects.

2. CC Conformance

As defined by the references [CC1], [CC2] and [CC3], this TOE conforms to the requirements of Common Criteria v3.1, Revision 3. The methodology applied for the evaluation is defined in [CEM].

The TOE is Part 2 conformant, Part 3 conformant, and meets the requirements of EAL2 augmented with ALC_FLR.2 Flaw Reporting Procedures.

This ST does not claim conformance to any PP.

3. Security Problem Definition

Note on terminology: in strict terms, a domain represents a running VM, but the terms ‘VM’ (or ‘virtual machine’) and ‘domain’ are used interchangeably in this section and the sections that follow.

3.1 Assets

The TOE protects data in each virtual machine (including dom0) from unauthorised access by other virtual machines, or by unauthorised users. dom0 is treated as having authorised access to any other domain[17], but no other pair of domU’s have authorised access to each other.

Thus VM data is the main asset identified for the TOE to protect. VM data requires protection in terms of both confidentiality and integrity.

However, the configuration data that defines a pool, a host, or a VM may also be relied on to support VM data separation, and is therefore identified as an additional asset. All configuration data is owned by dom0. This asset requires protection in terms of both confidentiality and integrity.

3.2 Users and Subjects

A single type of user is defined for the TOE:

XenServer Administrator — An administrator of XenServer, responsible for configuring and maintaining the TOE (including creation of pools of hosts and creation of virtual machines on those hosts according to certain configuration parameters). All XenServer administrators run as root in dom0.

Users of applications running under a Guest OS or of the Guest OS itself (i.e. within domU) – whether ordinary users or administrators of the Guest OS – are not considered as users of the TOE: they have no direct interaction with the TOE, and any indirect interactions are made through processes executing in the relevant domain.

The subjects in the TOE[18] are therefore:

- processes operating on behalf of XenServer administrators
- other processes.

[17] In fact dom0 communicates with other domains by the use of shared memory, and this limited access to dom0 data is obviously treated as an authorised access. Other pairs of domains do not share memory in this way.

[18] Guest OS’s are not part of the TOE and therefore processes in those OS’s are not subjects.

3.3 Threats

The following threats are to be countered by the TOE and its environment.

3.3.1 T.VM_Access Unauthorised access to data between VMs

A process executing on one VM might gain unauthorised access to read or modify the data belonging to another VM[19].

3.3.2 T.Intercept Unauthorised interception of communications

Communication channels on the management network might be intercepted by an attacker. This could lead to compromise of sensitive data in transit.

3.3.3 T.Mod_Conf_Data Unauthorised modification of configuration data

An attacker might make an unauthorised modification to configuration data associated with a pool, host or virtual machine.

3.4 Organisational Security Policies

No organisational security policies are defined for the TOE.

3.5 Assumptions

The following assumptions are made regarding the TOE:

3.5.1 A.Secure_Resource Physically secure IT resources

It is assumed that the following components of the TOE are kept physically secure so that no unauthorised persons have access to the components, either physically or for connection (e.g. via console ports):

- Hardware on which the TSF is running, and any connections between the hardware items (e.g. between hosts in a pool)
- The License Server[20]
- NTP server

[19] See footnote 17.

[20] Although this is not part of the TOE, it is assumed to be kept physically secure as a precaution, since it uses an unprotected communication channel to the TOE.

- Any local host dom0 console
- Storage devices used by the TOE, and their connections to the TOE.

These resources, and the protection boundary, are illustrated in Figure 2.

3.5.2 A.Separate_Networks Separated Networks

It is assumed that the storage connection and storage devices used by the TOE are physically isolated from the other networks used by the TOE, and that the management, storage, and guest networks each use separate NICs (more than one NIC may be used for the guest network).

4. Security Objectives

4.1 Security Objectives for the TOE

The security objectives for XenServer are defined as follows.

4.1.1 O.VM_Access Controlled access to data in VMs

The TOE shall protect the data associated with each VM, whether in memory or on disk, from unauthorised access (for reading or for modification) by processes executing in other VMs.

4.1.2 O.Admin_Access Controlled administrator access

The TOE shall ensure that only authorised XenServer administrators are given logical access to the TOE and its resources.

4.1.3 O.Secure_Traffic Protected network traffic

The TOE shall ensure the confidentiality and integrity of all data on the management network.

4.2 Security Objectives for the Operational Environment

The objectives that are required to be met by the TOE's operational environment are as follows.

4.2.1 OE.Secure_Resource Physically secure IT resources

The operational environment is required to ensure that the following components of the TOE are kept physically secure so that no unauthorised persons have access to the components, either physically or for connection (e.g. via console ports):

- Hardware on which the TSF is running, and any connections between the hardware items (e.g. between hosts in a pool)
- The License Server
- NTP server
- Any local host dom0 console
- Storage devices used by the TOE, and their connections to the TOE.

These resources, and the protection boundary, are illustrated in Figure 2.

4.2.2 OE.Secure_Keys Secure keys for communication security

The operational environment is required to ensure that all keys, public key certificates and other sensitive data used to support the confidentiality and integrity protection of the management network are managed securely (including generation, installation, storage and destruction as appropriate).

4.2.3 OE.Separate_Networks Networks are separated

The operational environment is required to ensure that the storage connection and storage devices used by the TOE are physically isolated from the other networks used by the TOE, and that the management, storage, and guest networks each use separate NICs (more than one NIC may be used for the guest network).

4.3 Security Objectives Rationale

The ways in which the threats are addressed by the security objectives are summarised in Table 1.

Threat/ OSP/ Assumption **Security Objectives**	**T.VM_Access**	**T.Intercept**	**T.Mod_Conf_Data**	**A.Secure_Resource**	**A.Separate_Networks**
O.VM_Access	X				
O.Admin_Access			X		
O.Secure_Traffic		X			
OE.Secure_Resource				X	
OE.Secure_Keys		X			
OE.Separate_Networks					X

Table 1: Threats/OSP/Assumptions addressed by Security Objectives

T.VM_Access is addressed by the requirement in O.VM_Access for separation of VM resources in memory or on disk.

T.Intercept is addressed by the protection of the confidentiality and integrity of the relevant data specified by O.Secure_Traffic. This is supported by the secure management of sensitive data (keys and certificates) in the environment.

T.Mod_Conf_Data is addressed by O.Admin_Access, which requires authentication of XenServer administrators before they are able to access the TOE and its resources.

A.Secure_Resource is addressed by OE.Secure_Resource, which specifically requires the physical protection of the relevant resources.

A.Separate_Networks is addressed by OE.Separate_Networks, which specifically requires the separation of the relevant networks.

5. IT Security Requirements

5.1 Conventions

The following conventions are used for the completion of operations:

- ~~Strikethrough~~ indicates text removed as a refinement and underlined text indicates additional text provided as a refinement.
- [**Bold text within square brackets**] indicates the completion of an assignment.
- [*Italicised text within square brackets*] indicates the completion of a selection.

5.2 Security Functional Requirements

The individual security functional requirements are specified in the sections below.

5.2.1 Administrator Authentication

The only users of the TOE are XenServer administrative users, who are required to authenticate before being given access to any operations.

FIA_UID.2 User identification before any action

Hierarchical to:	FIA_UID.1 Timing of identification
Dependencies:	No dependencies.

FIA_UID.2.1 The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

FIA_UAU.2 User authentication before any action

Hierarchical to:	FIA_UAU.1 Timing of authentication
Dependencies:	FIA_UID.1 Timing of identification

FIA_UAU.2.1 The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

Application note:

The users referred to in FIA_UID.2 and FIA_UAU.2 are XenServer administrators.

5.2.2 Protection of VM Data

The core requirement for the TOE is to prevent access to data held in a VM from access by another VM (apart from dom0, which has access to all VMs as part of its role in enabling domU VMs to use the physical resources on their host).

FDP_IFC.1/VMData Subset information flow control

Hierarchical to:	No other components.
Dependencies:	FDP_IFF.1 Simple security attributes

FDP_IFC.1.1/VMData The TSF shall enforce the **[VM data separation policy]** on **[all VMs, all VM data, and all operations]**.

FDP_IFF.1/VMData Simple security attributes

Hierarchical to:	No other components.
Dependencies:	FDP_IFC.1 Subset information flow control FMT_MSA.3 Static attribute initialisation

FDP_IFF.1.1/VMData The TSF shall enforce the **[VM data separation policy]** based on the following types of subject and information security attributes: **[VM processes and the identity of the VM that owns the VM data to which access is attempted]**.

Application note:

A VM is said to ‘own’ data when that data is ‘contained’ in a VM by virtue of being data in host memory that is mapped into that VM.

FDP_IFF.1.2/VMData The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold: **[the subject attempting to access the data is either the owner of the data or else is a process in dom0]**.

FDP_IFF.1.3/VMData The TSF shall enforce the additional information flow control rules: **[None]**.

FDP_IFF.1.4/VMData The TSF shall explicitly authorise an information flow based on the following rules: **[None]**.

FDP_IFF.1.5/VMData The TSF shall explicitly deny an information flow based on the following rules: **[None]**.

FDP_IFC.1/VDisk Subset information flow control

Hierarchical to:	No other components.
Dependencies:	FDP_IFF.1 Simple security attributes

FDP_IFC.1.1/VDisk The TSF shall enforce the **[VM disk separation policy]** on **[all domU VMs, all domU virtual disks, and all operations]**.

FDP_IFF.1/VDisk Simple security attributes

Hierarchical to:	No other components.
Dependencies:	FDP_IFC.1 Subset information flow control FMT_MSA.3 Static attribute initialisation

FDP_IFF.1.1/VDisk The TSF shall enforce the **[VM disk separation policy]** based on the following types of subject and information security attributes: **[domU VM processes and the identity of the VM that owns the virtual disk to which access is attempted]**.

Application note:

> A VM is said to 'own' a virtual disk when that virtual disk is defined and configured as a resource in that VM. In the case of read-only virtual disks, an administrator may configure the virtual disk as a resource in more than one VM.

FDP_IFF.1.2/VDisk The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold: **[the subject attempting to access the virtual disk is an owner of the virtual disk or else is a process in dom0]**.

FDP_IFF.1.3/VDisk The TSF shall enforce the <u>additional information flow control rules:</u> **[None]**.

FDP_IFF.1.4/VDisk The TSF shall explicitly authorise an information flow based on the following rules: **[None]**.

FDP_IFF.1.5/VDisk The TSF shall explicitly deny an information flow based on the following rules: **[None]**.

FDP_RIP.1 Subset residual information protection

Hierarchical to:	No other components.
Dependencies:	No dependencies.

FDP_RIP.1.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the *[deallocation of the resource from]* the following objects: **[memory mapped to a virtual machine]**.

5.2.3 Communications Protection

The TOE provides a secure channel for XenServer administrative operations (i.e. for the management network connection), providing authentication of the communicating parties, and confidentiality and integrity of traffic sent on the channel.

FTP_ITC.1 Inter-TSF trusted channel

Hierarchical to: No other components.

Dependencies: No dependencies.

FTP_ITC.1.1 The TSF shall provide a communication channel between itself and another trusted IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.

FTP_ITC.1.2 The TSF shall permit [*the TSF or another trusted IT product*] to initiate communication via the trusted channel.

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for [**master-slave connections and management network connections**].

Application note:

This SFR applies to the management network connections shown in Figure 2, excluding the License Server and NTP server connections. It therefore applies to the master-slave database connections (which although logically distinct also travel on the physical management network).

5.3 Security Assurance Requirements

The security assurance requirements are drawn from [CC3] and represent EAL2, with the addition of ALC_FLR.2 Flaw Reporting Procedures. The assurance components are identified in the table below.

Assurance Class	Assurance Components
Security Target (ASE)	ST introduction (ASE_INT.1)
	Conformance claims (ASE_CCL.1)
	Security problem definition (ASE_SPD.1)
	Security objectives (ASE_OBJ.2)
	Extended components definition (ASE_ECD.1)
	Derived security requirements (ASE_REQ.2)
	TOE summary specification (ASE_TSS.1)
Development (ADV)	Security architecture description (ADV_ARC.1)
	Security-enforcing functional specification (ADV_FSP.2)

Assurance Class	**Assurance Components**
	Basic design (ADV_TDS.1)
Guidance documents (AGD)	Operational user guidance (AGD_OPE.1)
	Preparative procedures (AGD_PRE.1)
Life cycle support (ALC)	Use of a CM System (ALC_CMC.2)
	Parts of the TOE CM coverage (ALC_CMS.2)
	Delivery procedures (ALC_DEL.1)
	Flaw reporting procedures (ALC_FLR.2)
Tests (ATE)	Evidence of coverage (ATE_COV.1)
	Functional testing (ATE_FUN.1)
	Independent testing – sample (ATE_IND.2)
Vulnerability assessment (AVA)	Vulnerability analysis (AVA_VAN.2)

Table 2: Security Assurance Requirements

The selection of EAL2 is consistent with the assurance levels commonly used for commercial products of this sort, and the augmentation with ALC_FLR.2 provides additional confidence for users that there is a process for reporting and addressing any vulnerabilities that might be subsequently discovered in the product, and hence that its security will be maintained over time.

5.4 Security Requirements Rationale

5.4.1 Mapping between SFRs and Security Objectives

The mapping between security objectives for the TOE and the SFRs that implement them is summarised in Table 3.

SFRs Security Objectives	FIA_UID.2	FIA_UAU.2	FDP_IFC.1/VMData	FDP_IFF.1/VMData	FDP_IFC.1/VDisk	FDP_IFF.1/VDisk	FDP_RIP.1	FTP_ITC.1
O.VM_Access			X	X	X	X	X	
O.Admin_Access	X	X						
O.Secure_Traffic								X

Table 3: Objectives implemented by SFRs

O.VM_Access is addressed by the information flow policies in FDP_IFC.1/VMData and FDP_IFF.1/VMData for data in memory, FDP_IFC.1/VDisk and FDP_IFF.1/VDisk for data on disk, and FDP_RIP.1 for protection of deallocated memory in a virtual machine.

O.Admin_Access is addressed by the requirements for identification and authentication of XenServer administrators in FIA_UID.2 and FIA_UAU.2.

O.Secure_Traffic is addressed by the provision of a secure channel in FTP_ITC.1 to protect the relevant traffic.

5.4.2 SFR Dependencies Analysis

The dependencies between SFRs implemented by the TOE are addressed as follows.

SFR	Dependencies	Rationale Statement
FIA_UID.2	None	
FIA_UAU.2	FIA_UID.1	Met by FIA_UID.2
FDP_IFC.1/VMData	FDP_IFF.1	Met by FDP_IFF.1/VMData
FDP_IFF.1/VMData	FDP_IFC.1	Met by FDP_IFC.1/VMData
	FMT_MSA.3	FMT_MSA.3 defines controls on initialisation of the attributes that are

SFR	Dependencies	Rationale Statement
		used to enforce the policy in FDP_IFF.1. However, for XenServer the attribute is simply the ownership of the data by a particular VM: this arises from the creation and operation of the VM and is not subject to separate management. An FMT_MSA.3 SFR is therefore not required in this case.
FDP_IFC.1/VDisk	FDP_IFF.1	Met by FDP_IFF.1/VDisk
FDP_IFF.1/VDisk	FDP_IFC.1	Met by FDP_IFC.1/VDisk
	FMT_MSA.3	FMT_MSA.3 defines controls on initialisation of the attributes that are used to enforce the policy in FDP_IFF.1. However, for XenServer the attribute is simply the ownership of the virtual disk by a particular VM: this arises from the creation and operation of the VM and is not subject to separate management. An FMT_MSA.3 SFR is therefore not required in this case.
FDP_RIP.1	None	
FTP_ITC.1	None	

Table 4: Analysis of SFR dependencies

6. TOE Summary Specification

The XenServer Security Functions correspond closely to the SFRs that they implement, as described below.

6.1 VM Data Separation

When a VM attempts to access its virtualised resources, the mapping between logical and physical resources is carried out by a driver that has a 'front-end' in the relevant domU guest and a back-end in dom0 (as illustrated in Figure 4). Dom0 makes the actual accesses to physical devices using its normal device drivers. The separation of VM data in primary memory (i.e. virtualised RAM) is implemented by mapping tables maintained by dom0 and the Hypervisor, which ensures that no VM can access pages of physical memory which have been mapped to a different VM. Only dom0 can make the privileged hypervisor calls necessary to set up the mapping of physical memory (the hypervisor checks the domain ID of its caller to determine whether a hypercall should be permitted) – this prevents any domU guest from accessing memory-mapped I/O resources directly. The same approach applies to the mapping of virtual disks to real storage devices. When memory is reused then the Hypervisor is also responsible for ensuring that no previous content is available to the new owner of the memory.

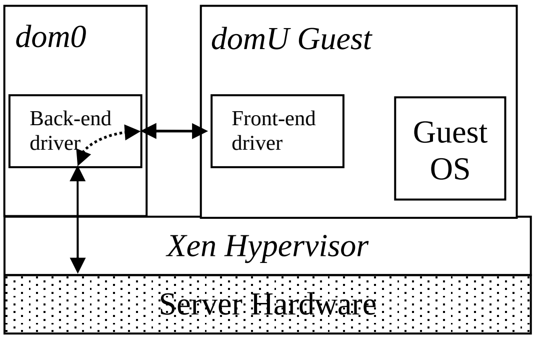

Figure 4: Overview of communication path between domU guest and hardware

Enforcement of protection implemented by the relevant Guest OS and/or application software (e.g. Windows file permissions) is outside the scope of the TOE.

This aspect of XenServer therefore implements FDP_IFC.1/VMData, FDP_IFF.1/VMData, FDP_IFC.1/VDisk, FDP_IFF.1/VDisk, and FDP_RIP.1.

6.2 Administrator Authentication

XenServer administrators gain access to XenServer using bulk data transfer/interactive services over the management network connection. The XenServer administrator is required to authenticate by submitting username and password credentials to dom0, which uses an implementation of PAM to check the credentials supplied.

This aspect of XenServer therefore implements FIA_UID.2 and FIA_UAU.2.

6.3 Channel Protection

XenServer protects the management network connection in two ways:

- The confidentiality and integrity of the master-slave connection is protected by the use of TLS[21]. The slave authenticates the master by checking its SSL certificate, while the master authenticates the slave by checking a shared secret supplied by the slave

- The confidentiality and integrity of all other management network traffic (except for the License Server and NTP server connections) is similarly protected by the use of TLS. Authentication in these cases is provided by submitting session credentials as in Administrator Authentication (section 6.2).

This aspect of XenServer therefore implements FIA_UID.2, FIA_UAU.2 and FTP_ITC.1.

End of Document

[21] Protection relies on correct configuration of the TOE according to its guidance documentation (see [CCECG]).