Microsoft Windows

Common Criteria Evaluation

Microsoft Windows Server 2008 Hyper-V

Security Target

Document Information	
Version Number	1.4
Updated On	Thursday, July 23, 2009

This is a preliminary document and may be changed substantially prior to final commercial release of the software described herein.

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, AS TO THE INFORMATION IN THIS DOCUMENT.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

The example companies, organizations, products, people and events depicted herein are fictitious. No association with any real company, organization, product, person or event is intended or should be inferred.

TABLE OF CONTENTS

1 ST Introduction

This is version 1.4 of the Security Target for Microsoft Windows Server 2008 Hyper-V RTM (called "Hyper-V" in this document).

1.1 Security Target (ST) reference

Title:	Microsoft Windows Server Core 2008: Hyper-V Server Role, Hyper-V Security Target
Version:	1.4
Keywords:	Virtualization, hypervisor

This document is the Security Target for the Common Criteria evaluation of the Microsoft Server 2008 Hyper-V RTM virtualization product. It is conformant to the Common Criteria for Information Technology Security Evaluation Version 3.1 [CC].

1.2 TOE Reference

The TOE is the Microsoft Hyper-V Server 2008 6.0.6001 Service Pack 1 with Hotfix KB950050. This product is a hypervisor or virtual machine monitor for Intel processors with Intel's VT-x support and for AMD processors with AMD VT support.

1.3 TOE Overview

The Target of Evaluation (TOE) is the Microsoft Hyper-V virtualization part of the Server 2008 product. Hyper-V allows the definition of partitions that have separate address spaces where they can load an operating system and applications operating on top of this operating system. The TOE consists of the Microsoft Server 2008 Server Core with the Hyper-V hypervisor running in a hypervisor configuration.

An operating system within such a partition has access to virtualized peripheral devices where access to those devices is controlled by Hyper-V. An operating system may either access devices using the same I/O related instructions as on a real system or it may use a specific interface offered by Hyper-V (called the VMBus) to communicate with Hyper-V for access to peripheral devices. In the first case the operating system can only access the devices virtualized by Hyper-V. When using the VMBus defined interface, an operating system in a guest partition needs to install "enlightenments" that set up the VMBus communication and use the "synthetic" devices accessible via VMBus. Note that the "enlightenments" within a guest operating system is part of the TOE, but not part of the TSF.

1.3.1 Summary of the TOE security functions

The TOE offers separation of partitions, controls access of partitions to resources like virtual hard disks or virtual network adapter, allows the definition of roles for the management of the TOE and enforces a role-based management policy, allows auditing of security critical events, authenticates administrative users in the root partition, and enforces quota for CPU time for partitions.

1.3.2 TOE usage

The TOE can be used to consolidate several physical servers based on Intel's x86 architecture onto one machine. The TOE allows the definition of so called partitions. Each instantiation of the TOE has one dedicated partition, called the root partition, and a variable number of so called "guest partitions". Resources accesses by guest partitions are virtualized by the TOE, i. e. the TOE performs a "translation" of the virtual resource accesses by a guest partition onto the real resources available to the TOE. Such resources include virtual CPUs, main memory, virtual hard disks, virtual network adapters, virtual CD/DVD drives or floppy disk drives as well as virtual video adapter and virtual mouse and keyboard. The root partition is used for support of the resource virtualization and for TOE management activities.

Each guest partition can take over the tasks of one physical server. Each guest will have its own operating system installed and is restricted by the TOE to the use of the resources that are assigned to the partition. The assignment of resources to partitions is performed by administrative roles defined in the root partition. The TOE allows separating each guest partition from others with a comparable degree as if they were executing on separate physical servers.

See the TOE description section in this chapter for more detailed information about the TOE structure and the security functionality implemented by the TOE.

1.3.3 Definitions

This section defines some of the TOE specific terms used throughout this document.

Hypervisor and Partitions

A hypervisor is a layer of software that sits just above the hardware and beneath one or more operating systems. Its primary job is to provide isolated execution environments called partitions. Each partition is provided with its own set of (physical or virtual) hardware resources (memory, devices, CPU cycles). The hypervisor is responsible for controlling and arbitrating access to the underlying hardware where necessary.

Guests

Software running within a partition is referred to as a guest. A guest might consist of a full-featured operating system like Windows Vista or a small, special-purpose kernel. The hypervisor is "guest-agnostic".

Specialized Partitions

In most respects, all partitions are equal. However, some partitions may be granted special privileges or assigned specialized functions.

In the evaluated version of Hyper-V there is only partition that has special privileges, which is called the root partition. The root partition acts as the default owner of all hardware resources. It is also typically in charge of power management, plug and play, and hardware failure events. The root partition is also responsible for creating and managing other partitions and assigning hardware resources. In some respects, it acts like the service processor on a machine with hardware partitioning facilities. To start Hyper-V, first a Server 2008 instance is started that is supposed to run in the root partition. In the evaluated configuration, this is the Server 2008 Server Core system. This then starts the Hypervisor, which creates the root partition and “moves” the running instance of Server 2008 into the root partition and assigns all devices to this partition.

Virtualization and Emulation

The hypervisor provides support for hardware virtualization. Virtualization provides multiple logical instances of CPUs and other hardware resources. These logical instances are mapped onto physical hardware resources using a variety of techniques. One such technique is emulation, the simulation of a processor or device using software. While the hypervisor facilitates processor and device emulation, the architecture attempts to provide or facilitate alternatives to emulation for performance reasons.

Legacy Guests, Worker Processes and Enlightenment

The hypervisor provides support for legacy guests. A legacy guest is an operating system that has no knowledge of the fact that it is running within a virtualized environment. Legacy guests require substantial infrastructure including a system BIOS and a wide variety of emulated devices. This infrastructure is not provided directly by the hypervisor. A separate piece of code called a worker process provides this infrastructure to legacy guests. Worker processes are part of the root partition and receive specific intercepts (i.e. notifications that specific events have occurred within a guest). This support for guest-mode intercept handlers provides added flexibility and reduced complexity within the hypervisor. Note that a worker process exists for each guest partition.

Device emulation provides broad compatibility, but it results in poor performance. Other operational assumptions made by legacy guests also add to virtualization overhead. This virtualization performance overhead can be mitigated by enlightening a legacy operating system. An enlightened guest has knowledge of the fact that it is running within a virtualized environment and changes its behavior accordingly. Various degrees of enlightenment are possible. For example, a guest might use a specialized block device driver to talk to an idealized virtual disk using a fast communication channel between itself and the root partition. More extensive enlightenment involves modifying or extending a guest hardware abstraction layer (HAL) so it talks to an idealized “synthetic” interrupt controller or access model specific register (MSR) that do not exist in real hardware.

Snapshots

A snapshot is a collection of data about a partition and its current state that allows restarting the partition in this state. A Hyper-V snapshot therefore includes all of the information and data that is required to roll back the status of a partition to the state when the snapshot was taken. Information that is collected when taken a snapshot include:

- Partition configuration settings (the contents of the .vmc file)
- Virtual network settings

- The current state of all virtual hard disks (VHDs) that are attached to the partition
- State information for the partition

1.4 TOE description

Hyper-V consists of a small hypervisor layer that uses the virtualization support functions of the Intel (Intel-VT) and AMD (AMD-V) to define and separate guest partitions, the Server 2008 Server Core installation in the root partition and the “enlightenments” that can be used in guest partitions to use the virtualization functions more efficiently. For more information about the hardware support for virtualization provided by Intel and AMD, please read [INTEL-VT] and [AMD-V]. Note that the “enlightenments” are not part of the TSF portion of the TOE. The TOE also includes the guidance documentation required to securely install, configure and operate Hyper-V with the Server 2008 core in the root partition.

A partition is defined by assigning a memory region (called “guest physical memory” GPA), a number of virtual processors and a set of virtualized devices and resources. Hyper-V differentiates between the single “root” partition and multiple “guest” partitions. The root partition is part of the trusted components of the TOE since it is responsible for the management of the partitions as well as the virtualization of devices and resources other than physical memory pages and virtual processors. All real devices except memory pages and processors are allocated to the root partition and all the mapping between the virtualized devices accessible by a guest partition and the real device that backs the virtual device is done by the root partition. The root partition runs the Server Core configuration of Server 2008 with the configuration defined in [Hyper-V_ECG].

When the administrator starts a partition, first the worker process for that partition is started. This worker process reads the configuration data of the partition and based on this data it defines the events within the partition that it wants to intercept. Additional events to be intercepted are defined by the root partition when the hypervisor layer is instructed by the root partition to start the new partition. The hypervisor itself defines additional events it wants to intercept and handle. Whenever such an event happens, a trap into the hypervisor is generated by the processor hardware and the hypervisor either handles the event itself or passes it on to the root partition to be handled there. For events where the hypervisor needs the support of the root partition, the hypervisor generates a message to the root partition, which then performs the required actions like mapping the access to a virtualized device to the real device that backs the virtual device. When this has been done, the root partition sends a message back to the hypervisor defining how to respond to the partition. The hypervisor then reflects this response back to the partition.

The hypervisor also provides a set of functions to the partitions they can use. One of those functions can be used by a partition to determine it is running on Hyper-V and to determine the version of Hyper-V that is installed. The software within the partition (usually a server operating system) can then install software that makes use of specific hypervisor functions that are not available if the operating system would execute directly on the system’s hardware. This software, called “enlightenments” can then optimize the performance of the operating system and use the flexibility provided by the hypervisor.

Beside the hypervisor calls that allow the use of specific functions of the hypervisor layer, this also includes a direct and fast communication mechanism between individual guest partitions and the root partition. This communication mechanism, called VMBus, allows a guest partition to “shortcut” requests for access to a virtualized device allocated to the partition and sends this request directly to the root partition without involving the hypervsisor layer and without undergoing the burden of virtualizing every hardware interface of this device. Using VMBus the root partition can also return any response to the request directly to the partition. The software within a partition that implements the “shortcuts” for accessing virtualized devices within a guest partition is called “Virtualization Service Client” or VSC. The corresponding software component within the root partition that handles the requests transferred via VMBus is called a “Virtualization Service Provider” or VSP.

Hyper-V ensures that the physical memory allocated to guest partitions do not overlap with memory allocated to another guest partition, providing each guest partition with its own, isolated guest physical address space. Real processors within a system are assigned by the hypervisor to guest partitions on a time-slice basis. There is no guarantee that the same virtual processor is always backed by the same real processor and real processors are not bound to a specific guest partition.

Hyper-V just ensures that when a real processor is assigned to a virtual processor, which then is executing on behalf of code within a guest partition, no information is passed via the processor’s state or register from its previous assignment to a virtual processor of another guest partition.

Hyper-V therefore enforces the separation of guest partitions as well as an access control policy between partition and virtualized devices. In addition Hyper-V allows assigning maximum quota for CPU time and memory for guest partitions to avoid that a single partition is able to use all of those resources.

The management of Hyper-V and its configuration is performed within the root partition. The Server 2008 components within the root partition provide the generic Server 2008 functionality for user authentication, auditing and audit management, access control and process isolation and management. In the evaluated configuration the root partition is solely used for the management of Hyper-V ,the virtualization of devices and passing remote users through to guest partitions using the RDP protocol. No untrusted program is allowed to be installed on the Server 2008 instance within the root partition. An organization that wants to have generic applications operating on Server 2008 needs to install Server 2008 in one of the guest partitions and install those programs on this instantiation of Server 2008.

Hyper-V is distributed as part of the Windows Server software editions (Windows Server 2008 Datacenter, Windows Server 2008 Enterprise, and Windows Server 2008 Standard). Those include the Hyper-V role, with the x64 version of the remote management tools, and integration services for the supported versions of the Windows operating system. Integration services for the supported versions of Linux distributions are distributed through the Microsoft Connect Web site and are identified as Linux Integration Components for Microsoft Hyper-V.

The Hyper-V management tools are available separately to allow remote management of a server running Hyper-V. Packages are available to install the tools on Windows Vista with Service Pack 1 (SP1) and on 32-bit editions of Windows Server 2008.

The guidance documentation provided for Hyper-V consists of the interface descriptions for the Hypervisor Calls, the description of the management interfaces (WMI interfaces for virtualization) and the description of the AzMan interfaces. In addition there is guidance for the secure installation, configuration and start-up of the TOE. Specific guidance for non-administrative users is not required, since the only non-administrative 'users' are the child partitions and remote users that are passed through to a guest partition they are allowed to access. Neither of those does require specific guidance other than the description of the programming interfaces that a partition can use resp. the list of VMs they may attempt to connect to.

Administrative users or users that are allowed to connect to a partition are identified and authenticated by the TOE using either locally defined and managed credentials or through the use of an Active Directory service. The protection and correct administration of the server operating the Active Directory service as well as the protection of the communication link to this server has to be ensured by the TOE environment.

1.4.1 Intended method of use

Hyper-V is used in conjunction with Windows Server 2008 to provide a computing environment that allows creating partitions within a single computer system where each partition can load and operate an operating system and its applications very much like the operating system and its application would execute directly on real hardware. Each partition gets virtualized hardware resources (memory, processors, storage, network devices, interrupt controller etc.) assigned where the virtualization of those resources is performed by the TOE. This allows the operator of a real server system to have multiple virtual servers installed on one server hardware system and assign each virtual server the resources it needs. The management functions of Hyper-V allow an authorized administrator to modify this allocation of resources to virtualized servers in order to optimize the use of resources and react to the different needs for resources by the virtualized servers.

Virtualized servers in the partitions are separated from each other such that they can only communicate using virtualized network functions or shared storage devices. This is the same way they would also communicate with each other if they were installed on separate hardware systems and the intention of the virtualization provided by Hyper-V is to provide a comparable degree of separation between virtual servers as there is between real servers on different hardware while being able to optimize the use of resources and ease the management of multiple servers.

For security reasons the evaluated configuration does not allow the installation of untrusted software in the root partition.

1.4.2 Summary of security functionality

Hyper-V provides the following primary security functionality:

- Access control between partitions and virtualized resources
- Auditing of security critical events detected by Hyper-V

- Object reuse for all resources managed by Hyper-V
- Management of the Hyper-V configuration including the configuration of the partitions
- Maximum quota for defined resources assigned to partitions (CPU time, memory, disk storage)

In addition the root partition provides the following security functionality within the Server 2008 parts:

- Identification and authentication of administrative users and users that request to be passed through to a guest partition
- Management and protection of the audit trail
- Access control of administrative users to management objects
- Access control to files and devices used
- Management of users and access control

In addition Hyper-V provides the following architectural properties:

- TSF protection against tampering from guest partitions and network devices
- Separation between the guest partitions
- Reference mediation for access of guest partitions to protected resources (including virtualized devices)
- Non-bypassability of the reference mediation
- Maintaining the separation mechanism provided by the underlying hardware when virtualizing resources and devices or responding to hypervisor calls for a guest partition.

1.5 Hardware Requirements and supported guest operating systems

Note: the requirements defined in this section apply for the version of Hyper-V subject to this evaluation. Future versions of Hyper-V may have different hardware requirements and may support other guest operating systems.

To install and use the Hyper-V role, the following is required:

- An x64-based processor. Hyper-V is available in 64-bit editions of Windows Server 2008—specifically, the 64-bit editions of Windows Server 2008 Standard, Windows Server 2008 Enterprise, and Windows Server 2008 Datacenter.

  Hyper-V is not available for 32-bit (x86) editions or Windows Server 2008 for Itanium-Based Systems. However, the Hyper-V management tools are available for 32-bit editions.

- Hardware-assisted virtualization. This is available in processors that include a virtualization option—specifically processors with Intel Virtualization Technology (Intel VT) or AMD Virtualization (AMD-V) technology.
- Hardware-enforced Data Execution Prevention (DEP) must be available and enabled. Specifically, you must enable Intel XD bit (execute disable bit) or AMD NX bit (no execute bit).

1.5.1 Memory

The maximum amount of memory that can be used is determined by the operating system, as follows:

- For Windows Server 2008 Enterprise and Windows Server 2008 Datacenter, the physical computer can be configured with up to 1 TB of physical memory, and partitions that run either of those editions can be configured with up to 64 GB of memory per partition.
- For Windows Server 2008 Standard, the physical computer can be configured with up to 32 GB of physical memory, and partitions that run either of those editions can be configured with up to 31 GB of memory per partition.

1.5.2 Processors

Hyper-V is supported on physical computers with up to 16 logical processors. A logical processor can be a core processor or a processor using hyper-threading technology. One can configure up to 4 virtual processors on a partition. However, the number of processors supported by a guest operating system might be lower.

The following are some examples of supported systems and the number of logical processors they provide:

- A single-processor/dual-core system provides 2 logical processors.
- A single-processor/quad-core system provides 4 logical processors.
- A dual-processor/dual-core system provides 4 logical processors.
- A dual-processor/quad-core system provides 8 logical processors.
- A quad-processor/dual-core system provides 8 logical processors.
- A quad-processor/dual-core, hyper-threaded system provides 16 logical processors.
- A quad-processor/quad-core system provides 16 logical processors.

1.5.3 Networking

Hyper-V provides the following networking support:

- Each partition can be configured with up to 12 virtual network adapters — 8 can be the “network adapter” type and 4 can be the “legacy network adapter” type. The network adapter type provides better performance and requires a dedicated driver that is included in the integration services packages.

- Each virtual network adapter can be configured with either a static or dynamic MAC address.
- Each virtual network adapter offers integrated virtual local area network (VLAN) support and can be assigned a unique VLAN channel.
- Hyper-V supports an unlimited number of virtual networks with an unlimited number of partitions per virtual network.

Note: Hyper-V cannot connect a virtual network to a wireless network adapter. As a result, wireless networking capabilities can not be provided to partitions.

1.5.4 Storage

Hyper-V supports a variety of physical storage options:

- Direct-attached storage: Serial Advanced Technology Attachment (SATA), external Serial Advanced Technology Attachment (eSATA), Parallel Advanced Technology Attachment (PATA), Serial Attached SCSI (SAS), SCSI, USB, and Firewire.
- Storage area networks (SANs): Internet SCSI (iSCSI), Fibre Channel, and SAS technologies cam be used.
- Network-attached storage

Hyper-V supports configuring a partition to use the following types of virtual storage.

- Virtual hard disks of up to 2040 GB. Hyper-V can use fixed virtual hard disks, dynamically expanding virtual hard disks, and differencing disks.
- Virtual IDE devices. Each partition supports up to 4 IDE devices. The startup disk (sometimes referred to as the boot disk) must be attached to one of the IDE devices. The startup disk can be either a virtual hard disk or a physical disk.
- Virtual SCSI devices. Each partition supports up to 4 virtual SCSI controllers, and each controller supports up to 64 disks. This means that each partition can be configured with as many as 256 virtual SCSI disks.
- Physical disks. Physical disks attached directly to a partition (sometimes referred to as pass-through disks) have no size limitation other than what is supported by the guest operating system.
- Partition storage capacity. Using virtual hard disks, each partition supports up to 512 TB of storage. Using physical disks, this number is even greater depending on what is supported by the guest operating system.

- Partition snapshots. Hyper-V supports up to 50 snapshots per partition.

1.5.5 Other hardware components

The following is information about the other types of physical and virtual hardware components that can be used with Hyper-V.

DVD drive	A partition has 1 virtual DVD drive by default when you create the partition. Partitions can be configured with up to 3 DVD drives, connected to an IDE controller. (Partitions support up to 4 IDE devices, but one device must be the startup disk.) A virtual DVD drive can access CDs and DVDs, either .iso files or physical media. However, only one partition can be configured to access a physical CD/DVD drive at a time.
Virtual COM port	Each partition is configured with 2 virtual serial (COM) ports that can be attached to a named pipe to communicate with a local or remote physical computer. **Note:** No access to a physical COM port is available from a partition.
Virtual floppy drive	Each partition is configured with 1 virtual floppy drive, which can access virtual floppy disk (.vfd) files. **Note:** No access to a physical floppy drive is available from a partition.

1.5.6 Supported guest operating systems

The following operating systems are supported for use in a partition as a guest operating system. 32-bit and 64-bit guest operating systems can be executed at the same time on one server running Hyper-V. Note that the TSF does not make any security-related assumptions based on the type of guest operating system used – see Section 1.3.1. Therefore, other operating systems not explicitly listed bellow, such as future versions of the listed ones, may also be used as guests.

- The following 32-bit and 64-bit editions of Windows Server 2008 can be used as a supported guest operating system in a partition configured with 1, 2, or 4 virtual processors:
  - Windows Server 2008 Standard and Windows Server 2008 Standard without Hyper-V
  - Windows Server 2008 Enterprise and Windows Server 2008 Enterprise without Hyper-V
  - Windows Server 2008 Datacenter and Windows Server 2008 Datacenter without Hyper-V
  - Windows Web Server 2008

  - Windows Server 2008 HPC Edition
- The following editions of Windows Server 2003 can be used as a supported guest operating system in a partition configured with 1 or 2 virtual processors:
  - Windows Server 2003 R2 Standard Edition with Service Pack 2
  - Windows Server 2003 R2 Enterprise Edition with Service Pack 2
  - Windows Server 2003 R2 Datacenter Edition with Service Pack 2
  - Windows Server 2003 Standard Edition with Service Pack 2
  - Windows Server 2003 Enterprise Edition with Service Pack 2
  - Windows Server 2003 Datacenter Edition with Service Pack 2
  - Windows Server 2003 Web Edition with Service Pack 2
  - Windows Server 2003 R2 Standard x64 Edition with Service Pack 2
  - Windows Server 2003 R2 Enterprise x64 Edition with Service Pack 2
  - Windows Server 2003 R2 Datacenter x64 Edition with Service Pack 2
  - Windows Server 2003 Standard x64 Edition with Service Pack 2
  - Windows Server 2003 Enterprise x64 Edition with Service Pack 2
  - Windows Server 2003 Datacenter x64 Edition with Service Pack 2
- The following versions of Windows 2000 can be executed in a partition configured with 1 virtual processor:
  - Windows 2000 Server with Service Pack 4
  - Windows 2000 Advanced Server with Service Pack 4
- The following Linux distributions can be executed in a partition configured with 1 virtual processor:
  - Suse Linux Enterprise Server 10 with Service Pack 2 (x86 edition)
  - Suse Linux Enterprise Server 10 with Service Pack 2 (x64 edition)
  - Suse Linux Enterprise Server 10 with Service Pack 1 (x86 edition)
  - Suse Linux Enterprise Server 10 with Service Pack 1 (x64 edition)

- The following 32-bit and 64-bit versions of Windows Vista can be executed in a partition configured with 1 or 2 virtual processors:
    - Windows Vista Business with Service Pack 1
    - Windows Vista Enterprise with Service Pack 1
    - Windows Vista Ultimate with Service Pack 1
- The following versions of Windows XP can be executed in a partition:
    - Windows XP Professional with Service Pack 3 (configured with 1 or 2 virtual processors)
    - Windows XP Professional with Service Pack 2 (configured with 1 virtual processor)
    - Windows XP Professional x64 Edition with Service Pack 2 (configured with 1 or 2 virtual processors)

1.5.7 Integration services (Enlightenments)

Integration services are available for supported guest operating systems as described in the following table.

Guest operating system	Device and service support
Windows Server 2008 (64-bit editions)	Drivers: IDE, SCSI, networking, video, and mouse Services: operating system shutdown, time synchronization, data exchange, heartbeat, and online backup
Windows Server 2008 (x86 editions)	Drivers: IDE, SCSI, networking, video, and mouse Services: operating system shutdown, time synchronization, data exchange, heartbeat, and online backup
Windows Server 2003 (x64 editions) with Service Pack 2	Drivers: IDE, SCSI, networking, video, and mouse Services: operating system shutdown, time synchronization, data exchange, heartbeat, and online backup
Windows Server 2003 (x86 editions) with Service Pack 2	Drivers: IDE, SCSI, networking, video, and mouse Services: operating system shutdown, time synchronization, data exchange, heartbeat, and online backup
Windows 2000 Server with Service Pack 4	Drivers: IDE, networking, video, and mouse Services: operating system shutdown, time synchronization,

Guest operating system	Device and service support
	data exchange, and heartbeat
Windows 2000 Advanced Server with Service Pack 4	Drivers: IDE, networking, video, and mouse Services: operating system shutdown, time synchronization, data exchange, and heartbeat
Suse Linux Enterprise Server 10 (x64 edition) with Service Pack 1 or 2	Drivers only: IDE, SCSI, networking, and mouse
Suse Linux Enterprise Server 10 (x86 edition) with Service Pack 1 or 2	Drivers only: IDE, SCSI, networking, and mouse
Windows Vista (64-bit editions) with Service Pack 1	Drivers: IDE, SCSI, networking, video, and mouse Services: operating system shutdown, time synchronization, data exchange, heartbeat, and online backup
Windows Vista (x86 editions) with Service Pack 1	Drivers: IDE, networking, video, and mouse Services: operating system shutdown, time synchronization, data exchange, heartbeat, and online backup
Windows XP Professional (x86 editions) with Service Pack 2 or 3	Drivers: IDE, SCSI, networking, video, and mouse Services: operating system shutdown, time synchronization, data exchange, and heartbeat
Windows XP Professional x64 Edition with Service Pack 2	Drivers: IDE, SCSI, networking, video, and mouse Services: operating system shutdown, time synchronization, data exchange, and heartbeat

1.6 TOE Guidance

The following guidance documentation exists for the startup and configuration as well as the operation of the TOE:

- Windows Server Core 2008: Hyper-V Server Role, Hyper-V Evaluated Configuration Guide [Hyper-V_ECG]

This document contains the entry point for installing, configuring and operating the TOE in its evaluated configuration. This document contains pointers to other documents that are also considered guidance, where [Hyper-V_ECG] has precedence whenever those documents contradict [Hyper-V_ECG].

1.7 References

[AMD-V] AMD64 Technology AMD64 Architecture Programmer’s Manual

Volume 2: System Programming, Revision 3.14, September 2007

[CC] Common Criteria for Information Technology Security Evaluation, Version 3.1 Revision 2, September 2007, Part 1 to 3, CCMB-2007-09-001 to CCMB-2007-09-003

[CEM] Common Methodology for Information Technology Security Evaluation, Version 3.1 Revision 2, September 2007, CCMB-2007-09-004

[INTEL-VT] Intel® 64 and IA-32 Architectures Software Developer’s Manual, Volume 3B: System Programming Guide, Part 2, July 2008

Hyper-V_ECG Microsoft Server Core 2008: Hyper-V Server Role, Hyper-V Evaluated Configuration Guide, Version 1.0, May 2009

2 Conformance Claim

This Security Target is conformant to the Common Criteria for Information Technology Security Evaluation Version 3.1 [CC]. It is CC Part 2 conformant and Part 3 conformant, with a claimed Evaluation Assurance Level of EAL4, augmented by ALC_FLR.3.

This Security Target does not claim conformance with any Protection Profile.

3 Security Problem Definition

3.1 Introduction

The intended purpose of the TOE is to allow the operation of different guest partitions on a single hardware server where each guest partition will operate like its own (virtualized) server system. Like in a data center that hosts a number of server systems used for different purposes and operate potentially even for different organizations, the separation of the guest partitions within the TOE is the main task to be achieved by the TSF. In addition, a guest partition should not be able to access TSF controlled resources other than those the TSF has assigned to the guest partition.

The configuration of the partitions is assumed to be performed by trusted administrator that are well trained and perform their duties in accordance with the policies defined by the organization that operates the TOE.

3.2 Threats

The **threat agents** in such a system are therefore the following:

- Subjects within a guest partition that attempt to interfere with the configuration or operation of other guest partitions, that attempt to escalate their privileges within the TOE, attempt to access or use resources not assigned to the partition, or attempt to escalate their privileges within a partition using a way that they would not have when the software in the partition would execute on a real server system.
- External entities that attempt to attack the TOE via an external interface in order to tamper with the TSF or the TSF data.
- Administrators that may misconfigure the TOE such that it does not protect assets in accordance with an organizations policy.

The first two threat agents are assumed to have a basic attack potential (compatible with the target assurance level). Administrators are assumed to be trustworthy and not assumed to actively attack the security functionality of the TOE. Especially they are not assumed to deliberately attempt to extend their privileges. (Note that this restriction may be lifted once the access control functions of Server 2008 have been evaluated independently in a separate evaluation). As a result the administrator is only seen as a threat agent to the respect that the guidance documentation may be misleading in a way that the administrator configures the TOE to an insecure state where the guidance does not provide sufficient information that this state is insecure.

The **assets** that need to be protected are the resources assigned to a partition, the virtualized and synthetic devices managed by the root partition and the TSF data, which includes the configuration data of the partitions and the audit data generated by the TOE.

The TOE is designed to counter the following threats:

T.CONFIGURATION_CHANGE	The lack of TSF-enforced constraints on the ability of an authorized subject to invoke or dictate how the TOE is reconfigured may result in the TOE transitioning to an insecure (unknown, inconsistent, etc) state.
T.DENIAL_OF_SERVICE	A malicious subject may block others from specific system resources (system memory, persistent storage, and processing time) via a resource exhaustion attack.
T.UNAUTHORIZED_ACCESS	A subject may gain access to resources or TOE security management functions for which it is not authorized according to the TOE security policy.
T.PARTITION_COMPROMISE	The TSF of a correctly configured TOE may allow software within a guest partition to escalate its processor or memory related privileges in a way that would not be possible when the software is executed on the real hardware. (Note: The software within a guest partition is not subject to this evaluation and therefore any flaw within a guest partition's software that allows such a privilege escalation is beyond the scope of this evaluation. Also the software a hypervisor-aware operating system installs when executing on the TOE is not part of this evaluation).

Table 2: List of Threats

3.3 Security Policies

An organizational security policy is a set of rules or procedures imposed by an organization upon its operations to protect its sensitive data. The following operational security policies are supported by the TOE:

P.ACCOUNTABILITY	The TOE shall provide the capability to make available information regarding the occurrence of security-relevant events.
P.CONFIGURATION	The TOE shall provide functions that allow authorized administrators to perform the setup and initialization of the TOE in accordance with the security policies for configuration and administration issued by the organization responsible for the operation of the TOE.

Table 3: List of Organisational Security Policies

3.4 Assumptions

This section describes the security aspects of the environment in which the TOE is intended to be used. It includes information about the physical, personnel, procedural, and connectivity aspects of the environment.

The TOE is assured to provide effective security measures in a cooperative non-hostile environment only if it is installed, managed, and used correctly. The operational environment must be managed in accordance with user/administrator guidance documentation. The following specific conditions are assumed to exist in an environment where the TOE is employed:

A.ADMIN	Authorized administrators of the TOE are assumed to be knowledgeable and trustworthy to follow the guidance and not misuse their privileges. This applies also to domain administrators that manage the AD-DS used by the TOE.
A.PHYSICAL	It is assumed that the non-IT environment provides the TOE with appropriate physical security commensurate with the value of the IT assets protected by the TOE.
A. SUBJECT_ALLOCATION	It is assumed that properly trained trusted administrators will create and manage the configuration data of partitions.
A.DEFINED_INSTALL	It is assumed that the administrator installs and configures the TOE in accordance with the guidance provided for the installation and configuration of the TOE.
A.REMOTE_ADMIN	It is assumed that remote administration is performed only using properly protected communication links.
A.REMOTE_IT_PRODUCTS	It is assumed that any other IT product that may be used to support the authentication of administrators, used to protect communication links, or used to assist administrators in their administrative tasks is trusted to perform its security related functions correctly and does not include side effects that may allow unauthorized persons to perform administrative functions on

	the TOE or perform administrative functions other than those explicitly initiated by the trusted administrator. Any communication to such a trusted IT product is assumed to be protected against unauthorized interception or modification of the network traffic. The applies especially to the communication with the Active Directory Directory Service (AD-DS) used by the TOE.
A.CLEAN_ROOT	It is assumed that no additional software than the one specified in the configuration guidance is installed in the root partition.
A.CORRECT_HARDWARE	It is assumed that the underlying hardware of the TOE operates correctly as described in the hardware manuals and does not expose undocumented critical side effects
A.PARTITION_CONNECT	It is assumed that users that are allowed to connect to a particular guest partition via the Remote Desktop Protocol (RDP) have all the same right to access information within this partition.
A.MEMORY_MANAGEMENT	The Windows Server 2008 instance that is running in the root partition provides memory management services to other components running in the server instance or the root partition with kernel-mode privileges. It exposes a dedicated and functionally-complete kernel memory management API to these components. In particular, the kernel memory manager ensures that any new request for allocating memory coming through the API is serviced by zeroizing the memory pages before making them available to the requestor.

Table 4: List of Assumptions

4 Security Objectives

This section defines the security objectives of the TSF and its supporting environment. Security objectives, categorized as either IT security objectives or non-IT security objectives, reflect the stated intent to counter identified threats, comply with any organizational security policies identified, or both. All of the identified threats and organizational policies are addressed under one of the following categories.

4.1 Security Objectives for the TOE

O.PARTITION_ACCESS	The TOE will ensure that subjects within a partition gain only authorized access to exported resources assigned to the partition.
O.AUDIT_GENERATION	The TOE will provide the capability to detect, generate audit records for security relevant auditable events.
O.AUTHORIZED_SUBJECT	The TOE will ensure that only authorized administrators are allowed to access security relevant TOE configuration data.
O.INIT_SECURE_STATE	The TOE will provide functions that allow administrators to setup and configure the TOE such that it is started in a secure state where all the other security objectives are enforced.
O.MANAGE	The TOE will provide all the functions necessary to support the administrative users and authorized subjects in their management of the TOE security functions and configuration data, and restrict these functions from use by unauthorized subjects.
O.RESIDUAL_INFORMATION	The TOE will ensure that any information contained in a protected resource is not released to subjects when the resource is reallocated. The TOE will utilize the facilities provided by A.MEMORY_MANAGEMENT to accomplish this.
O.RESOURCE_ALLOCATION	The TOE will provide mechanisms that enforce constraints on the allocation of TOE resources assigned to a partition.
O.PARTITION_ISOLATION	The TOE will provide mechanisms to protect each

	guest partition from unauthorized interference by other guest partitions.
O.PRESERVE_PART_PRIV	The TOE will preserve the hardware separation functions within a partition such that software within the partition is able to implement its own policy for separation in the same way as it would be when executing directly on the underlying hardware

Table 5: List of Security Objectives for the TOE

4.2 Security Objectives for the TOE environment

The TOE is assumed to be complete and self-contained and, as such, not dependent upon any other products to perform properly. However, certain objectives with respect to the general operating environment must be met. The following are the non-IT security objectives:

OE.PHYSICAL	Physical security will be provided for the TOE by the non-IT environment commensurate with the value of the IT assets protected by the TOE.
OE.HW_MECHANISMS	The underlying processor implements the privileges, ring separation, memory protection, and virtualization support as described in the hardware manuals. The underlying hardware implements interrupt handling, bus configuration, device controller configuration, timers and time, and other I/O related aspects as described in the related hardware manuals.
OE_HW_SIDE_EFFECTS	The underlying hardware has no undocumented side effects that may interfere with the security functions implemented by the TOE.
OE.SUBJECT_ALLOCATION	A properly trained trusted individual will create configuration vectors such that, for those partitions to which subjects are allocated, each partition is allocated one or more subjects (i.e., subjects with homogeneous access requirements, or subjects with heterogeneous access requirements) that are appropriate for the policy abstraction supported by the TOE.

OE.TRUSTED_INDIVIDUAL	Any individual allowed to perform procedures upon which the security of the TOE may depend must be trusted with assurance commensurate with the value of the IT assets. This applies also for individuals managing functions within the TOE environment that the TOE depends on.
OE.REMOTE_ADMIN	Any communication link used for remote administration or communication with another trusted IT product must be protected from unauthorized access or interference by unauthorized persons or systems.
OE.REMOTE_IT_PRODUCT	Any remote IT product used to assist in the authentication of administrative users, in the administration activities or in the storage of TSF data must protect all security related data against unauthorized access and must ensure that it performs the function the TOE expects from this product correctly and without any side effects that could undermine the security of the TOE.
OE.NETWORK	The physical networks the TOE is connected to, have controls that protect against attacks on the physical layer (e. g. high voltage) and against attacks on the data link layer (layer 2).
OE.TRUSTED_ROOT	The root partition is installed as defined for the evaluated configuration and has no software applications installed that are not required for the virtualization support or the management of Hyper-V. Especially there is no untrusted application installed in the root partition.
OE.PARTITION_USER	Procedures in the TOE environment exist to ensure that users that are allowed to connect to a specific partition have the right to access all information processed by this partition.
OE.MEMORY_MANAGEMENT	Software running in the root partition, namely Server Core 2008, ensures that memory allocated to guest partitions is being zeroized before the guest partition is able to access the memory.

Table 6: List of Security Objectives for the TOE environment

4.3 Security objectives rationale

This section provides a rationale for the existence of each threat, policy statement, security objective, and component that comprise the security target.

4.3.1 Complete coverage: threats and organizational security policies

This section provides evidence demonstrating coverage of the threats and Organizational Security Policies (OSPs) by both the IT and non-IT security objectives. The following table shows this objective to threat and policy mapping, and the table is followed by a discussion of the coverage for each threat and OSP.

T.CONFIGURATION_CHANGE	O.MANAGE O.AUTHORIZED_SUBJECT OE.TRUSTED_INDIVIDUAL OE.REMOTE_ADMIN OE.REMOTE_IT_PRODUCT
T.DENIAL_OF_SERVICE	O.RESOURCE_ALLOCATION
T.UNAUTHORIZED_ACCESS	O.PARTITION_ACCESS O.AUTHORIZED_SUBJECT O.RESIDUAL_INFORMATION O.PARTITION_ISOLATION OE.PHYSICAL OE.NETWORK OE.REMOTE_IT_PRODUCT OE.TRUSTED_ROOT
T.PARTITION_COMPROMISE	O.PRESERVE_PART_PRIV OE.HW_MECHANISMS OE.HW_SIDE_EFFECTS

Table 7: Mapping Threats to Security Objectives

P.ACCOUNTABILITY	O.AUDIT_GENERATION
P.CONFIGURATION	O.INIT_SECURE_STATE OE.SUBJECT_ALLOCATION OE.TRUSTED_INDIVIDUAL OE.REMOTE_IT_PRODUCT OE.TRUSTED_ROOT

Table 8: Mapping Organisational Security Policies to Security Objectives

Threats

T.CONFIGURATION_CHANGE

> *The lack of TSF-enforced constraints on the ability of an authorized subject to invoke or dictate how the TOE is reconfigured may result in the TOE transitioning to an insecure (unknown, inconsistent, etc) state.*

This threat is addressed by the security objective O.MANAGE that requires the existence of appropriate management functions together with the security objective O.AUTHORIZED_SUBJECT, which requires that only authorized administrators can perform those management functions. OE.TRUSTED_INDIVIDUALS supports countering this threat by requiring that those administrators are trusted to perform their job correctly and not misuse their privileges. OE.REMOTE_ADMIN also supports countering this threat by requiring that remote administration facilities are protected against unauthorized subjects attempting to access the communication link. OE.REMOTE_IT_PRODUCT addresses the issue that a remote product used for the management of the TOE configuration may be used by an attacker or an unauthorized person to impersonate as an authorized administrator and modify the configuration of the TOE.

T.DENIAL_OF_SERVICE

> *A malicious subject may block others from system resources (e.g., system memory, persistent storage, and processing time) via a resource exhaustion attack.*

This threat is countered by security objective O.RESOURCE_ALLOCATION requesting that constraints exist on the allocation of resources to partitions, which in turn prohibits a denial of service attack caused by resource exhaustion.

T.UNAUTHORIZED_ACCESS

A subject may gain access to resources or TOE security management functions for which it is not authorized according to the TOE security policy.

This threat is countered by the security objective O.PARTITION_ACCESS, which requests an access control mechanism for resources exported to partitions. Concerning access to partition configuration data and other TSF data, this threat is addressed by the security objective O.AUTHORIZED_SUBJECT, which requires restricting access to this data to authorized administrators. The aspect of subjects getting access to information by residuals left in resources assigned to the subject is addressed by O.RESIDUAL_INFORMATION. The aspect of a guest partition getting access to data from other partitions by sharing data with another guest partition is addressed by the security objective O.PARTITION_ISOLATION.

OE.PHYSICAL supports countering the threat by prohibiting unauthorized persons to access TOE resources by physically manipulating the TOE hardware.

OE.NETWORK supports countering the threat by prohibiting attempts to use irregular network signals to attack the TOE and potentially damage the network adapter in way that may cause changes in the TOE configuration.

OE.REMOTE_IT_PRODUCT supports countering the threat by prohibiting unauthorized access to the TOE using a remote product for accessing the TOE that cannot be trusted to perform its functions correctly.

OE.TRUSTED_ROOT supports countering the threat by prohibiting that any untrusted application is installed in the root partition. This prohibits that any untrusted program may attempt to use the Server 2008 system call interface in order to explore a way to bypass the access control policies enforced. Only the defined applications required for virtualization support and management are installed and those are part of the TSF.

T.PARTITION_COMPROMISE

The TSF of a correctly configured TOE may allow software within a guest partition to escalate its processor or memory related privileges in a way that would not be possible when the software is executed on the real hardware. (Note: The software within a guest partition is not subject to this evaluation and therefore any flaw within a guest partition's software that allows such a privilege escalation is beyond the scope of this evaluation. Also the software a hypervisor aware operating system installs when executing on the TOE is not part of this evaluation).

The threat of software within a partition escalating its hardware privileges within the partition by using functions provided by the TSF is addressed by security objective O.PRESERVE_PART_PRIV, which requires that this is not possible.

OE.HW_MECHANISMS supports countering the threat by requiring the hardware to work correctly in accordance with its specification. The TOE relies on the documented functions of the underlying hardware to correctly enforce the same separation and privilege policy that is enforced when the software runs on a real system.

OE.HW_SIDE_EFFECTS supports countering the threat by requiring the hardware to not expose undocumented side effects which, although not contradicting the specification, may undermine the separation within a partition.

Organisational Security Policies

P.ACCOUNTABILITY

> *The TOE shall provide the capability to make available information regarding the occurrence of security relevant events.*

This organizational security policy is addressed by the security objective O.AUDIT_GENERATION, which requires the TOE to provide an audit function for security relevant events that allows to make subjects responsible for their actions

P.CONFIGURATION

> *The TOE shall provide functions that allow authorized administrators to perform the setup and initialization of the TOE in accordance with the security policies for configuration and administration issued by the organization responsible for the operation of the TOE.*

This security policy is addressed by the security objective O.INIT_SECURE_STATE which requires the TOE to provide the administrative functions that allow an authorized administrator to setup and initialize the TOE such that it starts in a secure state.

OE.SUBJECT_ALLOCATION supports this organisational security policy such that authorized administrators do not define configurations that contradict the separation policy an organization wants to enforce.

OE.TRUSTED_INDIVIDUAL supports this organisational security policy by requiring authorized administrators to not misuse his privileges.

OE.REMOTE_IT_PRODUCT supports this organisational security policy by requiring that remote IT products used for managing or accessing the TOE are trusted to correctly pass their user's actions to the TOE and not to send requests to the TOE that have not been initiated by the user of the product.

OE.TRUSTED_ROOT supports this organisational security policy by requiring that no untrusted application is installed in the root partition that may attempt to tamper with the configuration as defined by the trusted administrator.

Mapping security objectives for the operational environment to the assumption backing them

The following table maps the security objectives for the operational environment to the assumptions that back those objectives.

OE.PHYSICAL	This security objective for the operational environment is backed by assumption A.PHYSICAL.
OE.HW_MECHANISMS	This security objective for the operational environment is backed by assumption A.CORRECT_HARDWARE.
OE_HW_SIDE_EFFECTS	This security objective for the operational environment is backed by assumption A.CORRECT_HARDWARE.
OE.SUBJECT_ALLOCATION	This security objective for the operational environment is backed by assumption A.SUBJECT_ALLOCATION
OE.TRUSTED_INDIVIDUAL	This security objective for the operational environment is backed by assumption A.ADMIN.
OE.REMOTE_ADMIN	This security objective for the operational environment is backed by assumption A.REMOTE_ADMIN.
OE.REMOTE_IT_PRODUCT	This security objective for the operational environment is backed by assumption A.REMOTE_IT_PRODUCTS.
OE.NETWORK	This security objective for the operational environment is backed by assumption A.PHYSICAL.
OE.TRUSTED_ROOT	This security objective for the operational environment is backed by assumptions A.DEFINED_INSTALL and A.CLEAN_ROOT.
OE.PARTITION_USER	This security objective for the operational environment is backed by assumption A.PARTITION_CONNECT
OE.MEMORY_MANAGEMENT	This security objective for the operational environment is backed by assumption A.MEMORY_MANGEMENT.

Table 9: Mapping security objectives for the operational environment to assumptions

5 Extended Components Definition

This Security Target does not include any extended components.

6 Security Requirements

This section defines the security functional and security assurance requirements that apply for the evaluation of Microsoft Hyper-V.

6.1 Security Functional Requirements

As mentioned in the TOE introduction and explained in more detail in the TOE summary specification, the TSF for this evaluation consist of the hypervisor layer and the software within the root partition. The root partition includes a Server 2008 Server Core installation augmented by a number of components specific to Hyper-V.

The security functionality described in this Security Target is partly implemented by the hypervisor layer and the Hyper-V specific components in the root partition and partly by security functionality of the Server 2008 Server Core part. To differentiate this, the security functional requirements have been split into two parts. The first part describes the security functional requirements implemented by the hypervisor layer together with the Hyper-V specific components of Server 2008, the second part describes security functionality implemented by the Server Core product as configured and used in the root partition. In the cases where a security functional requirement listed in the first part includes aspects implemented by the Server Core part, those aspects are marked blue.

In addition, assignments or selections performed on the components are marked in **bold** and refinements performed are marked in **<u>bold and underlined</u>**.

The security functional requirements have been derived using the following paradigms for Hyper-V:

- Hyper-V has two types of "users":
  - Human users that act as administrators to configure and manage the TOE
  - Partitions as entities outside of the TOE that "use" the functions of the TSF.

    Note that the human users need to be identified and authenticated while the partitions are created and operate under the complete control of the TOE and therefore only need to be identified.
- Hyper-V has two types of access control policies:
  - A "Partition Management Access Control Policy" which controls the actions of administrative users. This policy allows controlling access of administrative users to management operations. The functionality allows grouping of operations to "tasks" and the assignment of tasks and/or individual operations to roles. Roles can than be assigned to administrative users.
  - A "Partition Access Control Policy" which controls the access of partitions to virtualized devices

    Note that for the partition management access control policy the TOE relies on the generic access control functionality of Server 2008.

- The security attributes used in the enforcement of the policies are the following:
    - Human users (administrators):
        - Identity of the user
        - User role
    - Partitions:
        - Identity of the Partition
        - Partition privileges
    - Hyper-V configuration data:
        - Access control lists associated with the data

      Note that for the partition management access control policy the TOE relies on the generic access control functionality of Server 2008.
    - Virtualized devices:
        - Access control list associated with the virtualized device
- Residual information protection
    - Hyper-V removes all information from resources before a resource is assigned to a partition
- Audit Policy:
    - Hyper-V allows to audit the following Hyper-V specific events:
        - Failure to start the hypervisor
        - Partition creation
        - Partition deletion
        - Critical hypervisor error

      For the management and protection of the audit trail as well the evaluation of the audit records by authorized administrators the generic functions of Windows Server 2008 are used. Those include:
        - Access control for the audit trail
        - Selection of events that are audited

        - Tools to review the audit trail
        - Actions performed to ensure that audit records are not lost

- Management Policy:
    - Hyper-V allows authorized administrators to manage the following Hyper-V specific aspects:
        - Creation and deletion of partitions
        - Assignment of virtualized resources to partitions
        - Definition of maximum quota of resources (CPU time, memory) for partitions
- Protection of TSF data:
    - Hyper-V uses the time stamp provided by Server 2008

Of this security functionality the following parts are completely implemented by the Server Core components within the root partition:

- Identification and authentication of administrative users
- Management of the security attributes of administrative users
- Access control policy for Server 2008 files and objects
- Management of the Server 2008 access control policy
- Generation of audit events specific for Server 2008
- Management and protection of the audit trail
- Review of audit records
- Management of date and time

6.1.1 Security Functional Requirements implemented by Hyper-V

6.1.1.1 Security Audit (Class FAU)

FAU_GEN.1 (H) Audit data generation (Hyper-V)

FAU_GEN.1.1 The TSF shall be able to generate an audit record of the following auditable events:

1. Start-up and shutdown of the audit functions;
2. All auditable events for the **not specified** level of audit; and
3. **The following hypervisor specific events:**
   - **Failure to start the hypervisor**
   - **Creation of a partition (by the hypervisor)**
   - **Deletion of a partition (by the hypervisor)**
   - **Failure condition detected within the hypervisor**
   - **Modification of the Hyper-V AzMan policy**
4. **The following partition management specific events:**
   - **Access checks performed by AzMan on Hyper-V management operations**
   - **Reconfigure partition (Virtual Machine)**

Application Note: Hyper-V uses the Windows Server 2008 audit system. Startup and shutdown of this audit system are recorded by Server 2008 in the root partition without involvement of the Hyper-V specific functionality.

FAU_GEN.1.2 The TSF shall record within each audit record at least the following information:

a) Date and time of the event, type of event, subject identity (if applicable), and the outcome (success or failure) of the event; and

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, **and no other security relevant information**.

Application Note: Date and Time are inserted by the Server 2008 audit function in the root partition..

FAU_GEN.2 (H) User identity association (Hyper-V)

FAU_GEN.2.1 For audit events resulting from actions of identified users, the TSF shall be able to associate each auditable event with the identity of the user (**partition or the administrator**) that caused the event.

Application Note: Hyper-V supports two types of users: partitions and administrators. The identity of the "user" that causes the event is recorded for both types of users, although

6.1.1.2 User Data Protection (Class FDP)

FDP_ACC.1 (H-PM) Subset access control (Hyper-V partition management)

FDP_ACC.1.1 The TSF shall enforce the **Partition Management Access Control Policy** on **Server 2008 subjects acting on behalf of an administrative user, individual management operations as objects and performing the operation as functions.**

FDP_ACC.1 (H-DA) Subset access control (Hyper-V device access)

FDP_ACC.1.1 The TSF shall enforce the **Partition Device Access Control Policy** on **partitions as subjects, virtualized and synthesized devices as objects and device access as function.**

FDP_ACF.1 (H-PM) Security attribute based access control (Hyper-V partition management)

FDP_ACF.1.1 The TSF shall enforce the **Partition Management Access Control Policy** to objects based on the following:

a) The identity of the administrative user, his roles, the tasks and operations assigned to the roles and the additional BizRules assigned to the tasks.

Application Note: The identity of the administrative user is established and authenticated by the Server 2008 functions in the root partition.

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: **access (i. e performing an operation) is allowed if either the operation itself or a task that contains the operation is assigned to a role that has been assigned to the administrative user and the BizRules assigned to one of tasks involved in the access decision don't disallows access**.

FDP_ACF.1.3 The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: **none**

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the **none**.

FDP_ACF.1 (H-DA) Security attribute based access control (Hyper-V device access)

FDP_ACF.1.1 The TSF shall enforce the **Partition Device Access Control Policy** to objects based on the following:

a) **the type of device**

b) **the identity of the partition**

c) **the configuration data for the partition.**

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: **a partition is allowed to access a virtualized or synthesized device if the partition configuration has this device assigned to the partition**.

FDP_ACF.1.3 The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: **the device is a virtualized S3 Trio Video Card or a virtualized Intel 440 BX chipset.**

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on **no additional rule**.

FDP_RIP.1 (H) Subset residual information protection (Hyper-V)

FDP_RIP.1.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the **allocation of the resource to** the following objects: **virtual memory allocated to a partition, virtual and synthesized devices allocated to a partition, virtual processors allocated to a partition.**

6.1.1.3 Identification and Authentication (Class FIA)

FIA_ATD.1 (H) User attribute definition (Hyper-V partitions)

FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to individual users:

a) **For partitions:**

  a. **Partition identifier (number)**

  b. **Partition configuration data**

  c. **Partition privileges**

Application Note: The security attributes for administrative users are defined in the instantiation of FIA_ATD.1 in Server 2008 section of this chapter.

FIA_UID.2 (H) User identification before any action (Hyper-V partitions)

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

Application Note: The instantiation of FIA_UID.1 in Server 2008 section of this chapter covers the administrative user.

FIA_USB.1 (H) User-subject binding (Hyper-V)

FIA_USB.1.1 The TSF shall associate the following user security attributes with subjects acting on the behalf of that user:

a) For subjects acting on behalf of partitions:

  a. The identity of the partition

  b. The configuration data of the partition

  c. The privilege vector of the partition

FIA_USB.1.2 The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users:

- **Every subject acting on behalf of a partition will be assigned the security attributes associated with the partition on whose behalf the subject will act.**

FIA_USB.1.3 The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users: **subjects acting on behalf of partitions can not add additional security attributes beyond those initially assigned.**

Application Note: The instantiation of FIA_USB.1 in Server 2008 section of this chapter covers the user-subject binding for administrative user.

6.1.1.4 Security Management (Class FMT)

FMT_MSA.1 (H) Management of security attributes (Hyper-V)

FMT_MSA.1.1 The TSF shall enforce the **Partition Management Access Control Policy** to restrict the ability to **query and modify** the security attributes **partition configuration data** to **authorized administrators that have the authority for those operations assigned in the AzMan policy for Hyper-V**.

FMT_MSA.2 (H) Secure security attributes (Hyper-V)

FMT_MSA.2.1 The TSF shall ensure that only secure values are accepted for **partition configuration data**.

FMT_MSA.3 (H-PM) Static attribute initialization (Hyper-V)

FMT_MSA.3.1 The TSF shall enforce the **Partition Management Access Control Policy** to provide **restrictive** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow the **authorized administrator with access to the AzMan policy file** to specify alternative initial values to override the default values when an object or information is created.

FMT_MSA.3 (H-DA) Static attribute initialization (Hyper-V)

FMT_MSA.3.1 The TSF shall enforce the **partition device access control policy** to provide **restrictive** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow the administrator **authorized by the Hyper-V AzMan policy to modify partition configuration data** to specify alternative initial values to override the default values when an object or information is created except for those security attributes that nobody is allowed to modify.

FMT_MTD.1 (H-1) Management of TSF data (Hyper-V)

FMT_MTD.1.1 The TSF shall restrict the ability to **change_default and modify** the **partition's access to hypervisor calls related to inter-partition communication and the creation of guest partitions** to **nobody**.

Application Note: The pseudo-role "nobody" is used here to specify that those actions can not be performed by anybody, not even the authorized administrator. "nobody" is a "role" that can not be assigned to any user. This is the only way to define this without defining an extended requirement.

FMT_MTD.1(H-2) Management of TSF data (Hyper-V)

FMT_MTD.1.1 The TSF shall restrict the ability to **change_default and modify** the **resource quota assigned to partition** to **administrators authorized by the Hyper-V AzMan policy to perform those actions**.

FMT_MTD.1(H-3) Management of TSF data (Hyper-V)

FMT_MTD.1.1 The TSF shall restrict the ability to **modify** the **Hyper-V AzMan policy** to **administrators authorized to access the Hyper-V authorization store**.

FMT_REV.1 (H) Revocation (Hyper-V)

FMT_REV.1.1 The TSF shall restrict the ability to revoke **the assignment of devices** associated with the **partitions** under the control of the TSF to **administrators**.

FMT_REV.1.2 The TSF shall enforce the rules **that the administrator needs to be authorized by the Hyper-V AzMan policy to perform those actions**.

Application Note: Revocation is not bound to one specific role but to a privilege that can be assigned to installation defined management roles. The specification of FMT_REV.1 within part 2 of the CC is too narrow and therefore FMT_REV.1.2 has been used to correctly describe the way revocation can be performed in the TOE.

FMT_SMF.1 (H) Specification of Management Functions (Hyper-V)

FMT_SMF.1.1 The TSF shall be capable of performing the following management functions:

- **Management of partition configuration data**
- **Management of virtual switches**
- **Defining, deleting, starting and stopping partitions**
- **Management of the Hyper-V AzMan policy**.

6.1.1.5 Resource Utilisation (Class FRU)

FRU_RSA.1 (H) Maximum quotas (Hyper-V)

FRU_RSA.1.1 The TSF shall enforce maximum quotas of the following resources: **maximum CPU time per virtual CPU, maximum amount of the partition (guest) physical memory** that **partitions** can use **over a specified period of time**.

6.1.2 Security Functional Requirements for Server 2008 (Server Core)

The following security functional requirements are implemented by the generic functionality of Server 2008 Server Core in the root partition without using functionality of the Hyper-V specific parts of the TOE. They are listed in this Security Target because Hyper-V related functionality relies on those functions. Windows Server 2008 Server Core provides more security functions than specified here, but those functions are not directly supporting the security functionality of Hyper-V.

The Server 2008 Server Core in the root partition is protected from direct access by untrusted human users and is assumed to not have any untrusted application installed. The architecture of Hyper-V also ensures that the code and TSF data within the root partition is protected from direct access by any guest partition. The memory areas used by VMBus are the only objects shared between the root partition and individual guest partitions where there separate shared memory areas for each guest partition. This prohibits that there is any shared memory between guest partitions.

6.1.2.1 Security Audit (Class FAU)

FAU_GEN.1 Audit data generation (Server 2008)

FAU_GEN.1.1 The TSF shall be able to generate an audit record of the following auditable events:

a) Start-up and shutdown of the audit functions;

b) All auditable events for the **not specified** level of audit; and

c) The following Server Core 2008 specific events:

All authentication attempts of authorized administrators

Modification of the authentication information of authorized administrators

Modification to the security attributes of authorized administrators

Access to objects protected by the partition management access control policy

FAU_GEN.1.2 The TSF shall record within each audit record at least the following information:

a) Date and time of the event, type of event, subject identity (if applicable), and the outcome (success or failure) of the event; and

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST:

i. **The name of the object (if the audit event is for an operation on an object)**

ii. **For changes to TSF data (except for authentication data): the new and the old values of the data**

FAU_GEN.2 User identity association

FAU_GEN.2.1 For audit events resulting from actions of identified users, the TSF shall be able to associate each auditable event with the identity of the user that caused the event.

FAU_SAR.1 Audit review

FAU_SAR.1.1 The TSF shall provide **authorized administrators** with the capability to read **all audit information** from the audit records.

FAU_SAR.1.2 The TSF shall provide the audit records in a manner suitable for the **authorized administrator** to interpret the information **using a tool to access the audit records**.

Application Note: FAU_SAR.1.2 has been refined to describe more precisely that only authorized administrators are able to read the audit records (as indicated in FAU_SAR.1.1) and this can be done using a defined tool, which presents the audit records in human readable format. This refinement has been performed for consistency with the Security Target for the (ongoing) Server 2008 evaluation.

FAU_SAR.2 Restricted audit review

FAU_SAR.2.1 The TSF shall prohibit all users read access to the audit records, except those users that have been granted explicit read-access.

FAU_STG.1 Protected audit trail storage

FAU_STG.1.1 The TSF shall protect the stored audit records in the audit trail from unauthorised deletion.

FAU_STG.1.2 The TSF shall be able to **prevent** unauthorised modifications to the stored audit records in the audit trail.

FAU_STG.3 Action in case of possible audit data loss

FAU_STG.3.1 The TSF shall **generate an alarm to the authorized administrator** if the audit trail exceeds **the authorized administrator specified log size**.

FAU_STG.4 Prevention of audit data loss

FAU_STG.4.1 The TSF shall **provide the authorized administrator the capability to prevent audited events, except those taken by the authorized administrator (in the context of performing TOE maintenance) and generate an alarm to the authorized administrator** if the audit trail is full.

Application Note: FAU_STG.4.1 has been refined to describe more precisely the way the loss of audit records is prevented. This refinement has been performed for consistency with the Security Target for the (ongoing) Server 2008 evaluation.

6.1.2.2 Identification and Authentication (Class FIA)

FIA_AFL.1 Authentication failure handling

FIA_AFL.1.1 The TSF shall detect when **an authorized administrator configurable positive integer of consecutive** unsuccessful authentication attempts occur related to **any authorized administrator authentication process**.

FIA_AFL.1.2 When the defined number of **consecutive** unsuccessful authentication attempts has been **met or surpassed**, the TSF shall **disable the account for an authorized administrator configurable time period**.

Application Note: FIA_AFL.1.1 and FIA_AFL.1.2 have been refined for consistency with the Security Target for the (ongoing) Server 2008 evaluation.

FIA_ATD.1 User attribute definition

FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to individual **authorized administrators and user allowed to connect to a partition**:

a) **user identifier**

b) **group memberships**

c) **authentication data**

d) **security relevant roles**

e) **allowable time and day of logon**.

Application Note: FIA_ATD.1.1 has been refined to restricts its applicability to authorized administrators as users only. Partitions as the other type of users are covered by the instantiation of FIA_ATD.1 in the Hyper-V section of this chapter.

FIA_SOS.1 Verification of secrets

FIA_SOS.1.1 The TSF shall provide a mechanism to verify that secrets **used to authenticate an authorized administrator or a user that wants to connect to a partition** meet **the property that a random attempt to guess the secret will succeed with a probability that is less than one in 2 x 10^{15}**.

Application Note: FIA_SOS.1.1 has been refined to restricts its applicability to authorized administrators as users only. Partitions as the other type of users are not authenticated and therefore there is no secret used for them. Note that in the case AD-DS is used for user authentication, this requirement can only be enforced by AD-DS in the TOE environment.

FIA_UAU.1 User authentication

FIA_UAU.1.1 The TSF shall allow **read access to public objects** on behalf of the **authorized administrator** **or a user that wants to connect to a partition** to be performed before the **authorized administrator** **or the user that wants to connect to a partition** is authenticated.

FIA_UAU.1.2 The TSF shall require each **authorized administrator** to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

Application Note: FIA_UAU.1.1 has been refined to restricts its applicability to authorized administrators as users only. Partitions as the other type of users are not authenticated and therefore there are no user authentication requirements for partitions. Note that other users than the authorized administrators are only allowed to connect to a partition they are authorized to connect to.

FIA_UAU.7 Protected authentication feedback

FIA_UAU.7.1 The TSF shall provide only **obscured feedback** to the **authorized administrator or a user that wants to connect to a partition** while the authentication is in progress.

Application Note: FIA_UAU.7.1 has been refined to restrict its applicability to authorized administrators as users only. Partitions as the other type of users are not authenticated and therefore there are no user authentication requirements for partitions.

FIA_UID.1 User identification

FIA_UID.1.1 The TSF shall allow **read access to public objects** on behalf of the **authorized administrator** **or a user that wants to connect to a partition** to be performed before the **authorized administrator** **or the user that wants to connect to a partition** is identified.

FIA_UID.1.2 The TSF shall require each **authorized administrator or a user that wants to connect to a partition** to be successfully identified before allowing any other TSF-mediated actions on behalf of that **authorized administrator or the user that wants to connect to a partition**.

Application Note: FIA_UID.1.1 and FIA_UID.1.2 have been refined to restrict its applicability to authorized administrators as users only. The identification requirements for partitions as the other type of users are defined in the Hyper-V section of this chapter.

FIA_USB.1 User-subject binding

FIA_USB.1.1 The TSF shall associate the following **authorized administrator or a user that wants to connect to a partition** security attributes with subjects acting on the behalf of that **authorized administrator or that user that wants to connect to a partition**:

a) **The user identity which is associated with auditable events.**

b) **The user identity or identities which are used to enforce the Partition Management Access Control Policy.**

c) **The roles used to enforce the Partition Management Access Control Policy.**

FIA_USB.1.2 The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users:

a) **Every subject acting on behalf of an authorized administrator will be assigned a subset of the security attributes associated with the authorized administrator on whose behalf the subject will act.**

FIA_USB.1.3 The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users: **subjects acting on behalf of authorized administrators can not add additional security attributes beyond those initially assigned.**

Application Note: FIA_USB.1 has been refined to restrict its applicability to authorized administrators as users only. The requirements for user-subject binding for partitions as the other type of users are defined in the Hyper-V section of this chapter.

6.1.2.3 Security Management (Class FMT)

FMT_MTD.1 (1) Management of TSF data (Server 2008)

FMT_MTD.1.1 The TSF shall restrict the ability to **manage** the **values of security attributes associated with user authentication data** to **authorized administrators**.

Application Note: This applies for locally managed user authentication data only. In the case of authentication data managed in Active Directory, this aspect is addressed by a security objective for the TOE environment.

FMT_MTD.1 (2) Management of TSF data (Server 2008)

FMT_MTD.1.1 The TSF shall restrict the ability to **manage** the **security-relevant TSF data except for audit records, user security attributes, authentication data** to **the authorized administrator**.

FMT_MTD.1 (3) Management of TSF data (Server 2008)

FMT_MTD.1.1 The TSF shall restrict the ability to **query, delete, and clear** the **audit trail** to **the authorized administrator**.

FMT_MTD.1 (4) Management of TSF data (Server 2008)

FMT_MTD.1.1 The TSF shall restrict the ability to **initialize** the **user security attributes, other than authentication data** to **the authorized administrator**.

Application Note: This applies for locally managed user security attributes only. In the case of user security attributes managed in Active Directory, this aspect is addressed by a security objective for the TOE environment.

FMT_MTD.1 (5) Management of TSF data (Server 2008)

FMT_MTD.1.1 The TSF shall restrict the ability to **modify** the **authentication data** to **the authorized administrator and users authorized to modify their own authentication data**.

FMT_SMF.1 Specification of Management Functions (Server 2008)

FMT_SMF.1.1 The TSF shall be capable of performing the following management functions:

- **Management of audit data**
- **Management of access rights for the Partition Management Access Control Policy**

- **Definition of tasks as groups of controlled operations**
- **Assignment of tasks and/or controlled operations to roles**
- **Assignment of roles to users**
- **Management of user security attributes of authorized administrators**
- **Management of authentication data of authorized administrators.**

FMT_SMR.1 Security roles

FMT_SMR.1.1 The TSF shall maintain the roles **authorized administrator, nobody** **for administrative users**.

FMT_SMR.1.2 The TSF shall be able to associate **administrative** users with roles **except for the role “nobody” which can not be assigned to any user**.

Application Note: The TOE allows to define tasks as groups of operations and to assign tasks and/or individual controlled operations to roles. An arbitrary number of roles can be defined that way, allowing each installation to define the roles it requires and assign those roles to administrative users. Those users are then restricted in their administrative operations to those assigned to their role either directly or via a defined task. The refinement is required to indicate that roles only apply to administrative users, not to partitions. The “role” of “nobody” is used to specify that specific security attributes are not manageable but fixed.

6.1.2.4 Protection of the TSF (Class FPT)

FPT_STM.1 Reliable time stamps

FPT_STM.1.1 The TSF shall be able to provide reliable time stamps.

6.2 Security Assurance Requirements

The SARs for the TOE are the EAL 4 components augmented with ALC_FLR.3 as specified in Part 3 of the CC. No operations are applied to the assurance components.

Assurance Class	Assurance components
ADV: Development	ADV_ARC.1 Security architecture description
	ADV_FSP.4 Complete functional specification
	ADV_IMP.1 Implementation representation of the TSF
	ADV_TDS.3 Basic modular design
AGD: Guidance documents	AGD_OPE.1 Operational user guidance
	AGD_PRE.1 Preparative procedures
ALC: Life-cycle support	ALC_CMC.4 Production support, acceptance procedures

Assurance Class	Assurance components
	and automation
	ALC_CMS.4 Problem tracking CM coverage
	ALC_DEL.1 Delivery procedures
	ALC_DVS.1 Identification of security measures
	ALC_LCD.1 Developer defined life-cycle model
	ALC_TAT.1 Well defined development tools
	ALC_FLR.3 Systematic flaw remediation
ASE: Security Target evaluation	ASE_CCL.1 Conformance claims
	ASE_ECD.1 Extended components definition
	ASE_INT.1 ST introduction
	ASE_OBJ.2 Security objectives
	ASE_REQ.2 Derived security requirements
	ASE_SPD.1 Security problem definition
	ASE_TSS.1 TOE summary specification
ATE: Tests	ATE_COV.2 Analysis of coverage
	ATE_DPT.2 Testing: security enforcing modules
	ATE_FUN.1 Functional testing
	ATE_IND.2 Independent testing – sample
AVA: Vulnerability assessment	AVA_VAN.3 Focused vulnerability analysis

Table 10: Security assurance components (EAL 4 augmented by ALC_FLR.3)

6.3 Security requirements rationale

This section provides the rationale for the internal consistency and completeness of the security functional requirements defined in this Security Target.

6.3.1 Internal consistency of requirements

This section shows that the security functional requirements are internally consistent and mutually supportive:

FAU_GEN.1 (H) and FAU_GEN.2 (H) require that the TOE provides the capability to audit specific events related to Hyper-V functions. The events are general failure conditions that either do not allow a correct start of the hypervisor or failure conditions detected by the hypervisor. In addition the creation and deletion of partitions shall generate an audit event.

This functionality is supported by the identification of the partitions (defined in FIA_ATD.1 (H) which defines the partition identifier as a security attribute and FIA_UAU.2 (H) which requires the identification of a partition). In addition this function is supported by the provision of a time stamp

(FTP_STM.1) and the requirements for the Server 2008 Server Core on the protection, storage and management of the audit trail.

The functionality related to the Partition Device Access Control Policy is defined in FDP_ACC.1 (H-DA) and FDP_ACF.1 (H-DA) which define the subjects, objects, types of access and access control rules. In addition FMT_MSA.3 (H) defines that restrictive default values (no access) are assigned for the security attributes (access control entries). As a result, for a partition to have access to a device that is controlled by the device access control policy, the right to access must be assigned by an administrator when creating or managing the partition configuration data. FMT_MSA.1 (H) defines that only an authorized administrator is allowed to query and modify the partition configuration data and FMT_REV.1 defines that the revocation of a partitions access to a device is restricted to an administrator. This restricts all management activities to authorized administrators. On the other hand, the authorizations of individual administrators themselves with respect to access to specific partition configuration data and management functions can be controlled by the partition management access control policy defined by FDP_ACC.1 (H-PM) and FDP_ACF.1 (H-PM). To be able to start this system of management restrictions for authorized administrators, the TOE needs to start with an administrator that is not restricted and which can then define other administrative users with more restricted rights. This is defined in FMT_MSA.3 (H-PM), which requires permissive default values for the initial administrator.

6.3.2 Complete coverage: security objectives

This section demonstrates that the functional components selected for this profile provide complete coverage of the defined security objectives. The mapping of components to security objectives is depicted in the following table.

Security Objective	Related Security Functional Requirement
O.PARTITION_ACCESS	FDP_ACC.1 (H-DA) FDP_ACF.1 (H-DA) FIA_ATD.1 (H) FIA_UID.2 (H) FIA_USB.1 (H) FMT_MSA.3 (H-DA)
O.AUDIT_GENERATION	FAU_GEN.1 (H) FAU_GEN.2 (H) FAU_GEN.1 FAU_GEN.2 FAU_SAR.1 FAU_SAR.2 FAU_STG.1 FAU_STG.3 FAU_STG.4 FPT_STM.1
O.AUTHORIZED_SUBJECT	FIA_AFL.1 FIA_ATD.1 FIA_SOS.1 FIA_UAU.1 FIA_UAU.7 FIA_UID.1 FIA_USB.1
O.INIT_SECURE_STATE	FMT_MSA.2 (H) FMT_MTD.1 (H-1)

Security Objective	Related Security Functional Requirement
O.MANAGE	FDP_ACC.1 (H-PM) FDP_ACF.1 (H-PM) FMT_MSA.1 (H) FMT_MSA.3 (H-PM) FMT_MTD.1 (H-2) FMT_MTD.1 (H-3) FMT_REV.1 (H) FMT_SMF.1 (H) FMT_MTD.1(1) FMT_MTD.1(2) FMT_MTD.1(3) FMT_MTD.1(4) FMT_MTD.1(5) FMT_SMF.1 FMT_SMR.1
O.RESIDUAL_INFORMATION	FDP_RIP.1 (H)
O.RESOURCE_ALLOCATION	FRU_RSA.1 (H)
O.PARTITION_ISOLATION	ADV_ARC
O.PRESERVE_PART_PRIV	ADV_ARC

Table 11: Mapping Security Objectives to Security Functional Requirements

O.PARTITION_ACCESS is fully addressed by the access control policy defined by FDP_ACC.1 (H-DA) and FDP_ACF.1 (H-DA), the identification of the partitions as defined by FIA_ATD.1 (H), FIA_UID.2 (H) and FIA_USB.1 (H), and the management of the access control policy defined in FMT_MSA.3 (H-DA).

O.AUDIT_GENERATION is fully addressed by the audit generation of the hypervisor as defined in FAU_GEN.1 and FAU_GEN.2, supported by the generation of a reliable time stamp FPT_STM.1. In addition the generation of the audit records in the Server 2008 part is addressed by FAU_GEN.1 and FAU_GEN.2. The storage and protection of the audit trail as well as the evaluation of the audit records is addressed by the following requirements for the Server 2008 part: FAU_SAR.1, FAU_SAR.2, FAU_STG.1, FAU_STG.3, FAU_STG.4.

O.AUTHORIZED_SUBJECT is addressed by the SFRs related to the authentication of users in the Server 2008 part.

O.INIT_SECURE_STATE is addressed by FMT_MSA.2 (H) together with FMT_MTD.1 (H-1).

O.MANAGE is addressed by the various SFRs of the Security Management class as well as the management access control policy (FDP_ACC.1 (H-PM) and FDP_ACF.1 (H-PM)) which defines the access rights and privileges an administrative user needs to have to perform management operations.

O.RESIDUAL_INFORMATION is addressed by FDP_RIP.1 (H).

O.RESOURCE_ALLOCATION is addressed by FRU_RSA.1 (H).

The two security objectives O.PARTITION_ISOLATION and O.PRESERVE_PART_PRIV define architectural properties and are therefore not mapped to SFRs but to the TOE architecture.

6.3.3 Security requirements coverage

The following table shows that each security functional requirement addresses at least one objective.

SFR	Security Objective addressed by the SFR
FAU_GEN.1 (H)	O.AUDIT_GENERATION
FAU_GEN.2 (H)	O.AUDIT_GENERATION
FDP_ACC.1 (H-DA)	O.PARTITION_ACCESS
FDP_ACC.1 (H-PM)	O.MANAGE
FDP_ACF.1 (H-DA)	O.PARTITION_ACCESS
FDP_ACF.1 (H-PM)	O.MANAGE
FDP_RIP.1 (H)	O.RESIDUAL_INFORMATION
FIA_ATD.1 (H)	O.PARTITION_ACCESS
FIA_UID.2 (H)	O.PARTITION_ACCESS
FIA_USB.1 (H)	O.PARTITION_ACCESS
FMT_MSA.1 (H)	O.MANAGE
FMT_MSA.2 (H)	O.INIT_SECURE_STATE
FMT_MSA.3 (H-PM)	O.MANAGE
FMT_MSA.3 (H-DA)	O.PARTITION_ACCESS
FMT_MTD.1 (H-1)	O.INIT_SECURE_STATE

SFR	Security Objective addressed by the SFR
FMT_MTD.1 (H-2)	O.MANAGE
FMT_MTD.1 (H-3)	O.MANAGE
FMT_REV.1 (H)	O.MANAGE
FMT_SMF.1 (H)	O.MANAGE
FRU_RSA.1 (H)	O.RESOURCE_ALLOCATION
FAU_GEN.1	O.AUDIT_GENERATION
FAU_GEN.2	O.AUDIT_GENERATION
FAU_SAR.1	O.AUDIT_GENERATION
FAU_SAR.2	O.AUDIT_GENERATION
FAU_STG.1	O.AUDIT_GENERATION
FAU_STG.3	O.AUDIT_GENERATION
FAU_STG.4	O.AUDIT_GENERATION
FIA_AFL.1	O.AUTHORIZED_SUBJECT
FIA_ATD.1	O.AUTHORIZED_SUBJECT
FIA_SOS.1	O.AUTHORIZED_SUBJECT
FIA_UAU.1	O.AUTHORIZED_SUBJECT
FIA_UAU.7	O.AUTHORIZED_SUBJECT
FIA_UID.1	O.AUTHORIZED_SUBJECT
FIA_USB.1	O.AUTHORIZED_SUBJECT
FMT_MTD.1(1)	O.MANAGE
FMT_MTD.1(2)	O.MANAGE
FMT_MTD.1(3)	O.MANAGE
FMT_MTD.1(4)	O.MANAGE

SFR	Security Objective addressed by the SFR
FMT_MTD.1(5)	O.MANAGE
FMT_SMF.1	O.MANAGE
FMT_SMR.1	O.MANAGE
FPT_STM.1	O.AUDIT_GENERATION

Table 12: Mapping Security Functional Requirements to Security Objectives

This table shows that each security functional requirement contributes to satisfy at least one security objective.

6.3.4 Security requirements dependency analysis

The following table shows the dependencies which exist. In this table the SFRs are listed with their dependencies. In the case of multiple instantiations of SFRs it also shows, which of those SFRs actually resolves the dependency.

SFRs related to the Server 2008 Server Core functionality are marked in blue.

SFR	Dependency	Resolved?
	Hyper-V	
FAU_GEN.1 (H)	FPT_STM.1	Yes
FAU_GEN.2 (H)	FAU_GEN.1 (H) FIA_UID.1 (H)	Yes Yes
FDP_ACC.1 (H-PM)	FDP_ACF.1 (H-PM)	Yes
FDP_ACC.1 (H-DA)	FDP_ACF.1 (H-DA)	Yes
FDP_ACF.1 (H-PM)	FDP_ACC.1 (H-PM) FMT_MSA.3 (H-PM)	yes
FDP_ACF.1 (H-DA)	FDP_ACC.1 (H-DA) FMT_MSA.3 (H-DA)	yes
FDP_RIP.1 (H)	none	Yes
FIA_ATD.1 (H)	none	Yes

SFR	Dependency	Resolved?
FIA_UID.2 (H)	none	Yes
FIA_USB.1 (H)	FIA_ATD.1 (H)	Yes
FMT_MSA.1 (H)	FDP_ACC.1 or FDP_IFC.1 FMT_SMR.1 FMT_SMF.1	Yes Yes Yes
FMT_MSA.2 (H)	FDP_ACC.1 or FDP_IFC.1 FMT_MSA.1 FMT_SMR.1	Yes Yes Yes
FMT_MSA.3 (H-PM)	FMT_MSA.1 FMT_SMR.1	Yes Yes
FMT_MSA.3 (H-DA)	FMT_MSA.1 FMT_SMR.1	Yes Yes
FMT_MTD.1 (H-1)	FMT_SMR.1 FMT_SMF.1	Yes Yes
FMT_MTD.1 (H-2)	FMT_SMR.1 FMT_SMF.1	Yes Yes
FMT_MTD.1 (H-3)	FMT_SMR.1 FMT_SMF.1	Yes Yes
FMT_REV.1 (H)	FMT_SMR.1	Yes
FMT_SMF.1 (H)	none	Yes
FRU_RSA.1	none	Yes
Server 2008 Server Core		
FAU_GEN.1	FPT_STM.1	Yes
FAU_GEN.2	FAU_GEN.1 FIA_UID.1	Yes Yes
FAU_SAR.1	FAU_GEN.1 (H) FAU_GEN.1	Yes

SFR	Dependency	Resolved?
FAU_SAR.2	FAU_SAR.1	Yes
FAU_STG.1	FAU_GEN.1	Yes
FAU_STG.3	FAU_STG.1	yes
FAU_STG.4	FAU_STG.1	Yes
FIA_AFL.1	FIA_UAU.1	Yes
FIA_ATD.1	none	Yes
FIA_SOS.1	none	Yes
FIA_UAU.1	FIA_UID.1	Yes
FIA_UAU.7	FIA_UAU.1	Yes
FIA_UID.1	none	Yes
FIA_USB.1	FIA_ATD.1	Yes
FMT_MTD.1 (1-5)	FMT_SMR.1 FMT_SMF.1	Yes Yes
FMT_SMF.1	none	Yes
FMT_SMR.1	FIA_UID.1	Yes
FPT_STM.1	none	Yes

Table 13: Dependencies between Security Functional Requirements

This table shows that all the dependencies between the security functional requirements are resolved.

6.3.5 Operations performed on the SFRs

The following table maps the SFRs as stated in this Security Target to the definitions of the components in part 2 of the CC to show how the operations on the components have been performed.

FAU_GEN.1.1 The TSF shall be able to generate an audit record of the following auditable events: 5. Start-up and shutdown of the audit	**FAU_GEN.1.1** The TSF shall be able to generate an audit record of the following auditable events: 1. Start-up and shutdown of the audit

functions; 6. All auditable events for the **not specified** level of audit; and **7. The following hypervisor specific events:** • **Failure to start the hypervisor** • **Creation of a partition (by the hypervisor)** • **Deletion of a partition (by the hypervisor)** • **Failure condition detected within the hypervisor** • **Modification of the Hyper-V AzMan policy** **8. The following partition management specific events:** • **Defining the configuration parameter for a new partition** • **Modifying the configuration parameter for a partition** • **Deleting the configuration parameter associated with a partition** • **Access checks performed by AzMan on Hyper-V management operations**	functions; 2. All auditable events for the [selection, choose one of: *minimum, basic, detailed, not specified*] level of audit; and [assignment: *other specifically defined auditable events*].
FAU_GEN.1.2 The TSF shall record within each audit record at least the following information: • Date and time of the event, type of event, subject identity (if applicable), and the outcome (success or failure) of the event; and • For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, **and no other security relevant information**.	**FAU_GEN.1.2** The TSF shall record within each audit record at least the following information: a) Date and time of the event, type of event, subject identity (if applicable), and the outcome (success or failure) of the event; and b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [assignment: *other audit relevant information*].

FAU_GEN.2.1 For audit events resulting from actions of identified users, the TSF shall be able to associate each auditable event with the identity of the user (**partition or the administrator**) that caused the event.	**FAU_GEN.2.1** For audit events resulting from actions of identified users, the TSF shall be able to associate each auditable event with the identity of the user that caused the event.
FDP_ACC.1.1 The TSF shall enforce the **Partition Management Access Control Policy** on **Server 2008 subjects acting on behalf of an administrative user, individual management operations as objects and performing the operation as functions.**	**FDP_ACC.1.1** The TSF shall enforce the [assignment: *access control SFP*] on [assignment: *list of subjects, objects, and operations among subjects and objects covered by the SFP*].
FDP_ACC.1.1 The TSF shall enforce the **Partition Device Access Control Policy** on **partitions as subjects, virtualized and synthesized devices as objects and device access as function.**	**FDP_ACC.1.1** The TSF shall enforce the [assignment: *access control SFP*] on [assignment: *list of subjects, objects, and operations among subjects and objects covered by the SFP*].
FDP_ACF.1.1 The TSF shall enforce the **Partition Management Access Control Policy** to objects based on the following: **The identity of the administrative user, his roles, the tasks and operations assigned to the roles and the additional BizRules assigned to the tasks.**	**FDP_ACF.1.1** The TSF shall enforce the [assignment: *access control SFP*] to objects based on the following: [assignment: *list of subjects and objects controlled under the indicated SFP, and for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes*].
FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: **access (i. e performing an operation) is allowed if either the operation itself or a task that contains the operation is assigned to a role that has been assigned to the administrative user and the BizRules assigned**	**FDP_ACF.1.2** The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [assignment: *rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*].

to one of tasks involved in the access decision don't disallows access.	
FDP_ACF.1.3 The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: **none**	**FDP_ACF.1.3** The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: [assignment: *rules, based on security attributes, that explicitly authorise access of subjects to objects*].
FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the **none**.	**FDP_ACF.1.4** The TSF shall explicitly deny access of subjects to objects based on the [assignment: *rules, based on security attributes, that explicitly deny access of subjects to objects*].
FDP_ACF.1.1 The TSF shall enforce the **Partition Device Access Control Policy** to objects based on the following: **d) the type of device** **e) the identity of the partition** f) **the configuration data for the partition.**	**FDP_ACF.1.1** The TSF shall enforce the [assignment: *access control SFP*] to objects based on the following: [assignment: *list of subjects and objects controlled under the indicated SFP, and for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes*].
FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: **a partition is allowed to access a virtualized or synthesized device if the partition configuration has this device assigned to the partition**.	**FDP_ACF.1.2** The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [assignment: *rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*].
FDP_ACF.1.3 The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: **the device is a virtualized S3 Trio Video Card or a virtualized Intel 440 BX**	**FDP_ACF.1.3** The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: [assignment: *rules, based on security attributes, that explicitly authorise*

chipset.	*access of subjects to objects*].
FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on **no additional rule**.	**FDP_ACF.1.4** The TSF shall explicitly deny access of subjects to objects based on the [assignment: *rules, based on security attributes, that explicitly deny access of subjects to objects*].
FDP_RIP.1.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the **allocation of the resource to** the following objects: **virtual memory allocated to a partition, virtual and synthesized devices allocated to a partition, virtual processors allocated to a partition.**	**FDP_RIP.1.1** The TSF shall ensure that any previous information content of a resource is made unavailable upon the [selection: *allocation of the resource to, deallocation of the resource from*] the following objects: [assignment: *list of objects*].
FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to individual users: b) **For partitions:** a. **Partition identifier (number)** b. **Partition configuration data** c. **Partition privileges**	**FIA_ATD.1.1** The TSF shall maintain the following list of security attributes belonging to individual users: [assignment: *list of security attributes*].
FIA_UID.2.1 The TSF shall require each user (**partition)** to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.	**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_USB.1.1 The TSF shall associate the following user security attributes with subjects acting on the behalf of that user: **b) For subjects acting on behalf of partitions:** **a. The identity of the partition** **b. The configuration data of the**	**FIA_USB.1.1** The TSF shall associate the following user security attributes with subjects acting on the behalf of that user: [assignment: *list of user security attributes*].

partition c. **The privilege vector of the partition**	
FIA_USB.1.2 The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users: **Every subject acting on behalf of a partition will be assigned the security attributes associated with the partition on whose behalf the subject will act.**	**FIA_USB.1.2** The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users: [assignment: *rules for the initial association of attributes*].
FIA_USB.1.3 The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users: **subjects acting on behalf of partitions can not add additional security attributes beyond those initially assigned.**	**FIA_USB.1.3** The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users: [assignment: *rules for the changing of attributes*].
FMT_MSA.1.1 The TSF shall enforce the **Partition Management Access Control Policy** to restrict the ability to **query and modify** the security attributes **partition configuration data** to **authorized administrators that have the authority for those operations assigned in the AzMan policy for Hyper-V**.	**FMT_MSA.1.1** The TSF shall enforce the [assignment: *access control SFP(s), information flow control SFP(s)*] to restrict the ability to [selection: *change_default, query, modify, delete, [assignment: other operations]*] the security attributes [assignment: *list of security attributes*] to [assignment: *the authorised identified roles*].
FMT_MSA.2.1 The TSF shall ensure that only secure values are accepted for **partition configuration data**.	**FMT_MSA.2.1** The TSF shall ensure that only secure values are accepted for [assignment: *list of security attributes*].
FMT_MSA.3.1 The TSF shall enforce the **Partition Management Access Control Policy** to provide	**FMT_MSA.3.1** The TSF shall enforce the [assignment: *access control SFP, information flow control SFP*] to

restrictive default values for security attributes that are used to enforce the SFP.	provide [selection, choose one of: *restrictive, permissive, [assignment: other property]*] default values for security attributes that are used to enforce the SFP.
FMT_MSA.3.2 The TSF shall allow the **administrator authorized by the Hyper-V AzMan policy to modify partition configuration data** to specify alternative initial values to override the default values when an object or information is created except for those security attributes that nobody is allowed to modify.	**FMT_MSA.3.2** The TSF shall allow the [assignment: *the authorised identified roles*] to specify alternative initial values to override the default values when an object or information is created.
FMT_MTD.1.1 The TSF shall restrict the ability to **change_default and modify** the **partition's access to hypervisor calls related to inter-partition communication and the creation of guest partitions** to **nobody**.	**FMT_MTD.1.1** The TSF shall restrict the ability to [selection: *change_default, query, modify, delete, clear, [assignment: other operations]*] the [assignment: *list of TSF data*] to [assignment: *the authorised identified roles*].
FMT_MTD.1.1 The TSF shall restrict the ability to **change_default and modify** the **resource quota assigned to partition** to **administrators authorized by the Hyper-V AzMan policy to perform those actions**.	**FMT_MTD.1.1** The TSF shall restrict the ability to [selection: *change_default, query, modify, delete, clear, [assignment: other operations]*] the [assignment: *list of TSF data*] to [assignment: *the authorised identified roles*].
FMT_MTD.1.1 The TSF shall restrict the ability to **modify** the **Hyper-V AzMan policy** to **administrators authorized to access the Hyper-V authorization store**.	**FMT_MTD.1.1** The TSF shall restrict the ability to [selection: *change_default, query, modify, delete, clear, [assignment: other operations]*] the [assignment: *list of TSF data*] to [assignment: *the authorised identified roles*].
FMT_REV.1.1 The TSF shall restrict the ability to revoke **the assignment of devices** associated with the **partitions** under the control of the TSF to	**FMT_REV.1.1** The TSF shall restrict the ability to revoke [assignment: *list of security attributes*] associated with the [selection: *users, subjects, objects, [assignment: other additional*

administrators.	*resources]*] under the control of the TSF to [assignment: *the authorised identified roles*].
FMT_REV.1.2 The TSF shall enforce the rules **that the administrator needs to be authorized by the Hyper-V AzMan policy to perform those actions**.	**FMT_REV.1.2** The TSF shall enforce the rules [assignment: *specification of revocation rules*].
FMT_SMF.1.1 The TSF shall be capable of performing the following management functions: • **Management of partition configuration data** • **Management of virtual switches** • **Defining, deleting, starting and stopping partitions** • **Management of the Hyper-V AzMan policy.**	**FMT_SMF.1.1** The TSF shall be capable of performing the following management functions: [assignment: *list of management functions to be provided by the TSF*].
FRU_RSA.1.1 The TSF shall enforce maximum quotas of the following resources: **maximum CPU time per virtual CPU, maximum amount of the partition (guest) physical memory** that **partitions** can use **over a specified period of time**.	**FRU_RSA.1.1** The TSF shall enforce maximum quotas of the following resources: [assignment: *controlled resources*] that [selection: *individual user, defined group of users, subjects*] can use [selection: *simultaneously, over a specified period of time*].
FAU_GEN.1.1 The TSF shall be able to generate an audit record of the following auditable events: c) Start-up and shutdown of the audit functions; d) All auditable events for the **not specified** level of audit; and **e) The following Server Core 2008 specific events:**	**FAU_GEN.1.1** The TSF shall be able to generate an audit record of the following auditable events: 3. Start-up and shutdown of the audit functions; 4. All auditable events for the [selection, choose one of: *minimum, basic, detailed, not specified*] level of audit; and [assignment: *other specifically defined*

• **All authentication attempts of authorized administrators** • **Modification of the authentication information of authorized administrators** • **Modification to the security attributes of authorized administrators** • **Access to objects protected by the partition management access control policy**	*auditable events*].
FAU_GEN.1.2 The TSF shall record within each audit record at least the following information: a) Date and time of the event, type of event, subject identity (if applicable), and the outcome (success or failure) of the event; and b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST: **iii. The name of the object (if the audit event is for an operation on an object)** **iv. For changes to TSF data (except for authentication data): the new and the old values of the data**	**FAU_GEN.1.2** The TSF shall record within each audit record at least the following information: a) Date and time of the event, type of event, subject identity (if applicable), and the outcome (success or failure) of the event; and b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [assignment: other audit relevant information].
FAU_GEN.2.1 For audit events resulting from actions of identified users, the TSF shall be able to associate each auditable event with the identity of the user that caused the event.	**FAU_GEN.2.1** For audit events resulting from actions of identified users, the TSF shall be able to associate each auditable event with the identity of the user that caused the event.
FAU_SAR.1.1 The TSF shall provide **authorized administrators** with the capability to read **all audit information** from the audit records.	**FAU_SAR.1.1** The TSF shall provide [assignment: *authorised users*] with the capability to read [assignment: *list of audit information*] from the audit records.
FAU_SAR.1.2 The TSF shall provide the audit records in a	**FAU_SAR.1.2** The TSF shall provide the audit records in a

manner suitable for the **authorized administrator** to interpret the information **using a tool to access the audit records**.	manner suitable for the user to interpret the information.
FAU_SAR.2.1 The TSF shall prohibit all users read access to the audit records, except those users that have been granted explicit read-access.	**FAU_SAR.2.1** The TSF shall prohibit all users read access to the audit records, except those users that have been granted explicit read-access.
FAU_STG.1.1 The TSF shall protect the stored audit records in the audit trail from unauthorised deletion.	**FAU_STG.1.1** The TSF shall protect the stored audit records in the audit trail from unauthorised deletion.
FAU_STG.1.2 The TSF shall be able to **prevent** unauthorised modifications to the stored audit records in the audit trail.	**FAU_STG.1.2** The TSF shall be able to [selection, choose one of: *prevent, detect*] unauthorised modifications to the stored audit records in the audit trail.
FAU_STG.3.1 The TSF shall **generate an alarm to the authorized administrator** if the audit trail exceeds **the authorized administrator specified log size**.	**FAU_STG.3.1** The TSF shall [assignment: *actions to be taken in case of possible audit storage failure*] if the audit trail exceeds [assignment: *pre-defined limit*].
FAU_STG.4.1 The TSF shall **provide the authorized administrator the capability to prevent audited events, except those taken by the authorized administrator (in the context of performing TOE maintenance) and generate an alarm to the authorized administrator** if the audit trail is full.	**FAU_STG.4.1** The TSF shall [selection, choose one of: *"ignore audited events", "prevent audited events, except those taken by the authorised user with special rights", "overwrite the oldest stored audit records"*] and [assignment: *other actions to be taken in case of audit storage failure*] if the audit trail is full.
FIA_AFL.1.1 The TSF shall detect when **an authorized administrator configurable positive integer of consecutive** unsuccessful authentication attempts occur related to **any authorized administrator authentication process**.	**FIA_AFL.1.1** The TSF shall detect when [selection: *[assignment: positive integer number], an administrator configurable positive integer within[assignment: range of acceptable*

	values]] unsuccessful authentication attempts occur related to [assignment: *list of authentication events*].
FIA_AFL.1.2 When the defined number of **consecutive** unsuccessful authentication attempts has been **met or surpassed**, the TSF shall **disable the account for an authorized administrator configurable time period**.	**FIA_AFL.1.2** When the defined number of unsuccessful authentication attempts has been [selection: *met, surpassed*], the TSF shall [assignment: *list of actions*].
FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to individual **authorized administrators**: a) **user identifier** b) **group memberships** c) **authentication data** d) **security relevant roles** **e) allowable time and day of logon.**	**FIA_ATD.1.1** The TSF shall maintain the following list of security attributes belonging to individual users: [assignment: *list of security attributes*].
FIA_SOS.1.1 The TSF shall provide a mechanism to verify that secrets **used to authenticate an authorized administrator** meet **the property that a random attempt to guess the secret will succeed with a probability that is less than one in 2 x 10^{15}**.	**FIA_SOS.1.1** The TSF shall provide a mechanism to verify that secrets meet [assignment: *a defined quality metric*].
FIA_UAU.1.1 The TSF shall allow **read access to public objects** on behalf of the **authorized administrator** to be performed before the **authorized administrator** is authenticated.	**FIA_UAU.1.1** The TSF shall allow [assignment: *list of TSF mediated actions*] on behalf of the user to be performed before the user is authenticated.
FIA_UAU.1.2 The TSF shall require each **authorized administrator** to be successfully authenticated	**FIA_UAU.1.2** The TSF shall require each user to be successfully authenticated before allowing any

before allowing any other TSF-mediated actions on behalf of that user.	other TSF-mediated actions on behalf of that user.
FIA_UAU.7.1 The TSF shall provide only **obscured feedback** to the **authorized administrator** while the authentication is in progress.	**FIA_UAU.7.1** The TSF shall provide only [assignment: *list of feedback*] to the user while the authentication is in progress.
FIA_UID.1.1 The TSF shall allow **read access to public objects** on behalf of the **authorized administrator** to be performed before the **authorized administrator** is identified.	**FIA_UID.1.1** The TSF shall allow [assignment: *list of TSF-mediated actions*] on behalf of the user to be performed before the user is identified.
FIA_UID.1.2 The TSF shall require each **authorized administrator** to be successfully identified before allowing any other TSF-mediated actions on behalf of that **authorized administrator**.	**FIA_UID.1.2** The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.
FIA_USB.1.1 The TSF shall associate the following **authorized administrator** security attributes with subjects acting on the behalf of that **authorized administrator**: **a) The user identity which is associated with auditable events.** **b) The user identity or identities which are used to enforce the Partition Management Access Control Policy.** **c) The roles used to enforce the Partition Management Access Control Policy.**	**FIA_USB.1.1** The TSF shall associate the following user security attributes with subjects acting on the behalf of that user: [assignment: *list of user security attributes*].
FIA_USB.1.2 The TSF shall enforce the following rules on	**FIA_USB.1.2** The TSF shall enforce the following rules on

the initial association of user security attributes with subjects acting on the behalf of users: b) **Every subject acting on behalf of an authorized administrator will be assigned a subset of the security attributes associated with the authorized administrator on whose behalf the subject will act.**	the initial association of user security attributes with subjects acting on the behalf of users: [assignment: *rules for the initial association of attributes*].
FIA_USB.1.3 The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users: **subjects acting on behalf of authorized administrators can not add additional security attributes beyond those initially assigned.**	**FIA_USB.1.3** The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users: [assignment: *rules for the changing of attributes*].
FMT_MTD.1.1 The TSF shall restrict the ability to **manage** the **values of security attributes associated with user authentication data** to **authorized administrators**.	**FMT_MTD.1.1** The TSF shall restrict the ability to [selection: *change_default, query, modify, delete, clear, [assignment: other operations]*] the [assignment: *list of TSF data*] to [assignment: *the authorised identified roles*].
FMT_MTD.1.1 The TSF shall restrict the ability to **manage** the **security-relevant TSF data except for audit records, user security attributes, authentication data** to **the authorized administrator**.	**FMT_MTD.1.1** The TSF shall restrict the ability to [selection: *change_default, query, modify, delete, clear, [assignment: other operations]*] the [assignment: *list of TSF data*] to [assignment: *the authorised identified roles*].
FMT_MTD.1.1 The TSF shall restrict the ability to **query, delete, and clear** the **audit trail** to **the authorized administrator**.	**FMT_MTD.1.1** The TSF shall restrict the ability to [selection: *change_default, query, modify, delete, clear, [assignment: other operations]*] the [assignment: *list of TSF data*] to [assignment:

	the authorised identified roles].
FMT_MTD.1.1 The TSF shall restrict the ability to **initialize** the **user security attributes, other than authentication data** to **the authorized administrator**.	**FMT_MTD.1.1** The TSF shall restrict the ability to [selection: *change_default, query, modify, delete, clear, [assignment: other operations]*] the [assignment: *list of TSF data*] to [assignment: *the authorised identified roles*].
FMT_MTD.1.1 The TSF shall restrict the ability to **modify** the **authentication data** to **the authorized administrator and users authorized to modify their own authentication data**.	**FMT_MTD.1.1** The TSF shall restrict the ability to [selection: *change_default, query, modify, delete, clear, [assignment: other operations]*] the [assignment: *list of TSF data*] to [assignment: *the authorised identified roles*].
FMT_SMF.1.1 The TSF shall be capable of performing the following management functions: • **Management of audit data** • **Management of access rights for the Partition Management Access Control Policy** • **Definition of tasks as groups of controlled operations** • **Assignment of tasks and/or controlled operations to roles** • **Assignment of roles to users** • **Management of user security attributes of authorized administrators** • **Management of authentication data of authorized administrators.**	**FMT_SMF.1.1** The TSF shall be capable of performing the following management functions: [assignment: *list of management functions to be provided by the TSF*].

FMT_SMR.1.1 The TSF shall maintain the roles **authorized administrator, nobody for administrative users**.	**FMT_SMR.1.1** The TSF shall maintain the roles [assignment: *the authorised identified roles*].
FMT_SMR.1.2 The TSF shall be able to associate **administrative** users with roles **except for the role “nobody” which can not be assigned to any user**.	**FMT_SMR.1.2** The TSF shall be able to associate users with roles.
FPT_STM.1.1 The TSF shall be able to provide reliable time stamps.	**FPT_STM.1.1** The TSF shall be able to provide reliable time stamps.

Table 14: Mapping SFR components of this ST to the component definition in part 2

This mapping shows that all the operations have been performed correctly and marked accordingly.

6.4 TOE Summary Specification Rationale

6.4.1 Security functions justification

The TOE summary specification contains for each section that describes security functionality a part that maps the security functionality to the security functional requirements. This provides the link between the security functional requirements and the security functionality described in the TOE summary specification.

6.4.2 Assurance measures justification

The assurance measures are those defined by the EAL 4 assurance level, augmented by ALC_FLR.3. The selection of the EAL 4 assurance level is consistent with the assumed attack potential of the threat agents and is an assurance level used in the evaluation of products of the same kind.

7 TOE Summary Specification

This section describes the architecture and the security functions of Hyper-V. It starts with a description of architecture and the components followed by a description of the security functions.

7.1 Architecture of the TOE

The TOE consists of the following components:

- A small hypervisor responsible for the separation of the guest partitions and the management of virtual storage and virtual processors.
- A root partition (running the Server 2008 Server Core with some Hyper-V specific additions) that performs management activities and device virtualization
- Optional TOE software loaded into individual partitions (Virtualization Service Clients, VMBus driver and Enlightenments). This software is executed within a partition and is therefore not part of the TSF.

In addition the TOE requires server hardware that satisfies the hardware requirements for Hyper-V as laid out in chapter 1.

The following figure shows the general architecture of the TOE:

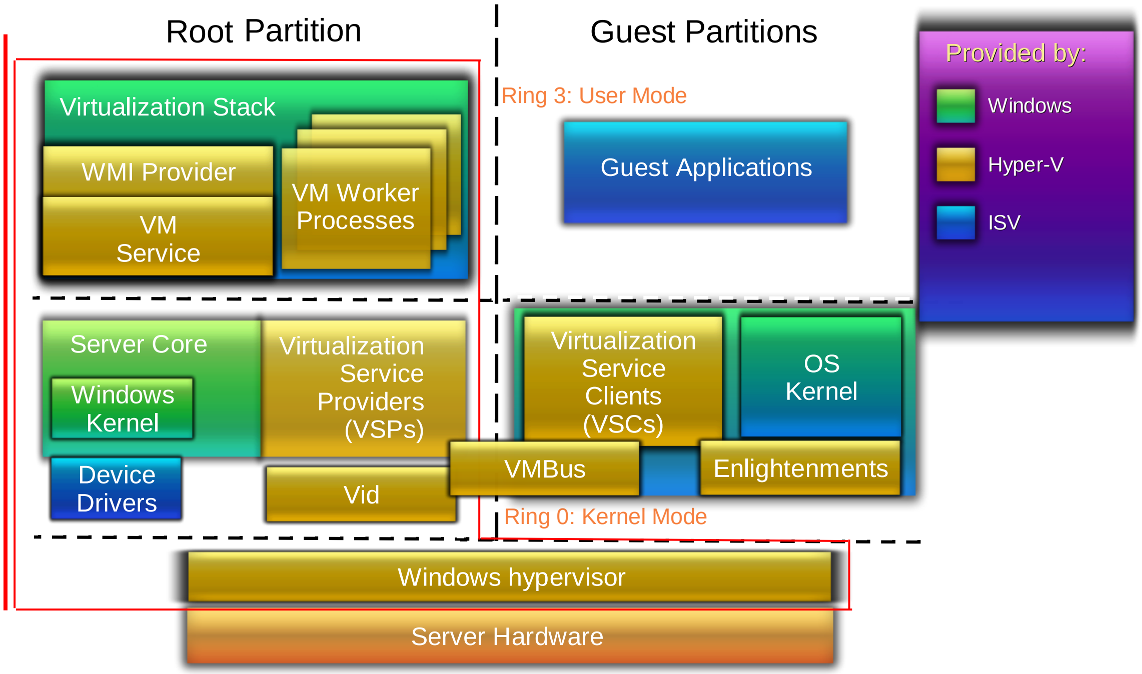

Figure 1: Hyper-V architecture components

The figure shows the architectural components of the TOE with the red line indicating the TSF boundary. For simplicity the figure shows only one guest partition, but if course Hyper-V supports multiple concurrent guest partitions. The example guest partition shown in the figure is for a partition that is “hypervisor aware” and has installed “Virtualizations Service Clients” (VSCs) and “Enlightenments”.

The colors in the figure indicate Hyper-V specific software (gold), Server Core 2008 software (green), and (potentially) third party software (blue). Note that no part of the TSF contains third party software.

The following section explains the purpose of the individual components shown in the figure:

7.1.1 Windows Hypervisor

By design, the hypervisor is a small and relatively simple piece of code. It runs on x64 systems and supports both 32-bit and 64-bit guests

The hypervisor runs in its own context and runs at a privilege level higher than that of guests. From this perspective, it makes sense to view the hypervisor as running at “ring
-1” (where rings 0 through 3 are used for guests).

The hypervisor is designed with concurrency in mind, and it is a design goal to provide scalability to large numbers of processors.

In an effort to minimize complexity, the hypervisor is not preemptable. External interrupts and inter-processor interrupts are disabled while code is running within the hypervisor. (SMIs and NMIs will remain enabled.)

The hypervisor is logically divided into two distinct strata. The lower layer is a simple microkernel that supports memory allocation, threads, signaling, and a hardware abstraction mechanism. The second layer runs on top of the microkernel and provides virtualization services (partitions, virtual processors, address translation, hypercalls, etc.). This layer is necessarily platform-specific because it virtualizes the underlying processor and interrupt controller hardware.

Partitions

A *partition* is the basic unit of isolation supported by the hypervisor. A partition is made up of a physical address space and one or more virtual processors. Furthermore, a partition can be assigned specific hardware resources (memory, devices and CPU cycles) and certain permissions and access rights. Each partition has a unique partition identity (unique per re-boot of the hypervisor, not globally unique), which is represented as a 64-bit number.

Partitions also have a set of attributes assigned to them by the hypervsisor. The security relevant attributes of a partition are:

- Amount of per virtual processor CPU time reserved for the partition
- Maximum amount of per virtual processor CPU time the partition is allowed to use

- Ability to create partitions (always disallowed for guest partitions in the evaluated configuration. This default value can not be changed)
- Access to memory pool related hypervisor calls
- Ability to initiate inter-process communication with other guest partitions (always disallowed for guest partitions in the evaluated configuration. This default value can not be changed)
- Access to debugging related hypervisor calls
- Access to CPU power management related hypervisor calls

Address Translation

The hypervisor provides a physical memory virtualization facility. This allows each partition to have a zero-based contiguous physical address space. Virtual processors support all x86 paging features and memory access modes (including large pages, global pages, write protection control, PAE, four-level page tables, paging disabled, etc.).

The hypervisor implements this behavior by means of two levels of address translation. The first level is controlled by the guest, allowing it to define a virtual address (VA) to guest physical address (GPA) translation. This is done via standard page tables maintained by the guest.

Second-level address translation is provided by the hypervisor without knowledge of the guest. This allows the hypervisor to virtualize physical memory, mapping guest physical addresses (GPAs) to system physical addresses (SPAs). The layout of a guest's physical address space is defined at partition creation time and can be modified at runtime.

Virtual Processors

Each partition has one or more *virtual processors* associated with it. A virtual processor is a virtualized instance of an x86 processor complete with user-level and system-level registers.

A virtual processor is scheduled on a physical processor (more specifically, on a hardware thread or core). The hypervisor does not use hard affinities, so a virtual processor may move from one physical processor to another. The hypervisor schedules virtual processors according to specified scheduling policies and constraints.

Note that while virtual processors are similar to threads within a traditional kernel, they differ from threads in the sense that they don't have an assigned address space. Rather, the guest operating system is able to switch the address space of a virtual processor by modifying its page table base (i.e. reload CR3 on x86 processors).

The hypervisor virtualizes all modes of an x86 processor including real mode, 16-bit and 32-bit protected mode, v86, long mode, etc. If virtualization support is missing for one or more processor modes (e.g. real mode), the hypervisor will provide the necessary emulation layer.

The hypervisor does not support dynamic addition and removal of logical processors. All potential logical processors must be declared at boot time.

Invoking the hypervisor

If guest code is actively executing, there are three ways code can enter the hypervisor:

1. **Interrupts**: Asynchronous events including external interrupts, IPIs, NMIs, SMIs, SIPIs, etc. (see section below on Interrupt Routing and Delivery)

2. **Intercepts:** Synchronous events the hypervisor has requested to intercept within the guest

3. **Hypercalls:** A programmatic interface to the hypervisor (analogous to "kernel calls")

Interrupt Routing and Delivery

The hypervisor is responsible for routing interrupts to individual guests. All real hardware interrupts are delivered to the hypervisor first (regardless of whether interrupts are currently masked by the guest). The hypervisor can then decide whether to handle the interrupt itself or reflect it to a partition.

Interrupts are used to signal a variety of asynchronous events from assigned hardware devices, VSPs, TSPs, other virtual processors, etc. While some interrupts are targeted at specific virtual processors (e.g. the virtual equivalent of IPIs), most interrupts are targeted at a partition. The hypervisor is responsible for routing the interrupt to an appropriate virtual processor.

From the perspective of a virtual processor, all interrupts delivered by the hypervisor are dispatched in the same way as traditional interrupts. On x86 processors, the behavior is defined by the guest's interrupt descriptor table (IDT). Most interrupts delivered by the hypervisor are maskable, so the guest can choose to hold off interrupts.

Intercept Handling, Redirection and Reflection

When the hypervisor is invoked due to some action performed by a guest (e.g. an access to an unmapped page, execution of a HLT instruction, etc.), the hypervisor can choose to do one of three things:

1. **Silently handle the intercept and return to the guest.** For example, the hypervisor may receive a page fault intercept because the page table entry in its shadow page table has not yet been populated despite the fact that the guest has already mapped the page in its page tables. In this case, the hypervisor would fill in the corresponding shadow page table entry and return back to the guest in such a way that the guest was unaware that the intercept occurred.

2. **Redirect the intercept to the partition related worker process in the root partition.** The worker process is a piece of code running in the root partition that is notified when an intercept occurs. This allows, for example, emulation of legacy devices. The hypervisor defines the events it would like to intercept. Examples on the x86 include: I/O port accesses, MSR accesses, CPUID, HLT, specific faults and exceptions, accesses to specified guest physical page ranges, and triple faults. Some intercepts (e. g. those related to address translation) are handled directly by the hypervisor. All other intercepts are forwarded to the worker processes to behandled there.

3. **Reflect the intercept back to the guest.** The hypervisor can simulate an exception within the guest. For example, if the hypervisor receives a page fault intercept and determines that the accessed page is unmapped within the guest's page tables, it can reflect the page fault back to the guest. The hypervisor also reflects the response it receives back from the worker process on intercepts

Device Ownership and Assignment

The hypervisor effectively "owns" the following hardware facilities: CPUs, memory, and APICs. All other hardware facilities and devices are assigned to the root partition.

Memory Intercepts

Intercepts can be requested for specific GPAs e. g. to simulate memory mapped I/O device register. The hypervisor supports read, write and execute intercepts. When a memory intercept is detected by the hypervisor, it sends an intercept message to the worker process related to the partition. The worker process typically responds by removing the intercept (e.g. paging in memory to back the GPA) and allowing the intercepted instruction to be re-executed. Alternatively, the external monitor can choose to complete the intercepted instruction in software. This is required for certain emulation scenarios (e.g. MMIO, VGA frame buffer accesses, etc.).

7.1.2 VMBus

VMBus is a component that allows inter-partition communication between a guest partition and the root partition. It is implemented as a ring buffer shared between the guest and root partition but the VMBus protocol also allows defining a broader range of shared memory between the two partitions used to transfer large amount of data especially in response to I/O request from a guest partition. VMBus is implemented by a driver in both the root and the guest partition.

The VMBus driver in a guest partition is optional and requires the operating system to be "hypervisor aware" and have such a driver available. Since such a VMBus driver in a guest partition operates as a part of the host operating system in the guest partition, it is considered to be outside of the TSF.

Communication via VMBus is used to access "synthetic devices" for

- Storage
- Networking
- Video
- USB

Synthetic devices are implemented using specific communication protocols over VMBus. Those are:

- SCSI and iSCSI (RFC 3720)
- RNDIS (Remote Network Driver Interface Specification)

- RDP (Remote Desktop Protocol)

The “Virtual Service Provider” in the root partition intercepts the protocols and the commands sent via those protocols by a guest partition, interprets them and passes them to specific handler within the root partition to perform the real I/O operations.

7.1.3 Operating System Enlightenments (outside of the TSF)

An operating system can detect that it operates on Hyper-V using the CPUID processor instruction. This instruction will be intercepted by the Hypervisor and returns a value indicating that the operating system is operating on a specific version of Hyper-V. The operating system can then use this information to install “Virtual Service Clients” that use the VMBus interface to communicate with the root partition for I/O operations and can install “Enlightenments” that make use of hypervisor calls to optimize the operation of the operating system on top of Hyper-V (e. g. memory management).

Enlightenments operate as part of the guest operating system and are considered to be outside of the TSF.

7.1.4 Virtual Service Clients (outside of the TSF)

Virtual Service Clients are an optional part of a guest partition and provide an interface between the guest operating system’s device access functions and the VMBus driver. Requests within the guest partition to access a virtual device can be directed to a virtual service client for the device, which then sends the request to the root partition via the VMBus communication interface. The response received from the root partition is then passed back to the virtual service client, which passes this response (eventually with some transformation) back to the requestor.

Note that virtual service clients operate within a guest partition as part of the guest partition’s operating system and are therefore considered to be outside of the TSF.

7.1.5 Virtualization Service Provider

The virtualization service provider is part of the root partition and takes requests for virtual device access passed by a guest partition via VMBus (using the protocols supported by VMBus) and redirects those requests to the functions that interpret the commands and translate them into the I/O operations on the real devices attached to the TOE. The following figure shows the flow of an example for a request to a virtual hard disk.

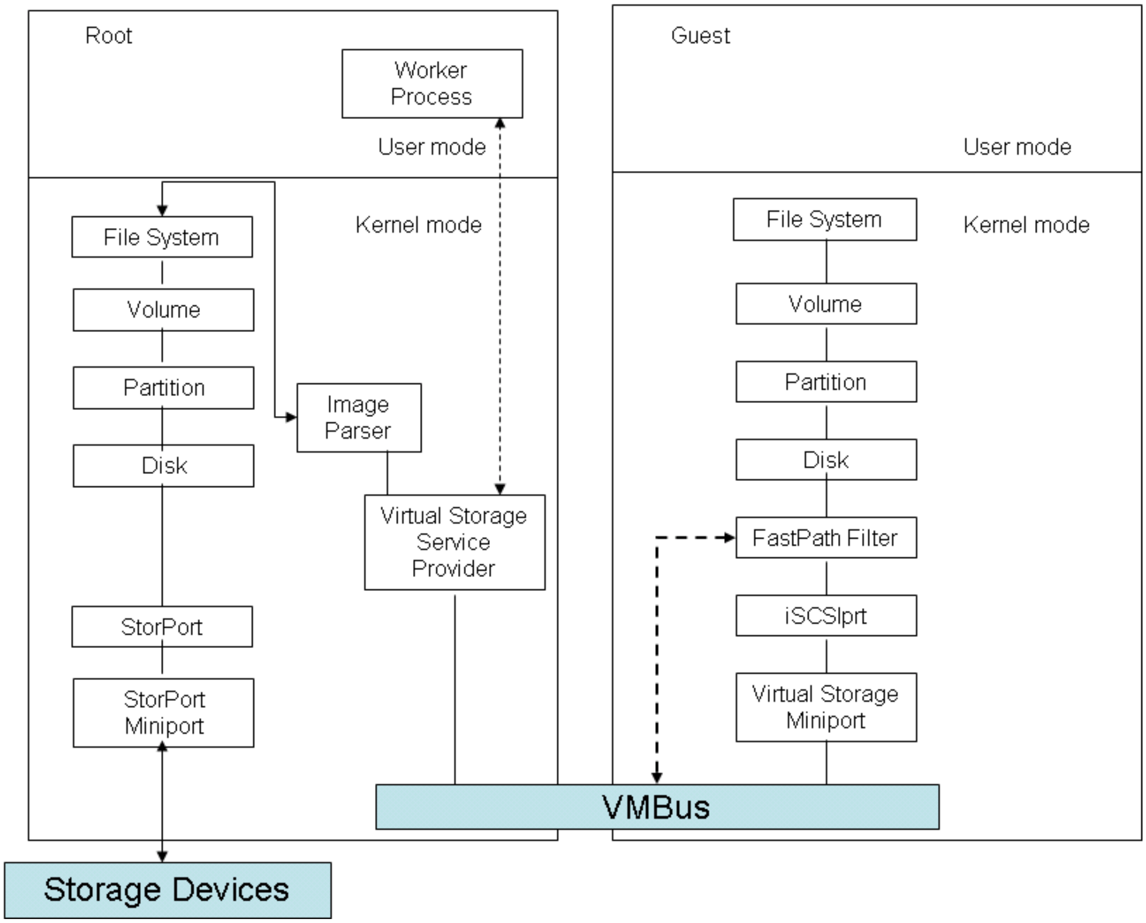

Figure 2: Example of a flow for access to a virtual hard disk (VHD)

In this example the request of an application within a guest partition is translated by the operating system within that guest partition into a request sent via VMBus to the root partition. Within the root partition this request is directed to the Virtual Storage Server which is the entry point of the Virtualization Service Provider for storage devices. Within the VSP and the Image Parser this request is translated to a request for access to a file that represents the virtual hard disk on the real storage media. In the case of a read request the data is read from this file and transferred back via the Virtual Storage Server and the VMBus to the guest partition, where the operating system passes this data back to the application.

7.1.6 VM Worker Processes

The VM worker processes are used for the virtualization of emulated devices. There is one separate VM worker process for each guest partition and all those VM worker processes operate as regular “user” processes on top of the Server 2008 operating system in the root partition. Each VM worker process operates under a different SID, which represents the guest partition within the root partition.

When the hypervisor intercepts an instruction related to device emulation (like access to an I/O port or access to memory mapped I/O), it signals this instruction and the guest partition identity (and virtual processor) to the root partition. The root partition signals this to the worker process that corresponds to the guest partition. The worker process performs the emulation and sends the response back to the Server 2008 kernel of the root partition (using specific ioctl interfaces), which in turn signals this response back to the hypervisor. The hypervisor takes the appropriate action to signal the result of the emulated instruction back to the guest partition.

A virtualized S3 Trio Video Card and a virtualized Intel 440 BX chipset are always available to a partition, providing the minimal virtualized hardware features that allow a partition to boot. The Intel 440 BX chipset provides the programming interfaces to the PCI bus, IDE controller, DMA controller, Interrupt controller, UART, floppy disk controller, keyboard, mouse, timer, real time clock, and power management. Note that even if the programming interfaces for all those devices exist, it depends on the configuration of the partition if all those programming interfaces are backed by virtualized devices (very much like in a real machine).

The VM worker process provides also a virtualized BIOS that is mapped into the guest partition.

7.1.7 WMI Provider

Windows Management Instrumentation (WMI) is the infrastructure for management data and operations on Windows-based operating systems. The WMI provider for virtualization and the hypervisor API enable developers, to quickly build custom tools, utilities, and enhancements for the virtualization platform. The WMI interfaces can manage defined aspects of the virtualization services.

Hyper-V exposes a rich interface which permits the administrator to monitor and control the partition environment. Most of Hyper-V can be controlled via the WMI provider, which allows easy, yet powerful customization of the partitions. For further details on what can be controlled with the WMI provider, see the following topics:

- Virtual System Management Service
- Networking
- Resource Management

Virtual System Management Service

The Virtual System Profile describes the objects which make up a partition. These objects include the base system, the devices that make up the system, the settings for the system and its devices, and the management service that performs operations on the system.

The physical computer and its hosted partitions are each represented by the ComputerSystem class. The ComputerSystem instance representing the physical computer is associated, via the HostedDependency association, to the ComputerSystem instances representing the partitions on the physical computer.

Each partition is associated with a Virtual System Global Setting Data (VSGSD) instances and one or more Virtual System Setting Data (VSSD) instances. For each VSSD there is a series of Resource Allocation Setting Data (RASD) objects. The RASD objects describe the settings for each device in a partition. Together, the VSGSD, VSSDs and RASDs describe the configuration of the partition.

The VSGSD represents the global settings for a partition. These settings are independent of those described in the VSSD, and thus do not change if a snapshot is applied to the partition. The VSSD is used to describe the virtualization-specific settings of a partition. An instance of the VSSD may describe either the current settings for the partition, or the settings of the partition at the time of a snapshot.

None of the objects in the model support direct creation, modification or deletion through intrinsic Put or Delete operations. Instead, the Virtual System Management Service (VSMS) contains methods to manipulate the objects.

Networking

The networking profile describes the objects used for configuring the system to allow partitions to communicate over the network. The global networking objects, used to configure the network switch in the root partition, include the Msvm_VirtualSwitchManagementService, Msvm_VirtualSwitch, and Msvm_SwitchPort classes. The partition networking objects, used to configure the network interface card (NIC) in the partition, include the Msvm_EmulatedEthernetPort, Msvm_ResourceAllocationSettingData, Msvm_VmLANEndpoint and Msvm_SwitchLANEndpoint classes.

The root of the global networking profile is the Msvm_VirtualSwitch class. This class represents a virtual switch device in the root partition. Msvm_VirtualSwitch is associated with instances of the Msvm_SwitchPort class, which represents the ports on the virtual switch. Instances of the Msvm_VirtualSwitch and Msvm_SwitchPort classes are created, deleted, and connected via the Msvm_VirtualSwitchManagementService class (not shown in the diagram above).

Virtual Switch Management Service (VSMS) represents the networking service present on a single Hyper-V host and contains methods for Msvm_VirtualSwitchManagementService used to control the definition, modification, and destruction of global networking resources such as virtual switches, switch ports and internal Ethernet ports.

The representation of the ethernet NIC device in the partition looks very similar to that of any other device, as described in the Virtual System Management Service. The Msvm_EmulatedEthernetPort and Msvm_SyntheticEthernetPort classes represent the virtual NIC device, and are configured via an associated Resource Allocation Setting Data (RASD) instance. The only unusual characteristic of this representation is that, when the partition is instantiated and in turn creates the Msvm_EmulatedEthernetPort and Msvm_SyntheticEthernetPort devices, it also creates an associated Msvm_VmLANEndpoint instance for the virtual NIC. Similarly, when the partition is saved or turned off and the Msvm_EmulatedEthernetPort and Msvm_SyntheticEthernetPort instances are destroyed, the associated Msvm_VmLANEndpoint instance is also destroyed. The purpose of the Msvm_VmLANEndpoint is to serve as a bridge for connecting two networking ports to each other. In this case, it is used to connect a virtual NIC to a port on the virtual switch device. In other words, it connects the Msvm_EmulatedEthernetPort and Msvm_SyntheticEthernetPort instances on the partition to a particular Msvm_SwitchPort instance on the virtual switch. To connect a switch to the outside, you must bind the physical ethernet port to the Msvm_VirtualSwitch through BindExternalEthernetPort. Adversely, when connecting a switch to the host networking stack, or internal NIC, use ConnectInternal to have a partition talk to the host and not the outside world. Msvm_ActiveConnection connects a switch port to the Msvm_SwitchLanEndpoint to which the port is connected inside of Hyper-V. The existence of this object means that the switch port and the Msvm_SwitchLanEndpoint are actively connected and the ethernet port associated with Msvm_VMLanEndPoint can communicate with the network through the switch port.

Resource Management

The Resource Virtualization Profile provides the means by which a client can discover the virtual resources supported by the virtualization system. It also describes the capacity – or number of allocations – that is supported for each type of virtual resource.

Two different classes of virtual resources are defined by the Resource Virtualization Profile:

- Shared Resource: Represents the resources managed by Hyper-V that are, or are capable of being shared among multiple partitions. Msvm_Processor is an example of a shared resource.
- Synthetic Resource: Represents the virtual resources that have no corresponding real resource. Msvm_SyntheticEthernetPort is an example of a synthetic resource.

The resource pool is used to collect a class of Hyper-V managed resources so that it can be easily discovered while its capabilities and settings can be described in a central location.

From the resource pool, one can access the associated Allocation Capabilities (AC). This class describes the capabilities of the resource described by this resource pool. For instance, it may indicate whether the Msvm_EmulatedEthernetPort represented by this resource pool supports virtual LANs (VLANs) or filters.

The AC Profile defines the means by which one can discover the valid range of and default settings for a given virtual resource. An AC object is associated with each resource pool. Four Resource Allocation Setting Data (RASD) objects are associated with the AC object to describe the minimum, maximum, default and incremental values for the given resource's allocation. Together, these classes describe the overall range of supported capabilities. The Msvm_AllocationCapabilities instance provides an anchor point for the set of Msvm_ResourceAllocationSettingData instances that specify the default and valid range of settings for a virtual resource. The Msvm_SettingsDefineCapabilities association class provides the link between the AC instance and the minimum, maximum, incremental and default settings for a resource supported by the virtualization platform.

7.1.8 AzMan as the framework for access control to management operations of Hyper-V

Authorization Manager (AzMan) is a set of COM-based runtime interfaces that allow applications to easily manage and verify a user's request to access application defined resources (usually operations defined by an application).

AzMan allows to group operations to "tasks", define roles as a collection of permissions to tasks and/or operations, and assign roles to users. In addition one can define "BizRules", which are scripts that are evaluated when the access check is performed. Those allow evaluating specific conditions at runtime like the time of day. BizRules can be associated with tasks.

AzMan uses an "authorization store" to store the policy including the tasks, roles and assignments of roles to users. The authorization store in the evaluated configuration is a XML file stored on the local disk. Other types of authorization stores (e. g. within Active Directory) are not part of the evaluated configuration.

AzMan allows auditing modifications to the policy store as well as all access checks performed.

AzMan also allows defining a "scope" as a collection of resources to which a policy applies.

7.2 Security functionality provided by the Hyper-V specific components of the TOE

This section presents the security functionality of Hyper-V specific components and the mapping to the SFRs as defined the previous chapter of this Security Target.

In general Hyper-V implements the following security functionality (using the headings from part 2 of the CC):

- Generation of Hyper-V specific audit records
- User and TSF data protection - protection of partition configuration data, partition resources and devices allocated to a partition
- Identification of partitions
- Management of partition configuration
- Resource utilization quota management and enforcement

7.2.1 Audit functionality

The hypervisor layer as well as the Hyper-V specific components in the root partition are able to generate auditable events. Those events are stored and managed by the Server 2008 audit function. Security relevant events generated by the hypervisor are signaled to the root partition and the root partition generates and records the audit record for those events. The security relevant events signaled by the hypervisor are:

- The creation of a partition
- The deletion of a partition
- A failure condition detected internal to the hypervisor component

In addition to those events the Hyper-V specific components within the root partition are capable to generate audit records for the following events:

- Modifications to the Hyper-V AzMan policy
- Access checks performed by AzMan

Each audit records is sent to the Event Logger in the Server 2008 Server Core in the root partition, which inserts the time and data and stores the record in the event log.

Mapping to SFRs

This implements the security functional requirements FAU_GEN.1 (H) and FAU_GEN.2 (H).

7.2.2 User and TSF data protection functionality

The Hyper-V user data protection function performs

- Discretionary access control of administrators to administration objects
- Discretionary access control of partitions to virtualized and synthetic devices
- Residual data protection

Hyper-V related administrative objects are protected using the Server 2008 data protection functions and not functions of the Hyper-V specific parts.

Mapping to SFRs

This implements the security functional requirements FDP_ACC.1 (H-PM), FDP_ACC.1 (H_DA), FDP_ACF.1 (H_PM), FDP_ACF.1 (H_DA) and FDP_RIP.1 (H).

7.2.3 Identification functionality

The TOE requires each partition to be identified and this is ensured by the TSF which assigns an identifier to a partition. This identifier is used to identify the partition when it traps into the hypervisor and when a communication channel to the root partition (via VMBus) is established. Since guest partitions are created and fully controlled by the TSF, authentication of the partitions is not necessary. When a partition is started, the privileges and the virtualized devices are assigned to the partition in accordance with the partition’s configuration data. Some privileges like the privilege to create partitions or the privilege to create ports that might be used for direct communication between guest partitions are not assigned to any guest partition and this default value can not be changed.

Mapping to SFRs

This implements the security functional requirements FIA_ATD.1 (H), FIA_UID.2 (H) and FIA_USB.1 (H)

7.2.4 Security Management Functionality

Hyper-V can be managed either directly using the management functions within the root partition, or remotely using a client that connects to the WMI and performs the management activities using this interface. In both cases the administrative user that wants to perform management activities is authenticated by the Server 2008 authentication function and gets his Server 2008 managed privileges assigned. Those define the management activities he is allowed to perform.

Hyper-V specific configuration data is stored in objects that are protected by the standard Server 2008 access control functions. Access to those objects is therefore managed by the access control management functions of Server 2008.

Hyper-V uses the "Authorization Manager" (AzMan) to define a role-based access control model for managing Hyper-V. The root partition protects a set of Hyper-V management operations. A default scope is provided as well as the set of operations that can be controlled. Tasks and roles can be defined by an application that uses the AzMan API for the management of the policy store. The command-line scripts HVRemote.wsf and HVRoles.wsf can be used to manage the authorization store of the AzMan policy for Hyper-V locally in the root partition.

The authorization store for Hyper-V can be found in the file "C:\ProgramData\Microsoft\Windows\Hyper-V\Initialstore.xml" in the root partition. This file needs to be selected when defining or editing role definitions, defining or editing task definitions, or defining or editing scopes. Roles can than be assigned to users, defining the management activities they are allowed to perform.

The authorization policy for Hyper-V has a set of operations which can be used to group to tasks and/or be assigned to roles. Those are:

- Operations on Hyper-V services
  - Read Service Configuration
  - Reconfigure Service
  - View Virtual Switch Management Service
- Operations on Virtual Machines
  - Create Virtual Machine
  - Delete Virtual Machine
  - Change Virtual Machine Authorization Scope
  - Start Virtual Machine

  - Stop Virtual Machine
  - Pause and Restart Virtual Machine
  - Reconfigure Virtual Machine
  - View Virtual Machine Configuration
  - Allow Input to Virtual Machine
  - Allow Output from Virtual Machine
- Operations on Virtual Switches
  - Create Virtual Switch
  - Delete Virtual Switch
  - Modify Switch Settings
  - View Switches
  - Create Virtual Switch Port
  - Delete Virtual Switch Port
  - Connect Virtual Switch Port
  - Disconnect Virtual Switch Port
  - Modify Switch Port Settings
  - View Switch Ports
  - View LAN Endpoints
  - Change VLAN Configuration Port
  - View VLAN Settings
  - Create Internal Ethernet Port
  - Delete Internal Ethernet Port
  - View Internal Ethernet Port
  - Bind External Ethernet Port
  - Unbind External Ethernet Port

- View External Ethernet Ports

Whenever a user attempts to perform an administrative operation on Hyper-V, the AzMan interfaces for checking access are called to validate the user's rights to perform the management function.

Mapping to SFRs

This implements the security functional requirements FMT_MSA.1 (H), FMT_MSA.2 (H), FMT_MSA.3 (H-PM), FMT_MSA.3 (H-DA), FMT_MTD.1 (H-1), FMT_MTD.1 (H-2), FMT_MTD.1 (H-3), FMT_REV.1 (H), FMT_SMF.1 (H)

7.2.5 TSF Protection Functionality

Hyper-V protects its TSF code and data by maintaining separate address spaces for those that are protected from access or unmediated interference by untrusted subjects or subjects outside of the TOE that may communicate with the TOE via its network interfaces. So the main TSF protection functionality of the TOE is the maintenance of this separate address spaces and the mediation of all references to TSF data and protected user data.

Mapping to SFRs

TSF Protection Functionality is implemented by the TOE architecture.

7.2.6 Resource utilization functionality

Hyper-V provides functions that allow an authorized administrator to define the maximum amount of memory and CPU time a guest partition is allowed to use. Hyper-V controls the use of those resources and ensures that a given guest partition does not use more of those resources than allocated to the partition. This prohibits that a single guest partition consumes all resources of a specific type and thereby cause a denial of service for other guest partitions.

Mapping to SFRs

This implements the security functional requirement FRU_RSA.1 (H)

7.3 Security functionality provided by Server 2008

As described above the root partition consists of a Server Core installation of Server 2008 with the Hyper-V specific components described above as being part of the root partition. In addition to those, the TOE also relies on non Hyper-V specific components within the root partition. The security functions provided by those non Hyper-V specific components are:

- Identification and authentication of administrative users and users that want to connect to a partition
- Management of the security attributes of administrative users
- Access control policy for Server 2008 files and objects

- Management of the Server 2008 access control policy
- Generation of audit events specific for Server 2008
- Management and protection of the audit trail
- Review of audit records
- Management of date and time

The TOE requires each administrative user to be identified and authenticated prior to performing TSF-mediated functions on behalf of that user. Administrative users are identified and authenticated by the root partition using the regular identification and authentication function implemented in Server 2008.

7.3.1 Identification and authentication of administrative users and users that want to connect to a partition

The TOE requires each user to be identified and authenticated prior to performing TSF-mediated functions on behalf of that user, with a few exceptions, regardless of whether the user is logging on interactively or is accessing the system via a network connection.

For initial interactive logon, a user must invoke a trusted path in order to ensure the protection of identification and authentication information. The trusted path is invoked by using the Ctrl+Alt+Del key sequence, which is always captured by the TSF (i.e., it cannot be intercepted by an untrusted process), and the result will be a logon dialog that is under the control of the TSF. Once the logon dialog is displayed, the user can enter their identity (username and domain) and authentication (password).

User's can change their password either during the initial interactive log or while logged on. To change a user's password, the user must invoke the trusted path by using the Ctrl+Alt+Del key sequence. The logon dialog displayed allows the user to select an option to change their password. If selected, a change password dialog is displayed which requires the user to enter their current password and a new password.

In case of a domain user, the TOE will submit the user ID and the user's authentication credentials to the Active Directory service and receives a token from this service indicating if the user could be successfully authenticated.

The TSF will change the password only if the TSF can successfully authenticate the user using the current password that is entered (see section Logon Process for a description of the authentication process).

Mapping to SFRs

This implements the security functional requirements FIA_UAU.1, FIA_UAU.7, FIA_UID.1, and FIA_USB.1.

7.3.2 Management of the security attributes of administrative users

The Server 2008 Server Core in the root partition maintains databases (collectively referred to as user attribute database) that fully define user and group accounts. These definitions include:

- Account name – used to represent the account in human-readable form;
- SID – a User Identifier (UID) or group identifier used to represent the user or group account within the TOE;
- Password (only for user accounts) – used to authenticate a user account when it logs on (stored in hashed form and is encrypted when not in use using a Rivest's Cipher (RC)4 algorithm and a RC4 system generated key). Note that passwords of domain users are managed in an Active Directory server;
- Groups – used to associate group memberships with the account; Note that groups of domain users are managed by an Active Directory server and used as read-only data on the root partition;
- Logon rights – used to control the logon methods available to the account (e.g. the "logon locally" right allows a user to interactively logon to a given system); Note that for domain users the logon rights parts of the domain security policy is managed by an Active Directory server and used as read-only data on the root partition.
- Miscellaneous control information – used to keep track of additional security relevant account attributes such as allowable periods of usage, whether the account has been locked, whether the password has expired, password history, and time since the password was last changed; Note that for domain users this type of information is managed in an Active Directory server and used as read-only data on the root partition.
- Other non-security relevant information – used to complete the definition with other useful information such a user's real name and the purpose of the account or to support functions not used in the evaluated configuration. Note that for domain users this type of information is managed in an Active Directory server and used as read-only data on the root partition.

The TSF provides a set of functions that allow the account policy to be managed. These functions include the ability to define account policy parameters, including minimum password length. The minimum password length can be configured to require as large as 14 characters. However, the administrator guide recommends that the minimum password length be configured to no less than eight (8) characters (with at least 90 available characters, the password space is 4,304,672,100,000,000 available combinations). Therefore, in the evaluated configuration, the probability that a random attempt will succeed is less than one (1) in $5x10^{15}$ and the probability that, for multiple attempts within one minute, the probability that a random attempt will succeed is less than one (1) in $25x10^{12}$.

During authentication, the TSF will not provide feedback that will reduce the probability before the metrics identified above. Furthermore, the TSF forces a delay between attempts, such that there can be no more than ten (10) attempts per minute.

For each subsequent failed logon following five (5) consecutive failed logon occurrences in the last 60 seconds, function sleeps for 30 seconds before showing a new logon dialog. It therefore supports the

I&A function that no more than ten (10) interactive logon attempts are possible in any 60 second (one minute) period.

Mapping to SFRs

This implements the security functional requirements FIA_AFL.1, FIA_ATD.1, FIA_SOS.1, FMT_MTD.1 (1), and FMT_MTD.1 (5).

7.3.3 Access control policy for Server 2008 files and objects

The Server 2008 Server Core instance in the root partition mediates access between subjects and user data objects, also known as named objects. Subjects consist of processes with one or more threads running on behalf of users, which in the case of the root partition are either the administrative users or the worker processes (which have their own Security Identifier).

Tokens contain the security attributes for a subject. Tokens are associated with processes and threads running on behalf of the user. The DAC related information in the token includes: the SID for the user, SIDs representing groups for which the user is a member, privileges assigned to the user, an owner SID identifying SID to assign as owner for newly created objects, a default DACL (for newly created objects), token type (primary or impersonation), impersonation level (for impersonation tokens), an optional list of restricting SIDs, and a logon ID for the session.

Every object has a unique Security Descriptor (SD) that includes an ACL. SDs contain all of the security attributes associated with an object. All objects covered by the access control policy have an associated SD. The security attributes from a SD used for access control are the object owner SID, the DACL present flag, and the DACL itself, if present.

DACLs contain a list of Access Control Entries (ACEs). Each ACE specifies an ACE type, a SID representing a user or group, and an access mask containing a set of access rights. Each ACE has inheritance attributes associated with it that specify if the ACE applies to the associated object only, to its children objects only, or to both its children objects and the associated object.

Mapping to SFRs

This implements parts of the security functional requirements FDP_ACC.1 (H-PM) and FDP_ACF.1 (H-PM).

7.3.4 Management of the Server 2008 access control policy

The notion of role within the TOE is generally realized by assigning group accounts and privileges to a given user account. Whenever that user account is used to logon, the user will be assuming the role that corresponds with the combination of groups and privileges that it holds. While additional roles could be defined, this ST defines just one logical role: the authorized administrator role.

The Administrator role is defined as any user account that is assigned one of the security-relevant privileges for managing the configuration of the partitions or to manage users and groups.

Mapping to SFRs

This implements the security functional requirements FMT_MTD.1 (2), FMT_MTD.1 (4), and FMT_SMR.1

7.3.5 Generation of audit events specific for Server 2008

The Event logger service of Server 2008 creates the security event log, which contains the security relevant audit records collected on a system. For each audit event, the Event Logger stores the following data in each audit record:

Date: The date the event occurred.

Time: The time the event occurred.

User: The security identifier (SID) of the user on whose behalf the event occurred that represents the user. SIDs are described in more detail in Section 7.3.1 under Identification and Authentication,

Event ID: A unique number identifying the particular event class.

Types: Indicates whether the security audit event recorded is the result of a successful or failed attempt to perform the action.

The Server 2008 Server Core in the root partition maintains an audit policy in its database that determines which categories of events are actually collected. Defining and modifying the audit policy is restricted to the authorized administrator.

Mapping to SFRs

This implements the security functional requirements FAU_GEN.1 and FAU_GEN.2.

7.3.6 Management and protection of the audit trail

The Event Logger controls and protects the security event log. To view the contents of the security log, the user must be an authorized administrator. The security event log is a system resource, created during system startup. No interfaces exist to create, destroy, or modify a security event within the security event log.

The TSF protects against the loss of events through a combination of controls associated with audit queuing and event logging. As configured in the TOE, audit data is appended to the audit log until it is full. The TOE protects against lost audit data by allowing the authorized administrator to configure the system to generate an audit event when the security log reaches a specified capacity percentage (e.g., 90%).

Additionally, the authorized administrator can configure the system not to overwrite events and to shutdown when the security log is full. When so configured, after the system has shutdown due to audit overflow, only the authorized administrator can log on. When the security log is full, a message is written to the terminal display of the authorized administrator indicating the audit log has overflowed.

Audit events may be lost if the audit event queues reach their high-water mark, or if the security log file is full. The TOE can be configured to crash when the audit trail is full. The security log file is limited in size by the resources available on the system.

Mapping to SFRs

This implements the security functional requirements FAU_STG.1, FAU_STG.3, and FAU_STG.4

7.3.7 Review of audit records

The event viewer administrator tool provides a user interface to view, sort, and search the security log. The security log can be sorted and searched by user identity, event type, date, time, source, category, event ID, and computer. The security log can also be searched by free form texts occurring in the audit records.

Reading and evaluation of the audit trail is restricted to administrators authorized to access the audit trail. They have tools available that allow them to transform the audit records to human readable format and further evaluate the records.

Mapping to SFRs

This implements the security functional requirements FAU_SAR.1, FAU_SAR.2 and FMT_MSA.1 (3).

7.3.8 Management of date and time

Each hardware platform supported by the TOE includes a real-time clock. The real-time clock is a device that is assigned to the root partition and can only be accessed using functions provided by the TSF. Specifically, the TSF provides functions that allow users, including the TSF itself, to query and set the clock, as well as functions to synchronize clocks within a domain. The ability to query the clock is unrestricted, while the ability to set the clock requires a privilege dedicated to that purpose. This privilege is only granted to authorized administrators to protect the integrity of the time service.

Mapping to SFRs

This implements the security functional requirements FPT_STM.1