User-defined Module

**SYS.2: Desktop Systems**

# SYS.2.8: Qubes OS Clients

## 1 Description

### 1.1 Introduction

Qubes OS is a free and open-source security-oriented operating system meant for single-user desktop computing. Qubes OS leverages Xen-based virtualization to allow for the creation and management of isolated virtual machines (VMs), which have specific :

- **Purposes :** with a predefined set of one or many isolated applications, for personal or professional projects, to manage the network stack, the firewall, or to fulfill other user-defined purposes.
- **Natures :** full-fledged or stripped-down virtual machines which are based on popular operating systems such as Fedora, Debian or Windows.
- **Levels of trust** : from complete to non-existent. All windows are displayed in a unified desktop environment with unforgeable colored window borders so different security levels are easily identifiable.

In Qubes, all programs are run in lightweight Virtual Machines called qubes which are isolated from each other (“security by compartmentalization”). Not every application runs in its own qube. Instead, each qube represents a security domain. A special administration domain called Dom0 is used to manage, i.e. create, start, stop, and delete the VMs defined in a Qubes installation.

By default all application VMs (AppVMs) are based on a single, common TemplateVM , although more TemplateVMs can be created and used. Each AppVM shares the root file system with its respective TemplateVM. An AppVM has read-only access to the file system of the Template on which it is based, so an AppVM cannot modify a TemplateVM in any way. This is important, as it means that if an AppVM is ever compromised, the TemplateVM on which it’s based (and any other AppVMs based on that TemplateVM) will still be safe.

Only the private files, usually located under a folder like `/home` or `Documents`, are stored in the AppVM itself and thus are permanent, surviving shutdown and reboot. So creating a large number of domains is cheap: each one needs only as much disk space as is necessary to store its private files. For operations concerning potentially malicious data, a special type of AppVM called Disposable VM can be used which has no permanent private storage and thus is completely destroyed on shutdown, reducing the risk of infecting the system with malware.

Common attack vectors such as network cards and USB controllers are isolated in their own hardware qubes, while their functionality is maintained through secure networking, firewalls and USB device management. Integrated file and clipboard copy and paste operations make it easy to work with different qubes without compromising security.

### 1.2 Objective

The objective of this module is to protect information created, processed, stored or sent on Qubes OS clients. The requirements of the module mainly address Linux and Windows clients running as virtual machines under the control of the Xen hypervisor and the Qubes OS operating environment.

### 1.3 Not in Scope

The module SYS.2.8 Clients under Qubes OS is to be used for all client systems where Qubes OS is used as central system.

This module includes basic requirements for operating clients under control of Qubes OS. It specifies and adds specifics of Qubes OS systems to the aspects addressed in module SYS.2.1 *General Client*. The specifics of the operating systems run in the individual VMs are addressed in the respective modules for these systems, e.g. SYS.2.2.3 *Windows 10 Clients* or SYS.2.3 *Unix Clients*.

The module does not include software that builds on the configurations of the operating systems run in the individual qubes, such as e-mail clients or Office software; the requirements in this regard can be found in layer APP.1 *Client Applications* of the IT-Grundschutz compendium.

## 2 Threat Landscape

The following specific threats and vulnerabilities are of particular importance for module SYS.2.8 *Qubes OS Clients*:

### 2.1 Malware

Malware is usually activated in secret without the knowledge or permission of the user. In Qubes OS, malware is restricted to the qube where the infecting file was executed; if this is an AppVM. it cannot modify the TemplateVM on which this AppVM is based. So, after a restart of the infected AppVM, malware which changed system data is eliminated while malware stored in AppVM private data will survive an AppVM restart.

### 2.2 Untrusted or Faulty Software from Third-Party Sources

In Qubes, it is easily possible to download and compile additional software independently from the software packages provided by the system. If ready-made software packages are used, these often are not always installed from the existing package sources of the operating system underlying a specific VM. They can also be procured from third-party sources without any further examination. Each of these alternative means of software installation entails additional risks because incorrect or incompatible software and malware may be installed. If this software is installed in a TemplateVM, immediately all AppVMs based on this TemplateVM are compromised.

### 2.3 Device Based Attacks

Attaching a PCI device to a qube has serious security implications. It exposes the device driver running in the qube to an external device. In many cases a malicious device can choose what driver will be loaded (for example by manipulating device metadata like vendor and product identifiers) – even if the intended driver is sufficiently secure, the device may try to attack a different, less secure driver. Furthermore that VM has full control of the device and may be able to exploit bugs or malicious implementation of the hardware, as well as plain security problems the hardware may pose. Conversely, it is also possible that in this way the security of the host system is compromised by targeted interference via this device.

The connection of an untrusted USB device to the Xen management VM Dom0 is a security risk since the device can attack an arbitrary USB driver, exploit bugs during partition-table-parsing or simply pretend to be a keyboard. The whole USB stack is put to work to parse the data presented by the USB device in order to determine if it is a USB mass storage device, to read its configuration, etc. Manipulation of the USB stack can therefore also be used to attack the integrity of Dom0.

### 2.4 Incomplete Separation of Windows 10 VMs

In order to have a clear separation between TemplateVMs containing the software and AppVMs holding the user data, user data need to be stored on a device different from the system device. For Windows 7, this is achieved by moving the directory `C:\Users` to `D:\Users` during template configuration. For Windows 10, this is not possible due to undocumented features of the NTFS file system, and usually results in a more or less defective system.

### 2.5 Multibooting Qubes

Dual booting Qubes OS poses special risks to the security of the system. One problem is that when you dual or multiboot, even if you are using encryption on your Qubes OS installation, `/boot` is normally unencrypted and thus still unprotected and could be maliciously modified by the other OS, possibly leading to Qubes itself being maliciously modified. The other problem is firmware security – for example the other system could infect the BIOS firmware, which might enable compromise or spying on the Qubes system.

## 3 Requirements

The specific requirements of module SYS.2.8 *Qubes OS Clients* are listed below. As a matter of principle, the IT Operation Department is responsible for fulfilling the requirements. The Chief Information Security Officer (CISO) must always be involved in strategic decisions. Furthermore, the CISO is responsible for ensuring that all requirements are met and verified according to the security concept agreed upon. There can be additional roles with further responsibilities for the implementation of requirements. They are listed explicitly in square brackets in the header of the respective requirements.

| | |
|---|---|
| Module Owner | IT Operation Department |
| Further Roles | User |

### 3.1 Basic Requirements

For module SYS.2.8 *Qubes OS Clients*, the following requirements MUST be implemented as a matter of priority

#### SYS.2.8.A1 Secure Installation (B)

The hardware on which Qubes OS is to run MUST support IOMMU-based virtualization. This MUST be enabled in the BIOS of the IT system along with the standard virtualization (Intel VT-x) and AMD virtualization (AMD-V) extensions.

The digital signature of the downloaded ISO MUST be verified before installation.

No other operating system SHOULD be installed on the drive where Qubes OS is installed.

#### SYS.2.8.A2 Authentication of Administrators and Users [User] (B)

In order to use the client, the users MUST be authenticated by the IT system.

#### SYS.2.8.A3 No Additional Software in Dom0 (B)

Before installing any additional software in Dom0, a thorough analysis of the possible consequences for the security of Dom0 MUST be performed. Additional software MUST ONLY be installed in Dom0 if it is absolutely necessary and there is no safe alternative.

#### SYS.2.8.A4 No Applications Run on a TemplateVM [User] (B)

With the exception of a text editor used to modify configuration files, one MUST NOT run user applications like mail or office software in either TemplateVMs or in Dom0. Only software installation and update is acceptable as usage of a TemplateVM. All applications MUST be run on AppVMs.

#### SYS.2.8.A5 No Network Access from TemplateVMs (B)

TemplateVMs MUST NOT have access to network VMs (e.g. sys-firewall) by default. Instead, the dedicated NetVM MUST be specified as “`none`”. Any software installation and update SHOULD only be done from

the dedicated update host, preferably via the update widget of the TemplateVM. Software not available from the update server, like special device drivers, MUST be installed in a TemplateVM only after a thorough analysis of potential consequences its security.

### SYS.2.8.A6 Installing Updates and Patches (B)

Updates and patches to Dom0 MUST be installed as quickly as possible whenever the Qubes OS Updater shows that such patches are available. Patches to TemplateVMs SHOULD be installed when it is convenient to do so. It may be advisable to first clone a TemplateVM and install risky patches in this clone to test them there.

### SYS.2.8.A7 Firewall Configuration (B)

The firewall VMs for AppVMs MUST be configured as restrictive as possible. The addresses to which an AppVM can connect MUST be restricted as much as possible. If additional rules for incoming connections are defined, they MUST also be limited to necessary cases. For AppVMs that should not be given network access, the network VM MUST be specified as "`none`".

## 3.2 Standard Requirements

For module SYS.2.8 *Qubes OS Clients*, the following requirements correspond to the state-of-the-art technology along with the basic requirements. They SHOULD be implemented as a matter of principle

### SYS.2.8.A8 Disk Encryption (S)

If physical security of the system running Qubes OS cannot be ensured all the time, disk encryption SHOULD be selected as an installation option.

### SYS.2.8.A9 Using a USB Qube (S)

If external USB devices are to be connected to the client system, a USB qube SHOULD be installed. However, this is only possible if the system does not use a USB keyboard and if it is not started from a USB drive, because in these cases the use of a USB Qube leads to a system that is no longer usable.

### SYS.2.8.A10 Restriction of Windows 10 VMs to the Local Network (S)

The firewall rules for Windows 10 VMs SHOULD be specified such that access is restricted to the local network and possibly selected external addresses in order to block any transmission of telemetry data.

### SYS.2.8.A11 Use of Disposable VMs [User] (S)

If any suspect data (such as mail attachments from unknown sources) have to be processed, this SHOULD be done using the functions provided for disposable VMs. For operations that do not need network access, the use of disposable VMs based on a TemplateVM without network access SHOULD be considered.

## 3.3 Requirements in Case of Increased Protection Needs

Generic suggestions for module SYS.2.8 *Qubes OS Clients* are listed below for requirements which go beyond the standard level of protection. These SHOULD be taken into account IN THE EVENT OF INCREASED PROTECTION NEEDS. Final specification is performed within a risk analysis.

### SYS.2.8.A12 Protection Against Unauthorized Logins (H)

The use of YubiKey SHOULD be considered to achieve enhanced login security.

### SYS.2.3.A13 Two-Factor Authentication via U2F Proxy (H)

The U2F ("Universal 2nd Factor") specification for hardware token authentication defines protocols for multiple layers from USB to the browser API, and the entire stack is designed for use with web applications (mostly websites) in browsers. The token-based two-factor authentication by installing the Qube U2F-Proxy SHOULD be used for sensitive web accesses.

### SYS.2.8.A14 Configuration of Windows 10 TemplateVM and AppVM (H)

Windows 10 Clients SHOULD always be installed as a combination of TemplateVM and dependent AppVMs and not as standalone VMs, as this is the only way to ensure a secure separation between system and user data. A connection to the open Internet necessary for the periodic check of the license activation should therefore only be allowed for the TemplateVM which does not contain any user data. Connections of

the AppVM to the open Internet SHOULD be blocked, especially if personal data are processed, whose possible transmission by telemetry is not permitted.

### SYS.2.8.A15 Anti Evil Maid Protection (H)

For mobile systems like notebooks, or if there is a risk that somebody may gain physical access to the computer when it is left powered down, or if Qubes is used in dual boot mode, then installation of Anti Evil Maid (or of other boot protections schemes like Heads) SHOULD be considered if the hardware preconditions are met. AEM will inform the user of any unauthorized modifications to the BIOS or boot partition.

### SYS.2.8.A16 Split GPG (H)

Split GPG implements a concept similar to having a smart card with your private GPG keys, except that the role of the "smart card" plays another Qubes AppVM. This way one, not-so-trusted domain can delegate all crypto operations, such as encryption/decryption and signing to another, more trusted, network-isolated, domain. This way the compromise of the domain where a client app is running does not allow the attacker to automatically also steal all the keys. If GNU Privacy Guard is used, Split GPG SHOULD be used.

### SYS.2.8.A17 Centralized Administration via SALT (H)

Qubes OS includes the Salt (also called SaltStack) management engine in dom0 as default (with some states already configured). Salt allows administrators to easily configure their systems. In a managed environment, Salt SHOULD be used to separate the roles of administrator and user, with VM management done by administrators without access to user data and VM use performed by users without access to VM management functions. Any Policies specified to control Salt SHOULD be as restrictive as possible.

### SYS.2.8.A18 Encrypted AppVMs (H)

For AppVMs containing sensitive information, the private volume SHOULD be encrypted. For Windows AppVMs, this could be done using BitLocker or tools like VeraCrypt to encrypt device `D:`, for Linux-based AppVMs, LUKS/dm-crypt encryption could be used for the directory `/rw` (including `/home` and `/usr/local`). This can also be used to restrict access to such AppVMs to selected users knowing the encryption passphrase or having access to a token containing the key.

As long as the system is not left unattended while an encrypted AppVM is running, there is no need to encrypt the corresponding TemplateVM, as any possible sensitive data copied form the AppVM to the TemplateVM are destroyed on AppVM shutdown.

If encrypted AppVMs are used, encryption of the system disk SHOULD be specified during Qubes OS installation, and an installation without swap file SHOULD be considered in order to avoid situations in which sensitive information could be left there.

# 4 Additional Information

For more information about threats and security safeguards for module SYS.2.8 *Qubes OS Clients*, see the following publications, among others, all last accessed on 04.10.2020:


[QubesDoc] Documentation of installation, configuration and use of Qubes OS
https://www.qubes-os.org/doc/,

[QubesInstall] Installation guide for Qubes OS
https://www.qubes-os.org/doc/installation-guide/

[QubesFirewall] Rules for configuring network interfaces between VMs
https://www.Qubes-os.org/doc/firewall/

[QubesWindows] Installation guide for Windows Clients in Qubes
https://www.Qubes-os.org/doc/windows-vm/
https://www.Qubes-os.org/doc/windows-tools/

[QubesDevelopment] Developer documentation, discussion of current / future enhancements

https://github.com/QubesOS/qubes-issues/issues

[QubesUsers] Google user group, discussion of usage problems
https://groups.google.com/g/qubes-users

[QubesForum] Community forum, discussion of usage problems and new developments
https://qubes-os.discourse.group/

# 5 Appendix: Cross-reference Table for Elementary Threats

The cross-reference table contains the assignment of Elementary Threats to the requirements. This table can be used to determine which Elementary Threats are covered by which requirements. The corresponding Elementary Threats are counteracted by implementing the safety measures derived from the requirements. The letters provided in the second column (C = confidentiality, I = integrity, A = availability) indicate the key security objectives which are primarily addressed by the requirement. The following Elementary Threats are relevant for module SYS.2.8 *Qubes OS Clients*:

- G 0.14 Interception of Information / Espionage
- G 0.19 Disclosure of Sensitive Information
- G 0.20 Information or Products from an Unreliable Source
- G 0.21 Manipulation with Hardware or Software
- G 0.22 Manipulation of Information
- G 0.23 Unauthorised Access to IT Systems
- G 0.25 Failure of Devices or Systems
- G 0.28 Software Vulnerabilities or Errors
- G 0.29 Violation of Laws or Regulations
- G 0.30 Unauthorised Use or Administration of Devices and Systems
- G 0.31 Incorrect Use or Administration of Devices and Systems
- G 0.32 Misuse of Authorisation
- G 0.38 Misuse of Personal Information
- G 0.39 Malware
- G0.46 Loss of integrity of sensitive Information

| Elementare Gefährdungen Anforderungen | CIA | G 0.14 | G 0.19 | G 0.20 | G 0.21 | G 0.22 | G 0.23 | G 0.25 | G 0.28 | G 0.29 | G 0.30 | G 0.31 | G 0.32 | G 0.38 | G 0.39 | G 0.46 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SYS.2.8.A1 | | | X | X | X | X | X | X | X | | | X | | | X | X |
| SYS.2.8.A2 | | X | X | | | | X | | | | X | | X | | | |
| SYS.2.8.A3 | | X | X | X | X | X | X | | X | | | X | | | X | X |
| SYS.2.8.A4 | | X | X | X | X | X | | | X | | | X | | | X | X |
| SYS.2.8.A5 | | X | X | | X | X | | | X | | | X | | | X | X |
| SYS.2.8.A6 | | | | X | | | | | X | | | X | | | X | X |
| SYS.2.8.A7 | | X | X | | | | | | X | | | X | | | X | |
| SYS.2.8.A8 | | X | | | | X | | | | X | | | | X | | X |
| SYS.2.8.A9 | | X | X | | X | | | X | | | | | | | X | X |
| SYS.2.8.A10 | | X | X | | | | | | | X | | | | | X | |
| SYS.2.8.A11 | | | X | X | | | | | | | | | | | X | X |
| SYS.2.8.A12 | CIA | X | X | | X | X | X | | | | X | | | | | X |
| SYS.2.8.A13 | CIA | X | | | | X | | | X | | | | X | X | | X |
| SYS.2.8.A14 | CI | X | X | | | | | | X | X | | X | | X | X | X |
| SYS.2.8.A15 | CIA | X | X | X | X | X | X | | | | | | | | X | X |
| SYS.2.8.A16 | CI | X | X | | | | | | X | | | | | | | |
| SYS.2.8.A17 | CIA | | X | | X | | X | | | | X | X | X | | | |
| SYS.2.8.A18 | CI | X | X | | | X | | | | | | | X | X | | X |