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 rsik 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 block 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 on commercially available IT systems. 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. If the client has interfaces for data exchange (e.g. CD/DVD, USB, Bluetooth or WLAN), the security specifications of module SYS.3.4 *Mobile Storage Media* must be fulfilled.

Within the framework of this client module, it is assumed that, in addition to the administrator, only one unchanging person makes constant active use of an interactive user account. Clients used by several persons consecutively or simultaneously require additional safeguards not addressed within the framework of this module.

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 developed with the objective of executing unwanted and usually damaging functions. Malware is usually activated in secret without the knowledge or permission of the user. These days, malware provides an attacker with extensive communication and control capabilities, as well as a number of functions. Amongst other things, malware may be used to obtain specific passwords, control systems remotely, disable protective software and obtain data without authorization. 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, most malware is eliminated.

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. On the other hand, if such software is installed in an AppVM, it is automatically removed on AppVM shutdown.

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.

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.

2.4 Evil Maid Attacks

An evil maid attack is an attack on an unattended device, in which an attacker with physical access alters it in some undetectable way so that they can later access the device, or the data on it. The name refers to the scenario where a maid could subvert a device left unattended in a hotel room – but the concept itself also applies to situations such as a device being intercepted while in transit, or taken away temporarily by airport or law enforcement personnel.

The attack begins when the victim leaves their device unattended. The attacker can then proceed to tamper with the system. If the victim's device does not have password protection or authentication, an intruder can turn on the computer and immediately access the victim's information. However, if the device is password protected, as with full disk encryption, the firmware of the device needs to be compromised, usually done with an external drive. The compromised firmware often provides the victim with a fake password prompt identical to the original. Once the password is input, the compromised firmware sends the password to the attacker and removes itself after a reboot. In order to successfully complete the attack, the attacker must return to the device once it has been unattended a second time to steal the now-accessible data.

2.5 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 not working anyway. Furthermore, such a configuration is no longer supported according to the Windows documentation.

2.6 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)

Qubes OS has very specific system requirements. Even on supported hardware, it MUST be ensured that IOMMU-based virtualization is activated in the BIOS. Without it, Qubes OS will not be able to enforce isolation. For Intel-based boards, this setting is called Intel Virtualization for Directed I/O (Intel VT-d) and for AMD-based boards, it is called AMD I/O Virtualization Technology (or simply AMD-Vi). This parameter MUST be activated in the computer's BIOS, alongside the standard Virtualization (Intel VT-x) and AMD Virtualization (AMD-V) extensions.

Qubes OS developers have absolutely no control over the servers providing the installation ISO, so it is possible that they might be compromised, or just be serving compromised ISOs because their operators decided so, for whatever reason. Therefore the digital signature on the downloaded ISO MUST always be verified prior to the installation.

Qubes cannot be installed as a virtual machine under another hypervisor like VirtualBox, VMware or Hyper-V, because the hypervisor Xen, on which Qubes is based, is of type 1 (hardware hypervisor) and does not support nested virtualization.

Qubes SHOULD be installed on a separate device, preferably an SSD with at least 40 GB free space, without any other operating system installed on this device. For testing purposes, the system may be installed on a fast USB 3.0 stick; using only USB 2.0 will result in an unacceptably slow system.

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)

Normally there should be few reasons for updating software in the administration domain Dom0. This is because there is no networking in Dom0, which means that even if some bugs will be discovered e.g. in the Dom0 Desktop Manager, this really is not a problem for Qubes, because all the third-party software running in Dom0 is not accessible from VMs or network in any way. Additional software MUST be installed in Dom0 only after a thorough analysis of potential consequences for the security of Dom0, as this has a high probability of compromising the security of the whole system by introducing errors or even malware in Dom0. Such installations in Dom0 MUST be a rare exception.

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, as this has a high probability of compromising the security of the the TemplateVM and all AppVMs depending on this TemplateVM.

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

The persons in charge MUST obtain information on vulnerabilities that have become known. 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; normally, there is no need to install them immediately when available. It may be advisable to first clone a TemplateVM and install risky patches in this clone to test them there. After shutdown of a patched TemplateVM, its patches are applied automatically to the dependent AppVMs whenever they are booted again.

SYS.2.8.A7 Firewall Configuration (B)

Every qube in Qubes is connected to the network via a FirewallVM, which is used to enforce network-level policies. By default there is one default FirewallVM, but the user is free to create more, if needed. The firewall for an AppVM is controlled by iptables rules restricting the addresses to which this AppVM can connect. These addresses SHOULD be selected as restrictive as possible in order to avoid data leakage from sensitive AppVMs; it may even be necessary to allow no network connections from some AppVMs.

Without explicit additional rules, no access from the outside or from other VMs is allowed. These rules SHOULD be specified as restrictive as possible, e.g. only allowing access to a local network and no access from other qubes or from outside the system to an AppVM. The AppVM cannot modified the rules to which it is subjected.

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, during installation or later on, a USB qube SHOULD be installed. A USB qube acts as a secure handler for potentially malicious USB devices, preventing them from coming into contact with dom0 (which could otherwise be fatal to the security of the whole system). It thereby mitigates some of the security implications of using USB devices.

Special consideration to this step has to be given if the system uses a USB keyboard, because no typing is possible if the keyboard is connected to a non-running USB qube.

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. This effectively blocks any transmission of telemetry data, since no VM can change the firewall rules specified for it. The firewall rules are enforced by the firewall VM selected for that VM (normally sys-firewall).

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)

You can use a YubiKey to enhance Qubes user authentication, for example to mitigate risk of someone snooping the password. This can also slightly improve security when you have a USB keyboard. The use of YubiKey SHOULD be considered to achieve enhanced login security.

SYS.2.3.A13 Two-Factor Browser Authentication (H)

Two-factor authentication SHOULD be used by installing the Qubes U2F Proxy. The U2F (“Universal 2nd Factor”) specification defines protocols for multiple layers from USB to the browser API, and the whole stack is intended to be used with web applications (most commonly websites) in browsers.

SYS.2.8.A14 Configuration of Incoming Network Access to VMs (H)

Normally any networking traffic between qubes is prohibited for security reasons. However, in special situations, one might want to selectively allow specific qubes to establish networking connectivity between each other. In order to allow networking between two qubes, appropriate iptables rules have to be specified in the firewall VM to which both qubes are connected. Additionally, in each qube an iptables rule allowing access from the other qube has to be specified. In order to make these rules permanent, they must be entered into the configuration files of all these qubes. These rules SHOULD be specified as restrictive as possible, avoiding the specification of access to whole subnets.

In order to allow a service present in a qube to be exposed to the outside world in the default setup (where the qube has sys-firewall as network VM, which in turn has sys-net as network VM) the following needs to be done:

- In the sys-net VM:
  - Route packets from the outside world to the sys-firewall VM
  - Allow packets through the sys-net VM firewall
- In the sys-firewall VM:
  - Route packets from the sys-net VM to the VM
  - Allow packets through the sys-firewall VM firewall
- In the qube:
  - Allow packets through the qube firewall to reach the service

SYS.2.8.A15 Configuration of Windows 10 TemplateVM and AppVM (H)

The individual user data directories `Documents` SHOULD be manually moved from drive `C:` to drive `D:` after TemplateVM installation. So any user data stored in these directories are kept in the AppVMs instead of in the TemplateVM.

Configuration data like those stored in directories like `AppData` still remain in the TemplateVM, such that their Changes are lost each time the AppVM shuts down. In order to make permanent changes to these configuration data, they have to be changed in the TemplateVM, meaning that applications have to be started there, which violates and perhaps even endangers the security of the TemplateVM. Such changes SHOULD be done only if absolutely necessary and with great care. It is a good idea to test them first in a cloned TemplateVM before applying them in the production VM.

File copy operations to a Windows 10 VM are possible, if the Qubes OS default user property is set to the user name used for access to that VM, which can be done in the administrative domain Dom0 via the command

`qvm-prefs` *`VMname`* `default_user` *`username`*

If this property is not set or set to a wrong value, files copied to this VM are stored in the folder `C:\Windows\System32\config\systemprofile\Documents\QubesIncoming\`*`srcVM`*
If the target VM is an AppVM, this has the consequence that the files are stored in the corresponding TemplateVM and so are lost on AppVM shutdown.

SYS.2.8.A16 Windows 10 Usage According to GDPR (H)

If Windows 10 is used to process personal data, no automatic data transfer to countries outside the EU is allowed without explicit consent of the person(s) concerned, or other legal consent, as applicable. Since no reliable way is found to completely control the sending of telemetry from Windows 10, the system containing personal data must be completely shielded from the internet.

This SHOULD be achieved by installing Windows 10 on a TemplateVM with the user data directory moved to a separate drive (usually D:). Personal data MUST NOT be stored within the TemplateVM, but only in AppVMs depending on this TemplateVM. Network access by these AppVMs MUST be restricted to the local network. Any data exchange of the AppVMs MUST be restricted to file and clipboard operations to and from other VMs in the same Qubes system.

SYS.2.8.A17 Anti Evil Maid Protection (H)
Anti Evil Maid is an implementation of a TPM-based dynamic trusted boot for dracut / initramfs-based OSes (Fedora, Qubes, etc.) with a primary goal to prevent Evil Maid attacks. In short, AEM relies on TPM and a feature found in Intel's vPro CPUs (TXT) to detect tampering of various boot components. 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.A18 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. For high security requirements, Split GPG SHOULD be used.

SYS.2.8.A19 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.

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 01.08.2020:

[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/

[QubesIssues] Developer documentation, discussion of current / future enhancements
https://github.com/QubesOS/qubes-issues/issues

[QubesUsers] User forum, discussion of usage problems
https://groups.google.com/forum/?nomobile=true#!forum/qubes-users

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
- G 0.46 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	
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				
SYS.2.8.A15	CI	X	X						X	X		X			X	X
SYS.2.8.A16	CI	X	X							X				X		
SYS.2.8.A17	CIA	X	X	X	X	X	X								X	X
SYS.2.8.A18	CI	X	X						X							
SYS.2.8.A19	CIA		X		X		X				X	X	X			