

**BAHIR DAR UNIVERSITY**

**OPERATING SYSTEMS AND SYSTEMS PROGRAMMING**

**INDIVIDUAL ASSIGNMENT**

**QUBES OS**

**Section B**

**Name ID**

**Biruk Yenebihon BDU 1601161**

**Submission date: 16/08/17 E.C**

**Submission to: Lec. Wendmu Baye**

**Contents**

**Title Page**

Introduction to Qubes OS ------------------------------------------- 3

Background ------------------------------------------------- 3

Motivation ------------------------------------------------- 3

Objectives of Qubes os ----------------------------------------- 4

Requirements of Qubes os ----------------------------------- 5

Hardware Requirement ------------------------------------ 5

Software Requirement ------------------------------------ 6

Installation Steps For Qubes OS ------------------------------ 7

Problem and solution --------------------------------------- 9

File system support -------------------------------------- 14

Advantage and Disadvantage of Qubes OS -------------- 15

Conclusion about Qubes OS ---------------------------------- 18

Future Out look and Recommandation of Qubes os ------- 18

Virtuallization ------------------------------------------------- 20

Reference ---------------------------------------------------------- 23

**QUBES OS**

INTRODUCTION TO QUBES OS

Written by Joanna Rutkowska, Qubes OS is based on the principle of "security through isolation," providing an open-source model, security-focused operating systems for users in an age where modern-day digital threats have advanced to such levels that traditional operating systems either do not or cannot provide required levels of security and isolation for sensitive data. First released in 2012, it was developed by Invisible Things Lab.

BACKGROUND

Normal operating systems, such as Windows, macOS, and mainstream Linux distributions, use a structure called monolithic design. This typically creates a policy where one application, like web browsers or PDF readers, can completely give an attacker access to the entire operating system. This is a problem and a major risk to users managing confidential information, like journalists, developers, activists, and people working in business.

Recognizing this vulnerability, the creators of Qubes OS envisioned developing an environment in which compromised application/system component damage could be contained, so that it wouldn't spread to the rest of the system. This is accomplished through lightweight virtual machines (VMs), or qubes, each of which has its level of trust and specific role (e.g., networking, personal files, web browsing).

The central idea of Qubes OS is to deliver a desktop environment that will very closely approximate that available on high-security desktop installations but without sacrificing usability. Compartmentalized VMs that behave like individual workspaces, each independent of others, are created using the Xen hypervisor for compartmentalizing purposes. Compartmentalization greatly restricts the effects of malware, exploits, or user error. For example, a compromised PDF reader in one qube cannot access files or resources in another.

Qubes OS actually adopts an approach to compute that hews toward being "reasonably secure." User breaches can and do happen, but minimizing trust and segmenting tasks reduces the negative consequences of those breaches. The concern for increased amounts of evidence of surveillance, data theft, and zero-day attacks makes Qubes OS a modern and pragmatic solution for users who want privacy combined with digital sovereignty.

MOTIVATION

The main purpose of Qubes OS is a desktop environment that delivers high security without usability compromises. It achieves this using the Xen hyervisor in generating compartmentalised VMs, which operate as separate workspaces, each isolated from all others. Compartmentalisation limits the effect of malware, exploits, or user error. For example, a compromised PDF reader within one qube cannot attain files or resources from another.

Qubes OS is about “reasonably secure” computing in assuming that breaches indeed happen without disposing trust and segmenting them into tasks to lessen the effects of such breaches. Against a backdrop of increasing surveillance, theft of data, and zero-day attacks, here is a realistic, futuristic weapon for today’s users, including those with strong digital sovereignty arguments.

**OBJECTIVES OF QUBES OS**

1. Security

Implementing security through isolation is Qubes OS's main goal. Qubes OS makes sure that a compromise in one area doesn't result in a complete system breach by executing distinct applications and system components in isolated qubes. Opening a malicious email attachment in one Qube, for instance, won't impact another Qube that houses credentials or personal files.

1. Reduce attack surface

Qubes OS seeks to lessen the operating system's attack surface. It divides networking, USB handling, and system administration into distinct, isolated virtual machines (VMs) rather than executing all services on a single kernel or environment. Because fewer parts are exposed to the external environment, it is more difficult for hackers to take advantage of the system.

1. Flexible security policies

Qubes allows users to implement fine-grained security policies that dictate how information is shared between qubes. This includes strict copy/paste and file-sharing policy between VMs, allowing users to exert total control over how information is shared through the system. This capability allows users to configure the system to suit their threat model and workflows.

1. Usability without compromising security

Another core objective is to make strong security feasible and accessible. While other security-focused solutions are difficult to deploy, Qubes OS offers an easy-to-use desktop environment, with the system's inherent complexity of VM management being taken care of to a large extent. This makes it accessible to both security professionals and privacy-conscious everyday users.

1. Transparent trust model

Qubes OS promotes an open trust model by enabling individuals to observe easily distinct qubes with color-coded windows. This visual reinforcement improves user awareness about where they are operating and what information or programs they are dealing with, thus precluding unintentional cross-contamination.

**Requirements of Qubes OS**

A.Hardware Requirements

Qubes OS is designed to run on **64-bit Intel or AMD hardware** with specific virtualization support. Not all systems are compatible, even if they appear to meet the basic specs. The developers maintain a **Hardware Compatibility List (HCL)** on the Qubes OS website, where users can confirm whether specific models work well with Qubes.

### **1. Processor Requirements**

**Architecture:** 64-bit x86\_64 CPU (required)

**Virtualization Support:**

**Intel:** Must support **VT-x** and **VT-d**

**AMD:** Must support **AMD-V** and **AMD-Vi (IOMMU)**

These features are necessary for efficient virtualization and for assigning physical devices (like USB controllers or network cards) to specific qubes securely.

### **2. Memory (RAM)**

**Minimum:** 8 GB

**Recommended:** 16 GB or more

Since each virtual machine (qube) runs its own OS and services, RAM usage can grow quickly. For users who plan to run many qubes simultaneously, more RAM is crucial.

### **3. Storage**

**Minimum:** 128 GB SSD or HDD

**Recommended:** 256 GB SSD or higher

Solid-state drives (SSDs) are highly recommended for performance reasons, especially considering that each qube creates its own virtual disk image

### **And also It is recommanded to use External storage devices external hard disk, SSD or USB devices because the operating system may case all the files on the computer to halt. So it is better to use the operating system external device**

### **4. Graphics and Display**

**Preferred:** Integrated Intel or AMD graphics

**Not Recommended:** Nvidia GPUs, due to poor driver support and complications with Xen.

**Hybrid Graphics (Intel + Nvidia):** Often problematic and not supported out of the box.

Qubes OS does not support 3D acceleration in dom0 (the admin domain), so users should not expect to run games or graphics-intensive applications.

1. **Software Requirments**

Qubes OS includes all necessary software during installation, but it does rely on certain technologies to operate properly.

### **1. Base Software Architecture**

**Hypervisor:** Xen

This is the core virtualization layer. It allows Qubes OS to run multiple virtual machines securely and efficiently.

**Dom0 (Domain 0):**

A minimal and highly trusted Fedora-based Linux VM that controls the Xen hypervisor and the GUI.

Users do not install applications in dom0; instead, it is strictly used for management.

**Template VMs:**

These are special VMs that serve as bases for creating application qubes. Qubes OS supports various templates like:

**Fedora Whonix** (for anonymity via Tor)

**Debian Arch Linux** (community-supported)

Software is installed in these templates and used in linked qubes.

**Installation Steps for Qubes OS**

Step 1: Check Hardware Compatibility

Before we began, ensure your hardware meets the minimum requirements for Qubes OS. This includes having a compatible CPU with virtualization support (Intel VT-x/AMD-V and IOMMU), at least 4 GB of RAM (8 GB or more recommended), and sufficient disk space.

Step 2: Download Qubes OS

1. Visit the Qubes OS official website (https://www.qubes-os.org/).

2. Go to the "Download" section and choose the latest stable release.

3. Download the ISO file to your computer.

Step 3: Create a Bootable USB Drive

We need to create a bootable USB drive with the downloaded ISO file. You can use tools like Rufus or balena etcher (for Windows) or dd (for Linux/macOS). In our case we try to use balena etcher.

1. Insert a USB drive

2. Open balena etcher.

3. Select your USB drive under "Device."

4. Choose the Qubes OS ISO file under "Boot selection."

5. Click "Start" and wait for the process to complete.

1. Open a terminal.

2. Identify your USB drive using lsblk or diskutil list.

3. Use the following command (replace /path/to/qubes.iso and /dev/sdX with your actual paths):

4. Wait for the command to finish and safely eject the USB drive.

▎Step 4: Boot from USB Drive

1. Insert the bootable USB drive into your computer.

2. Restart your computer and enter the BIOS/UEFI settings (usually by pressing F2, F12, ESC, or DEL during startup).

3. Change the boot order to prioritize booting from the USB drive.

4. Save changes and exit BIOS/UEFI.

▎Step 5: Start Qubes OS Installer

1. When your computer boots from the USB drive, you will see the Qubes OS installer menu.

2. Select "Install Qubes OS" and press Enter.

▎Step 6: Follow the Installation Wizard

1. Language Selection: Choose your preferred language and click "Continue."

2. Installation Destination: Select the disk where you want to install Qubes OS. Make sure to back up any important data on that disk, as it will be formatted.

3. Partitioning: You can choose automatic partitioning or configure it manually if needed.

4. Network Configuration: If prompted, configure your network settings.

5. Installation Summary: Review your settings and click "Begin Installation."

▎Step 7: Set Password and creat user.

1. During installation, you will be prompted to set a root password.

2. Create a user account with a username and password.

Step 8: Complete Installation

1. Once the installation is complete, you will be prompted to reboot your system.

2. Remove the USB drive when prompted.

Step 9: First Boot

1. After rebooting, you should see the Qubes OS boot menu.

2. Select your installation and press Enter.

3. Log in with the user account you created.

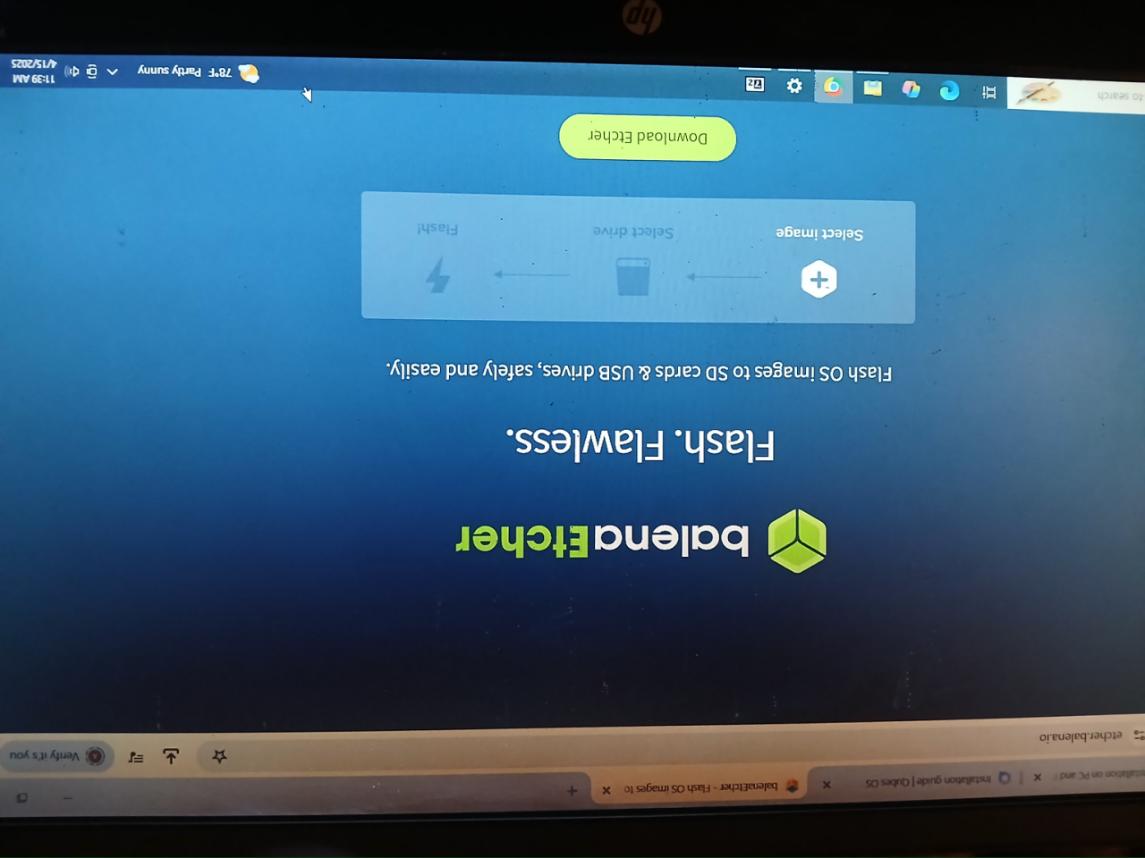
**Problem And Solution**

The main problem that face while working this assignment is the difficulty of the Installation process as the Hardware requirements os this operating system is high I can’t find appropriate hardware device to run the operating System. But I try to work With available device as I can..

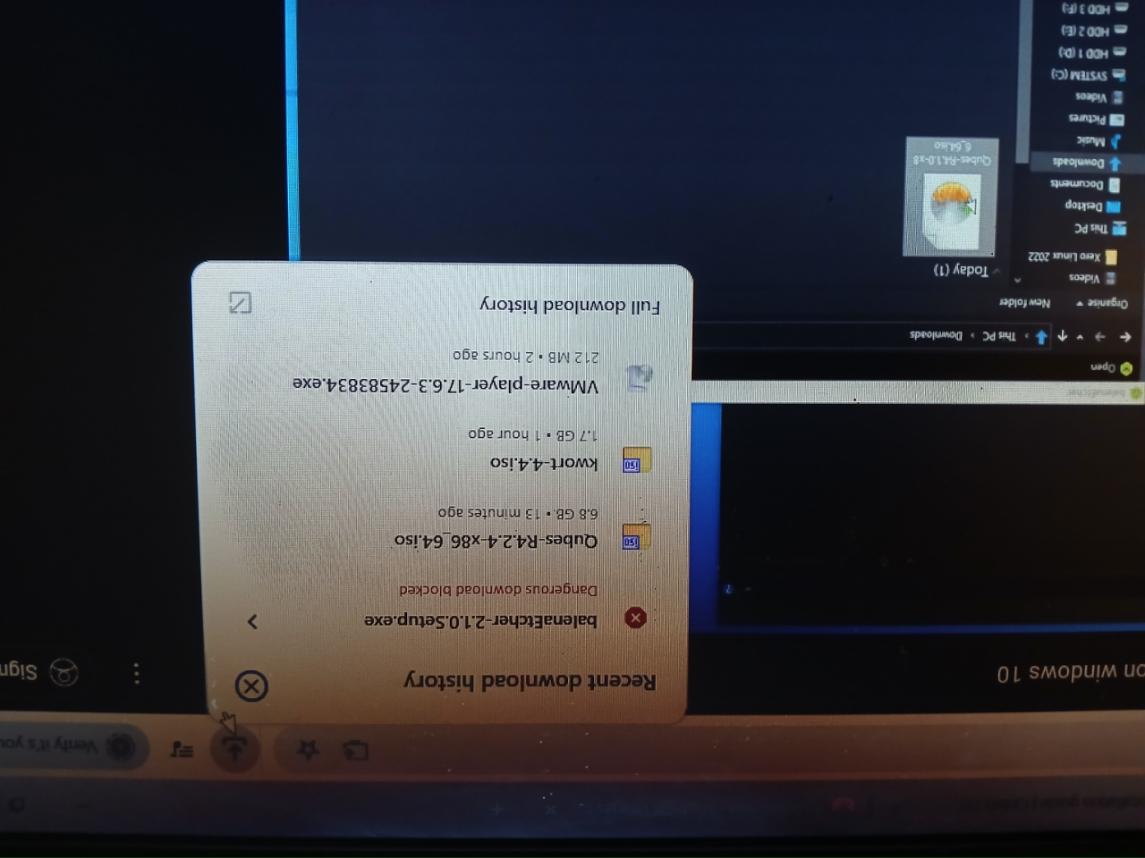
* List of Hardware Shortage

To run the qubes OS as I listed on the Requirements above it needs a bunch of Hardware to work the Operating System corrrectly on the Computer.

64 GB or above external bootable USB,SSD or Hard disk is needed for the safe work of qubes os and I can not find an external device with this specification. So use only 32GB USB device(Flash Disk) which is the only available device that I found. Then I proceed to the installation process for bootable software which is balena etcher. Then when I attempt to download it does not go well.

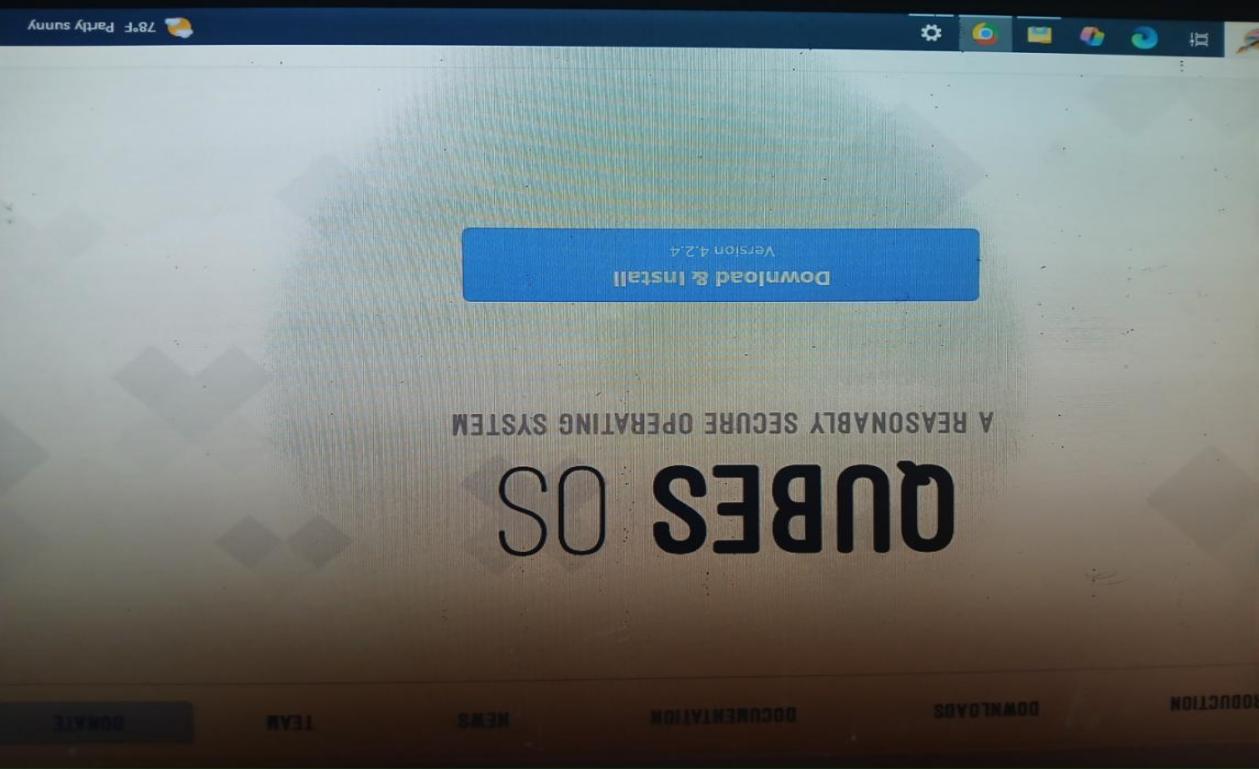


Then when I attempt to download it does not go well. And with my computer specification it is difficult and it may cause halt the computer I f I run balena etcher. And the following warning appears.

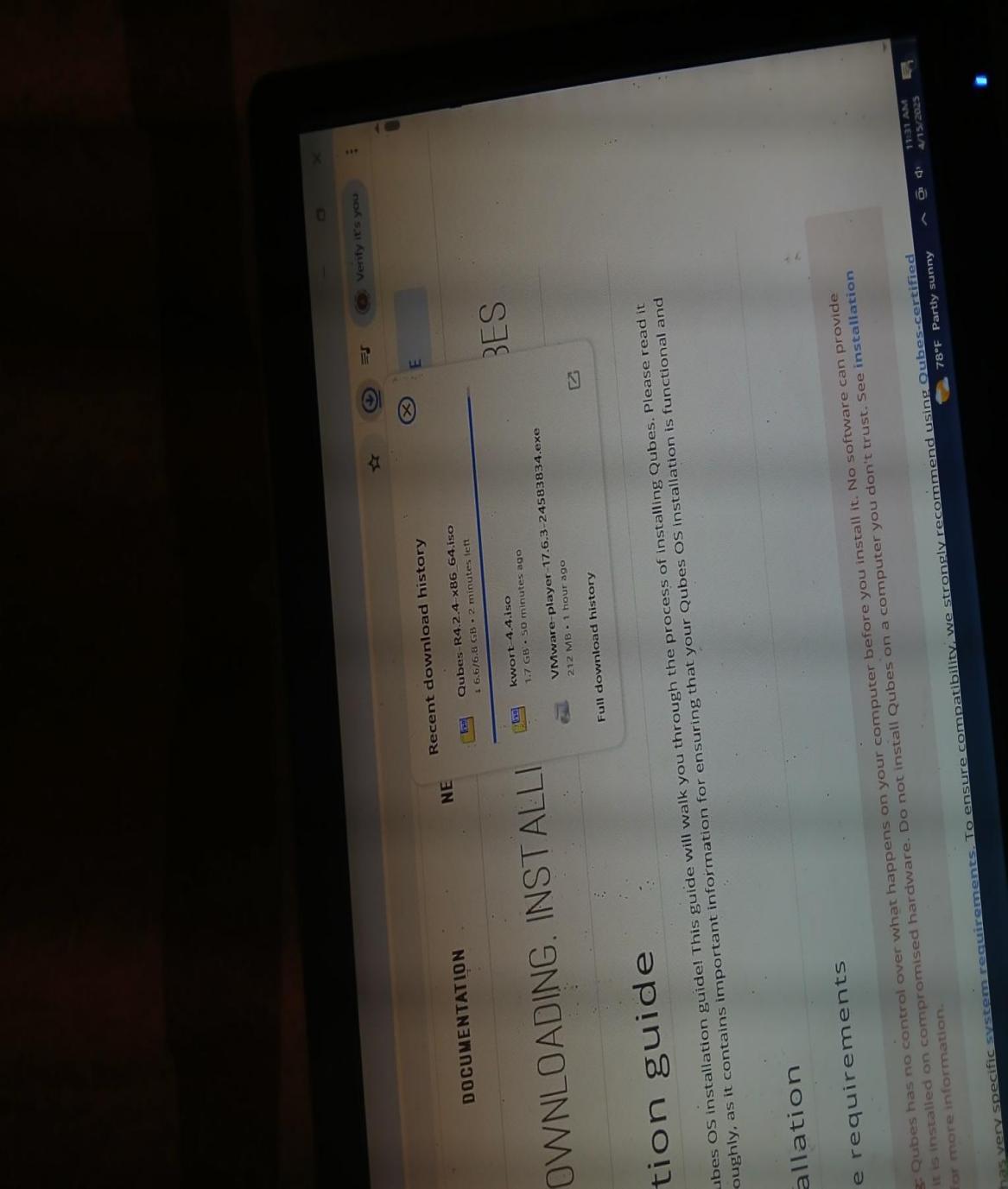


Then I forced to use the Main internal disk on the computer with out considering the issues faced on the computer when it install directly to the computer side to side with the other operating system which is Window.

First I download the Latest Qubes os Qubes OS 4.2 from the official website of Qubes os which is [www.qubes](http://www.qubes) os.org.



On the process of Downloading qubes os



Then when I try to install it does not go well so I try to figure out the problem I found it is the performance of the computer I use computer with this specification. As I listed to make qubes os work in computer the computer needs Intel VT-x and VT-d. I try to install with the following Computer processor specifications:-

Intel(R) Core(TM) i3-8130U CPU @ 2.20GHz 2.20 GHz

My Computer, a venerable desktop . Its specifications – an Intel Core i3-8130U, 4GB of RAM, and HDD storage device – seemed, on paper, sufficient for a reasonable Qubes OS experience. The i7-8130U has the crucial VT-x and VT-d boxes, enabling the necessary hardware virtualization features.

Unfortunately, upon attempting the Qubes OS installation, I hit a wall. The installer would consistently hang at a specific point, seemingly during the hardware detection phase. Initial troubleshooting steps yielded little progress. I meticulously verified the downloaded ISO's signature, ensuring it wasn't corrupted. I tried different ways and tools, ruling out issues with the installation media. I even ventured into the BIOS/UEFI settings, double-checking that VT-x and VT-d were enabled, and experimenting with various other settings, such as disabling Secure Boot (which, while generally recommended for troubleshooting, didn't resolve the issue).

After much head-scratching and forum searching, I began to suspect a deeper hardware incompatibility. The i3-8130U, while supporting VT-x and VT-d, is an aging platform. The motherboard, a somewhat generic OEM board, might be lacking in specific features or updates necessary for proper Qubes OS compatibility. Further research revealed reports of similar issues with older chipsets, even when the CPU itself met the requirements.

This experience, while frustrating, has been a valuable lesson in the practical realities of operating system deployment. It highlighted the importance of not just meeting the minimum hardware requirements, but also ensuring broader hardware compatibility, particularly when dealing with a specialized operating system like Qubes OS. While the CPU itself supported the necessary virtualization features, the limitations of the motherboard and chipset ultimately prevented a successful installation.

So, lacking the ability to physically install Qubes OS, I've had to shift my focus. Instead of a hands-on practical exploration, this assignment has become a more theoretical analysis, relying on the wealth of information available. the Qubes OS documentation, community forums, research papers, and the experiences of other users.

## ****File System Support in Qubes OS****

Qubes OS is a security-focused Linux-based operating system that relies on the Xen hypervisor to compartmentalize applications into isolated virtual machines (qubes). Like other Linux systems, it supports various file systems depending on **where and how** they are used — for internal storage, external drives, or inter-VM data exchange.

There is a list of file systems that Qubes OS support. They are sorted below.

### **Supported File Systems**

#### **1. ext4 (Recommended and Fully Supported)**

· **Usage**: Primary file system for dom0 and internal qubes.

· **Why Supported**:

Native to Linux and extremely stable.

Offers journaling and good performance.

Supported out-of-the-box by all Linux templates (Fedora, Debian, Whonix).

· **Use Case in Qubes**: Used by default for system installation (dom0), VM root and home volumes.

**Usage**: Primary file system for dom0 and internal qubes.

**Why Supported**:

Native to Linux and extremely stable.

Offers journaling and good performance.

Supported out-of-the-box by all Linux templates (Fedora, Debian, Whonix).

**Use Case in Qubes**: Used by default for system installation (dom0), VM root and home volumes.

#### **2. Btrfs (Partially Supported)**

**Usage**: Optional file system for advanced users.

**Why Supported**:

Available in Linux, supports snapshotting and compression.

Can be used manually in VMs or as custom dom0 partitioning.

The limitation of Btrfs : Not used by default; some tools may not support Btrfs fully in dom0.

#### **3. FAT32 / exFAT (Supported for External Devices)**

**Usage**: USB drives and external storage.

**Why Supported**:

Supported by Linux kernel; useful for exchanging files with Windows/macOS.

Qubes allows temporary mounting of USB drives in qubes like sys-usb or work.

**Limitations**:

No journaling.

exFAT may require installing extra packages in some templates (e.g., exfat-utils).

#### **4. NTFS (Read/Write Supported via FUSE or Kernel Driver)**

**Usage**: External drives or file exchange with Windows systems.

**Why Supported**:

Available via ntfs-3g or kernel modules.

Often used when reading or writing to USB drives or backup disks.

**Limitations**:

Performance is lower than native Linux file systems.

Not suitable for root or system partitions.

### **Not Supported or Not Recommended**

#### **5. ZFS (Not Officially Supported)**

**Why Not Supported**:

Requires external packages not included in Qubes dom0 (for security and stability reasons).

Licensing incompatibility with Linux kernel (ZFS is CDDL, Linux is GPL).

**Possible Workaround**:

Can be used in specific VMs if ZFS tools are manually installed.

Not supported for dom0 or LVM-based qube volumes.

#### **6. HFS+ (Limited Read Support)**

**Why Not Recommended**:

macOS file system; limited and outdated support in Linux.

Read support is available via kernel module, but write support is unstable.

**Use Case**:

Can be used for reading Mac-formatted drives in a qube.

#### **7. APFS (Not Supported)**

**Why Not Supported**:

Proprietary file system developed by Apple for macOS and iOS.

No official or reliable Linux support, especially for writing.

**Consequence**:

Qubes OS cannot access APFS drives unless reformatted or accessed from a macOS VM (which is not officially supported on Qubes)

SUMMARY TABLE

|  |  |  |  |
| --- | --- | --- | --- |
| File System | Supported in Qubes OS | Read/Write | Typical use case |
| ext4 | YES | R/W | Dom0,VM sisks |
| Btrfs | YES | R/W | Advanced VM use |
| FAT32 | YES | R/W | Use drives |
| exFAT | YES(with tool) | R/W | Modern USB drives |
| NTFS | YES(with tool) | R/W | External storage |
| ZFS | Expermental | R/W(Manual) | Advanced users |
| HFS+ | Read Only | Read-only | macOS drives |
| APFS | No | None | Not supported |

## ****Advantages and Disadvantages of Qubes OS****

**Qubes OS** is a security-oriented operating system designed to provide strong isolation between applications, based on the principle of “security through compartmentalization.” While it offers unparalleled security benefits, it also introduces certain usability and hardware compatibility challenges. This note highlights the key **advantages** and **disadvantages** of using Qubes OS.

## **Advantages of Qubes OS**

### 1. **Security Through Isolation**

**Key Feature**: Each task or application runs in its own virtual machine (VM), called a “qube.”

**Benefit**: If one qube is compromised, the others remain unaffected, minimizing the risk of full system compromise.

**Example**: You can open an untrusted file in a Disposable VM (dispVM) without endangering your personal data.

### 2. **Compartmentalization of Workflows**

Users can create qubes for specific purposes such as work, personal, banking, development, etc.

Helps prevent accidental data leaks or cross-contamination between different activities.

### 3. **Whonix and Tor Integration**

Qubes supports **Whonix** templates for anonymous internet browsing through **Tor**.

This offers enhanced privacy without needing to run a full Tor OS separately.

### 4. **Minimal Trusted Computing Base**

The central part of the system (dom0) is small and isolated, reducing the number of components that need to be trusted.

Applications are installed in template VMs, not in the core OS.

### 5. **Visual Security Cues**

Qube windows have colored borders (e.g., red for untrusted, green for work) to help users stay aware of what context they are in.

This minimizes the chance of mistakenly entering credentials in the wrong VM.

### 6. **Support for Multiple Linux Distros and Windows**

Qubes allows different templates such as Fedora, Debian, and even Windows-based VMs.

This is helpful for users who need to work with different environments or applications.

### 7. **Disposable VMs for One-Time Use**

Qubes enables launching temporary VMs that are destroyed after use, perfect for opening suspicious files or links securely.

These VMs leave no traces after closing.

## **Disadvantages of Qubes OS**

### 1. **High Hardware Requirements**

Qubes needs a 64-bit processor with **VT-x/AMD-V** and **VT-d/AMD-Vi** virtualization support.

**Minimum**: 8 GB RAM and SSD

**Recommended**: 16 GB+ RAM, high-performance CPU, and good I/O devices

Many modern ultrabooks, and especially older laptops, do not meet these requirements.

### 2. **Steep Learning Curve**

The concept of qubes, dom0, template VMs, and Disposable VMs can be confusing to new users.

Managing data between qubes (copy/paste, file sharing) is not as seamless as in traditional systems.

### 3. **Reduced Compatibility with Certain Software**

Many software applications assume direct hardware access or a monolithic OS, which Qubes does not allow.

Applications like video editors, 3D games, or design tools may not work due to lack of GPU passthrough or 3D acceleration.

### 4. **Limited Peripheral Support**

Devices like **NVIDIA GPUs**, **some Wi-Fi adapters**, **Bluetooth**, or **fingerprint scanners** may not work properly.

USB isolation adds complexity: users must manually assign USB devices to specific qubes.

### 5. **Not Ideal for Average Users**

While Qubes offers incredible security, it’s not designed for everyday casual users or beginners.

Tasks like installing software, setting up printers, or configuring VPNs are more complex compared to Ubuntu or Windows.

### 6. **Resource Intensive**

Running multiple VMs simultaneously consumes significant system memory and CPU.

Performance might degrade on systems with lower specs, even during basic multitasking.

### 7. **Dom0 Software Limitation**

You cannot browse the internet or install regular apps in dom0 (for security reasons).

This adds a layer of protection but also limits flexibility and convenience for quick tasks.

**Conclusion about Qubes OS**

Qubes OS is a highly secure, privacy-focused operating system that relies on the security through isolation principle. Through compartmenting different tasks, applications, and data in light virtual machines (qubes), it significantly reduces the chance of system compromise by malware or user error.  
It is ideal for those who are handling sensitive information, such as reporters, developers, researchers, and security engineers, and who require complete command over how apps and data converse. Disposable VMs, Whonix integration, and reliable hardware isolation are some of the elements that result in Qubes OS possessing one of the best desktop security designs to be available in contemporary computing.  
But Qubes is not ideal for everyone. It requires deep hardware, high learning curve, and bad support for some in-demand applications and peripherals. But it is suited for power users who are capable of sacrificing convenience for security and are okay dealing with multiple virtual environments.  
In general, Qubes OS is the most secure desktop OS available—but with great power comes great complexity. If you're willing to invest time learning and have the proper hardware, Qubes offers control and protection unparalleled.

## ****Future Outlook and Recommendations for Qubes OS****

### **Future Outlook**

Qubes OS has carved a unique space in the world of secure operating systems, and its future looks promising as the need for privacy and compartmentalized computing continues to grow. With increasing cyber threats, surveillance concerns, and remote work environments, the demand for systems that can isolate and control digital risks is higher than ever. Qubes OS is well-positioned to meet these needs.

In the coming years, we can expect:

**Better Hardware Support**: As more users adopt Qubes, compatibility with a wider range of modern hardware (including newer AMD and Intel processors) is anticipated.

**Improved Performance**: Optimization efforts in virtualization and memory usage will likely improve user experience and reduce resource consumption.

**GUI Enhancements**: Efforts to make the user interface more intuitive and beginner-friendly may help reduce the learning curve.

**Community Growth**: An expanding user base and developer community will contribute to more templates, documentation, and tools.

### **Recommendations**

**For Users**:

**Advanced Users**: Qubes OS is highly recommended for privacy-conscious individuals such as developers, journalists, and cybersecurity professionals.

**New Users**: Start with tutorials and community forums. Gradually adopt Qubes alongside a traditional OS while learning its mode

**For Developers**:

Contribute to improving user experience, especially around hardware compatibility and graphical tools.

Build more Qubes-compatible applications or automate repetitive tasks (e.g., updates, backups).

**For Organizations**:

Adopt Qubes OS for high-security environments or personnel who manage sensitive data.

Offer employee training to help staff use the OS efficiently and securely.

1. Briefly explain the what, why, and how virtualization in modern operating system.

## **Virtualization in Modern Operating Systems**

## **1. What is Virtualization?**

**Virtualization** is the process of creating a **virtual version of computing resources**, such as operating systems, servers, storage devices, or networks. In simpler terms, it allows multiple "virtual machines" (VMs) to run on a single physical computer.

Each VM behaves like a separate computer, complete with its own OS, applications, memory, and storage. but they are all running on shared underlying hardware. This is made possible using a software layer called a **hypervisor**.

### **2. Why Virtualization?**

#### A. **Better Hardware Utilization**

Before virtualization, most computers ran a single OS and one workload, leading to underutilization of CPU and memory. Virtualization allows multiple systems to share the same hardware efficiently.

#### B. **Isolation and Security**

Virtual machines are **isolated** from each other. If one VM crashes or gets infected with malware, it doesn't affect the others. This is a core benefit for secure environments like **Qubes OS**.

#### C. **Flexibility and Portability**

You can run different operating systems (e.g., Linux, Windows, BSD) on the same machine. It also enables moving VMs between physical machines easily.

#### D. **Testing and Development**

Developers can test new code in a safe environment without impacting the host system. You can clone or reset VMs quickly.

#### E. **Disaster Recovery and Snapshots**

Virtual machines can be backed up, paused, or reverted to earlier states using **snapshots**, helping with disaster recovery and quick rollbacks.

#### F. **Cloud and Server Efficiency**

Cloud computing depends heavily on virtualization. Data centers use virtualization to serve many users on fewer physical servers, reducing cost and energy use.

### ⚙️

### **3. How Does Virtualization Work?**

#### a. **Hypervisor**

A **hypervisor** (or virtual machine monitor) is the software that manages virtual machines. It acts as a bridge between the physical hardware and the virtual environments.

There are two types:

**Type 1 Hypervisor (Bare-metal)**  
Runs directly on hardware. Examples: **Xen (used in Qubes OS)**, **VMware ESXi**, **Microsoft Hyper-V**  
 Offers better performance and security.

**Type 2 Hypervisor (Hosted)**  
Runs on top of a host OS. Examples: **VirtualBox**, **VMware Workstation**, **QEMU/KVM**  
 Easier to set up but less efficient.

#### b. **Virtual Machine (VM)**

Each VM includes:

A **virtual CPU**, memory (RAM), and storage

A **virtual BIOS or UEFI**

A **virtual OS**, which runs as if it's on its own machine

The hypervisor allocates real hardware resources to these virtual machines as needed.

#### c. **Hardware Virtualization Support**

Modern CPUs from Intel and AMD support virtualization through:

**Intel VT-x** / **AMD-V**: Enables the CPU to efficiently run multiple OSes.

**Intel VT-d** / **AMD-Vi**: Allows secure, direct device access to VMs (used for GPU or USB passthrough).

Operating systems like Qubes OS, Windows 10 Pro (with Hyper-V), and Linux distributions use these features to improve virtualization performance and security.

### 🛠️ **Common Use Cases in Modern OS**

**Qubes OS**: Uses virtualization for strict security — each app runs in its own isolated VM.

**Linux Servers**: Use KVM (Kernel-based Virtual Machine) for hosting multiple services.

**Windows 10/11**: Offers Hyper-V for testing and sandboxing.

**macOS**: Developers use VirtualBox or UTM to run other OSes.

**References**

"Qubes OS: A Security Architecture for Desktop Computing" by Joanna Rutkowska

"The Qubes OS Security Model" by Joanna Rutkowska

Rutkowska, J., Kwiatkowska, M. (2015). "Qubes OS: A Security Architecture for Desktop Computing."

"What is Qubes OS?" by TechRadar

"Qubes OS: A Practical Guide" by Linux Journal.

[www.qubes](http://www.qubes) -os.org

You tube: Qubes OS: install, You tube channel: ”Bluewire solution”

You tube: How to install QUBES OS on external USB drive, You tube channel: Knowledge sharing Tech

You tube: Qubes OS Review/ Window Integration, You tube channel:Chris Tikus Tech

You tube: Qubes OS Tutorial, You tube channel:Chris Tikus Tech

You tube: Qubes OS Installation on PC and preview 2022, You tube channel: Apk Heaven

<https://xenproject.org/docs/>

<https://kernel.org>