BAHIR DAR INSTITUTE OF TECHNOLOGY

FACULTY OF COMPUTING

DEPARTMENT OF SOFTWARE ENGINEERING

Individual ASSIGNMENT FOR ossp

PREPARED BY: YIHALEM GIRMA

ID NUMBER: 1602775

Submitted to: Lec.wendimu

Submission date: 16/08/2017

BAHIRDAR, ETHIOPIA

1.Installation of Parrot OS in a Virtual Environment

A. Introduction (Background & Reasons)

Background: Parrot OS, a Linux distribution based on Debian, provides security, privacy along with development features. The OS comes with pre-installed utilities for penetration testing, reverse engineering, digital forensics as well as private internet access. It fits security professionals plus students who study secure computing. It functions on less advanced computers and newer models.

Reasons: I am a second-year Software Engineering student. I wished to learn how secure systems obtain installation, administration next to function. Parrot OS gives me experience in Linux administration besides security tests. Through installation inside a virtual machine, I can test its utilities for ethical hacking and digital forensics. This helps me as a full-stack developer also system programmer later.

B. Goals

Secure OS Installation Study: Acquire abilities for installing and setting up a Linux system inside a virtualized space, concentrating on Parrot OS.

Cybersecurity Tools Study: Become acquainted with Parrot OS integrated utilities for ethical hacking, penetration testing in addition to digital forensics.

Virtualization Study: Obtain knowledge of how virtualization platforms, such as VirtualBox, function plus administrate virtual resources.

Filesystem Knowledge Application: Discover how to pick and set up applicable filesystems, like ext4, for secure besides lasting OS function.

Linux Experience Gain: Practice Linux commands and administration utilities. The practice supports development also system programming.

C.Requirements

i. Hardware:

Processor: A 64-bit CPU with virtualization support (Intel VT-x or AMD-V).

Memory (RAM): At least 4 GB is needed, but 8 GB or more is recommended for smooth

performance.

Hard Drive: Minimum 20 GB of available disk space.

Graphics: A graphics adapter compatible with virtualization.

ii. Software:

Virtualization Tool: Oracle VM VirtualBox

Operating System ISO: Parrot OS ISO file (Home or Security Edition) from https://www.parrotsec.org

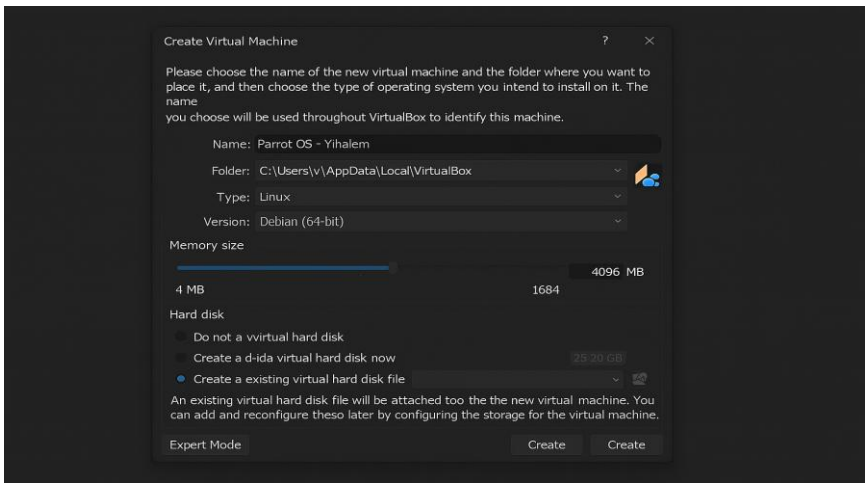
Additional Utilities: Tools like 7-Zip to extract ISO files and VirtualBox Extension Pack for enhanced USB and graphics support

D. Installation Steps

The following are step-by-step instructions to install Parrot OS on a VirtualBox virtual machine:

1. Download ISO File: Go to Parrot OS download page and select your preferred edition.

2. Launch VirtualBox: Open the VirtualBox application and click the “New” button to create a new virtual machine.



3. Configure the VM:

Name: Parrot OS - Yihalem

Type: Linux

Version: Debian (64-bit)

4. Set Memory Size: Allocate at least 4096 MB (4 GB) of RAM.

5. Create Virtual Hard Disk:

Choose VDI (VirtualBox Disk Image) format.

Set disk size to at least 20 GB.

6. Attach ISO File:

Go to the virtual machine's settings → Storage → Empty → Click the disk icon → Choose the downloaded Parrot OS ISO file.

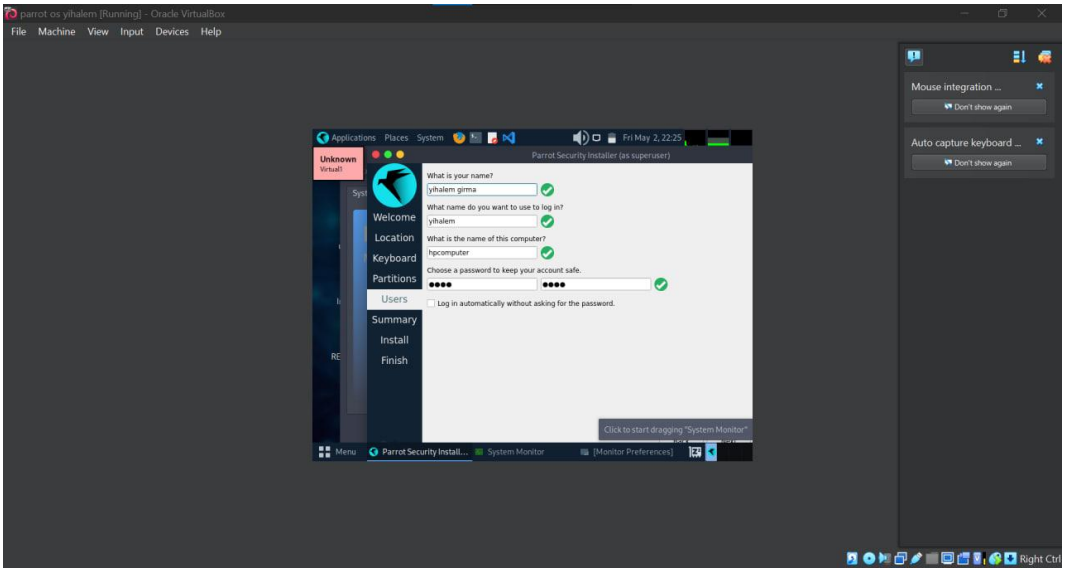
Search on Google: VirtualBox attach ISO screenshot

7. Start the Virtual Machine: Click "Start" to boot the VM and begin the installation.

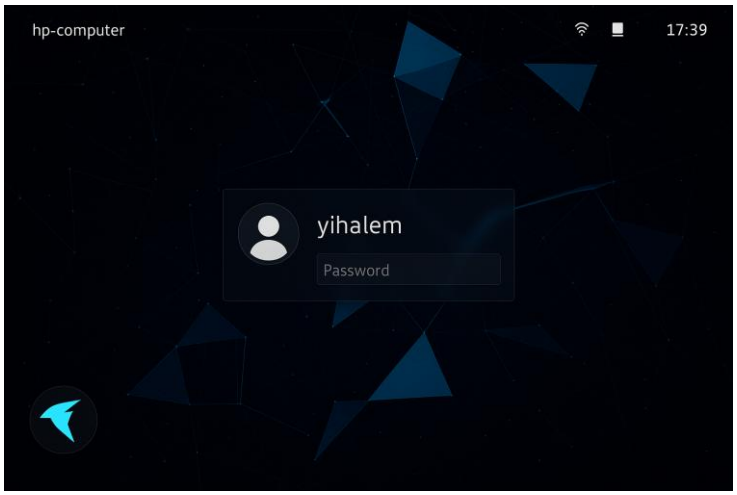
8. Installer Configuration:

Choose language, location, and time zone.

Set up user credentials (Full name: Yihalem Girma, Username: yihalem



9. Partition the Disk: Choose "Guided – Use entire disk" and select ext4.



10. Install GRUB Bootloader: Install the GRUB bootloader to allow the system to boot properly.

11. Finish and Reboot: Remove the ISO from virtual disk settings and reboot the VM.

Search on Google: Parrot OS login screen

12. Log in and Start Exploring: Use the credentials created earlier to access the Parrot OS desktop environment.

E. Issues (Problems Faced)

1. Installation Freeze: The setup process froze during the package installation. This was resolved by re-downloading the ISO file.

2. Network Adapter Issues: The VM did not automatically connect to the internet. Switching to a "Bridged Adapter" in VirtualBox fixed this issue.

3. Low Screen Resolution: Initially, the display was limited. Installing VirtualBox Guest Additions solved this.

4. Slow Performance: Due to low RAM, the VM was slow. Increasing the RAM allocation fixed this.

5. Guest Additions Not Installing: Guest Additions failed due to missing dependencies. Installing linux-headers and build-essential packages resolved the issue.

F. Solutions

1. Re-downloaded the ISO: A corrupted ISO file was replaced to fix the installation freeze.

2. Enabled VT-x in BIOS: This improved virtual machine performance.

3. Switched Network Mode: The network mode was set to “Bridged Adapter” for better internet connectivity.

4. Increased RAM: Allocating more memory (4 GB or higher) improved the speed.

5. Installed Guest Additions: Required packages were installed, making it work as expecte

G. Filesystem Support

Parrot OS supports multiple file systems, but the most common ones are:

ext4: Default and recommended for most Linux systems.

Btrfs: Used by advanced users who need features like snapshots.

FAT32, exFAT, NTFS: Used for compatibility with Windows, especially for external drives.

H. Advantages and Disadvantages

Advantages:

Comes with a range of hacking, analysis, and privacy tools.

Lightweight and optimized for low-end hardware.

Strong community and regular updates.

Disadvantages:

Requires basic Linux knowledge.

Not as user-friendly as other Linux distributions like Ubuntu.

Limited support for certain Wi-Fi or graphics hardware.

I. Conclusion

Installing Parrot OS in a VirtualBox environment helped me learn important system administration skills such as partitioning, selecting filesystems, and troubleshooting installation issues. These skills are essential for cybersecurity, full-stack development, and system programming.

J. Future Outlook / Recommendations

1. Learn More Linux Commands: Continue practicing with shell scripting and administrative tools in Parrot OS.

2. Explore Cybersecurity Tools: Dive deeper into the tools provided by Parrot OS to understand ethical hacking, penetration testing, and forensics.

3. Try Dual-Booting: For better performance, consider setting up Parrot OS alongside your main OS on a physical machine.

4. Contribute to the Open Source Community: Contribute to Parrot OS or other open-source projects to improve my development skills.

2.What is Virtualization in Modern Operating Systems?

Virtualization is the technology that allows the creation of virtual (rather than physical) versions of resources, such as servers, storage devices, and networks. This enables a single physical machine to host multiple virtual environments, each of which behaves like a separate computer. These environments, called virtual machines (VMs), run their own operating systems and applications.

This process is made possible by a software layer called the hypervisor, which manages the allocation of physical resources (like CPU, memory, and storage) to each virtual machine.

Why Virtualization is Important?

1. Better Resource Utilization:

Virtualization maximizes the use of physical hardware. Instead of having several underutilized physical machines, it allows multiple virtual machines to share the same physical resources, such as CPU, memory, and storage. This results in improved efficiency, ensuring hardware is being used to its fullest potential.

2. Cost Efficiency:

By consolidating multiple servers or systems into a single physical machine, organizations can reduce their capital expenses (hardware costs), operational expenses (power, cooling, space), and maintenance costs. Virtualization helps businesses lower overall IT infrastructure expenses.

3. Increased Flexibility and Scalability:

Virtualization provides unparalleled flexibility. You can quickly create, modify, or destroy virtual machines based on demand. This flexibility is especially valuable in cloud environments, where resources can be allocated dynamically. Virtualized environments can be scaled up or down with minimal effort, making them ideal for businesses with fluctuating workloads.

4. Improved Security and Isolation:

Since virtual machines are isolated from one another, any issues or security breaches in one VM do not affect the others. This isolation enhances security, as malware or errors in one VM can't easily spread to others. Virtual machines can also be configured with different security settings, providing granular control over each environment.

5. Better Disaster Recovery:

Virtualization simplifies disaster recovery. Virtual machines can be backed up easily and moved between different physical servers without compatibility issues. In case of hardware failure, a virtual machine can be quickly restored on another machine, minimizing downtime and data loss.

6. Simplified Testing and Development:

Developers and IT teams can use virtualization to create isolated environments for testing software, configurations, or updates. These environments can be spun up quickly and do not interfere with production systems. Developers can simulate different operating systems, test multiple configurations, and quickly roll back changes without affecting the primary system.

7. Environment Replication:

Virtualization allows for easy replication of environments. For example, testing a production environment in a virtual machine makes it possible to simulate real-world conditions without the risks associated with testing directly on production hardware. This feature is beneficial in quality assurance, development, and training scenarios.

8. Legacy System Support:

Virtualization can run older operating systems and software that may not be compatible with newer physical hardware. By creating virtual environments, businesses can continue to run legacy applications without needing to maintain outdated physical servers.

How Virtualization Works

Virtualization works by using a software layer that emulates or abstracts physical hardware, allowing multiple virtual machines (VMs) to run on a single physical machine. This software layer is called a hypervisor. Here's how it works step by step:

1. Hypervisor (Virtual Machine Monitor - VMM):

The hypervisor is the key component in virtualization. It sits between the hardware (physical machine) and the operating systems running inside virtual machines. There are two types of hypervisors:

Type 1 (Bare Metal): This hypervisor runs directly on the physical hardware, without an underlying host operating system. It has direct access to the hardware and is more efficient for large-scale environments. Examples include VMware ESXi, Microsoft Hyper-V, and Xen.

Type 2 (Hosted): This hypervisor runs on top of a host operating system, which in turn manages the hardware. While this is easier to set up for personal use or testing, it is less efficient than a Type 1 hypervisor. Examples include VMware Workstation, Oracle VirtualBox, and Parallels Desktop.

2. Virtual Machine (VM):

A virtual machine is a software-based computer that runs its own operating system and applications. Each VM operates independently, as though it were a separate physical computer, even though they are sharing the underlying physical resources.

Guest OS: Each VM has its own operating system (the "guest OS"), which is installed on top of the virtualized hardware provided by the hypervisor. The guest OS can be any operating system, such as Windows, Linux, or macOS.

Virtual Hardware: The hypervisor emulates the hardware for each VM. It provides virtualized resources such as CPU, memory, storage, network adapters, and other devices. These virtualized resources allow the guest OS to interact with the VM as if it were running on physical hardware.

3. Resource Allocation:

The hypervisor manages and allocates physical resources (like CPU power, RAM, and disk space) among all running VMs. The hypervisor ensures that each VM gets a fair share of the resources, and it can dynamically allocate more or less based on demand.

CPU Virtualization: The hypervisor divides the physical CPU into multiple virtual CPUs, which are assigned to different VMs. It manages CPU time to ensure that each VM gets its required processing power without interfering with others.

Memory Virtualization: The hypervisor allocates physical RAM to each VM. VMs may not need all the allocated memory at once, so the hypervisor can use techniques like memory overcommitment or paging to optimize memory usage.

Storage Virtualization: Virtual disks are created for each VM, which appear as physical hard drives to the guest OS. These virtual disks can be stored on physical disks (or in the cloud) and managed by the hypervisor.

4. Isolation and Security:

Each virtual machine is isolated from the others. This means that processes or problems within one VM (e.g., software crashes or security breaches) won’t affect other VMs or the host machine. Isolation is achieved by the hypervisor, which enforces boundaries between VMs and ensures that resources are allocated safely.

5. Virtual Networks:

Virtual machines can be connected to one another and the external network through virtual network interfaces. The hypervisor can create virtual switches and routers to manage traffic between VMs and physical networks. This allows VMs to communicate just like physical machines in a network environment.

6. Snapshot and Cloning:

One of the advantages of virtualization is the ability to take snapshots of a VM, which is a point-in-time copy of the entire VM (including its OS, applications, and data). If something goes wrong, you can roll back the VM to its snapshot state. Cloning allows you to create identical copies of a VM to replicate environments or distribute workloads.

7. Live Migration (Optional):

In some advanced virtualization setups, VMs can be moved from one physical host to another without shutting them down. This process is called live migration and is often used in cloud computing or for load balancing.

How Virtualization Enables Different Environments

1. Consolidation:

Virtualization allows multiple operating systems to run on a single physical machine. For example, you could run Windows, Linux, and macOS on different VMs on the same computer, each isolated from the others. This reduces hardware requirements and simplifies management.

2. Testing and Development:

Developers can use virtualization to create isolated environments for testing different applications or configurations. They can test software in a variety of operating systems without needing separate physical machines, enabling faster development cycles.

3. Cloud Computing:

Virtualization is at the core of cloud computing. Virtual machines can be provisioned on-demand, allowing users to rent computing power from a cloud provider without needing to own or manage physical hardware. The cloud provider uses virtualization to allocate resources dynamically based on customer needs.

4. Disaster Recovery:

Virtualization makes it easy to back up and restore entire systems. Since VMs are portable, they can be moved to another server or data center if one fails, ensuring minimal downtime. Disaster recovery processes are faster and less costly than traditional methods

.In summary, virtualization enhances resource efficiency, scalability, and security in modern OS environments.