Introduction

Please comply with the following rules: - Remain polite, courteous, respectful and constructive throughout the evaluation process. The well-being of the community depends on it. - Identify with the student or group whose work is evaluating the possible dysfunctions in their project. Take the time to discuss and debate the problems that may have been identified. - You must consider that there might be some differences in how your peers might have understood the project's instructions and the scope of its functionalities. Always keep an open mind and grade them as honestly as possible. The pedagogy is useful only and only if the peer-evaluation is done seriously.

Guidelines

- Only grade the work that was turned in the Git repository of the evaluated student or group.
- Double-check that the Git repository belongs to the student(s). Ensure that the project is the one expected. Also, check that "git clone" is used in an empty folder.
- Check carefully that no malicious aliases were used to fool you and make you evaluate something that is not the content of the official repository.
- To avoid any surprises and if applicable, review together any scripts used to facilitate the grading (scripts for testing or automation).
- If you have not completed the assignment you are going to evaluate, you have to read the entire subject prior to starting the evaluation process.
- Use the available flags to report an empty repository, a non-functioning program, a Norm error, cheating, and so forth. In these cases, the evaluation process ends and the final grade is 0, git@vogsphere.42istanbul.com.tr:vogsphere/intra-uuid-8465856c-659c-495a-a398-e83 (https://profile.intra.42.fr) (https://profile.intra.42.fr/searches) gyildiz Intra Projects Born2beroot Edit https://projects.intra.42.fr/scale_teams/7587171/edit 1 of 8 12/14/24, 18:30 or -42 in case of cheating. However, except for cheating, students are strongly encouraged to review together the work that was turned in, in order to identify any mistakes that shouldn't be repeated in the future.

Attachments

subject.pdf

Preliminaries

If cheating is suspected, the evaluation stops here. Use the "Cheat" flag to report it. Take this decision calmly, wisely, and please, use this button with caution. Preliminary tests

• Defense can only happen if the student being evaluated or group is present. This way everybody learns by sharing knowledge with each other.

• If no work has been submitted (or wrong files, wrong directory, or wrong filenames), the grade is 0, and the evaluation process ends.

• For this project, you have to clone their Git repository on their station.

General instructions

• During the defense, as soon as you need help to verify a point, the student evaluated must help you.

• Ensure that the "signature.txt" file is present at the root of the cloned repository.

• Check that the signature contained in "signature.txt" is identical to that of the ".vdi" file of the virtual machine to be evaluated. A simple "diff" should allow you to compare the two signatures. If necessary, ask the student being evaluated where their ".vdi" file is located.

• As a precaution, you can duplicate the initial virtual machine in order to keep a copy.

• Start the virtual machine to be evaluated.

• If something doesn't work as expected or the two signatures differ, the evaluation stops here

Mandatory part

The project consists of creating and configuring a virtual machine following strict rules. The student being evaluated will have to help you during the defense. Make sure that all of the following points are observed.

Project overview

• The student being evaluated should explain to you simply:

How a virtual machine works?

A virtual machine (VM) is a software-based emulation of a computer that runs an operating system and applications like a physical computer. Here's how it works:

- **Hypervisor:** The VM relies on a hypervisor, which is software that manages and isolates VMs. It can be a bare-metal hypervisor (e.g., VMware ESXi, Microsoft Hyper-V) or a host-based one (e.g., *VirtualBox*, KVM).
- **Resource Sharing:** The hypervisor allocates physical resources (CPU, RAM, disk, etc.) from the host machine to the VM. This allows multiple VMs to run on the same physical hardware.
- **Isolation:** Each VM operates independently with its own OS, applications, and files. Changes in one VM don't affect others or the host.
- **Virtual Hardware:** The VM uses virtualized hardware such as virtual CPUs, network adapters, and storage devices, which the hypervisor maps to real hardware.

Their choice of operating system?

Choosing Debian over Rocky Linux can be based on several factors, depending on the intended use case for your virtual machine. Here are some reasons why Debian might be a better choice for you:

1. Stability with Flexibility: Debian is known for its stability, similar to Rocky Linux, but it also offers flexibility with its three branches:

- **Stable**: Ideal for production environments.
- **Testing**: Balances stability with newer software versions.
- Unstable (Sid): Provides cutting-edge features for development and experimentation.

This makes Debian suitable for a wide range of purposes, from **server deployments** to **desktop systems**, and even **experimental use cases**.

2. Broader Package Ecosystem:

- Debian's package repository is one of the largest among Linux distributions, with over **59,000 packages** available. It supports a wide variety of use cases, including development, multimedia, and scientific computing.
- **Rocky Linux, while robust, focuses more on enterprise environments and inherits RHEL's smaller, enterprise-focused repository.**

3. Independence and Philosophy:

- Debian is an independent, community-driven project that adheres to the Debian Free Software Guidelines (DFSG). It prioritizes software **freedom and transparency**, which might align with your personal or organizational philosophy.
- Rocky Linux, on the other hand, is based on Red Hat Enterprise Linux (RHEL), so it depends on Red Hat's upstream decisions for its direction.

4. Package Management – APT vs. DNF:

- Debian uses the APT (Advanced Package Tool) system, which is simpler, faster, and **more intuitive for beginners and advanced users alike**.
- Rocky Linux uses DNF, which is more suited for enterprise-grade package management but may feel slower or less intuitive.

5. Versatility for Virtual Machines:

- Debian is lightweight and versatile, making it an excellent choice for:
    - Minimalist installations (e.g., headless VMs).
    - Customizing the operating system to specific needs.
- Rocky Linux is optimized for enterprise use cases, particularly as a replacement for CentOS. It might feel heavier or more enterprise-oriented for non-server use cases.

6. Broader Hardware and Software Compatibility:

- Debian supports a broader range of architectures and hardware, including older and niche systems, which makes it highly adaptable for diverse virtual machine environments.

7. Community and Support:

- Debian's long-standing and active community provides extensive documentation, forums, and mailing lists, making it easier to find support for any issues.
- While Rocky Linux also has a growing community, it is relatively newer (released in 2021) and not yet as established as Debian.

8. Use Case Alignment:

If your VM is for general-purpose usage (development, testing, or even running applications), Debian's versatility and simplicity make it a great choice. On the other hand, Rocky Linux might be overkill if you don't need an enterprise-oriented environment.

When You Might Still Prefer Rocky Linux:

- **If your virtual machine is for enterprise-grade applications that require RHEL compatibility or certifications, Rocky Linux might be a better fit due to its focus on stability in such environments.**

In summary, Debian is a better choice for users who need ***versatility***, ***ease of use***, and a ***balance of stability and cutting-edge features*** in ***virtual machine environments.***

The basic differences between Rocky and Debian.

Feature	Rocky Linux	Debian
Base	Based on Red Hat Enterprise Linux (RHEL).	Independent distribution.
Purpose	Designed for **enterprise servers** and stability.	General-purpose **OS** for **desktops and servers**.
Package Manager	**dnf and rpm** (Red Hat Package Manager).	apt and dpkg (Debian Package Manager).
Release Cycle	Follows RHEL's release and lifecycle.	**Stable (long-term), Testing, and Unstable branches.**

Community	Focuses on CentOS replacement and enterprise use.	Broader community with various use cases.

The purpose of virtual machines.

Resource Optimization: Run multiple systems on a single physical machine to maximize resource utilization.

Testing and Development: Provide isolated environments for testing software, OS configurations, or updates without affecting the host system.

Disaster Recovery: Easily create snapshots and backups of VMs for recovery in case of failure.

Cross-Platform Compatibility: Run an OS or application not natively supported on the host machine.

Security: Isolate processes or applications in separate VMs to enhance security.

◦ If the evaluated student chose Debian: the difference between aptitude and apt, and what APPArmor is. During the defense, a script must display information all every 10 minutes. Its operation will be checked in detail later. If the explanations are not clear, the evaluation stops here.

Difference Between aptitude and apt and aptitude:

- A higher-level package manager that includes a text-based user interface.
- Provides **advanced dependency** resolution and **interactive conflict resolution**.
- **More suitable for power users and administrators**.

apt:

- A simpler, modern command-line tool introduced as a front-end to dpkg.
- Focuses on basic package management commands like **installing, upgrading, and removing packages**.
- **Easier to use for new users compared to aptitude**.

What is AppArmor?

- AppArmor (Application Armor) is a Linux kernel security module that enforces mandatory access control (MAC) policies for applications.
- It restricts programs' access to **files, system resources,** and **capabilities**, **reducing the risk of exploits**.
- How It Works: AppArmor uses profiles to define allowed and denied actions for applications. Profiles can be in **enforcing mode** (actively restrict) or **complain mode** (log violations without blocking them).
- Debian Support: AppArmor is included and supported in Debian to enhance security for critical applications.

Simple setup

• Ensure that the machine *does not have a graphical environment at launch.* A password will be requested before attempting to connect to this machine. Finally, connect with a user with the help of the student being evaluated. *This user must not be root*. *Pay attention to the password chosen, it must follow the rules imposed in the subject.*

passwd

• Check that the UFW service is started with the help of the student being evaluated.

sudo systemctl status ufw

• Check that the SSH service is started with the help of the student being evaluated.

sudo systemctl status ssh

• Check that the chosen operating system is **Debian** or Rocky with the help of the student being evaluated. If something does not work as expected or is not clearly explained, the evaluation stops here.

cat /etc/os-release

User Remember:

Whenever you need help checking something, the student being evaluated should be able to help you. The subject requests that a user with the login of the student being evaluated is present on

the virtual machine. Check that it has been added and that it belongs to the **"sudo"** and **"user42"** groups.

getent group sudo

getent group user42

groups bakpulat

Make sure the rules imposed in the subject concerning the password policy have been put in place by following the following steps.

First, create a new user.

sudo adduser deneme

Assign it a password of your choice, respecting the subject rules. The student being evaluated must now explain to you **how they were able to set up the rules** requested in the subject on their virtual machine.

A,a,1,333,deneme,oldpass=X7,minlen10

1. *sudo apt install libpam-pwquality -y*
2. ***dpkg -l | grep libpam-pwquality***
3. ***sudo nano /etc/pam.d/common-password***
4. *ucredit=-1 : En az 1 adet büyük karakter olması için*

   lcredit=-1 : En az 1 adet küçük karakter olması için

   dcredit=-1 : En az 1 adet sayı olması için

   maxrepeat=3 : En fazla 3 adet karakterin ardışık olması için

   usercheck=1 : Şifre kullanıcı adını içeriyorsa şifrenin geçersiz olması için

   difok=7 : Yeni oluşturulacak şifrenin, eski şifrenin içermediği en az 7 karakteri içermesi için

   enforce_for_root : Tüm bu değişikliklerin "root" kullanıcısına da uygulamak için

   minlen=10 : Şifrenin en az 10 karakter uzunluğunda olması için
5. *İkinci satırın da sonundakileri silip aşağıdaki kodlarla düzeltin:*

   obscure : Şifrenin daha güvenli olması için

   sha512 : Şifreyi sha512 formatında şifrelemek için

Normally there should be one or two modified files. If there is any problem, the evaluation stops here.

- Now that you have a new user, ask the student being evaluated to create a group named "evaluating" in front of you and assign it to this user.

  sudo addgroup evaluating

  sudo adduser deneme evaluating
- Finally, check that this user belongs to the "evaluating" group.

  sudo groups evaluating

  getent group evaluating
- Finally, ask the student being evaluated to explain the advantages of this password policy, as well as the advantages and disadvantages of its implementation. Of course, answering that it is because the subject asks for it does not count. If something does not work as expected or is not clearly explained, the evaluation stops here

  Advantages:

  Enhanced Security: Strong password policies reduce the risk of unauthorized access by requiring passwords that are hard to guess or crack. This helps protect sensitive data from attacks such as brute force, dictionary attacks, and credential stuffing.

  Compliance with Standards: Many industries and organizations are required to follow security regulations (e.g., GDPR, HIPAA, or ISO 27001). Strong password policies ensure compliance, reducing the risk of legal penalties and data breaches.

  Disadvantages:

  User Frustration: Complex password requirements (e.g., minimum length, special characters, and frequent changes) can frustrate users. This may lead to users forgetting their passwords or resorting to insecure practices like writing them down.

  Increased IT Support Overhead: A strong password policy often results in more frequent password resets due to users forgetting their passwords. This increases the workload for IT support teams, which can lead to higher operational costs.

Check that the hostname of the machine is correctly formatted as follows: login42 (login of the student being evaluated).

sudo hostname

• Modify this hostname by replacing the login with yours, then restart the machine. If on restart, the hostname has not been updated, the evaluation stops here.

sudo hostnamectl set-hostname deneme42

sudo nano /etc/hosts

sudo reboot

• You can now restore the machine to the original hostname.

sudo hostnamectl set-hostname bakpulat42

sudo nano /etc/hosts

sudo reboot

• Ask the student being evaluated how to view the partitions for this virtual machine.

sudo lsblk

df -h

sudo fdisk -l

lsblk komutu, sistemdeki tüm blok cihazlarını (diskler, bölümler, sürücüler vb.) listelemek için kullanılır. Çıktısı, fiziksel ve mantıksal diskleri ve onların sahip oldukları bölümleri (partitions) içerir.

• Compare the output with the example given in the subject. Please note: if the student evaluated makes the bonuses, it will be necessary to refer to the bonus example. This part is an opportunity to discuss the scores! The student being evaluated should give you a brief explanation of how LVM works and what it is all about. If something does not work as expected or is not clearly explained, the evaluation stops here.

LVMGroup Nedir?

LVM Group (Logical Volume Manager Group), birden fazla fiziksel depolama birimini bir araya getirerek, mantıksal olarak daha esnek bir şekilde yönetim sağlayan bir disk yönetim yapısıdır. Linux işletim sistemlerinde özellikle büyük veri kümeleriyle çalışırken sıkça kullanılır.

LVM Group'un temel bileşenleri şunlardır:

1. Physical Volume (PV): Fiziksel diski ya da bir disk bölümünü temsil eder.Örneğin: /dev/sda1, /dev/sdb.

2. Volume Group (VG): Birden fazla fiziksel birimi birleştirerek bir grup oluşturur. Bu grup, mantıksal birimlerin oluşturulacağı kaynak havuzudur. Örneğin: VG1, storage_vg.

3. Logical Volume (LV): Volume Group içinde oluşturulan ve dosya sistemiyle kullanılabilen mantıksal birimlerdir. LV'ler, ihtiyaç oldukça boyutlandırılabilir veya yeniden yapılandırılabilir. Örneğin: /dev/VG1/home, /dev/VG1/root.

SUDO

Remember: Whenever you need help checking something, the student being evaluated should be able to help you.

• Check that the "sudo" program is properly installed on the virtual machine.

dpkg -l | grep sudo

• The student being evaluated should now show assigning your new user to the "sudo" group.

sudo visudo

getent group sudo

• The subject imposes strict rules for sudo. The student being evaluated must first explain the value and operation of sudo using examples of their choice. In a second step, it must show you the implementation of the rules imposed by the subject.

sudo visudo

Defaults log_input,log_output : Sudo ile yapılan işlemlerin girdilerinin ve çıktılarının loglarını tutması için

Defaults logfile=”/var/log/sudo/sudo.log” : Sudo işlemlerinin loglarının kaydedileceği yeri belirlemek için

Defaults requiretty : TTY modunu aktif etmek için

Defaults secure_path=”/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin” : PATH’in değiştirilme ihtimaline karşılık sadece bizim belirlediğimiz güvenli dizinlerden çalışması için

Defaults passwd_tries=3 : sudo için bir şifre istediğinde yanlış girilme durumunda en fazla kaç kez yanlış girme hakkı olduğunu ayarlamak için

Defaults badpass_message=”<belirttiğiniz hata mesajı>” : Hatalı şifre girildiğinde gösterilecek olan mesaj

• Verify that the "/var/log/sudo/" folder exists and has at least one file. Check the contents of the files in this folder, You should see a history of the commands used with sudo.

cd /var/log/sudo

ls

sudo cat sudo.log

Finally, try to run a command via sudo. See if the file (s) in the "/var/log/sudo/" folder have been updated. If something does not work as expected or is not clearly explained, the evaluation stops here

The sudo command allows a regular user to perform administrative tasks with elevated privileges temporarily, ***without requiring full root access.*** *It enhances system security by restricting access to critical operations while maintaining accountability through activity logs.*

Example of sudo cat:

To view the contents of a file that requires root privileges (e.g., /etc/shadow):

cat sudo.log

sudo cat sudo.log

UFW / Firewalld

Remember: Whenever you need help checking something, the student being evaluated should be able to help you.

• Check that the "UFW" (or "Firewalld" for rocky) program is properly installed on the virtual machine.

dpkg -l | grep ufw

• Check that it is working properly.

sudo nano /etc/ssh/sshd_config

• The student being evaluated should explain to you basically what UFW (or Firewalld) is and the value of using it.

UFW (Uncomplicated Firewall) and Firewalld are tools for managing firewall rules in Linux, providing a user-friendly way to control incoming and outgoing network traffic. They add an essential layer of security by allowing only authorized traffic to access your system while blocking potential threats. Using these tools ensures better system protection, simplifies rule configuration, and helps enforce network security policies.

• List the active rules in UFW (or Firewalld). A rule must exist for port 4242.

Settings > network > pathforwarding

• Add a new rule to open port 8080. Check that this one has been added by listing the active rules.

• Finally, delete this new rule with the help of the student being evaluated. If something does not work as expected or is not clearly explained, the evaluation stops here.

SSH

Remember:

Whenever you need help checking something, the student being evaluated should be able to help you.

• Check that the SSH service is properly installed on the virtual machine.

• Check that it is working properly.

• The student being evaluated must be able to explain to you basically what SSH is and the value of using it.

• Verify that the SSH service only uses port 4242 in the virtual machine.

• The student being evaluated should help you use SSH in order to log in with the newly created user. To do this, you can use a key or a simple password. It will depend on the student being evaluated. Of course, you have to make sure that you cannot use SSH with the "root" user as stated in the subject. If something does not work as expected or is not clearly explained, the evaluation stops here.

APT NEDİR?

APT (Advanced Package Tool), Debian tabanlı Linux sistemlerinde paket yönetimi için kullanılan bir araçtır.

Görevi: Yazılımları kolayca kurmak, güncellemek, kaldırmak ve yönetmek.

Komutlar:

apt install <paket> → Paket kurar.

apt update → Depo bilgilerini günceller.

apt upgrade → Sistemdeki tüm yazılımları günceller.

apt remove <paket> → Paketi kaldırır.

APT, arka planda dpkg aracını kullanarak çalışır ve bağımlılıkları otomatik yönetir. Bu, yazılım yüklemeyi hızlı ve sorunsuz hale getirir.

APTİTUDE NEDİR?

Gelişmiş Arayüz:

Konsol tabanlı bir menü sunar, böylece paketleri bir arayüz üzerinden görüp yönetebilirsiniz.

Komut satırı modunda da çalışır. APT'ye Göre Avantajları:

Bağımlılıkları daha iyi yönetir ve çözüm önerileri sunar.

Daha detaylı hata mesajları verir.

APT'nin yaptığı tüm işleri (kurma, güncelleme, kaldırma) yapabilir.

Kullanımı:

aptitude yerine apt gibi komutlarla kullanılabilir:

aptitude install <paket> → Paket yükler.

aptitude remove <paket> → Paket kaldırır.

aptitude search <paket> → Paket arar.

APT ve APTİDUDE ARASINDAKİ FARK

APT ve Aptitude, Debian tabanlı Linux sistemlerde kullanılan iki paket yönetim aracıdır. İşlevleri benzer olsa da, kullanım ve özellik açısından farklılıkları vardır:

1. Kullanım Farklılıkları

APT:

Daha basit ve hızlıdır.

Daha az kaynak kullanır.

Komut satırında sık kullanılan temel paket yönetimi işlevlerini gerçekleştirir.

Aptitude:

Konsol tabanlı bir arayüzü vardır (kullanıcı dostudur).

Bağımlılık sorunlarında daha iyi öneriler sunar.

Daha karmaşık komutlar ve özellikler içerir.

2. Bağımlılık Yönetimi

APT: Bağımlılıkları yükler, ancak bağımlılık sorunları ortaya çıkarsa çözümü size bırakır.

Aptitude: Bağımlılık sorunlarını analiz eder ve farklı çözüm yolları önerir.

3. Modernlik

APT: Debian ve Ubuntu gibi sistemlerde standart hale gelmiştir. Geliştiriciler APT'yi daha aktif günceller ve destekler.

Aptitude: Daha eski bir araçtır. Özellikle bağımlılık çözümü gerektiren karmaşık sistemlerde tercih edilebilir.

4. Komut Farklılıkları

Görev	APT	Aptitude:
Paket Yükleme	apt install paket	aptitude install paket

Paket Kaldırma	apt remove paket	aptitude remove paket
Paket Arama	apt search paket	aptitude search paket
Güncelleme	apt update && apt upgrade	aptitude update && aptitude upgrade

Hangisini Kullanmak Daha Avantajlı?

• APT: Daha hızlı ve yaygın olduğu için modern sistemlerde önerilir.

• Aptitude: Bağımlılık sorunlarıyla uğraşıyorsanız veya kullanıcı dostu bir arayüze ihtiyacınız varsa tercih edilebilir.

APPARMOR NEDİR?

AppArmor, Linux'ta çalışan uygulamalar için erişim kontrolü sağlayan bir güvenlik modülüdür.

• Ne yapar?: Uygulamaların yalnızca izin verilen kaynaklara erişmesini sağlar (örneğin dosyalar, ağ, sistem çağrıları).

• Nasıl çalışır?: Her uygulama için bir profil tanımlanır, bu profil uygulamanın yapabileceklerini sınırlar.

• Modlar:

Complain Mode: İhlalleri sadece loglar.

Enforce Mode: İhlalleri engeller.

Kullanım:

sudo aa-status: Profilleri görüntüler.

sudo systemctl restart apparmor: AppArmor'u yeniden başlatır.

Ubuntu gibi sistemlerde varsayılan olarak gelir ve SELinux'a daha kolay bir alternatiftir.

SELİNUX NEDİR?

SELinux (Security-Enhanced Linux), Linux çekirdeği için geliştirilmiş bir güvenlik modülüdür. Sistem güvenliğini artırmak için Zorunlu Erişim Kontrolü (MAC) mekanizması kullanır.

Ne Yapar?

Kullanıcıların, süreçlerin ve uygulamaların sistem kaynaklarına erişimini katı kurallarla sınırlar. Sistem yöneticisi, hangi işlemlerin hangi kaynaklara erişebileceğini belirler.

Özellikleri

1. Erişim Kontrolü: Dosyalara, ağ bağlantılarına ve sistem çağrılarına erişimi sınırlar.

2. Politika Tabanlı Yönetim: Detaylı güvenlik politikaları ile süreçleri izler ve denetler.

3. Modlar:

Enforcing: Kuralları uygular ve ihlalleri engeller.

Permissive: Kuralları loglar ama engellemez.

Disabled: SELinux devre dışıdır.

Kullanımı

• Durumu kontrol etmek: sestatus

• Mod değiştirmek:

sudo setenforce 0 # Permissive moda alır

sudo setenforce 1 # Enforcing moda alır

SELİNUX VE APPARMOR ARASINDAKİ FARK

SELinux ve AppArmor, Linux sistemlerinde güvenlik sağlamak için kullanılan iki farklı zorunlu erişim kontrolü (MAC) mekanizmasıdır. İşlevsel olarak benzerlikleri olsa da, uygulama yöntemleri ve kullanım kolaylığı açısından farklılık gösterirler.

1. Temel Farklar

Özellik	SELinux	AppArmor
Politika Türü	Etiket (Label) tabanlı	Yol (Path) tabanlı
Karmaşıklık	Daha karmaşık ve güçlü	Daha kullanıcı dostu ve basit
Kullanım Alanı	RedHat,CentOS,Fedoragibidağıtımlardavarsayılan	
	Ubuntu,Debian gibidağıtımlarda varsayılan	

Özellik SELinux AppArmor

Profil

Yönetimi Ayrıntılı, dinamik etiketleme Belirli dosya ve uygulama yollarına bağlı

Performans Biraz daha yüksek sistem yükü

oluşturabilir Daha hafif bir çözüm

2. Çalışma Yöntemleri

• SELinux:

o Sistem kaynaklarını etiketler (ör. dosyalar, süreçler) ve etiketler üzerinden erişim kurallarını uygular.

o Daha ayrıntılı ve esnek kontrol sağlar ama yapılandırması zordur.

• AppArmor:

o Uygulama veya dosya yollarına dayalı profiller kullanır.

o Kullanımı daha kolaydır, özellikle küçük ölçekli sistemlerde tercih edilir.

3. Modlar

Mod	SELinux	AppArmor
Zorlayıcı	"Enforcing"	"Enforce Mode"
İzin Verici	"Permissive"	"Complain Mode"
Devre Dışı	"Disabled"	"Disabled"

4. Yönetim Araçları

Görev	SELinux	AppArmor
Durum kontrolü	sestatus	aa-status
Görev	SELinux	AppArmor
Politika değiştirme	semanage	aa-complain, aa-enforce

Hangisi Daha İyi?

• SELinux:

o Büyük ve karmaşık sistemler için daha güçlü ve güvenlidir.

o Ancak yapılandırması daha zordur.

• AppArmor:

o Daha küçük sistemler ve kolay yönetim isteyen kullanıcılar için uygundur.

o Daha az karmaşık ve hızlı bir çözüm sunar.

SSHD NEDİR?

SSHD, SSH hizmetini sunan bir sunucu tarafı yazılımıdır. Tam adı "Secure Shell Daemon"dır. SSHD, gelen SSH bağlantı taleplerini dinler, bunları işler ve uygun kimlik doğrulama işlemlerini gerçekleştirir.

Özellikleri:

1. SSH bağlantıları için bir arka plan hizmeti olarak çalışır.
2. Genelde bir sunucuda başlatılır ve 22 numaralı portta istemcilerden gelen bağlantıları bekler.
3. Kullanıcı doğrulama, şifreleme, bağlantı yönetimi gibi işlemleri gerçekleştirir.

SSH VE SSHD ARASINDAKİ FARK

Özellik	SSH	SSHD
Amacı	SSH, istemci tarafıdır ve kullanıcılarınsunuculara bağlanmasını sağlar.	SSHD, sunucu tarafıdır ve bağlantıları kabul eder, yönetir.
Kim Kullanır?	Kullanıcılar, SSH istemcisiyle sunucuya bağlanır.	Sunucu, SSHD hizmetini çalıştırarak bağlantıları bekler.
Çalışma Şekli	Kullanıcı tarafından çalıştırılır.	Arka planda çalışan bir hizmettir (daemon).
Örnek Yazılımlar	OpenSSH istemcisi.	OpenSSH sunucusu (sshd).

KULLANILAN KOMUT TANIMLARI

dpkg -l | grep <kelime> : “dpkg” bir paket yöneticisidir. Bilgisayarda ki var olan dosyalarla ilgili işlem yapmamız için kullanırız. Bu komutu “-l” ile kullandığımızda tüm paketleri sıralamasını isteriz. “|” işareti “ile çalış” anlamına gelir. Bu işaretin yanına yazdığımız şey “dpkg -l” komutu ile çalışacak. “grep <kelime>” kısmı ise “dpkg -l” komutunun sadece belli bir kelimeye göre çalıştırılması anlamına gelir. Örn. : dpkg -l | grep ufw
Bu örnek komutun çıktısında içinde “ufw” geçen satırlar gözükür.

getent <veritabanı> <anahtar> : “getent” sistemdeki veritabanlarını sorgulamak için kullanılan bir komuttur. <veritabanı> ile belirtilen veritabanını ya komple ya da <anahtar> ile sorgular. Örn. : getent group

sudo

Bu örnek komutta “group” adlı veritabanında “sudo” grubunu aramasını istedim.

Eğer “sudo” diye bir grup varsa ekrana (varsa) içindeki veri ile yazılacak.

Örn. Çıktı: sudo:x:27:beergin

cut -d: -f1 /etc/passwd : Bu kod bir dosyanın içindeki verileri belli bir ayraca göre kesip bize verir. “cut” kısmı zaten adından da anlaşılacağı üzere kesmek için kullanılıyor. “-d” kısmı bir ayraç belirtmemiz için kullanılır. Bu ifadeden sonraki gelen ilk karakter ayraç olarak belirlenir. Biz ayraç için “:” kullanmışız. “-f” ifadesi sütun belirtmemiz için kullanılan bir ifade. Bu ifadeden sonra gelen ilk sayısal değer sütun olarak ele alınır. Kodda “1” yapmışız bu durumda ilk sütunu alıcak. Sonra gelen “etc/passwd” kısmı ise dizin. Tüm bu işlemlerin hangi dizinde uygulanacağını belirtiyoruz.

Yukardaki örnek için bakalım.

sudo:x:27:beergin

Ayracımız “:” idi. Kod uygulandığında ilk aşamada ayraçtan sonra gelen her kelimeyi bölüyor (ft_split yaptıysanız onun gibi). “-f1” kullandığımızda bu bölünenlerden ilki yani ilk sütunu alıyor. Bu stringin sütunlarını aşağıdaki gibi hayal edebilirsiniz:

sudo x 27 beergin

1 2 3 4

Koda göre bana “sudo” yu vericek. Eğer “-f2” olarak yapsaydım 2. Sütun olan “x” i verecekti. Olay bu kadar.

TTY: TTY, kullanıcı ile işletim sistemi arasında etkileşim sağlamak için kullanılır. Modern bilgisayarlarda genellikle komut satırı erişimi veya terminal arayüzü anlamına gelir.

TTY'nin İşlevleri ve Ne İşe Yaradığı

1. Kullanıcı Girişi Sağlama:

o TTY, kullanıcıların sisteme giriş yapmasını sağlar.

Örneğin, Linux'ta bir kullanıcı oturum açtığında, sistem bir TTY kullanır.

2. Komut Çalıştırma:
o Komut satırı üzerinden işletim sistemine talimatlar göndermek için kullanılır.
o Örneğin, ls, cd, veya mkdir gibi komutları çalıştırarak dosyaları listelemek, dizin değiştirmek veya yeni bir dizin oluşturmak mümkün olur.
3. Uzak Sunuculara Bağlantı (SSH ile):
o SSH kullanarak bir sunucuya bağlandığınızda, TTY benzeri bir ortam oluşturulur ve bu sayede sunucuya komut gönderilebilir.
4. Fiziksel Terminal Erişimi:
o Eğer bir sunucu veya bilgisayarın ekranı bağlı değilse, seri port üzerinden fiziksel TTY cihazları kullanılarak sisteme erişim sağlanabilir.
5. Sanal Konsollar (Virtual Consoles):
o Linux'ta Ctrl + Alt + F1-F6 gibi tuş kombinasyonlarıyla sanal terminaller arasında geçiş yapılabilir. Bu, bir kullanıcı komut satırı tabanlı bir oturumda çalışırken başka bir terminal açmasına olanak tanır.
6. Başka Kullanıcılarla Paralel Çalışma:
o Birden fazla kullanıcı farklı sanal TTY'lerde çalışabilir. Örneğin, bir kullanıcı tty1 üzerinde çalışırken, başka bir kullanıcı tty2 üzerinden çalışabilir.
7. Grafik Ortam ile Komut Satırı Geçişi:
o Linux sistemlerinde grafik arayüz (örneğin, GNOME, KDE) genellikle tty7 veya başka bir sanal terminal üzerinde çalışır. Kullanıcılar bu grafik oturumdan komut satırı tabanlı oturumlara geçiş yapabilir.
Monitoring.sh kodları:
arc=$(uname -a): İşletim sisteminin temel bilgilerini “arc” adında bir değişkende tutar.
pcpu=$(grep "physical id" /proc/cpuinfo | sort | uniq | wc -l):

grep "physical id" /proc/cpuinfo: Her işlemci fiziksel kimliğini /proc/cpuinfo dosyasından alır.
sort | uniq: Aynı "physical id" değerlerini birleştirir.
wc -l: Sonuçtaki satır sayısını sayar, yani fiziksel işlemci sayısını verir.
vcpu=$(grep "^processor" /proc/cpuinfo | wc -l):
grep "^processor" /proc/cpuinfo: Başında “processor” olan satırları sayar.
wc -l: İşlemci sayısını sayar ve sanal işlemci sayısını verir.
fram=$(free -m | awk '$1 == "Mem:" {print $2}'):
free -m: RAM kullanım bilgilerini MB cinsinden verir.
awk '$1 == "Mem:" {print $2}':
• awk komutu ile, free çıktısındaki ilk sütunda ($1) "Mem:" olan satır seçilir.
• Bu satırda, toplam RAM miktarı 2. sütunda ($2) yer alır ve bu değer yazdırılır.
uram=$(free -m | awk '$1 == "Mem:" {print $3}'):
awk '$1 == "Mem:" {print $3}':
• free çıktısındaki "Mem:" satırından kullanılan RAM miktarı 3. sütunda yer alır.
• Bu değeri alır ve yazdırır.
pram=$(free | awk '$1 == "Mem:" {printf("%.2f"), $3/$2*100}'):
awk '$1 == "Mem:" {printf("%.2f"), $3/$2*100}':
• awk komutu, "Mem:" satırındaki kullanılan RAM ($3) ve toplam RAM ($2) değerleriyle RAM kullanım oranını hesaplar.
• Hesaplanan oranı yüzde olarak yazdırır. %.2f ifadesi, sonucu iki ondalıklı şekilde yazdırır.
fdisk=$(df -Bg | grep '^/dev/' | grep -v '/boot$' | awk '{ft += $2} END {print ft}'):
df -Bg: Disk alanını GB cinsinden gösterir.
grep -v ‘/boot$’: Sonu “/boot” ile biten satırları görmezden gelir.
awk '{ft += $2} END {print ft}':
• awk ile her satırdaki 2. sütundaki (toplam disk alanı) değeri alır ve

biriktirir (ft += $2).

• END {print ft}: Tüm satırlar işlendikten sonra birikmiş toplam değeri yazdırır.

udisk=$(df -Bm | grep '^/dev/' | grep -v '/boot$' | awk '{ut += $3} END {print ut}'):

awk '{ut += $3} END {print ut}':

• Burada da, her satırdaki 3. sütunda bulunan (kullanılan disk alanı) değeri alır ve biriktirir (ut += $3).

• END {print ut}: Tüm satırlar işlendiğinde toplam değeri yazdırır.

pdisk=$(df -Bm | grep '^/dev/' | grep -v '/boot$' | awk '{ut += $3} {ft+= $2} END {printf("%d"), ut/ft*100}'):

awk '{ut += $3} {ft+= $2} END {printf("%d"), ut/ft*100}':

• Bu kısımda, her satırdaki 3. sütundaki (kullanılan disk) ve 2. sütundaki (toplam disk) değerleri toplanır.

• ut/ft*100 ile disk kullanım oranı hesaplanır ve yüzde olarak yazdırılır (%d tam sayı biçiminde).

cpul=$(top -bn1 | grep '^%Cpu' | cut -c 9- | xargs | awk '{printf("%.1f%%"), $1 + $3}'):

top -bn1

• top komutu, sistemin anlık performansını ve kaynak kullanımını gösteren bir komuttur. top komutunun çıktısı sürekli güncellenir.

• -b: Bu seçenek, top komutunu batch mode'a alır; yani sürekli güncellenmeden sadece tek bir kez çalışmasını sağlar.

• -n1: Bu seçenek, top komutunun yalnızca bir kez çalışmasını sağlar. Yani 1 defa çıktı alırız.

cut -c 9-

• cut komutu, metin üzerinde sütunlar kesmeye yarar.

• -c 9-: Bu seçenek, her satırın 9. karakterinden sonrasını alır. Çünkü, %Cpu satırında CPU kullanım yüzdeleri genellikle 9. karakterden başlar.

xargs

• xargs komutu, gelen veriyi tek bir satıra dönüştürür ve boşluklarla ayırır.
• Bu, cut komutundan gelen birden fazla değeri (örneğin, 7.2 us, 2.3 sy, 0.0 ni, 90.3 id) tek bir satırda birbirinden ayırarak alır.
awk '{printf("%.1f%%"), $1 + $3}'
• awk komutu, veriyi işlemenin güçlü bir yoludur. Burada $1 ve $3 değişkenleri, metni alanlar (fields) olarak temsil eder:
o $1: İlk sayıyı (kullanıcı CPU zamanı) alır, yani 7.2'yi.
o $3: Üçüncü sayıyı (sistem CPU zamanı) alır, yani 2.3'ü.
• $1 + $3: Kullanıcı ve sistem CPU kullanım yüzdelerini toplar.
o 7.2 + 2.3 = 9.5 (örnek olarak).
• printf("%.1f%%", $1 + $3): Bu işlem sonucu %9.5 formatında, bir ondalıklı sayı (örneğin 9.5%) çıktısı üretir.
lb=$(who -b | awk '$1 == "system" {print $3 " " $4}'):
awk '$1 == "system" {print $3 " " $4}':
• who -b komutundan gelen son başlatılma tarihi verilerini işler.
• awk ile "system" kelimesini içeren satır bulunur ve bu satırdaki 3. ve 4. sütunlar (tarih ve saat) yazdırılır.
lvmt=$(lsblk | grep "lvm" | wc -l):
lsblk komutu, sistemdeki tüm blok cihazlarını (diskler, bölümler, sürücüler vb.) listelemek için kullanılır. Çıktısı, fiziksel ve mantıksal diskleri ve onların sahip oldukları bölümleri (partitions) içerir.
grep "lvm": Bu komut, lsblk çıktısını tarar ve sadece LVM ile ilgili satırları arar. LVM genellikle disk adlarında lvm ifadesini içerir.
wc -l: grep'in çıktısını sayar ve bu satırların sayısını verir. Sonuç, LVM kullanılan disk sayısını sayısal olarak döndürecektir.
lvmu=$(if [$lvmt -eq 0]; then echo no; else echo yes; fi):
if [$lvmt -eq 0]: Bu, lvmt (LVM kullanımı sayısı) değişkeninin 0 olup olmadığını kontrol eder. Eğer 0 ise, yani sistemde LVM kullanılmıyorsa:
• echo no: LVM kullanılmıyor anlamına gelir.
Eğer lvmt değeri 0 değilse, yani sistemde LVM kullanılıyorsa:

• echo yes: LVM kullanılıyor anlamına gelir.
• fi: En başta yazdığımız “if” koşulunun bittiği anlamına gelir.
ctcp=$(cat /proc/net/sockstat{,6} | awk '$1 == "TCP:" {print $3}'):
cat /proc/net/sockstat{,6}
• /proc/net/sockstat dosyası, sistemdeki ağ bağlantıları hakkında istatistikleri içerir. Bu dosya, TCP ve UDP soket bağlantılarını, aktif bağlantı sayıları ve bağlantı türleri hakkında bilgi sağlar.
• /proc/net/sockstat6 ise IPv6 bağlantılarına dair aynı bilgileri içerir.
• {,6} kullanımı, hem /proc/net/sockstat hem de /proc/net/sockstat6 dosyalarını birden okur. Bu, hem IPv4 hem de IPv6 bağlantılarını bir arada almanızı sağlar.
| awk '$1 == "TCP:" {print $3}'
• |: Bu bir "pipe" komutudur ve önceki komutun çıktısını, bir sonraki komutun girdi olarak kullanmasına olanak tanır.
• awk '$1 == "TCP:" {print $3}': awk komutu, verileri satır satır işler ve belirli alanlara odaklanır. Buradaki awk komutu şu şekilde çalışır:
o $1 == "TCP:": awk komutu, her satırın ilk kolonunda ($1) "TCP:" ifadesi olup olmadığını kontrol eder. Bu, TCP bağlantılarının başlangıcını belirtir.
o {print $3}: Eğer bir satır "TCP:" ile başlıyorsa, o satırın 3. sütunundaki değeri ($3) yazdırır. Bu sütun, o anda aktif olan TCP bağlantı sayısını içerir.
ulog=$(users | wc -w):
Bu komut, sisteme giriş yapmış aktif kullanıcı sayısını hesaplar:
1. users: Sistemde oturum açmış kullanıcıları listeler.
2. wc -w: Listelenen kullanıcıları kelime bazında sayar, yani aktif kullanıcı sayısını verir.
Sonuç olarak, ulog değişkeni aktif kullanıcı sayısını tutar.
ip=$(hostname -I): Sistem IP adreslerini listeler.
mac=$(ip link show | awk '$1 == "link/ether" {print $2}'):
awk '$1 == "link/ether" {print $2}':

• ip link show komutundan ağ arayüzü bilgilerini alır.

• link/ether satırındaki 2. sütunda MAC adresi bulunur ve yazdırılır.

cmds=$(journalctl _COMM=sudo | grep COMMAND | wc -l):

Bu komut, sudo komutuyla çalıştırılmış komut sayısını bulur:

1. journalctl _COMM=sudo

• journalctl: Sistem günlüklerini (log) görüntülemek için kullanılan bir komuttur.

• _COMM=sudo: journalctl komutuna verilen bu filtre, sadece sudo komutuyla çalıştırılmış işlemleri gösterir. Yani, yalnızca sudo kullanılarak yapılan işlemler kaydedilecektir.

2. | grep COMMAND

• grep COMMAND: journalctl çıktısındaki satırlarda "COMMAND" kelimesini arar. Bu, sudo komutuyla çalıştırılan komutların başlık kısmındaki COMMAND anahtarını arar ve sadece bu satırları alır.

3. | wc -l

• wc -l: wc komutu, girdi olarak aldığı satır sayısını döndürür. Bu komut, grep tarafından filtrelenmiş satırları sayarak, sudo ile çalıştırılmış toplam komut sayısını verir.

wall "... mesaj içeriği ...":

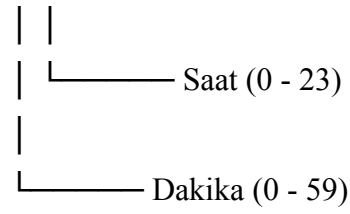
wall: Tüm terminallere bir mesaj yayınlar.

Yukarıda toplanan bilgileri formatlı bir şekilde yazdırır.

crontab komutu:

*/10 * * * * bash /usr/local/bin/monitoring.sh: Bu kod belirlediğimiz bir işlemi belirlediğimiz bir sürede çalıştırmayı sağlar. Kodun şeması şöyledir:

```
* * * * * komut
│ │ │ │ │
│ │ │ │ └──────── Hafta günü (0 - 6) (Pazar = 0 veya 7)
│ │ │ │
│ │ │ └──────── Ay (1 - 12)
│ │ │
│ │ └──────── Gün (1 - 31)
```



Bizim kullanımımızda şu şekilde çalışır:

/10: Bu kısım komutun 10 dakikada bir çalışmasını sağlar. Eğer başına “/” koymasaydık her 10 dakikada bir değil her saatin 10. Dakikasında çalışırdı.

: Eğer alanlara “” koyarsak bu o alanın görmezden gelineceğini belirtmiş oluruz.

NOT: 0 ve “*” aynı şey değildir. 0’ında burada bir işlevi var.

bash /usr/local/bin/monitoring.sh: Bu kısım ise belirlediğimiz süre geldiğinde çalışacak olan komutu temsil ediyor. Her 10 dakikada bir monitoring.sh dosyamız çalışacak.