Assignment 1

Name: Adnan Altukleh
Course: DV1466

Problem 1 (1 Point).

You own a GNU/Linux server that runs a user authentication system. The authentication system verifies user information and grants access to an external e-mail service. Your user authentication system uses single-factor authentication (i.e. username + password combination). The company providing the e-mail service wants you to update the authentication system to offer multi factor authentication (i.e. fingerprint, QR codes, facial recognition, SMS tokens, etc.). However, you do not have the resources (neither money nor infrastructure) to develop, maintain, and interconnect multiple custom authentication systems in-dependently. How may the GNU/Linux OS running in your server help you address this issue?

Solution:

We can implement multi-factor authentication using OpenSSH, a suite of secure networking tools based on the SSH (Secure Shell) protocol[1][2]. OpenSSH is distributed under a BSD-style license, which is a permissive free software license[3]. This solution enables multi-factor authentication by combining a password with public and private key authentication[4], improving security without any extra costs or infrastructure requirements.

1. Ylonen, T., & Lonvick, C. (2006). *RFC 4252: The Secure Shell (SSH) Authentication Protocol*. RFC Editor. https://www.rfc-editor.org/rfc/rfc4252.
2. Barrett, D. J., Silverman, R. E., and Byrnes, R. G. *SSH, The Secure Shell: The Definitive Guide*. 2nd ed., O'Reilly Media, Inc., 2005. Available at: https://learning.oreilly.com/library/view/ssh-the-secure/0596008953/ch01s03.html#sshtdg2-CHP-1-SECT-3.1.
3. OpenSSH. *OpenSSH is distributed under a BSD-style license*. https://www.openssh.com.
4. (2006). Authentication. In Pro OpenSSH (pp. 113-114). Apress. https://doi.org/10.1007/978-1-4302-0076-5_6

Problem 2 (1 Point).

You are the system administrator of a computing server, running GNU/Linux,
in a web development company. The company is currently executing three different projects in parallel. Each project has a dedicated team working on its development, and all of them require access to processing power and memory allocation from the computing server, simultaneously. There has been complaints that one team is starving the others by monopolizing the resources from the computing server. How can you monitor the computing requirements of each team and configure GNU/Linux to ensure that every team has fair access to the resources they need?

Solution:

In this scenario, the goal is to monitor and control the allocation of system resources (such as CPU and memory) to ensure that all teams have equitable access without manual intervention.

By using cgroup, it can create an interactive group limiting resources to N GB memory and N% of a CPU core[1]. We can use the (/etc/cgrules.conf) file which can be configured to automatically assign specific users or groups to this resource-limited cgroup[2].

1. Jain, S. M. Linux Containers and Virtualization, Chapter 4, APress, 2023. Available at: https://learning.oreilly.com/library/view/linux-containers-and/9781484297681/html/500466_2_En_4_Chapter.xhtml.
2. Ramesh, J. Linux Service Management Made Easy with systemd, Chapter 11, Packt Publishing, 2022. Available at: https://learning.oreilly.com/library/view/linux-service-management/9781801811644/B17491_11_Final_NM_ePub.xhtml#_idParaDest-150.

Problem 5 (1 Point).

A company uses a server to monitor and manage the network traffic generated by its employees. This server runs GNU/Linux. The company has discovered that some services employees use while working (such as streaming music or videos, or reviewing social media) can sometimes saturate the company's bandwidth, negatively affecting other services that are considered priorities (such as video conferencing or transferring large files). The company doesn't have the resources to improve their bandwidth or network speed. Also, the company doesn't want to ban the access to services employees commonly use. How can this issue be addressed from the server perspective?

Solution:

To solve the company's issue of bandwidth overload caused by non-essential services (such as streaming music, videos, or social media), a solution can be implemented through bandwidth prioritization. In GNU/Linux, the iproute2 package provides the tc (Traffic Control) command, which is specifically designed for traffic shaping[1][2].

By using tc, the company can allocate more bandwidth to essential services (such as video conferencing or file transfers) while limiting the bandwidth available for non-essential services, such as streaming music or videos[3].

1. SUSE. (n.d.). Iproute2 components: Traffic Control. SUSE. Retrieved from https://www.suse.com/c/iproute2-traffic-control/
2. Gheorghe, L. Designing and Implementing Linux Firewalls and QoS using netfilter, iproute2, NAT, and L7-filter, Chapter 3: iproute2 and Traffic Control. Packt Publishing, 2006. Available at: https://learning.oreilly.com/library/view/designing-and-implementing/9781904811657/ch03s02.html
3. Anastasi, G. F., Coppola, M., Dazzi, P., & Distefano, M. "QoS Guarantees for Network Bandwidth in Private Clouds." *Procedia Computer Science*, vol. 97, 2016, pp. 4-13. DOI: 10.1016/j.procs.2016.08.275.