**Valeria Mudzindiko**

**SAT 5424: Architecture Development Part 2: Installation, Configuration, and Security of OpenEMR**

**QUESTION B: Securing OpenEMR: Steps and Commands**

1. **Updating the System**: Regularly update and upgrade Ubuntu Server packages to keep

security patches up to date by using the following commands:

“ sudo apt-get update

sudo apt-get upgrade”

2. **Automatic Security Updates**: Enable automatic security updates with the `unattended-

upgrades` package to ensure timely application of security patches with the following command:

“ sudo apt-get install unattended-upgrades”

3. **Configuring a Firewall**: Use `ufw` to set up a firewall that allows only essential traffic

(HTTP, HTTPS, SSH) and blocks unnecessary access using the command:

“ sudo apt-get install ufw”

4. **Securing Apache**: Configure Apache&#39;s security settings by modifying the `security.conf`

file to disable unnecessary features and enable security headers that prevent web-based

Attacks with the following commands:

“ sudo ufw allow http

sudo ufw allow https

sudo ufw allow ssh”

5. **Restarting Apache**: After making changes to the configuration, restart Apache to apply

the security updates using the following commands:

“ sudo ufw enable”

“ ServerTokens Prod

ServerSignature Off

TraceEnable Off (You may leave this commented for diagnostics

purposes for this class)

Header set X-Content-Type-Options: &quot;nosniff&quot;

Header set X-Frame-Options: &quot;sameorigin&quot;

Header set X-XSS-Protection: &quot;1; mode=block&quot;

Header set X-Robots-Tag: &quot;none&quot;

Header set X-Download-Options: &quot;noopen&quot;

Header set X-Permitted-Cross-Domain-Policies: &quot;none&quot;”

“sudo a2enconf security”

6. **Using Strong Passwords**: Ensure that all user accounts in OpenEMR use strong, unique

passwords and consider enabling two-factor authentication for added security.

**QUESTION C: OpenEMR can still be susceptible to THE following types of cyberattacks:**

**SQL Injection** – If the platform does not properly sanitize user inputs, attackers could manipulate SQL queries to gain unauthorized access to the database, extract sensitive patient data, or modify records.

**Cross-Site Scripting** – Malicious scripts can be injected into OpenEMR’s web interface, allowing attackers to steal session cookies, redirect users to phishing sites, or perform unauthorized actions on behalf of a user.

**Cross-Site Request Forgery** – Attackers could trick authenticated users into performing unintended actions, such as modifying patient records or changing system settings, without their consent.

**Brute Force Attacks** – Weak or default credentials can make the system vulnerable to brute force attacks, where attackers repeatedly guess login credentials to gain unauthorized access.