Requirement 1: Install and Maintain a Firewall Configuration

1.1 Establish firewall and router configuration standards

- **Explanation**: Define rules for controlling inbound and outbound traffic, ensuring unauthorized access is restricted.
- **Example**: Set a policy that blocks all inbound traffic except HTTP/HTTPS (ports 80 and 443).

1.2 Build firewall configuration to protect CHD

- **Explanation**: Firewalls should segment the cardholder data environment (CDE) from other networks.
- **Example**: Use a firewall to isolate the database containing cardholder data from the corporate network.

1.3 Restrict connections between untrusted networks and the CDE

- **Explanation**: Limit traffic from external sources or other networks, ensuring only trusted systems communicate with CHD.
- **Example**: Only allow VPN access to the network segment handling card payments.

Requirement 2: Do Not Use Vendor-Supplied Defaults

2.1 Change all vendor-supplied defaults

- **Explanation**: Replace default passwords and settings in all systems to prevent unauthorized access.
- **Example**: Change the default credentials for the admin panel of a newly installed server.

2.2 Develop configuration standards for system hardening

- **Explanation**: Ensure systems are configured securely by default, disabling unnecessary services.
- **Example**: Remove services like FTP if not needed, and apply secure protocols such as SFTP.

Requirement 3: Protect Stored Data

3.1 Keep cardholder data storage to a minimum

- **Explanation**: Store only necessary CHD and securely discard it after use.
- **Example**: Use a tokenization service to store tokens instead of actual card numbers.

3.2 Do not store sensitive authentication data after authorization

- **Explanation**: Prohibit storing full magnetic stripe, CVV, or PIN block data after transaction approval.
- **Example**: Ensure your system discards CVV immediately after processing the transaction.

3.4 Render CHD unreadable wherever it is stored

- **Explanation**: Encrypt or tokenize stored CHD to make it unreadable to unauthorized users.
- **Example**: Use AES-256 encryption to secure stored credit card numbers.

Requirement 4: Encrypt Transmission of CHD Across Open Networks

4.1 Use strong cryptography to protect CHD in transit

- **Explanation**: Encrypt cardholder data whenever it's transmitted over public networks.
- **Example**: Use TLS 1.2 or higher to encrypt payment data sent between customers and your payment gateway.

4.2 Never send unencrypted PAN via email or messaging technologies

- **Explanation**: Do not send unencrypted CHD over email or messaging platforms like SMS.
- **Example**: Ensure employees use encrypted email systems or secure file transfer methods to share sensitive data.

Requirement 5: Use and Regularly Update Anti-Virus Software

5.1 Deploy anti-virus software on all systems commonly affected by malware

- **Explanation**: Install and maintain anti-virus solutions on all endpoints (e.g., workstations, servers).
- **Example**: Install anti-virus software on all Windows-based systems that handle payment data.

5.2 Ensure anti-virus programs are capable of generating audit logs

- **Explanation**: Configure anti-virus solutions to log events for later review.
- **Example**: Enable logging on your anti-virus software to track which files were scanned and quarantined.

Requirement 6: Develop and Maintain Secure Systems and Applications

6.1 Establish a process to identify security vulnerabilities

- **Explanation**: Regularly review vulnerability reports and apply patches promptly.
- **Example**: Subscribe to security bulletins for your software and apply patches within a month of release.

6.2 Develop software securely and test for vulnerabilities

- **Explanation**: Follow secure coding practices and test for vulnerabilities during development.
- **Example**: Use OWASP security guidelines to develop your Laravel API and conduct penetration testing before deployment.

6.3 Ensure that all web-facing applications are protected against known attacks

- **Explanation**: Implement security features such as WAF (Web Application Firewall) for applications facing the internet.
- **Example**: Use ModSecurity as a WAF for your web application to mitigate SQL injection attacks.

Requirement 7: Restrict Access to CHD by Need to Know

7.1 Limit access to CHD to only those whose job requires it

- **Explanation**: Ensure only specific roles have access to sensitive data, applying the Principle of Least Privilege.
- **Example**: Grant database access only to the database administrator, not the entire IT staff.

7.2 Control access via role-based access control (RBAC)

- **Explanation**: Define access levels based on job roles and responsibilities.
- **Example**: Use RBAC to ensure that customer service representatives can view only the last four digits of a credit card number.

Requirement 8: Assign a Unique ID to Each Person with Computer Access

8.1 Assign a unique ID to each user

- **Explanation**: Ensure that each person accessing the system has a unique identifier.
- **Example**: Require employees to log in with individual usernames and passwords, never shared accounts.

8.3 Implement multi-factor authentication (MFA) for access to the CDE

- **Explanation**: Require multiple factors (something you know, something you have) to verify identity.
- **Example**: Use a hardware token in addition to a password for access to sensitive systems.

Requirement 9: Restrict Physical Access to Cardholder Data

9.1 Use physical access controls

- **Explanation**: Implement keycard or biometric controls to restrict access to areas where CHD is stored.
- **Example**: Only authorized personnel can enter the server room using keycards or fingerprints.

9.4 Log access to sensitive areas

- **Explanation**: Keep logs of physical access to secure areas, such as server rooms.
- **Example**: Maintain an electronic access log of all personnel entering the data center.

Requirement 10: Track and Monitor All Access to Network Resources and CHD

10.1 Implement logging mechanisms to track user activity

- **Explanation**: Set up audit logs to track who accesses systems handling CHD.
- **Example**: Ensure that all database access is logged and retained for future analysis.

10.6 Review logs daily

- **Explanation**: Review security logs regularly to detect anomalies or unauthorized access.
- **Example**: Use SIEM (Security Information and Event Management) tools to analyze logs daily and detect suspicious activities.

Requirement 11: Regularly Test Security Systems and Processes

11.1 Perform quarterly vulnerability scans

- **Explanation**: Conduct vulnerability scans of all systems, especially those handling CHD.
- **Example**: Run quarterly vulnerability scans on all servers using tools like Nessus or Qualys.

11.3 Perform annual penetration testing

- **Explanation**: Test systems for vulnerabilities by simulating an attack.
- **Example**: Engage a third-party security firm to perform a penetration test on your network once a year.

Requirement 12: Maintain an Information Security Policy

12.1 Establish, publish, and maintain an information security policy

- **Explanation**: Create an InfoSec policy that covers how CHD is handled, updated regularly, and accessible to all relevant employees.
- **Example**: Your organization publishes an InfoSec policy on its intranet and reviews it annually to ensure compliance with the latest PCI standards.

12.6 Implement a security awareness program

- **Explanation**: Train employees on security best practices, including handling CHD and identifying potential threats.
- **Example**: Conduct annual security awareness training sessions for all employees who handle CHD.