Kong API Gateway Benchmark

19th November 2023

Document Version 1.0

Mohammed Alhamadah

Contents

Overview

This document delivers prescriptive guidance for establishing a secure configuration posture for the Kong API Gateway. It is intended to serve as a benchmark to ensure the security of Kong installation. This includes initial setup and hardening, securing communications and encryption, monitoring and logging, patch management, database security, and plugin security. The goal is to provide administrators with detailed steps to enhance the security of the Kong API Gateway and its associated data. This benchmark was created using Kong API Gateway version 3.5.

To acquire the most recent version of this guide, please visit https://appsec.sa/ If you have any queries, feedback, or have spotted ways to refine this guide, please contact us at info@appsec.sa.

Special Note: The configuration files mentioned in this benchmark may vary based on the deployment method and platform used. References to any configuration file should be tailored to the actual configuration files employed in the particular deployment.

Intended Audience

This document is designed for system and application administrators, security specialists, auditors, help desk professionals, and platform deployment experts who aim to design, launch, evaluate, or secure solutions that incorporate the Kong API Gateway.

1 Initial Setup and Hardening

1.1 Ensure admin API is not publicly accessible

Description:

Ensure that the administrative API endpoint of your Kong deployment is configured to be inaccessible from public networks. The Admin API is a powerful tool that grants comprehensive control and insights over the Kong deployment and should be restricted to trusted networks and individuals only.

Rationale:

Securing the Admin API is crucial because it can be used to configure routes, services, plugins, and other critical components of your Kong deployment. If malicious actors gain unauthorized access to the Admin API, they could severely compromise the confidentiality, integrity, and availability of your API services. Hence, limiting the access to Admin API to specific IP addresses or networks helps to enhance the security posture of your Kong deployment.

Audit:

1. Review the Kong configuration files, typically located at /etc/kong/kong.conf.
2. Confirm that the admin_listen directive is configured to bind to a private network interface or localhost (127.0.0.1).

```
admin_listen = 127.0.0.1:8001
```

Remediation:

1. Open the Kong configuration file (/etc/kong/kong.conf).
2. Modify the admin_listen directive to bind the Admin API to a private network interface or to the localhost.

```
admin_listen = 127.0.0.1:8001
```

3. Restart the Kong service to apply the changes.

```
systemctl restart kong
```

Default Value:

By default, the admin_listen directive might be configured to listen on all network interfaces (0.0.0.0). It is essential to change this to a more secure default, such as binding it to localhost (127.0.0.1).

References:

1. https://docs.konghq.com/gateway/latest/production/running-kong/secure-admin-api/
2. https://docs.konghq.com/gateway/
3. https://github.com/Kong/kong/

1.2 Ensure access to admin API is restricted when exposed beyond local host

Description:

Ensure that when the administrative API of your Kong deployment must be accessible beyond the localhost, proper restrictions are implemented to prevent unauthorized access. These restrictions might include IP whitelisting, implementing firewall rules, or configuring VPN access to safeguard the Admin API endpoint.

Rationale:

In certain scenarios, it becomes necessary to expose the Admin API beyond the localhost for administrative convenience or operational requirements. However, exposure beyond the localhost increases the risk of unauthorized access. Implementing appropriate access controls ensures that only trusted entities can access the Admin API, thereby protecting the Kong deployment from potential misuse and exploitation.

Audit:

1. Review the Kong configuration file (/etc/kong/kong.conf).
2. Confirm that appropriate access controls, such as IP whitelisting or VPN requirements, are implemented. Check for directives such as:

```
admin_listen = <private_IP>:8001
```

3. Validate the implemented restrictions by trying to access the Admin API from an IP address that should not have access.

Remediation:

1. Open the Kong configuration file (/etc/kong/kong.conf).
2. Bind the `admin_listen` directive to a private IP and set up access controls like IP whitelisting or VPN.

```
admin_listen = <private_IP>:8001
```

3. Implement firewall rules to allow access only from trusted networks or IP addresses.
4. Restart the Kong service to apply the changes:

```
systemctl restart kong
```

Default Value:

By default, the admin_listen directive might be configured to listen on all network interfaces (0.0.0.0).

References:

1. https://docs.konghq.com/gateway/latest/production/running-kong/secure-admin-api/
2. https://docs.konghq.com/gateway/
3. https://github.com/Kong/kong/

1.3 Ensure Kong Admin API Loopback is Configured

Description:

Ensure that the Kong Admin API loopback is configured correctly. This setup involves using Kong's routing design to have Kong serve as a proxy for the Admin API, allowing for finer access control over the Admin API while still making it available over specific controlled paths.

Rationale:

Configuring a loopback for the Kong Admin API through Kong's proxy capabilities enables a secure and controlled environment for accessing the Admin API. This arrangement safeguards the Admin API against unauthorized external access, whilst still allowing for necessary external communications through a controlled pathway, mitigating potential security risks.

Audit:

1. List services to confirm that a service is set up with the for the admin API.

```
curl http://127.0.0.1:8001/services
```

2. Check that a route is established for the `<adminAPIservice>` service with the appropriate path:

```
curl http://127.0.0.1:8001/services/<adminAPIservie>/routes
```

3. Test the setup by attempting to access the Kong Admin API through the established proxy path from an external system:

```
curl 127.0.0.1:8000/<adminAPIroute>/services
```

Remediation:

1. Create a new service for the admin API:

```
curl -X POST http://127.0.0.1:8001/services --data name=admin-api --data
host=127.0.0.1 --data port=8001
```

2. Set up a route for the new service:

```
curl -X POST http://127.0.0.1:8001/services/admin-api/routes --data
paths[]=/admin-api
```

3. Verify the setup by attempting to access the Kong Admin API through the newly established route:

```
curl myhost.dev:8000/admin-api/services
```

Default Value:

By default, the Kong Admin API might not be configured to operate over a loopback set up through a Kong proxy.

References:

1. https://docs.konghq.com/gateway/latest/production/running-kong/secure-admin-api/

1.4 Ensure Authentication is Configured for Kong Admin API

Description:

Ensure that robust authentication mechanisms are set up for the Kong Admin API. Adequate authentication configurations are vital to ensure only authorized users and applications have access to Kong's administrative functions.

Rationale:

The Kong Admin API provides a centralized interface for managing various aspects of the Kong API Gateway. Without proper authentication measures, malicious actors could potentially access and modify the Kong configuration, leading to unauthorized changes, data exposure, or service disruption.

Audit:

1. Access the Admin API through various endpoints with and without the required authentication parameters to verify if the configured authentication is functioning correctly:

```
curl 127.0.0.1:8000/admin-api/services
=> HTTP/1.1 401 Unauthorized
```

```
curl 127.0.0.1:8000/admin-api/services?apikey=secret
=> HTTP/1.1 200 OK
```

Remediation:

1. Configure Kong-specific security controls such key authentication, IP restrictions, or access control lists for the Admin API. Refer to the Kong documentation for guidance on setting up these security controls.
2. For Docker-hosted Kong Enterprise environments, use a declarative configuration to establish authentication as illustrated in the initial setup scripts provided. Ensure to tailor the key-auth plugin and consumer credentials to meet your specific security requirements.
3. Consider leveraging Kong's integration with Nginx for environments with complex security/access control requirements. This integration allows for the utilization of native Nginx authorization and authentication mechanisms, among other benefits.
4. Set up role-based access control (RBAC) to secure access to the Admin API, allowing fine-grained control over resource access based on a model of user roles and permissions.

Default Value:

By default, the Kong Admin GUI may not have authentication configured, leaving it open to access by unauthorized users.

References:

1. https://docs.konghq.com/gateway/
2. https://docs.konghq.com/gateway/latest/admin-api/
3. https://docs.konghq.com/gateway/latest/production/running-kong/secure-admin-api/

1.5 Ensure HTTPS/TLS is used for the admin API

Description:

Ensure that HTTPS/TLS is configured and utilized for the Kong Admin API to encrypt data in transit and protect it from interception and unauthorized access.

Rationale:

Utilizing HTTPS/TLS for the Admin API is a fundamental security measure that ensures confidentiality and integrity of the data transmitted over the network. It protects sensitive information from being intercepted and accessed by unauthorized entities, thereby reducing the risk of data breaches and other security incidents.

Audit:

1. Review the Kong configuration files to check if HTTPS/TLS has been enabled for the Admin API.
2. Try accessing the Admin API using a web browser or a tool like curl and verify if it is accessible only over HTTPS and not HTTP.

```
curl http://gateway:8001/
=> HTTP/1.1 200 OK
```

```
curl https://gateway:8443/
=> curl: (60) Issuer certificate is invalid
```

Default Value:

By default, HTTPS/TLS may not be enabled for the Kong Admin API, making it vulnerable to data interception and unauthorized access.

Remediation:

1. Configure the Admin API to use HTTPS by setting up TLS certificates and enforcing HTTPS connections.
2. Test the setup by attempting to access the Admin API over HTTPS to ensure it is working correctly and that HTTP requests are either redirected to HTTPS or return an error.

References:

1. https://docs.konghq.com/gateway/latest/
2. https://docs.konghq.com/gateway/latest/admin-api/

2 Secure Communication and Encryption

2.1 Ensure SSL/TLS certificates are valid and up to date

Description:

Ensure that the SSL/TLS certificates used in the Kong environment are valid and up-to-date to maintain the integrity and confidentiality of data transmitted over the network.

Rationale:

Using valid and up-to-date SSL/TLS certificates is essential to secure communications and to establish trust with clients by verifying that they are communicating with the intended server and that the data is encrypted.

Audit:

1. Regularly review the expiration dates of all SSL/TLS certificates in use to ensure they are current.
2. Verify the certificates' validity. The following command can be employed to verify the validity dates of the certificate for the gateway.

```
echo | openssl s_client -servername gateway -connect gateway:8443 2>/dev/null |
openssl x509 -noout -dates
```

Remediation:

1. Obtain new certificates from a reliable and recognized Certificate Authority (CA).
2. Install and configure the new certificates correctly in the Kong environment.
3. Set up a system for regular monitoring and alerting to be notified before certificates expire to avoid downtimes and security risks.

Default Value:

By default, SSL/TLS certificates have a defined validity period, and it is the responsibility of the system administrator to ensure they remain valid and up-to-date to maintain a secure environment.

References:

1. https://docs.konghq.com/gateway/latest/

2.2 Ensure Kong is using a strong SSL cipher suite

Description:

Ensure that Kong is configured to use a strong SSL cipher suite to facilitate secure communications by using robust encryption algorithms, thereby safeguarding data integrity and confidentiality during transmission over networks.

Rationale:

Utilizing a strong SSL cipher suite is pivotal in securing sensitive data from potential eavesdropping and man-in-the-middle attacks. It ensures that the encryption protocols in place are robust and resistant to known vulnerabilities, helping maintain the security posture of the Kong environment.

Audit:

1. Examine the `ssl_cipher_suite`, `ssl_ciphers`, and `ssl_protocols` settings in kong.conf configuration file.

```
ssl_cipher_suite = intermediate

ssl_ciphers =

ssl protocols = TLSv1.1 TLSv1.2 TLSv1.3
```

2. Ensure the returned value for `ssl_cipher_suite` is set to `intermediate` or another strong suite.
3. Confirm that the returned protocols are limited to secure versions such as `TLSv1.1`, `TLSv1.2`, and `TLSv1.3`.

Remediation:

1. Research and identify a strong and secure SSL cipher suite that is compliant with current best practices and industry standards.
2. Update the Kong configuration files to use the identified strong SSL cipher suite.

Default Value:

By default, Kong may be configured with a cipher suite that meets the security requirements at the time of its release. However, given the evolving nature of security threats, it is essential to regularly review and update the cipher suite to a strong one.

References:

1. https://docs.konghq.com/gateway/latest/

2.2 Ensure secure communication between Kong and its backend database.

Description:

Ensure that the communication between Kong and its backend database is secure and encrypted. This involves using secure protocols and trusted certificates for encrypted communication, ensuring database authentication credentials are safeguarded, and isolating the database in a protected network segment.

Rationale:

Kong requires a backend database to store its configuration data, which may include sensitive information, like API routes, consumers, plugins, and possibly more. Ensuring secure communication between Kong and its database will prevent eavesdropping, man-in-the-middle attacks, and unauthorized data access.

Audit:

1. Review the Kong configuration to check if TLS/SSL encryption is enabled for database connections. Look for configurations related to ssl, tls, or database connection strings with sslmode parameters.
2. Check for the presence and validity of SSL/TLS certificates used for encrypted communication between Kong and its database.
3. Examine network configurations to ensure the database is not publicly accessible and resides in a protected network segment.
4. Verify that database credentials used by Kong are stored securely and not hardcoded in plain text.

Remediation:

1. Configure Kong to use SSL/TLS encryption when connecting to its backend database. For databases like PostgreSQL, ensure sslmode is set to require or a stricter setting in the connection string.
2. Use trusted and valid SSL/TLS certificates for encrypted communication. Regularly renew these certificates and ensure they're from a trusted certificate authority.
3. Isolate the backend database in a protected network zone, ensuring it's only accessible by trusted entities.
4. Store database credentials in a secure manner, using tools like secret management systems or environment variables, avoiding hardcoding in configuration files.

References:

1. https://www.postgresql.org/docs/current/ssl-tcp.html
2. https://docs.konghq.com/gateway/latest/admin-api/#security

3 Monitoring and Logging

3.1 Ensure monitoring is configured for the Kong gateway

Description:

Ensure that comprehensive monitoring solutions are in place for the Kong Gateway. A proper monitoring setup provides insights into the system's health, performance, and potential security threats, allowing for timely detection and response.

Rationale:

Monitoring is essential for maintaining the availability, performance, and security of the Kong Gateway. With effective monitoring, administrators can detect and address issues proactively, reducing downtime and ensuring consistent service delivery.

Audit:

1. Fetch a list of all enabled plugins in Kong:

```
curl http://127.0.0.1:8001/plugins/enabled
```

2. Review the returned list to see if monitoring plugins, like `prometheus`, `datadog`, or others, are active.

Remediation:

1. If monitoring isn't configured, decide on a suitable monitoring solution that's compatible with Kong.
2. Activate and configure the chosen monitoring plugin for Kong Gateway, ensuring it captures key metrics.
3. Establish alert mechanisms for critical events or thresholds to guarantee a timely response to potential issues.

References:

1. https://docs.konghq.com/gateway/latest/production/logging/

3.2 Ensure logs don't contain sensitive information

Description:

Ensure that the Kong API gateway logs are sanitized and free from sensitive information. This includes but is not limited to credentials, tokens, personally identifiable information (PII), and credit card details. Storing such data in logs could make them a target for malicious actors and lead to data breaches.

Rationale:

Logs are invaluable for diagnostics, monitoring, and security analysis. However, unintentionally logging sensitive information poses a significant risk. Malicious actors can exploit these logs to access restricted areas of an application or even conduct identity theft. Sanitizing logs prevents exposing sensitive data and complies with data protection regulations.

Audit:

1. Review the Kong logging configuration in the kong.conf file and ensure that no detailed data logging is enabled in production environments.
2. Analyse a sample set of logs to verify the absence of sensitive data.
3. Use `grep` to search for typical sensitive keywords in your Kong logs, such as "password", "token", "key", etc.

```
grep -irnE 'username' /usr/local/kong/logs/*
grep -irnE 'password' /usr/local/kong/logs/*
grep -irnE 'key' /usr/local/kong/logs/*
grep -irnE 'token' /usr/local/kong/logs/*
grep -irnE 'authorization":"Basic' /usr/local/kong/logs/*
```

4. Validate that no plugins or custom logging mechanisms inadvertently log sensitive information.

Remediation:

1. Adjust Kong's logging configuration to a less detailed level.
2. Use Kong's transform plugin or similar functionalities to mask or remove sensitive information from logs.
3. For custom plugins or logging solutions, review and update the logging logic to ensure sensitive information is not logged.

3.3 Ensure Kong database logging is enabled

Description:

Ensure that logging is enabled for the Kong backend database. Logging provides valuable insights into database activity, can be used to detect suspicious or malicious activities, and is crucial for forensic analysis and auditing purposes.

Rationale:

Database logs can provide information about executed queries, login attempts, and other significant activities. Enabling logging helps administrators identify unauthorized access attempts, changes made to the data, or other anomalies, thus enhancing the overall security posture.

Audit:

1. Access the configuration settings of the Kong database.
2. Verify that logging is enabled and that essential activities, such as logins, queries, and transactions, are being logged.

Remediation:

1. If logging is not enabled, modify the database configuration to turn on logging.
2. Configure the logging level to capture essential details without overwhelming the storage or impacting performance.
3. Regularly review the logs to detect anomalies and ensure storage rotation to prevent space issues.

References:

1. https://docs.konghq.com/gateway/latest/configuration/#database
2. https://www.postgresql.org/docs/current/runtime-config-logging.html

4 Patch Management

4.1 Ensure Kong version is up-to-date and free from known vulnerabilities

Description:

Ensure that the Kong API gateway deployed is using the most recent version and that it is free from any known vulnerabilities. Regularly updating Kong is crucial as updates often include patches for security vulnerabilities and other critical bugs.

Rationale:

Running outdated versions of any software, including Kong, can expose the system to vulnerabilities that have already been identified and addressed in newer releases. Keeping Kong updated ensures that the system benefits from the latest security patches, bug fixes, and performance improvements.

Audit:

1. Check the currently running version of Kong with the command:

```
kong version
```

2. Compare the obtained version with the latest available version on the official Kong repository or Gateway Changelog
3. Use vulnerability databases such as CVE Details or NIST's National Vulnerability Database to check if the current Kong version has any known vulnerabilities.

Remediation:

1. If you're not using the latest version or if known vulnerabilities are found in your current version, plan an update strategy. Ensure to test the newer version in a staging environment first to verify compatibility with your setup.
2. Follow Kong's official upgrade documentation to safely upgrade to the newest version.
3. Regularly review and subscribe to official channels or mailing lists related to Kong for timely information about updates and patches.

References:

1. https://github.com/Kong/kong
2. https://docs.konghq.com/gateway/latest/
3. https://docs.konghq.com/gateway/changelog/
4. https://docs.konghq.com/gateway/latest/upgrade/

4.2 Ensure Kong database is updated

Description:

Ensure that the Kong backend database is regularly updated to the latest stable version. Keeping the database updated ensures that it benefits from the latest security patches, performance improvements, and feature enhancements.

Rationale:

Outdated databases might be vulnerable to known security issues that have been fixed in the newer releases. Regularly updating the database helps in reducing the attack surface and ensuring that the system remains resilient against known vulnerabilities.

Audit:

1. Check the current version of the Kong database.

```
postgres --version
```

2. Compare the installed version with the latest stable version available from the official database release notes or repository.

Remediation:

1. If the database is not updated, plan an update strategy considering potential downtimes, data migrations, and compatibility issues.
2. Before updating, take a full backup of the database and test the update process in a staging environment.
3. Regularly monitor the official database channels or websites for any new updates, patches, or security advisories.

References:

1. https://www.postgresql.org/docs/current/

4.3 Ensure plugins are updated and secure

Description:

Ensure that no outdated, unmaintained, or known vulnerable plugins are installed or active on the Kong API Gateway. Unmaintained plugins might not receive updates or security patches, while plugins with known vulnerabilities can expose the system to potential threats.

Rationale:

Relying on unmaintained or vulnerable plugins compromises the security and stability of the Kong environment. Such plugins may contain outdated code, dependencies, or vulnerabilities that can be exploited, thus introducing unnecessary risks.

Audit:

1. List all active plugins installed on the Kong API Gateway. Official Kong plugins will have the same version number as the installed Kong release.

```
curl http://127.0.0.1:8001/plugins/enabled
```

2. Identify the default plugins in Kong and the custom ones by saving the plugins names in a file and using this find_plugin_type.py script.

```
import requests
import sys

def get_plugin_type(file_path):
    try:
        with open(file_path, 'r') as file:
            plugins = file.readlines()

        for plugin in plugins:
            plugin_name = plugin.strip()
            if plugin_name:
                url = f"https://docs.konghq.com/hub/kong-inc/{plugin_name}"
                response = requests.get(url)
                if response.status_code == 200:
                    print(f"{plugin_name} default ({response.status_code})")
                else:
                    print(f"{plugin_name} custom ({response.status_code})")
    except FileNotFoundError:
        print(f"The file '{file_path}' does not exist.")
    except Exception as e:
        print(f"An error occurred: {e}")
```

```
if __name__ == "__main__":
    if len(sys.argv) != 2:
        print("Usage: python script.py plugins_names.txt")
    else:
        get_plugin_type(sys.argv[1])
```

Remediation:

1. For any plugin identified as unmaintained or vulnerable during the audit, consider its removal or replacement with a maintained and secure alternative.
2. If direct removal isn't feasible due to dependencies or other reasons, take measures to isolate or limit the plugin's functionalities to mitigate potential risks.
3. Implement a regular monitoring process for plugin updates, deprecations, and security advisories.

References:

1. https://docs.konghq.com/hub/

5 Database Security

5.1 Ensure Kong database is not publicly accessible

Description:

Ensure that the backend database used by Kong is not exposed to public networks and is only accessible by authorized services and personnel. Proper network configurations, firewalls, and access control measures should be in place to prevent unauthorized access.

Rationale:

Kong's backend database stores critical configuration data which, if compromised, can lead to significant security risks including unauthorized access to APIs, data breaches, and potential malfunction of the Kong gateway. Protecting the database from public access minimizes the surface area for potential attacks.

Audit:

1. Review network configurations to check if the Kong database port (e.g., 5432 for PostgreSQL) is exposed to public networks.
2. Use tools such as nmap to scan for open ports on the server hosting the database.
3. Check firewall rules and security group settings associated with the database server to ensure only trusted IPs or subnets have access.
4. Review database logs for any unexpected or unauthorized access attempts.

Remediation:

1. Modify network configurations to ensure that the database port is not exposed to public networks.
2. Set up firewalls or cloud security groups to restrict access to the database, allowing only trusted IP addresses or subnets.
3. Regularly review and update access rules to ensure that only necessary entities have access.
4. Monitor database logs for suspicious activities and set up alerts for unauthorized access attempts.

References:

1. https://www.postgresql.org/docs/current/auth-pg-hba-conf.html
2. https://docs.konghq.com/gateway/latest/admin-api/#security

5.2 Ensure strong credentials and appropriate roles/permissions are set for the database user

Description:

Ensure that the database user that Kong uses to connect to its backend database has strong credentials and is granted only the necessary roles and permissions required for operation. Limiting the privileges of the database user helps in reducing the potential impact in the event of a compromise.

Rationale:

Database users with weak credentials can be easily compromised, leading to unauthorized access. Furthermore, over-privileged database users can be exploited to perform actions beyond their intended scope, such as dropping tables, viewing sensitive information, or making unauthorized modifications.

Audit:

1. Review the database configuration in the kong.conf file and assess the strength of the password set for the Kong database user.

```
Pg_user = kong
Pg_password = password
Pg_datanase = kong
```

2. Ensure that the database user does not have administrative or superuser privileges unless necessary.

Remediation:

1. If the password is found to be weak, update the Kong database user's password to a strong, complex one.
2. Remove any superuser or administrative privileges from the Kong database user unless there's a specific need.

References:

1. https://www.postgresql.org/docs/current/user-manag.html
2. https://docs.konghq.com/gateway/latest/admin-api/#security

5.3 Ensure database backups are configured and functional

Description:

Ensure that backups for the Kong configurations and database are regularly scheduled, configured properly, and functional. These backups should capture the current state of the configuration and database and allow for the restoration of data in the event of data loss, corruption, or other disasters.

Rationale:

A properly configured backup strategy is essential for data durability and availability. Without regular and reliable backups, there's a risk of irreversible data loss, leading to potential service disruptions and loss of critical configuration and data.

Audit:

1. Review the backup configuration for the Kong database to ensure it's set up at regular intervals.
2. Look for running backup services such “rsync” and pg_dump.
3. Verify the location where backups are stored for redundancy and accessibility. Ideally, backups should be stored in geographically diverse locations.
4. Periodically perform a restoration test from a backup to ensure the backup's integrity and functionality.
5. Check the retention policy for backups, ensuring that backups are stored for an adequate amount of time based on the business needs and compliance requirements.

Remediation:

1. If backups are not configured, set up a regular backup schedule for the Kong database.
2. Ensure backups are stored in a secure location, preferably with encryption at rest.
3. Regularly test backups by performing restore operations in a test environment to verify their integrity.
4. Implement monitoring and alerts for backup failures or issues to ensure timely awareness and action.

References:

1. https://www.postgresql.org/docs/current/backup.html
2. https://docs.konghq.com/gateway/latest/admin-api/#backup-and-restore

5.4 Ensure database backups are encrypted

Description:

Ensure that backups for the Kong backend database are encrypted at rest. This ensures that the data, even if accessed without authorization, remains confidential and cannot be read without the appropriate decryption keys.

Rationale:

Encrypting database backups provides an additional layer of security against unauthorized access and potential data breaches. Without encryption, an attacker gaining access to backup files can easily restore and access sensitive information, which could include configurations, user data, and other critical information.

Audit:

1. Review the backup configuration for the Kong database to verify encryption at rest settings.
2. Check for the presence and secure management of encryption keys, ensuring they are stored separately from the backups.
3. Confirm the encryption algorithm used is considered strong and aligns with current best practices.

Remediation:

1. If backups are not encrypted, reconfigure the backup mechanism to enable encryption at rest.
2. Use strong and industry-recommended encryption algorithms like AES-256.
3. Store encryption keys securely, separate from the backups, and ensure regular rotation and secure management of these keys.

References:

1. https://www.postgresql.org/docs/current/backup-dump.html#BACKUP-DUMP-ENCRYPTION

5.5 Ensure no hardcoded secrets in Kong configuration

Description:

Ensure that there are no hardcoded secrets, such as API keys, passwords, or tokens, within the Kong Gateway configuration or any associated scripts. Using hardcoded secrets can expose systems to unnecessary security risks.

Rationale:

Hardcoded secrets present a security risk as they can be easily discovered by malicious actors, especially if the code or configuration is exposed or leaked. It's essential to use secure methods for secret management, such as environment variables or secret management tools, rather than embedding them directly in configurations or code.

Audit:

1. Review and scan the Kong configuration files for potential secrets:

```
grep -irnE 'username' /usr/local/kong/*
grep -irnE 'password' /usr/local/kong/*
grep -irnE 'key' /usr/local/kong/*
```

Remediation:

1. If hardcoded secrets are discovered, remove them from the Kong configuration or scripts.
2. Use secure methods to store and retrieve secrets, such as Secret Management tools like HashiCorp's Vault or AWS Secrets Manager.

References:

1. https://www.postgresql.org/docs/current/user-manag.html
2. https://docs.konghq.com/gateway/latest/admin-api/#security

6 Plugin Security

6.1 Ensure proper configuration of security-critical plugins

Description:

Ensure that plugins which have a critical impact on security, such as authentication, rate limiting, or logging plugins, are correctly configured on the Kong API Gateway. Proper configuration ensures that these plugins effectively enhance the security posture of the environment.

Rationale:

Misconfiguration or incomplete setup of security-critical plugins can lead to vulnerabilities, unauthorized access, or ineffectual security mechanisms. It is imperative to ascertain that these plugins function as intended and provide the required security measures.

Audit:

1. List all active plugins installed on the Kong API Gateway.

```
curl http://127.0.0.1:8001/plugins
```

2. Retrieve the schema and configuration settings of each plugin.

```
curl http://127.0.0.1:8001/plugins/schema/pluginName
```

3. Review the configuration settings of each plugin against best practices and recommended configurations.
4. Validate the effective functioning of these plugins in real-world scenarios (e.g., authentication attempts, rate limit thresholds).

Remediation:

1. For each security-critical plugin, adjust its settings to align with best practices and ensure optimum security. This might involve strengthening authentication processes or tightening rate limits.
2. Document any deviations from recommended configurations and justify them based on specific business or technical requirements.
3. Regularly review the configurations to accommodate changes in the threat landscape or business needs.

References:

1. https://docs.konghq.com/hub/

6.2 Ensure rate limiting is configured

Description:

Ensure that rate limiting is correctly configured for the Kong API Gateway. Rate limiting controls the number of API calls a user or IP can make within a specific time window, providing a mechanism to protect backend services from abuse or overwhelming traffic.

Rationale:

Rate limiting is crucial for maintaining the quality of service, preventing misuse, and safeguarding the backend services from potential DDoS attacks or other forms of abuse. It ensures that no single user or IP can monopolize the resources, leading to service degradation for other users.

Audit:

1. Check the Kong rate-limiting plugin is enabled.

```
curl http://127.0.0.1:8001/plugins/enabled
```

2. Retrieve the schema and configuration settings for the rate-limiting plugin.

```
curl http://127.0.0.1:8001/plugins/schema/rate-limiting
```

3. Verify the configured rate limits for various endpoints, users, or IPs to ensure they are reasonable and align with the service's capacity and

Remediation:

1. If rate limiting is not configured, enable the appropriate rate limiting plugin in Kong.
2. Define clear rate limits based on user roles, endpoints, or IP addresses.
3. Monitor the rate limits in real-time and adjust them based on usage trends and backend service capacity.

References:

1. https://docs.konghq.com/hub/kong-inc/rate-limiting/
2. https://docs.konghq.com/gateway/latest/admin-api/#rate-limiting-plugin-api