Infrastructure Endpoint Management

Guidelines Document

Version 1

Produced:
March 10th, 2023

Edited:
April 19, 2023

Written by:
Michael Goll - Expresia Full Stack Developer

Approved by:
Alex Phillips - Acting Expresia Technical Lead and Director of Infrastructure

Paula Amador - Chief Product Officer

1. Policy Statement

Expresia recognizes the critical importance of maintaining robust security measures for all endpoints within its network infrastructure. To that end, the Infrastructure Endpoint Management Guidelines document outlines the specific policies and procedures that must be followed in order to ensure that Expresia's public and non-public API endpoints, SSH server sessions of developers, and server, command, and administrative logging are secured to the highest possible standards.

2. Publicly Exposed Endpoints

A key feature of Expresia is the ability to implement publicly exposed endpoints[1] that allow end users to perform certain actions or functions within a given web application[2]. Some examples are signing up for an account, purchasing products or services, and downloading and viewing site content which includes articles within web pages and documents available for download. These endpoints are exposed to the web application built within Expresia that orchestrates incoming and outgoing data that can be viewed within the debug tools in most modern browsers.

Publicly exposed endpoints that handle and process data objects served by the Expresia API do not contain sensitive, non-public information such as credit card numbers or passwords.

3. Private Endpoints

Expresia also allows for the creation and use of non-publicly exposed endpoints[3] (or *private* endpoints) that are intended to only be accessed and used by a web application internally. These endpoints can provide any number of functions such as interfacing with external services (payment processors, user management systems, etc), and helper functions to break down complex operations into separate steps for easier management by developers.

These endpoints, and the data handled therein, are not publicly exposed. The only way to access these functions is via the web application's internal operations or by certified developers of the web application or Expresia. This allows developers to dictate exactly what is publicly available and what is not.

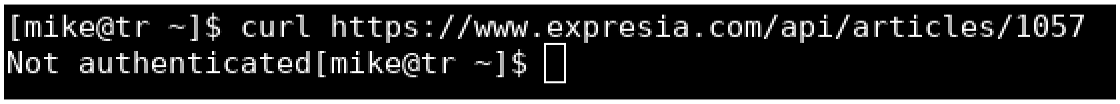
4. Request Security Credentials

Every request made to Expresia's internal API needs to specify a security code, otherwise known as an "access token". This allows Expresia to determine if the request made has valid authorization in order to request API operations.

[1] **Publicly Exposed Endpoint** is defined by an endpoint or URL that is openly accessible by a web application in order to orchestrate data transfer between the web application and the Expresia environment.
[2] **Web Application** is defined by the application (or website) built within the Expresia application and is separate from the internal processes, operations, and functions of the Expresia application itself.
[3] **Non-publicly Exposed Endpoint** is defined by an endpoint or URL that is not openly accessible by a web application. These endpoints typically only orchestrate data from internal systems of an application.

Any requests made to Expresia's API without the appropriate security headers are immediately denied and receive a "Not authenticated" response (see example below).

Furthermore, Expresia allows for the safeguarding of API endpoints developed within Expresia by requiring a specific, independent access token for authorization. Developers are able to create their own access token and implement additional security to endpoints their web application interfaces with. This access token is stored server-side and is not propagated publicly. This allows for the denying any and all requests that either omit or specify an incorrect access token.



```
[mike@tr ~]$ curl https://www.expresia.com/api/articles/1057
Not authenticated[mike@tr ~]$ 
```

Example: Showcasing an attempted API request omitting the required access token / security headers from an unauthorized source via the curl operation. This example specifies the request for an article object however the same security principles apply to all Expresia data objects.

5. Endpoint Security Best Practices

All Expresia developers are encouraged to follow industry standards and best practices when developing both publicly exposed API endpoints as well as non-publicly exposed API endpoints. These practices are both followed when creating API endpoints within the Expresia DXP application itself as well as any web application built within Expresia.

These best practices include:

- Never expose sensitive information within public API endpoints. This information includes:
  - Credit card numbers (including CVV numbers)
  - Passwords
  - Access Tokens
  - Private Keys
  - Encryption / Decryption Methods
- Safeguarding endpoints that handle sensitive data or operations with an access token and immediately deny any and all requests that do not contain correct access tokens.
  - If the endpoint does not need to be publicly accessible (ex. frontend administrative controls) then keep all operations in the Expresia backend only.
- Keep access tokens and private keys in server-side sources only.
- Manage / delete any and all endpoints that are no longer to be used in production.
- Update any and all endpoints that may contain any security flaws.
- Peer reviews of endpoints for security and operational concerns.
- Never perform data validation client-side except for minor tasks such as confirming existence of required input data.
  - Significant data validation methods should only be performed server-side.

These best practices mitigate security risks by not exposing sensitive information when in transit.

6. Remote Connections and Access Monitoring

Expresia acknowledges that remote access to administrative controls can be a severe security risk if it is not handled correctly. Exprsia has implemented security measures to ensure that unauthorized or unauthenticated sources are not allowed to access internal files of Expresia.

6.1. Network Access

Authorized internal Expresia developers are granted access to the internal servers for making modifications to the Expresia application. Access to the internal servers requires a valid, configured VPN file for establishing connection in addition to having login credentials to internal servers. Only authorized developers are allowed to create new user accounts on internal servers for development purposes. Passwords for server-side accounts are also to be generated in order to ensure password strength.

6.2. Server Access Logging

All internal Expresia servers are configured to create multiple log files that outline activity of Expresia developers. Internal Expresia servers create log files that track the following:

- General server logging
- SSH session logging
- System Authentication ("Auth") logging

6.3. General Server Logging

General server logs outline what is occurring on Expresia's internal servers. This includes errors that arise during program execution, as well as which authorization the program was executed by. This shows where the program is located in addition to all errors that caused issues.

6.4. SSH Session Logging

SSH session logging tracks which users are logging into the internal servers and when the login occurs. SSH session logging also tracks the commands that a user has given to the internal servers and can be traced back at a later time for review if needed.

6.5. System Authentication Logging

System authentication logging tracks what ascended-privileged operations are executed and which user executed them. This allows for a review of any and all operations that a developer has performed while using a heightened privileged level. Only authorized developers are allowed to perform such operations and require the Linux standard of using the **sudo** command in order to do so. Expresia developers are also encouraged to only use the sudo command when absolutely necessary and to exercise caution when doing so.

7. Endpoint Patch Management

Patch management for Expresia endpoints are an important part of maintaining a fully functioning system, to bring new functionality to Expresia, and address any issues that arise during Expresia's operation. All patches that are appointed to developers must undergo code reviews as well as testing before the submitted patch is approved and included in the newest Expresia versions henceforth.

7.1. Patch Priority

Expresia patches can be decreased and increased in priority based on the severity of the fixes the patch provides. Security patches are normally top priority to keep the system up to date and as protected as possible.

7.2. Code Reviews

Any and all changes made to any internal Expresia API endpoints must undergo code reviews. The current policy for these code reviews is as follows:

- All code changes must be approved by as many developers as possible
- Developers are not to approve their own work once submitted
- Once approved by the repository owner / appointee it can be merged with the rest of the codebase

7.3. Quality Assurance Testing

Submitted patches to the Expresia codebase are expected to have undergone numerous quality assurance tests developed by the submitting developer in order to prove that new, fixed, or improved functionality is working as intended or otherwise described.

Developers are expected to create appropriate test cases for the submitted patches that prove working functionality. Appropriate test cases include any conceivable edge cases that can affect operations as well as any and all scenarios that affect existing functionality.

Deployments of new Expresia patches are also tested and monitored after being officially deployed. Any deploys that experience issues or that do not perform as expected are either reverted or are supplemented by additional supporting patches to bring Expresia performance to the quality expected.

8. Device Management

Expresia emphasizes security on all devices that have access to the internal infrastructure and development tools. Most data and assets stored by Expresia are stored via Expresia's cloud and are not stored local to personal / development devices. Any information stored locally is expected to be password protected using strong passwords unless it is publicly available. This prevents issues in the event that a device is compromised that sensitive information is stored in the cloud and requires appropriate authentication credentials in order to access.

8.1. Data Storage

All employees are expected to not store sensitive information unencrypted on personal devices. Data and assets that are required for various instances built within Expresia are to be stored in the cloud that requires appropriate authentication in order to access if public availability does not apply.

8.2. Password Management

All Expresia team members are strongly encouraged to use trusted and powerful password managers in order to store passwords. These services can also be used to generate strong passwords that prevent duplication of passwords across various accounts to strengthen security. More information can be found in the **Expresia Application Security Standard**.

9. Incident Response

If a security incident does occur, tasks are created as quickly as possible in order to solve the incident. Damage assessments are performed and next steps are discussed immediately with all relevant Expresia team developers. The created tasks then receive the highest priority and are actioned immediately when a developer is available. If required, developers can also be redirected to assist with security focussed tasks.

This allows security-specific tasks to be actioned as fast as possible and resolve any incidents in a timely manner.

10. Continuous Improvement

Endpoint security is an ongoing battle that Expresia is no stranger to fighting. Our continuous improvement policy allows for the review and improvement of various endpoints in an attempt to prevent security issues.

One such policy is to review and investigate any and all endpoints for further security improvements. This includes running various security tests, monitoring of endpoints, and reporting of security issues straight away once they are discovered.

Expresia is dedicated to ensure that all internal systems are up to date and functioning as expected. If a security threat is discovered on any supported version then it is handled appropriately by the Expresia team as quickly as possible.