Application Security

Standards Document

Version 1

Produced:
March 10th, 2023

Edited:
April 19th, 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

To further strengthen the security of the Expresia application, it is crucial to maintain a strict policy for the management of passwords and roles & permissions. This policy ensures that user accounts are protected from unauthorized access and that users have access only to the resources necessary to perform their job functions. Expresia acknowledges that the proper management of passwords and roles & permissions is a critical aspect of its application security standards and therefore has implemented measures to ensure that these aspects of security are managed in a manner that is consistent with industry best practices. This policy applies to all users of the Expresia application and is aimed at maintaining the confidentiality, integrity, and availability of data and resources within the application.

2. Important Definitions

This document discusses the various parts of the Expresia user experience and configuration of user accounts and permissions for authorized access to an instance.

A key distinction between the **Expresia application** and the **Expresia instance** is as follows:

Expresia Application pertains to the operations, functions, and information that is handled by the overarching Expresia application that houses, at minimum, one Expresia instance. This includes the standardized core functionality of Expresia that cannot be manipulated from within an Expresia instance and is similar between multiple Expresia instances.

Expresia Instance pertains to the operations, functions, and information that is handled and displayed by the user-facing Expresia application available via the **/xpr** URL. The instance is housed within the Expresia application.

The use of the term **instance** throughout this document is implied as **Expresia Instance** unless otherwise specified.

3. Password Management Policy

Expresia's password management policy dictates that all users must hold at least one user account that must contain a strong password in order to be able to access the Expresia application.

3.1. Password Protection

All user accounts are authenticated through the use of strong passwords. All users must adhere to Expresia's minimum password strength requirements in order to assign a new password to a user account. The criteria for an acceptable Expresia password is defined as follows:

- Minimum of 8 characters in length
- Must contain at least one numeric character
- Must contain at least one special character

Instance developers are free to impose additional password requirements for their own system, however these requirements are the bare minimum for creation of instance user accounts.

Passwords stored within the Expresia application are encrypted using industry standard encryption algorithms. The resultant hashes are then stored server-side and are never decrypted from their hash forms. Hashes are compared and only correct passwords will result in a positive hash comparison. This allows all passwords to remain hashed / encrypted when at rest on Expresia servers.

3.2. Account Creation Activation

When an account is created external to the Expresia instance's account system, it is strongly recommended by Expresia to issue an account creation and activation email to the new account's email address to ensure that the email address owner can be verified.

User accounts created within the instance's backend user interface are activated immediately upon creation within the backend of Expresia.

3.3. Multi Factor Authentication

Expresia application accounts are able to enable optional multi-factor authentication. This process allows for the setup of the industry standard two factor authentication (2FA). Expresia's implementation of two factor authentication requires a user to scan a QR code using popular authentication mobile applications such as Google Authenticator in order to complete the login process.

Expresia strongly encourages all users to enable two factor authentication in order to prevent unauthorized access to the backend of the Expresia application.

4. Permissions Management Policy

User accounts within the Expresia application are subject to standardized account roles and allows for administrators to assign specific permissions to various roles. These permissions allow and disallow specific sets of users from accessing different parts of the instance in order to provide a streamlined experience for all user roles.

Expresia instances refer to the roles and permissions system as **User Groups**.

All user roles and permissions can be viewed by authorized roles and users under the **Users -> Groups** section of the instance.

4.1. Default User Roles and Permissions

The Expresia application offers various standardized roles that can be assigned freely to any user accounts created in the backend of Expresia. The default roles for all Expresia instances are:

- **Root** - A unique, specialized role that only is assigned to a singular account held by the Expresia team. This allows for access to Expresia instances for support services as well as the creation of additional user accounts. This role is limited to only the **master** branch of an instance and all Expresia team members are only to use this account when absolutely necessary.

- **Site Developer** - This role is intended to be assigned to any and all developers of a web application inside of the Expresia application. This role grants access to the entire instance for both the **development** and **master** branches.

- **Content Only** - This role is intended to be assigned to content management users for an instance. This role is only able to see content-specific features and functionality in order to create content for the instance.

These default roles can be edited by users with an authorized role. By default Expresia developers and support staff are set to the **Site Developer** role. This is required to be able to help with support cases and provide assistance to any part of the application as needed.

These user roles are only scoped to the access data, source code, assets, and user accounts of a web application housed inside of the instance via the user interface. Roles assigned here do not impact authentication of user accounts of Expresia's internal development team for developing the Expresia application itself. Those roles and permissions are handled externally from inside the Expresia application.

4.2. Custom User Roles and Permissions

The Expresia application allows for the creation of custom user roles. These roles can be freely created by the instance owner and can be associated with a number of different rules that dictate specific permissions. All changes to user roles are documented within the **Activity Log** that specifies user, time, and change(s) made.

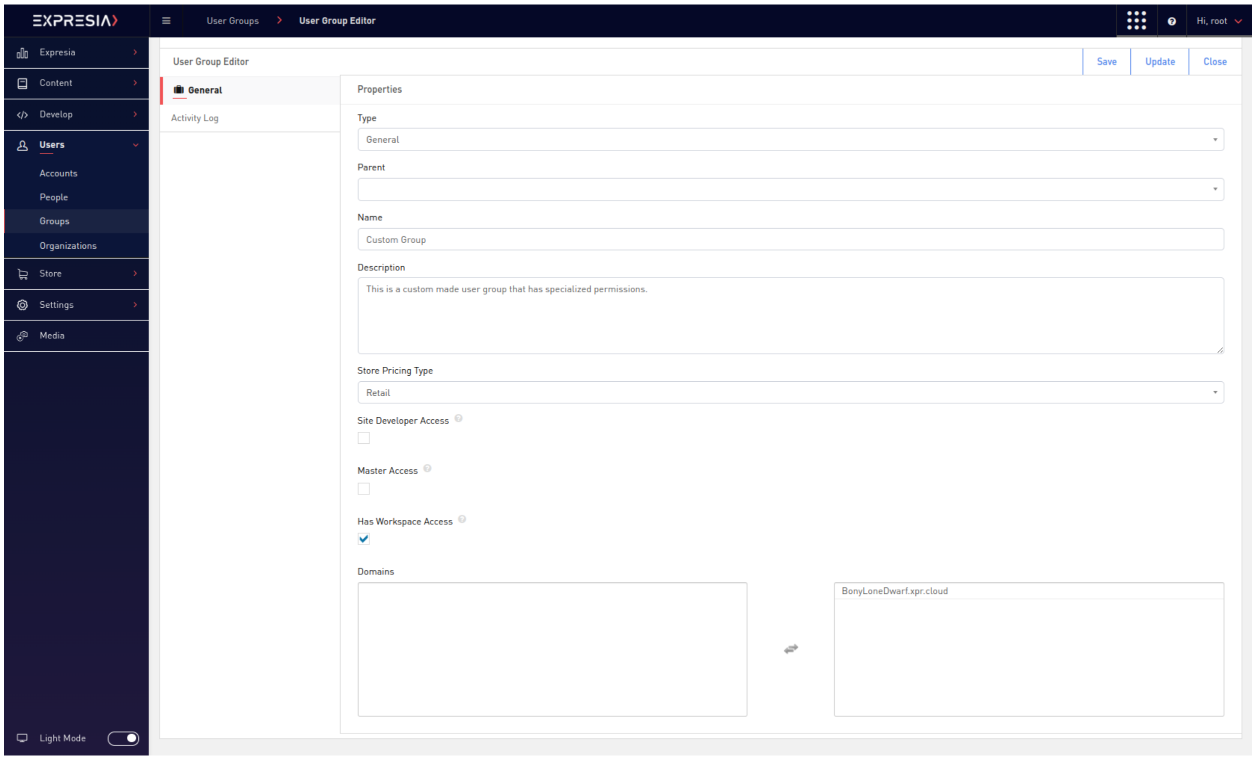
User roles have the ability to be grouped together by assigning a different user role as the "parent" role. Although being children of a specific parent role, all permissions are separate from the assigned parent role and operate independently. This functionality is available to provide effective categorization of roles.

An example of this categorization would be:

- Site Developer
    - Frontend Developer
    - Backend Developer
    - QA Tester

With this hierarchy, it would be possible to set different permissions for each level while keeping all children roles categorized to the parent **Site Developer** role.

User roles can also be active on specific domains as well if the instance has been configured for the use of multiple domains to allow for another level of granularity.



User Group Editor

Type: General

Parent:

Name: Custom Group

Description: This is a custom made user group that has specialized permissions.

Store Pricing Type: Retail

Site Developer Access: ☐

Master Access: ☐

Has Workspace Access: ✓

Domains: BonyLoneDwarf.xpr.cloud

Example: The user interface for the role creation system. This example showcases a custom made user role that offers limited access to the instance's backend systems and is available within all domains.