This application note will cover the changes to user authentication in the 2.9.0 release.

With an increased focus on network security and legislative changes regarding networked devices, Pharos has implemented changes to improve security, functionality and accessibility.

DESIGNER AUTHENTICATION

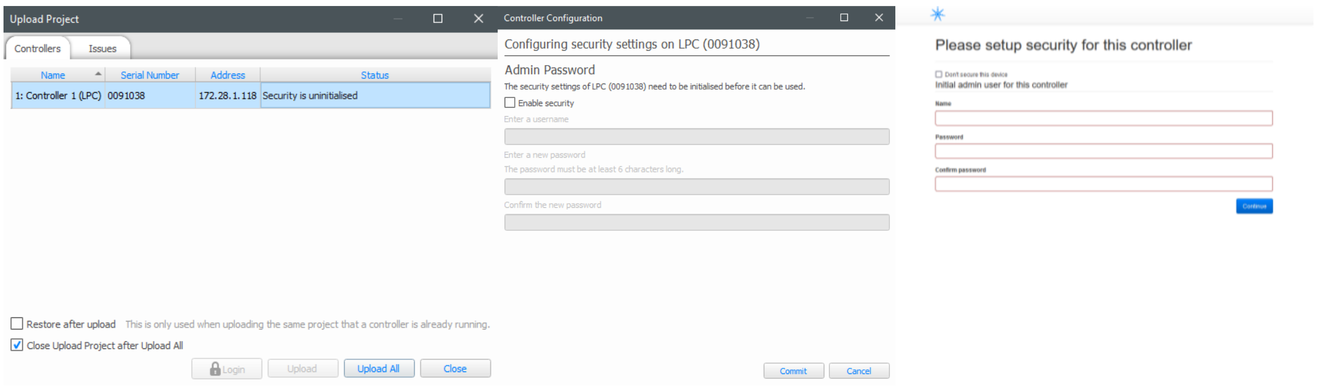
A summary of the changes to authentication in Designer v2.9.0 are as follows:

- New controllers require security choice before use
- Unification of controller and web interface user accounts
- Access levels for controller users
- Custom usernames and passwords
- Removal of session limit from web interface authentication
- Update of Controller Configuration window in Designer

New Controller Setup

A new controller requests at least one user account to be created before it can be used, either via Designer or via the web interface.

When a new controller is found in Designer, a pop-up will appear before the device can be used. This pop-up will ask the user to create an admin-level user account, with the option of turning security off (leaving the controller with no user accounts or password protection).


Upload Project

Name	Serial Number	Address	Status
1: Controller 1 (LPC)	0091038	172.28.1.118	Security is uninitialised

☐ Restore after upload This is only used when uploading the same project that a controller is already running.
☑ Close Upload Project after Upload All

Controller Configuration

Configuring security settings on LPC (0091038)

Admin Password

The security settings of LPC (0091038) need to be initialised before it can be used.

☐ Enable security

Enter a username

Enter a new password

The password must be at least 6 characters long.

Confirm the new password

Please setup security for this controller

☐ Dont secure this device

Initial admin user for this controller

Name

Password

Confirm password

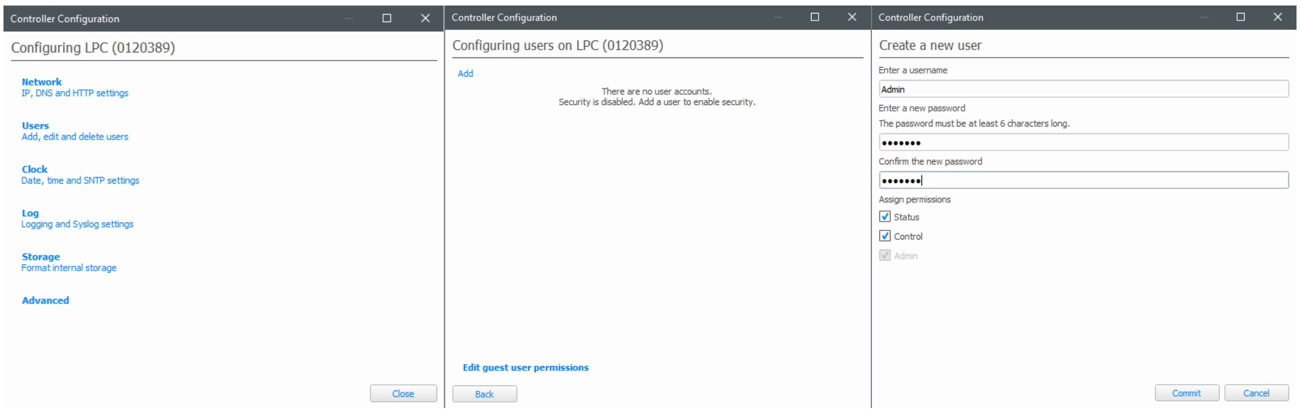
Fig1. When first: attempting to upload (left); opening the configuration window (centre); accessing the Web Interface (right) of a new or factory-reset controller

This initial admin account can then be managed, and more user accounts created, in the Controller Configuration window, which is covered in the next section of this application note.

If a controller is reset to factory defaults, this same process will be required again before the controller can be used.

Controller Configuration Window

The Controller Configuration window has a new, decluttered menu interface. Controller user accounts are edited within the **Users** menu item. User accounts control access to the controller via Designer and via the web interface.


Controller Configuration

Configuring LPC (0120389)

Network
IP, DNS and HTTP settings

Users
Add, edit and delete users

Clock
Date, time and SNTP settings

Log
Logging and Syslog settings

Storage
Format internal storage

Advanced

Controller Configuration

Configuring users on LPC (0120389)

There are no user accounts.
Security is disabled. Add a user to enable security.

Controller Configuration

Create a new user

Enter a username: Admin

Enter a new password
The password must be at least 6 characters long.
•••••••

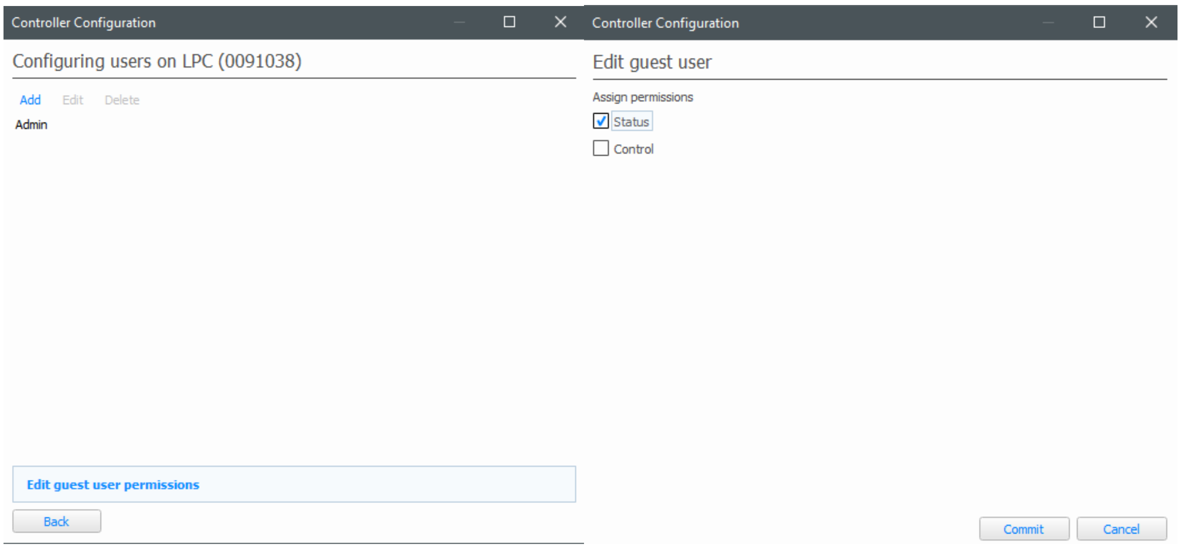
Confirm the new password
•••••••

Assign permissions
[✓ Status]
[✓ Control]
[✓ Admin]

Fig2. New Controller Configuration window (left), User Configuration window (centre) and New User window (right)

Create a new user account with a username and a password, and choose one or more permission groups. The permission groups are not cumulative – the Admin permission group does not necessarily grant the same access as the Status and Control groups. This separation will allow an expanded list of permission groups in a future release.

Click "Edit guest user permissions" to configure which sections of the web interface can be accessed without needing to log in – these will be accessible to everyone connected to the same network as the controller. This can be useful to allow any user access to the log or the project status, whilst protecting the ability to fire triggers or make changes to controller configuration.


Controller Configuration

Configuring users on LPC (0091038)

Admin

Controller Configuration

Edit guest user

Assign permissions
[✓ Status]
Control

Fig3. User Configuration window with "Edit guest user permissions" highlighted (left) and the Guest User Permission window (right)

Unification of Controller and Web Interface User Accounts

Before these changes were made in 2.9, there were two separate ways to secure a controller. Firstly, a **controller password**, which would secure the controller from alterations or uploads within Designer and the built-in web interface. Secondly, there was a **custom web interface user account system** that could be configured to enable granular permissions on the web interface for multiple users. These were both administered in different ways; the controller password was saved on the controller, and the web interface user accounts were saved in the project file. With 2.9, the two systems have been merged, so users can have granular permissions for the controller and the web interface, and all are administered directly on the controller – project files no longer contain any user account information.

Please note: in 2.9 we will no longer respect user accounts created within project files from earlier versions; user accounts must be (re-)created on the controller. The new, controller-based system will use the same permission groups as the old web interface system, whilst providing the option of creating up to 10 custom groups via the .webconfig (see below) or legacy .htaccess file. And, unlike the old system where user accounts were stored in the project file, controller user accounts will be wiped on a factory reset.

Designer 2.9 will display a warning when loading projects created in versions prior to 2.9 if custom web interface user accounts are stored in the project: *This project contains user accounts created in an earlier version of Designer for custom web interface access control. These user accounts are no longer supported - user accounts are now created directly on controllers. Please see the Help for more information*. These user accounts will be ignored if the project is uploaded to a controller but group access restrictions set within the custom web interface will still be honoured, so the custom web interface will remain secure assuming the same accounts are created on the controller with the same groups as before. The same is true for a controller upgraded to 2.9 with a project uploaded from an earlier version that contains custom web interface user accounts – the user accounts will be ignored but any group access restrictions will still be enforced.

Session Limit Lifted

Another change we have made is the method of authenticating a user on the controller web interface. Prior to 2.9, users were authenticated with a session token. This has been replaced by a JSON web token, which removes the need to store sessions on the controller and has removed the limitation of four active, authenticated sessions. Access is now only limited to the eight WebSockets we can maintain concurrently.

Replacing .htaccess with .webconfig

To setup authorisation for custom web interfaces, the use of .htaccess and groups files has been replaced in 2.9 by a single .webconfig file. Similar functionality is offered but it’s far more logical and simple to set up.

This will be covered more fully in the Designer Help upon release, but the general function of the .webconfig file is the ability to restrict access by permission group to folders within a custom web interface. The .webconfig file must be at the root level of the custom web interface.

For example:

```
[/admin]
AllowedGroups = Admin
LoginFile = login.html

[/timeline]
AllowedGroups = Control, Status
LoginFile = login.html

[/timeline/controls]
AllowedGroups = Control
LoginFile = login.html
```

In the above, only a user with the Admin permission group could access any page in the **/admin/** folder, but they could not access any pages in the **/timeline/** folder. Users with the Control or Status access permissions could access pages in the **/timeline/** folder, but only those with Control could access **/timeline/controls/**. **LoginFile** specifies the page that is loaded when a user tries to access a folder they don't have permission for. In this example, login.html might be written as follows:

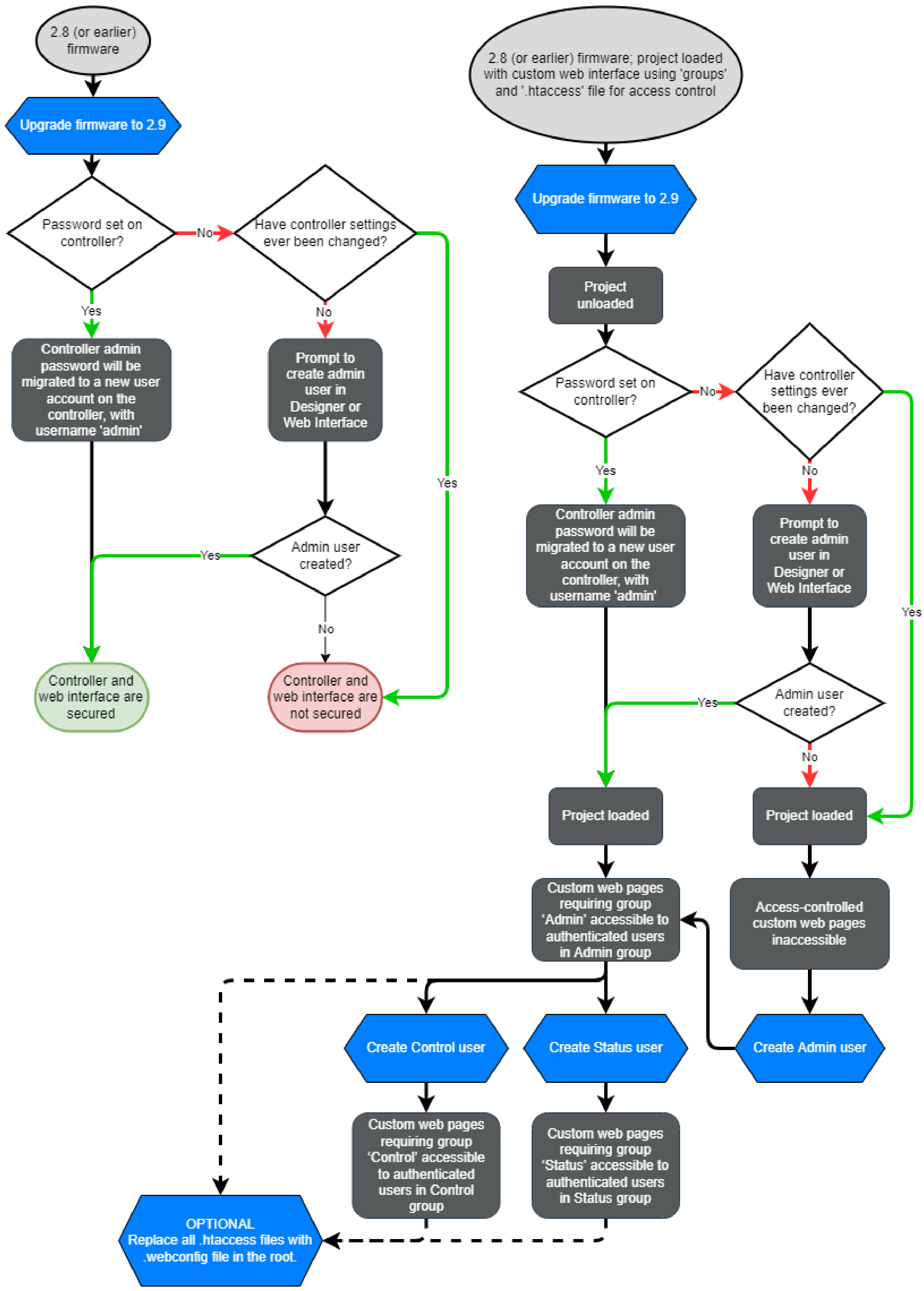
```
<html>
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-
scale=1, user-scalable=yes">
    </head>
    <body>
        <form action="/authenticate" method="POST">

            <input type="text" name="username" placeholder="Username">

            <input type="password" name="password" placeholder="Password">

            <button type="submit">Login</button>

        </form>
    </body>
</html>
```


2.8 (or earlier) firmware
Upgrade firmware to 2.9
Password set on controller?
No
Have controller settings ever been changed?
Yes
Controller admin password will be migrated to a new user account on the controller, with username 'admin'
No
Prompt to create admin user in Designer or Web Interface
Yes
Admin user created?
Yes
No
Controller and web interface are secured
Controller and web interface are not secured
2.8 (or earlier) firmware; project loaded with custom web interface using 'groups' and '.htaccess' file for access control
Upgrade firmware to 2.9
Project unloaded
Password set on controller?
No
Have controller settings ever been changed?
Yes
Controller admin password will be migrated to a new user account on the controller, with username 'admin'
No
Prompt to create admin user in Designer or Web Interface
Yes
Admin user created?
Yes
No
Project loaded
Project loaded
Custom web pages requiring group 'Admin' accessible to authenticated users in Admin group
Access-controlled custom web pages inaccessible
Create Control user
Create Status user
Create Admin user
Custom web pages requiring group 'Control' accessible to authenticated users in Control group
Custom web pages requiring group 'Status' accessible to authenticated users in Status group
OPTIONAL
Replace all .htaccess files with .webconfig file in the root.

v6.0 API

The key changes made in the API around authentication are:

- JWT tokens used for authentication instead of session IDs
- `username` instead of `user` in authentication payloads
- Endpoints `/token` and `/authorise` replaced with one `/authenticate` endpoint
  - If the `original_url` cookie or query parameter is passed in the POST request then `/authenticate` will behave in the same way as `/authorise` did in API v5 and earlier. If `original_url` is not present then `/authenticate` will behave as `/token` did before.
- .htaccess file has been replaced with the .webconfig file for custom web interface authorisation
  - similar functionality but easier to setup and manage
  - more information on the .webconfig file will be available in the Help on the release of v2.9.0
- query.js redirect handler
  - The query.js library now includes the function `set_redirect_handler()`. This can be passed as a callback, which will be invoked upon an unauthorised request. The callback can determine where to redirect the user. For example:
    - `Query.set_redirect_handler(() => {return "/login.html"})`
- API error messages are now returned as JSON, as an array of strings; API v5.0 returned error messages as text