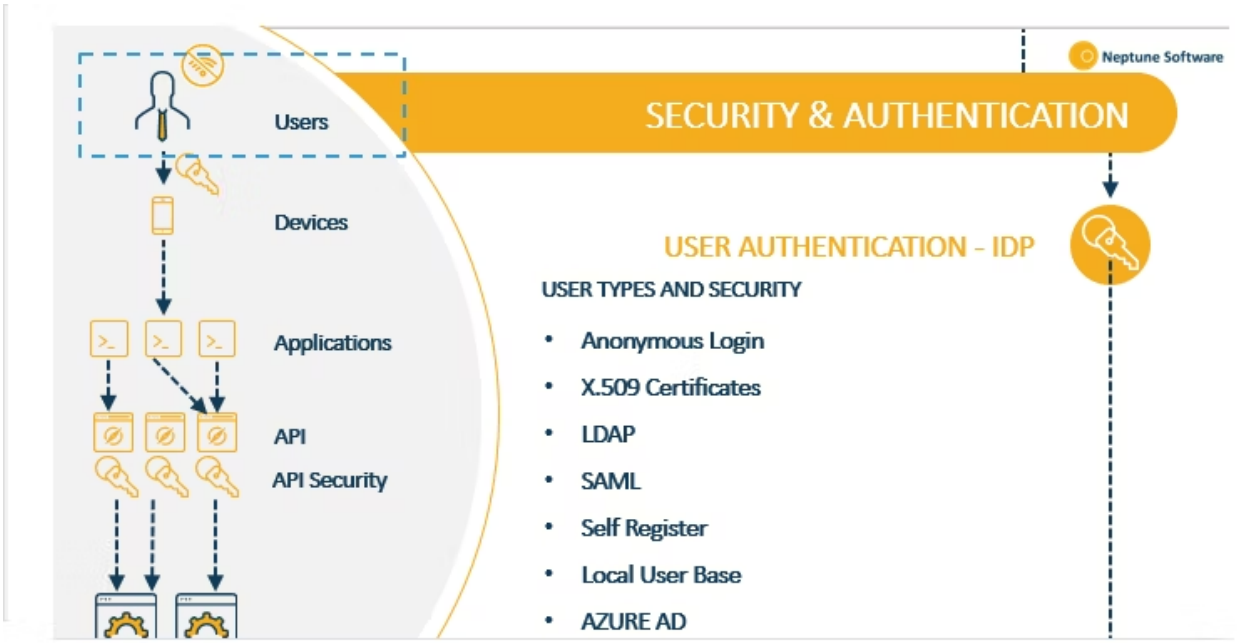
Neptune DXP - Open Edition - Security Process Overview

community.neptune-software.com/topics/neptune-dxp/blogs/planet-9-security-process-overview

Priyanka Jain



Neptune DXP - Security Features Overview

Security Features Overview

Planet 9 is a self-contained stateless nodejs server, which listens to a pre-configured network interface. It supports Linux, OSX, Windows and Raspberry Pi. It can be thought of as both an app development platform and an app run time. When a user creates an app, that app is served by the Planet 9 server. That is, the app is not a separate process - all apps are running through the same Planet 9 run time.

In a new installation, Planet 9 will start up with a local internal database and no other external dependencies. By default, it will bind to port 8080; however, this can be changed either through the settings page in Planet 9 itself, or by manually changing the config, or starting Planet 9 with the environment variable PORT set to the desired port (for instance in Linux, `PORT=3000 ./planet9-linux`).

It is possible to configure Planet 9 to run in HTTPS mode. For this, add certificates to the Cockpit ->Settings and configure the desired port (by default, this is port 8081, and can be overridden with an SSL_PORT environment variable)

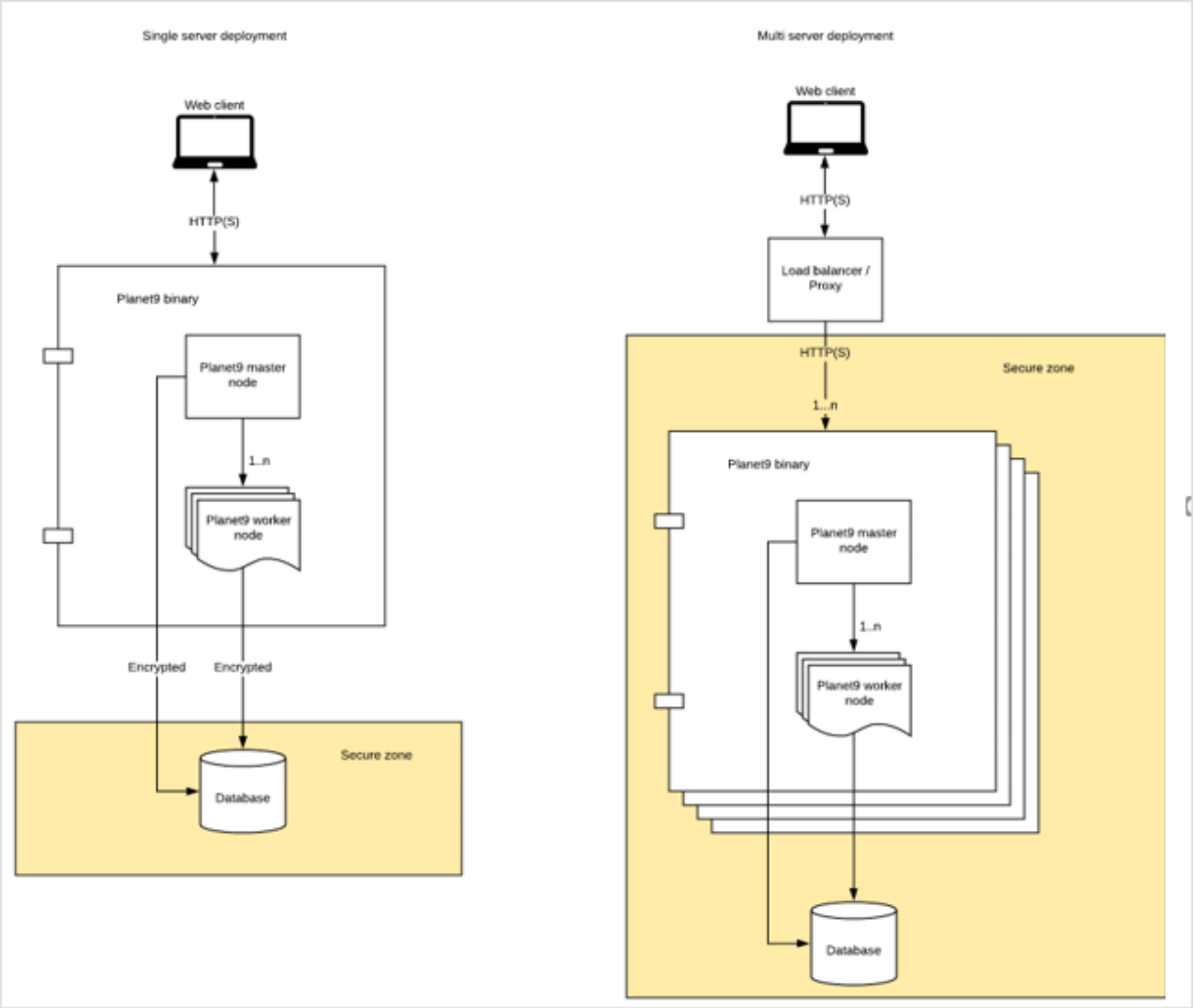
If Planet 9 running in HTTPS mode, an internal process will still bind to the configured HTTP port, but only listen to the loopback interface (127.0.0.1). This is used to run internal server script processes.

Load Balancing

However, in a production environment, it is recommended that you run Planet 9 behind a proxy such as 'nginx' or 'HAProxy', as binding to the default HTTPS port requires root permissions. That way, you do not have to run Planet 9 with a privileged account.

It is recommended that you switch from the default SQLite database to a specialised external database such as PostgreSQL. This database should preferably not be accessible to the general internet, just the Planet 9 installations.

More on Load Balancing HERE



Scaling within one server

To utilise the many cores computers usually have, Planet 9 has a master/worker architecture where you can spawn many Planet 9 workers. They all share the same TCP port and are stateless, so scaling these up and down is not- problematic. If a worker process should die, it responds immediately; this is by design.

Scaling across servers

In a high volume environment, more than one server could be needed. From a Planet 9 standpoint, this is not -problematic since the nodes themselves do not have to sync between each other. Planet 9 runs fine in docker and can be set up an orchestration tool such as Kubernetes. In this case, pods can be set up to run Planet 9 and can be scaled up and down depending on load. For security, it is recommended that they run on an internal network and only communicate with the load balancer. To make sure that the node is actually up, you can add a health check to the /healthz endpoint.

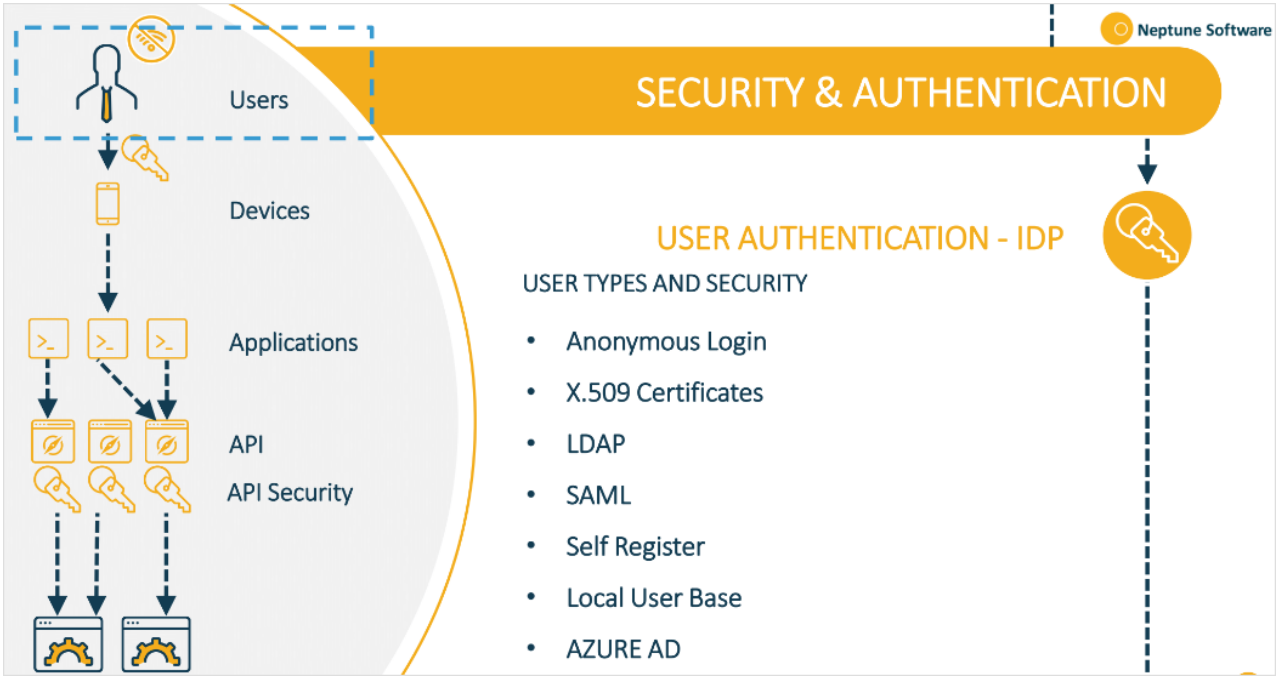
App Servers and System Logs

Check the logs for different systems; Add systems in App Servers under Settings, and then you can check Logs under Monitor -> System Logs.

User Authentication

By default, a user will have its credentials stored in Planet 9 itself (using bcrypt). There are a number of password policies that can be configured, such as minimum password length or time-based expiration. This is convenient if you don't have a lot of users.

However, for large organisations, Planet 9 supports multiples Users Security and Authentication Processes, including certification-based authentication like X.509 certification, cloud-based authentication like Azure AD and SAML based authorisation and authorisation.



Anonymous Login

The Anonymous authentication provider allows users to log in to your application without providing credentials. Each time someone authenticates anonymously, the provider generates a new anonymous user object for that session.

Enable Anonymous access for the Launchpad from Launchpad settings and enable Anonymous users settings in App Designer.

X.509 Certificates

Data security is an essential aspect of every modern data platform. Micro-service based architectures becoming more of a common pattern across every high-scale app. Existing password-based authentication mechanisms for user authentication is hard to manage at scale, let alone the fact that passwords are hard to remember or could be cracked easily.

Planet 9 is like a cloud connector that can generate X.509 certificate, which can be used by STRUST for generating identity authentication on SAP Cloud or SAP on-Premise.

Read the guide to set up X.509 Certificate-based authentication in Planet 9.

https://community.neptune-software.com/documentation/certificates

LDAP

Mostly, In a large organisation, an external authentication service such as LDAP makes more sense. In this case, when a user logs in, we query the LDAP server, and if the result is a success, we log the user in and add the groups to the user returned from LDAP, if any. Planet 9 does not store any passwords if an external authentication source is used. If you are using an external system, it should have SSL enabled and preferably not be a public-facing service. Note that the connection made to the system is done by Planet 9, not the client.

Read how to configure LDAP connection by Planet 9

SAML

Most organisations already know the identity of users because they are logged in to their Active Directory domain or intranet. It makes sense to use this information to log users into other applications, such as web-based applications, and one of the more elegant ways of doing this is by using SAML.

Planet 9 Supports SAML Authentication and Authorization. SAML is a standard authentication process for logging users into applications based on their sessions in another context.

Learn how to Enable SAML and Single Sign-On in Planet 9.

Self Register

Self Registration is the process of allowing users to give access to create their own account. Planet 9 supports this process also. Read how to enable Self Registration from Launchpad Settings.

Local User

Sometimes an organisation requires to generate role-based local users to access the Planet 9 cockpit, Launchpad and applications. You can generate local user and can assign roles to them, which can maintain the authentication and authorisation Planet 9 user management.

Roles are what define what a user has access to; a role can be assigned directly to a user or a group. The most permissive role always wins. That means that if a user is part of two groups, and group A only has read access to an operation, but group B has to write access, then the user will have write access. Read how to add Local User in Planet 9.

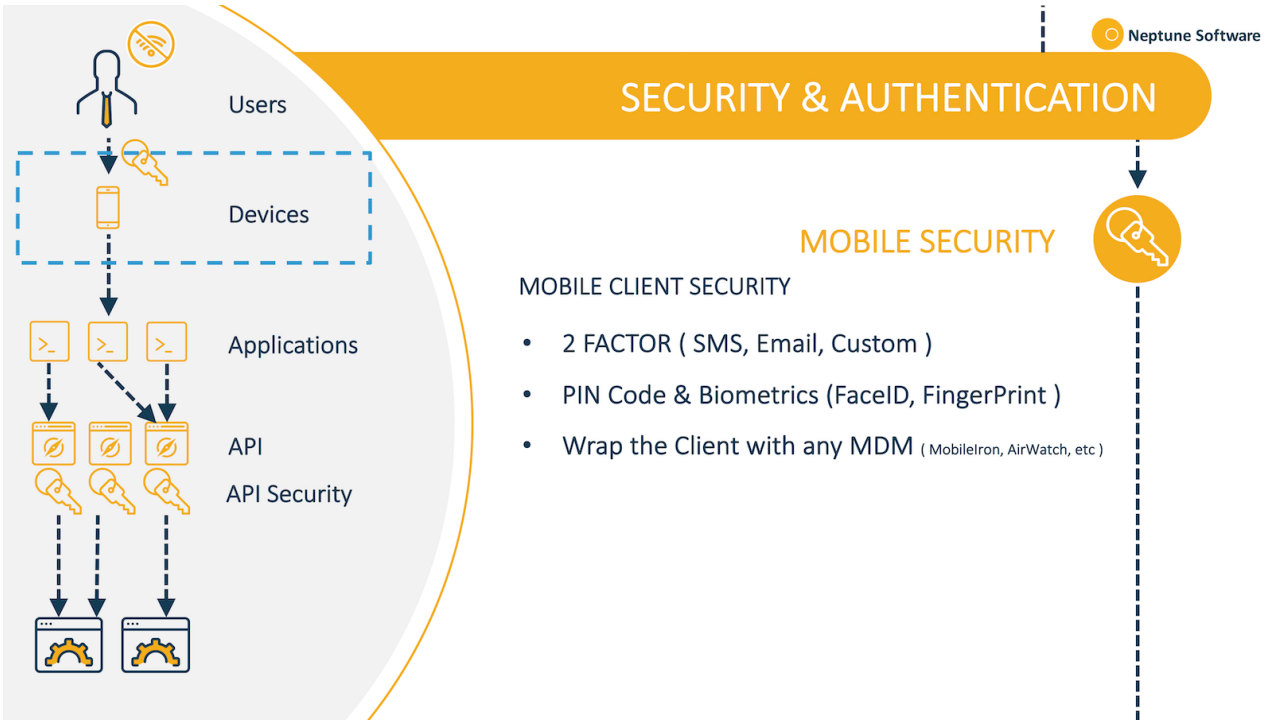
Azure AD

Currently, Azure AD authentication is one of the most popular methods. Large organisations prefer to set up Azure AD Authentication, and planet 9 fully supports this.

Read how to setup Azure AD Bearer Authentication in Planet 9.

Mobile Client Security

As Neptune Planet 9 is used for desktop as well as mobile-based applications. And Mobile application required mobile clients to publish the apps. Keeping this in consideration, Neptune supports multiple ways to provide security and authentication for mobile clients.



2-Factor Authentication(SMS, Email)

To support Mobile client Security Planet 9 is capable of setting 2 Factor Authentication through SMS and Email like enabling Pin Code Setup.

SMS Planet 9 can be configured for Firebase Push Notifications

Email: Planet 9 can be configured to send out emails. This requires that you set an SMTP server in the Cockpit - > Settings. Again, this should use TLS to make sure the password is not sent unencrypted.

Pin code Biometrics(Face ID, FingerPrint)

Planet 9 supports Pin Code as well as Face ID and Finger Point authentication for iOS and Android Mobile Devices.

Enable Pin code and set up Biometrics authentication in Planet 9 from Cockpit -> Run -> Mobile Client -> Authentication

Wrap the mobile client with MDM

The application binary files resulting from packaging with Cordova CLI or Build Phonegap can then be further secured by wrapping with a Mobile Device Management solution such as Mobile Iron, InTune, AirWatch, etc.

API Authentication

APIs handle enormous amounts of data, and securing API Data is always one of the primary concern. At some point, your APIs will need to allow limited access to users, servers, or servers on behalf of users.

Planet 9 supports many API authorisation methods, i.e. Protocol-based, certification based methods and Role-based methods(Auditing).

Auditing

When a user creates, update or delete an entity, that operation is added to the audit log, along with the timestamp, username and contents of the operation.

The audit log interface is read-only in Planet 9, and should only be added to users that need this kind of functionality.

oAuth 2.0

By using the oAuth 2.0 authorisation framework, you can give your own applications limited access to your APIs on behalf of the application itself.

Basic

Enable basic API Authentication by simply providing a usernameand password to prove their authentication.

SAP Principal Propagation

Enable SAP Principal Propagation to set up API Authentication with SAP Cloud or SAP on-Premise.

Tokens - JWT

If you don't want to use any of the authentication type defined above, Planet 9 gives an option to enable token-based authorisation to enable through JWT Bearer token

Certificates

Certificate-based authentication refers to the two parties authenticating each other through verifying the provided digital certificate so that both parties are assured of the others' identity. In technology terms, it refers to a client (web browser or client application) authenticating themselves to a server (website or server application) and that server also authenticating itself to the client through verifying the public key certificate/digital certificate issued by the trusted Certificate Authorities (CAs).

In Planet 9, you can generate a self-signed certificate to access the authorised resource or import the certificate from the authorised resource.

Customised Authentication

Planet 9 supports the ability to extend the functionality by writing your own JavaScript, in addition to including your own node modules, care must be taken when accessing external systems and what is exposed through the API.

For instance, if a script accesses another database decrypted, an external user might be able to pick up the username and password, if that is part of the connection string.

In general, having access to server scripts gives you a lot of power and should thus be given only to users that need it.