Security

Security and compliance features that is built in to the platform

- Bruteforce
- Compliance built-in
- Compliance external
- Data restrictions
- Encryption
- Security baseline
- Security built-in
- Security external
- Single sign-on

Bruteforce

In order to prevent bruteforce attacks on passwords to meaures are implemented

- Maximum number of retries for passwords
- Detection of spread attacks across multiple accounts

Maximum login retries

Configuration options for Maximum number of login retries are

After the defined amount of retries have been reached, the user account is suspended.

There is an option for automatic password reset (password is sent to user).

Policy_reference#Security

Brute force detection

Detection of spread attacks are implemented by registering the number of failed login attempts during a defined amount of time.

If a certain threshold is passed, the server will temporaryly deny further login attempts, for a defined aamout of time.

During this period the server will function normally for allready logged in users.

Configuration options for Brute force detection are

Policy_reference#Protection

Compliance built-in

Activity and data logging (optional)

Activity and Data Logging includes the automatic creation of a series of log files. Logging can be set up for each entity in an application providing insight and transparency in relation to: user activity, creation, changes and status of different records in an application.

- **Access Log**: Can be activated on an entity in an application. This automatically generates a log of which users have accessed and/or edited a given record.
- **Status Log**: Can be activated for an entity in an application. This automatically generates a log of the history of the created records, which shows how long a record has been in each status
- **Change Log**: Can be activated for an entity in an application. This automatically generates a log of what changes have been made to the individual records. Including who has changed what and when (timestamp).

How to: Each option is activated on the entity Advanced page.

Pro tip: Especially the status log can be used for setting up performance charts on dashboards, as it can give detailed information of how much time was spent in each step.

Versioning (optional)

By default file versioning is supported on the "Documents" and "Files" field types. In addition, data revisions can be supported on each entity. This automatically builds an audit log for each record.

In addition data revisions can be supported on each individual entity.

How to: Data revisions is activated on the entity Advanced page.

GDPR Deletion Policies (optional)

For each entity in a TS Application, a GDPR Deletion Policy can be set up, enabling automatic deletion or anonymization in accordance with the specified rules. The application will thus automatically delete or anonymize data and files in the application, cf. specified criteria.

How to:

1. Set up an action on a entity status
2. Check of deletion policy
3. Choose between anonoumization or deletion
4. Optionally select log data to also be deleted

In case you choose "anonoumization" you should define how each field should be handled

1. Click on a field
2. Click on Assignment
3. Check of anoumization
4. Optionally set value after change

Event and System Logging (recommended)

When Event and System Logging are turned on for an application, the following events are logged automatically:

- User logins
  - Succesfull normal user logins are hidden
  - Also contains client IP (used for MFA)
- System events
  - User errors
  - Scheduled services
  - Administrator logins

Error events will include stacktraces if available.

The eventlog can be cleaned automatically on a regular schedule.

Compliance external

Request logging

The webserver itself can be set up to do make detailed logs in file , containing for example

- Request timestamp, IP and session ID
- Stacktraces on errors

Depending on your security setup you might want to log these to a central repository

Data restrictions

Understanding permissions

Data access is retricted in two ways

- **Mandatory** permissions granting access to
  - Certain groups of fields (blocks)
  - Records in certain status

- **Optional** filters binding certain data to certain users
  - Owner user (the user that created the record)
  - List of users (dynamic list of users for each record)
  - Per group (group property)

If a user has no active permissions, they will not have any kind of access to the solution. Filters on the otherhand is just considered to be additional restrictions, limiting the access granted by permissions.

In both cases the security restrictions allways apply, even during system access, API interaction, intergration etc.

Permissions [mandatory]

Permissions to solutions are granted as a sum of multiple permissions.

Each permission contains

- Group
- 0-1 Status (records have status)
- 0-1 Blocks (fields belong to blocks)
- Allow read
- Allow write

Permissions stack in an aggregate like manner, allowing to build complex structures from different fragments. This is also the reason that the Allow read and Allow write properties can be set to empty values (typically for generic permissions).

Differentiated FIELD level access

Fields belong to blocks. Permissions may be bound to such a block.

A permission with a block specified will ONLY apply to the fields belonging to this block.

Differentiated STATE level access

Permissions may be bound to a certain status.

A permission with a status specified will ONLY apply to records in this status.

Filters [optional]

All ownership options can be overrided by belonging to a certain group, that ignores all types of filters (3 below).

Access to configuration:

Designer > [solution] > Security - Filters

Ownership by data exclusive group

Designer attribute: Use Exclusive groups for access control

The solution contains a **Exclusive group** that defines a group with access to this piece of data.

- Scope: Group
- Cardinality: One

Ownership by data member lists

Designer attribute: Use Lists of members for each item

The solution contains a **memberlist** field where users can have their access added or removed. Behind the scenes a table with a relation between the record and the user is maintained.

- Scope: User
- Cardinality: Many

Ownership by being the creator

Designer attribute: Use Creator only restriction (ignore group recommended)

You must have created this record in order to see access it.

- Scope: User
- Cardinality: One

Encryption

HTTPS / SSL is readily supported as the interface is indifferent of which protocol is used by the user.

It is however possible to force the user to use SSL by tweaking the configuration

- Force SSL during logins
- Force SSL in interface

Users making specific requests, will receive a rewritten redirect including all parameters, from the original request. The recommended setting is always using SSL at all times.

Guide to set up SSL: Setting_up_SSL/HTTPS

Security baseline

Security by design

The platform is security designed in accordance with OWASP version 4:

- Level 2: Compliant
- Level 3: +90% supported

In addition the platform supports a wide range of security schemes and logging features, needed for support of ISO27001, ISAE3000 etc.

The platform is subject to periodic penetration testing. Last customer testing was july 2021.

Default security

By default the Tempus Serva is verifiably secure to all common threat vectors, such as

- SQL injection
- Cross-Site Scripting
- Session highjacking
- Login replays
- etc

Protective measure includes common hardening efforts, such as

- Data sanitization
- Request throtling
- CSRF tokens
- etc

In addition to this baseline additional features can be activated per installation.

- Built-in: Features in the product itself
- External: Measures controlled in application server or operating system

Security built-in

Password Policies (recommended)

TS No-code Platform allows you to setup and enfoce the use of strong passwords through an appropriate password policy. Specify attribute requirements that control complexity and lifetime of passwords such as:

- Mimimum password length
- Special characters
- Maximum password age
- Maximum number of tries

The struictural restrictions will be honored when

- The user changes his password
- A new randow password is generated and sent

Passwords can also be set to expire after an amount of time.

How to: The polices can be changed in server configurations in the group *Password policies*

Note: The password polices will have no impact on SSO authentication

Multi-factor Authentication (recommended)

Device MFA

TS No-code Platform offers native Multi-factor Authentication to protect against unauthrised access by requiring a user to provide multiple authentication factors to prove their identity. At present two different options are available:

- MFA using session-specific, one-time-passcodes sent to the users mobile phone via regular or Flash SMS
  - You will need to create an account for sending SMS
  - Cost is approx. 0,30 DKK per message)
- MFA using af dedicated app from
  - Apple

    - Google
    - Microsoft

SMS requires very little of the users, while App based MFA is considered (slightly) more secure.

Note: If using singlesignon (SSO) the MFA will not be used

Location MFA

IP can be used as factor. In some cases slightly less secure, but much easier for the user.

Options include

- Country whitelisting (via IP)
- Static whitelisting of IP's
- Adaptive whitelisting of IP's

Adaptive whitelisting happens when the same user logs in from the same IP multiple times (typically 5).

IP MFA can be used together with normal MFA, so that SMS/App check is only required in case the IP is not whitelistet.

Geolocation blocking (optional)

Geoblocking will allow the servers to deny requests from certain countries.

The geoblocking will match the clients IP against af Geo service. The county will be matched to the servers whitelist of country names.

How to: Change the system configurations starting with *ipBlocker*

- Activate setting **ipBlockerActive**
- Set allowed countries in **ipBlockerAllowedCountries**

Request throttling (optional)

As specified in OWASP v4 system should be able to limit the amount of request a user can carry out in a system.

Limitations can be set on

- Pages hit
- WebDAV requests

- Upload (size/count)
- REST operations

How to: Edit server configurations starting with *limit*

Brute force prevention (optional)

This protection is handled by not serving too many requests to the login page, regardless of the source in question. This will prevent brute force attacks on distributed accounts using multiple machines. In case the detection triggers, login requests will be ignored for at set amount of time, while allready logged in users can continue their work.

How to: Define systems configurations starting with *bruteforce*

Additional configurations

- File whitelisting (uploadWhitelist)
- OWASP compliance (owaspCompliance)

Security external

Virus scanning

Scanning of uploaded files are left to software installed on the system.

The upload feature will temporarily store the files on the file system, so that detection mechanisms can quarantine the files in case they are infected.

Storage encryption

Storage encryption is normally supported by the underlying technologies, with the possible exception of password hashes (handled with BCrypt).

MySQL (+8) supports multiple encryption schemes

- The whole database
- Single schema (each TS installation)

Read more about encryption for MySQL and MariaDB

O/S level encryption technology includes

- Windows: BitLocker
- Linux: LUKS

Transport encryption (https)

Minimum requirements are SSL certificates. On Linux these can easily be obtained for free via LetsEncrypt.

Optionally the server can also apply to HSTS, using the following guideline for Tomcat.

Denial of service attacks

Protection against DOS attacks are best handled using dedicated services such as Cloudflare.

Single sign-on

TS NoCode contains its own user management. In order to minimize the effort in maintaining the profiles, and require less effort for users already authenticated in other systems.

Oauth2

There are multiple Oauth 2 sources available

- Azure
- Office 365
- Google
- LinkedIn
- Facebook
- MitID

Note than only the Office 365 source can be used to synchronize group membership.

Implementation

Setup will require 2 steps

1. Setting up the SSO source
2. Configuring your TS platform

In Designer > Modules > Configuration set up the following properties

- oauth___Allow: Set to true
- oauth___Tenant: From step 1 above
- oauth___Secret: From step 1 above
- oauth___Client: From step 1 above

LDAP integration

LDAP integration is not SSO per se, but rather using LDAP as an authentication source.

- Use LDAP to authenticate

- Import and link groups

Implementation

Learn about how to set up LDAP integration (insert link)

TS as Oauth2 provider

In case you want other systems to use TS to authenticate users, the platform can be set up to respond to Oauth2 requests.

Implementation

Contact TS support team to get a link to the required Wordpress SSO plugin

Also

- oauthWPAllow: Set true
- oauthWPClient: Set to anything
- oauthWPSecret: Set to anything
- oauthWPHost: URL of the Wordpress server