Proxedo API Security based on VM environment: Migration manual from 4.11.0 to 4.12.1

2025-07-02

1. Introduction

This guide describes the necessary steps and commands to upgrade PAS instance from version 4.11.0 to 4.12.1.

There are four main stages to describe the different phases and scenarios of the upgrade.

1. Creating a backup for safety
2. Upgrading a single node setup
3. Upgrading a multi node setup
4. Restoring the pre-upgrade state

2. Creating a backup for safety

Before starting the upgrade process, make sure there is a backup to which the current state of the actual PAS setup can be restored.

The following steps describe how to create a backup of an actual configuration manually.

All instructions need to be executed on the management node even in case of a multi node setup.

The password for the management component's admin user will be necessary to be able to backup the running configuration.

2.1. Bootstrap configuration

1. Log in to `pas` user by running `sudo -iu pas`.
2. Save the following configuration files on the node in a zip file:
   - `/opt/balasys/etc`
   - `/opt/balasys/.ssh` for a multi node setup
   - Parts of the automated core deployment tool:
     - `/opt/balasys/usr/share/automation/deploy-core.yml`
     - `/opt/balasys/usr/share/automation/host_vars`
     - `/opt/balasys/usr/share/automation/inventory.yml`
     - `/opt/balasys/usr/share/automation/roles/deploy-core/vars/main.yml`

Example command to compress bootstrap files in a single node setup

```
zip --recurse-paths bootstrap-config.zip --symlinks \
    /opt/balasys/etc/ \
    /opt/balasys/usr/share/automation/{deploy-core.yml,host_vars,inventory.yml} \
    /opt/balasys/usr/share/automation/roles/deploy-core/vars/main.yml
```

Example command to compress bootstrap files in a multi node setup

```
zip --recurse-paths bootstrap-config.zip --symlinks \
    /opt/balasys/.ssh/ \
```

```
/opt/balasys/etc/ \
/opt/balasys/usr/share/automation/{deploy-core.yml,host_vars,inventory.yml} \
/opt/balasys/usr/share/automation/roles/deploy-core/vars/main.yml
```

2.2. Creating a backup of the running configuration

1. Log in to the Web UI as admin user and navigate to the *Configuration Backup* page from the top bar.
2. On the *Configuration Backup* page select *Running* from the *Export configuration* dropdown menu.

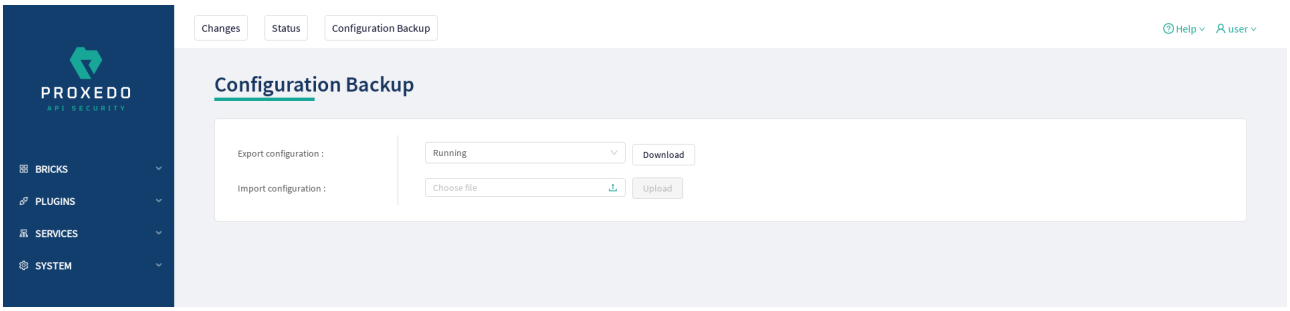

Figure 1. Backup and restore services with Proxedo API Security configuration

3. To export the running configuration, press the *Download* button. This will save the running configuration to a file named `running-config-backup.zip` in the working directory.

Save both files (`bootstrap-config.zip` and `running-config-backup.zip` displayed in the examples) to a backup server.

To restore the backup, follow the instructions in section Restoring the pre-upgrade state.

3. Upgrading a single node setup

This section describes how to upgrade PAS in a single node setup. In case any problem occurs during the upgrade and the version 4.11.0 needs to be restored, follow the instructions in section Restoring the pre-upgrade state.

3.1. Prerequisites

The following requirements need to be met before carrying out the upgrade process:

- The management, storage and core components of version 4.11.0 are running and healthy.
- The password for the administrator user of the management component is available.
- The new Debian packages are downloaded and available for installation on the node:
  - proxedo-api-security_4.12.1_all.deb
  - proxedo-api-security-mgmt_4.12.1_all.deb
  - proxedo-api-security-storage_4.12.1_all.deb

3.2. Upgrade steps

The user who performs the upgrade, needs to be logged in to docker.balasys.hu via the docker login command line tool, and needs to have access to the PAS docker images.

1. Run the following command in a new shell on the management node as `pas` user. We recommend using the pas home directory as your working directory to ensure the upgrader can correctly locate the license file. After the pre-upgrade process is run, a pas-config-4.11.0.zip can be found in the working directory.

```
docker run --user $(id -u):$(id -g) --rm -it --network=host --volume $(pwd):/upgrade
docker.balasys.hu/api-security/config-upgrader:4.12.1 --frontend-baseurl
http://localhost pre-upgrade
```

A custom base url can be defined using `--frontend-baseurl <url> pre-upgrade`. It is useful when the management node is configured to only accept requests on a specific HTTP host or when the TLS certificate is not valid for localhost.

When the frontend certificate is not trusted by the computer that runs pas-mgmt-update-config, the signing CA bundle can be provided using the `--cacert <ca-bundle-file>` option. The CA bundle file must contain the full CA chain and has to be placed in the working directory in PEM format.

2. Run the following command in the same shell and working directory. It attempts to convert the running configuration to make it compatible with the new PAS version. If the conversion is successful, the new configuration is saved as `pas-config-4.11.0.upgraded-to-4.12.1.zip`. However, if the automated conversion fails, the converter prompts the user for input.

```
docker run --user $(id -u):$(id -g) --rm -it --network=host --volume $(pwd):/upgrade
docker.balasys.hu/api-security/config-upgrader:4.12.1 convert-config
```

3. If the previous `convert-config` step was successful, run the following command to clean up the storage content:

```
docker run --user $(id -u):$(id -g) --rm -it --network=host --volume $(pwd):/upgrade
docker.balasys.hu/api-security/config-upgrader:4.12.1 delete-storage-content
```

4. Install the new Proxedo API Security packages as `root` user. This will not restart *systemd* services.
   - Use the simplified installer windows for a directed and easier way of installing the PAS packages.
   - When the installer asks whether the existing configuration files shall be overwritten, select 'yes'. This question will be asked for each package.

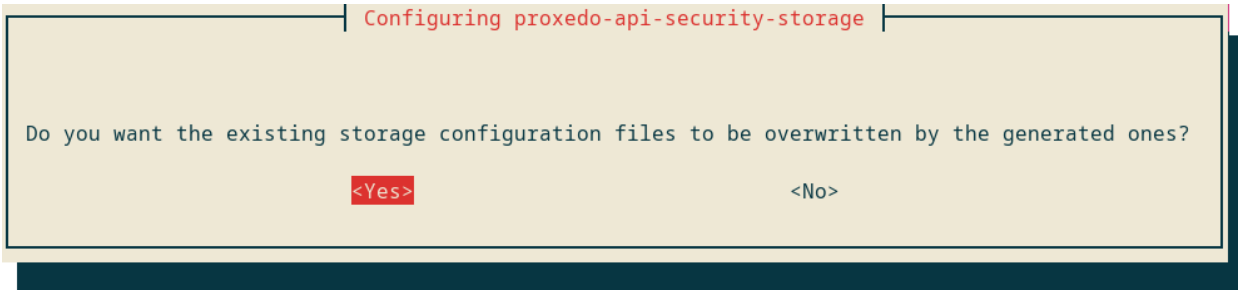

Figure 2. Question about overriding existing files

By agreeing the installer to overwrite the existing files, the installer creates a working, single node configuration, consequently, there is no need to complete any manual updates. The newly created configuration is saved in a backup subdirectory next to its original location.

- `/opt/balasys/etc/infrastructure/storage/backups` for storage infrastructure configuration
- `/opt/balasys/etc/storage/backups` for storage configuration
- `/opt/balasys/etc/infrastructure/mgmt/backups` for management infrastructure configuration
- `/opt/balasys/etc/mgmt/backups` for management configuration
- `/opt/balasys/etc/infrastructure/core/backups` for core infrastructure configuration

◦ When the Debian installer asks what to do with the new versions of the configuration files, select `N` for keeping installed versions:

Example question from the Debian installer

```
Installing new version of config file
/opt/balasys/etc/infrastructure/storage/docker-compose.conf ...

Configuration file '/opt/balasys/etc/storage/config.yml'
==> Modified (by you or by a script) since installation.
==> Package distributor has shipped an updated version.
  What would you like to do about it ?  Your options are:
   Y or I  : install the package maintainer's version
   N or O  : keep your currently-installed version
     D     : show the differences between the versions
     Z     : start a shell to examine the situation
The default action is to keep your current version.
*** config.yml (Y/I/N/O/D/Z) [default=N] ?
```

5. Log in to `pas` user by executing `sudo -iu pas`. Carry out the following operations as `pas` user.
6. Verify that the upgraded configuration file, named `pas-config-4.11.0.upgraded-to-4.12.1.zip`, exists in your current working directory.
7. Run the update command for each component:
   ◦ `pas-update` for core
   ◦ `pas-mgmt-update` for management
   ◦ `pas-storage-update` for storage
8. If LDAP authentication is used for the management component, verify the `/opt/balasys/etc/mgmt/config.yml` configuration file. LDAP attribute names are now case-sensitive. For example, if you previously used `memberof`, it must be changed to `memberOf` to be valid.
9. Run the *checkconfig* command for each component for which it is available:
   ◦ `pas-mgmt-checkconfig` for management
   ◦ `pas-storage-checkconfig` for storage
10. Stop the PAS management, storage and core components.

By stopping the core component, the proxy functionality is suspended, and the network traffic going through is interrupted. The traffic will be served again when the core component is restarted in the last step.

11. Delete the blob-store's (MinIO) system files, as it will be replaced by Garage. This step ensures that the new software can initialize the storage without conflicts and prevents leftover files from MinIO that are no longer needed. Use the following command to remove the files: `find /opt/balasys/var/persistent/blob-`

```
store/ -mindepth 1 -delete
```

12. Start the PAS management and storage components.
13. Run `pas-mgmt-upgrade-config post-upgrade --config-apply-timeout 20` in the same directory where the pre-upgrade and the configuration conversion steps were performed. Follow the instructions of the script to complete.

If a new administrator password was provided in the installer, that one needs to be used.

A custom base url can be defined using `pas-mgmt-upgrade-config --frontend-baseurl <url> post-upgrade`. It is useful when the management node is configured to only accept requests on a specific HTTP host or when the TLS certificate is not valid for localhost.

When the frontend certificate is not trusted by the computer that runs pas-mgmt-update-config, the signing CA bundle can be provided using the `--cacert <ca-bundle-file>` option. The CA bundle file must contain the full CA chain and has to be placed in the working directory in PEM format.

14. Start the PAS core component.

4. Upgrading a multi node setup

This section describes how to upgrade PAS in a multi node setup. In case any problem occurs during the upgrade and the version 4.11.0 needs to be restored, follow the instructions in section Restoring the pre-upgrade state.

4.1. Prerequisites

The following requirements need to be met before carrying out the upgrade process:

- The management and storage components of version 4.11.0 are running and healthy on the management node.

If the core component is also run on the management node, it needs to be running and functioning too.

- The storage and core components of version 4.11.0 are running and healthy on the core node.
- The password for the administrator user of the management component is available.
- The new Debian packages are downloaded and available for installation on the management node:
  - proxedo-api-security_4.12.1_all.deb
  - proxedo-api-security-mgmt_4.12.1_all.deb
  - proxedo-api-security-storage_4.12.1_all.deb

4.2. Upgrade steps

The user who performs the upgrade, needs to be logged in to docker.balasys.hu via the docker

login command line tool, and needs to have access to the PAS docker images.

Execute all steps on the management node, unless otherwise indicated for specific steps.

1. Make sure *local* PAS management and storage components of version 4.11.0 are running.
2. Make sure *remote* PAS storage and core components of version 4.11.0 are running.
3. Run the following command in a new shell on the management node as `pas` user. After the pre-upgrade process is run, a pas-config-4.11.0.zip can be found in the working directory.

```
docker run --user $(id -u):$(id -g) --rm -it --network=host --volume $(pwd):/upgrade
docker.balasys.hu/api-security/config-upgrader:4.12.1 --frontend-baseurl
http://localhost pre-upgrade
```

A custom base url can be defined using `--frontend-baseurl <url> pre-upgrade`. It is useful when the management node is configured to only accept requests on a specific HTTP host or when the TLS certificate is not valid for localhost.

When the frontend certificate is not trusted by the computer that runs pas-mgmt-update-config, the signing CA bundle can be provided using the `--cacert <ca-bundle-file>` option. The CA bundle file must contain the full CA chain and has to be placed in the working directory in PEM format.

4. Run the following command in the same shell and working directory. It attempts to convert the running configuration to make it compatible with the new PAS version. If the conversion is successful, the new configuration is saved as `pas-config-4.11.0.upgraded-to-4.12.1.zip`. However, if the automated conversion fails, the converter prompts the user for input.

```
docker run --user $(id -u):$(id -g) --rm -it --network=host --volume $(pwd):/upgrade
docker.balasys.hu/api-security/config-upgrader:4.12.1 convert-config
```

5. If the previous `convert-config` step was successful, run the following command to clean up the storage content:

```
docker run --user $(id -u):$(id -g) --rm -it --network=host --volume $(pwd):/upgrade
docker.balasys.hu/api-security/config-upgrader:4.12.1 delete-storage-content
```

6. Install the new Proxedo API Security packages locally as `root` user. This will not restart *systemd* services.

The simplified installer is designed to help single node installation. Ignore all questions asked by the installer during the upgrade and select "No" where possible. Ignore any questions asked twice and any error prompts.

   - When the Debian installer asks what to do with the new versions of the configuration files, select `N` for keeping the installed versions.

     Example question from the Debian installer

```
Installing new version of config file
/opt/balasys/etc/infrastructure/storage/docker-compose.conf ...
```

```
Configuration file '/opt/balasys/etc/storage/config.yml'
==> Modified (by you or by a script) since installation.
==> Package distributor has shipped an updated version.
  What would you like to do about it ?  Your options are:
   Y or I  : install the package maintainer's version
   N or O  : keep your currently-installed version
     D     : show the differences between the versions
     Z     : start a shell to examine the situation
The default action is to keep your current version.
*** config.yml (Y/I/N/O/D/Z) [default=N] ?
```

7. Log in to `pas` user by executing `sudo -iu pas`. Carry out the following operations as `pas` user.
8. Update the configuration file of the storage component. The configuration file is located at **`/opt/balasys/etc/storage/config.yml`**.
   - The following attributes need to be updated:
     - `blob_store.access_key` to a new value. It starts with "GK", followed by 12 hex-encoded bytes. `echo "GK$(openssl rand -hex 12)"`
     - `blob_store.secret_key` to a new value. It is a 32-byte hex-encoded random string, which can be generated with a command such as `openssl rand -hex 32`.
   - The following attributes need to be added:
     - `blob_store.rpc_secret` to a new value. It is a 32-byte hex-encoded random string, which can be generated with a command such as `openssl rand -hex 32`.
     - `blob_store.admin_token` to a new value. It can be any string. A random token generated with `openssl rand -base64 32` is recommended.
9. Make sure the `DOCKER_IMAGE_TAG` variable is updated to 4.12.1 in all `docker-compose.conf` files:
   - **`/opt/balasys/etc/infrastructure/pas/docker-compose.conf`** for core
   - **`/opt/balasys/etc/infrastructure/mgmt/docker-compose.conf`** for management
   - **`/opt/balasys/etc/infrastructure/storage/docker-compose.conf`** for storage
10. Update storage keys and values in the management **`/opt/balasys/etc/infrastructure/mgmt/docker-compose.conf`** file:
    - Rename `MINIO_ACCESS_KEY` to `S3_ACCESS_KEY` and set its value to match the `blob_store.access_key` value from the storage configuration file (**`/opt/balasys/etc/storage/config.yml`**).
    - Rename `MINIO_SECRET_KEY` to `S3_SECRET_KEY` and set its value to match the `blob_store.secret_key` value from the storage configuration file (**`/opt/balasys/etc/storage/config.yml`**).
11. If core services are also installed on the management node, update storage keys and values in the core component's **`/opt/balasys/etc/infrastructure/pas/docker-compose.conf`** file:
    - Rename `MINIO_ACCESS_KEY` to `S3_ACCESS_KEY` and set its value to match the `blob_store.access_key` value from the storage configuration file (**`/opt/balasys/etc/storage/config.yml`**).
    - Rename `MINIO_SECRET_KEY` to `S3_SECRET_KEY` and set its value to match the `blob_store.secret_key` value from the storage configuration file (**`/opt/balasys/etc/storage/config.yml`**).
12. Verify that the upgraded configuration file, named `pas-config-4.11.0.upgraded-to-4.12.1.zip`, exists in your current working directory.
13. Update the configuration of the automated deployment tool.
    a. At **`/opt/balasys/etc/automation/common_vars.yml`**
       - Update the `storage_deb_path` to the new `.deb` file.
       - Update the `core_deb_path` to the new `.deb` file.
       - Update the `common.docker.PAS_IMAGE_TAG` to 4.12.1.

- Add the `common.docker.PAS_DOCKER_REPO` key. The value has to be `api-security`.
- Update the `common.storage.config.blob_store.access_key` to a new value. The value has to be the same as the one in the storage configuration file.
- Update the `common.storage.config.blob_store.secret_key` to a new value. The value has to be the same as the one in the storage configuration file.
- Add the `common.storage.config.blob_store.rpc_secret` key. The value has to be the same as the one in the storage configuration file.
- Add the `common.storage.config.blob_store.admin_token` key. The value has to be the same as the one in the storage configuration file.

Example extract of values for the updated attributes

```
storage_deb_path: /tmp/proxedo-api-security-storage_4.12.1_all.deb
core_deb_path: /tmp/proxedo-api-security_4.12.1_all.deb
common:
  docker:
    PAS_IMAGE_TAG: 4.12.1
    PAS_DOCKER_REPO: api-security
  storage:
    config:
      blob_store:
        access_key: GK65d0d9fb0b5a70076f19cd0a
        secret_key:
d8b62bfd1f2202d2505f7baa12292aac9ae6615ad97bb700d40acdb395bc38cf
        rpc_secret:
631d41aec186f895f9f4fbe7023fca158ee4e45a8c8f795248c1f985150d661b
        admin_token: osdMc7euvXy8vfToHycCWLLOsqqGgveeRdLjNil4CFk=
```

14. Run the update command for each component:
    - `pas-update` for core if core is also run on the management node
    - `pas-mgmt-update` for management
    - `pas-storage-update` for storage
15. If LDAP authentication is used for the management component, verify the `/opt/balasys/etc/mgmt/config.yml` configuration file. LDAP attribute names are now case-sensitive. For example, if you previously used `memberof`, it must be changed to `memberOf` to be valid.
16. Run the *checkconfig* command for each component for which it is available:
    - `pas-mgmt-checkconfig` for management
    - `pas-storage-checkconfig` for storage
17. Stop the *local* PAS management and storage components.
18. Stop the *remote* core and storage component by running `pas-mgmt-deploy-core --stop-core`.
19. Delete the blob-store's (MinIO) system files on every node. This is necessary because MinIO will be replaced by Garage, and this step ensures that Garage can initialize the storage without conflicts and prevents leftover files from MinIO that are no longer needed. On each such node, use the following command to remove the files: `find /opt/balasys/var/persistent/blob-store/ -mindepth 1 -delete`
20. Restart the *local* PAS management and storage components.
21. Deploy and restart the *remote* core component by running `pas-mgmt-deploy-core --deploy-core`. This will restart the *remote* storage and core services with PAS version 4.12.1.
22. Run `pas-mgmt-upgrade-config post-upgrade --config-apply-timeout 20` in the same directory where the pre-upgrade and the conversion of the steps were performed. Follow the instructions of the script to complete.

A custom base url can be defined using `pas-mgmt-upgrade-config --frontend-baseurl <url> post-upgrade`. It is useful when the management node is configured to only accept requests on a specific HTTP host or when the TLS certificate is not valid for localhost.

When the frontend certificate is not trusted by the computer that runs pas-mgmt-update-config, the signing CA bundle can be provided using the `--cacert <ca-bundle-file>` option. The CA bundle file must contain the full CA chain and has to be placed in the working directory in PEM format.

23. Restart the *local* PAS core component.
24. If HA is run, also restart the *local* HA component.
25. If HA is run, also upgrade and restart the *remote* HA component by running `pas-mgmt-deploy-core --deploy-ha --restart-ha`.

5. Restoring the pre-upgrade state

5.1. Cleaning up to pre-upgrade state

1. Stop all PAS services on *all nodes*.
   - `systemctl stop proxedo-api-security` for core
   - `systemctl stop proxedo-api-security-mgmt` for management
   - `systemctl stop proxedo-api-security-storage` for storage
2. Remove PAS packages as `root` user on *all nodes*. Remove packages only from those nodes where they are installed.
   - `apt remove --purge proxedo-api-security` for core
   - `apt remove --purge proxedo-api-security-mgmt` on management
   - `apt remove --purge proxedo-api-security-storage` on storage
3. Remove the `pas` user by running `userdel --force --remove pas`.

5.2. Restoring the configuration to pre-upgrade state

All instructions need to be executed on the management node.

1. Reinstall PAS version 4.11.0 packages.
2. Log in to `pas` user by running `sudo -iu pas`.
3. Copy the files saved during the backup to the `pas` user's home directory.

5.2.1. Bootstrap configuration

It is important to run all commands as `pas` user to prevent from accidentally overwriting system files.

1. Unzip the saved bootstrap configuration files in the `/opt/balasys` directory as `pas` user by running `unzip -u -o bootstrap-config.zip -d /`.

Example bootstrap configuration restore command and output

```
$ unzip -u -o bootstrap-config.zip -d /
Archive:  bootstrap-config.zip
   creating: /opt/balasys/etc/
   creating: /opt/balasys/etc/ha/
  inflating: /opt/balasys/etc/ha/config.yml
   creating: /opt/balasys/etc/mgmt/
 extracting: /opt/balasys/etc/mgmt/users.htpass
  inflating: /opt/balasys/etc/mgmt/config.yml
   creating: /opt/balasys/etc/storage/
[...]
```

2. Start all PAS services including the HA component if previously an HA setup was run.
3. If a multi node setup is being restored, also deploy the remote node by running the remote deployment command.

```
pas-mgmt-deploy-core --deploy-core
```

4. If an HA setup is run, also start the HA service on the remote node.

```
pas-mgmt-deploy-core --deploy-ha
```

5.2.2. Restoring the running configuration

1. Log in to the Web UI as the administrator user and navigate to the *Configuration Backup* page from the top bar.
2. To import the running configuration, on the *Configuration Backup* page choose a configuration file from the computer and press *Upload* to upload the configuration.

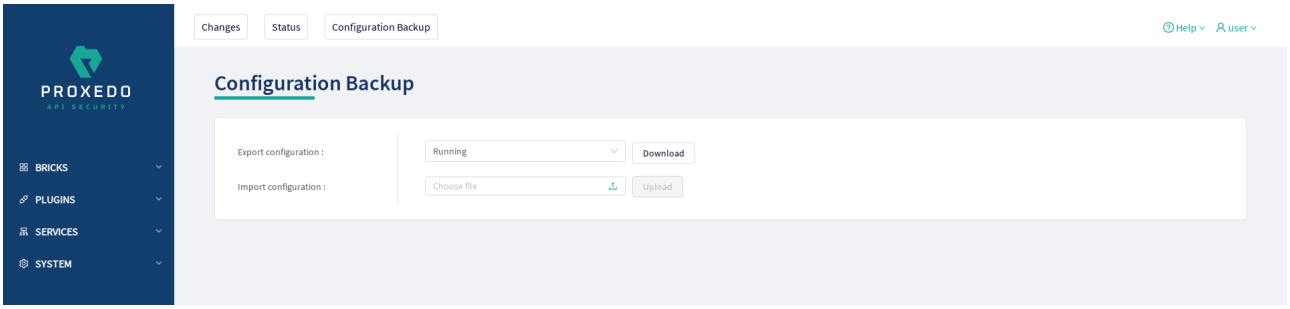

Figure 3. Backup and restore services with Proxedo API Security configuration

3. Apply the configuration.