



PROXEDO

API SECURITY

Proxedo API Security
Migration manual 4.0.0 to 4.0.1

Copyright (C) Balasys IT Ltd. 4.0.1, 2022-08-29

Copyright © 2019 Balasys IT Ltd.. All rights reserved. This document is protected by copyright and is distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this document may be reproduced in any form by any means without prior written authorization of Balasys.

This documentation and the product it describes are considered protected by copyright according to the applicable laws.

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>). This product includes cryptographic software written by Eric Young (eay@cryptsoft.com)

Linux™ is a registered trademark of Linus Torvalds.

Windows™ 10 is registered trademarks of Microsoft Corporation.

The Balasys™ name and the Balasys™ logo are registered trademarks of Balasys IT Ltd.

The Zorp™ name and the Zorp™ logo are registered trademarks of Balasys IT Ltd.

The Proxedo™ name and the Proxedo™ logo are registered trademarks of Balasys IT Ltd.

AMD Ryzen™ and AMD EPYC™ are registered trademarks of Advanced Micro Devices, Inc.

Intel® Core™ and Intel® Xeon™ are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries.

All other product names mentioned herein are the trademarks of their respective owners.

DISCLAIMER

Balasys is not responsible for any third-party websites mentioned in this document. Balasys does not endorse and is not responsible or liable for any content, advertising, products, or other material on or available from such sites or resources. Balasys will not be responsible or liable for any damage or loss caused or alleged to be caused by or in connection with use of or reliance on any such content, goods, or services that are available on or through any such sites or resources.

2022-08-29

Introduction

This guide describes the necessary steps and commands to upgrade your PAS instance from version 4.0.0 to 4.0.1.

There are four main parts to describe 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

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.

Bootstrap configuration

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

Running configuration

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

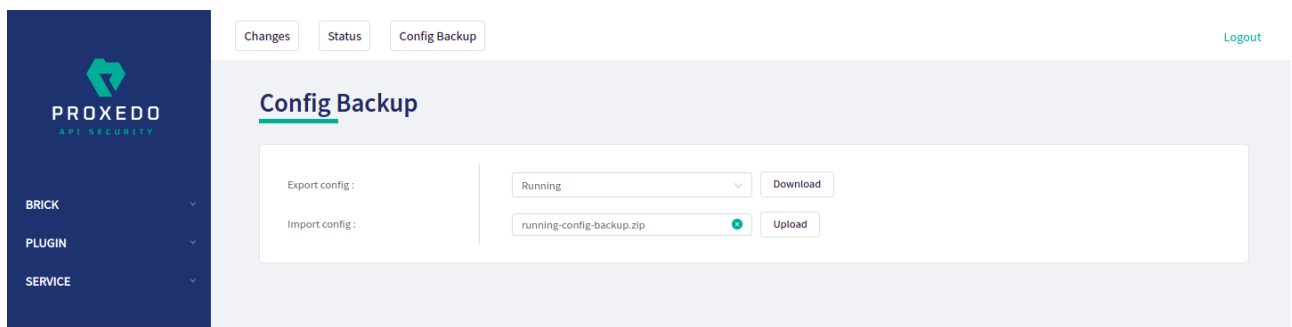


Figure 1. Backup and restore services with Proxedo API Security configuration

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 [Restoring the pre-upgrade state](#).

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 4.0.0 needs to be restored, follow the instructions in [Restoring the pre-upgrade state](#).

Prerequisites

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

- The management, storage and core components of version 4.0.0 are running and healthy.
- The password for the admin user of the management component is available.
- The new Debian packages are downloaded and available for installation on the node:
 - `proxedo-api-security_4.0.1_all.deb`
 - `proxedo-api-security-mgmt_4.0.1_all.deb`
 - `proxedo-api-security-storage_4.0.1_all.deb`

Upgrade steps

1. 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.

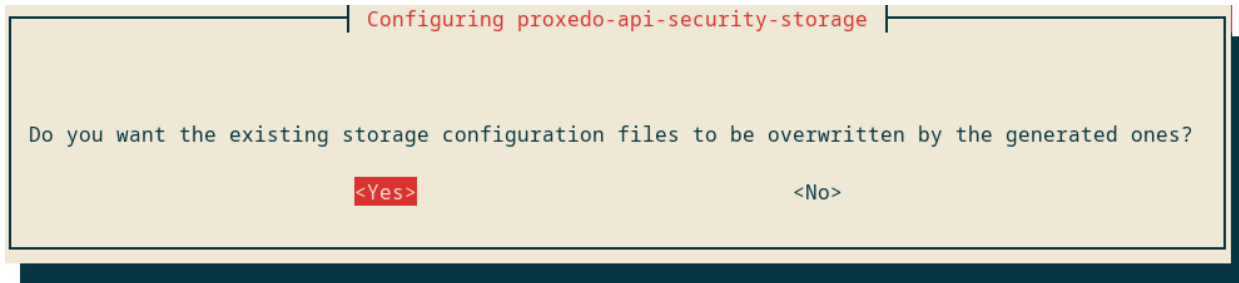


Figure 2. Question about overriding existing files

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/balasy/etc/infrastructure/storage/backups` for storage infrastructure configuration
- `/opt/balasy/etc/storage/backups` for storage configuration
- `/opt/balasy/etc/infrastructure/mgmt/backups` for management infrastructure configuration
- `/opt/balasy/etc/mgmt/backups` for management configuration
- `/opt/balasy/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/balasy/etc/infrastructure/storage/docker-compose.conf ...

Configuration file '/opt/balasy/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] ?
```

2. Log in to `pas` user by executing `sudo -iu pas`. Carry out the following operations as `pas` user.

- Run `pas-mgmt-upgrade-config pre-upgrade`. This will guide you through the pre-upgrade process. Once the pre-upgrade finishes, you will find a `pas-config-4.0.0.zip` in your working directory.



The old administrator password needs to be used, even if you provided a new one during the installation.

- Run `pas-mgmt-upgrade-config convert-config` in the same directory. It will convert your old running configuration to be compatible with the new version. The new configuration will be saved as `pas-config-4.0.0.upgraded-to-4.0.1.zip`.
- Run the update command for each component:
 - `pas-update` for core
 - `pas-mgmt-update` for management
 - `pas-storage-update` for storage
- Run the `checkconfig` command for each component for which it is available:
 - `pas-mgmt-checkconfig` for management
 - `pas-storage-checkconfig` for storage
- Restart the PAS management and storage components.
- Stop the PAS core component.



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

- Run `pas-mgmt-upgrade-config post-upgrade` in the same directory where the pre-upgrade and 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.

- Start the PAS core component.

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 4.0.0 needs to be restored, follow the instructions in [Restoring the pre-upgrade state](#).

Prerequisites

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

- The management and storage components of version 4.0.0 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 4.0.0 are running and healthy on the core node.
- The password for the admin user of the management component is available.
- The new Debian packages are downloaded and available for installation on the management node:
 - proxedo-api-security_4.0.1_all.deb
 - proxedo-api-security-mgmt_4.0.1_all.deb
 - proxedo-api-security-storage_4.0.1_all.deb

Upgrade steps

Execute all steps on the management node.

1. Make sure *local* PAS management and storage components of version 4.0.0 are running.
2. Make sure *remote* PAS storage and core components of version 4.0.0 are running.
3. 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/balasy/etc/infrastructure/storage/docker-compose.conf ...

Configuration file '/opt/balasy/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] ?
```

4. Log in to `pas` user by executing `sudo -iu pas`. Carry out the following operations as `pas` user.
5. Run `pas-mgmt-upgrade-config pre-upgrade`. This will guide you through the pre-upgrade process. Once the pre-upgrade finishes, you will find a `pas-config-4.0.0.zip` in your working directory.
6. Run `pas-mgmt-upgrade-config convert-config` in the same directory. It will convert your old running configuration to be compatible with the new version. The new configuration will be saved as `pas-config-4.0.0.upgraded-to-4.0.1.zip`.

7. Update the configuration of the automated deployment tool.

- a. At `/opt/balasy/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 4.0.1.

Example extract of values for the updated attributes

```
storage_deb_path: /tmp/proxedo-api-security-storage_4.0.1_all.deb
core_deb_path: /tmp/proxedo-api-security_4.0.1_all.deb
common:
  docker:
    PAS_IMAGE_TAG: 4.0.1
```

8. Make sure the `DOCKER_IMAGE_TAG` variable is updated to 4.0.1 in all `docker-compose.conf` files:

- `/opt/balasy/etc/infrastructure/pas/docker-compose.conf` for core
- `/opt/balasy/etc/infrastructure/mgmt/docker-compose.conf` for management
- `/opt/balasy/etc/infrastructure/storage/docker-compose.conf` for storage

9. Run the update command for each component:

- `pas-update` for core if you also run core on the management node
- `pas-mgmt-update` for management
- `pas-storage-update` for storage

10. Run the `checkconfig` command for each component for which it is available:

- `pas-mgmt-checkconfig` for management
- `pas-storage-checkconfig` for storage

11. Restart the *local* PAS management and storage components.



By stopping the *local* core component, the proxy functionality is hung up, and the network traffic going through is interrupted. The traffic will only be served again when the *local* core component is restarted after the `post-upgrade` command.



This step will restart the HA component if it was running.

12. Restart the *remote* core component by running `pas-mgmt-deploy-core --restart-core`. This will restart the *remote* nodes with PAS version 4.0.0 but that is expected. Upgrading the *remote* components will be done in a later step.



By stopping the *remote* core component, the proxy functionality is hung up, and the network traffic going through is interrupted. The traffic will only be served again when the *remote* core component is restarted after the `post-upgrade` command.



This step will restart the HA component if it was running.

13. Run `pas-mgmt-upgrade-config post-upgrade` in the same directory where you performed the pre-upgrade and convert steps. Follow the instructions of the script to complete.
14. Restart the *local* PAS core component.
15. If you are running HA, also restart the *local* HA component.
16. Upgrade and restart the *remote* PAS core component by running `pas-mgmt-deploy-core --deploy-core --restart-core`.
17. If you are running HA, also upgrade and restart the *remote* HA component by running `pas-mgmt-deploy-core --deploy-ha --restart-ha`.
18. Open the UI as admin user and go to the *Status* page. If you see **FAILED** in the `Reload success` column, go to the *Changes* page, and click *Config Apply* even if there are no pending changes.

Restoring the pre-upgrade state

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`.

Restoring to pre-upgrade state

All instructions need to be executed on the management node.

1. Reinstall PAS version 4.0.0 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.

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/balasy` 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/balasy/et/
  creating: /opt/balasy/et/ha/
 inflating: /opt/balasy/et/ha/config.yml
  creating: /opt/balasy/et/mgmt/
 extracting: /opt/balasy/et/mgmt/users.htpass
 inflating: /opt/balasy/et/mgmt/config.yml
  creating: /opt/balasy/et/storage/
[...]
```

2. Start all PAS services including the HA component if you previously ran a HA setup.
3. If you are restoring a multi node setup, also deploy the remote node by running the remote deployment command.

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

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

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

Running configuration

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

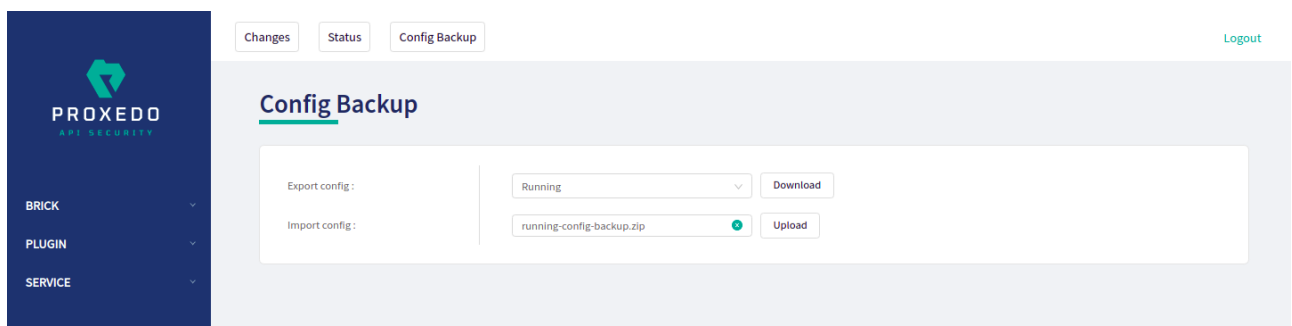


Figure 3. Backup and restore services with Proxedo API Security configuration

3. Apply the configuration.