



EDB

Lasso

Release 4.0.0

EnterpriseDB Corporation

Sep 01, 2023

CONTENTS

1	Phase 1	3
1.1	Automated diagnostics gathering	3
1.2	Downloading Lasso	3
1.3	Linux systems	4
1.3.1	Requirements	4
1.3.2	Postgres and BDR3 report	4
1.3.3	Barman report	4
1.3.4	Replication Manger (repmgr) report	5
1.3.5	Postres Enterprise Manager (PEM) report	6
1.3.6	Failover Manager (EFM) report	6
1.3.7	Replication Server (xDB) report	6
1.3.8	PgBouncer report	7
1.3.9	HARP report	8
1.3.10	etcd report	8
1.3.11	System-only report	9
1.4	Windows systems	9
1.4.1	Requirements	9
1.4.2	Usage	9
1.4.3	Postgres Enterprise Manager (PEM) report	9
1.4.4	Replication Server (xDB) report	10
1.5	Common considerations	11
1.5.1	Security considerations	14
2	Phase 2	15
2.1	Supervised Lasso report generation	15
2.1.1	General architecture	15
2.1.2	Hardware related	15
2.1.3	Software related	16
2.1.4	Amazon Web Services (AWS)	16
3	Phase 3	17
3.1	Returning the Lasso report	17
3.2	Troubleshooting	17
3.3	What do we gather in our Lasso report?	17
3.4	Detailed breakdown of gathered data	18
3.4.1	GNU/Linux operating system ### barman crontab/cron (barman_crontab_cron)	18
3.4.2	Microsoft Windows operating system ### PEM configuration - Windows (pem_configuration_windows)	40
3.4.3	PostgreSQL/PGD3 nstance ### Current archiver stats (postgresql_archiver)	46

3.4.4	Details for every Postgres/BDR3 database ### PGD1 replication slots (postgresql_db_bdr1_replication_slots)	57
3.4.5	Barman (Backup and Recovery Manager) ### Barman check (barman_check)	80
3.5	Servers accepting upload of reports	82

A *Lassoreport* is the first step of any evaluation and assessment process delivered by EDB on existing installations of Postgres or related tools, including supportability checks , health checks ,**and** incident response investigations.

You can generate it both in an *_automated_*

and in a *_supervised_* way.

Some information can be gathered through automated tools in a noninvasive way, in phase 1 of the Lasso report. Some other information isn't automatically detected (at least in a noninvasive way). You must communicate it as part of the Lasso report, in phase 2.

After you return the gathered information to EDB engineers in phase 3, EDB proceeds with data analysis. Then comes either:

- An offer, in the case of an initial assessment
- A final report assessing the status and the *health* of your Postgres server, in the case of a health check service

Important: You must provide EDB with all relevant information about the underlying system to check, in particular network and hardware configurations.

1.1 Automated diagnostics gathering

The process of automated diagnostics gathering is performed by a small multi-platform application written in Python. You can also run Lasso on systems where Postgres isn't installed to gather all relevant information regarding the underlying operating system.

Consult the specific information for your operating system.

Important: Lasso can't affect your data. It gathers statistics and diagnostics information from your Postgres server, with imperceptible effects on the workload. *No actual data* from the rows of your Postgres user tables is gathered.²

inspection.

1.2 Downloading Lasso

EDB has developed a small multi-platform application called Lasso that can safely gather relevant diagnostics data on a system where Postgres and other relevant supported software, such as Barman, is running.

EDB distributes the application in a binary form by way of the
[Customer Portal](#)

and grants usage to its customers using their *companytoken* as a means of authentication.

Important: Consider Lasso a disposable application. Always download the application before exporting a report (or a series of reports), as the application is in continuous deployment mode.

To download Lasso for your system, use the **Lasso** link in the left panel of the Portal.

Unless your company requested a version of Lasso for an isolated network system, without external access to EnterpriseDB's infrastructure, you can upload the produced tarball directly through Lasso using the `--upload` option. For more information, see [Servers accepting upload of reports](#).

Alternatively, you can attach the report to a specific support ticket through the Portal or use the **Supportoperations>Uploadreport** menu from your company's page.

² : Gathered diagnostics data files are available for your

1.3 Linux systems

1.3.1 Requirements

Glibc 2.5 or newer

1.3.2 Postgres and BDR3 report

As the user postgres, copy the Lasso binary into a folder.

Important: The same user that runs the Postgres server process must run the command. The user must have permission to write in the current working directory.

A standard Lasso run consists of:

```
./lasso
```

Lasso writes the output file in the current working directory.

Warning: Lasso needs to execute files from the temporary directory. If your system temporary directory is mounted with the *noexec* flag, the following error occurs:

```
./lasso: error while loading shared libraries:
libz.so.1: failed to map segment from shared object:
Operation not permitted
```

In that case, you can specify a different temporary file location using the TMPDIR environment variable:

```
TMPDIR=/some/alternate/tmp/location ./lasso`
```

Important: When run on a local instance, Lasso requires a user that belongs to the `pg_monitor` role, which is available in Postgres 10 and later. Before 10, you must run as a superuser. For more information, see [Default Roles](#) in the PostgreSQL documentation.

Lasso transparently gathers information related to all supported versions of EDB Postgres Distributed (PGD) and pg-logical.

1.3.3 Barman report

Lasso can run on systems where Barman is installed. In that case, it gathers Barman-related information, as well as system information, that helps during analysis.

As the user barman, copy the Lasso binary into a folder.

Important: You must run the command as the same user that runs the Barman process, and you must have enough permission to write in the current working directory. If you installed Barman using RPM/DEB packages, Barman is configured to run as the user barman.

The Barman report is enabled by default if Lasso runs as the barman user and looks for configuration files in the expected locations, typically `/etc/barman.conf`. Enter `man 5 barman` for details.

In general, if you installed Barman using EDB-certified RPM and DEB packages, all you need to do is execute Lasso as the barman user.

Managing custom installations of Barman

If you have custom installations of Barman, you can enable the barman report by passing the `--barman` option to Lasso:

```
./lasso --barman
```

You can also point to a specific global configuration file by using the `--barman-configuration` option as follows:

```
./lasso --barman-configuration /opt/barman/barman.conf
```

1.3.4 Replication Manger (repmgr) report

Lasso can run on systems where repmgr is installed. In that case, it gathers repmgr-related information.

In general, if you installed repmgr using EDB-certified RPM and DEB packages, all you need to do is execute Lasso.

How Lasso finds the repmgr configuration file

Lasso uses the following approach while trying to identify the repmgr configuration file. It uses the first one it finds.

1. Use the one provided through the `--repmgr-configuration` option, if given.
2. Check the paths provided by packages. For example, in CentOS/RHEL packages, the configuration file is usually put under `/etc/repmgr/<PG_VERSION>/repmgr.conf`. Lasso inspects those folders, if they exist, in descending order, and uses the first repmgr configuration file that it finds.
3. Check if a `repmgr.conf` file exists in the current directory.
4. Check if a `/etc/repmgr.conf` file exists.
5. Check the path given by `pg_config --sysconfdir` for the `repmgr.conf` file.
6. Check the output of `systemctl cat <service_name>`, where `<service_name>` is any service whose name contains `repmgr`, and try to get the `repmgr.conf` file from the service unit file. It checks the units in descending order, just as it does in approach 2, and uses the first match.

Managing custom installations of repmgr

You can point to a specific repmgr configuration file by using the `--repmgr-configuration` option as follows:

```
./lasso --repmgr-configuration /opt/repmgr/repmgr.conf
```

This approach is preferred, as it guarantees Lasso will use the correct repmgr configuration file instead of trying to find it automatically.

1.3.5 Postres Enterprise Manager (PEM) report

Lasso can run on systems where PEM is installed. In that case, it also gathers PEM-related information, like some configuration files and information about services.

Lasso inspects the well-known paths for configuration files, as well as well-known service names. If you have custom PEM installations, Lasso doesn't gather the related information.

1.3.6 Failover Manager (EFM) report

Lasso can run on systems where EFM is installed. In that case, it gathers EFM-related information.

In general, if you installed EFM using EDB-certified RPM and DEB packages, all you need to do is execute Lasso.

How Lasso finds the EFM configuration file

Lasso uses the following approach while trying to identify the EFM configuration file. It uses the first one it finds.

1. Use the one provided through the `--efm-configuration` option, if given.
2. Check the paths provided by packages. The configuration file is usually put under `/etc/edb/efm-<EFM_VERSION>/efm.properties`. Lasso inspects those folders, if they exist, in descending order, and uses the first EFM configuration file that it finds.
3. Check the output of `systemctl cat <service_name>`, where `<service_name>` is replaced with any service whose name starts with `edb-efm-`. Then try to get the `efm.properties` file from the service unit file. It checks the units in descending order, just as it does in approach 2, and uses the first match.

How Lasso finds the EFM binary

Lasso inspects the paths provided by packages. The binary file is usually put under `/usr/edb/efm-<EFM_VERSION>/bin/efm`. Lasso inspects these folders, if they exist, in descending order, and uses the first EFM binary file that it finds.

Managing custom installations of EFM

You can point to a specific EFM configuration file by using the `--efm-configuration` option, as follows:

```
./lasso --efm-configuration /opt/EFM/efm.properties
```

This approach is preferred, as it guarantees Lasso will use the correct EFM configuration file instead of trying to find it automatically.

1.3.7 Replication Server (xDB) report

Lasso can run on systems where xDB is installed. In that case, it gathers xDB-related information.

In general, if you installed xDB using EDB-certified RPM packages or the bitrock installer, all you need to do is execute Lasso.

How Lasso finds the xDB configuration file

Lasso uses the following approach while trying to identify the xDB configuration files. It uses the first one it finds.

1. Use the ones provided through the `--xdb-pubserver-configuration` and `--xdb-subserver-configuration` options, if given.
2. Check the paths provided by xDB 7 RPM package and bitrock installer. The configuration files are usually put under `/etc/edb/xdb/etc`. Lasso inspects this folder, if it exists, and uses the `xdb_pubserver.conf` and `xdb_subserver.conf` files found under that folder.
3. Check the paths provided by xDB 6 RPM package and bitrock installer, which are usually `/usr/ppas-xdb-<XDB_VERSION>/etc` and `/opt/PostgreSQL/EnterpriseDB-xDBReplicationServer/etc`, respectively. Lasso inspects these folders and, if they exist, uses the `xdb_pubserver.conf` and `xdb_subserver.conf` files found under the folder. In the case of RPM packages, as there can be a lot of folders, they're inspected from newest XDB version to oldest.

How Lasso finds the xDB binary

Lasso uses the following approach while trying to identify the xDB binary file. It uses the first one it finds.

1. Check the paths provided by xDB 7 RPM package and bitrock installer. The binary files are usually put under `/etc/edb/xdb/bin`. Lasso inspects this folder, if it exists, and uses the `edb-repcli.jar` file found under that folder.
2. Check the paths provided by the xDB 6 RPM package and bitrock installer, which are usually `/usr/ppas-xdb-<XDB_VERSION>/bin` and `/opt/PostgreSQL/EnterpriseDB-xDBReplicationServer/bin`, respectively. Lasso inspects these folders and, if they exist, uses the `edb-repcli.jar` file found under the folder. In the case of RPM packages, as there can be a lot of folders, they're inspected from newest XDB version to oldest.

Managing custom installations of EFM

You can point to a specific xDB publication or subscription server configuration file by using the `--xdb-pubserver-configuration` and `--xdb-subserver-configuration` options, as follows:

```
./lasso --xdb-pubserver-configuration /opt/xDB/pubserver.conf --xdb-subserver-
configuration /opt/xDB/subserver.conf
```

This approach is preferred, as it guarantees Lasso will use the correct xDB configuration files instead of trying to find it automatically.

1.3.8 PgBouncer report

Lasso can run on systems where PgBouncer is installed. In that case, it gathers PgBouncer-related information.

In general, if you installed PgBouncer using EDB-certified RPM and DEB packages, and PgBouncer instances are up and running, all you need to do is execute Lasso.

How Lasso finds the PgBouncer configuration file

Lasso uses the list of running processes to identify the PgBouncer processes and uses the configuration file retrieved from there.

You can also specify one or more PgBouncer configuration files through the Lasso argument `--pgbouncer-configuration`. It's mainly useful in scenarios where a PgBouncer instance isn't running or if Lasso can't detect the configuration file from the process list.

Lasso collects information from all identified PgBouncer instances.

1.3.9 HARP report

Lasso can run on systems where HARP is installed. In that case, it gathers HARP-related information.

In general, if you installed HARP using EDB-certified RPM and DEB packages, all you need to do is execute Lasso.

How Lasso finds the HARP configuration file

Lasso uses the following approach while trying to identify the HARP configuration file. It uses the first one it finds.

1. Use the one provided through the `--harp-configuration` option, if given.
2. Check if the `/etc/harp/config.yml` file exists.

Managing custom installations of HARP

You can point to a specific HARP configuration file by using the `--harp-configuration` option, as follows:

```
./lasso --harp-configuration /opt/harp/config.yml
```

This approach is preferred, as it guarantees Lasso will use the correct HARP configuration file instead of trying to find it automatically.

Running Lasso on a HARP proxy node

The HARP proxy nodes don't have a Postgres or EDB Postgres Advanced Server service running on them, as they're intended only as a proxy for connections coming in from the applications.

When using Lasso to gather information from a HARP proxy node, you must use the `--system-only` option so it doesn't try to connect to the database server.

1.3.10 etcd report

Lasso can run on systems where etcd is installed. In that case, it gathers etcd-related information.

How Lasso finds the etcd configuration file

Lasso uses the following approach while trying to identify the etcd configuration file. It uses the first one it finds.

1. Use the one provided through `--etcd-configuration` option, if given.
2. Check if the `/etc/etcd/etcd.conf` file exists.

1.3.11 System-only report

Lasso can also run on systems where Postgres or Barman aren't installed to gather all relevant information regarding the underlying operating system.

You can run a system-only report with the following command:

```
./lasso --system-only
```

1.4 Windows systems

1.4.1 Requirements

Windows Server 2008 r2 or newer

1.4.2 Usage

Expand the distribution zip file in a folder.

A standard Lasso run consists of:

```
lasso.exe
```

Important: The process must run in an elevated environment, such as an elevated command prompt or with the *Run as administrator** context menu option.

Lasso writes the output file in the working directory.

Important: Lasso is distributed as a 64-bit executable.

1.4.3 Postgres Enterprise Manager (PEM) report

Lasso can run on systems where PEM is installed. In that case, it also gathers PEM-related information, like some configuration files and information about services.

Lasso inspects the well-known paths for configuration files, as well as well-known service names. If you have custom PEM installations, Lasso doesn't gather the related information.

1.4.4 Replication Server (xDB) report

Lasso can run on systems where xDB is installed. In that case, it gathers xDB-related information.

In general, if you installed xDB using EDB-certified packages, all you need to do is execute Lasso.

How Lasso finds the xDB configuration file

Lasso uses the following approach while trying to identify the xDB configuration files. It uses the first one it finds:

1. Use the ones provided through `--xdb-pubserver-configuration` and `--xdb-subserver-configuration` options, if given.
2. Check the paths provided by xDB 7 packages. The configuration files are usually put under `<drive>:\Program Files\edb\EnterpriseDB-xDBReplicationServer\etc` . Lasso inspects this folder, if it exists, and uses the `xdb_pubserver.conf` and `xdb_subserver.conf` files found under that folder.
3. Check the paths provided by xDB 7 packages. The configuration files are usually put under `<drive>:\Program Files\PostgreSQL\EnterpriseDB-xDBReplicationServer\etc` . Lasso inspects this folder, if it exists, and uses the `xdb_pubserver.conf` and `xdb_subserver.conf` files found under that folder.

How Lasso finds the xDB binary

Lasso uses the following approach while trying to identify the xDB binary files. It uses the first one it finds:

1. Check the paths provided by xDB 7 packages. The binary files are usually put under `<drive>:\Program Files\edb\EnterpriseDB-xDBReplicationServer\bin` . Lasso inspects this folder, if it exists, and uses the `edb-repcli.jar` file found under that folder.
2. Check the paths provided by xDB 7 packages. The binary files are usually put under `<drive>:\Program Files\PostgreSQL\EnterpriseDB-xDBReplicationServer\bin` . Lasso inspects this folder, if it exists, and uses the `edb-repcli.jar` file found under that folder.

Managing custom installations of EFM

You can point to a specific xDB publication or subscription server configuration file by using the `--xdb-pubserver-configuration` and `--xdb-subserver-configuration` options, as follows:

```
lasso.exe --xdb-pubserver-configuration C:\xDB\pubserver.conf --xdb-subserver-  
configuration C:\xDB\subserver.conf
```

This approach is preferred, as it guarantees Lasso will use the correct xDB configuration files instead of trying to find it automatically.

System-only report

Lasso can also run on systems where Postgres isn't installed to gather all relevant information regarding the underlying operating system.

You can run a system-only report with the following command:

```
lasso.exe --system-only
```

1.5 Common considerations

Lasso uses the libpq environment variables to get the Postgres connection parameters. You can find the list of the environment variables in the [PostgreSQL documentation](#).

The connection parameters, among other options, can also be passed as command line arguments:

```
usage: lasso [-h] [-H HOST_NAME] [-p PORT] [--password PASSWORD]
            [--lock-timeout LOCK_TIMEOUT]
            [--statement-timeout STATEMENT_TIMEOUT] [--bindir BINDIR]
            [--depth [{surface,shallow,deep}]]
            [--describe [{short,json,full}]] [--version] [--latest-version]
            [--system-only | --barman]
            [--barman-configuration BARMAN_CONFIGURATION]
            [--repmgr-configuration REPMGR_CONFIGURATION]
            [--efm-configuration EFM_CONFIGURATION]
            [--xdb-pubserver-configuration XDB_PUBSERVER_CONFIGURATION]
            [--xdb-subserver-configuration XDB_SUBSERVER_CONFIGURATION]
            [--pgbouncer-configuration PGBOUNCER_CONFIGURATION]
            [--harp-configuration HARP_CONFIGURATION]
            [--etcd-configuration ETCD_CONFIGURATION]
            [--upload] [--keep-report] [--is-latest-version]
            [dbname] [user]
```

EDB Lasso

positional arguments:

dbname	Database name to connect to (default root)
user	Database user name (default root)

optional arguments:

-h, --help	show this help message and exit
-H HOST_NAME, --host-name HOST_NAME	Database host name or socket directory (default local socket)
-p PORT, --port PORT	Database server port (default 5432)
--password PASSWORD	Database server password
--lock-timeout LOCK_TIMEOUT	Database connection lock timeout (default 3s)
--statement-timeout STATEMENT_TIMEOUT	Database connection statement timeout (default 5min)
--bindir BINDIR	Postgres binaries directory (autodetect by default)
--depth [{surface,shallow,deep}]	Depth of the report (default deep)
--describe [{short,json,full}]	Describes every single module, in terms of action and output
--version	Shows Lasso version and modules revision
--latest-version	Shows the latest available version of EDB Lasso, taken from the EDB Web Services
--system-only	Gather only system-related information - without requiring a Postgres connection
--barman	Gather Barman status, enabled by default when Lasso is

(continues on next page)

(continued from previous page)

```

run as barman user and the executable exists.
Defaults to False. When Barman reporting is enabled,
we do not gather Postgres related information
--barman-configuration BARMAN_CONFIGURATION
    Barman configuration file. By default use the native
    algorithm in Barman to find the configuration file.
    Valid only if Barman reporting is enabled
--repmgr-configuration REPMGR_CONFIGURATION
    Path to the repmgr.conf file, if using a non-default
    path
--efm-configuration EFM_CONFIGURATION
    Path to the EFM properties file, if using a non-
    default path
--xdb-pubserver-configuration XDB_PUBSERVER_CONFIGURATION
    Path to the xDB publication server configuration file,
    if using a non-default path
--xdb-subserver-configuration XDB_SUBSERVER_CONFIGURATION
    Path to the xDB subscription server configuration
    file, if using a non-default path
--pgbouncer-configuration PGBOUNCER_CONFIGURATION
    Path to the pgbouncer.ini file. You can specify multiple
    files separated by comma
--harp-configuration HARP_CONFIGURATION
    Path to the config.yml file, if using a non-default
    path
--etcd-configuration ETCD_CONFIGURATION
    Path to the etcd.conf file, if using a non-default
    path
--upload
    Report tarball file is sent to EDB at the end of the
    execution. The file will be removed if successfully
    uploaded unless --keep-report is specified
--keep-report
    Keep a local copy of the report even after a
    successful upload to EDB
--is-latest-version
    Only check if this is the latest available version of
    Lasso and returns exit code 0 if this is the latest
    version and 1 otherwise

```

You can also use a configuration file for Lasso, so you can omit most of the command line options that your environment might require. Lasso looks for configuration files in the following paths, in this order:

1. ./edb-lasso.cfg in the same directory where Lasso is running
2. ./edb-lasso.conf in the same directory where Lasso is running
3. \$HOME/.edb-lasso.conf
4. /etc/edb-lasso.conf

It uses the first match .

A template file for the configuration file looks like this:

```

; Lasso template configuration file
;
; Copyright (C) EnterpriseDB UK Limited 2015-2022 - All Rights Reserved.

```

(continues on next page)

(continued from previous page)

```

; Licensed only for use with an EnterpriseDB subscription

[customer]
; the "Company code" from the customer page in the EDB Portal
id=
; the "Token" from the customer page in the EDB Portal
token=
; the depth of the Lasso report. Must be one between: surface, shallow and deep
depth=deep

[postgresql]
; Lasso uses the following connection string parameters to connect to your
; PostgreSQL cluster and take diagnostics. By default it will attempt a peer
; connection to socket under directory /var/run/postgresql
; dbname=
; user=
; port=
; hostname can be a host or a socket directory
; hostname=/var/run/postgresql
; password=

; session variable to control timeout on lock waits
; lock_timeout=3s
; session variable to control statement execution time
; statement_timeout=5min

[environment]
; PostgreSQL binaries directory. Lasso will try to detect this automatically if
; not set
; bindir=
; if the host is able to access the internet and reach the EDB servers to upload
; Lasso reports
; external_access=yes

[barman]
; path to barman.conf, if using a non-default one
; configuration=

[repmgr]
; path to repmgr.conf, if using a non-default one
; configuration=

[efm]
; path to efm.properties, if using a non-default one
; configuration=

[xdb]
; path to xdb_pubserver.conf, if using a non-default one
; pubserver_configuration=
; path to xdb_subserver.conf, if using a non-default one
; subserver_configuration=

```

(continues on next page)

(continued from previous page)

```
[pgbouncer]
; path to pgbouncer.ini, if using a non-default one
; configuration=

[harp]
; path to config.yml, if using a non-default one
; configuration=
```

You can use this template to set up your configuration file. Uncomment the desired parameters and set their values according to your environment.

If you installed Lasso from DEB or RPM packages, You can find that same template configuration file at `/etc/edb-lasso.conf.templ`.

Important: If you installed Lasso from DEB or RPM packages, then you must have a configuration file that contains at least the customer ID and token set. That isn't required if you're running the Lasso disposable binary, which contains those variables embedded in the binary.

The script produces a *TAR file* containing the gathered data. You must return this file to EDB engineers as part of the process or send it automatically with the `--upload` option.

Important: Some companies have requested for Lasso to operate in an isolated network. In that case, the `--upload` option isn't permitted. (You can easily verify this setting from your company page in the Portal or through the `--version` runtime option.) To take advantage of this feature, make sure that the server that's analyzed by Lasso can access the 443 port of any of the front-line websites listed in *Servers accepting upload of reports*.

You can also use Lasso on a server that has no Postgres installation. In that case, use the `--system-only` option, and the *TAR file* produced will contain only system-related information.

1.5.1 Security considerations

When running queries in the database, Lasso tries to use a role that has enough privileges to gather the required information from the tool from which metrics are being gathered.

The following are the tools and the roles that Lasso tries to use for each of them. Lasso tries to use the first available role in each tool role list. *Initialconnectionrole* means the role that was provided through Lasso CLI when running the tool—usually postgres or enterprisedb.

- PostgreSQL: - pg_monitor - initial connection role
- PgLogical: - pglogical_superuser - initial connection role
- PGD: - bdr_monitor - initial connection role
- PEM: - pem_user - initial connection role
- repmgr: - initial connection role
- xDB: - initial connection role

Most of the PGD gatherings try using the bdr_monitor role. However, one of them, which is in charge of gathering conflicts, tries to use the role bdr_read_all_conflicts for that purpose. That's the only exception.

In any of the cases, it uses a read-only transaction while querying metrics and configurations from the database.

PHASE 2

2.1 Supervised Lasso report generation

Lasso can't detect certain information. You must provide detailed information about the server running Postgres and the database system architecture, including business continuity implementations, such as availability and disaster recovery.

The following is a non-exhaustive list of questions for your network, system, and database administrators. (You might find you need additional information.)

2.1.1 General architecture

- What are the main use cases for your databases? - Client applications and tools - Frameworks and programming languages
- What's your current high-availability strategy for Postgres?
- What's your current disaster-recovery strategy for Postgres? - Describe your backup strategy. - Is it regularly being tested? - Do you have dedicated hardware for the recovery and testing of your backups?
- Are you using any connection poolers for Postgres? If yes, which ones?
- What's the network architecture around your database server?

2.1.2 Hardware related

- What's the configuration of the disks? - Brand and model of the disks - Brand and model of the controller/interface, where applicable - Brand and model of any device for disk management (SAN, NAS, and so on), where applicable - RAID configuration, where applicable - Configuration of write cache - Battery status, where applicable - Multipath configuration, where applicable - LVM physical volume dump, where applicable
- Is the server protected by a UPS?

2.1.3 Software related

- Is the server virtual? - Which technology is used for virtualization?
- Software RAID configuration, where applicable
- Are you using any trending/monitoring tools (that is, Cacti, ZenOS, Zabbix, and so on)?
- Are you using any alerting tools (that is, Nagios/Icinga, ZenOS, Zabbix, and so on)?
- Do you use any record-based software for replication in Postgres (that is, Slony, Bucardo, Londiste, and so on)?
- Have you ever used the pg_upgrade tool to upgrade your Postgres server? What version did you upgrade this way?

2.1.4 Amazon Web Services (AWS)

If you have Postgres on Amazon instances:

- If using AWS, what's the size of your instances?
- Can you provide information on the regions of the servers?
- Are you using CloudWatch?
- Are you using VPC?
- In case of S3 backups, are you using any encryption for your backups and archives?

PHASE 3

3.1 Returning the Lasso report

The Lasso report on a single instance ends when you return the following information to your contact person at EDB:

1. The TAR files from phase 1 by uploading them in the Portal using the **Uploadreport** menu or as a specific ticket attachment
2. A document containing the answers to the questions from phase 2 by uploading it in the Portal in the **MyDocuments** section

3.2 Troubleshooting

For support and assistance on Lasso, refer to your contact person at EDB.

3.3 What do we gather in our Lasso report?

A Lasso report is designed to allow any EDB engineer to rapidly feel the pulse of your database by getting relevant information from your operating system and your Postgres server.

You can get a list with a detailed description of the information that's gathered from your system through lasso by entering:

```
lasso --describe
```

If you need more information, contact your account manager at EDB.

Lasso informs its level of depth when scraping your system. There are three levels:

- **Surface** : Noninvasive gathering process, mainly used to perform a supportability check of the instance and focused on the current moment. In terms of Postgres, the focus is on the instance, its configuration, and its general health. This level is the minimal level of scraping.
- **Shallow** : Gathering of performance-related information, including basic trending data. In terms of Postgres data, its focus is moved down to the database. This level is suitable for support-incident resolution.
- **Deep** : Most detailed level of scraping, suitable for advanced and security health checks but also for severe support incidents. In terms of Postgres data, its focus is on tables and indexes.

3.4 Detailed breakdown of gathered data

3.4.1 GNU/Linux operating system ### barman crontab/cron (barman_crontab_cron)

Description

Output from `crontab -l`, if running as barman. Content of `/etc/cron.d/barman`, if it exists.

Report output

- File `/linux/barman_cron.data` : Content of `/etc/cron.d/barman`, if it exists
- File `/linux/barman_crontab.data` : Output from `barman crontab -l`, if barman user

Depth

Surface

Security impact

Low — Might have entries in crontab/cron with sensitive data

debug_sources (debug_sources)

Description

Counts files under `/usr/src/debug` to detect the application source code present in the system and ease live debugging

Report output

File `/linux/debug_sources.data` : Sources for GNU debugger

Depth

Surface

Security impact

Low — No known security impact

EFM CLI (`efm_cli`)

Description

Gets output of `efm cluster-status` command

Report output

File `/tools/efm/cli/cluster_status.out` : Output of `efm cluster-status <cluster_name>` command

Depth

Surface ##### Security impact

Low — No known security impact.

EFM configuration (`efm_configuration`)

Description

EFM properties and nodes configuration files

Report output

- File `/tools/efm/config/efm.nodes` : EFM nodes file
- File `/tools/efm/config/efm.properties` : EFM properties file

Depth

Surface ##### Security impact

Low — No known security impact.

EFM systemctl (`efm_systemctl`)

Description

When EFM services are detected, collects status and cat of the corresponding services. Checks for any service whose name starts with `edb-efm-`.

Report output

- File `/tools/efm/systemd/<service_name>_cat.data` : Output of `'systemctl cat`
- File `/tools/efm/systemd/<service_name>_status.data` : Output of `'systemctl status`

Depth

Surface

Security impact

Low — No known security impact.

etcd CLI (`etcd_cli`)

Description

Gathers the output of some `etcdctl` commands, if `etcdctl` is available in the server. The commands are `endpoint status` and `endpoint health` .

Report output

- File `/tools/etcd/cli/endpoint_status.out` : Output of `etcdctl endpoint status` command
- File `/tools/etcd/cli/endpoint_health.out` : Output of `etcdctl endpoint health` command

Depth

Surface

Security impact

Low — No known security impact.

etcd configuration (`etcd_configuration`)

Description

Collects `etcd` configuration file that's found in the server

Report output

File `/tools/etcd/config/<basename>` : etcd configuration file

Depth

Surface

Security impact

Low — No known security impact.

etcd systemctl (etcd_systemctl)**Description**

When etcd services are detected, collects status and cat of the corresponding services. Checks for any service whose name starts with etcd .

Report output

- File `/tools/etcd/systemd/<service_name>_cat.data` : Output of 'systemctl cat
- File `/tools/etcd/systemd/<service_name>_status.data` : Output of 'systemctl status

Depth

Surface

Security impact

Low — No known security impact.

HARP CLI (harp_cli)**Description**

Gathers output of a few harpctl command outputs using the `config.yml` file that's found in the server. The commands are: `cluster` , `proxies` , `locations` , `nodes` , and `version` .

Report output

- File `/tools/harp/cli/version.out` : Output of `harp -f <conf_file_path> version` command
- File `/tools/harp/cli/proxies.out` : Output of `harp -f <conf_file_path> get proxies -o yaml` command
- File `/tools/harp/cli/nodes.out` : Output of `harp -f <conf_file_path> get nodes -o yaml` command
- File `/tools/harp/cli/locations.out` : Output of `harp -f <conf_file_path> get locations -o yaml` command
- File `/tools/harp/cli/cluster.out` : Output of `harp -f <conf_file_path> get cluster -o yaml` command

Depth

Surface

Security impact

Low — No known security impact.

HARP configuration (`harp_configuration`)

Description

Collects HARP configuration file that's found in the server

Report output

- File `/tools/harp/config/harp.cluster.init.yaml` : HARP bootstrap configuration file
- File `/tools/harp/config/<basename>` : HARP configuration file

Depth

Surface

Security impact

Low — No known security impact.

HARP systemctl (harp_systemctl)

Description

When HARP services are detected, collects status and cat of the corresponding services. Checks for any service whose name starts with harp .

Report output

- File /tools/harp/systemd/<service_name>_cat.data : Output of 'systemctl cat
- File /tools/harp/systemd/<service_name>_status.data : Output of 'systemctl status

Depth

Surface

Security impact

Low — No known security impact.

Block devices layout (linux_block_devices_layout)

Description

Information on block devices layout from the lsblk command

Report output

File /linux/lsbk.data : lsbk command output

Depth

Surface

Security impact

Low — No known security impact.

Processor governor (linux_cpu_governor)

Description

Processor scaling governor from the files in /sys/devices/system/cpu

Report output

- File `/linux/sys/energy_perf_bias.data` : Intel Performance and Energy Bias attributes
- File `/linux/sys/intel_pstate.data` : Intel pstate configuration
- File `/linux/sys/cpu_scaling_driver.data` : available CPU scaling driver
- File `/linux/sys/cpu_scaling_available_governors.data` : available CPU scaling governors
- File `/linux/sys/cpu_scaling_governor.data` : active CPU scaling governor

Depth

Surface

Security impact

Low — No known security impact.

Mounted file systems and available space (`linux_devices_info`)

Description

Lists mounted filesystems through the `mount` command and free space using `df`

Report output

- File `/linux/diskspace.data` : amount of available disk space
- File `/linux/mount.data` : Output of the `mount` command

Depth

Surface

Security impact

Low — No known security impact.

File systems configuration (`linux_disk_configuration`)

Description

Disk configuration obtained through the `/etc/fstab` file

Report output

File `/linux/fstab.data` : contents of `/etc/fstab`

Depth

Surface

Security impact

Low — No known security impact.

OS distribution, kernel and device data (`linux_distro_collector`)**Description**

Information about the Linux distribution currently in use, as returned by the `lsb_release` command

Report output

- File `/linux/release.data` : The Linux distribution currently in use
- File `/linux/release_source.data` : The name of the collected file or the executed command

Depth

Surface

Security impact

Low — No known security impact.

Hardware (`linux_hardware_info`)**Description**

Hardware info through `lspci`

Report output

File `/linux/lspci.data` : Hardware info from `lspci`

Depth

Surface

Security impact

Low — No known security impact.

Hypervisor (linux_hypervisor_collector)

Description

Information about the type of virtualization used, as returned by the `systemd-detect-virt` command

Report output

File `/linux/hypervisor.data` : The name of the collected file or the executed command

Depth

Surface

Security impact

Low — No known security impact.

Kernel (linux_kernel_info)

Description

Kernel info, transparent huge pages status, and disk scheduler configuration obtained by combining the output of the commands `uname` and `ipcs` with the contents of the `/proc` and `/sys` filesystems

Report output

- File `/linux/read_ahead.data` : Info on the read ahead
- File `/linux/schedulers.data` : Scheduler info from `/sys` dir
- File `/linux/sys/kernel_mm_transparent_hugepage.data` : Transparent huge pages info
- File `/linux/ipcs.data` : `ipcs` command output
- File `/linux/uname.data` : `uname` command output

Depth

Surface

Security impact

Low — No known security impact.

Kernel limits (linux_kernel_limits)**Description**

Configuration file for the pam_limits module

Report output

File /linux/limits.data : content of the limits.conf file

Depth

Surface

Security impact

Low — No known security impact.

Processor usage statistics (linux_mpstat)**Description**

Processor statistics from the mpstat command.

Report output

File /linux/mpstat.data : Output from 'mpstat -P ALL 1 10'

Depth

Surface

Security impact

Low — No known security impact.

Network interfaces (`linux_network_interfaces`)

Description

Network interface information from the `ip` and `ifconfig` commands

Report output

- File `/linux/ifconfig.data` : Output from 'ifconfig'
- File `/linux/ip_address_list.data` : Output from 'ip address list'

Depth

Surface

Security impact

Low — No known security impact.

Installed packages via rpm or dpkg (`linux_packages_info`)

Description

Information about the system packages installed using `rpm` or `dpkg`

Report output

- File `/linux/packages-dpkg.data` : List of packages installed using `dpkg`
- File `/linux/packages-rpm.data` : List of packages installed using `rpm`

Depth

Surface

Security impact

Low — No known security impact.

PostgreSQL disk layout (linux_postgresql_disk_layout)**Description**

Lists all files in the Postgres data directory using `find` for links and `ls` for files

Report output

- File `/linux/pg_ls.data` : List of files inside the data directory
- File `/linux/pg_links.data` : List of links inside the data directory

Depth

Surface

Security impact

Low — No known security impact.

SELinux (linux_sestatus)**Description**

SELinux status from `sestatus`

Report output

File `/linux/sestatus.data` : Output from ‘setstatus’

Depth

Surface

Security impact

Low — No known security impact.

System identification (`linux_system_identity`)

Description

Collects hostname, network interfaces, system info (uname), system identifier, and release info

Report output

- File `/linux/id/system_release.data` : OS information from `/etc/system-release`
- File `/linux/id/os_release.data` : OS information from `/etc/os-release`
- File `/linux/id/machine_id.data` : Machine ID contained in `/etc/machine-id`
- File `/linux/id/uname.data` : Information about the running kernel
- File `/linux/id/hostname.data` : Fully qualified domain name
- File `/linux/id/interfaces.data` : Network addresses of the host

Depth

Surface

Security impact

Low — No known security impact.

dmesg and /proc information (`linux_system_info`)

Description

System info from the contents of the `/proc` filesystem and through the output of `dmesg` command.

Report output

- File `/linux/lsmmod.data` : Lsmmod output
- File `/linux/dmesg_with_timestamp.data` : Dmesg output (human readable timestamps)
- File `/linux/dmesg.data` : Dmesg output
- File `/linux/proc/sys_net_ipv4.data` : Network info from `/proc`
- File `/linux/proc/sys_vm.data` : vm info from `/proc`
- File `/linux/proc/sys_kernel.data` : Kernel info from `/proc`
- File `/linux/vmstat.data` : VM statistics from `/proc`
- File `/linux/proc/mounts.data` : Mount points from `/proc`
- File `/linux/proc/uptime.data` : Uptime info from `/proc`
- File `/linux/proc/loadavg.data` : Load avg from `/proc`
- File `/linux/proc/meminfo.data` : Memory info from `/proc`

Depth

Surface

Security impact

Low — No known security impact.

System status - iostat (linux_system_status_iostat)**Description**

System status from the `iostat` command

Report output

File `/linux/iostat.data` : Info on I/O statistics

Depth

Surface

Security impact

Low — No known security impact.

System status - nfsiostat (linux_system_status_nfsiostat)**Description**

System status from the `nfsiostat` command

Report output

File `/linux/nfsiostat.data` : nfs I/O statistics

Depth

Surface

Security impact

Low — No known security impact.

System status - ps (linux_system_status_ps)

Description

System status from the ps command

Report output

File /linux/ps.data : Active processes info

Depth

Surface

Security impact

Low — Some processes might contain sensitive data in their names.

System status - sar (linux_system_status_sar)

Description

System status from the sar command

Report output

- File /linux/sar.data : Actual sar info
- File /linux/sar-yesterday.data : sar info from yesterday

Depth

Surface

Security impact

Low — No known security impact.

System status - top (linux_system_status_top)**Description**

System status from the top command

Report output

File /linux/top.data : Process information

Depth

Surface

Security impact

Low — Some processes might contain sensitive data in their names.

System status - vmstat (linux_system_status_vmstat)**Description**

System status from the vmstat command

Report output

File /linux/vmstat.data : Info on processes, memory, paging, block IO, traps, disks and cpu activity

Depth

Surface

Security impact

Low — No known security impact.

systemctl units (linux_systemctl_units)**Description**

Systemctl list-units on a systemd server

Report output

File `/linux/systemctl-list-units.data` : Output of ‘systemctl list-units’

Depth

Surface

Security impact

Low — No known security impact.

tuned (linux_tuned)

Description

Tuned status and profiles

Report output

- Directory `/linux/tuned/tune-profiles` : Files from `/etc/tune-profiles`
- Directory `/linux/tuned/tuned` : Files from `/etc/tuned`
- File `/linux/tuned/tuned.conf` : The file `/etc/tuned.conf`
- File `/linux/tuned/tuned-list.data` : Output from `tuned_adm list`
- File `/linux/tuned/tuned-active.data` : Output from `tuned_adm active`

Depth

Surface

Security impact

Low — No known security impact.

PEM configuration (pem_configuration)

Description

PEM configuration files from PEM agent, PEM server, and PEM web server

Report output

- File `/tools/pem/config/edb-ssl-pem.conf` : PEM web server SSL configuration file
- File `/tools/pem/config/edb-pem.conf` : PEM web server configuration file
- File `/tools/pem/config/install-config` : PEM server configuration file (installation config file)
- File `/tools/pem/config/config_setup.py` : PEM server setup configuration file
- File `/tools/pem/config/pem.wsgi` : PEM server WSGI definition file
- File `/tools/pem/config/agent.cfg` : PEM agent configuration file

Depth

Surface

Security impact

Low — No known security impact.

PEM systemctl (pem_systemctl)**Description**

When PEM is detected, collects PEM agent and PEM web server status and content

Report output

- File `/tools/pem/systemd/<service_name>_cat.data` : Output of 'systemctl cat
- File `/tools/pem/systemd/<service_name>_status.data` : Output of 'systemctl status

Depth

Surface

Security impact

Low — No known security impact.

PgBouncer configuration (pgbouncer_configuration)**Description**

PgBouncer configuration files

Report output

File `/tools/pgbouncer/<num>/config/<basename>` : PgBouncer configuration file from instance `<num>`

Depth

Surface

Security impact

Low — No known security impact.

PgBouncer systemctl (pgbouncer_systemctl)

Description

When PgBouncer services are detected, collects status and cat of the corresponding services. Checks for any service that contains any of the PgBouncer configuration files.

Report output

- File `/tools/pgbouncer/<num>/systemd/<service_name>_cat.data` : Output of 'systemctl cat from instance `<num>`
- File `/tools/pgbouncer/<num>/systemd/<service_name>_status.data` : Output of 'systemctl status from instance `<num>`

Depth

Surface

Security impact

Low — No known security impact.

postgres/enterprisedb crontab (postgres_enterprisedb_crontab)

Description

Output from crontab `-l` , if running as postgres or enterprisedb

Report output

- File `/linux/enterprisedb_crontab.data` : Output from enterprisedb crontab -l, if enterprisedb user
- File `/linux/postgres_crontab.data` : Output from postgres crontab -l, if postgres user

Depth

Surface

Security impact

Low — Might have entries in crontab/cron with sensitive data.

repmgr CLI (repmgr_cli)

Description

Collects output of `repmgr cluster crosscheck` and `repmgr daemon status` using the `repmgr.conf` file that's found in the server

Report output

- File `/tools/repmgr/cli/daemon_status.out` : Output of `repmgr daemon status -f <conf_file_path>` command
- File `/tools/repmgr/cli/cluster_crosscheck.out` : Output of `repmgr cluster crosscheck -f <conf_file_path>` command

Depth

Surface

Security impact

Low — No known security impact.

repmgr configuration (repmgr_configuration)

Description

Collects `repmgr` configuration file that's found in the server

Report output

File `/tools/repmgr/config/repmgr.conf` : repmgr configuration file

Depth

Surface

Security impact

Low — No known security impact.

repmgr systemctl (repmgr_systemctl)

Description

When repmgr services are detected, collects status and cat of the corresponding services. Checks for any service whose name starts with repmgr .

Report output

- File `/tools/repmgr/systemd/<service_name>_cat.data` : Output of 'systemctl cat
- File `/tools/repmgr/systemd/<service_name>_status.data` : Output of 'systemctl status

Depth

Surface

Security impact

Low — No known security impact.

xDB CLI (xdb_cli)

Description

xDB output from several CLI commands, from the running xDB publication and subscription server

Report output

- Directory `/tools/xdb/cli` : xDB CLI print commands

Depth

Surface

Security impact

Low — No known security impact.

xDB configuration (xdb_configuration)**Description**

xDB configuration files

Report output

- File /tools/xdb/config/xdbReplicationServer.config : xDB startup configuration
- File /tools/xdb/config/edb-repl.conf : xDB replication configuration
- File /tools/xdb/config/xdb_subserver.conf : xDB subscription server configuration
- File /tools/xdb/config/xdb_pubserver.conf : xDB publication server configuration

Depth

Surface

Security impact

Low — No known security impact.

xDB systemctl (xdb_systemctl)**Description**

When xDB services are detected, collects status and cat of edb- xdbpubserver and edb-xdbsubserver

Report output

- File /tools/xdb/systemd/<service_name>_cat.data : Output of systemctl cat <service_name>
- File /tools/xdb/systemd/<service_name>_status.data : Output of systemctl status <service_name>

Depth

Surface

Security impact

Low — No known security impact.

3.4.2 Microsoft Windows operating system ### PEM configuration - Windows (pem_configuration_windows)

Description

PEM configuration files from PEM agent, PEM server, and PEM web server in a Windows environment

Report output

- File /tools/pem/config/edb-ssl-pem.conf : PEM web server SSL configuration file
- File /tools/pem/config/edb-pem.conf : PEM web server configuration file
- File /tools/pem/config/pem.wsgi : PEM server WSGI definition file
- File /tools/pem/config/agent.cfg : PEM agent configuration file

Depth

Surface

Security impact

Low — No known security impact.

PEM sc (pem_sc)

Description

When PEM is detected, collects PEM agent and PEM web server status and content

Report output

File /tools/pem/sc/<service_name>_query.data : Output of sc query <service_name>

Depth

Surface

Security impact

Low — No known security impact.

Disk information (win_disk_information)**Description**

Disk and controller information from the system registry

Report output

- File `/windows/enum_ide.reg` : Local machine ide device settings
- File `/windows/enum_scsi.reg` : Local machine scsi device settings

Depth

Surface

Security impact

Low — No known security impact.

Hosts file (win_hosts)**Description**

Host files and network-related information

Report output

- File `/windows/services.data` : Windows services file
- File `/windows/protocol.data` : Windows protocol file
- File `/windows/networks.data` : Windows networks file
- File `/windows/hosts.sam` : Windows hosts.sam file
- File `/windows/hosts.data` : Windows hosts file

Depth

Surface

Security impact

Low — No known security impact.

MsInfo (win_msinfo)

Description

MsInfo32 report in NFO and TXT format

Report output

- File `/windows/msinfo_report.txt` : Information from the MsInfo32 in textual format
- File `/windows/msinfo_report.nfo` : Information from the MsInfo32 in NFO

Depth

Surface

Security impact

Low — No known security impact.

ODBC/64 (win_odbc32_info)

Description

ODBC configuration from the 64-bit registry section

Report output

- File `/windows/user_odbc_wow64.reg` : User DSN list
- File `/windows/localmachine_odbcinst_wow64.reg` : List of installed ODBC drivers
- File `/windows/localmachine_odbc_wow64.reg` : System DSN list

Depth

Surface

Security impact

Medium — ODBC connection information can expose the presence of other databases or connection information to Postgres that can be used to attack the system.

ODBC/32 (win_odbc64_info)**Description**

ODBC configuration from the 32-bit registry section

Report output

- File /windows/user_odbc.reg : User DSN list
- File /windows/localmachine_odbcinst.reg : List of installed ODBC drivers
- File /windows/localmachine_odbc.reg : System DSN list

Depth

Surface

Security impact

Medium — ODBC connection information can expose the presence of other databases or connection information to Postgres that can be used to attack the system.

systeminfo (win_systeminfo)**Description**

Output of the systeminfo command

Report output

File /windows/systeminfo_report.txt : Information from the systeminfo command

Depth

Surface

Security impact

Low — No known security impact.

Disk volumes (`win_volumes`)

Description

Volume list from WMI

Report output

- File `/windows/association_structure` : Association between drive letters and physical drives
- File `/windows/volume_disk` : Volume list from the WMI subsystem
- File `/windows/logical_disk_list` : Logical disk list from the WMI subsystem
- File `/windows/disk_partition_list` : Disk partition list from the WMI subsystem
- File `/windows/disk_drive_list` : Disk list from the WMI subsystem

Depth

Surface

Security impact

Low — No known security impact.

xDB CLI - Windows (`xdb_cli_windows`)

Description

xDB output from several CLI commands, from the running xDB publication and subscription server

Report output

- Directory `/tools/xdb/cli` : xDB CLI print commands

Depth

Surface

Security impact

Low — No known security impact.

xDB configuration - Windows (xdb_configuration_windows)**Description**

xDB configuration files

Report output

- File /tools/xdb/config/xdbReplicationServer.config : xDB startup configuration`
- File /tools/xdb/config/edb-repl.conf : xDB replication configuration`
- File /tools/xdb/config/xdb_subserver.conf : xDB subscription server configuration
- File /tools/xdb/config/xdb_pubserver.conf : xDB publication server configuration

Depth

Surface

Security impact

Low — No known security impact.

xDB sc (xdb_sc)**Description**

When xDB is detected, collects xDB Publication and Subscription server status

Report output

File /tools/xdb/sc/<service_name>_query.data : Output of 'sc query

Depth

Surface

Security impact

Low — No known security impact.

3.4.3 PostgreSQL/PGD3 nstance ### Current archiver stats (postgresql_archiver)

Description

Statistics about the archiver process's activity from pg_stat_archiver

Report output

File postgresql/archiver.out

Depth

Surface

Security impact

Low — No known security impact.

Available extensions (postgresql_available_extensions)

Description

List of extensions available on the server

Report output

File postgresql/available_extensions.out

Depth

Surface

Security impact

Low — No known security impact.

Current bg_writer stats (postgresql_bgwriter)**Description**

Statistics about the background writer process's activity from pg_stat_bgwriter

Report output

File postgresql/bgwriter.out

Depth

Surface

Security impact

Low — No known security impact.

Directory with binaries (postgresql_bin_dir)**Description**

PostgreSQL binary directory

Report output

File /postgresql/postgresql_bin_path.data : Path to the Postgres bin directory

Depth

Surface

Security impact

Low — No known security impact.

Current configuration (postgresql_configuration)

Description

Postgres current configuration

Report output

File postgresql/configuration.out

Depth

Surface

Security impact

Medium — postgresql.conf might contain bad security policies

Configuration files (postgresql_configuration_files)

Description

Postgres configuration files and the data directory path. Passwords contained in well-known connection strings are redacted for information security reasons.

Report output

- File /postgresql/pg_ident.conf : Postgres ident configuration file
- File /postgresql/pg_hba.conf : Postgres host-based authentication file
- File /postgresql/postgresql.auto.conf : Postgres auto configuration file
- File /postgresql/recovery.done : Postgres recovery.done file
- File /postgresql/recovery.conf : Postgres recovery.conf file
- File /postgresql/postgresql.conf : Postgres configuration file

Depth

Surface

Security impact

Medium — `pg_hba.conf` and `pg_ident.conf` might expose potential security holes, such as trusted connections.

Databases (postgresql_databases)**Description**

List of databases in the Postgres node

Report output

File `postgresql/databases.out`

Depth

Surface

Security impact

Low — No known security impact.

postgresql_db_bdr_tables_and_views (postgresql_db_bdr_tables_and_views)**Description**

Collects all the tables and views of the BDR extension, except for:

- `bdr.apply_log`
- `bdr.conflict_history`
- `bdr.consensus_kv_data`
- `bdr.internal_node_pre_commit`
- `bdr.replication_status`
- `bdr.state_journal`
- `bdr.stat_activity`

Report output

File `/postgresql/dbs/<dbname>/bdr/*` : Content of all tables under the BDR schema

Depth

Shallow

Security impact

Low — No known security impact.

postgresql_db_pglogical_tables_and_views (postgresql_db_pglogical_tables_and_views)

Description

Collects all the tables and views of the pglogical extension

Report output

File /postgresql/dbs/<dbname>/pglogical/* : Content of all tables under the pglogical schema

Depth

Shallow

Security impact

Low — No known security impact.

Database/Role Setting (postgresql_db_role_setting)

Description

List of database/role settings in the Postgres node

Report output

File postgresql/db_role_setting.out

Depth

Shallow

Security impact

Low — No known security impact.

Node and snapshot data (postgresql_node)**Description**

Information about the running Postgres node

Report output

File postgresql/node.out

Depth

Surface

Security impact

Low — No known security impact.

pg_config (postgresql_pg_config)**Description**

Postgres pg_config command output

Report output

File /postgresql/pg_config.data : pg_config command output

Depth

Surface

Security impact

Low — No known security impact.

pg_controldata (postgresql_pg_controldata)

Description

Postgres pg_controldata information

Report output

File /postgresql/pg_controldata.data : pg_controldata command output

Depth

Surface

Security impact

Low — No known security impact.

Version (postgresql_pg_version)

Description

Postgres client and server version

Report output

- File /postgresql/postgresql_server_version.data : Postgres server version
- File /postgresql/postgresql_client_version.data : Postgres client version

Depth

Surface

Security impact

Low — No known security impact.

Current pg_prepared_xacts contents (postgresql_prepared_xacts)

Description

Status of prepared xacts from pg_prepared_xacts

Report output

File postgresql/prepared_xacts.out

Depth

Surface

Security impact

Low — No known security impact.

Current pg_replication_origin_status contents (postgresql_replication_origin)**Description**

Status of replication origins from pg_replication_origin_status

Report output

File postgresql/replication_origins.out

Depth

Surface

Security impact

Low — No known security impact.

Current pg_replication_slots contents (postgresql_replication_slots)**Description**

Replication slots from pg_replication_slots

Report output

File postgresql/replication_slots.out

Depth

Surface

Security impact

Low — No known security impact.

Roles (postgresql_roles)

Description

Database roles from `pg_roles`

Report output

File `postgresql/roles.out`

Depth

Shallow

Security impact

Medium — `pg_roles` might contain bad security policies

Current activity stats (postgresql_running_activity)

Description

Information related to the current activity on running processes from `pg_stat_activity`

Report output

File `postgresql/running_activity.out`

Depth

Shallow

Security impact

Low — queries in `pg_stat_activity` could contain user names and application names.

Active locks (postgresql_running_locks)**Description**

List of active locks

Report output

File `postgresql/running_locks.out`

Depth

Surface

Security impact

Low — No known security impact.

Current pg_shmem_allocations contents (postgresql_shmem_allocations)**Description**

Status of shared memory allocations from `pg_shmem_allocations`

Report output

File `postgresql/shmem_allocations.out`

Depth

Surface

Security impact

Low — No known security impact.

Current pg_stat_replication contents (postgresql_stat_replication)

Description

Replication connections from pg_stat_replication

Report output

File postgresql/replication.out

Depth

Surface

Security impact

Low — No known security impact.

Server subscription statistics (postgresql_subscription_statistics)

Description

Statistics of subscriptions

Report output

File postgresql/subscription_statistics.out

Depth

Shallow

Security impact

Low — No known security impact.

Server subscriptions (postgresql_subscriptions)

Description

List of subscriptions

Report output

File postgresql/subscriptions.out

Depth

Shallow

Security impact

Low — No known security impact.

Tablespaces (postgresql_tablespace)**Description**

Tablespaces information and location

Report output

File postgresql/tablespaces.out

Depth

Surface

Security impact

Low — No known security impact.

**3.4.4 Details for every Postgres/BDR3 database ### PGD1 replication slots
(postgresql_db_bdr1_replication_slots)****Description**

List of replication slots with 9.6 format for PGD1

Report output

File bdr1_replication_slots.out

Depth

Surface

Security impact

Low — No known security impact.

PGD2 replication slots (postgresql_db_bdr2_replication_slots)

Description

List of replication slots with 9.6 format for PGD2

Report output

File bdr2_replication_slots.out

Depth

Surface

Security impact

Low — No known security impact.

PGD conflict_history_summary aggregation (postgresql_db_bdr3_conflict_history_summary_agg)

Description

Collects aggregate count for all types of conflicts

Report output

File bdr_conflict_history_summary_agg.out

Depth

Surface

Security impact

Low — No known security impact.

PGD current activity stats (postgresql_db_bdr3_stat_activity)**Description**

Information related to the current activity on running processes from `bdr.stat_activity`

Report output

File `bdr_stat_activity.out`

Depth

Shallow

Security impact

Low — queries in `bdr.stat_activity` could contain user names and application names.

PGD sequences (postgresql_db_bdr_sequences)**Description**

List of the PGD sequences

Report output

File `bdr_sequences.out`

Depth

Surface

Security impact

Low — No known security impact.

PGD version (postgresql_db_bdr_version)

Description

Currently used version of PGD

Report output

File bdr_version.out

Depth

Surface

Security impact

Low — No known security impact.

Database extensions (postgresql_db_extensions)

Description

List of extensions in the database

Report output

File extensions.out

Depth

Shallow

Security impact

Low — No known security impact.

Database indexes (postgresql_db_indexes)

Description

List of indexes in the database

Report output

File `indexes.out`

Depth

Shallow

Security impact

Low — No known security impact.

Database procedural languages (postgresql_db_languages)**Description**

Procedural languages in the database

Report output

File `language.out`

Depth

Shallow

Security impact

Low — No known security impact.

PGD monitor_group_raft (postgresql_db_monitor_group_raft)**Description**

Check the raft status in the bdr cluster

Report output

File `bdr_monitor_group_raft.out`

Depth

Surface

Security impact

Low — No known security impact.

PGD monitor_group_versions (postgresql_db_monitor_group_versions)

Description

Check the version of all PGD nodes

Report output

File `bdr_monitor_group_versions.out`

Depth

Surface

Security impact

Low — No known security impact.

PGD monitor_local_replslots (postgresql_db_monitor_local_replslots)

Description

Checks all the replication slot status

Report output

File `bdr_monitor_local_replslots.out`

Depth

Surface

Security impact

Low — No known security impact.

Database operators (postgresql_db_operators)**Description**

Operators in the database

Report output

File operator.out

Depth

Shallow

Security impact

Low — No known security impact.

Database partitioned tables (postgresql_db_partitioned_tables)**Description**

Information about partitioned tables using declarative partitioning

Report output

File partitioned_table.out

Depth

Shallow

Security impact

Low — No known security impact.

Database PEM pem.agent table tuples (postgresql_db_pem_agent)

Description

Data from metatable pem.agent

Report output

File pem_agent.out

Depth

Shallow

Security impact

Low — No known security impact.

Database PEM pem.agent_config table tuples (postgresql_db_pem_agent_config)

Description

Data from metatable pem.agent_config

Report output

File pem_agent_config.out

Depth

Shallow

Security impact

Low — No known security impact.

Database PEM pem.agent_heartbeat table tuples (postgresql_db_pem_agent_heartbeat)

Description

Data from metatable pem.agent_heartbeat

Report output

File pem_agent_heartbeat.out

Depth

Shallow

Security impact

Low — No known security impact.

Database PEM pem.agent_server_binding table tuples (postgresql_db_pem_agent_server_binding)

Description

Data from metatable pem.agent_server_binding

Report output

File pem_agent_server_binding.out

Depth

Shallow

Security impact

Low — No known security impact.

Database PEM pem.config table tuples (postgresql_db_pem_config)

Description

Data from metatable pem.config

Report output

File pem_config.out

Depth

Shallow

Security impact

Low — No known security impact.

Database PEM pem.email_group table tuples (postgresql_db_pem_email_group)

Description

Data from metatable pem.email_group

Report output

File pem_email_group.out

Depth

Shallow

Security impact

Low — No known security impact.

Database PEM pem.email_group_option table tuples (postgresql_db_pem_email_group_option)

Description

Data from metatable pem.email_group_option

Report output

File pem_email_group_option.out

Depth

Shallow

Security impact

Low — No known security impact.

Database PEM pem.probe table tuples (postgresql_db_pem_probe)**Description**

Data from metatable pem.probe

Report output

File pem_probe.out

Depth

Shallow

Security impact

Low — No known security impact.

Database PEM pem.probe_schedule table tuples (postgresql_db_pem_probe_schedule)**Description**

Data from metatable pem.probe_schedule

Report output

File pem_probe_schedule.out

Depth

Shallow

Security impact

Low — No known security impact.

Database PEM pem.schema_version() output (postgresql_db_pem_schema_version)

Description

Output from function `pem.schema_version()`

Report output

File `pem_schema_version.out`

Depth

Shallow

Security impact

Low — No known security impact.

Database PEM pem.server table tuples (postgresql_db_pem_server)

Description

Data from metatable `pem.server`

Report output

File `pem_server.out`

Depth

Shallow

Security impact

Low — No known security impact.

Database PEM pem.server_heartbeat table tuples (postgresql_db_pem_server_heartbeat)

Description

Data from metatable `pem.server_heartbeat`

Report output

File pem_server_heartbeat.out

Depth

Shallow

Security impact

Low — No known security impact.

Database PEM pem.smtp_spool table tuples (postgresql_db_pemsmtp_spool)**Description**

Data from metatable pem.smtp_spool

Report output

File pemsmtp_spool.out

Depth

Shallow

Security impact

Low — No known security impact.

Database PEM pem.snmp_spool table tuples (postgresql_db_pemsnmp_spool)**Description**

Data from metatable pem.snmp_spool

Report output

File pemsnmp_spool.out

Depth

Shallow

Security impact

Low — No known security impact.

Pglogical subscription status (postgresql_db_pglogical_subscription_status)

Description

List of tables replicated by pglogical

Report output

File pglogical_subscription_status.out

Depth

Surface

Security impact

Low — No known security impact.

Database functions (postgresql_db_procs)

Description

Functions in the database

Report output

File proc.out

Depth

Shallow

Security impact

Low — No known security impact.

Database publication tables (postgresql_db_publication_tables)**Description**

List of tables of publications of the database

Report output

File `publication_tables.out`

Depth

Shallow

Security impact

Low — No known security impact.

Database publications (postgresql_db_publications)**Description**

List of publications of the database

Report output

File `publications.out`

Depth

Shallow

Security impact

Low — No known security impact.

Database repmgr repmgr.events table tuples (postgresql_db_repmgr_events)

Description

Data from metatable repmgr.events

Report output

File repmgr/events.out

Depth

Shallow

Security impact

Low — No known security impact.

Database repmgr repmgr.monitoring_history table tuples (postgresql_db_repmgr_monitoring_history)

Description

Data from metatable repmgr.monitoring_history

Report output

File repmgr/monitoring_history.out

Depth

Shallow

Security impact

Low — No known security impact.

Database repmgr repmgr.nodes table tuples (postgresql_db_repmgr_nodes)

Description

Data from metatable repmgr.nodes

Report output

File `repmgr/nodes.out`

Depth

Shallow

Security impact

Low — No known security impact.

Database `repmgr` `repmgr.replication_status` table tuples (`postgresql_db_repmgr_replication_status`)

Description

Data from metatable `repmgr.replication_status`

Report output

File `repmgr/replication_status.out`

Depth

Shallow

Security impact

Low — No known security impact.

Database `repmgr` `repmgr.show_nodes` table tuples (`postgresql_db_repmgr_show_nodes`)

Description

Data from metatable `repmgr.show_nodes`

Report output

File `repmgr/show_nodes.out`

Depth

Shallow

Security impact

Low — No known security impact.

Database schema (postgresql_db_schemas)

Description

List of schemas in the database

Report output

File schemas.out

Depth

Shallow

Security impact

Low — No known security impact.

Database statistics (postgresql_db_statistics)

Description

Statistics of the database

Report output

File statistics.out

Depth

Deep

Security impact

Low — No known security impact.

Database subscription tables (postgresql_db_subscription_tables)**Description**

List of tables of subscriptions of the database

Report output

File subscription_tables.out

Depth

Shallow

Security impact

Low — No known security impact.

Database tables (postgresql_db_tables)**Description**

List of tables in the database

Report output

File tables.out

Depth

Shallow

Security impact

Low — No known security impact.

Database types (postgresql_db_types)

Description

Types in the database

Report output

File type.out

Depth

Shallow

Security impact

Low — No known security impact.

Database	xDB	_edb_replicator_pub.xdb_mmr_pub_group	table	tuples
(postgresql_db_xdb_mmr_pub_group)				

Description

Data from metatable _edb_replicator_pub.xdb_mmr_pub_group

Report output

File xdb_mmr_pub_group.out

Depth

Shallow

Security impact

Low — No known security impact.

Database	xDB	_edb_replicator_pub.xdb_pub_database	table	tuples
(postgresql_db_xdb_pub_database)				

Description

Data from metatable _edb_replicator_pub.xdb_pub_database

Report output

File xdb_pub_database.out

Depth

Shallow

Security impact

Low — No known security impact.

Database xDB _edb_replicator_pub.xdb_pub_replug table tuples (postgresql_db_xdb_pub_replug)

Description

Last 50 rows from metatable _edb_replicator_pub.xdb_pub_replug

Report output

File xdb_pub_replug.out

Depth

Shallow

Security impact

Low — No known security impact.

Database xDB _edb_replicator_pub.xdb_publication_subscriptionstable tuples (postgresql_db_xdb_publication_subscriptions)

Description

Data from metatable _edb_replicator_pub.xdb_publication_subscriptions

Report output

File xdb_publication_subscriptions.out

Depth

Shallow

Security impact

Low — No known security impact.

Database `xDB_edb_replicator_pub.xdb_publications` **table tuples** (`postgresql_db_xdb_publications`)

Description

Data from metatable `_edb_replicator_pub.xdb_publications`

Report output

File `xdb_publications.out`

Depth

Shallow

Security impact

Low — No known security impact.

Database `xDB_edb_replicator_pub.rrep_mmr_pub_group` **table tuples** (`postgresql_db_xdb_rrep_mmr_pub_group`)

Description

Data from metatable `_edb_replicator_pub.rrep_mmr_pub_group`

Report output

File `xdb_rrep_mmr_pub_group.out`

Depth

Shallow

Security impact

Low — No known security impact.

Database `xDB_edb_replicator_pub.rrep_mmr_txset` table tuples (`postgresql_db_xdb_rrep_mmr_txset`)

Description

Last 10 rows from metatable `_edb_replicator_pub.rrep_mmr_txset`

Report output

File `xdb_rrep_mmr_txset.out`

Depth

Shallow

Security impact

Low — No known security impact.

Database `xDB_edb_replicator_pub.rrep_properties` table tuples (`postgresql_db_xdb_rrep_properties`)

Description

Data from from metatable `_edb_replicator_pub.rrep_properties`

Report output

File `xdb_rrep_properties.out`

Depth: **** shallow**

Security impact

Low — No known security impact.

Database **xDB** **_edb_replicator_pub.rrep_publication_tablestable** **tuples**
(postgresql_db_xdb_rrep_publication_tables)

Description

Data from metatable _edb_replicator_pub.rrep_publication_tables

Report output

File xdb_rrep_publication_tables.out

Depth

Shallow

Security impact

Low — No known security impact.

Database xDB _edb_replicator_pub.rrep_txset table tuples (postgresql_db_xdb_rrep_txset)

Description

Data from metatable _edb_replicator_pub.rrep_txset

Report output

File xdb_pub_rrep_txset.out

Depth

Shallow

Security impact

Low — No known security impact.

3.4.5 Barman (Backup and Recovery Manager) ### Barman check (barman_check)

Description

Collects the status of the Barman check framework for all configured servers

Report output

File `/barman/barman_check.data` : Output of `barman check`

Depth

Surface

Security impact

Low — No known security impact.

Barman diagnose (`barman_diagnose`)**Description**

Collects the barman diagnosis information

Report output

File `/barman/diagnose.data` : Output from ‘barman diagnose’

Depth

Surface

Security impact

Low — No known security impact.

Barman executable location (`barman_executable`)**Description**

Collects the barman executable location

Report output

File `/barman/barman_location.data` : The path of the main barman executable

Depth

Surface

Security impact

Low — No known security impact.

Barman module location (`barman_modules_path`)

Description

Collects the location of the barman Python modules

Report output

File `/barman/barman_python_verbose.data` : The list of Barman python modules

Depth

Surface

Security impact

Low — No known security impact.

3.5 Servers accepting upload of reports

Note: This information applies only to Lasso executables with external network access. You can verify that your Lasso can access the internet using the `--version` option. Look for the line *External network access enabled*.

The Lasso `--upload` command sends the tarball to one of the front-line application servers that are configured in high availability and load balancing to receive Lasso reports for EDB as part of the Customer Portal.

These servers are part of an ISO 27K1 infrastructure and fully comply with the European Union's General Data Protection Regulation (GDPR).

Provide this list of servers to your infrastructure team so that they can open outbound connections to port 443 of the following IPv4 addresses:

1. 78.46.204.124
2. 88.99.39.37
3. 88.99.39.42
4. 88.99.80.86
5. 116.203.69.173
6. 116.203.69.177

7. 116.203.69.179
8. 116.203.69.185
9. 116.203.69.248
10. 116.203.69.253
11. 159.69.178.25
12. 159.69.178.96
13. 159.69.181.84
14. 159.69.181.86
15. 195.201.138.157
16. 195.201.99.62

If you also use IPv6, add this list:

1. 2a01:4f8:1c17:482e::1
2. 2a01:4f8:1c17:4f1a::1
3. 2a01:4f8:1c17:5197::1
4. 2a01:4f8:1c17:529d::1
5. 2a01:4f8:c010:1210::1
6. 2a01:4f8:c010:154c::1
7. 2a01:4f8:c010:1550::1
8. 2a01:4f8:c010:1551::1
9. 2a01:4f8:1c1c:c82e::1
10. 2a01:4f8:1c1c:c836::1
11. 2a01:4f8:1c1c:c837::1
12. 2a01:4f8:1c1c:c95f::1
13. 2a01:4f8:1c1c:c97e::1
14. 2a01:4f8:1c1c:ca43::1
15. 2a01:4f8:1c1c:ca4e::1
16. 2a01:4f8:1c1c:dab3::1