HEXACLUSTER

PostgreSQL Security Assessment

FOSSASIA – Hanoï 2024

Purpose of this Talk

Checks to ensure that the security policies applies to your PostgreSQL Cluster.

- Operating and file system
- PostgreSQL configuration
- Logs / Audit
- Access / Objects permissions
- Encryption …

How to automate these checks as much as possible?

Gilles DAROLD

CTO at HexaCluster Corp / PostgreSQL Expert

Author of Ora2Pg, pgBadger, pgFormatter, pgCluu, ….

HexaCluster Corp.

Database migration to PostgreSQL specialist and PostgreSQL support.

Official sponsor for support and development for Ora2Pg, pgBadger and others tools

- https://github.com/darold/
- https://github.com/hexacluster/
- https://github.com/MigOpsRepos/

Contact: https://hexacluster.ai/contact-us/

How to Secure PostgreSQL?

PostgreSQL Security Checks

A long list of items to check for PostgreSQL security. The majority of the security checks have been listed and well described.

Center for Internet Security (CIS) Benchmark PostgreSQL

- https://www.cisecurity.org/cis-benchmarks

Security Technical Implementation Guides (STIGs) from DISA

- https://public.cyber.mil/stigs/downloads

Data Security Checks

European GDPR data protection requirements:

- To hide some data elements
- Data masking for certain groups of users (dev, 3rd parties, ...)
- Data encryption (card number, etc.)

Not covered yet.

Operating System

Operating System Level

- Ensure packages are obtained from authorized repositories
- Ensure systemd service files are enabled
- Ensure PostgreSQL versions are up-to-date
- Ensure sudo is configured correctly

File System

File System Level

- Ensure Data Cluster Initialized correctly
- Ensure Data Cluster have checksum enabled
- Ensure PGDATA, WALs and temporary files are not on the same partition
- Ensure that the PGDATA partition is encrypted
- Check directory and files permissions

PostgreSQL Configuration

PostgreSQL Configuration Level

- Ensure backend runtime parameters are configured correctly
- Ensure others runtime parameters are configured conforming to your security policy
    - Postmaster runtime parameters
    - SIGHUP runtime parameters
    - Superuser runtime parameters
    - User runtime parameters

Logs / Audit

Log / Audit Level

- Ensure all log settings are well configured
- Ensure connections / disconnections are traced
- Ensure pgBadger can be used for deep analyze and reporting
- Ensure that log_directory is outside the PGDATA
- Ensure the PostgreSQL Audit Extension (pgAudit) is enabled

Access

HEXA CLUSTER

Connection and Logging

- Ensure Password Complexity is configured
- Ensure authentication timeout and delay are configured
- Ensure specific database and users are used
- Ensure superusers are not allowed to connect remotely
- Ensure pg_hba.conf is well configured

Objects Permissions

User Access and Authorization

- Ensure excessive administrative privileges are revoked
- Ensure excessive function privileges are revoked
- Ensure excessive DML privileges are revoked
- Ensure Row Level Security (RLS) is configured correctly
- Make use of predefined roles

Encryption

Encryption Level

- Ensure that the PGDATA partition is encrypted
- Ensure SSL / TLS is enabled and well configured
- Ensure FIPS 140-2 OpenSSL cryptography is used (how to enable it)
- Ensure a cryptographic extension is installed (pgcrypto or pgsodium)

Introducing PGDSAT

HEXACLUSTER

PostgreSQL Database Security Assessment Tool

Standalone Perl Script

- No dependencies
- A single program to collect information and build reports
- Support multi language (English, French)
- HTML or text reports
- Around 80 security checks

Example of Use

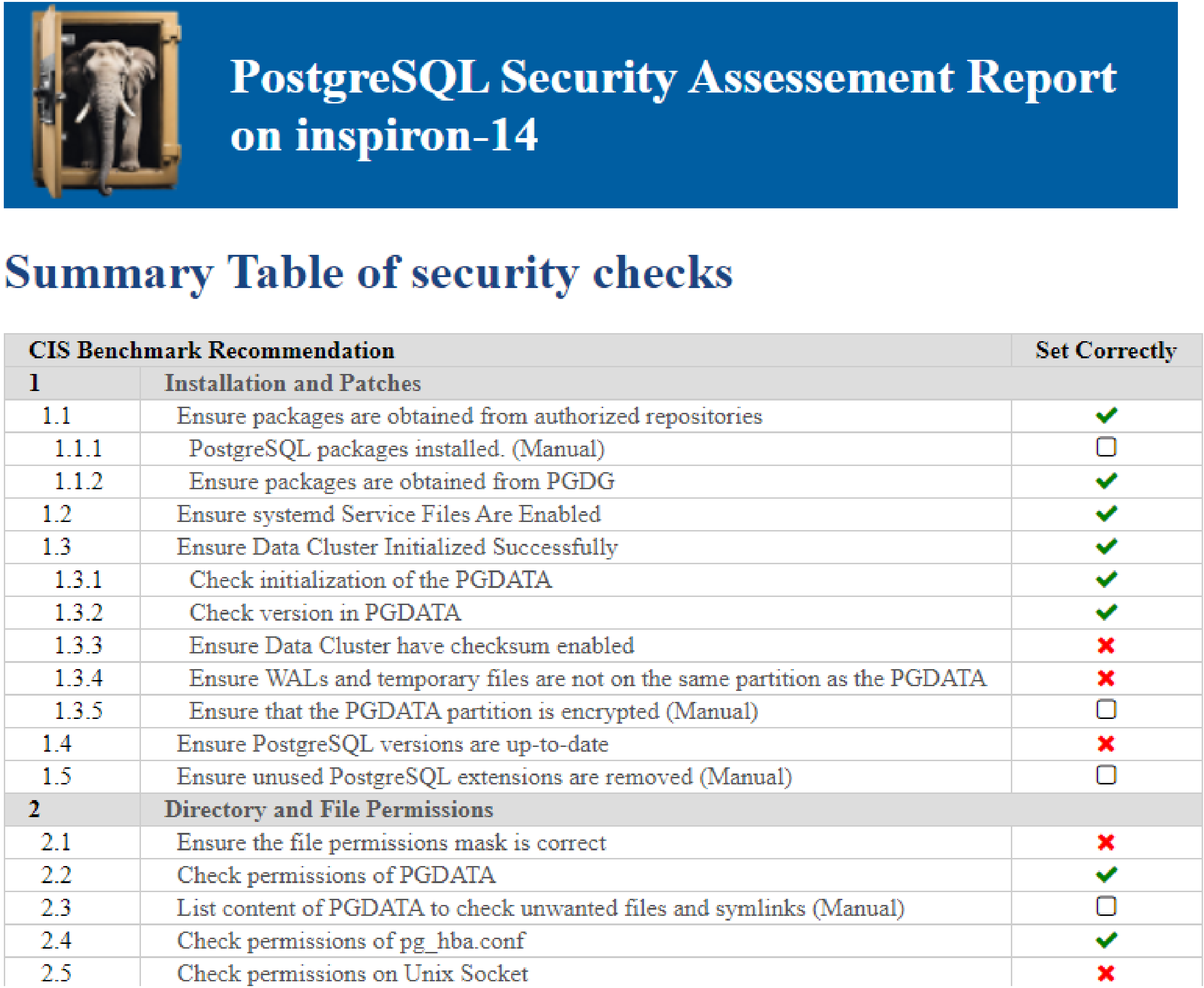
```
$ sudo su – postgres

$ pgstat > report.text

$ pgdsat -U postgres -h localhost -d postgres -f html > report.html

$ pgdsat -U postgres -h localhost -d postgres -o report.html
```

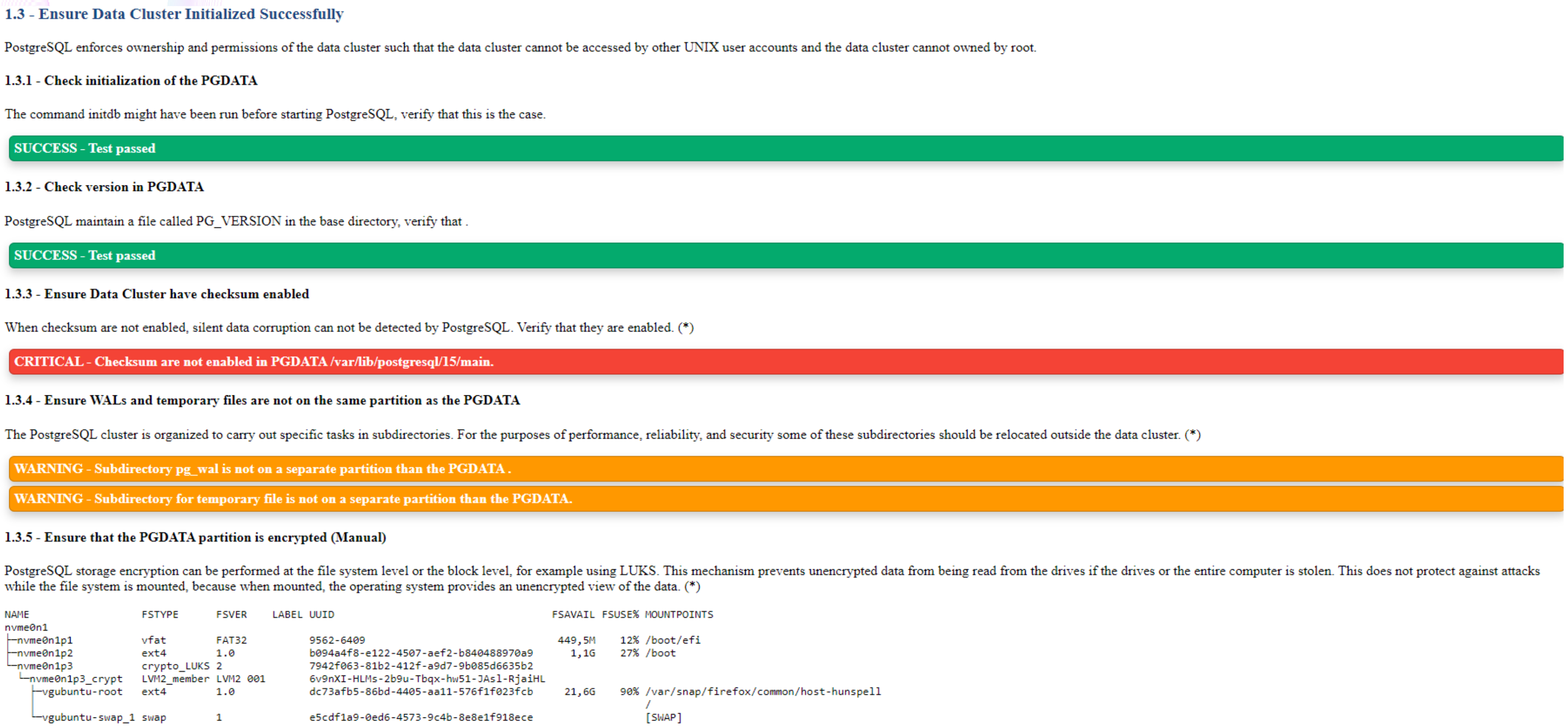
PGDSAT use psql to query PostgreSQL, you can use .pgpass and the PG* environment variables.



PostgreSQL Security Assesement Report on inspiron-14

Summary Table of security checks

CIS Benchmark Recommendation		Set Correctly
1	**Installation and Patches**	
1.1	Ensure packages are obtained from authorized repositories	✓
1.1.1	PostgreSQL packages installed. (Manual)	□
1.1.2	Ensure packages are obtained from PGDG	✓
1.2	Ensure systemd Service Files Are Enabled	✓
1.3	Ensure Data Cluster Initialized Successfully	✓
1.3.1	Check initialization of the PGDATA	✓
1.3.2	Check version in PGDATA	✓
1.3.3	Ensure Data Cluster have checksum enabled	✗
1.3.4	Ensure WALs and temporary files are not on the same partition as the PGDATA	✗
1.3.5	Ensure that the PGDATA partition is encrypted (Manual)	□
1.4	Ensure PostgreSQL versions are up-to-date	✗
1.5	Ensure unused PostgreSQL extensions are removed (Manual)	□
2	**Directory and File Permissions**	
2.1	Ensure the file permissions mask is correct	✗
2.2	Check permissions of PGDATA	✓
2.3	List content of PGDATA to check unwanted files and symlinks (Manual)	□
2.4	Check permissions of pg_hba.conf	✓
2.5	Check permissions on Unix Socket	✗



1.3 - Ensure Data Cluster Initialized Successfully

PostgreSQL enforces ownership and permissions of the data cluster such that the data cluster cannot be accessed by other UNIX user accounts and the data cluster cannot owned by root.

1.3.1 - Check initialization of the PGDATA

The command initdb might have been run before starting PostgreSQL, verify that this is the case.

SUCCESS - Test passed

1.3.2 - Check version in PGDATA

PostgreSQL maintain a file called PG_VERSION in the base directory, verify that .

SUCCESS - Test passed

1.3.3 - Ensure Data Cluster have checksum enabled

When checksum are not enabled, silent data corruption can not be detected by PostgreSQL. Verify that they are enabled. (*)

CRITICAL - Checksum are not enabled in PGDATA /var/lib/postgresql/15/main.

1.3.4 - Ensure WALs and temporary files are not on the same partition as the PGDATA

The PostgreSQL cluster is organized to carry out specific tasks in subdirectories. For the purposes of performance, reliability, and security some of these subdirectories should be relocated outside the data cluster. (*)

WARNING - Subdirectory pg_wal is not on a separate partition than the PGDATA .

WARNING - Subdirectory for temporary file is not on a separate partition than the PGDATA.

1.3.5 - Ensure that the PGDATA partition is encrypted (Manual)

PostgreSQL storage encryption can be performed at the file system level or the block level, for example using LUKS. This mechanism prevents unencrypted data from being read from the drives if the drives or the entire computer is stolen. This does not protect against attacks while the file system is mounted, because when mounted, the operating system provides an unencrypted view of the data. (*)

```
NAME                FSTYPE      FSVER    LABEL UUID                                   FSAVAIL FSUSE% MOUNTPOINTS
nvme0n1
├─nvme0n1p1         vfat        FAT32          9562-6409                               449,5M    12% /boot/efi
├─nvme0n1p2         ext4        1.0            b094a4f8-e122-4507-aef2-b840488970a9      1,1G    27% /boot
└─nvme0n1p3         crypto_LUKS 2              7942f063-81b2-412f-a9d7-9b085d6635b2
  └─nvme0n1p3_crypt LVM2_member LVM2 001       6v9nXI-HLMs-2b9u-Tbqx-hw51-JAsl-RjaiHL
    ├─vgubuntu-root ext4        1.0            dc73afb5-86bd-4405-aa11-576f1f023fcb     21,6G    90% /var/snap/firefox/common/host-hunspell
    │                                                                                                /
    └─vgubuntu-swap_1 swap      1              e5cdf1a9-0ed6-4573-9c4b-8e8e1f918ece                  [SWAP]
```

HTML / TEXT Reports

- HTML report: https://www.darold.net/sample_pgdsat/report.html
- Text report: https://www.darold.net/sample_pgdsat/report.txt

TODO List

The Wish List

- Add report for GDPR rules / Data securing
- Implement SQL injection checks
- Implements an extension that can act as a SQL firewall
- Check in detail the configuration of credcheck and other extensions
- Add a JSON output format for processing by other tools
- Implements an incremental mode to detect drift between two runs
- A minimal version for Cloud? … and your feature requests …

Contributing

Feedbacks matter

The objective is to have a common and free tool to help the PostgreSQL users to enforce the security of their PostgreSQL clusters.

- Bug reports: https://github.com/HexaCluster/pgdsat/issues
- Feature requests: https://github.com/HexaCluster/pgdsat/issues
- Patches: https://github.com/HexaCluster/pgdsat/pulls
- Download: https://github.com/hexacluster/pgdsat

Thanks !

https://github.com/hexacluster/pgdsat

Any Questions?