CSPN Security Target

KeePassXC CSPN

Version 1.7

Document identification

Document specifications

Object	CSPN Security Target – KeePassXC CSPN
Number of pages	15
Diffusion	PUBLIC

Document history

Version	Date	State
1.0	28/03/2024	First version (DRAFT)
1.1	06/05/2024	Adding exact dependencies in 2.4 Removing H4 Modifying TA1 Adding TA3, T7 and T8 Modifying SF3 and SF4 Adding SF6 and SF7
1.2	20/05/2024	Modifying S3 and TA2.
1.3	28/05/2024	Modifying SF6.
1.4	29/10/2024	KeepassXC version updated to 2.7.9 Adding H4, H5, H6 Adding 3.4 Attack surface Modifying Disclaimer
1.5	17/07/2025	Adding amd64 CPU requirement
1.6	11/09/2025	Removing T3 and renaming T8 to T3 Adding Argon2 algorithm details
1.7	28/10/2025	PUBLIC Classification

Table of contents

1. Introduction

1.1. Product identification

Product maintainer	KeePassXC Team (https://keepassxc.org/team/)
Product official website	https://keepassxc.org/
Commercial name of the product	*KeePassXC*
Evaluated version	*KeePassXC* 2.7.9
Product category	Stockage sécurisé

1.2. Document structure

This document is divided up into five parts (this introduction excluded) describing:

- The target of evaluation (TOE).
- The environment in which the product is evaluated.
- The sensitive assets that are to be protected by the product.
- The threats and the threat actors.
- The security features handling the threats previously described.

1.3. References

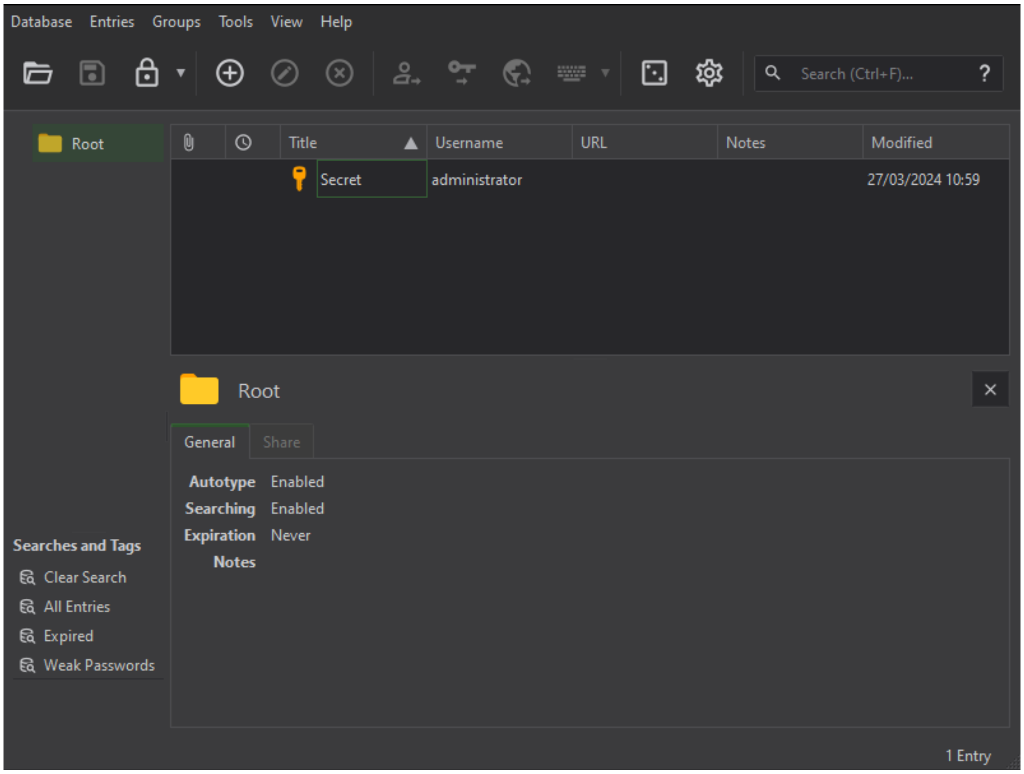
Identifier	Title	Reference	Version	Classification
CSPN	Certification de sécurité de premier niveau	ANSSI-CSPN-CER-P-01	1.1	Public
RGS_B_1	Rules and recommendations regarding usage of cryptographic mechanisms.	RGS_v-2-0_B1	2.0	Public

2. Product description

2.1. General description

KeePassXC is a modern, secure, and open-source password manager that stores and manages your most sensitive information.

You can run KeePassXC on Windows, macOS, and Linux systems. KeePassXC is for people with extremely high demands of secure personal data management. It saves many types of information, such as usernames, passwords, URLs, attachments, and notes in an offline, encrypted file that can be stored in any location, including private and public cloud solutions.[1]



Title	Username	URL	Notes	Modified
Secret	administrator			27/03/2024 10:59

Autotype Enabled
Searching Enabled
Expiration Never
Notes

Figure 1: KeePassXC UI once unlocked.

2.2. Product usage

The common usage of *KeePassXC* is the storage and usage of passwords. Passwords are stored in entries that can be organized according to the user and UI functions allows browsing and accessing them. Then, *KeePassXC* offers features to use the password, either by simply copying it in the clipboard or by automatically typing it in the previously focused window or in specific windows. *KeePassXC* can also generate passwords for users allowing them to never know an application' specific password.

To facilitate the usage of *KeePassXC*, a browser extension KeePassXC-Browser has been developed to easily interact with the main *KeePassXC* process and insert credentials according to the visited domain. This extension is available in multiple browsers: Google Chrome, Mozilla Firefox, Microsoft Edge and corresponding variants.

Other utilities are included in *KeePassXC*:

[1] https://keepassxc.org/#project

- Synchronisation of other database using KeeShare. This feature allows importing another existing database as a group in the original database tree. Changes are tracked using certificates in order to identify authors' changes.

- SSH-agent interaction. This feature configures *KeePassXC* to directly add or remove SSH private keys in a ssh-agent or a pagent process.

2.3. Environment description

KeePassXC is available on Windows, macOS and Linux systems and can also be compiled from the source code that is provided on its website. By design, *KeePassXC* is designed to work for single users especially given the sensitiveness of the information stored in KeePassXC databases. However, databases can be shared inside a same team and *KeePassXC* offers a feature allowing to merge different databases, rendering the asynchronous usage of the same database by multiple users possible. Its main use case remains for single user though.

2.4. Description of dependencies

The source code of *KeePassXC* is available to be built on multiple platforms while prebuild binaries are available for aforementioned operating systems, Linux distributions recompile and distributes their own versions. No specific dependencies are needed to run *KeePassXC*, binary version downloadable from the website are either statically compiled binaries or come with their owned linked libraries.

The dependencies and their version can be found in the file *vcpkg.json* in the *KeePassXC* GitHub repository:

```
argon2: 20190702
botan: 3.1.1
minizip: 1.3
libqrencode: 4.1.1
libusb: 1.0.26.11791
libxi: 1.8
libxtst: 1.2.4
qt5: 5.15.11
qt5-imageformats: 5.15.11
qt5-macextras: 5.15.11
qt5-svg: 5.15.11
qt5-tools: 5.15.11
qt5-translations: 5.15.11
qt5-wayland: 5.15.11
qt5-x11extras: 5.15.11
readline: 0#5
zlib: 1.3
```

2.5. Hypothesis on the environment

H1: Installation

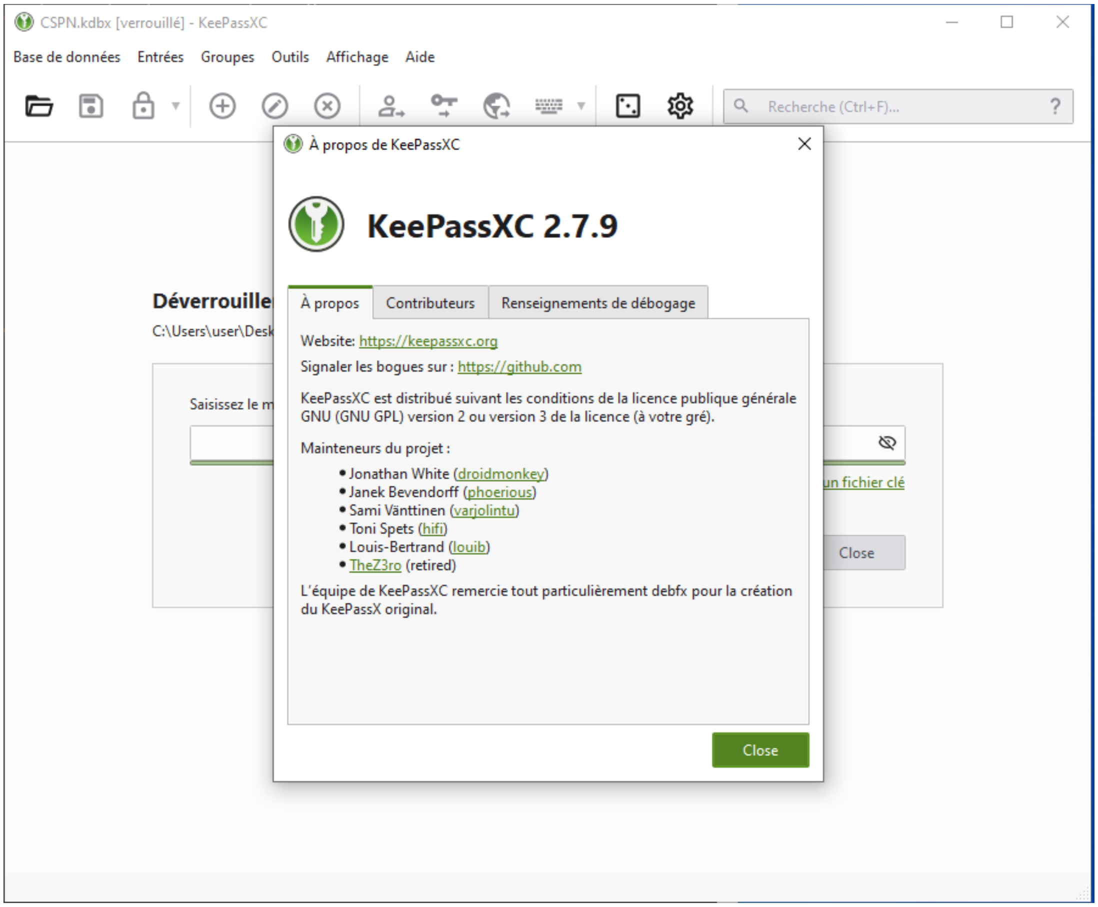
The TOE[2] is installed on a dedicated system considered healthy. This system runs Windows 10 on an amd64 CPU.

```
PS C:\> systeminfo
OS Name:                    Microsoft Windows Server 2022 Standard
OS Version:                 10.0.20348 N/A Build 20348
OS Manufacturer:            Microsoft Corporation
```

The *KeePassXC* binary is the one prebuilt and available from the developer's website: https://keepassxc.org at the following link:

```
https://github.com/keepassxreboot/keepassxc/releases/download/2.7.9/KeePassXC-2.7.9-Win64.msi
```

The version installed is the 2.7.9 which is accessible from the *about* submenu:



Website: https://keepassxc.org

Signaler les bogues sur : https://github.com

KeePassXC est distribué suivant les conditions de la licence publique générale GNU (GNU GPL) version 2 ou version 3 de la licence (à votre gré).

Mainteneurs du projet :

- Jonathan White (droidmonkey)
- Janek Bevendorff (phoerious)
- Sami Vänttinen (varjolintu)
- Toni Spets (hifi)
- Louis-Bertrand (louib)
- TheZ3ro (retired)

L'équipe de KeePassXC remercie tout particulièrement debfx pour la création du KeePassX original.

[2] Target Of Evaluation

The Windows version of *KeePassXC* comes with all the needed dependencies as linked libraries (Botan, Qt, etc.).

The integrity of the overall installer has been validated following *KeePassXC* tutorial at https://keepassxc.org/verifying-signatures/.

The extension KeePassXC-Browser is installed in a Google Chrome installable at the following link: https://chromewebstore.google.com/detail/keepassxc-browser/oboonakemofpalcgghocfoadofidjkkk. The version of the installed extension is 1.3.3 (26-03-2019).

Databases used to store passwords rely on the KDBX4 format.

H2: Administration

The system administrators are neither malicious nor hostile and are properly trained to operate the user system. They also follow administration best practices.

H3: Authentication

Authentication configured on the TOE rely on a master password and a master key file. Integration with operating system authentication mechanisms such as Windows Hello or macOS Touch ID are not considered. These mechanisms may be normally used for the *Quick Unlock* operation.

H4: Master key file protection

In case of a master key file is used to unlock the *KeepassXC* database, this master key file must be securely protected (Encryption, storage on an encrypted drive...).

H5: Execution environment

The operating system is supposed to be up-to-date and provide a protection against malware (Windows defender for example). In that context, a key-logger malware that records every keystroke entered by the trusted user is not considered a threat.

H6: Operating system protection

KeepassXC's process runs with low privileges on the operating system, thus limiting the risks for the host operating system.

2.6. Description of users and roles

KeePassXC is a single user application used and managed by a single user. Apart from the installation process, there is no need for administration privileges to run the TOE.

ID	Actor's description	Trust level
A.1	Connected user	Trustful

2.7. Evaluation scope

The evaluation scope includes the following elements:

- Check that the unlock and the decryption process of a database respect cryptography's best practices and prevent illegitimate access.

- Check that sensitive information stored and manipulated by *KeePassXC* are protected, for example, the clipboard is correctly cleared once a password has been paste or by preventing screenshots.

- Check that metadata of the database are correctly protected and do not leak in log files or by other mechanisms.

- Check the surface exposed by the enabling of the *browser integration* feature.

3. Description of the technical environment for the product to operate

3.1. Material prerequisite

There is no specific prerequisite.

3.2. Architecture selected

The selected architecture is Windows 10.0.20348 version with the following build number version: 20348.

3.3. Configuration

No specific configuration. As stated by H3, users do not used the *KeePassXC*'s Quick Unlock feature relying on Windows Hello.

3.4. Attack surface

The following table summarizes the attack surface of a KeepassXC instance running on Windows 10.

Attack surface type	Interfaces	Feature involved	Actors having access to this interface
Logical interfaces	Graphical User Interface	Display secret information (credentials, password..) **SF1**	Connected users (A.1) OS
	Chrome Browser's extension	Have access to KeepassXC secrets from the browser **SF5**	Connected users (A.1) Browser
	Keyboard input and clipboard	Clipboard clearance after delay **SF2** Auto-type in another window **SF2**	Connected users (A.1) OS
	File system	Database protections **SF3**	Connected users (A.1) OS
	Multi-device synchronisation software	Data sharing between devices (not evaluated), Keeshare segregation **SF6**	Connected users (A.1) OS
	SSH agent	SSH connection software (not evaluated), SSH agent software (not evaluated), ssh-agent interaction **SF7**	Connected users (A.1) OS
	Memory	Dump the process memory **SF4**	OS Local Administrator

4. Sensitive assets the product must protect

Sensitive assets that *KeePassXC* must protect are the following:

S1: Passwords stored

Passwords and similar sensitive information (passkey, TOTP seed) have to be protected in confidentiality and integrity.

S2: Entries metadata

Information regarding stored entries (present/absence of passwords, timestamp of last access/modification) protected in confidentiality and integrity.

S3: Master password, key file and intermediary keys

Information used to unlock a database, being either a master password and/or a master key file and the different intermediary keys used in the process of encryption and decryption, protected in confidentiality and integrity.

KeePassXC does not provide any protection for the master key file in terms of data integrity or confidentiality. The responsibility of securing the master key file lies entirely with the user.

5. Threats description

5.1. Threatening agents

The identified threatening agents are the following:

TA1: Attacker with access to the system where *KeePassXC* is running

An attacker with the capability to execute codes and commands in a session where *KeePassXC* is running. This attacker can dump the RAM, run keyloggers or screenshot programs.

TA2: Attacker with access to the database

The attacker has access to the database file. They may have access to the master key file as well, but only if the database is secured with both a passphrase and a master key file. If the database is protected by a master key file alone, it is presumed that the attacker does not have access to the file.

TA3: Attacker with control of an imported database using KeeShare

An attacker compromised or controls a database that is imported using KeeShare in another database.

5.2. Threats

The identified threats are the following:

T1: Take a screenshot or record the current screen to see passwords

TA1 executes a program taking regular screenshot or recording the whole screen to see plaintext passwords when generated or manipulated by the legitimate *KeePassXC* user.

T2: Memory dump of the *KeePassXC* process

TA1 dumps the memory of the *KeePassXC* process in order to retrieve sensitive information regarding entries, the master password or metadata.

T3: Stealing ssh private keys by exploiting the ssh-agent interaction

TA1 intercepts ssh private keys when *KeePassXC* interacts with either ssh-agent or pageant.

T4: Clipboard snooping

TA1 monitors the clipboard and retrieve its content before being cleared by *KeePassXC*.

T5: Offline database bruteforce

TA2 performs a bruteforce attacks on the offline database to retrieve the master password or the master keyfile.

T6: Instrumenting the browser integration interface

TA1 uses the browser integration interface to retrieve passwords and leak passkeys information from the *KeePassXC* process.

T7: Tampering or stealing entry information from an imported KeeShare database

TA3 exploits the fact that it controls a database imported in another database to corrupt or obtain information from the receiving database.

Threats on sensitive assets	
T1: Take a screenshot or record the current screen to see passwords	**S1**: Passwords stored **S2**: : Entries metadata **S3**: Master password and keyfile
T2: Memory dump of the KeePassXC process	**S1**: Passwords stored **S2**: : Entries metadata **S3**: Master password and keyfile
T3: Stealing ssh private keys by exploiting the ssh-agent interaction	**S1**: Passwords stored **S2**: Entries metadata
T4: Clipboard snooping	**S1**: Passwords stored
T5: Offline database bruteforce	**S3**: Master password and keyfile
T6: Instrumenting the browser integration interface	**S1**: Passwords stored **S2**: : Entries metadata
T7: Tampering or stealing entry information from an imported KeeShare database	**S1**: Passwords stored **S2**: Entries metadata

6. Security features

The TOE implements the following security features:

SF1: Anti-screenshot/recording

The *KeePassXC* window disappears when being recorded or screenshot.

SF2: Clipboard protection

KeePassXC implements two protections against clipboard snooping:

- The autotype feature that directly types the password's characters into the target window.
- The clearing of the clipboard after a certain timeframe.

SF3: Database protection

Master passwords are computed using cryptography's best practices (Argon2d) into a master key that is used to encrypt the database. The algorithm can be parameterized in order to take a certain amount of time to decrypt the database. The encryption and decryption process also include an integrity check in order to identify whether the database file was tampered with or corrupted.

KeePassXC advises regarding the robustness of user-supplied master password and/or key file and if requested, generates strong random passwords/key files.

SF4: Memory protection

KeePassXC renders its memory impossible to access for unprivileged users. This protection is not effective against attackers with administration privileges. This protection is persistent after the process *KeePassXC* has been terminated as well as when the database is locked.

SF5: Prompt of each password access from a browser extension

Each access to passwords or passkey triggers pop-ups from the *KeePassXC* process prompting for acceptance regarding the requested access.

SF6: KeeShare segregation

Imports using KeeShare are correctly segregated and do not permit to obtain any information from the receiving database.

SF7: ssh-agent interaction

KeePassXC securely interacts with ssh-agent/pageant to import private keys and submitting unlocking passphrases.

Threat coverage by security features	
SF1: Anti-screenshot/recording	**T1**: Take a screenshot or record the current screen to see passwords
SF2: Clipboard protection	**T4**: Clipboard snooping
SF3: Database decryption	**T5**: Offline database bruteforce
SF4: Memory protection	**T2**: Memory dump of the *KeePassXC* process (unprivileged)
SF5: Prompt of each password access from a browser extension	**T6**: Instrumenting the browser integration interface
SF6: KeeShare segregation	**T7**: Tampering or stealing entry information from an imported KeeShare database
SF7: ssh-agent interaction	**T3:** Stealing ssh private keys by exploiting the ssh-agent interaction

Disclaimer

- SF6 and SF7 are low priority security features and should be analyzed solely once the other features have been analyzed.