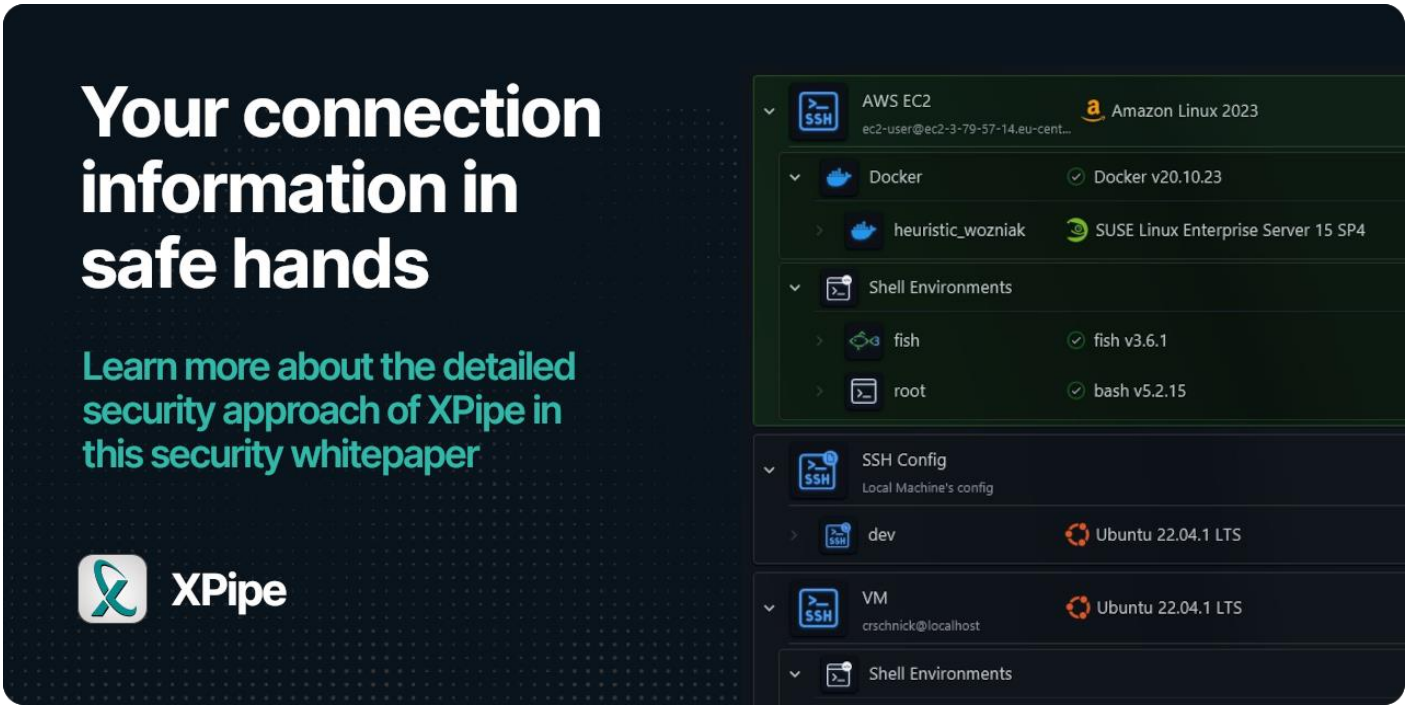

Due to its nature, XPipe has to handle a lot of sensitive data. This can range from passwords for all kinds of servers, to SSH keys, and more. Therefore, the security model of XPipe plays a crucial role. This document summarizes the approach of XPipe when it comes to the security and privacy of your data.

Contents

Security principles

XPipe is designed and built upon certain fundamentals that provide additional pillars for security and privacy compared to many other existing solutions.

Isolation through a rich client

XPipe is implemented as a rich client application, i.e., a desktop application that you install on your local machine. Any data managed by XPipe is stored and used on your local machine, plus your own systems you choose to connect to. There is no external component to XPipe which comes in contact with your connection data. As a result, no connection data managed with XPipe leaves your network and your controlled systems. There is no type of mandatory web service that the XPipe application sends any kind of connection data to. We have no possibility of monitoring or accessing any of your data that is managed in XPipe.

Advanced security through integrations

Another component is the compatibility and integration with external tools. As described later in this document, XPipe supports and integrates with many different password managers, git providers, key types, and more. It focuses on allowing you to use your existing and trusted solutions together with XPipe. For example, the XPipe application can be configured to fetch all types of secrets from external sources so that you can use your solution of your choice for secret storage. As a result, you can use XPipe effectively without having to store any secrets in XPipe itself.

Transparency through open-source

XPipe is an open-core project, which means that you can find the application core on GitHub licensed under the Apache License 2.0. Select parts are not open-source, such as some paid plan features, however, all relevant security implementations for the secret vault storage and others are available for everyone to check out and verify. This provides transparency of our development workflows and visibility into the software you are using, especially when compared to other closed-source solutions.

Local vaults

By default, XPipe will only store data on your local machine.

As a result, there already is a base level of security inherent with XPipe as any potential threat actors do not have access to the XPipe vault, unless, of course, your system is already compromised. Implementing robust security measures for your local systems is therefore already a big step in establishing the best first line of defense for your data in XPipe.

When it comes to the security of the XPipe data itself, we implement and maintain the latest established security standards to securely encrypt and store any sensitive information.

Types of data

When referring to any part of vault data, it can be generally categorized as either general data or sensitive data.

Sensitive data refers to things like:

- Passwords
- Private keys

The focus of the vault is securing sensitive data through secure encryption and storage.
Any other data stored by XPipe can be categorized as general data. This includes data like:

- Usernames
- Hostnames

With other general data, XPipe does not perform the same encryption by default. This is in line with other tools, e.g., with SSH where the username and hostname will show up in any shell histories and connection logs outside of XPipe. If preferred, you can choose to also encrypt any other general data as well by enabling this in the XPipe settings menu.

External secrets

For passwords, you have the option to fetch them from outside sources like password managers or enter them at connection time through a prompt window. In that case, XPipe doesn't have to store any secrets itself. This has the advantage of being able to rely on your trusted password manager for security.

You can configure a password manager setup in the XPipe settings menu. There are already a few templates included for common password managers. Furthermore, it is also possible to implement a fully custom secret retrieval operation workflow for any other kind of password manager.

Other secrets like private keys can also be managed and stored outside of XPipe. As long as the files are retrievable at runtime by XPipe, connections will work fine. You can therefore store and secure your sensitive key files in any way you like with XPipe.

XPipe also supports the use of smartcards and FIDO2 keys which offer a hardware-based authentication solution that does not have the possibility of exposing the underlying private key.

XPipe also supports storing passwords and key files in its own vault as seen next.

Managed secrets

In case you choose to store passwords and other secrets within XPipe, all sensitive information is encrypted with either:

- A dynamically generated vault key file (The data can then only be decrypted with that file present)
- A custom master passphrase that can be set by you in the settings menu, combined with the vault key file (This option is only as secure as the passphrase you choose)

The detailed encryption algorithm is an AES-128 GCM block-cipher with 12 IV bytes and 128 tag bits. The encryption keys are generated with a Password-based key-derivation algorithm (PBKDF) using SHA-256 HMAC. The full secret encryption implementation is open source and available on GitHub It is important that your supplied passphrase provides a high level of entropy in order to guarantee the best possible encryption.

Prompted secrets

Alternatively, you can also choose to prompt for some required secrets at runtime so you don't have to store a password for a connection anywhere. With XPipe, you have the option to input the password when you connect to that system and optionally cache it for the current XPipe session, so you don't have to reenter it until

XPipe is restarted. This has the advantage of the password not being stored on any file system as it only persists in the main memory while XPipe is running.

Remote vaults

To seamlessly synchronize your connection data with other devices and team members, XPipe allows you to synchronize your connections with a remote git repository. Any updates and connection data from the local vault are then regularly committed to that repository by XPipe. This repository can then be cloned on other systems or by other collaborators to have access to the same connections.

Repository access

This repository can be set up by yourself with any git service provider of your choice using the authentication scheme you prefer. XPipe supports all authentication schemes, both HTTP and SSH with advanced options like password prompts, key files, smartcards, and more. This guarantees the best possible control of your repository security as you can use your existing git solution to host the XPipe vault and your own tools to connect to it.

File encryption

When choosing to add a key file to the vault, you have the ability to also sync it with the remote git repository. This allows you to synchronize any required key files across all systems so you don't have to be concerned about missing key files on other systems. Any secret files committed to the repository are only committed in encrypted form similar to vault passwords. They will only be decrypted on the local system after pulling the remote data. This ensures that your remote vault does only contain encrypted sensitive data.

Security of remote vaults

Having connection data stored in a remote git service always bears the risk of this service being compromised and the git repository connection data being stolen. This risk also applies to local systems and its vaults being compromised, with the difference being that remote vaults are usually accessible from many users within a network.

The same security principles of local vaults also apply to remote vaults. The remote vault does not have to store any secrets if the vault is set up to fetch external secrets from providers like password managers. If the vault contains managed secrets, encrypting the vault with a strong custom passphrase should prevent any possible decryption of connection data.

Connections

When establishing connections, XPipe essentially delegates any form of connection and shell handling to your existing command-line tools. It does not come with any remote handling capabilities of its own. Therefore, the security of the established connection depends on your used command-line programs.

If, for example, your ssh command-line program or its connections are outdated and susceptible to MITM attacks or vulnerable in any other way, there is no way for XPipe to guarantee that data can be transferred securely. It is your responsibility to use external programs that XPipe interacts with in a secure environment and keep them up to date with security patches and more. Connections established with XPipe can only be as secure as your underlying command-line tools itself.

Secret transfer

When any kinds of passwords are required by a command-line program, these secrets have to be passed to it somehow.

In case a program accepts password input via stdin, this process is relatively straightforward. Then the secret information is just written into the stdin of the program and does not show up in any shell history.

In case the program supports askpass programs that can supply passwords, as is the case with tools like ssh and sudo, XPipe will act as an askpass program that will provide the necessary secrets to programs. These programs are then configured via environment variables to call the XPipe executable for any secret requests. Any returned secrets by XPipe do not show up in any shell history or file system in this case.

Outbound requests

The XPipe application itself only sends requests to two external services under certain conditions. These requests are initiated from the XPipe application, it does not listen or respond to requests that were initiated from any external services.

Licensing

The first one is the licensing API of our payment provider LemonSqueezy to which the XPipe application communicates to validate a license. This will send one request on startup to https://api.lemonsqueezy.com with only the license key for validation purposes. This request is only sent if a license is active. The community edition of XPipe does not perform this request.

We offer fully offline licenses that do not require an internet connection for our customers at no additional cost. If you are planning to deploy XPipe in an air-gapped environment with no outside internet connectivity, you can request offline licenses by sending an email to sales@xpipe.io. These offline licenses can also be used if the LemonSqueezy API servers are blocked or not whitelisted and can't be reached by the XPipe application in other cases.

Error reports

The second service the XPipe application communicates with is the error-tracking service Sentry This service is used to diagnose newly introduced problems, keep track of errors, and prevent exploits. Requests are sent when an unexpected error occurs in the XPipe application. So under normal operation, no requests are sent. Requests are sent to ingest.sentry.io. These requests do not contain any personal or sensitive data and are stripped of any contextual information. They are intended to show trends of new developing errors and are not intended to diagnose or troubleshoot individual error cases. The only information they contain is the OS name/version/architecture, XPipe version, an anonymized user ID that is used to group issues, and the stack trace without error message.

Other data handling

There are also a few other miscellaneous areas where your data is used such as log files and voluntary issue reports.

Logging

By default, XPipe creates log files. These files do not contain any sensitive data.

If you are troubleshooting an issue, launching XPipe in debug mode might sometimes be useful. In debug mode, log information is printed to the console instead and will contain a lot more and finer grained information, some of which might be sensitive. However, this information is not written to a file and will only be displayed temporarily in a console window.

In summary, log files will not contain any sensitive data.

Issue reports

Whenever an error occurs within XPipe, you can choose to automatically send an error report with feedback and attachments. This is a purely optional choice if you want to quickly provide additional error context information to help the development team speed up their bugfixing process.

If you do so, the error report will contain various types of general data such as hostnames, usernames, and more. This error report can contain personal or sensitive data depending on the error trace and context.

We are an european company, so we adhere to the GDPR. The privacy policy for the error reporting feature can be found at https://docs.xpipe.io/reporter-privacy-policy. The general privacy policy for the XPipe application itself can be found at https://docs.xpipe.io/privacy-policy.

Corporate security

The security of our technology is not the only focus. We also place high importance on organizational practices and processes throughout our development and business operations. Our development workflow incorporates best practices from OWASP to build secure software.

This includes, for example, static code analysis, which is a testing methodology that analyzes source code to find security vulnerabilities that make our applications susceptible to attacks. We also monitor any supply-chain vulnerabilities and monitor any third-party dependencies for potential vulnerabilities and updates.

By prioritizing security within our organization, we are able to develop and deliver robust and trustworthy applications.

Outlook

We hope that this document summarized the approach of XPipe when it comes to security and privacy. If any of your questions are left unanswered by this document, feel free to reach out to us at security@xpipe.io so your question can be answered individually and can also potentially be included in this document.

Reporting a security vulnerability

If you believe that you found a security vulnerability in XPipe, you can make use of the private security report feature of GitHub to ensure a confidential handling of this matter.