XPORTA Project Final Evaluation

Introduction

The goal of this project has been to improve the portability of user accounts between different XMPP services and software. In the previous phase of DAPSI, we successfully created and updated the relevant standards specifications needed to achieve this. We also began work on implementations of these new standards.

During the second phase, we have successfully improved the Prosody XEP-0227 implementation and we've built a friendly user interface in Snikket to access it and allow users to export and import their account data in the self-service account dashboard.

We can also report that other projects than ours have also now implemented the specifications too, notably ejabberd[1].

Design and approach

Prosody is the core component that powers the Snikket chat solution. In particular, it is the component where all a user's data is managed and stored. We developed an API plugin for Prosody that allows a user to request their data in XEP-0227 format using a HTTP REST API.

Upon clicking the 'Export' button, the Snikket web interface connects to this API to download a user's data in XEP-0227 format, and sends it to the user's browser as an XML file download.

Next we built the import functionality into the user onboarding process. As soon as a user registers an account successfully, we show them the option to import any existing data they may have (e.g. from a previous XMPP account). If the user chooses to upload a file, the web interface performs some minor validation and then forwards the uploaded XML data to Prosody via the API so the data can be imported into the new user's account.

Testing methods

Verifying correctness

When handling data, correctness is one of the most important aspects to the user. We want to ensure that if someone exports and then imports their data, it will not have been unexpectedly changed in any way. Such changes are to be considered bugs, and are easily introduced during data transformation if no tests are in place. Therefore we performed end-to-end testing of an export and import cycle.

[1] https://github.com/processone/ejabberd/issues/3676

The following 4 steps were taken for these tests:

1. Generate realistic test users (we used existing tooling for this).
2. Export the user data as XML in XEP-0227 format using our new export implementation.
3. Create a new (empty) account and import the data from the XML file using our new import implementation.
4. Check for differences between the original data structures and the data stored for the new user. structures after import to those of the original data. We found and fixed two bugs in our original implementation using this method.

Measuring performance

Although correctness is important, and import/export operations are not expected to happen very often during the lifecycle of a user account, we wanted to ensure that our implementation behaved within reasonable expectations. We determined to measure the average CPU, RAM and wall clock time required by data import and export operations.

For testing, we generated 1000 accounts with realistic test data and performed a bulk migration to XEP-0227 format, and then back again.

Operation	**Peak RAM**	**CPU time**	**Elapsed time**
Export 1000 users	18.69 MB	9.19s	9.77s
Import 1000 users	19.70 MB	14.02s	14.52s

With per-user import/export operations measured in fractions of a second, we were happy with these results.

As a final check, we inspected the resulting file size of the XEP-0227 XML export files. For our test accounts (which are designed to be representative of typical users, such as avatars, contacts and other data) the output file sizes ranged from 20KB to 40KB file size.

Interoperability with other implementations

Now that other implementations have begun to implement the updated XEP-0227 standard, we were finally able to test cross-implementation interoperability. We obtained an XML data export from another new implementation (ejabberd by Process One). We imported this data to Prosody using our import code, and were able to successfully verify the user data import succeeded as expected.

Security and privacy considerations

After reviewing the implementation and a risk assessment of the final project, we decided to selectively disable the self-service export of some user data. Specifically we disabled the ability to export a user's credentials (even though these are hashed, an attacker could subject them to an offline dictionary/brute-force attack) and we disabled export of a user's message history as this may contain sensitive information and metadata. These choices were made to limit the potential data exposure from a successful Cross-Site Request Forgery, or from a user account compromise.

With consideration to the requirements of the GDPR, a Snikket operator has the ability to generate a full export on behalf of a user, upon request. Indeed, our work makes it far easier for XMPP service operators to comply with GDPR requests from their users, as previously no suitable standard data format was available in which to send the data.

Final summary

Over the course of this project we have achieved the following:

- Developed and published new standard data formats and protocols for account portability in XMPP.
- Implemented support for these formats and protocols in Prosody, a widely deployed XMPP protocol server.
- Tested interoperability of our implementation with an independent implementation (ejabberd) of the standards we developed.
- Developed an independent web migrator prototype in Javascript that allows any XMPP user to convert their account data to and from an XEP-0227 XML file, regardless of server support for the new standard.
- Added support for exporting and importing user account data in Snikket, a modern open-source self-hostable chat solution based on Prosody and XMPP.

We are very grateful to NGI DAPSI for supporting this project and allowing us the funding and resources necessary to design and complete it.

Appendix: Links and resources

XMPP Account Portability (XPORTA)

Information and links related to this project.

- **Project homepage:** https://docs.modernxmpp.org/projects/portability/

Standards documents

Documents we have created/updated as part of this project.

- **XEP-0227: Portable Import/Export Format for XMPP-IM Servers**
  Published at: https://xmpp.org/extensions/xep-0227.html
  Pull request: https://github.com/xsf/xeps/pull/1064
- **XEP-0283: Moved**
  Published at: https://xmpp.org/extensions/xep-0283.html
  Pull request: https://github.com/xsf/xeps/pull/1071

Implementations

Developed partly or fully as part of this project.

Prosody plugins

Prosody is a module XMPP protocol server. Most of its functionality is implemented through different plugins, and we developed multiple as part of this project.

- **mod_storage_xep0227**
  Description: A Prosody storage driver that translates between Prosody's native data structures and the XEP-0227 XML format.
  Documentation: https://prosody.im/doc/modules/mod_storage_xep0227
  Source: https://hg.prosody.im/trunk/file/tip/plugins/mod_storage_xep0227.lua
- **mod_http_xep227**
  Description: Provides a HTTP API to import/export user account data in XEP-0227 format (uses mod_storage_xep0227 to perform the translation).
  Documentation: https://modules.prosody.im/mod_http_xep227
  Source code:
  https://hg.prosody.im/prosody-modules/file/541b2cf68e93/mod_http_xep227/mod_http_xep227.lua
- **mod_auto_moved**
  Description: Automatic handling of 'moved' notifications received from contacts, as described in XEP-0283.
  Documentation: https://modules.prosody.im/mod_auto_moved
  Source code:

https://hg.prosody.im/prosody-modules/file/f95a1e197a07/mod_auto_moved/mod_auto_moved.lua

Snikket

Snikket is an open-source project that combines Prosody and other components to create a modern self-hostable chat solution based on open standards (XMPP). As part of this project we have incorporated new features.

- Added support for account import/export in Snikket's Prosody: https://github.com/snikket-im/snikket-server/pull/113
- Added support for account import/export in Snikket's web interface: https://github.com/snikket-im/snikket-web-portal/pull/110

Web migrator (prototype)

This project was not in the original project plan submitted to DAPSI, but developed as an additional prototype to validate the method and provide an (emergency!) way for users to export their data from an independent German XMPP service that shut down in July 2021.

- **Migrator homepage:** https://migrate.modernxmpp.org/
- **Source code:** https://github.com/snikket-im/xmpp-account-exporter