

Scrum iteration I

Rolf Jagerman, Laurens Versluis and Martijn de Vos

May 13, 2014

1 Introduction

Hier iets in het algemeen over deze eerste scrum iteratie en een uitleg wat er besproken wordt in dit bestand

2 Our work

In deze sectie komt wat we deze sprint gedaan hebben

2.1 Porting the anonymous tunnels to Android

In deze subsectie iets over de anontunnels die we hebben geport naar android, wat over Python for Android en de packages. Misschien de packages in subsubsections bespreken?

2.1.1 Anonymous tunnels

This package is the core of our application and contains the code needed for the anonymous tunnels. This package actually contains another package: the support for the Socks5 proxies. The files for this package come from pull request 525 on the Tribler Github.

We've made some minor changes to this code: the Main.py file has been removed and the class definition of AnonTunnel has been moved to it's own file (atunnel.py). We import this file in our main.py so we can use the AnonTunnel class. We also adjusted the master key in community.py so we have our own community to test communication between devices.

2.1.2 Tribler core

The anonymous tunnels are using some parts of the core of Tribler. These files have been bundled in the tribler_core_minimal package. We've looked closely to the various imports the code for the anonymous tunnels is using and when a script imports a script from the Tribler core, it is added to the tribler_core_minimal package. For example, this package contains the RawServer and the SocketHandler classes. It also contains the code needed for the cryptography such as support for elliptic curve cryptography and ELGamal.

2.1.3 Dispersy

Since the code of the anonymous tunnels are using Dispersy for node discovery and data synchronization, we've created a Python package with all the code that's needed for Dispersy. This package does not depend on other custom packages we made and can be used standalone.

The files we have bundled are from pull request 525 on the Tribler Github. We didn't use the files from the official Dispersy Github because this build was missing some classes we needed (for example, the decorator.py). Besides

that, some changes have been made to the Dispersy core to add support for the anonymous tunnels.

2.2 Creating a GUI with Kivy

The first version of the anonymous tunnels used the standard output for printing information about what's going on. This required the phone to be connected to a computer so we can examine the log with the adb logcat tool. That's why we decided to create a graphical user interface for our application. The purpose of this application is to provide a button to start the tunneling and a log to see on the screen what's happening.

Creating a GUI was a small stap for us: we already included the Kivy package in our Python for Android distribution. Since Kivy is a GUI framework for desktop (but has been ported to Python for Android), we first tried to create a desktop interface. Creating interfaces in Kivy is quite straightforward and is related to creating user interfaces in Android: you specify your layout elements in Kivy files which have the kv extension. In the main Python file, you load this interface file and you can access properties of the UI elements.

2.3 Attempt (?) to compile Libtorrent for Android

Hier iets over onze uitdaging om Libtorrent werkend te krijgen in Python for Android

3 Conclusion

Hier komen onze conclusies over deze sprint (wat ging er goed/fout, wat willen we anders doen, wat gaan we volgende sprint doen etc.)