Group 22: Initial Report

Rohit Panda

rohit.panda@tum.de

Benedikt Brandner

ga49xel@mytum.de

Abstract

This is the initial report of group 22, working on the onion module. This report details choices made with regard to methods, platforms, programming languages and organization that we plan to employ over the course of the project.

1. Preamble

1.1. Team

Teamname: voidphone-onion-qt
Module: onion

- Bendikt Brandner, ga49xel@mytum.de
- Rohit Panda, rohit.panda@tum.de

2. Programming Language, OS, Building & Libraries

We plan to use C++ with the Qt framework. We chose C++ for its good methods in manipulating low level payloads, on top of that Qt provides useful abstractions for common networking cases (e.g. TCP/UDP sockets) and alleviates many more common pains of a plain C++ application. For example it also provides readers for the ini format used by the configuration files, string argument parsing, logging, etc.

Qt is completely crossplatform, thus we don't require any specific OS. Nevertheless we use Windows at the moment. This also matches with the

purpose of the application which allows running the client on a wide variety of platforms.

Qt brings it's own build system (qmake) that compiles to makefiles. All of that is handled by the IDE included.

In terms of libraries, Qt provides most of the needed things, for cryptography a number of libraries could be considered, most prominently QCA [1], crypto++ [2] or directly OpenSSL (which would probably be rather painful).

3. Quality Control

Qt provides a testing module to perform unit tests, e.g. for testing connectivity, key exchange, encryption, ... For more complex test scenarios scripted setups could be evaluated, either starting multiple instances on localhost or using a more elaborate system like mininet.

In terms of debugging, Qts parenting system should save us from most of the memory leaks. For static analysis tools like cppcheck could be considered.

4. License

We will choose the MIT license [3] because it allows usage, modification and distribution of source code in each and every way, and we don't see much added value by using a more restrictive license.

5. Previous Knowledge

5.1. Benedikt Brandner

Roughly 5 years software engineering, lots of experience with Qt, including low level communication, more general networking, protocol design. Network security lecture, ACN master course.

5.2. Rohit Panda

Roughly 5 years of Software Engineering experience. Worked on developing services on SS7 protocol stack.

---

[1] `https://userbase.kde.org/QCA`

[2] `https://www.cryptopp.com/`

[3] `https://opensource.org/licenses/MIT`

6. Project Management

We will be organizing our project work in a dynamic manner, using the integrated issue tracker of gitlab for long term tasks and a messenger like WhatsApp for direct communication. We haven't done any general division of the work by now, since we don't think it makes much sense at this stage of the project. We will try to figure out a reasonable labor division along the way.

7. Issues

-