The Free Secure Network Systems Group: Secure Peer-to-Peer Networking and Beyond

Christian Grothoff
Free Secure Network Systems Group
Department of Computer Science
Technische Universität München
Email: grothoff@net.in.tum.de

***Abstract*—This paper introduces the current research and future plans of the Free Secure Network Systems Group at the Technische Universität München. In particular, we provide some insight into the development process and architecture of the GNUnet P2P framework and the challenges we are currently working on.**

I. Introduction

The Free Secure Network Systems Group (FSNSG) was established in Fall 2009 by a grant from the Deutsche Forschungsgesellschaft (DFG) in the Emmy-Noether Program. It currently consists of four full-time researchers (including three PhD students) and five Master's students working on research projects or theses. The group is largely working in the area of secure peer-to-peer (P2P) networks, but is also looking at networking issues in general, including work on scaleable graph algorithms and distributed programming [1].

In terms of security, our focus is on secure network protocol design and implementation. Secure software engineering is a key component when building secure systems. Our software engineering practice is tool-centric; as part of our implementation work, we use, extend and sometimes develop software engineering tools, in particular static analysis tools, portability and regression testing tools. Since availability and performance often are closely related, we are currently developing a new tool for cross-platform performance regression analysis.

II. Current Research

The main focus of our group is the development of GNUnet[1], GNU's framework for secure P2P networking. One of the characteristics of the GNUnet system is that it uses a multi-process architecture for fault isolation; failures in individual components are isolated in their respective address spaces and rarely affect other parts of the system. While GNUnet is currently mostly written in C, the multi-process architecture also enables the development of extensions in other languages.

GNUnet uses a layered architecture (Figure 1). At the bottom layer, transport plugins enable P2P communication [2]. GNUnet can currently communicate using UDP, TCP, HTTP or HTTPS. Support for IP-less direct communication using WLAN is under development. Our next goal here is to formalize an effective strategy for efficiently selecting the best communication method while satisfying resource and security constraints. The transport layer also includes support for NAT traversal [3]. Above the transports, the GNUnet core provides link-encrypted peer-to-peer communication, bandwidth allocation, peer discovery and other general-purpose functions necessary for any secure P2P network. GNUnet applications (for example, anonymous file-sharing [4]) and services (for example, our DHT) then use the core to communicate with other peers.

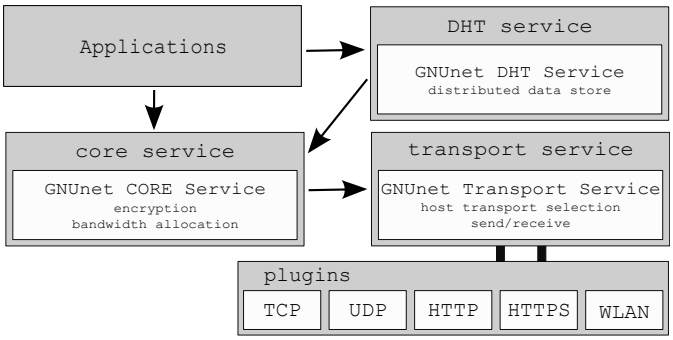

Fig. 1. GNUnet Architecture.

Our main research focus at this point is on demonstrating the security and scalability of a new randomized DHT design and its use for various applications. Our DHT has the special property that it can work on top of a restricted-route underlay; in other words, it does not make the assumption that any peer can directly communicate with any other peer. Using this DHT, we plan to develop a mesh routing abstraction for the construction of redundant tunnels between clients and network services integrated into the P2P overlay. As a first service, we plan to offer a virtual network interface (IP-VPN) that uses the P2P overlay to provide IP connectivity (for both IPv4 and IPv6).

For the evaluation of our designs, we emulate large P2P networks using the testing support library available in GNUnet. Using the GNUnet framework, we can easily setup thousands of peers on a single desktop system (or tens of thousands using a cluster), connect them using various topologies and run experiments. A transport plugin using UNIX-domain sockets can be used to avoid problems with the 64k port limitation of UDP or TCP.

[1] https://gnunet.org/

III. Teaching

Our teaching goals are to enable students to design and implement secure systems as well as to analyze existing designs for flaws. The GNUnet framework provides a starting point for the design and implementation of secure P2P network protocols. For example, we have in the past asked students in the "Peer-to-Peer Systems and Security" course to implement a range of proposed DHT designs in the framework. While some groups succeeded, it is clear that making the framework accessible from multiple languages would be a major improvement: many of our students are more proficient in Java or Python than in C. This need reinforces our belief that the multi-process architecture is the right design choice for a P2P framework. Currently, we are asking students to design and implement a distributed web search engine using the framework; however, it is too early to draw any conclusions from this. However, students at other universities have already created new applications using the GNUnet framework.[2]

On the analysis side, we regularly supervise students who, as part of their master's thesis, analyze the design of an existing free software P2P network and then devise, implement and evaluate an attack based on vulnerabilities caused by particular design choices. The goal is not to find simple bugs in the existing implementation but to find and exploit weaknesses that the developers build-in by design. For example, our attack on Tor [5] is based on source-routing and low-latency routing, two fundamental design choices for Tor. For Freenet, our attack [6] exploits a key step in their routing algorithm. Our recent work on I2P [7] builds on their use of unidirectional paths and performance-based peer selection. The resulting thesis is typically publishable work and the students are keenly aware of the security implications of certain design choices and have learned to understand complex software systems to a sufficient degree to find design flaws by studying documentation and source code.

Finally, we of course encourage all of our students to famlilarize themselves with the various software engineering tools that we have deployed. This generally improves their ability to write correct code quickly. Furthermore, we believe that knowing available tools is key for secure software engineering.

IV. Future Plans

We currently see an urgent need for an Internet architecture that is resilient to malicious participants and not under the control of cooperations or governments. With the widespread use of wireless networking equipment, a secure, scalable and most of all easy-to-use P2P network with support for DNS and an IP-VPN could solve the problem of three-strike-Internet-kill-switches and further the agenda of free software: user's freedom. Naturally, a large number of technical hurdles need to be overcome: secure routing, scaleable creation of virtual tunnels with TCP-like semantics, secure naming for DNS and design and integration of secure variants of important Internet applications into the P2P network. Finally, we hope to face the challenge of making the resulting system easy to use while maintaining security for ordinary users.

In the near term, we also plan to further extend on our tool suite for secure software engineering. In particular, we want to customize off-the-shelf tools to better support the idioms of a particular large software system. Furthermore, we are working a tool that can be deployed at end-user systems to help developers automate key steps in the diagnosis of problems, especially those that they cannot reproduce on their own systems. This could be particularly useful if the end-user experiences system-specific problems, such as an external attack, and the developer requires more than simple logs or heap images for the diagnosis.

V. Conclusion

Our group offers expertise in the areas of analysis, design and implementation of secure P2P networks. We will be happy to support other groups that want to build systems using the GNUnet P2P framework for teaching or research. Feedback on the various libraries, software-engineering and language tools maintained by our group is also always welcome. We would be interested in models or measurement data to help make our security and performance analyses more realistic.

Acknowledgements

This work was funded by the Deutsche Forschungsgemeinschaft (DFG) under ENP GR 3688/1-1.

References

[1] N. Evans, C. GauthierDickey, C. Grothoff, K. Grothoff, J. Keene, and M. J. Rutherford, "Simplifying parallel and distributed simulation with the DUP system," in *Proceedings 43rd Annual Simulation Symposium (ANSS-43 2010)*. Orlando, FL, USA: Society for Modeling & Simulation International, April 2010, pp. 208–215. [Online]. Available: http://dupsystem.org/anss2010

[2] R. A. Ferreira, C. Grothoff, and P. Ruth, "A Transport Layer Abstraction for Peer-to-Peer Networks," in *Proceedings of the 3rd International Symposium on Cluster Computing and the Grid (GRID 2003)*. IEEE Computer Society, 2003, pp. 398–403. [Online]. Available: https://gnunet.org/transports

[3] A. Müller, N. Evans, C. Grothoff, and S. Kamkar, "Autonomous nat traversal," in *10th IEEE International Conference on Peer-to-Peer Computing (IEEE P2P 2010)*. IEEE, 2010, pp. 61–64. [Online]. Available: https://gnunet.org/pwnat

[4] K. Bennett and C. Grothoff, "gap - Practical Anonymous Networking," in *Designing Privacy Enhancing Technologies*. Springer-Verlag, 2003, pp. 141–160. [Online]. Available: http://gnunet.org/gap

[5] N. S. Evans, R. Dingledine, and C. Grothoff, "A practical congestion attack on tor using long paths," in *18th USENIX Security Symposium*. USENIX, 2009, pp. 33–50. [Online]. Available: https://gnunet.org/torattack

[6] N. S. Evans, C. GauthierDickey, and C. Grothoff, "Routing in the dark: Pitch black," in *23rd Annual Computer Security Applications Conference (ACSAC 2007)*. IEEE Computer Society, December 2007, pp. 305–314. [Online]. Available: https://gnunet.org/pitchblack

[7] M. Herrmann and C. Grothoff, "Privacy-implications of performance-based peer selection by onion-routers: A real-world case study using i2p," in *Privacy Enhancing Technologies Symposium (PETS 2011)*, 2011. [Online]. Available: https://gnunet.org/i2p_2011_pet

[2] http://sourceforge.net/projects/s-n-a-g/