Decentralized Bootstrap for Social Overlay Networks

Rodolphe Marques, André Zúquete

Decentralized Bootstrap for Social Overlay Networks

Rodolphe Marques[1] and André Zúquete[2]

[1] IT, University of Aveiro
[2] DETI / IEETA, University of Aveiro
{rodolphe.marques,andre.zuquete}@ua.pt

Abstract. In this paper we show how we can use social networks to bootstrap a social overlay network. This overlay network is different from others, in the sense that it enables participants to share services on a personal basis, unlike other overlay networks that provide a single service for all peers. Since the overlay network is not supposed to have central servers for managing a single service, its bootstrap and the direct communication among pairs of participants is challenging. However, the actual social networks, such as Twitter, Facebook and Google+ already provide an API that enables participants to exchange direct messages, which will be the basis of our bootstrap mechanism.

Keywords: Privacy, P2P interactions, social networks

1 Introduction

Privacy is hard to achieve in centralized architectures [1], since one needs to trust in service providers to mediate all the information that we disclose while being out of the clients' control. On the other hand, more private communication channel in the Internet could be achieved if one could interact directly to the intended persons or entity, without central services.

The goal of our work is to provide human-to-human (H2H) private services using the Internet, as stated in [5]. We distinguish H2H from peer-to-peer (P2P) because, on the latter, peers are just participants (on particular protocols) that are alike and don't cooperate strictly on a one-to-one basis, while we want to provide means for clear, personal interactions, where persons can act differently.

H2H private services allow pairs of clearly identified persons to provide services to one another without service-oriented mediators. The set of services provided by each person involved in a H2H interaction can be different, there is no need to have reciprocity. Such service provisioning takes place over a Virtual Private Link (VPL, see Fig. 1). We don't see a VPL as a Virtual Private Network, since the former will enable only a controlled access to a set of (well-defined) services, while the latter usually provides an access to a network, where many (ill-defined) services may exist.

The VPLs used by all persons exploring our system will form an overlay network (of services). This overlay network is not oriented to a single service,

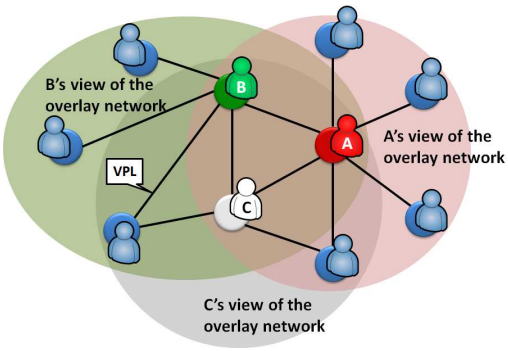

Fig. 1. Overview of the overlay network, formed by many different human-centric, H2H interactions on top of VPLs. Private interaction between A and B can start either because A invited B to join his (view of the) overlay network or vice-versa. A, B and C can provide services among themselves in a private way, without knowing the full extent of the entire overlay network (e.g., C may not know that A interacts with B).

such as routing (e.g. TOR [3]) or content sharing (e.g. BitTorrent [2]). There is no global definition of the services provided in the overlay network by the participant; they are free to create their own services and provide them privately to others). Furthermore, there is no global notion of who is involved in the overlay network. Each participant will have his own view of the overlay network, which will be formed (to him) by the persons with whom he has a VPL established. That's why we say that we have social overlay networks (one for each person).

1.1 Problem

Bootstrapping overlay networks has been a longstanding problem [7] that is usually solved by one of two ways: using the binding information of a least one node in the network (e.g. for DHTs); or using a centralized directory service (e.g. TOR directory servers [3]). In the first case the binding information can change frequently and needs to be obtained through an out-of-band mode. The second case requires dedicated network infrastructure to aid the bootstrap. Moreover it leaks information about the entire network since the directory server contains information about all the nodes in the network, which besides the privacy implications that it may bring, it provides a single point of failure that can be open to attacks or that can be easily blocked.

Yet another problem with current overlay network designs is that users joining the network have little or no control on the network. Users have no control regarding the nodes they connect to or which nodes connect to them. And even if they had the control to choose that, there is not enough information about the other nodes in the network except for their binding information. In short, overlay networks are cooperative and service-oriented by nature, but not social. This is not what we are looking for, since we want persons to build their own overlay network by explicitly exploring H2H interactions with known persons.

1.2 Contribution

Since we want to bootstrap in a distributed way an overlay network formed by an arbitrary number of H2H interactions, its seems natural in our days to explore social networks for that purpose. This could enable persons to create and manage their personal view of the H2H overlay network (i.e. create their own VPLs) by reusing their previous work in the management of their social graph in Web-based social networks. In other words, we can use social networks to extract existing relationships with persons with whom one may be interested in setting up a VPL.

2 Decentralized Bootstrap for our Social Overlay Network

Nowadays social networking platforms (Twitter, Facebook, Google+, etc.) have an API that enables applications to exchange private messages with friends within the same social network. This facility enables us to use social networks to bootstrap our overlay network. In particular, we can use social network to send our personal communication endpoint to friends, this way using the social network as a rendez-vous point, or a mailbox, for exchanging this information.

Personal communication endpoints are UDP/IP or TCP/IP transport endpoints that can be used to contact a person in our overlay network. Such endpoint needs to use a public IP address, otherwise it may not be reachable from outside its own network. However, the current Internet architecture makes this difficult, since Internet clients are frequently behind NAT (Network Address Translation) routers that raise many issues regarding the direct addressing of hosts behind them [6].

Currently we foresee three strategies for enabling client hosts to get their public transport endpoint: (i) management of the egress NATs to set up a public endpoint as a forwarding transport port; (i) exploitation of transport addresses of TURN servers [4]; and (iii) exploitation of a TURN server as a service provided indiviadually by participants in our own overlay network.

The first possibility is the preferable one, since it allows the most direct communication between participants. However, in many cases it may not be possible to explore, because existing NAT equipments may not allow hosts behind them to manage port forwarding policies.

The second possibility may overcome this limitation but requires the exploitation of TURN (Traversal Using Relays around NAT) servers. These servers simply relay traffic over allocated, public transport endpoints. A host behind a NAT router can allocate a single TURN public endpoint to receive incoming traffic from several hosts. The identification of the contacting peers is provided in TURN messages that are used to encapsulate the traffic between the TURN server and the TURN endpoint allocator.

The third possibility is in fact a combination of the previous ones. A hosts capable of having a public transport endpoints can run a TURN server and provide this service to friends that may use it to set up their public endpoints.

In any case, for the handshake protocol through a social network direct messaging channel all we need is to send, along with some distinctive keyword, the transport endpoint that should be used to contact the message sender, regardless of being a public address of his own or the public address of a TURN server.

This bootstrap protocol is completely decentralized, since each participant manages the bootstrap of his own VPLs. Furthermore, even for each VPL, which connects only a pair of participants, each of them may take the initiative to propose to the other its creation, just by publishing on a social network his public endpoint.

3 Conclusions and Future Work

In this paper we have presented a strategy for bootstraping an overlay social network of services. Unlike other overlay networks, this one does not target a single service, but rather a H2H personal exchange of services. Each participant in the overlay network has its own view of it, formed by a set of VPLs established with friends. Thus, for bootstraping such an overlay network we may use social relationships established through social networks to make a first handshake towards the creation of VPLs. This is currently facilitated by the fact that the most popular social networks have APIs for sending and receiving arbitrary information, and through which we can send the public communication endpoint that a person makes available to a friends for establishing VPLs.

The next step that needs to be tackled is related with the authentication of the participants in the overlay network. This authentication is fundamental for preventing a person from being fooled by the social network (with fake messages) or by someone else that gets to know his public endpoint without being explicitly contacted. This authentication is also fundamental to perform an authenticated key distribution protocol for deriving session keys for protecting VPLs' traffic.

References

1. Boyd, D.: Facebook's Privacy Trainwreck. Convergence: The Int. Journal of Research into New Media Technologies 14(1), 13–20 (Feb 2008)
2. Cohen, B.: Incentives build robustness in BitTorrent. In: Proc. of the First Workshop on the Economics of Peer-to-Peer Systems. Berkeley, CA, USA (Jun 2003)
3. Dingledine, R., Mathewson, N., Syverson, P.: Tor: The Second-Generation Onion Router. In: Proc. of the 13th USENIX Security Symp. (Aug 2004)
4. Mahy, R., Matthews, P., Rosenberg, J.: Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN). RFC 5766 (Proposed Standard) (Apr 2010)
5. Marques, R., Zúquete, A.: User-centric, private networks of services. In: Int. Conf. on Smart Communications in Network Technologies (SaCoNeT). pp. 1–5 (Jun 2013)
6. Srisuresh, P., Ford, B., Kegel, D.: State of Peer-to-Peer (P2P) Communication across Network Address Translators (NATs). RFC 5128 (Informational) (Mar 2008)
7. Wolinsky, D., Juste, P., Boykin, P., Figueiredo, R.: Addressing the P2P Bootstrap Problem for Small Overlay Networks. In: Peer-to-Peer Computing (P2P), 2010 IEEE Tenth International Conference on. pp. 1–10 (Aug 2010)