Security Analysis of Leap-of-Faith Protocols

Viet Pham[1] and Tuomas Aura[2]

[1] Royal Holloway, University of London, Egham, Surrey, TW20 0EX, UK
viet.pham.2010@rhul.ac.uk
[2] Aalto University, P.O.Box 15400, FI-00076 Aalto, Finland
tuomas.aura@aalto.fi

Abstract. Over the Internet, cryptographically strong authentication is normally achieved with support of PKIs or pre-configured databases of bindings from identifiers to credentials (e.g., DNS to public keys). These are, however, expensive and not scalable solutions. Alternatively, Leap-of-Faith (LoF) provides authentication without additional infrastructure. It allows one endpoint to learn its peer's identifier-to-credential binding during first time communication, then stores that binding for future authentication. One successful application of LoF is SSH server authentication, encouraging its introduction to other protocols.

In this paper we analyze the security of LoF protocols. Various aspects are discussed to show that several proposed LoF protocols have weaker security than SSH, and that their security also depends on design and implementation details. Several protocols were analyzed, including SSH, TLS, BTNS, and HIP, revealing attacks such as impersonation, man-in-the-middle attacks, and credentials flooding. Consequently, additional mechanisms and best practices are proposed to strengthen LoF applications.

Keywords: leap-of-faith, authentication, key management, SSH, TLS, BTNS IPsec, HIP, decentralized system, infrastructureless.

1 Introduction

Due to physical separation, Internet communication suffers from identity-spoofing attacks, such as impersonation and man-in-the-middle (MitM). When two parties communicate, they need a way to name each other. Each party is represented by a *communication identifier*. For example, in `telnet` remote login, the communicating parties are the server and the client user (not the client computer). The server is identified by its DNS name or IP address, whereas the client user is identified by a username. However, since `telnet` transmits username and password in plaintext across the Internet, these identifiers could be easily spoofed.

To prevent identity-spoofing attacks, there are authentication methods based on *cryptographic credentials*. Each credential is owned by one entity and can be used to verify its identity. For instance, in public-key authentication, public keys are used as credentials. To facilitate authentication, the identifier of an entity must be mapped to that entity's credential. When someone claims to have an

M. Rajarajan et al. (Eds.): SecureComm 2011, LNICST 96, pp. 337–355, 2012.

identifier, the authenticator can use the corresponding credential to verify the ownership. The main problem with authentication is to maintain such kind of identifier-to-credetial mappings, or *security bindings* in our terminology.

Authentication should be strong in the sense that all bindings accepted by the authenticator are correct. In distributed systems, strong authentication is usually supported by a trusted third party (TTP) or a public-key infrastructure (PKI). For example, in TLS, each binding is represented by a certificate. Each certificate must be signed by a certificate authority (CA) within the PKI hierarchy, and its correctness could be securely verified given the public key of the root CA. Similarly, symmetric-key systems like Kerberos provides strong authentication [1] using a key distribution center (KDC) as a TTP. The strong authentication with a PKI or TTP, however, does not come without costs:

- Registration effort: correct bindings must be registered with the CA or TTP, which requires administrative effort [2]. For example, the TTP administrator must carefully perform background check on the binding owner, or otherwise attacks are possible, e.g., [3].
- Cost: as a business process, each registration incurs a cost to the registering party to have its binding certified. Most individuals and many businesses are unwilling to pay such fees especially for local or temporary IT systems.
- Scalability: with the current size of the Internet, no TTP is capable of maintaining a database of bindings for every network entity. This is especially true in peer-to-peer communication in which all endpoints are equal the their number can be very large.

These limitations have led to a search for alternative forms of authentication. One possibility is *recommendation systems* with PGP Web of Trust [4] as an early example. In these systems, the reliance on the trustworthiness of CAs is replaced by trust between people based on experiences and recommendations. As certification of bindings is decentralized to a community rather than a single PKI hierarchy, the registration and management costs could be lower. However, modelling of trust in these systems requires complicated mathematical techniques (e.g. [5]) and their applicability to non-human entities like computers is still unclear. Also, both the authenticator and the peer being authenticated must be in the same community, making it less scalable globally.

Another idea is the use of *self-certifying identifiers*. An identifier is generated by evaluating some collision-resistant function f on a credential, so that the mappings between identifiers and credentials are one-to-one, hence avoiding impersonation. As identifiers can be generated locally, no third party is needed and there is no administrative overhead. This idea works well when identifiers can include arbitrary bit strings, for instance, in the SEND protocol [6] with *cryptographically generated address* [7,8]. In contrast, user-interactive applications require human-readable identifiers, which cannot be arbitrary hash values. *Identity-based cryptography* [9] attempts to solve this, but still requires a TTP, i.e., a private key generator and thus suffers from similar problems as PKI.

In this paper, we focus on the *Leap-of-Faith (LoF)* method as an alternative to strong authentication. LoF is familiar to most university users from SSH server