FAST and Pre-Authentication Framework

Sam Hartman

Painless Security, LLC

IETF 72

July 29, 2008

Moving Forward FAST

- ➜ Changes since version 7
- ➜ Protocol walk-through results
- ➜ Open issues
- ➜ Case study: channel binding and *encrypted challenge*

Changes since Version 7

ENCRYPTED CHALLENGE

Version 7 used *authenticated timestamp*. Version 8 introduces *Encrypted Challenge* which should be simpler and avoids time synchronization on the client.

- ➜ Based on *Encrypted Timestamp* from RFC 4120; the timestamp is only used to limit the replay window. Facilities are available if the client time is out of sync.
- ➜ The resulting ticket is sent in a new reply key rather than the long-term key.
- ➜ **Needs security review:** some problems already found during the walk through

Authentication Sets

Several open issues with *authentication sets* have been cleaned up.

- ➜ The heart-beet mechanism is removed; KDCs double up messages as appropriate.
- ➜ Clients indicate which set they select. Per mailing list discussion, clients include the full set they select not an index.

Other Changes

- *Armor keys* are required to be fresh in order to prevent cross-conversation cut&paste.
- The previous spec allowed too much flexibility in when parties could ignore messages that they might not understand. Once a party has used an extension, they are presumed to understand that extension now.
- A *well-known name* is used when clients hide their identity in the outer request. Currently the anonymous name.

Protocol Walk-through Results

Protocol Walk-Through

Monday, a group got together to analyze the FAST protocol. We hoped to come up with recommended solutions for a number of open issues. Instead, many new open issues were discovered. The meeting was quite productive; Larry and I would like to thank the participants.

What FAST is Not

It's easy to think of FAST as a full tunnel or as a complete replacement for messages. However:

- ➜ FAST does not wrap errors; it does provided a protected container within errors.
- ➜ FAST does not wrap the *AS-REP;* it does allow the reply key to be replaced and provide checksumming.

Is this the right trade-off? Not wrapping errors may be problematic.

Clarity Problems

- Where does the *cookie* go, what is covered by the *finish* checksum?
- How do *armor tickets* interact with validating or proxying tickets where you are presenting a service ticket not a TGT?
- FAST should be advertised in the non-FAST PREAUTH_REQUIRED error.

Deployment and Operational Concerns

- ➜ Like all pre-authentication mechanisms FAST needs to be available on all KDCs in a realm before it is offered by any.
- ➜ FAST involves a implementation-defined *state cookie* that must be passed back and forth with requests. You cannot mix and match KDC implementations from different vendors if we adopt FAST.
- ➜ We need to work through how unprivileged processes can use FAST to get tickets without gaining the ability to authenticate as the host.

Security and Extensibility

- ➜ State cookies need to include the initial PREAUTH_REQUIRED error so that the negotiation of mechanisms is protected. That means even one-round-trip mechanisms need the cookie.
- ➜ *Encrypted Challenge* is vulnerable to a serious man-in-the-middle attack if the KDC's identity is not known. Fixes were proposed at multiple levels.
- ➜ How important is replay detection for Encrypted Challenge? Doing that cross-KDC is hard.
- ➜ We need to use strengthen-reply-key more than replace-reply-key
- ➜ Hosts **MUST NOT** print their own tickets for extensibility reasons.

Open Issues

Summary of Walkthrough Issues

- Should FAST protect more?
- Which approach do we take for fixing Encrypted Challenge ? What are the more general/abstract things we take away in terms of security requirements and mechanism design guidelines?
- How do we handle service tickets presented to the KDC?

Other Open Issues

- ➜ Several of the *FAST options* have confusing names; Ken proposes fixing them.
- ➜ Should KDCs allow any TGT to be used as an armor ticket?
- ➜ When can a reply key be replaced? Limiting options would limit testing complexity.
- ➜ What errors should be used for decryption failure in Encrypted Challenge?

Case Study: Encrypted Challenge and Channel Binding