Extensibility in the Kerberos Protocol

Sam Hartman
Mekinok, Inc.
IETF 52

Table of Contents

Table of Contents

I. Why Extensibility?

Arguments for Extensibility

- Vendors need extensibility to get technologies to market.
- The IETF needs extensibility to change the protocol while maintaining backward-compatibility.

Protocol Requirements from Vendors

- Vendors choose standards based on what is available when they start a project.
- Market forces may make a new technology critical at any time.
- Waiting for standard changes is not an option.

IETF Concerns About Vendor Extensibility

We want vendors to be able to use our standards, but *in such a way that interoperability between vendors is maintained.*

Evolution of the Protocol within the IETF

Developing a new, incompatible protocol to make changes has high cost. Thus, the IETF needs to be able to extend the protocol in future.

Mechanisms to Support Protocol Evolution

- Mechanism for making phased upgrades to core protocol messages
- Extensibility model that minimizes likelihood of changes today precluding other important changes in the future

II. Kerberos Extensibility Track Record

Extensibility Proposals Presented at IETF

- Kerberos error checksums (IETF 38)
- Ticket extensions (IETF 39)
- Authorization Data clarifications (IETF 39)
- Make some fields optional
- Checksum of AS-REP
- Crypto system selection

Where these Extensions Stand Today

With the exception of crypto system selection, *none of these extensions work in an interoperable manner*. Many have been removed from the draft pending a better solution; others would present significant problems if implemented.

Common Problems with Extensions

- Many proposed extensions assume fields can be added to ASN.1 sequences. Unfortunately, doing so breaks backward compatibility.
- Significant effort was spent trying to avoid adding capability negotiation to Kerberos. As such, clients cannot tell whether extensions they want to use are supported.
- Extensions were considered one-at-a-time; we didn't take advantage of common elements.

Why Avoid Negotiation

- Negotiation adds complexity.
- Negotiation often adds round trips
- Negotiation is harder in Kerberos than other protocols because Kerberos involves three parties. The KDC must know the capabilities of the service.

III. A General Solution

Why can we do better with a general solution?

- Storing one bit of state on the KDC to facilitate general negotiation is reasonable, even if storing a bit for each option is not.
- We can spend more time reviewing the ASN.1 for a general solution than for any specific option.
- We can abstract out common elements of proposed extensions. For example, we can solve all instances of authenticated cleartext rather than specific cases.

Goals of Our General Solution

- Provide the IETF with a mechanism for future protocol evolution.
- Provide vendors with hooks to extend the protocol in interoperable ways.
- Provide a means to authenticate cleartext carried in Kerberos messages.

ASN.1 Extensibility Allows IETf Protocol

We add ASN.1 extension markers to most Kerberos messages.

- New fields can be added to the end of messages
- Not useful for vendor extensions because we must coordinate tag assignment; only the IETF can take advantage of the ASN.1 extensibility markers.

Meeting Vendor Needs with Typed Holes

- Typed holes (sub-messages that contain an octet-string along with an integer that defines how to interpret the octet-string) were added to messages that did not already have them.
- Vendors and the IETF can use these holes to add new extensions.

Protecting Cleartext with Signed Type

- Messages containing cleartext fields have been wrapped in types containing keyed checksums.
- Guidelines clearly specify when checksums should be sent and what key is used.
- A checksum is added to the AS-REP in order to authenticate the AS-REQ.

The Golden Rule

Be liberal in what you accept and conservative in what you send. New Messages should only be sent when they will be understood by the recipient. Recipients should ignore extensions they do not understand, preserving them if the message is reencoded.

Determining Capabilities of a Recipient

- If you have received a new-format message then you can send new-format messages.
- The ticket type tells the client about server capabilities.
- For bootstrapping, a new AS-REQ can be included in an old AS-REQ.