SPNEGO, Kerberos, GSS-API and Negotiate support and how to make them better

Isaac Boukris
Nürnberg 2017

Agenda

- Brief review of authentication schemes
- HTTP Negotiate
  - Terminology
  - Advantages
  - How does it work
  - Typical usage and credentials acquisition
  - Problems and possible improvements

Authentication Schemes

- **HTTP (from RFC 7236):**

Authentication Scheme Name	Reference	Notes
Basic	[RFC2617], Section 2	
Bearer	[RFC6750]	
Digest	[RFC2617], Section 3	
Negotiate	[RFC4559], Section 3	This authentication scheme violates both HTTP semantics (being connection-oriented) and syntax (use of syntax incompatible with the WWW-Authenticate and Authorization header field syntax).
OAuth	[RFC5849], Section 3.5.1	

- **Form-cookie based**
- **TLS based**

Negotiate

- What is negotiated and how?
  - SPNEGO vs Kerberos
  - GSS-API vs SSPI
- Advantages
  - Centralized authentication model
  - Authenticated authorization data
  - Interoperability

Negotiate: how does it work

Curl built on UNIX with GSS-API:

gss_init_sec_context()

A service running on Windows:

AcceptSecurityContext()

Curl built on Windows with SSPI:

InitSecurityContext()

A service running on UNIX:

gss_accept_sec_context()

Essentially, GSS tokens are exchanged in a loop.

Negotiate: typical usage

 # curl -u: --negotiate http://host/

Credentials are acquired from the environment.

On Windows if username and password are specified they will be used (probably NTLM).

- Improving credentials acquisition in GSS-API
  - Tool vs library point of view

Negotiate: challenges

Problems:

- Connection oriented
  - Not required for Kerberos mechanism
- Posting problem
- Mutual authentication problem

Possible improvements:

- Allow specifying mechanism to use
- Allow fallback to other schemes
- Use gss-ntlmssp for NTLM
- Add tests