Kerberos and Single Sign-On with HTTP

Joe Orton

Senior Software Engineer, Red Hat

Overview

- Introduction
- The Problem
- Current Solutions
- Future Solutions
- Conclusion

Introduction

- WebDAV: common complaint of poor support for authentication in HTTP
- Kerberos is “The” network authentication protocol

The Problem

- How to integrate HTTP servers into a Kerberos infrastructure?
- Single Sign-On: reducing the number of times people enter passwords
- Ideal: user authentication exactly once per “session”; not per-server and/or per-service

The Problem: Scope

- Covering intranet/enterprise/organisation-wide HTTP authentication
- Out of scope: SSO for “The Web”
- In scope? Proxy authentication

GSSAPI vs HTTP

- GSSAPI: protocol-agnostic token-based API
- Authentication, optional integrity and/or confidentiality – but not really optional
- Confidentiality/integrity = transport layer
- In HTTP, authentication is independent from the transport layer

Current Solutions

- Stanford WebAuth: forms and cookies
- HTTP “Basic” authentication
- HTTP “Negotiate” authentication

Stanford WebAuth

- Cookie-based authentication
- Token-passing via browser redirects between web server and “WebKDC”
- Kerberos credentials passed to WebKDC via HTML form
- WebKDC passes token back to web server

Stanford WebAuth

- “Application layer” solution
- Cookies + HTML != HTTP authentication
- Requires SSL when passing credentials
- Requires a real web browser: won't work with generic WebDAV clients
- Requires a special server to be WebKDC

Stanford WebAuth

- Training users to enter Kerberos credentials into web forms is Very Bad™ - phishing
- Cannot authenticate to proxies
- Session termination? Flush cookies
- Session scope: within one web browser but then covers all servers

Kerberos via Basic Auth

- Use standard HTTP Basic authentication
- Send Kerberos credentials as Basic auth credentials
- Web server authenticates as user directly to KDC
- Works with any generic HTTP client

Kerberos via Basic Auth

```
GET /secret/ HTTP/1.1
HTTP/1.1 401 Unauthorized
WWW-Authenticate: Basic realm=”Blah”

GET /secret/ HTTP/1.1
Authorization: Basic QWxuIHNlc2FZQ==
HTTP/1.1 200 OK
```

Kerberos via Basic Auth

- Requires SSL when passing credentials
- Training users to enter credentials into HTTP authentication dialogs is also Very Bad™
- Can authenticate to proxies
- Session scope: one web browser, one server
- Session termination: flush cached credentials

The “Negotiate” Scheme

- New HTTP authentication scheme (kind of)
- Written by Microsoft; I-D published 2001
- Became “Informational” RFC 4559 in 2006
- Uses GSSAPI with “SPNEGO” for NTLM
- Implemented as HTTP client extension, custom server module

Negotiate: Protocol trace

```
1. GET /secret/ HTTP/1.1
2. HTTP/1.1 401 Unauthorized
   WWW-Authenticate: Negotiate [token]

3. GET /secret/ HTTP/1.1
   Authorization: Negotiate Y.....Q==
   [goto 2, or...]
   HTTP/1.1 200 OK
```

The “Negotiate” scheme

- Supported at HTTP client level; works with WebDAV etc
- Implemented by Firefox, MSIE
- Requires SSL to secure the connection
- Could almost work with proxies

The “Negotiate” Scheme

- Even the name is bad
- Per-connection authentication!
- Breaks RFC2617 challenge grammar
- Abuses RFC2617 headers

mod_auth_kerb

- Module for Apache httpd 1.3/2.x
- Maintained by Daniel Kouril, BSDy license
- Version 5.0 released August 2006, first non-beta release
- Supports both Negotiate and Kerberos-over-Basic authentication

mod_auth_kerb Configuration

- Obtain a service key from the KDC
- Name, for example: `HTTP/www.example.com@EXAMPLE.COM`
- Service key in keytab – check permissions!
- Load module and add access control configuration, either httpd.conf or .htaccess

Access control Configuration

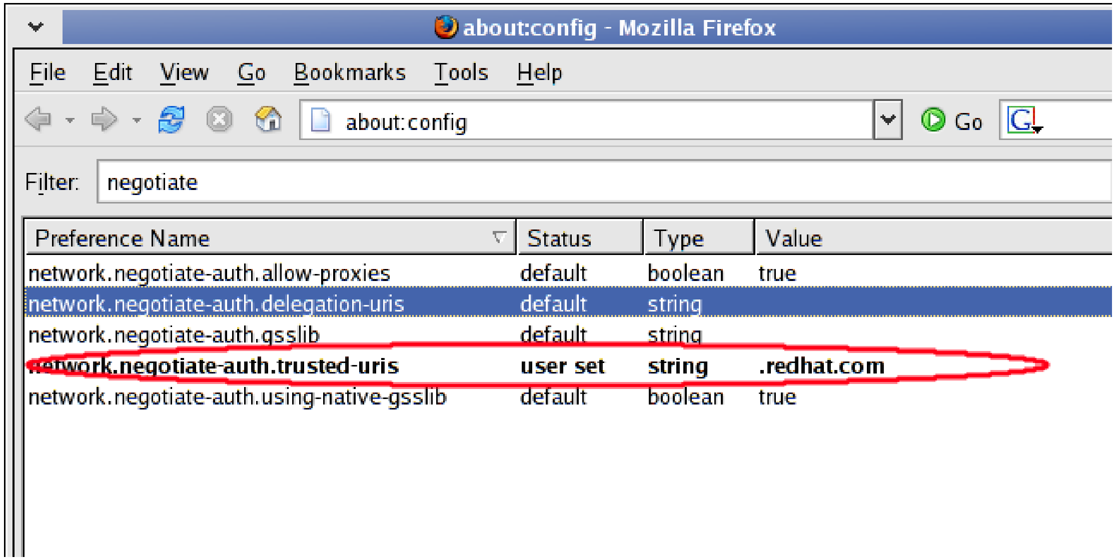
```
<Location /private>
 AuthType Kerberos
 AuthName "Kerberos Login"
 KrbMethodNegotiate On
 KrbMethodK5Passwd Off
            ...
```

Access control continued

```
    KrbAuthRealms  EXAMPLE.COM
    Krb5KeyTab /etc/httpd/conf/keytab
    require valid-user
    SSLRequireSSL
</Location>
```

Client configuration

- Firefox:


about:config - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
about:config
Go

Filter: negotiate

Preference Name	Status	Type	Value
network.negotiate-auth.allow-proxies	default	boolean	true
network.negotiate-auth.delegation-uris	default	string	
network.negotiate-auth.gsslib	default	string	
network.negotiate-auth.trusted-uris	**user set**	**string**	**.redhat.com**
network.negotiate-auth.using-native-gsslib	default	boolean	true

- MSIE should work within “Intranet zone”

Conclusion

- Strong authentication as an HTTP authentication scheme alone is not enough
- “Negotiate” is a practical if flawed solution for Kerberos Single Sign-On with HTTP
- But MUST be used over SSL

Future Solutions

- RFC2712: TLS with Kerberos ciphersuites
- Implemented in OpenSSL; no deployment
- A “GSSAPI Transport Layer” for HTTP?
- Implement via Upgrade: header (RFC2817)

Resources

- http://webauth.stanford.edu/
- http://modauthkerb.sourceforge.net/
- http://www.ietf.org/rfc/rfc4559.txt
- http://www.ietf.org/rfc/rfc2712.txt
- These slides: http://people.apache.org/~jorton/ac06us/

Q&A

Any questions?