Kerberosを用いたシングルサインオンの実装

笠原 卓也† 関口 聖美†† 黒羽 秀一†† 齋藤孝道†
†明治大学 ††明治大学大学院

1 はじめに

ユーザにとって，ネットワーク認証の数が増えると，ユーザは，認証システムごとに，パスワードを管理することになる．それを解決する一つの方法としてシングルサインオンがあり，シングルサインオンを実現する認証システムとして，Kerberos[1] が挙げられる．しかしながら，Kerberos を利用するには，仕様上，クライアントとサーバの全てを Kerberos に対応するように修正しなければならない．したがって，Kerberos に対応していない既存のクライアント，またはサーバにおいて，何らかの変更を行わずに，Kerberos を利用することはできない．

そこで，本論文では，ユーザ認証を行う複数の Web アプリケーションにおいて，クライアントである Web ブラウザとそれらの Web アプリケーションを Kerberos に対応させることなく，Kerberos を利用したシングルサインオンを実現するシステムを提案し，実装を示す．

2 Kerberos

Kerberos は，クライアントとサービスを提供するアプリケーションサーバの相互認証を「信頼できる第三者機関」によって行う認証システムであり，Windows, Mac, Unix のユーザ認証に採用されている．信頼できる第三者機関とは，アプリケーションサーバに代わって，クライアントを認証するサーバのことで，KDC (Key Distribution Center) と呼ばれる．また，クライアントとアプリケーションサーバは principal と呼ばれ，KDC には principal に関する情報，すなわち，KDC 上での名前 (principal 名)，パスワード，認可情報，パスワード変更や認証に関するタイムスタンプ (以降，principal 情報と表記) が保存される．principal を含む管理単位領域を realm と呼ぶ．よって，Kerberos 認証システムは，KDC と複数の realm から構成される．クライアントは，KDC の認証を一度受けていれば，自分と同じ realm に所属しているアプケーションサーバの提供しているサービスを利用することができる．

realm に所属するには，所属したいクライアント，またはアプリケーションサーバの principal が KDC に作成されていること (principal 情報が保存されていること) と，その principal が使用するアプケーションが Kerberos に対応していて，所属する realm の情報と認証を要求する KDC の情報が設定されていることが必要である．

本論文のシステムは，クライアントとアプリケーションサーバにおいて，Kerberos 認証システムを利用する上で，realm に所属する必要性をなくすものである．

3 提案システム

3.1 概要

提案システムは，クライアントである Web ブラウザと Apache[2] を利用した Web サーバ間に配置されるリバースプロキシである．このリバースプロキシは，KDC と同じ realm に所属していて，realm に所属していないクライアントに対する，KDC を用いた認証を行う．その後，KDC に保存されているクライアントの principal 情報を利用して，realm に所属していない Web サーバに適切な認証情報を送信する．

クライアントとリバースプロキシ間，リバースプロキシと Web サーバ間では，認証情報は Basic 認証 [3] の方式で送られる．

ここで，図 1 にクライアント，Web サーバ，提案システム，KDC の配置関係を示す．

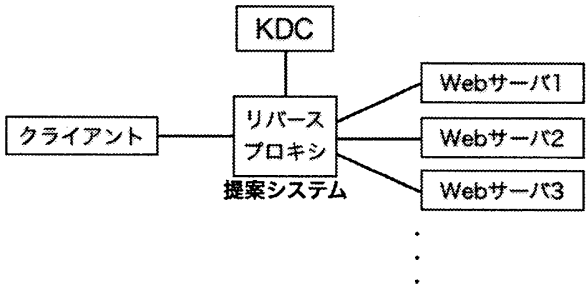

図 1: 提案システムの構成例

3.2 構成要素

クライアント

クライアントは，Internet Explorer や Firefox といった Basic 認証を利用することが可能な Web ブラウザであり，リバースプロキシと KDC を含む realm には所属しない．

Web サーバ

Web サーバは，Apache で，Basic 認証を行う．クライアントには，リバースプロキシを通してのみ，Web サーバにアクセスさせることとする．この Web サーバも，リバースプロキシと KDC を含む realm には所属しない．

リバースプロキシ

リバースプロキシは，Apache を改変したものである．この Apache には，Kerberos 認証を利用するためのモ

† Takuya KASAHARA
†† Kiyomi SEKIGUCHI
†† Shuichi KUROBA
† Takamichi SAITO
Meiji University (†)
Graduate School of Meiji University (††)

ジュール mod_auth_kerb[4] と，KDC から principal 情報を取得し，その情報から Web サーバの Basic 認証に使うクレデンシャル (ユーザ名とパスワード) を取り出し，それを Web サーバへのリクエストに付与する独自モジュールが用意されている．

リバースプロキシは，クライアントからのアクセスに対して，mod_auth_kerb モジュールの Basic 認証を利用した認証を行う．これにより，クライアントに Kerberos 認証で使用するクレデンシャルの提示を求める．

KDC

KDC は，krb5-server[1] パッケージを利用したものである．KDC には，リバースプロキシとクライアントの principal を作成しておく．リバースプロキシの principal は，krb5-server パッケージに含まれる管理ツールで作成した．クライアントの principal は，独自の principal 作成プログラムで作成した．

独自の principal 作成プログラムは，任意の ASCII 文字列を principal 情報に追加し，principal を作成するプログラムである．追加する文字列以外の principal 情報は，管理ツールで作成するときと同じものが格納される．本論文の実装では，このプログラムを使用して，Web サーバに送るクレデンシャルを，クライアントの principal 情報に格納した．

3.3 実装環境

Web サーバ，リバースプロキシ，KDC は Fedora Core 6 (kernel2.6.22) 上に構築した．Web サーバ，リバースプロキシで利用した Apache はいずれもバージョン 2.2.6 である．リバースプロキシの独自モジュールと，独自の principal 作成プログラムは，Fedora Core 6 (kernel2.6.22) 上で gcc-4.1.2 (C 言語) を用いて，Kerberos API と kadm5 ライブラリ [1] を利用して実装した．

3.4 動作

提案システムの動作を図 2 に，各動作の内容を以下に示す．図中の灰色枠の要素が，本論文で独自に実装したものである：

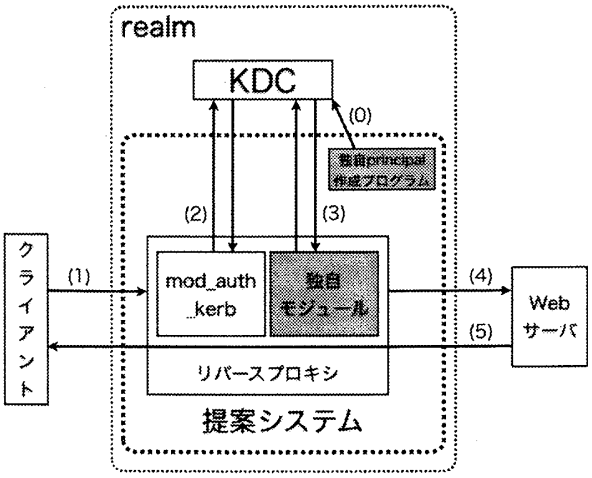

図 2: 提案システムの動作

(0) あらかじめ，独自の principal 作成プログラムで，Web サーバの Basic 認証に使うクレデンシャルを principal 情報に持つ principal を，クライアントの principal として作成しておく．

(1) クライアントからリバースプロキシへ，リクエストが送られる．リバースプロキシは，そのリクエストに対して，Basic 認証のクレデンシャルを求めるレスポンスを返す．そして，クレデンシャルの要求を受けたクライアントから，クライアントの principal 名とパスワードが Basic 認証のクレデンシャルとして付与されたリクエストが，リバースプロキシへ送られる．

(2) mod_auth_kerb モジュールは，クライアントから送られてきたクレデンシャルを，クライアントの principal 名とパスワードとして KDC に問い合わせ，その真正性を確認することで，クライアントの認証を行う．

(3) 独自モジュールは，クライアントから送られてきたクレデンシャルの principal 情報を利用して，クライアントの principal 情報を KDC から取り寄せる．その情報から Web サーバの Basic 認証に使うクレデンシャルを取り出し，それを Web サーバに転送されるリクエストに付与する．

(4) リバースプロキシは，独自モジュールがクレデンシャルを付与したリクエストを，Web サーバへ送る．

(5) クレデンシャルが正しければ，認証の成功を示すレスポンスが，Web サーバからリバースプロキシを通して，クライアントへ送られる．

4 まとめ

本論文では，Kerberos を利用することで，Web ブラウザと Web サーバ間の認証で，シングルサインオンを実現した．

本論文の実装では，クライアントの principal 情報に格納できるクレデンシャルは一つのみなので，認証を行う Web サーバが複数存在する場合，シングルサインオンを実現するためには，その全ての Web サーバにおいて同じクレデンシャルで認証が通るように設定しなければならない．したがって，今後の課題には，クライアントの principal 情報に複数のクレデンシャルを格納するためのフォーマットの策定と，リクエストを送る Web サーバによって適切なクレデンシャルを選択する機能の実装がある．

参考文献

[1] http://web.mit.edu/Kerberos/

[2] http://www.apache.org/

[3] http://httpd.apache.org/docs/2.2/howto/auth.html

[4] http://modauthkerb.sourceforge.net/