6S-09

パスワードを保持しない代行入力システムによるSSOの実現

田崎優典，安井浩之，横山孝典
東京都市大学

1. はじめに

Web サービスの普及に伴い，ユーザが管理すべき ID とパスワードが増えている．このような問題への対策として，一度の認証処理で連携するシステム全てを利用できるシングルサイオン(SSO)の仕組みがある．

SSO を実現する方法には，相互の認証結果を信頼しあうフェデレーション型やエージェント型などがあるが，いずれも Web サービスが SSO に対応していなければならない．

一方，ユーザの代わりに Web サービスの ID とパスワードを記憶し，代行入力する仕組みを用いれば，SSO に非対応の Web サービスにも導入可能であるが，パスワードをそのままシステムに保持させなければいけないというリスクがある．

そこで本研究では，ユーザのパスワードを保持しない代行入力システムを開発し，SSO を実現する．

2. 代行入力方式

コストや技術的な問題により，機能を拡張することが難しい既存のシステムに対して，SSO を実現する方法として代行入力方式がある．代行入力方式とは，認証時にユーザ ID とパスワードを自動的に代行入力することによって，擬似的な SSO を実現する方式である．しかし，代行入力システムは内部に生のパスワードを保持しておく必要があり，リスクが高くなる．

3. 提案内容

ID・パスワードの統合認証化を目的に，多くの Web サービスが LDAP 認証に対応している点に着目して，生のパスワードを保持しない代行入力システムを実現する．代行入力システムは，最初に Web サービスで認証を行う際，連携した LDAP サーバからユーザ情報を参照し，LDAP 上でハッシュ化されている既存のパスワード情報を一時的に保管しておく．次に，代行入力システムが一時的に生成したパスワードで LDAP 上のパスワード情報を上書きし，そのパスワードを Web サービスに代行入力して認証を行う．これにより，代行入力システムがパスワードを生の形で保持する必要をなくすことができる．

ユーザが Web サービスを利用するまでの流れは図 1 に示すようになる．

① ユーザが代行入力システムにアクセスし，認証を行う．
② 代行入力システムが LDAP 認証を行い，その後ハッシュ化した元のパスワードを保管し，一時的なパスワードを LDAP サーバに登録する．
③ 代行入力システムが web サービスのログイン画面にアクセスする．
④ ユーザ名，新規のパスワードを web サービスに送る．
⑤ Web サービスが LDAP 認証を行う．

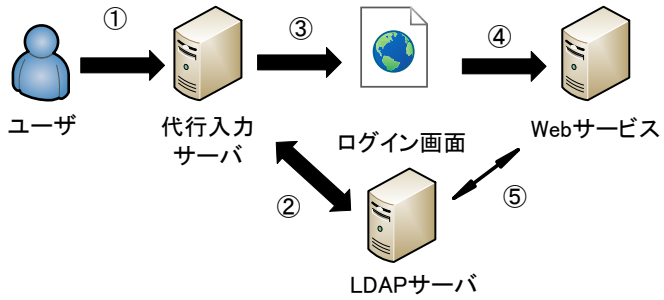

図 1　システム全体の流れ

4. 実現方法

代行入力システムをリバースプロキシ型として実装することにより，ユーザは Web サービスの ID とパスワードを入力せずに，Web サービスを利用できるようにする．

代行入力システムのほか，LDAP に OpenLDAP，Web サービスには WordPress をそのまま利用する．なお，本実装では Web サービスとして WordPress のみを対象としているが，リバースプロキシをアプリケーションゲートウェイとして実装し，複数の Web サービスに対応させることで SSO を実現できる．

5. 代行入力システムの実装

LDAP 代行入力システムのパスワード書き換え機能とリバースプロキシ機能の実装について説明する．

Realization of SSO method by reverse proxy systems without holding passwords
Yusuke Tasaki
Hiroyuki Yasui
Takanoru Yokoyama
Tokyo City University

パスワードの書き換え機能については，まず代行入力システムで認証を行う際に入力されたIDとパスワードでLDAP認証後，LDAPにハッシュ化されているユーザのパスワードを参照し，代行入力システムに保管しておく．次にLDAPのパスワード変更機能を用いて，代行入力システムが作成した一時パスワードに変更する．これ以降は生成した一時パスワードでWeb認証が可能となる．なお，Webサービスの利用終了時にはLDAPの更新機能を用いて，保管されている元のハッシュ化されたパスワード情報で一時パスワードの情報を上書きする．

リバースプロキシ機能については，PHPのライブラリであるPEARのHTTP_Clientを利用し，図2に示すようにユーザからのHTTPリクエストとWebサービスからのHTTPレスポンスを中継している．その際，絶対パス指定されたURLの書き換えなどの処理を行っている．

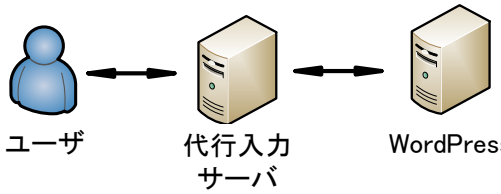

図2　代行入力システムの流れ

6. 評価

提案手法により一時パスワードに変更してからWebサービスにログインする場合と，パスワードを変更せずにそのままWebサービスにログインする場合のレスポンス時間，LDAPサーバのCPU負荷率を比較する．

測定にはJMeter[1]を用い，5~30人分のユーザ認証を同時に行ったときのレスポンス時間（すべてのアクセスが終了するまで）と，その時のLDAPサーバ上でのCPUの負荷率の中央値（試行回数5回）をそれぞれ求めた．なお，Webサービスの付加による影響を排除するため，計測は代行入力システムがWebサービスへユーザ認証を行う直前までの時間とした．

図3がレスポンス時間，図4がLDAPに対するCPU負荷率の結果である．

Webサービスを大学等の授業で利用すると仮定した場合，50人程度が同時にアクセスすることが想定される．仮に50人同時アクセスの場合，書き換え無しが29.4ms，書き換えありが2323msになった．しかし，一斉にアクセスしようとした場合でも同時にアクセスはなかなか起こらないと考えられる．また，50人が1秒の間でアクセス(同時ではない)する場合ならば1312ms，2秒の間なら338msになった．

ニールセン[2]によると，レスポンス時間は1秒までならば良質な応答時間とユーザは感じることから，書き換えを行っていても不満なく操作をすることができると考えられる．

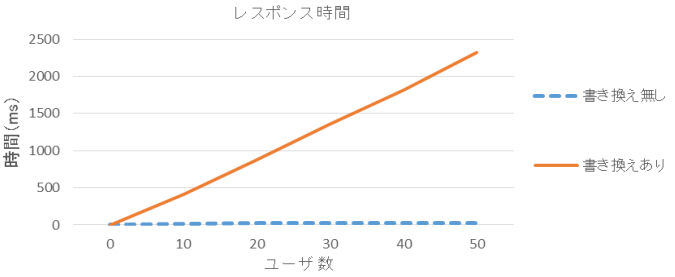

図3　レスポンス時間

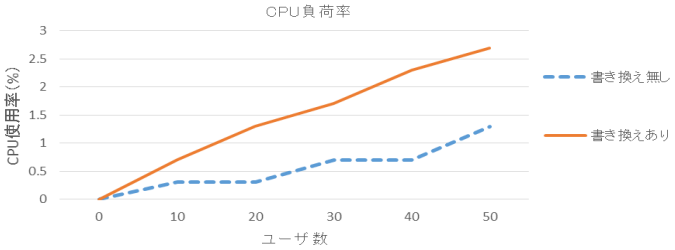

図4　CPU負荷率

7. まとめ

パスワードを保持しない代行入力システムによるSSOの実現手法として，LDAP上のパスワードを一時的に書き換える手法を提案し、リバースプロキシ型のシステムとすることでSSOが可能なシステムを実現した．

今回はLDAPを重点的に測るため，Webサービスと連携せず測定した．より正確な数値を取るためにはWebサービス側もアクセス数に耐えられるようして，代行入力システム全体として評価を行う予定である．

今後の課題として，フェデレーション型などのSSO[3]への対応や，さらに負荷をかけた場合でも認証の速度維持についても改善を行っていきたい．

参考文献

[1]JMeter　http://jmeter.apache.org/
[2]ヤコブ・ニールセン著，三好かおる訳，2002.7，東京電機大学出版局「ユーザビリティエンジニアリング原論：ユーザのためのインターフェースデザイン」
[3] 渡辺健次，大谷誠，江藤博文，「全面的に Shibboleth に対応した佐賀大学の学術情報基盤システム」，教育システム情報学会研究報告 25(3) 43-48,2010