Vol.2011-CSEC-54 No.29
Vol.2011-SPT-1 No.29
2011/7/13

WEB アカウント自動収集による省力的疑似 SSO 環境の研究

森岡卓哉　大西克実　中野秀男

大阪市立大学 創造都市研究科 知識情報基盤分野

既存 WEB アプリケーションのアカウントをリバースプロキシでアカウント収集し，擬似的シングルサインオン環境を簡便に実装する方法を作成する リバースプロキシにパスワード管理機能を実装，リバースプロキシにパスワード・キーチェインを作る．プロキシサーバーが WEB アプリケーションに代理ログインさせる．新しい認証方式に対応させ、既存 WEB アプリケーションの延命を図るネットワーク装置の開発を目標としている．

Pseudo SSO for Web Application supported by Reverse Proxy Auto Login System

Takuya Morioka/Katsumi Onishi/Nakano Hideo†

Today many web applications are used. We have many accounts, and want to connect them. Taking a look Web browser function, many browsers has master password. So we mimic that function and make it into reverse proxy and that we can easily gather our accounts and password. Legacy web application with form based authentication will be able to live another long time life

1. はじめに

1.1 既存 WEB の認証対応.

WEB でアプリケーションを動かすことは一般的になった. アプリケーションは多様化しそれぞれに認証が必要になっている．大学も統合認証が進み，ワンアカウントでサービス提供が一般的になった.

1.2 統合認証に未対応の WEB をどうするか

統合認証で便利になった．一部システムなどで新しい認証に未対応のままのモノが多くある．これらシステムが認証に対応できないという理由で放棄・新設するとコストが掛かる．また認証の非対応を理由に，導入時に選択肢から除外される OSS アプリケーションが数多くあると懸念される.

本研究では，ブラウザのパスワードでのログイン情報保存を参考に手軽な統合アカウントを実現，これを組込んだマシンをローカル環境で稼働させ，追加開発が起きないような方法を考えた．この実現には従来からのリバースプロキシ便利でそれ以外に選択肢は考えにくくリバースプロキシを強化する案について論じる.

2. アカウント統合作業について

WEB アカウント統合では，パスワード保存先の共通化（LDAP など）やデータベース共有，認証 API（OpenID，OAuth など）を HTTP のドメイン Cookie と組み合わせて使う．これらを勘案しながらアカウント統合は行われる.

その際にシングルサインオン製品を導入し，URL 単位でのアクセス制御を登録し実装．ソフトウェアにより Cookie(HTTP Header)の共有化を実現．または，LDAP などにまとめてパスワードを保存する事が考えられる．しかし旧システムはバージョンアップとモジュール導入テストが必要になったりする.

アカウント統合作業ではユーザー毎に情報をまとめていくが，ユーザー固有情報はデータベースのテーブルスキーマが異なる，ハッシュ化方式が異なるなど差異が多く容易ではない．そのためシステム間パスワードの共有は難しい．他にソースコードのメンテナンスの問題がある．追加認証モジュールを導入するとバージョンアップやセキュリティー対応パッチを当てるためにチェック事項が多くなる．このような対応作業を見積もると，それなりの経費が必要になる．追加開発と設定がオープンソースの良さをともに殺してしまう．また，○○認証対応済みの狭まった選択肢の中で WEB アプリケーションを選んでいると，ベンダロックインが解決せず，狭まった選択肢に足りない機能を追加開発するといった事例が起きる.

Vol.2011-CSEC-54 No.29
Vol.2011-SPT-1 No.29
2011/7/13

特定のベンダや認証方式に依存せず，簡単にアカウント統合実現する選択肢を増やしたく，調査を行った．リバースプロキシを利用すれば．シンプルで簡便な実装が出来るはずと仮定し，プログラム作成を行って実験を行った．

3. 代理ログインを用いた単純認証連携

フォームを使った認証が WEB アプリケーションでは一般的に使われる，そこでフォーム認証機能を拡大した．フォーム認証をプロキシ側で代理ログインすることにした．

1. ユーザーはリバースプロキシにログインする
2. ユーザーが WEB アプリケーションへアクセスする
3. リバースプロキシ側で，フォーム認証を行う．
4. ログイン済みセッション（Cookie）を返す．

リバースプロキシでユーザーのフォーム入力を代行することにした．これであれば数多くの WEB アプリケーションに対応が可能である．

本研究はこの代理ログインをメインに据え，認証方式そのものではなく機能強化を実験し提案するものである．

3.1 フォーム認証の自動化について

フォーム認証の自動化については，シングルサインオン製品で実装されている事もあるが，ロジックを一般化し，シンプルに実装する事にした．ログインフォームを次のように一般化した

- HTML(content-type: text/html)
- Form 要素
- input[type=text]要素：１ケ
- input[type=password]要素：１ケ

これらを全て満たすログインフォームと見なすおことにした．ちなみに password 要素が複数あるモノは必ずパスワード変更画面であった．またパスワード要素のみの場合は，セッションタイムアウトで，それらは再度ログインすることでログインフォームを使うので回避可能であった．

4. 代理ログインの動作

リバースプロキシでユーザーとサーバーの HTML を監視し代理ログインを行う事になった．ログインフォームを探すログインフォームが見つかった場合には，リバースプロキシ側でパスワードを入力・送信する．ログイン成功後，ユーザーにセッションを引き継がせる．このさいログインフォームの URL を学習し，他ユーザーにも再利用して HTML 内部を探す手間を省略した．

学習していないログインフォーム
ユーザーにパスワードを入力送信してもらう
リバースプロキシ側でパスワード保存して良いかユーザーに尋ねる．
ユーザーの同意の下でパスワード保存し，次回から代理ログインを行う

4.1 実験リバースプロキシについて．

本実験では Ruby を用いてリバースプロキシとアプリケーションを作成した．Ruby を用いた理由は HTML 解析の容易さである．Ruby と XPATH でログインフォーム検出は数行で書くことが出来た．またパスワード保存は LDAP と RDBMS に保存した．

4.2 パフォーマンスについて

リバースプロキシを挟むことによる速度低下の懸念があった．リバースプロキシを経由することによる速度低下は，従来の速度に＋３００ｍｓ程度であった．

静的コンテンツ（画像，スタイルシート）は，ログインフォームチェックをせず素通しすることで速度を稼ぐことにした．HTTP の Content-type ヘッダに注目し text/html だけをリバースプロキシの処理対象とし，その他コンテンツには一切処理を挟んでいない．

動的ページ（HTML）部分に関しても＋３００ｍｓ程度の速度低下が見られた．しかし動的ページはもともと応答時間が大きいため，リバースプロキシによる速度低下は目立った差に表れなかった．元々７００ｍｓの応答速度が１０００ｍｓに伸びても体感速度に影響が見られなかった．

5. アカウントの自動収集について

WEB サーバーのアカウント統合を行うため，リバースプロキシでアカウントの自動収集を実験した．これを無断で行うことは問題を引き起こすが，今回の目的は，アカウント統合作業負荷を減らすことにあり，組織内で行う事が前提であり問題は少ないと考えている．

5.1 自動収集したパスワードの保存について

代理ログインのためパスワードを保存し再利用する．にパスワードはプレインテキストで保存される．滋賀って厳重な取り扱いが必要である．そこでパスワード保存はリバースプロキシではなく，外部から切り離された LDAP サーバーや RDBMS に暗号

Vol.2011-CSEC-54 No.29
Vol.2011-SPT-1 No.29
2011/7/13

化保存する事でセキュリティーレベル確保できると考えている.

5.2 HTTPS 通信について

リバースプロキシは HTTPS 通信を復号化するわけではない. リバースプロキシは HTTP の通信が見られる場所に設置し, SSL アクセラレーターとともに用いることで HTTPS 通信環境下でも代理ログインが可能になる. 検討の結果この手法が従来の WEB との親和性が高い.

6. まとめ

代理ログインを用い, パスワードを自動収集するリバースプロキシを設置することで, WEB アプリケーションで LDAP 非対応のもの, 情報部門お手製 WEB アプリケーションで使うパスワードを簡単に統合アカウントに紐づけることが出来る. パスワード保存には厳重な注意が必要だが, 組織内部で使うには十分であると考えられる.

WEB アプリケーションをしばらく延命させ, 非対応 WEB アプリケーションでもユーザー視点からは利便性を損なわず利用が可能になる.

ブラウザのパスワード保存機能をユーザーに利用されるより, サーバーに一括保存していく方が漏洩ポイントを減るために安全性は高いと考えられる. また経路上に設置するだけの小さなアプライアンスボックスとして手軽に導入できる可能性がある. 企業の製品版が高価なため必要な機能だけに絞った簡易リバースプロキシ SSO が OSS で提供されることに意味があるのではないかと考えている.

参考文献

1) 西村 健,佐藤 周行:レガシーWeb アプリケーションに対応する PKI を用いた簡易 Single Sign-On の実現, 情報処理学会研究報告. QAI, [高品質インターネット] 2007(3), 61-65, 2007-01-18
2) オブジェクト指向スクリプト言語 Ruby,http://www.ruby-lang.org/ja/
3) 植田 浩光,力宗 幸男:エージェントサーバにおけるパスワードの保護方法について,電子情報通信学会技術研究報告. ISEC, 情報セキュリティ 110(281), 81-86, 2010-11-10
4) 野林 大起,中村 豊,池永 全志:ハードウェアトークンと鍵管理サーバを用いたシングルサインオンシステムの開発,電子情報通信学会技術研究報告. IA, インターネットアーキテクチャ 106(62), 7-12, 2006-05-17
5) 土屋 英亮,丸山 一貴,高田 昌之:ゆるやかに結合した LDAP ツリーを用いた統合認証システムに関する研究,IPSJ SIG Technical Reports 2008(72), 23-28, 2008-07-17