

Outsourcing Auth and Auth (with Web Services)

Dave Burchell

2006-08-10

Copyright © 2006 HeroicMarkup.com

What are “Outsourcing,” “Auth,” & “Auth?”

What are “Outsourcing,” “Auth,” & “Auth?”

The talk is about “Outsourcing Auth & Auth (with Web Services).” What does this mean?

Authentication

Proving you are who you say you are

Authentication may be based on:

- **Something you know (e.g., a password)**
- **Something you have (e.g., a house key)**
- **Something you are (e.g., your fingerprint)**

If you have a Website and want to know who is asking for accessing it (or requesting access), then you need a way to authenticate.

Authorization

Proving you should have access to a resource you request

If you have a Website where you want to restrict access to only certain people, you need to authorize them.

Outsourcing

Transferring an ancillary function to another entity

- **Not to be confused with “offshoring”**
- **Can contribute to creation of “virtual” companies**

Web Services

Application-to-application programmatic interfaces using Web protocols

Think of them as Web browsing for computer programs.

- Nearly always use HTTP, XML
- Often use SOAP and REST
- Commercially available
- Facilitate outsourcing

Why Outsource?

Web Services facilitate outsourcing. But why outsource?

- **The internal group lacks expertise (*Barcode Reader*)**
- **The system or resource is burdensome to maintain (*tax rates*)**
- **Special equipment is required (*phone verification*)**
- **Need access to a controlled information resource (*D-U-N-S*)**
- **The resources cannot be spared to develop the feature or subsystem at the moment (*CAPTCHA*)**

A Simple Real-World Scenario

A Simple Real-World Scenario

A family reunion is coming up. You are helping to plan and organize it.

Scenario

- **Relatives in Nebraska are hosting the reunion**
- **Relatives in other places are planning to attend**
- **You want to divide the universe into three groups:**
 - **Nebraska relatives (hosts)**
 - **Non-Nebraska relatives (attendees)**
 - **Others (non-family)**

Stand-Alone Authentication

If you wish to build stand-alone authentication, you could:

1. **Gather a list of each relative's email address**
2. **Prepare a list of usernames (email addresses?) and generate a random password for each**
3. **Email the passwords to each relative**
4. **Put the user list on your system (e.g., .htpasswd file)**

Stand-Alone Authorization

You could:

1. **Sort the relatives usernames by Nebraska/Non-Nebraska**
2. **Protect the entire reunion site by requiring users to be authorized (e.g., .htaccess file, Unix username established).**
3. **Protect the reunion host section of the site with an access control list (ACL), Unix group, etc. to allow only Nebraska relatives to access the Nebraska section**

QUICK QUIZ

Q: When Great Uncle Hollis forgets his password, who will he ask for help?

QUICK QUIZ

A: You

QUICK QUIZ

(Bonus question: Who will he call if he's been placed in the wrong group?)

Outsourcing Auth & Auth

Outsourcing Auth & Auth

You can avoid your uncle's phone calls if you let him contact someone else for help. If you outsource the functions of auth & auth, you can outsource the support of your users (relatives).

Authentication with PayPal

PayPal's Web Services are designed to allow merchants to interact behind-the-scenes with PayPal. A PayPal feature called Express Checkout can be used to authenticate that a user has the email address he or she claims to have.

The process (from an API Sourcebook page on Express Checkout [http://paypaltech.com/Dave/api_sourcebook/html/shell/shell_ec.html]):

1. Call *SetExpressCheckout*
2. Send the buyer to the PayPal site with the token. Example:

```
https://www.sandbox.paypal.com/  
cgi-bin/webscr?  
cmd=_express-checkout&token=EC-29879344RH8987624
```

3. Call *GetExpressCheckoutDetails*
4. Call *DoExpressCheckoutPayment*

By using the first three of the four steps only, you can verify that the user was able to log in to PayPal with the email address they claimed. From the *GetExpressCheckoutDetails* call:

Authentication with PayPal (Continued)

```
<Payer xsi:type="ebl:EmailAddressType">dave_buyer@fake.com</Payer>  
<PayerID xsi:type="ebl:UserIDType">RRJPLTMFCREJQ</PayerID>  
<PayerStatus xsi:type="ebl:PayPalUserStatusCodeType">verified</PayerS  
<PayerName xsi:type="ebl:PersonNameType">  
  <FirstName xmlns="urn:ebay:apis:eBLBaseComponents">Dave</FirstName>  
  <LastName xmlns="urn:ebay:apis:eBLBaseComponents">Burchell</LastNar  
</PayerName>
```

For details, see http://paypaltech.com/Dave/api_sourcebook/html/example_SOAP/ec_examples/get_ec_ex.html.

Authentication with Strikelron

Strikelron's Real Time Telephone Verification lets you verify that a person is calling from the phone number they claim. It is implemented as a Web Service using SOAP and HTTP.

Once you know a user's phone number, you can uniquely identify that user by phone number. If you have a list of authorized users's phone numbers on your Website you can compare against it.

(See Resources slide for a demo of another Strikelron service.)

Auth/Auth Using Flickr and Strikelron

Q: "Who is this dog?"



Auth/Auth Using Flickr and Strikelron

A:

```
m/r[ie]{1,2}ll?[ie]{0,2}y?[ie]{0,2}/i
```

(Use a regular expression to check for the name “Rilley” and all for its likely misspellings.)

Only relatives will know this dog's name (we assume). Use this to grant authorization.

- 1. Authenticate the user's phone number using Strikelron**
- 2. Quiz the user on the dog's name (only allow a few tries per phone number)**
- 3. Check the area code to see if it is a Nebraskan or non-Nebraskan**

We now know who it is (what phone number) and if they are a Nebraska relative or non-Nebraska relative. Our auth & auth is complete.

Conclusion

Conclusion

Outsourcing Advantages

- **Fast**
- **Easy**
- **Cheap***

Outsourcing Drawbacks

- **New skills may be required**
- **Expensive**
- **What if the service goes away?**

Other services to use in outsourcing?

- eBay?
- Facebook?
- Friendster?
- Google?
- LinkedIn?
- Microsoft/MSN?
- Monster?
- MySpace?
- Tru?
- VeriSign?
- Yahoo!?
- YouTube?

Who lets us store data? Google Base? (Encrypt it, *a la* Unix passwords.) PayPal? (yes!)

And while we are thinking about it: Why don't people have to pay you to get authorization send you email? (“Monetize your inbox!”)

Resources

- **Strikelron** [<http://www.strikeiron.com>]
 - **Demo**
[<http://www.heroicmarkup.com/~burchell/O/Strikelron/CartUploadTax/CartUploadTax.cgi>] of Strikelron's Web Services
 - **Reverse Phone Lookup:** Look up a residential phone number and get associated phone book information, such as name and address. <http://www.strikeiron.com/ProductDetail.aspx?p=157>
 - **Real Time Telephone Verification:** Verify a site's visitor is really at the phone number she claims by placing a phone call to the number. <http://www.strikeiron.com/ProductDetail.aspx?p=219>
 - **Text Disguise CAPTCHA-Image Service:** make sure you are dealing with a human, not a computer program. <http://www.strikeiron.com/ProductDetail.aspx?p=196>
- **Google Base** [<http://www.google.com/base>]: post information publicly
- **PayPal** [<http://www.paypal.com>]: send and receive money
 - **Simple PayPal Web Services example code in the API Sourcebook** [http://paypaltech.com/Dave/api_sourcebook/html/].
 - **PayPal Integration Center** [<http://paypal.com/integration/>].

Resources (Continued)

- **Flickr [<http://www.flickr.com>]: share photos**

Q & A

Questions?

- <http://www.heroicmarkup.com/O/cert>
- burchell@heroicmarkup.com
- Skype me! User ID: “evaddnomaid”
- Search me out on gmail, Yahoo!, MySpace, eBay, etc.