

# **Mistaken Identity**

**How Not To Build an Account Recovery Process** 

Nick Freeman
Senior Security Consultant
Security-Assessment.com

Copyright © The OWASP Foundation Permission is granted to copy, distribute and/or modify this document under the terms of the OWASP License.

# The OWASP Foundation <a href="http://www.owasp.org/">http://www.owasp.org/</a>

### **Introduction**

- How users can regain control of their account after forgetting their password
  - ▶ Forgotten your password?
  - Reset your password
  - Send me my password
  - ▶ Help! I can't Access My Account!
- Why talk about it?
  - ▶ I encounter too many webapps that screw this up
  - ▶ The consequences can be dire



### What we'll cover

■ Username enumeration

■ Not-so-secret questions

■ 'Send me my password'

■ Other Bad Ideas

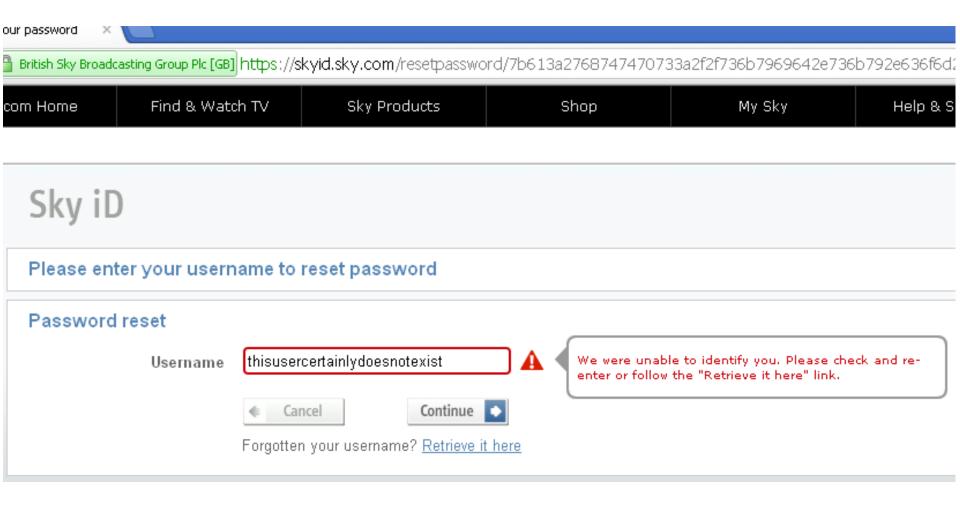


### 1 – Username Enumeration

- The first stage of the recovery process asks for a username / email address
  - If the username exists, no email/notification is sent to the user
  - No CAPTCHA is in place



### 1 – Username Enumeration



### 1 – Username Enumeration

# The problem

- User not notified of password reset initiation
  - Provides a simple true/false condition for username enumeration

Usernames are ½ of account brute-forcing



### 1 - Username Enumeration

- Send an email to the user when recovery is initiated
- Don't immediately reset user passwords
- A CAPTCHA will ease the symptoms but not solve the underlying issue



# 2 – Not-So-Secret Questions

- The application allow unlimited secret answer attempts
- Limited choice of secret questions with a finite answer set – for example:
  - What is your favourite sport?
  - What was the make of your first car?
  - What is your favourite colour?
- AND/OR, questions which can be answered by looking at someone's Facebook profile (e.g. DOB, first school, MMN)



# 2 - Not-So-Secret Questions

#### My Lycamobile - Online Registration

Please complete the details below to register for your free credit, to top-up, to receive your FREE Lycamobile saving card and for My Lycamobile. You will need your SIM holder with the PUK code in order to register.

<sup>\*</sup> Your Lycamobile Saving card will be sent to your address within 7 working days.



<sup>\*</sup> As a valued customer and in recognition of your commitment to Lycamobile you will receive £2 free credit once you have registered your details below and purchased 2 Lycamobile top ups. Your free credit pin will be sent to your address within 7 working days of your second top-up

# 2 – Not-So-Secret Questions

## The problem

- Secret answers can be brute forced
- Many user bases will have similar interests
  - If 'allblacks' is the most popular .NZ password...
- Social networking vastly increases the amount of info available on a target
  - Not as much of a problem for big sweeping brute force attacks, but a big problem for targeted attacks



# 2 - Not-So-Secret Questions

- DON'T ALLOW UNLIMITED GUESSES!
  - Consider lockout / contact customer support after 5 wrong guesses
- Choose (multiple?) questions with many possible answers
  - Let users choose their own question
  - First teacher
  - First home phone number
  - Favourite TV/Movie character
- Require the user to have performed an out of band (email/SMS) check before this step

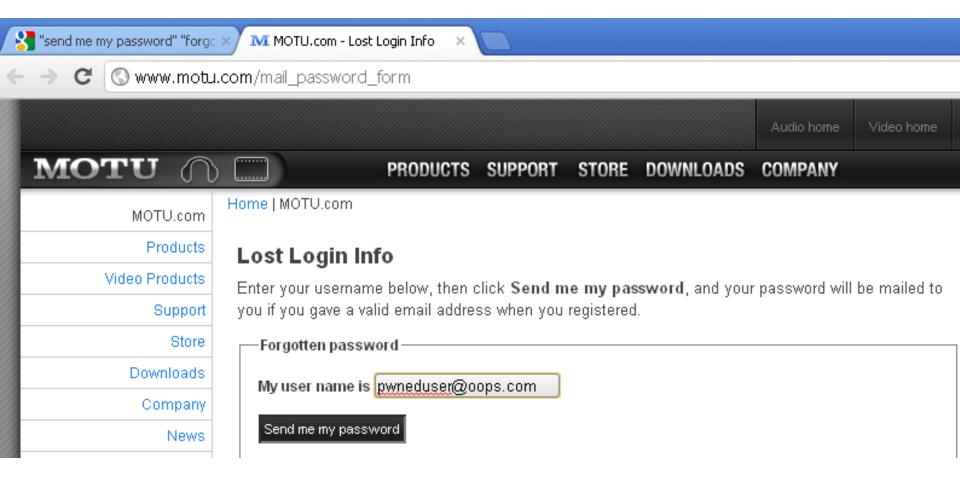


# 3 – 'Send Me My Password'

- A temporary (often weak) password is sent via Email (often without Q/A), or worse:
- Their current (stored plaintext..) password is sent via Email (often without Q/A), or worse:
- Their password is simply displayed to them through the application (rare but not extinct).



# 3 - 'Send Me My Password'



# 3 - 'Send Me My Password'

# The problem

- Passwords stored in plaintext :(
- If the user's email account is compromised, their account is toast
  - If the users reuse passwords (which they do) then several accounts could be compromised
- Many applications don't force users to change temporary passwords



# 3 – 'Send Me My Password'

- DON'T STORE PLAINTEXT PASSWORDS!
  - Seriously. This ^
- Don't Email passwords (temporary or otherwise)
- Email a single-use link with a random token (e.g. GUID) then get them to answer a question
  - Ensure the link expires after an hour
  - Additional layer of defense for users with compromised email accounts



### 4 - Other Bad Ideas

- Poor / Lack of input filtering
- UserID can be specified in the 'choose a new password' phase
- No XSRF protection
- App served unencrypted over HTTP



### 4 - Other Bad Ideas

#### The problem

- SMTP injection User password / token sent to bad guy
- XSS secret answer / new password sent to attacker
- HTTP Parameter Pollution (HPP)
  - e.g.: http://a.com/?email=attacker@ownyou.com& username=attacker\_account&username=victim\_account
- Reused functionality users can change any user's password
- XSRF to change a user's password for them



### 4 - Other Bad Ideas

- Filter all inputs!
- Store the userid of the user in the session, server side
- Use random form tokens for XSRF protection
- Serve the app over HTTPS



# My idea of a safe password reset process:

- 1. User supplies email address or username
  - ▶ CAPTCHA required & Input filtered
- 2. Application emails single-use random link to user
  - ▶ Token sufficiently random, expires after a set period of time
- 3. User visits link and answers one or more complex secret questions
  - ▶ Limited number of attempts to answer correctly
- 4. User is forced to choose a new, complex password
  - ▶ Password is hashed before being stored in the database



### **Conclusion**

- Secure password reset is not hard but there are a lot of things to take into account
- The sensitivity of your application may demand more stringent measures (reset code sent via SMS, more stringent lockouts)

https://www.owasp.org/index.php/Forgot Password Cheat Sheet - OWASP Cheat Sheet for Forgotten Password functionality