CSE 409, Fall 2011, Rob Johnson, http://www.cs.stonybrook.edu/~rob/teaching/cse409-fa11/
Alin Tomescu, December 5th, 2011

Phorcefield

Phorcefield overview

Motivation: Many users still type in their password into phishing pages that do not display their SiteKey image.

With Phorcefield, the **password is no longer text** it is actually a sequence of 12 images.

When a user registers, he picks 12 images from a set of a million images or something and then he picks an order for his 12 images, which he needs to remember in order to login.

How it works:

- Do a condition safe ceremony and set a secure cookie with the user's 12 images in it (SSL only cookie for the bank) on the user's browser
- At login, the bank retrieves the cookie and displays the 12 images, and now the user has to put them in the right order to login

Difference from SiteKey: Now the user has to see the images to login.

Phisher's options:

- Phisher could build a search interface where he shows you the million images and then he basically says "We have lost your password, please help us recover it"
    - One fix is to allow users to have one of the images be a private one
    - Usually, users get so frustrated with finding their images that they give up and they end up calling the bank.
- Guess images (too many images)
    - Unfeasible

User study:

- 100% of people could not reconstruct the password when presented with a phishing page of 1,000,000 images.

People have also looked into **face image passwords**, but…

- Users tend to choose the most attractive faces
- Users tend to choose faces of people with similar race

Another idea is to use a **big image password**, where you have a single scene image and you have to click on points in a scene to login, however this presents some **problems:**

- Hotspots. People like to click on certain things, like the corner of things.
- People choose simple click patterns like a straight line in the middle of the image.

Good things to have in a login system:

- **Attack scenario should look as different as possible from non-attack scenarios**, so it's obvious to the user when he's being phished
- Break **click-whirr behavior** or make click-whirr behavior safe (like condition safe ceremonies do)
- Part of the reason Phorcefield has succeeded is because **you could not ignore** it. This is called a "**forcing function**".
    - Example: you cannot start engine unless your foot's on the break

CSE 409, Fall 2011, Rob Johnson, http://www.cs.stonybrook.edu/~rob/teaching/cse409-fa11/
Alin Tomescu, December 5th, 2011

Device pairing

If there is no certificate authority (CA) or no pre-shared secret, then there is no way to prevent a man in the middle attack (MITM) during an authentication/key-exchange protocol.

General proposal for solving this: The human holding the laptop and the phone should be the trusted third party.

The devices can do a Diffie-Hellman key-exchange and then each device can display the hashed key to the user. The user can then check that the two devices have the same key. The problem with this is that the user will often ignore the hash checking messages (dialog fatigue).

Solution: Force user to type in the hash of the key on one of the devices.

Instead of presenting the hash of the key to the user, the device can prompt the user and ask him which one of the following hashes is displayed on the other device. This **forces the user to pay attention**.

Another proposal:

- Use **slower medium** to do key-exchange, like sound, and ensure that transactions take a certain amount of time, such that if there were a MITM, he would slow down the transmission considerably and be detected, unless he was close around you in which case you would see him
- Use **infra-red**, since it only works around you, so you would see the MITM
- **Vibration** (hold two phones next to each other, one phone vibrates a one or zero pattern to send the $g^a \ mod \ p$ and $g^b \ mod \ p$ Diffie-Hellman numbers to each other)
- **Bump** (take your device and gently bump your phone on the other phone, each phone will send a message to a bump server, which will connect the two phones who sent the messages)
    - However, this turns out to be insecure.