All Your Face Are Belong to Us: Breaking Facebook's Social Authentication

Iasonas Polakis[*]
FORTH-ICS, Greece
polakis@ics.forth.gr

Marco Lancini
Politecnico di Milano, Italia
marco.lancini@mail.polimi.it

Georgios Kontaxis
Columbia University, USA
kontaxis@cs.columbia.edu

Federico Maggi
Politecnico di Milano, Italia
fmaggi@elet.polimi.it

Sotiris Ioannidis
FORTH-ICS, Greece
sotiris@ics.forth.gr

Angelos D. Keromytis
Columbia University, USA
angelos@cs.columbia.edu

Stefano Zanero
Politecnico di Milano, Italia
zanero@elet.polimi.it

ABSTRACT

Two-factor authentication is widely used by high-value services to prevent adversaries from compromising accounts using stolen credentials. Facebook has recently released a two-factor authentication mechanism, referred to as Social Authentication, which requires users to identify some of their friends in randomly selected photos. A recent study has provided a formal analysis of social authentication weaknesses against attackers inside the victim's social circles. In this paper, we extend the threat model and study the attack surface of social authentication in practice, and show how any attacker can obtain the information needed to solve the challenges presented by Facebook. We implement a proof-of-concept system that utilizes widely available face recognition software and cloud services, and evaluate it using real public data collected from Facebook. Under the assumptions of Facebook's threat model, our results show that an attacker can obtain access to (sensitive) information for at least 42% of a user's friends that Facebook uses to generate social authentication challenges. By relying solely on publicly accessible information, a casual attacker can solve 22% of the social authentication tests in an automated fashion, and gain a significant advantage for an additional 56% of the tests, as opposed to just guessing. Additionally, we simulate the scenario of a determined attacker placing himself inside the victim's social circle by employing dummy accounts. In this case, the accuracy of our attack greatly increases and reaches 100% when 120 faces per friend are accessible by the attacker, even though it is very accurate with as little as 10 faces.

Categories and Subject Descriptors

K.6.5 [**Security and Protection**]: Authentication

General Terms

Security

Keywords

Authentication, Face recognition, Online social networks

1. INTRODUCTION

Online social networks (OSNs) have become some of the fastest growing Web services with a massive user base and, at the same time, an appealing target for malicious activities: Facebook reports over 900 million active users as of March 2012, all the while encouraging its users to share more and more information online for a richer experience. Such accumulated data and the interconnections between users have made OSNs an attractive target for the Internet miscreants, which harvest account credentials using both technical and social-engineering attacks. Studies [22] have shown that traditional underground economies have shifted their focus from stolen credit card numbers to compromised social network profiles, which are sold for the highest prices. A recent study [12] reports that the vast majority of spamming accounts in OSNs are not dummy profiles created by attackers, but legitimate, existing user accounts that have been compromised. Additionally, new Facebook phishing attacks use compromised accounts to steal personal information [15].

As a standard method for strengthening the security of online user accounts, high-value services such as online banking, and recently Google services, have adopted two-factor authentication where users must present two separate pieces of evidence in order to authenticate. The two factors are such that the risk of an adversary acquiring both is very low. Typically, the two factors consist of something the user knows (e.g., a password) and something the user possesses (e.g., a hardware token). Physical tokens, however, are inconvenient for users, who may not always carry them, and costly for the service that deploys them.

In 2011 Facebook, in an effort to combat stolen account passwords, introduced its so-called Social Authentication (SA), a second authentication factor based on user-related social information that an adversary "half way around the world" supposedly lacks and cannot easily trick the owners into divulging. Following the standard password-based authentication, if Facebook deems it necessary, users are presented with photos of 7 of their friends and are asked to identify them. SA appears to be more user-friendly and practical as (i) users are required to identify photos of people

[*]Iasonas Polakis and Sotiris Ioannidis are also with the University of Crete

ACSAC '12 Dec 3-7, 2012, Orlando, Florida USA
Copyright 2012 ACM 978-1-4503-1312-4/12/12 ...$15.00.

they know and (ii) they are accustomed to tagging photos of their friends—thus implicitly providing the necessary labeled dataset for Facebook.

In this paper we identify the vulnerable nature of SA and empirically confirm a series of weaknesses that enable an adversary to carry out an effective automated attack against Facebook's SA. The key of SA is the knowledge a user has about his online social circle, whereas an attacker trying to log into the account with stolen credentials, lacks. Facebook acknowledges that its heuristics and threat model do not cover the case of friends and family (i.e., anyone inside a user's online social circle) hacking into one's account. The intuition behind our research is that any stranger who obtains a user's password can gain enough data to defeat the SA mechanism.

To this end, we initially conduct a series of experiments to validate our assumptions about the access that an adversary might have to such information. The core of this paper is the design and implementation of an automated, modular system that defeats Facebook's SA mechanism. The general principles of our approach allow it to be extended and applied to any photo-based SA system. Initially, during a preparatory reconnaissance phase we obtain a victim's list of friends and the photos accessible from his OSN profile. This includes crawling the publicly-accessible portion of the victim's social graph and (optionally) performing actions that bring us inside the restricted part of the social circle, such as issuing friendship requests to the victim's friends. We then process the collected photos using face detection and recognition software to build each friend's facial model. An attacker is highly unlikely to be familiar with the friends of a victim—at least under the threat model assumed by Facebook—and there lies the security of recognizing one's friends as a security mechanism. However, by acquiring accurate facial models of a victim's friends we are in possession of the key to solving SA challenges. When the SA test is triggered, we lookup the identity of the depicted friends and provide an answer.

At a first glance, it might seem that our attack only affects Facebook users that leave their friends list and published photos publicly accessible. According to Dey R. et al. [9] (2012), 47% percent of Facebook users leave their friends list accessible by default. However, an attacker can always attempt to befriend his victims, thus gaining access to their protected information. Such actions may achieve up to a 90% success rate [4, 5, 19, 24]. That way, the set of vulnerable users may reach 84% of the Facebook population. At the same time, our experiments show that 71% of Facebook users expose at least one publicly-accessible photo album. Similarly, an attacker has very good chances of getting access, through online friendship requests, to profiles with private photo albums. Moreover, even if user A's photos are protected from public view and A does not accept friend requests from unknown people, user B might have a photo of A in which A is tagged (i.e., their face framed and labeled with his real name and Facebook ID). If user B has their photos public, A's tags are implicitly exposed to crawling. Overall, dynamics of OSNs such as Facebook, make it very hard for users to control their data [18, 23] and thereby increase the attack surface of threats against SA. We show that anyone can gain access to crucial information for at least 42% of the tagged friends used to build SA challenges that will protect a user's profile.

Under such minimal attack-surface assumptions we manually verify that our implemented SA breaker, powered by a face recognition module, solves 22% of the real SA tests presented by Facebook (28 out of 127 tests), in less than 60 seconds for each test. Moreover, our attack gives a significant advantage to an attacker as it solves 70% of each test (5 out of 7 pages) for 56% of the remainder tests (71 out of 99 tests). Note that we obtain this accuracy in real-world conditions by relying solely on publicly-available information, which anyone can access: We do not send friendship requests to the victims or their friends to gain access to more photos. Furthermore, our simulations demonstrate that within a maximized attack surface (i.e., if a victim, or one of his friends, accepts befriend requests from an attacker, which happens in up to 90% of the cases), the success rate of our attack increases to 100%, with as little as 120 faces per victim for training, and takes about 100 seconds per test.

A recent study [17], provided a formal analysis of the social authentication weaknesses against attacker within the victim's social circle. We expand the threat model and demonstrate in practice that any attacker, inside and outside the victim's social circle, can carry out automated attacks against the SA mechanism in an efficient manner. Therefore we argue that Facebook should reconsider its threat model and re-evaluate this security mechanism.

In summary, the key contributions of this work are the following:

- We systematize and expand previous work, which pointed out (i) the feasibility of recognizing people's faces using Facebook photos, and (ii) the theoretical issues with face-based SA. This systematization allows us to describe an attack that breaks Facebook's SA mechanism, while retaining the assumptions of their threat model.
- We present our black-box security analysis of Facebook's SA mechanism and point out its weaknesses (and implementation flaws) when employed as the second factor of a two-factor authentication scheme.
- We design and implement an automated, modular system that leverages face detection and recognition to break Facebook's SA efficiently. We, thus, show the feasibility of such an attack in large-scale.
- We show that publicly-available face recognition services offer a very accessible and precise alternative to building a custom face recognition system.
- We discuss how Facebook's SA scheme should be modified so that users can trust it as a second authentication factor.

2. SOCIAL AUTHENTICATION

We hereby describe the nature of Facebook's SA in terms of functionality and heuristics. We go beyond a general description and evaluate its behavior under real-world conditions. Facebook's SA was announced in January 2011 and, to the best of our knowledge, is the first instance of an authentication scheme based on the "who you know" rationale: A user's credentials are considered authentic only if the user can correctly identify his friends.

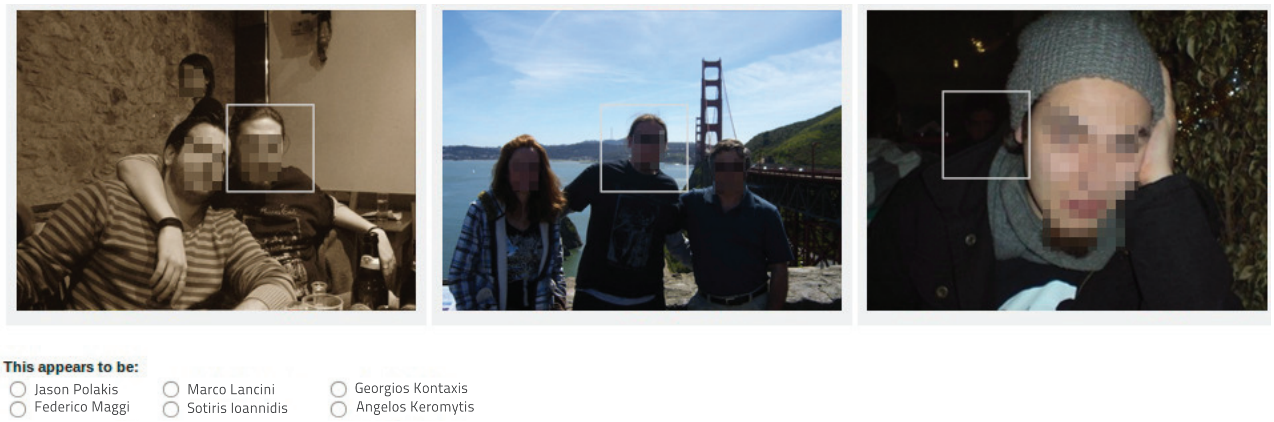
2.1 How Social Authentication Works

After the standard, password-based authentication, the user is presented with a sequence of 7 pages featuring authentication challenges. As shown in Fig. 1, each challenge is comprised of 3 photos of an online friend; the names of 6 people from the user's social circle are listed and he has to select the one depicted. The user is allowed to fail in 2 challenges, or skip them, but must correctly identify the people in at least 5 to pass the SA test.

2.2 Requirements for Triggering

Based on our analysis, Facebook activates the SA only for the fraction of accounts that have enough friends with a sufficient amount of tagged photos that contain a human face.

Friend list. SA requires that the user to be protected has a reasonable number of friends. From our experiments we have concluded that, in the case of Facebook, a user must have at least 50



This appears to be:

○ Jason Polakis ○ Marco Lancini ○ Georgios Kontaxis
○ Federico Maggi ○ Sotiris Ioannidis ○ Angelos Keromytis

Figure 1: Example screenshot of the user interface of a Facebook SA page. The screenshot is synthetic due to copyright reasons, but is an exact replica of a real-world Facebook SA page. Faces have been pixelated for privacy reasons.

friends. To obtain this information, we created 11 distinct dummy profiles and increased the number of friends of these accounts on a daily basis, until we managed to trigger the SA (detailed in §4.3).

Tagged photos. The user's friend must be tagged (placed in a labeled frame) in an adequate number of photos. Keep in mind that since these are user-submitted tags, Facebook's dataset can get easily tainted. People often erroneously tag funny objects as their friends or publish photos with many friends tagged, several of whom may not actually be present in the photo.

Faces. SA tests must be solvable by humans within the 5 minute (circa) time window enforced by Facebook. We argue that Facebook employs a face detection algorithm to filter the dataset of tagged people to select photos with tagged faces. From our manual inspection of 127 instances of real SA tests (2,667 photos), we have noticed that Facebook's selection process is quite precise, despite some inaccuracies that lead to SA tests where some photos contain no face. Overall, 84% of these 2,667 photos contained at least one human-recognizable face, and about 80% of them contained at least one face such that an advanced face detection software can discern—in this test, we used `face.com`. To validate our argument on the use of face detection filtering, we repeated the same manual inspection on a different set of 3,486 photos drawn at random from our dataset of 16,141,426 photos (detailed in §4.1). We then cropped these images around the tags; hence, we generated a SA dataset in the same manner that Facebook would if it naively relied only on people's tagging activity. Only 69% ($< 84\%$) of these photos contain at least one recognizable human face, thus the baseline number of faces per tag is lower in general than in the photos found in the real SA tests. This confirms our hypothesis that Facebook employs filtering procedures to make sure each SA test page shows the face of the person in question in at least one photo.

Triggering. Facebook triggers the SA when it detects a suspicious login attempt, according to a set of heuristics. Our experiments reveal that this happens when (i) the user logs in from a different geographical location, or (ii) uses a new device (e.g., computer or smartphone) for the first time to access his account.

2.3 Advantages and Shortcomings

The major difference from the traditional two-factor authentication mechanisms (e.g., confirmation codes sent via text message or OTP tokens) is that Facebook's SA is less cumbersome, especially because users have grown accustomed to tagging friends in photos. However, as presented recently by Kim et al. [17], designing a usable yet secure SA scheme is difficult in tightly-connected social graphs, not necessarily small in size, such as university networks.

Our evaluation suggests that SA carries additional implementation drawbacks. First of all, the number of friends can influence the applicability and the usability of SA. In particular, users with many friends may find it difficult to identify them, especially when there are loose or no actual relationships with such friends. A typical case is a celebrity or a public figure. Even normal users, with 190 friends on average[1], might be unable to identify photos of online contacts that they do not interact with regularly. Dunbar's number [11] suggests that humans can maintain a stable social relationship with at most 150 people. This limit indicates a potential obstacle in the usability of the current SA implementation, and should be taken into account in future designs.

Another parameter that influences the usability of SA is the number of photos that depict the actual user, or at least that contain objects that uniquely identify the particular user. As a matter of fact, feedback [15] from users clearly expresses their frustration when challenged by Facebook to identify inanimate objects that they or their friends have erroneously tagged for fun or as part of a contest which required them to do so.

Finally, in certain cases, Facebook currently presents users with the option to bypass the SA test by providing their date of birth. This constitutes a major flaw in their security mechanism. Obtaining the victim's date of birth is trivial for an adversary, as users may reveal this information on their Facebook profile.

2.4 Threat Model and Known Attacks

Throughout this paper we refer to the people inside a user's online social circle as friends. Friends have access to information used by the SA mechanism. Tightly-connected social circles where a user's friends are also friends with each other are the worst scenarios for SA, as potentially any member has enough information to solve the SA for any other user in the circle. However, Facebook designed SA as a protection mechanism against strangers, who have access to none or very little information. Under this threat model, strangers are unlikely to be able to solve an SA test. We argue that any stranger can position himself inside the victim's social circle, thereby gaining the information necessary to defeat the SA mechanism. Kim et al. [17] suggest that the progress made by face-recognition techniques may enable automated attacks against photo-based authentication mechanisms. At the same time, Dantone et al. [8] have demonstrated that social relationships can also be used to improve the accuracy of face recognition. Moreover, Ac-

[1] `https://www.facebook.com/notes/facebook-data-team/anatomy-of-facebook/10150388519243859`

quisti et al. [1] went beyond the previous approach and presented a system that can associate names to faces and, thus, de-anonymize a person solely by using a picture of his or her face. Although no scientific experimentation on real-world data has been made to measure the weakness of SA, these studies suggest that the face-to-name relation, which is the security key behind SA, may be exploited further to demonstrate that the scheme is insecure. Our intuition that attackers can overcome the limitations of Facebook's perceived threat model has been the motivation behind this paper.

2.5 Attack Surface Estimation

In our attack model, the attacker has compromised the user's credentials. This is not an unreasonable assumption; it is actually the reason behind the deployment of the SA. This can be accomplished in many ways (e.g., phishing, trojan horses, key logging, social engineering) depending on the adversary's skills and determination [10]. Statistically speaking, our initial investigation reveals that Facebook's current implementation results in 2 out of 3 photos of each SA page (84% of 3 is 2.523) with at least one face that a human can recognize. This makes SA tests solvable by humans. However, our investigation also reveals that about 80% of the photos found in SA tests contain at least one face that can be detected by face-detection software. This rationale makes us argue that an automated system can successfully pass the SA mechanism. To better understand the impact of our attack, we provide an empirical calculation of the probabilities of each phase of our attack. In other words, if an attacker has obtained the credentials of any Facebook user, what is the probability that he will be able to access the account? What is the probability if he also employs friend requests to access non-public information on profiles? To derive the portion of users susceptible to this threat, we built the attack tree of Fig. 2.

We distinguish between a casual and a determined attacker, where the former leverages publicly-accessible information from a victim's social graph whereas the latter actively attempts to gather additional private information through friendship requests.

Friends list. Initially, any attacker requires access to the victim's friends list. According to Dey et al. [9] $\mathrm{P}(F) = 47\%$ of the user's have their friends list public—as of March 2012. If that is not the case, a determined attacker can try to befriend his victim. Studies have shown [4, 5, 19, 24] that a very large fraction of users tends to accept friend requests and have reported percentages with a 60–90% chance of succeeding (in our analysis we use 70%, lower than what the most recent studies report). Therefore, he has a combined 84% chance of success so far, versus 47% for the casual attacker.

Photos. Ideally the attacker gains access to all the photos of all the friends of a victim. Then with a probability of 1 he can solve any SA test. In reality, he is able to access only a subset of the photos from all or a subset of the friends of a victim. Our study of 236,752 Facebook users revealed that $P(P) = 71\%$ of them exposed at least one public photo album. Again we assume that a determined attacker can try to befriend the friends of his victim to gain access to their private photos with a chance of $P(B) \simeq 70\%$ to succeed, which is a conservative average compared to previous studies. At the end of this step, the determined attacker has on average at least one photo for 77% of the friends of his victim while a casual attacker has that for 33%. This is versus Facebook which has that for 100% of the friends with uploaded photos.

Tags. The next step is to extract labeled frames (tags) of people's faces from the above set of photos to compile $\langle \texttt{uid}, \texttt{face} \rangle$ tuples used by Facebook to generate SA tests and by the attacker to train facial models so as to respond to those tests. By analyzing $16,141,426$ photos from out dataset, corresponding to the 33% of friends' photos for the casual attacker, we found that 17% of these photos contain tags (hence usable for generating SA tests), yet only the 3% contain tags about the owner of the photo. This means that by crawling a profile and accessing its photos it is more likely to get tags of friends of that profile than of that profile itself. The astute reader notices that Facebook also has to focus on that 17% of photos containing tags to generate SA tests: Facebook will utilize the 17% containing tags of all the photos uploaded by a user's friends and therefore generate SA tests based on 100% of the friends for whom tags are available, whereas an attacker usually has access to less than that. In the extreme case, having access to a single friend who has tagged photos of all the other friends of the target user (e.g., he is the "photographer" of the group), the attacker will acquire at least one tag of each friend of the user and will be able to train a face recognition system for 100% of the subjects that might appear in an SA test. In practice, by collecting the tags from the photos in our dataset we were able to gather $\langle \texttt{uid}, \texttt{face} \rangle$ tuples for 42% of the people in the friend lists of the respective users. Therefore, assuming that all of a user's friends have tagged photos of them on Facebook, a casual attacker is able to acquire this sensitive information for 42% of the tagged friends used by Facebook to generate SA tests. As we show in §4.3, with only that amount of data, we manage to automatically solve 22% of the real SA tests presented to us by Facebook, and gain a significant advantage for an additional 56% with answers to more than half the parts of each test. We cannot calculate the corresponding percentage for the determined attacker without crawling private photos (we discuss the ethical reasons for this in §5). However, we simulate this scenario in §4.2 and find that we are able to pass the SA tests on average with as little as 10 faces per friend.

Faces. Finally, from the tagged photos, the attacker has to keep the photos that actually feature a human face and discard the rest—we can safely hypothesize Facebook does the same, as discussed in §2.2. We found that 80% of the tagged photos in our dataset contain human faces that can be detected by face-detection software, and Facebook seems to follow the same practice; therefore, the advantage for either side is equal. Overall, our initial investigation reveals that up to 84% of Facebook users are exposed to the crawling of their friends and their photos. They are, thus, exposed to attacks against the information used to protect them through the SA mechanism. A casual attacker can access $\langle \texttt{uid}, \texttt{face} \rangle$ tuples of at least 42% of the tagged friends used to generate social authentication tests for a given user. Such information is considered sensitive, known only to the user and the user's circle, and its secrecy provides the strength to this mechanism.

3. BREAKING SOCIAL AUTHENTICATION

Our approach applies to any photo-based SA mechanism and can be extended to cover other types of SA that rely on the proof of knowledge of "raw" information (e.g., biographies, activities, relationships and other information from the profiles of one's social circle). We focus on Facebook's SA, as it is the only widespread and publicly-available deployment of this type of social authentication. As detailed in §3.1, our attack consists of three preparation steps (steps 1-3), which the attacker runs offline, and one execution step (step 4), which the attacker runs in real-time when presented with the SA test. Fig. 3 presents an overview of our system's design.

3.1 Implementation Details

3.1.1 Step 1: Crawling Friend List

Given the victim's UID, a crawler module retrieves the UIDs and names of the victim's friends and inserts them in our database. As discussed in §2.5, casual attackers can access the friend list when this is publicly available (47% of the users), whereas determined attackers can reach about 84% of the friend lists by issuing befriend

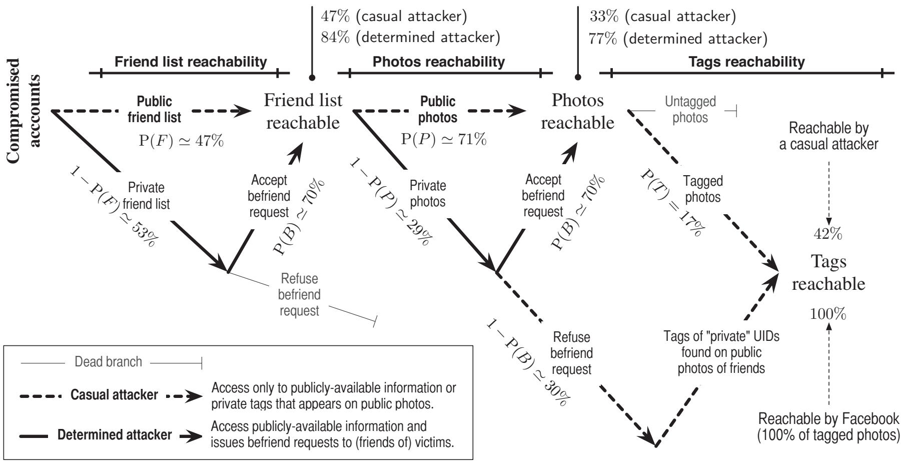


Figure 2: Attack tree to estimate the vulnerable Facebook population. Not all the branches are complete, as we consider only the events that are relevant to the case study.

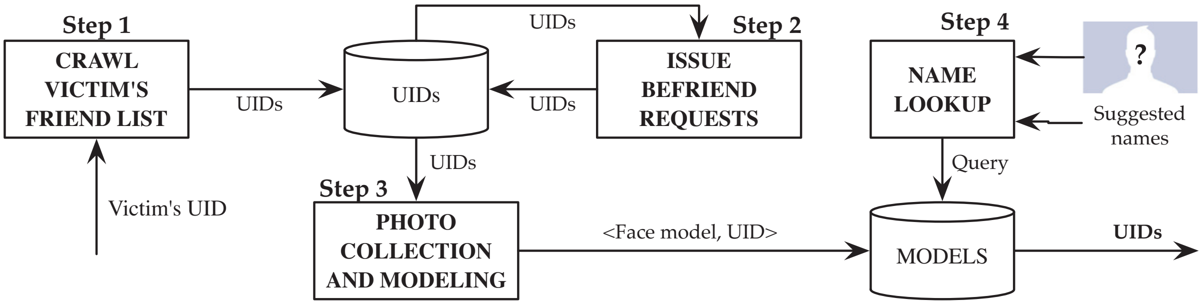


Figure 3: Overview of our automated SA-breaking system. It operates in four steps. In Step 1 we retrieve the victim's friend list using his or her UID. Then, in Step 2 (optional), we send befriend requests, so that we have more photos to extract faces from and build face classifiers in Step 3. In Step 4, given a photo, we query the models to retrieve the corresponding UID and thus match a name to face.

requests. We implement the crawling procedures using Python's `urllib` HTTP library and regular expression matching to scrape Facebook pages and extract content. We store the retrieved data in a MongoDB database, a lightweight, distributed document-oriented storage suitable for large data collections, and keep the downloaded photos in its GridFS filesystem.

3.1.2 Step 2: Issuing Friend Requests

An attacker can use legitimate-looking, dummy profiles to send friendship requests to all of the victim's friends. As shown in Fig. 2, this step can expand the attack surface by increasing the reachable photos. We implement a procedure that issues befriend requests via the fake accounts we have created for our experimental evaluation (see §4.1). Even though we do not collect any private information or photos of these users for our experiments, we need an adequate number of friends in our accounts to be able to trigger the SA mechanism. We select users for our requests, based on the friends suggested by Facebook. Also, as shown by Irani et al. [14], to achieve a high ratio of accepted friend requests, we create profiles of attractive women and men with legitimate-looking photos[2] (i.e., avoiding the use of provocative or nudity photos). In addition, we inject some random profile activity (e.g., status messages, like activities). If Facebook triggers CAPTCHA challenges at some point, our system prompts a human operator to intervene. However, Bilge et al. [4] have demonstrated the use of automated systems against the CAPTCHA countermeasure. Moreover, to hinder spammers, Facebook limits the number of friend requests each profile is allowed to issue in a short period of time and enforces a "cooldown" period of two days on misbehavior. To overcome this obstacle and still have profiles with an adequate amount of friends, we spread our friend requests over a period of one week. We also noticed that for profiles that have education and employment information and send requests to people within these circles, Facebook enforces more relaxed thresholds and allowed us to send close to 100 requests in a single day. In addition, the method described by Irani et al. [14] allows to increase the number of friends passively as opposed to requesting friendships explicitly.

3.1.3 Step 3: Photo Collection/Modeling

Step 3.1: Photo collection We collect the URLs of all the photos contained in the albums of the target's friends using the same screen-scraping approach that we described in Step 3.1.1. We then feed the collected URLs into a simple module that does the actual download. This module stores in the database the metadata associated with each downloaded photo: URL, UID of the owner, tags and their coordinates (in pixels).

[2] We selected photos from a database of models.

Step 3.2: Face Extraction and Tag Matching We scan each downloaded photo to find faces. Specifically, we use a face detection classifier part of the OpenCV toolkit[3]. There are plenty of face detection techniques available in the literature, more precise than the one that we decided to use. However, our goal is to show that face-based SA offers only a weak protection, because even with simple, off-the-shelf tools, an adversary can implement an automated attack that bypasses it.

Subsequently, we label each face with the UID of the nearest tag found in the adjacent 5%-radius area, calculated with the euclidean distance between the face's center and the tag's center. Unlabeled faces and tags with no face are useless, thus we discard them. We save the selected faces as grayscale images, one per face, resized to 130 × 130 pixels.

Step 3.3: Facial Modeling We use the `sklearn` library[4] to construct a supervised classifier. We first preprocess each face via histogram equalization to ensure uniform contrast across all the samples. To make the classification job feasible with these many features (i.e., 130 × 130 matrices of integers), we project each matrix on the space defined by the 150 principal components (i.e., the "eigenfaces"). We tested K-nearest-neighbors (kNN), tree, and support-vector (with a radial-basis kernel) classifiers using a K-fold cross-validation technique. We found that support-vector classifiers (SVC) yield the highest accuracy, but are very expensive computationally. Therefore, we use kNN classifiers, with $k = 3$ as they provide a faster alternative to SVC with comparable accuracy.

3.1.4 *Step 4: Name Lookup*

When Facebook challenges our system with a SA test, we submit the photos from the SA test to the classifier, which attempts to identify the depicted person and select the correct name. We detect the faces in each of the 7 photos of an SA page and extract the 150 principal components from each face's 130 × 130 matrix. Then, we use the classifier to predict the class (i.e., the UID) corresponding to each unknown face, if any. If, as in the case of Facebook, a list of suggested names (i.e., UIDs) is available, we narrow its scope to these names. Then, we query the classifier and select the outcome as the correct UID for each unknown face, choosing the UID that exhibits more consensus (i.e., more classifiers output that UID) or the highest average prediction confidence.

3.2 Face Recognition as a Service

Automatic face recognition is approaching the point of being ubiquitous: Web sites require it and users expect it. Therefore, we investigate whether we can employ advanced face recognition software offered as a cloud service. We select `face.com` which offers a face recognition platform for developers to build their applications on top of. Incidentally, `face.com` was recently acquired by Facebook[5]. The service exposes an API through which developers can supply a set of photos to use as training data and then query the service with a new unknown photo for the recognition of known individuals. The training data remains in the cloud. Developers can use up to two different namespaces (i.e., separate sets of training data) each one able to hold up to 1,000 users, where each user may be trained with a seemingly unbound number of photos. Usage of the API is limited to 5,000 requests an hour. Such a usage framework may be restrictive for building popular applications with thousands of users but it is more than fitting for the tasks of an adversary seeking to defeat photo-based social authentication. Assuming the free registration to the service, one may create a training set for up to 1,000 of a victim's friends (the max limit for Facebook is 5,000 although the average user has 190 friends). After that, one can register more free accounts or simply delete the training set when no longer necessary and reclaim the namespace for a new one. We develop a proof-of-concept module for our system that leverages the `face.com` API as an alternative, service-based implementation of steps 3 and 4 from Fig. 3. We submit the photos to the service via the `faces.detect` API call to identify any existing faces and determine whether they are good candidates for training the classifier. The next step is to label the good photos with the respective UIDs of their owners (`tags.save`). Finally we initiate the training on the provided data (`faces.train`) and once the process is complete we can begin our face recognition queries—the equivalent of step 4 from Fig. 3. Once the training phase is finished, the service is able to respond within a couple of seconds with a positive or negative face recognition decision through the `faces.recognize` call. We take advantage of the ability to limit the face matching to a group of uids from the training set and we do so for the suggested names provided by Facebook for each SA page.

	Total	Public	Private
UIDs	236,752	167,359	69,393
Not tagged	116,164	73,003	43,161
Tagged	120,588	94,356	26,232
Mean tags per UID:		19.39	10.58
Tags[9]	2,107,032	1,829,485	277,547
Photos	16,141,426	16,141,426	(not collected)
Albums	805,930	805,930	(not collected)

Table 1: Summary of our collected dataset. The terms "public", and "private" are defined in §4.1.

4. EXPERIMENTAL EVALUATION

Here we evaluate the nature of Facebook's SA mechanism and our efforts to build an automated SA solving system. We first assess the quality of our dataset of Facebook users (§4.1). We consider this a representative sample of the population of the online social network. We have not attempted to compromise or otherwise damage the users or their accounts. We collected our dataset as a casual attacker would do. Next we evaluate the accuracy and efficiency of our attack. In §4.2 we use simulation to play the role of a determined attacker, who has access to the majority of the victims' photos. In §4.3 we relax this assumption and test our attack as a casual attacker, who may lack some information (e.g., the victims may expose no photos to the public, there are no usable photos, no friend requests issued). More details on the capabilities of these two types of attacker are given in §2.5.

For part of our experiments we implemented custom face recognition software. This was done for two reasons. First, because we needed something very flexible to use, that allowed us to perform as many offline experiments as needed for the experiments of the determined attacker. Second, we wanted to show that even off-the-shelf algorithms were enough to break the SA test, at least in ideal conditions. However, superior recognition algorithms exist, and we conducted exploratory experiments that showed that face.com, although less flexible than our custom solution, has much better accuracy. Therefore, we decided to use it in the most challenging conditions, that is to break SA tests under the hypothesis of the casual attacker.

[3] `http://opencv.itseez.com/`

[4] `http://scikit-learn.org`

[5] `http://face.com/blog/facebook-acquires-face-com/`

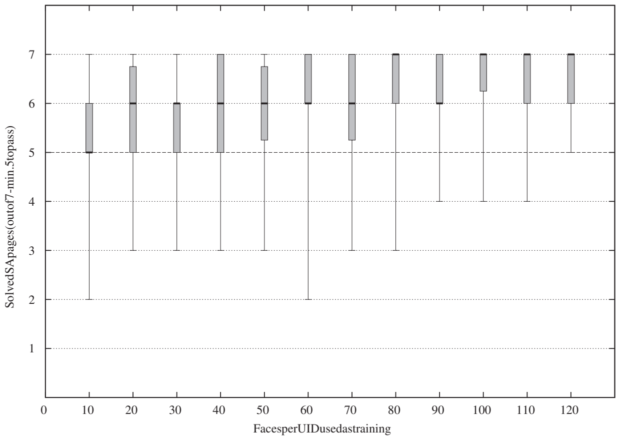


Figure 4: Percentage of successfully-passed tests as a function of the size of the training set. For each iteration, 30 randomly-generated offline SA tests were used.

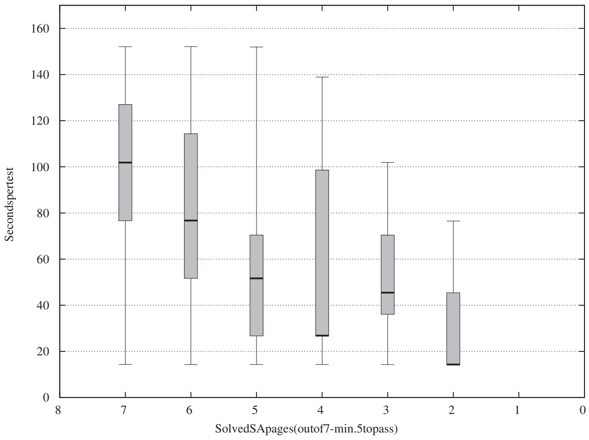


Figure 5: Time required to lookup photos from SA tests in the face recognition system.

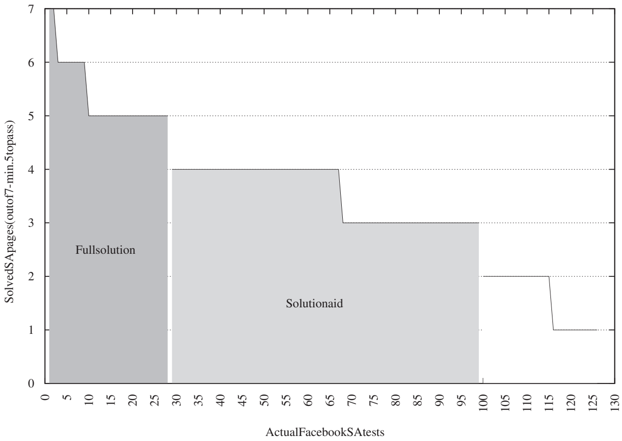


Figure 6: Efficiency of automated SA breaker against actual Facebook tests.

4.1 Overall Dataset

Our dataset contains data about real Facebook users, including their UIDs, photos, tags, and friendship relationships, as summarized in Table 1. Through public crawling we collected data regarding 236,752 distinct Facebook users. 71% (167,359) of them have at least one publicly-accessible album. We refer to these users as public UIDs (or public users). The remaining 29% of UIDS (69,393) keep their albums private (i.e., private UIDs, or private users). We found that 38% of them (26,232 or 11% of the total users) are still reachable because their friends have tagged them in one of the photos in their own profile (to which we have access). We refer to these UIDs as semi-public UIDs (or semi-public users). Data about the remaining 62% of UIDs (43,161 or 18% of the total users) is not obtainable because these users keep their albums private, and their faces are not found in any of the public photos of their friends. The public UIDs lead us to 805,930 public albums, totaling 16,141,426 photos and 2,107,032 tags that point to 1,877,726 distinct UIDs. It is therefore evident that people exposing (or making otherwise available) their photos are not only revealing information about themselves but also about their friends. This presents a subtle threat against these friends who cannot control the leakage of their names and faces. Albeit this dataset only covers a very small portion of the immense Facebook user base, we consider it adequate enough to carry out thorough evaluation experiments.

4.2 Breaking SA: Determined Attacker

The following experiment provides insight concerning the number of faces per user needed to train a classifier to successfully solve the SA tests. We create simulated SA tests using the following methodology. We train our system using a training set of $K = 10, 20, \ldots, 120$ faces per UID. We extract the faces automatically, without manual intervention, using face detection as described in §3.1.3. We then generate 30 SA tests. Each test contains 3 target photos per 7 pages showing the face of the same victim. The photos are selected randomly from the pool of public photos we have for each person, from which we exclude the ones used for the training. For each page and K we record the output of the name-lookup step (step 4), that is the prediction of the classifier as described in §3.1.4, and the CPU-time required. Fig. 4 shows the number of pages solved correctly out of 7, and Fig. 5 shows the CPU-time required to solve the full test (7 pages).

For an SA test to be solved successfully, Facebook requires that 5 out of 7 challenges are solved correctly. Our results show that our attack is always successful (i.e., at least 5 pages solved over 7) on average, even when a scarce number of faces is available. Clearly, having an ample training dataset such as $K > 100$ ensures a more robust outcome (i.e., 7 pages solved over 7). Thus, our attack is very accurate. As summarized in Fig. 5, our attack is also efficient because the time required for both "on the fly" training—on the K faces of the 6 suggested users—and testing remains within the 5-minute timeout imposed by Facebook to solve a SA test. An attacker may choose to implement the training phase offline using faces of all the victim's friends. This choice would be mandatory if Facebook—or any other Web site employing SA—decided to increase the number of suggested names, or remove them completely, such that "on the fly" training becomes too expensive.

4.3 Breaking SA: Casual Attacker

In the following experiment we assume the role of a casual attacker, with significantly more limited access to tag data for the training of a face recognition system. At the same time we attempt to solve real Facebook SA tests using the following methodology.

[9] On 11 April 2012, our crawler had collected 2,107,032 of such tags, although the crawler's queue contains 7,714,548 distinct tags.

We have created 11 dummy accounts that play the role of victims and populate them with actual Facebook users as friends and activity. Then, we employ a graphical Web browser scripted via Selenium[6] to log into these accounts in an automated fashion. To trigger the SA mechanism we employ Tor[7] which allows us to take advantage of the geographic dispersion of its exit nodes, thus appearing to be logging in from remote location in a very short time. By periodically selecting a different exit node, as well as modifying our user-agent identifier, we can arbitrarily trigger the SA mechanism. Once we are presented with an SA test, we iterate its pages and download the presented photos and suggested names, essentially taking a snapshot of the test for our experiments. We are then able to take the same test offline as many times necessary. Note that this is done for evaluation purposes and that the same system in production would take the test once and online. Overall, we collected 127 distinct SA tests.

We tried breaking the real SA tests using our module for `face.com` described in §3.2. Fig. 6 presents the outcome of the tests. Overall we are able to solve 22% of the tests (28/127) with people recognized in 5–7 of the 7 test pages and significantly improve the power of an attacker for 56% of the tests (71/127) where people were recognized in 3–4 of the 7 test pages. At the same time, it took 44 seconds on average with a standard deviation of 4 seconds to process the photos for a complete test (21 photos). Note that the time allowed by Facebook is 300 seconds.

We further analyzed the photos from the pages of the SA tests that failed to produce any recognized individual. In about 25% of the photos `face.com` was unable to detect a human face. We manually inspected these photos and confirmed that either a human was shown without his face being clearly visible or no human was present at all. We argue that humans will also have a hard time recognizing these individuals unless they are very close to them so that they can identify them by their clothes, posture or the event. Moreover, in 50% of the photos `face.com` was able to detect a human face but marked it as unrecognizable. This indicates that it is either a poor quality photo (e.g., low light conditions, blurred) or the subject is wearing sunglasses or is turned away from the camera. Finally, in the last 25% of the photos a face was detected but did not match any of the faces in our training set.

Overall, the accuracy of our automated SA breaker significantly aids an attacker in possession of a victim's password. A total stranger, the threat assumed by Facebook, would have to guess the correct individual for at least 5 of the 7 pages with 6 options per page to choose from. Therefore, the probability [8] of successfully solving an SA test with no other information is $O(10^{-4})$, assuming photos of the same user do not appear in different pages during the test. At the same time, we have managed to solve SA tests without guessing, using our system, in more than 22% of the tests and reduce the need to guess to only 1–2 (of the 5) pages for 56% of the tests, thus having a probability of $O(10^{-1})$ to $O(10^{-2})$ to solve those SA tests correctly. Overall in 78% of the real social authentication tests presented by Facebook we managed to either defeat the tests or offer a significant advantage in solving them.

After these experiments, we deleted all the photos collected from the real SA tests, as they could potentially belong to private albums of our accounts' friends, not publicly accessible otherwise.

5. ETHICAL CONSIDERATIONS

In this paper we explore the feasibility of automated attacks against the SA mechanism deployed by Facebook. As our experiments involve actual users, the question of whether this is ethically justifiable arises. We believe that research that involves the systematic exploration of real attacks is crucial, as it can reveal weaknesses and vulnerabilities in deployed systems, and provide valuable insight that can lead better solutions. This opinion is also shared among other security researchers [5, 16].

Nonetheless, we designed our experiments such that we minimize the impact of our research and preserve the privacy of these users. First, we never retained verbatim copies of sensitive information, besides the photos that we clearly needed for running the experiments. Secondly, our attack can optionally issue friend requests with the purpose of expanding the number of accessible photos. However, we issued friendship requests exclusively to reach the 50-friends threshold, required by Facebook to trigger the SA mechanism. We never took advantage of accepted requests to collect photos or other private information otherwise unavailable; we solely collected public photos. In particular, in §4.2 we simulated a determined attacker, by assuming he has obtained access to all the photos (both public and private) needed to launch the attacker under ideal conditions. We simulated these conditions using publicly-available photos.

6. REMEDIATION AND LIMITATIONS

Facebook has already devised some mechanisms that aim at hindering casual attackers and the practices presented in this paper. We explain why these mechanisms are not very effective or have some drawbacks that make them impractical. We continue with some proposed modifications to SA to make it safer based on the insights we obtained through our experiments.

6.1 Compromise Prevention and Notification

Facebook has recently deployed some security features that can help further defend against stolen credentials being used for compromising accounts. However, these mechanisms are opt-in and disabled by default. Therefore, users may not have them enabled, and will remain susceptible to the threat that we study in this paper.

First, users can add certain devices to a list of recognized, trusted devices. Whenever a user logs in from an unrecognized device, a security token is sent to the owner's mobile phone. This token must be entered in the log-in form for the user to be successfully logged in. This security setting, called login approval, follows the traditional second-token authentication scheme and only works in combination with the recognized, trusted devices feature. This approach can completely deter our attack, because it implements a truly-strong, two-factor authentication: The adversary would need physical access to the user's mobile phone to obtain the security token and successfully login.

Second, a user who fails to complete an SA challenge is redirected to an alert page, upon the next successful login, which reports the attempted login, and shows the time and place information. Unfortunately, if the adversary manages to solve the SA test in a subsequent attempt, he will be redirected to the notification page and the account owner will never see the alert. In addition to the default notification, users may enable an optional login-notification feature: Whenever their account is accessed, an alert message is sent via text or email message. This notification feature does not prevent an adversary from logging in and, therefore, does not prevent our attack, which takes less than one minute. Furthermore, if the adversary has compromised the email account—which is not an unrealistic assumption, as users may reuse their credentials across

[6] `http://seleniumhq.org`

[7] `http://www.torproject.org`

[8] Calculated using the binomial probability formula used to find probabilities for a series of Bernoulli trials.

services—he can delete the notification email. If that is not the case, the adversary will still have access until the owner changes the password and terminates any illegal active sessions.

Moreover, these mechanisms present three additional drawbacks. First, users must link their mobile phone number to their Facebook account, which many may not wish to do. Second, and more importantly, users typically access their account from many devices some of which may be public (e.g., computers in libraries or their workplace). In this case, adding all these devices to the list of trusted devices is both impractical and insecure, and users will not wish to receive alerts every time they log in from one of those machines. Finally, involving the cellular network may result in monetary charges, a factor which could seriously discourage users from opting in to the mechanism.

6.2 Slowing Down the Attacker

When the attacker is prompted with an SA challenge, he must solve a CAPTCHA before the actual SA test. Although this topic falls outside the scope of this paper, it is worth noticing that solving a CAPTCHA is trivial and only takes a human a few seconds. In addition, as previous work [4, 5, 7] has shown, breaking CAPTCHAs automatically is feasible and, in many cases, easy. Furthermore, it is well known that adversaries can perform laundry attacks [2, 13] and crowd-source the solution of CAPTCHAs. In conclusion, CAPTCHAs may create a technical obstacle to automated attacks, but they should not be considered a definitive countermeasure.

The presence of suggested names in SA tests is the major disadvantage of the current implementation as it greatly limits the search space for adversaries. By removing suggestions, there is a high probability of face-recognition software returning multiple users with similar confidence scores. Also, the time needed for face recognition might increase for certain systems although, as we have shown, cloud-based face recognition systems are unlikely to be seriously affected. On the downside, it will be harder for users to identify their friends and the system will be less usable as one would have to manually type the exact names of his friends. Automatic “type ahead” features may lessen the burden, although they are still vulnerable to exhaustive enumeration.

6.3 SA revisited

Designing effective and usable CAPTCHAs [6] is as hard as designing effective and usable authentication schemes that exploit social knowledge [17]. The downside of CAPTCHAs is that they are either too easy for machines or too difficult for humans. This paper and previous work show that the main weakness of social-based authentication schemes is that the knowledge needed to solve them is too public: Ironically, the purpose of social networks and the nature of human beings is to share knowledge. However, we believe that SA tests could be more secure yet still solvable by humans.

Facebook can build SA tests from photos showing human faces that fail or achieve very low confidence scores in Facebook’s own face recognition system. Photos may contain faces of people wearing glasses, with masks on or slightly turned away from the camera. Humans are able to recognize their friends in the general image of a person, the environment, or the event. On the other hand, face-recognition algorithms have a hard time matching these human and social characteristics across different photos. In terms of feasibility, Facebook can piggy-back on users uploading photos as part of their daily routine and prompt them to tag a person for which no face has been detected, thus creating the necessary labeled dataset for generating SA tests. Even if an adversary is able to capture that photo and the tag provided by the user, chances are his face recognition algorithm will fail to find a resemblance with other photos of the same person. Also, if the adversary carries out an informed training process this might introduce unwanted noise which will reduce the overall accuracy of the classifier.

7. RELATED WORK

Previous work showed that information available in users’ profiles in social networks can be used to break authentication mechanisms, or deduce information that threatens their privacy. A study performed by Rabkin [21] attempted to assess the security properties of personal knowledge questions that are used for fallback authentication. In §2.5 we discuss a similar study, although focused on Facebook SA. Rabkin argues that since such mechanisms owe their strength to the hardness of an information-retrieval problem, in the era of online social networks and the vast availability of personal information, their security is diminishing. In this study 12% of the sampled security questions from online banking sites is automatically attackable (i.e., the answers are on a user’s profile).

Polakis et al. [20] demonstrate and evaluate how names extracted from OSNs can be used to harvest e-mail addresses as a first step for personalized phishing campaigns. Using information from Facebook, Twitter and Google Buzz, over 40% of the users’ e-mail addresses could be directly mapped to their social profiles. A similar study was conducted by Balduzzi et al. [3]. They focus on the ability to use search utilities in social networking sites as an oracle; an attacker can search for an e-mail address, and if a user profile has been registered with that address, the profile is returned. Thus, the attacker can map e-mail addresses to social profiles.

The work most related to this paper is a recent study by Kim et al. [17], already discussed in §2.4. They formally quantify the advantage an attacker has against SA tests when he is already inside the victim’s social circle. The researchers thus demonstrate that SA is ineffective against one’s close friends and family or highly connected social sub-networks such as universities. However, in this paper we extend the threat model to incorporate any attacker located outside the victim’s social circle. Furthermore, we implement a proof-of-concept infrastructure, and use publicly available information to quantify the effectiveness of such attacks. Thus, we are able to show the true extent to which SA is susceptible to automated attacks. Previous work [4, 5, 19, 24] has proved the feasibility of positioning one’s self among a target’s social circle using a mix of active and passive [14] techniques ranging from social engineering (e.g., attractive fake profiles) to forgetful users accepting friendship requests from fake profiles of people they are already linked. As such, the proposed countermeasures by Kim et al. [17] for a more secure social authentication mechanism remain equally vulnerable to our attack. Finally, we also present a theoretical estimation of the attack surface based on empirical data from our experiments as well as those reported by previous studies.

Boshmaf et al. [5] explore the feasibility of socialbots infiltrating social networks, and operate a Socialbot Network in Facebook for 8 weeks. A core aspect of their operation is the creation of new accounts that follow a stealthy behavior and try to imitate human users. These actions are complimentary to our attack, as a determined attacker can use them to infiltrate the social circles of his victims and expand the attack surface by gaining access to private photos. This will result in much higher percentages of solved SA tests. Gao et al. [12] found that 97% of malicious accounts were compromised accounts of legitimate users. This reflects the importance of social authentication as a mechanism for preventing attackers from taking over user accounts using stolen credentials. Accordingly, in this paper we explore the feasibility of an automated attack that breaks SA tests though the use of face recognition techniques. Our results validate the effectiveness of our attack even when the attacker uses only publicly available information.

8. CONCLUSIONS

In this paper we pointed out the security weaknesses of using social authentication as part of a two-factor authentication scheme, focusing on Facebook's deployment. We found that if an attacker manages to acquire the first factor (password), he can access, on average, 42% of the data used to generate the second factor, thus, gaining the ability to identify randomly selected photos of the victim's friends. Given that information, we managed to solve 22% of the real Facebook SA tests presented to us during our experiments and gain a significant advantage to an additional 56% of the tests with answers for more than half of pages of each test. We have designed an automated social authentication breaking system, to demonstrate the feasibility of carrying out large-scale attacks against social authentication with minimal effort on behalf of an attacker. Our experimental evaluation has shown that widely available face recognition software and services can be effectively utilized to break social authentication tests with high accuracy. Overall we argue that Facebook should reconsider its threat model and re-evaluate the security measures taken against it.

Acknowledgements

We thank the anonymous reviewers for their valuable comments and Alessandro Frossi for his support. This paper was supported in part by the FP7 project SysSec funded by the EU Commission under grant agreement no 257007, the Marie Curie Reintegration Grant project PASS, the ForToo Project of the Directorate General Home Affairs, and ONR MURI N00014-07-1-0907. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of ONR or the US Government.

9. REFERENCES

[1] A. Acquisti, R. Gross, and F. Stutzman. Faces of Facebook: How the largest real ID database in the world came to be. BlackHat USA, 2011, `http://www.heinz.cmu.edu/~acquisti/face-recognition-study-FAQ/acquisti-faces-BLACKHAT-draft.pdf`.

[2] E. Athanasopoulos and S. Antonatos. Enhanced CAPTCHAs: Using animation to tell humans and computers apart. In *Proceedings of the 10th IFIP Open Conference on Communications and Multimedia Security*. Springer, 2006.

[3] M. Balduzzi, C. Platzer, T. Holz, E. Kirda, D. Balzarotti, and C. Kruegel. Abusing social networks for automated user profiling. In *Proceedings of the 13th International Conference on Recent Advances in Intrusion Detection*. Springer, 2010.

[4] L. Bilge, T. Strufe, D. Balzarotti, and E. Kirda. All your contacts are belong to us: automated identity theft attacks on social networks. In *Proceedings of the 18th International Conference on World Wide Web*. ACM, 2009.

[5] Y. Boshmaf, I. Muslukhov, K. Beznosov, and M. Ripeanu. The socialbot network: when bots socialize for fame and money. In *Proceedings of the Annual Computer Security Applications Conference*. ACM, 2011.

[6] E. Bursztein, S. Bethard, C. Fabry, J. C. Mitchell, and D. Jurafsky. How good are humans at solving CAPTCHAs? A large scale evaluation. In *Proceedings of the 2010 IEEE Symposium on Security and Privacy*. IEEE, 2010.

[7] E. Bursztein, M. Martin, and J. C. Mitchell. Text-based CAPTCHA strengths and weaknesses. In *Proceedings of the 18th ACM Conference on Computer and Communications Security*. ACM, 2011.

[8] M. Dantone, L. Bossard, T. Quack, and L. V. Gool. Augmented faces. In *Proceedings of the 13th IEEE International Workshop on Mobile Vision*. IEEE, 2011.

[9] R. Dey, Z. Jelveh, and K. Ross. Facebook users have become much more private: A large-scale study. In *Proceedings of the 4th IEEE International Workshop on Security and Social Networking*. IEEE, 2012.

[10] R. Dhamija, J. D. Tygar, and M. Hearst. Why phishing works. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*. ACM, 2006.

[11] R. Dunbar. *Grooming, Gossip, and the Evolution of Language*. Harvard University Press, 1998.

[12] H. Gao, J. Hu, C. Wilson, Z. Li, Y. Chen, and B. Y. Zhao. Detecting and characterizing social spam campaigns. In *Proceedings of the 10th Annual Conference on Internet Measurement*. ACM, 2010.

[13] C. Herley. The plight of the targeted attacker in a world of scale. In *Proceedings of the Ninth Workshop on the Economics of Information Security*, 2010.

[14] D. Irani, M. Balduzzi, D. Balzarotti, E. Kirda, and C. Pu. Reverse social engineering attacks in online social networks. In *Proceedings of the 8th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*. Springer, 2011.

[15] D. Jacoby. Facebook Security Phishing Attack In The Wild. Retrieved on January 2012 from `http://www.securelist.com/en/blog/208193325/Facebook_Security_Phishing_Attack_In_The_Wild`.

[16] M. Jakobsson and J. Ratkiewicz. Designing ethical phishing experiments: A study of (ROT13) rOnl query features. In *Proceedings of the 15th International Conference on World Wide Web*. ACM, 2006.

[17] H. Kim, J. Tang, and R. Anderson. Social authentication: harder than it looks. In *Proceedings of the 2012 Cryptography and Data Security conference*. Springer, 2012.

[18] M. Madejski, M. Johnson, and S. M. Bellovin. A study of privacy settings errors in an online social network. In *Proceedings of the 4th IEEE International Workshop on Security and Social Networking*. IEEE, 2012.

[19] F. Nagle and L. Singh. Can friends be trusted? Exploring privacy in online social networks. In *Proceedings of the 2009 International Conference on Advances in Social Network Analysis and Mining*. IEEE, 2009.

[20] I. Polakis, G. Kontaxis, S. Antonatos, E. Gessiou, T. Petsas, and E. P. Markatos. Using social networks to harvest email addresses. In *Proceedings of the 9th Annual ACM Workshop on Privacy in the Electronic Society*. ACM, 2010.

[21] A. Rabkin. Personal knowledge questions for fallback authentication: Security questions in the era of Facebook. In *Proceedings of the 4th Symposium on Usable Privacy and Security*. ACM, 2008.

[22] A. Shulman. The underground credentials market. *Computer Fraud & Security*, (3), 2010.

[23] J. Staddon and A. Swerdlow. Public vs. publicized: content use trends and privacy expectations. In *Proceedings of the 6th USENIX Conference on Hot Topics in Security*. USENIX, 2011.

[24] B. E. Ur and V. Ganapathy. Evaluating attack amplification in online social networks. In *Proceedings of the 2009 Web 2.0 Security and Privacy Workshop*.