2014 IEEE 34th International Conference on Distributed Computing Systems Workshops

Secure Fallback Authentication and the Trusted Friend Attack

Ashar Javed, David Bletgen, Florian Kohlar, Markus Dürmuth and Jörg Schwenk
Horst Görtz Institut for IT Security
Ruhr-University Bochum
Bochum, Germany
{ashar.javed|david.bletgen|florian.kohlar|markus.duermuth|joerg.schwenk}@rub.de

***Abstract*—Fallback authentication, i.e., recovering access to an account after the password is lost, is an important aspect of real-world deployment of authentication solutions. However, most proposed and deployed mechanisms have substantial weaknesses that seriously degrade security and/or usability. A promising new fallback authentication mechanism is social authentication, which bases authentication on information about the social context of the user (e.g., on his social graph). We consider fallback authentication mechanisms deployed in practice on a number of social network sites (we concentrate on social networks because those can realistically implement social authentication). Our main contribution is a novel attack against Facebook's social authentication mechanism called *Trusted Friends*, which is the prime example for social authentication. Our attack is different from previous attacks in that it does not exploit bias in user choice but exploits tests that are realized client-side (but should be server-side) and POST-data fields that can be manipulated by an attacker. Furthermore, we found problems with all fallback authentication mechanisms used by social network sites, and demonstrate a number of cases where we can circumvent the schemes used.**

These findings are problematic as successfully breaking the fallback authentication gives full access to an account, just as breaking the main authentication mechanism. We conclude that implementations of fallback authentication mechanisms require more attention, both on a conceptual and an implementation level, as even seemingly minor implementation details can have a broad impact on the overall security. We have responsibly reported all attacks to the respective security teams well in advance of publication.

I. Introduction

User authentication is prevalent when accessing a large number of services on the Internet (and beyond). However, frequently users fail to authenticate due to loss of the authenticator: Using knowledge-based authentication the secret is frequently forgotten, using token-based authentication the token can be lost or stolen, and when using biometric-based authentication the biometric feature can become temporarily or permanently altered (a fingerprint may be altered by a cut, the voice might be unrecognizable due to illness, ...).

After loss of the authenticator, *fallback authentication* provides a mechanisms to allow the users to regain access to their accounts (the literature refers to fallback authentication also as *backup authentication*, *emergency authentication*, *last-resort authentication*, or *account recovery*). The requirements for fallback authentication are different from the requirements for "normal" authentication, as fallback authentication is only used rarely under exceptional circumstances: *Memorability* must be better than for "normal" passwords, as the frequent recall of passwords will help in learning, which is not the case for fallback authentication. Obviously, the overall security of the fallback authentication must be at least as high as for the normal mode of authentication, as otherwise fallback authentication provides a shortcut circumventing otherwise secure authentication. (See the (in)famous hack of Sarah Palin's email account in 2008 [25]). But as fallback authentication is not intended to be used regularly, time requirements are much more relaxed and stricter limitations can be applied, such as strict rate-limiting or a 24 hour lockout period, and thus a somewhat lower entropy of the secret can be tolerated.

A number of fallback authentication schemes have been proposed and are deployed in practice: Perhaps the most widely known scheme is based on *security questions*, where the website asks the user several questions such as the brand of his first car and compares the answers with the ones recorded when enrolling, resetting the password if these are answered correctly. However, the entropy of the answers to typically asked questions is low and guessing attacks are possible [19]. *Manually checking credentials*, either by sending in a copy of the user's passport or by a personal encounter, scales badly for Internet-wide services. It is sometimes used for such services if other fallback authentication fails, and it can be useful in corporate settings (where the entity checking in-person can be located in close proximity and security requirements are high). *Authentication by email* tests if the user has access to a pre-set email account by sending a reset-link or a temporary password to the email account; a similar approach sends such a code via SMS to a *mobile phone*. Both ideas have the drawback that neither email nor SMS are secure services, that simultaneous loss of passwords (e.g., caused by loosing several passwords that are stored in browser cache due to switching to another computer) cannot be recovered, or if the email access or phone number do no longer exist.

A relatively novel approach is *social authentication*, where a user is authenticated by his social contacts, either by sending secrets to selected contacts that, combined, allow him to reset his password, or by querying him about his contacts and preferences. This method has gained interest in the recent past, due to the rise of social networks, which facilitate implementations.

We take a closer look on fallback authentication methods that are deployed in the real world. We concentrate on social network sites, because (the most interesting method of) so-

1545-0678/14 $31.00 © 2014 IEEE
DOI 10.1109/ICDCSW.2014.30

cial authentication works reasonably only for social networks where some amount of social information is present. Our main contribution is an attack against Facebook's *Trusted Friends* fallback authentication. The attack is based on the idea of getting access to the secret codes required for account recovery by adding fake accounts to the victim's friends list. Facebook has measures in place to stop similar attacks, however, by cleverly gaming the interface with the help of POST data manipulation (see Section III-B) and Chain Trusted Friends attack (see Section III-E), we can work around these countermeasures and conduct a successful attack. This teaches us valuable lessons on what pitfalls there exist in implementing social authentication in an open network. We also explore the potential for successively applying this attack to compromise larger fractions of the user base, and discuss the effectiveness of the applied countermeasures. Furthermore, we round up this consideration by looking at social networks more broadly and evaluating their fallback authentication mechanisms.

A. Related Work

Probably the best known form of fallback authentication are *security questions*, sometimes also called *cognitive passwords*, and their security is well studied. One of the earlier studies on the security of security questions was conducted in 1993 [28] and found good usability and security. However, by modern standards the security is rather low, as demonstrated by a number of more recent studies. Griffith et al. showed [6] that the *mother's maiden name*, which is frequently used for such questions, can often be reconstructed from public databases, rendering them insecure. More generally, Rosenblum has shown the simplicity of learning private information about members of social networking sites [18]. This information can be used to narrow down potential answers for the security questions. Secrecy of those answers in the age of Facebook was studied by Rabkin [17], and Bonneau et al. studied the entropy of names [2]. Schechter et al. demonstrated [19] that for a number of such questions the answers can typically be guessed easily. A more general discussion on designing security questions including usability, privacy, and security is given by Just [11]. A potentially better domain for the security questions, namely personal questions similar to those used on online dating sites, where studied by Jakobson et al. [10] and found to provide better security than most other commonly used questions.

Another common form of fallback authentication uses a *registered email address* or mobile phone of the user [5], where an access code is sent to that registered device if the user lost his regular password and requested a password reset. This can work quite well, but several facets are problematic. First, sometimes a user looses more than one password, e.g., they might have been stored in the password cache on the same computer (which might be defective or was a work computer which was returned). Second, neither mobile phones nor the SMS service were designed with security as a main concern, and in fact Trojans for mobile phones have been found that capture authentication tokens that are sent to the phone [8]. Fallback authentication by support team is susceptible to social engineering attacks [14]. Social authentication, or vouching, was proposed by [3], a more recent design is given by Schechter et al. [20]. Problems with social authentication were pointed out in [12].

Considering the security of social networks more broadly, Huber et al. have shown how social networks in general (and Facebook specifically) can be exploited to distribute spam [7]. Also, Potharaju et al. have studied efficient techniques to befriend members of a targeted community in social networks [16]. After the attacker befriends the "weakest link" in this community, i.e., the initial access, its chances of befriending other, less accessible members rise. Our results directly benefit from their approach, as we only need to befriend three arbitrary friends or control three arbitrary accounts of a given community in order to gain access to the other community account via our Trusted Friends Attack. Wang *et al.* have analyzed the security of popular Single Sign-On schemes, including Facebook, leading to the discovery of worrisome security flaws [24]. Note that our attacks similarly allow an attacker to sign in to web services over Facebook on behalf of the compromised user. Irani *et al.* abused friend-finding features in an online social networking sites and have proposed reverse engineering attack in [4]. In reverse engineering attack, the attacker trick victim to send friendship request to him. Our attack can also take advantage from reverse engineering attack because if attacker is on victim's friend list then he can use his account in order to receive secret code from Facebook. Parwani *et al.* have exploited vulnerabilities in email accounts to gain access to Facebook accounts [15]. Note that in contrast to this work we do not consider a *compromise/hack* of a legitimate user's email account. Bilge *et al.* have shown that users on social networks accept friend requests and once friendship relationship has been established, the attacker may later use the private information against victim [13]. Our work can directly benefit from this work because if attacker is able to trick victim to accept friend request from three accounts, he may later use these accounts in order to receive secret code from Facebook. Boshmaf *et al.* were able to operate a botnet of fake accounts on Facebook for around 8 weeks and have shown that Facebook's defenses are not reliable in detecting socialbot [27]. In Trusted Friend Attack, we were also able to bypass Facebook's security measures and have shown that they are not effective. At the same time, TFA may also leverage the idea of socialbot in order to evaluate it on scale.

II. Facebook Trusted Friends

Facebook is probably the most important example for a provider using social authentication as an authentication mechanism. The prime reason is that Facebook has rich information about their users, including their social graphs, and in addition Facebook has a huge user-base, so efficient fallback authentication is important as forgotten passwords are frequent. Facebook is an attractive target for attackers, due to it's large user base and the vast information that is stored about users. The attacker's goals range from retrieving private information, spamming using Facebook's messaging mechanism, infection of computers with Malware (e.g., to launch clickjacking attacks), and more (e.g., [21]).

The *Trusted Friends* (TF) mechanism (also called *Guardian Angels* [9]) is a fallback authentication feature introduced by Facebook in October 2011 [23]. The work-flow of password recovery including the Trusted Friends feature is depicted in Figure 1, we will describe the relevant parts of it in the sequel.

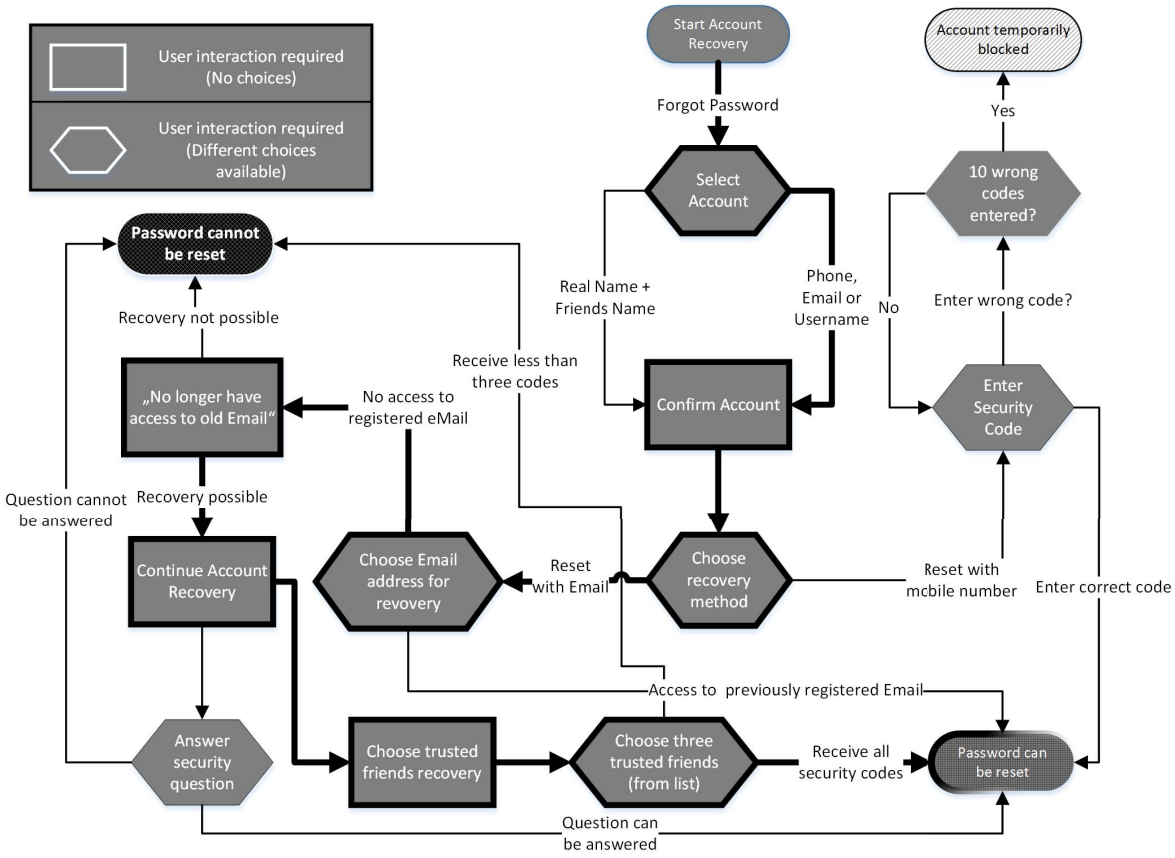

Fig. 1. Facebook Password Recovery Flow. (Bold arrows indicate our Trusted Friends attack scenario.)

A. Invoking Trusted Friends

A user can initiate account recovery by clicking a link on the Facebook homepage, where he can identify the account he wants to recover by a number of identifiers such as the email address, phone, full name, or username. If the user still has access to a previously registered email address or mobile phone, he can choose to have an access code sent there. However, there are a number of reasons why a user has lost access to his email as well: the user has lost access to his computer, e.g., by theft or defect, and stored the passwords in the browser cache, he used the same password for both and forgot it, he registered an email that went inactive in the meantime, e.g., a work address from a previous employer, or a freemail address that became inactive (in the case of HotMail, if an account is not accessed for about eight month it becomes inactive.[1]) In these cases, Facebook uses an unknown algorithm to determine if the user is offered the *Trusted Friends* authentication or not. (The Facebook security team gave us some hints on how this decision is made, in particular it requires more than 100 friends and that the friend list is public; we also suspect that the user has to have a sufficient number of "long-term friends".)

[1] http://email.about.com/od/windowslivehotmailtips/qt/Know_When_Your_Windows_Live_Hotmail_Account_Becomes_Inactive.htm

B. Trusted Friends Authentication

If Facebook allows Trusted Friends authentication, they display three lists of some of the user's friends and asks the user to pick one friend on each list. (Again, no information about how these friends are selected is available, but we believe that Facebook is trying to select "trustworthy" friends from different clusters (e.g., co-workers, family and friends), in particular to avoid an attack where an attacker befriends the victim using fake identities and selects those for circumventing the Trusted Friends authentication.) During the friend selection process Facebook initially shows a list of 100 names, reducing this number in half each time one person was selected. (Again, we assume this is done to counter attacks.)

The final selection of friends is then confirmed by the user, and if he continues, different four-digit secret codes are sent to each of those friends. These friends are supposed to verify the identity of their friend and hand him over their code (preferably using out-of-band communication, e.g., by phone). Upon entering all three codes the user may regain access to his account. None of the selected friends on their own receives enough information to access the user's account, as they have only access to one of the three secret codes. To prevent misuse, Facebook will send an email to the registered email address of the account, for which the recovery process has been started, containing the three friends' names to which recovery codes have been sent.

C. Trusted Contacts

Just recently, Facebook has implemented a feature called *Trusted Contacts*[2], which is similar to the Trusted Friends recovery feature. In contrast to the latter, which is only activated arbitrarily when trying to register a new email address for password recovery, this new feature can be activated for every user in the account security settings. While Trusted Contacts recovery is disabled by default, our study shows that the Trusted Friends feature is nevertheless available.

III. Trusted Friend Attack

Next, we describe the Trusted Friend Attack (TFA).

A. Recovery Flow of an Attacker

To start the attack, an adversary needs some information that identifies the victim, e.g., one of the following: email address, phone number, full name, or Facebook username (which is public information and part of every Facebook users' URL). He performs the following steps:

1) Ahead of time, the attacker befriends the victim using three fake accounts that are under his control. (Studies show that users quite often accept friendship requests from strangers [22]. We confirmed these findings in a small and informal study where we sent friendship requests from 3 fake accounts to 20 users, and 8 out of 20 accepted all three requests.)
2) Initiate the password recovery process and identify the intended victim as described in Section II.
3) Then there are two options:
   a) Facebook offers the attacker the option to register a new email address. (This decision is based on the number and status of friends, but the exact criteria are not publicly known.)
   Then the attacker can choose between two authentication mechanisms, classical security questions on the one hand, and Trusted Friends on the other hand. As we are, for now, interested in the Trusted Friend mechanism, we will choose that option and proceed with the attack.
   b) Facebook requires access to the original email address. The Trusted Friend mechanism is not involved, so this case is not interesting for us and we abort.
4) Select the three friends that are under the attacker's control by using POST data manipulation (see Section III-B). The attacker's controlled accounts can be fake accounts that are part of trusted friends lists of the victim, or the attacker may use already compromised accounts (e.g., by answering a security question) that are part of the victim's friend list.

B. POST Data Manipulation

For an attacker to gain access to some user's account by the Trusted Friend feature, he has to learn the three security codes sent to the trusted friends of this user. While an attacker that initiates the recovery process may select these friends in way that is beneficial for him (e.g., users that may be susceptible to social engineering), he is still very restricted as Facebook only offers subsets of users to choose from. (We learned empirically that Facebook is less likely to present freshly added friends on the trusted friends list.)

We found a way to circumvent the pre-selection of users by a manipulation of POST data in a way the attacker has complete freedom in choosing the friends from different clusters that will receive the three security codes. We were able to select arbitrary friends that were not even on the presented list of 100 users, which is substantially weakening the security of the scheme and allows to easily use arbitrary fake accounts for account recovery. (A common Facebook user has about 342 friends [26] and at the same time social network users are likely to accept friend requests even from unknown persons. According to [22], 50% of Facebook users accept friendship requests from unknown profiles.) Since we contacted the Facebook security team, this problem was (partially) fixed; currently the selection is restricted to the presented list of 100 users. (But still with our POST data manipulation, we can select any user from the full list in each of the three steps, and not just from the shortened lists that are presented in step two and three.)

Our POST data manipulation proceeds as follows. When selecting one of the three friends from the presented friend list, the POST data consists of the following information:

```
lsd=AVqJim1g&profileChooserItems={"FBID":
1}&checkableitems[]=FBID
```

Here, FBID is a placeholder for a numeric value, which may contain any Facebook user ID. `lsd` is a Cross-Site Request Forgery (CSRF) token and the next two parameters are `profileChooserItems` and `checkableitems`. The value of both parameters is the user ID of the selected friend, which will subsequently receive one of the three codes. An attacker can simply change the ID value in both parameters, replay the request and thus choose any friend of his choice.

Note that the attacker needs to know the user ID, which, in contrast to the user name, is not readily available in most cases. However, Facebook provides a *Graph API Explorer* tool,[3] that the attacker can use to learn user IDs with given user names in the following manner: The Graph API Explorer's GET request has the following format:

```
https://developers.facebook.com/tools/explorer?met
hod=GET&path=FBID?fields=id,name
```

The attacker can input any Facebook username as a value of the `path` parameter and replay the request. The API responds with the complete name along with the user ID. We have used the Live HTTP Headers extension[4] for replaying POST data. In the same manner the attacker is able to repeat the process for the selection of the second and third friend and the attacker's chosen accounts will receive the secret codes for recovery.

C. URL Manipulation

Before the final confirmation of requesting an account recovery, the URL has the following format:

[2]https://www.facebook.com/notes/facebook-security/introducing-trusted-contacts/10151362774980766

[3]https://developers.facebook.com/tools/explorer

[4]https://addons.mozilla.org/en-US/firefox/addon/live-http-headers/

https://www.facebook.com/guardian/confirm.php?guardians[0]=FBID0&guardians[1]=FBID1&guardians[2]=FBID2&cuid=AYi[...]XrNQaw&email=attackersuppliedemailaddressgoeshere

Here FBID0, FBID1, FBID2 are placeholders for the user IDs, cuid is the placeholder for hash value that Facebook calculates at real time, and attackersuppliedemailaddressgoeshere is a placeholder for the attacker's created email address. While the attacker cannot prevent Facebook from sending a notification email to the account for which the recovery was started (see Section IV), he still can tweak this URL in a way that the legitimate user will receive an email from Facebook that does not contain the names of the three friends. To this end the attacker proceeds as follows: The URL contains three user IDs, i.e., `FBID0`, `FBID1` and `FBID2`.

Suppose that the attacker has chosen himself as one friend during the recovery process of the victim and selected two other random friends. An obvious thought would be for the attacker to replace the other two user IDs with his own ID so that he alone will receive all three secret codes. But tweaking the URL this way only provokes an error message stating that the requested URL is invalid or expired. We think that the main reason is that the URL contains a parameter "`cuid`", which consists of some hash that Facebook calculates at real time during the selection of friends. But, although Facebook correctly says that the URL is invalid or expired, behind the scenes it will nevertheless send an email to the legitimate user, which however does not contain any friend name. Other possible ways for the attacker to tweak the URL are: change the hash value (i.e., value of query parameter `cuid`), change the array values (i.e., `guardians[1]`) by Cross-Site Scripting (XSS) vectors or send himself secret codes for different friends. In all cases the URL becomes invalid or expired but behind the scene the legitimate user gets an email.

D. Applicability

To evaluate the applicability of the *Trusted Friend Attack* we tested 250 accounts of one of the authors' friends. (This is, of course, not a representative sample, but will give us a fair idea on how wide-spread this problem is.) We test those accounts for the pre-requisites necessary to conduct the Trusted Friend Attack. We used Facebook usernames to start the recovery procedure. A Facebook username is public information and part of the URL, e.g., https://www.facebook.com/first_name.last_name.50596.

Out of the 250 accounts tested, 69 accounts allow account recovery if the user no longer has access to the registered email. (Note that the standard way to recover an account is by email to a registered email account; we are discussing account recovery if this email account is no longer accessible.) Out of these 69 accounts, the Trusted Friends recovery is available for 58 accounts (approx. 23% of all accounts). In a subsequent discussion with Facebook's security team, we learned that the Trusted Friends account recovery is generally available for accounts with a friends list which is publicly available and at least 100 friends.

Fig. 2. Users' Reaction on Facebook's Email or SMS

E. Chained Trusted Friends Attack

It should be clear that compromised accounts increase the number of accounts affected by TFA, thus creating a *Chained Trusted Friend Attack*. In CTFA, an attacker uses already compromised accounts in order to compromise more accounts, all without the consent of the legitimate users. This is not restricted to the Trusted Friends Attack but holds for any compromised account. This highlights that compromised accounts are not only a threat to the owner of the account, but also to his contacts. Further note that once the account recovery is started, further attempts to initiate account recovery are blocked, which is a form of denial-of-service attack.

IV. Facebook Security Measures and Bypasses

Facebook implements some protection measures to make attacks against the Trusted Friend scheme harder. In this section we discuss the effectiveness of these countermeasures, and will see that their effect is very limited.

A. Security Alert via Email or Mobile SMS

Facebook has a security measure in place in the form of email alerts. An attacker may start the account recovery process without knowing the email address of the victim, with the user name alone. Upon initiating the password recovery process, Facebook's automated system sends an email alert to the email address associated to the Facebook account of the user. In this email, Facebook informs the user that someone has initiated to reset either email address or password of his/her account. Most people ignored those emails, as they have never initiated the process. To learn more about this we have asked several of the persons involved and learned that they considered it as one of the many spam emails circulating. Some also were unaware that Facebook had such a recovery system in place, and nobody tried to contact the three friends selected during the process. As anecdotal evidence, we display some user reactions on Facebook in Figure 2.

B. 24 Hour Locked-out Period

When changing the password, Facebook does not grant immediate access to the targeted user's profile. Instead, the attacker has to obey a 24 hour locked-out period. During this 24 hour locked-out period the legitimate user may abort the

password recovery process by clicking on the link in the email he received and for the attacker the locked-out period link becomes invalid. It can be useful to monitor the online behavior of a targeted user and time the attack accordingly (e.g., on weekends or when they are on vacation).

C. Temporarily Locked

We have observed that Facebook temporarily locked the profile, if their system noticed that the access was attempted over an unrecognized device. Facebook's system identifies the device by matching the recognized browser version, estimated location, IP address, and operating system against a list of previously observed configurations. However, Facebook's temporarily locked functionality can be easily bypassed if the attacker clicks on `Continue`. Facebook then will asks the same security question that the attacker has already answered in order to reach this point. Along with the same security questions, Facebook may alternatively offer the user to identify his/her friends. As soon as the attacker answers the question again, Facebook will grant access to the victim's account.

V. Other Means of Fallback Authentication

To better understand our results for the Trusted Friends mechanism we reviewed alternative methods of fallback authentication. We concentrated on social network sites because they have a comparable user-base and thus we expect the results to be comparable, and consider the 50 social network sites with the highest Alexa-rank (see Appendix A). All sites offer account recovery by email to a registered email account. Further methods that were available were: registering a new email address with the support team, SMS to a registered mobile device, and answering security questions.

A. Recovery by Email to a Support Team

A commonly offered method for fallback authentication in those cases when the user has no longer access to his registered email is with the help of the support team, usually by phone or email. Basically all (49 out of 50) surveyed sites offered help from a support team.

We tested the support teams by creating a (fake) account and then trying to reset the password for this account from an unrelated email address. We were able to compromise accounts on six popular social networks (Academia, Delicious, GetGlue, FreizeitFreunde, Lokalisten and Meetup) with straight-forward *social engineering*, claiming our account was hacked and we needed assistance. Three sites (MeinVZ, Kwick and Jappy) asked for copies of official documents, MyLife's support team asked for a personal call on their call center and one site (Habbo) asked for details of our "avatar" on the site. Four social networks (Badoo, Experience Project, FriendScout24 and Yelp) respond by deleting the victim's account. 34 social networks did not respond at all, and it is unclear if they suspected cheating or have a bad service quality. See also the story of *Mat Honan* [1] and Mitnick's book [14] for more information on the relevance and techniques of social engineering attacks.

B. Recovery by SMS

One popular social network (VK 100 million active users) uses a registered mobile phone number for password recovery. It sends a verification code to the pre-registered mobile number which allows the user to reset the password. A potentially problematic behavior is that, for their specific implementation, the verification code does not change for subsequent recovery attempts (even when the phone number is changed). While we are not aware of an actual exploit this seems problematic. More generally, recovery by SMS suffers from the same problems as recovery by email: SMS is not a secure service, and phone numbers may change.

C. Recovery by Answering Security Questions

Two social networks (Facebook and StayFriends) offer this method. There is plenty of work demonstrating the weaknesses of security questions, which range from finding the answers on social network profiles to low-entropy answers for typical questions; we refer to Section I-A for more information.

VI. Conclusion

We have investigated fallback authentication mechanisms, with a focus on Facebook's Trusted Friends mechanism, as this is the prime example of a social authentication mechanism deployed in practice; for perspective we also tested other fallback authentication mechanisms as deployed by a number of social networks. In particular, we found the following issues:

- We found an error in the way that Facebook handles the friends that are selected to receive the codes, with the effect that the scheme is easy to break.
- We demonstrated several failed attempts to check user credentials in the fallback authentication by support team, which in several cases reset the password where the provided information was clearly not sufficient.
- We discussed the (known) weaknesses of other forms of fallback authentication.
- In general, they acknowledge that fallback authentication is still an open problem for them to solve, given their large user base.

The lesson learned from our work is that fallback authentication is not only conceptually hard to realize securely, but that also implementation details *do* matter. Implementors of those mechanisms need to understand in detail what the specific elements of the scheme are for. Facebook was apparently aware of the (straightforward) attack using several fake identities and implemented measures against it, but practice has shown that these are not sufficient (in particular client-side checking is inappropriate) and effective. Finally, we conclude that fallback authentication is still an unsolved problem.

VII. Acknowledgments

The authors would like to thank anonymous reviewers for their comments. The first author is supported by the Ministry of Economic Affairs and Energy of the State of North Rhine-Westphalia (Grant 315-43-02/2-005-WFBO-009), and the fourth author is supported by the German Research Foundation through the Graduiertenkolleg UbiCrypt (GRK 1817).

REFERENCES

[1] How Apple and Amazon Security Flaws Led to My Epic Hacking. Online at http://www.wired.com/gadgetlab/2012/08/apple-amazon-mat-honan-hacking/all/.

[2] Joseph Bonneau, Mike Just, and Greg Matthews. What's in a name? evaluating statistical attacks on personal knowledge questions. In *Financial Cryptography and Data Security*, volume 6052 of *Lecture Notes in Computer Science*, pages 98–113. Springer Berlin Heidelberg, 2010.

[3] John Brainard, Ari Juels, Ronald L. Rivest, Michael Szydlo, and Moti Yung. Fourth-factor authentication: somebody you know. In *Proceedings of the 13th ACM conference on Computer and communications security*, pages 168–178. ACM, 2006.

[4] Davide Balzarotti Engin Kirda Danesh Irani, Marco Balduzzi and Calton Pu. Reverse social engineering attacks in online social networks. In *DIMVA*, 2011.

[5] Simson L Garfinkel. Email-based identification and authentication: An alternative to pki? *Security & Privacy, IEEE*, 1(6):20–26, 2003.

[6] Virgil Griffith and Markus Jakobsson. Messin' with texas deriving mother's maiden names using public records. In *Applied Cryptography and Network Security*, volume 3531 of *Lecture Notes in Computer Science*, pages 91–103. Springer Berlin Heidelberg, 2005.

[7] Markus Huber, Martin Mulazzani, Edgar Weippl, Gerhard Kitzler, and Sigrun Goluch. Exploiting social networking sites for spam. In *Proceedings of the 17th ACM conference on Computer and communications security*, pages 693–695. ACM, 2010.

[8] ZeuS Mitmo: Man in-the mobile. Online at http://securityblog.s21sec.com/2010/09/zeus-mitmo-man-in-mobile-i.html.

[9] Facebook Security Infographic. Online at http://sophosnews.files.wordpress.com/2011/10/facebook-security-infographic.pdf.

[10] Markus Jakobsson, Erik Stolterman, Susanne Wetzel, and Liu Yang. Love and authentication. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 197–200. ACM, 2008.

[11] Mike Just. Designing authentication systems with challenge questions. *Security and Usability: Designing Secure Systems That People Can Use*, pages 143–155, 2005.

[12] Hyoungshick Kim, John Tang, and Ross Anderson. Social authentication: Harder than it looks. In *Financial Cryptography and Data Security*, volume 7397 of *Lecture Notes in Computer Science*, pages 1–15. Springer Berlin Heidelberg, 2012.

[13] Davide Balzarotti Engin Kirda Leyla Bilge, Thorsten Strufe. tacts are belong to us: Automated identity theft attacks on social networks. In *WWW*, 2009.

[14] Kevin D. Mitnick and William L. Simon. *The Art of Deception: Controlling the Human Element of Security*. Wiley, 2002.

[15] Tarun Parwani, Ramin Kholoussi, and Panagiotis Karras. How to hack into facebook without being a hacker. In *Proceedings of the 22nd international conference on World Wide Web companion*, pages 751–754. International World Wide Web Conferences Steering Committee, 2013.

[16] Rahul Potharaju, Bogdan Carbunar, and Cristina Nita-Rotaru. ifriendu: leveraging 3-cliques to enhance infiltration attacks in online social networks. In *Proceedings of the 17th ACM conference on Computer and communications security*, pages 723–725. ACM, 2010.

[17] Ariel 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*, pages 13–23. ACM, 2008.

[18] D. Rosenblum. What anyone can know: The privacy risks of social networking sites. *Security Privacy, IEEE*, 5(3):40–49, 2007.

[19] Stuart Schechter, A. J. Bernheim Brush, and Serge Egelman. It's no secret. measuring the security and reliability of authentication via "secret" questions. In *Proceedings of the 2009 30th IEEE Symposium on Security and Privacy*, pages 375–390. IEEE Computer Society, 2009.

[20] Stuart Schechter, Serge Egelman, and Robert W. Reeder. It's not what you know, but who you know: a social approach to last-resort authentication. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1983–1992. ACM, 2009.

[21] Sophos Security. Online at https://www.facebook.com/SophosSecurity.

[22] Robert Siciliano. Fake friends fool facebook users. Online at http://blogs.mcafee.com/consumer/fake-friends, 2013.

[23] National Cybersecurity Awareness Month Updates. Online at https://www.facebook.com/notes/facebook-security/national-cybersecurity-awareness-month-updates/10150335022240766.

[24] Rui Wang, Shuo Chen, and XiaoFeng Wang. Signing me onto your accounts through facebook and google: A traffic-guided security study of commercially deployed single-sign-on web services. In *Proceedings of the 2012 IEEE Symposium on Security and Privacy*, pages 365–379. IEEE Computer Society, 2012.

[25] Wikipedia. Online at http://en.wikipedia.org/wiki/Sarah_Palin_email_hack.

[26] Stephen Wolfram. Data science of the facebook world. Online at http://blog.stephenwolfram.com/2013/04/data-science-of-the-facebook-world/, 2013.

[27] Konstantin Beznosov Matei Ripeanu Yazan Boshmaf, Ildar Muslukhov. The socialbot network: When bots socialize for fame and money. In *ACSAC*, 2011.

[28] M. Zviran and W. J. Haga. A comparison of password techniques for multilevel authentication mechanisms. *The Computer Journal*, 36(3):227–237, 1993.

APPENDIX

A. List of the social networks used in the study.

Academia	FourSquare	last.fm	Pinterest	Viadeo
Badoo	FreizeitFreunde	LinkedIn	Plaxo	VK
Bebo	FriendScout24	Lokalisten	Plurk	Wayn
Cafemom	Friendster	Meetme	Schueler.cc	WeeWorld
Care2	Gaiaonline	Meetup	Sonico	Twitter
Classmates	GetGlue	MeinVZ	Spin	Xanga
Couchsurfing	Habbo	MyHerritage	StayFriends	XING
Delicious	Hi5	mylife	Stumbleupon	Yammer
Experienceproject	Jappy	MySpace	Tagged	Yelp
Flickr	Kwick	Netlog	Facebook	Zoosk