Friend-in-the-middle Attacks

Technical Report TR-SBA-Research-0710-01

Markus Huber*, Martin Mulazzani*, Edgar Weippl*,
Gerhard Kitzler*, Sigrun Goluch*

*SBA Research
Favoritenstrasse 16
AT-1040 Vienna, Austria
{mhuber,mmulazzani,eweippl,gkitzler,sgoluch}@sba-research.org

Abstract. In the ongoing arms race between spammers and the multi-million dollar anti-spam industry, the number of unsolicited e-mail messages (better known as "spam") and phishing has increased heavily in the last decade. In this paper, we show that our novel *friend-in-the-middle attack* on social networking sites (SNSs) can be used to harvest social data in an automated fashion. This social data can then be exploited for large-scale attacks such as context-aware spam and social-phishing. We prove the feasibility of our attack exemplarily on Facebook and identify possible consequences based on a mathematical model and simulations. Alarmingly, all major SNSs are vulnerable to our attack as they fail to secure the network layer appropriately.

1 Introduction

Criminals, as well as direct marketers, continue to clog mailboxes with unsolicited bulk e-mails (e.g., spam and phishing) in the hope of financial gain. So far, their strategy is straightforward, namely to send out a vast numbers of unsolicited e-mails in order to maximize profit on the tiny fraction that falls for their scams. Their pool of target e-mail addresses is normally based upon data harvested with web crawlers or trojans, sometimes even including plain dictionary-based guessing of valid targets. Previous research indicates that social networking sites (SNSs) might change the playing field of spam attacks in the near future. SNSs contain a pool of sensitive information which can be misused for spam messages, namely contact information (email addresses, instant messaging accounts, etc.) and personal information which can be used to improve the believability of spam messages. A successful extraction of sensitive information from SNSs would result in spam attacks that are based upon a pool of verified e-mail addresses. Thus messages may have higher conversion rates, increasing the success rate of spam.
Gaining access to the pool of personal information stored in SNSs and impersonating a social network user poses a non-trivial challenge. Gross and Acquisti [22] as well as Jones and Soltren [32] were among the first researchers to raise awareness for information extraction vulnerabilities of SNSs. While their techniques were rather straightforward (automated scripts which retrieve web pages), their results eventually led to security improvements of SNSs. Existing attempts to extract information from SNSs focus on the application layer and can thus be mitigated by adapting a specific social network's application logic. Recent publications devoted to information extraction from SNSs introduced elaborate methods such as the inference of a user's social graph from their public listings [11] or cross-platform profile cloning attacks [8]. The leakage of personal information from these platforms creates a remarkable dilemma as this information forms the ideal base for further attacks. Jagatic et al. [30] showed that they could increase the success rate of phishing attacks from 16 to 72 % using "social data". In social engineering, additional available information on targets could lead to automated social engineering attacks [28]. The main obstacle for large-scale spam attacks on basis of SNSs are the various access protection measures providers offer to keep sensitive information private or at least limit access to a closed circle of friends. Our friend-in-the-middle attack overcomes this obstacle by hijacking HTTP sessions on the network layer, which the majority of SNSs providers fail to secure.

The main contributions of our work are:

- A novel friend-in-the-middle attack on social networks and how it can be used for context-aware spam and social phishing on a large scale.
- An evaluation of the feasibility of our attack on basis of Facebook.
- A simulation to estimate the impact a friend-in-the-middle spam campaign would have.
- A discussion on protection measures and mitigation strategies.

The rest of the paper is organized as follows: Section 2 provides brief background information on spam, social graphs models and the prevalence of vulnerable social networking sites. Section 3 outlines the friend-in-the-middle attack. We explain our methodology used to verify the practicability of our attack for large-scale information extraction in Section 4, while describing our model to simulate the possible effects on real-world SNSs in Section 5. Our findings and mitigation strategies are discussed in Section 6, followed by our overall conclusion in Section 7.

2 Background

In the following section we give a brief introduction to related research in the areas of spam, phishing and how data from SNSs might change the success rate of these malicious messages. Furthermore we discuss which social networking services are vulnerable to our novel attack. Finally, we outline theoretical graph models for social networks.

Spam and Phishing. It is believed that the vast majority of emails sent today are spam, accounting for more than 90% of all emails. Empirical studies showed that while the conversion rate of spam is quite low, it is apparently still sufficient for the spammers to make money [33]. Phishing, on the other hand, can be seen as the marriage between social engineering and spam, where attackers try to fool victims into entering their login credentials into malicious websites that mimic a real website (e.g., a bank). As phishing uses the same attack vector (email) and infrastructure as spamming (phishing websites are hosted on fast flux networks [27]), research in this area is closely related to spam and botnet research. Current research focuses on preventing spam delivery and how botnets are used for sending spam (e.g., the Storm [35] or Szizbi [49] botnets) to defeat IP blacklisting from email service providers. In practice, a combination of various techniques is used to minimize spam: sender-based systems such as SPF [54], IP blacklisting such as the Spamhouse blocklist [48], and content matching systems such as SpamAssassin [47].
Social networking sites might change the way large-scale spam and phishing attacks are carried out. Jagatic et al. [30] showed how information extracted from online social networks significantly increased the success rate of phishing, while Brown et al. [12] raised awareness that these emerging online services could also form the basis of context-aware spam. With information extracted from social networking sites, spam and phishing messages become tailored to the receiver. Tailored messages eventually result in fewer messages that are required to gain the same effect as with huge spam campaigns. For example, attackers could use pictures extracted from a friend's social networking services or sign messages with the name of the target's friend in order to increase the apparent of spam and phishing messages.

Social Networking Site		
Name	Claimed users	HTTPS
Facebook	400×10^6	Login only
Friendster	110×10^6	No
Orkut	100×10^6	Login only
hi5	80×10^6	No
LinkedIn	60×10^6	Login only

Table 1. Top five social networking sites and their support for HTTPS.

Social Networking Sites (SNSs). SNSs have become one of the most popular online services and are used by millions of users around the world. They offer a platform to foster social relationships over the Internet and are in general free of charge. The business paradigm these services follow is similar to Google's in the sense that users do not have to pay for service usage and profit is generated via online advertising. Because people provide plenty of personal information about themselves on social networks, advertisers can effectively target specific demographic groups (e.g., an advertisement for male college students between the age of 18 and 22 years in the US). Hence, in order to make SNSs attractive platforms for advertisers, SNSs encourage their users to share as much information about themselves as possible [16].

The European Network and Information Security Agency (ENISA) published a position paper on the information security of SNSs [26] and introduced four threat categories which are useful to understand all the information security risks that are involved with SNSs usage. Within this paper we focus however only on two attack vectors: unencrypted network communication between users and SNSs providers, as well as information leakage through third party applications.

Data between the user and the social networking platform is usually transmitted over the unencrypted HTTP protocol, while only some networks protect the transmitted login credentials with TLS. This means that the entire communication content (including the information with whom a user is friends with, status updates and pictures) are vulnerable to eavesdropping. We hypothesize that the social network operators refrain from offering their services over a secure channel for performance and cost reasons (compare in [24]). Table 1 shows the biggest social networking sites at the time of writing and their support for HTTPS. The ranking of the different SNSs is based on their self-claimed user-bases [15,50,18,41,36] and no reliable numbers on the size of their networks exist.

Like other web services, SNSs rely on session cookies to track the state of a certain user. These session cookies are saved locally at the client side, containing among other information a shared, hashed secret. This shared secret is used as a proof that the users has been successfully authenticated by providing username and password. As these cookies are transmitted unencrypted, the communication between a user and a SNS provider is vulnerable to cookie hijacking. Thus, an

Social Networking Site		
Name	App. Registration	Client Libraries
Facebook	open	PHP5 JS library[a]
Friendster	open	OpenSocial API
Orkut	open	OpenSocial API
hi5	open	OpenSocial API
LinkedIn	closed	OpenSocial API

Table 2. Support for custom applications.

[a] A number of unsupported libraries for other common scripting languages are available as well. The PHP5 library though, is officially supported by Facebook.

attacker could take over a user's social networking sessions by sniffing out the HTTP cookies, since the majority of SNSs providers do not support HTTPS.

Another related risk is the support for third party applications. SNSs providers offer a developer API for third party applications. These APIs provide yet another way to tap into the pool of personal information stored within SNSs. Once a user adds a certain third party application to her/his profile, the application is automatically granted access to this user's personal information. Table 2 shows that today's biggest SNSs already support custom applications, while LinkedIn is the only SNS which has an application review process. With the remaining four platforms, custom applications can be added by anyone without any prior security or privacy screenings.

SNS and graph theory. Since the dawn of the Internet, there has been extensive theoretical work in modeling the structure of social networks [17]. The various proposed models are focusing on the realization of certain properties of the social graphs observed in practice. Recent work on social networks within mathematics has focused on three distinctive features of network structure: the small average path length (APL) or *small world* effect [39], the high *clustering* effect and the property of having a *scale free degree distribution* [7]. Within network simulations and modeling there are basically three main classes of paradigms. The first one, which is probably the simplest useful model of a network, is the classic random graph or *Poisson* random graph [14,46]. Random graph theory has been well studied by mathematicians [10,31]. It is still widely used in many fields and serves as a benchmark for many modeling studies. However, the properties of these classic random graphs are not consistent with the properties observed in real social networks. The second class, motivated by the small world effect, are small-world models. The idea of shortcuts for small-world models was proposed by Watts and Strogatz [53], followed by many other scientists of all specialities [5,44]. These models account for the so called *six degrees of separation* phenomenon. Finally there are scale-free models [6] which are basi-

cally networks with a given degree distribution, more precisely with a power law distribution. The discovery of the power-law degree distribution has led to the construction of various scale-free models, which try to provide a universal theory of network evolution and the realization of the skewed degree distribution. For an overview of various analytical models, we refer the interested reader to [4].

3 FITM Attacks

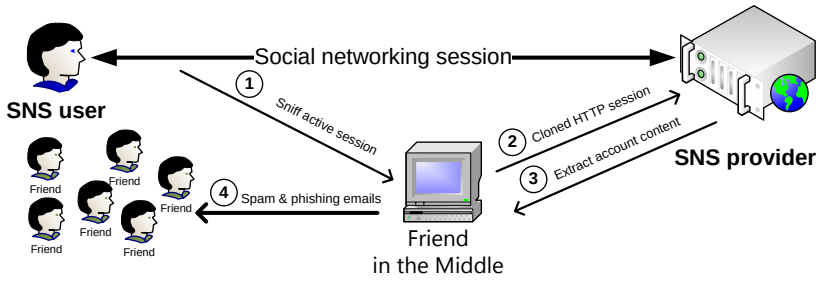

Fig. 1. Outline of a large-scale spam campaign via the friend-in-the-middle attack: A social networking session is hijacked to fetch personal information from a victim's profile. The extracted information is then used for spam and phishing emails targeted at the victim's friends.

We define friend-in-the-middle attacks as active eavesdropping attacks against social networking sites. Our FITM attack is based on the missing protection of the communication link between users and social networking providers. By hijacking session cookies, it becomes possible to impersonate the victim and interact with the social network without proper authorization. While at first glance the risk of hijacking social networking seems like yet another threat to privacy, we claim that FITM attacks enable large-scale spam attacks. Within this section, we first explain various attack scenarios on basis of session hijacking and describe how FITM attacks could be misused for large-scale spam campaigns on basis of Facebook.

HTTP Session Hijacking Attacks on SNSs. As a precondition the attacker needs to have access to the communication between the SNSs and the user. This can be achieved either passively (e.g., by monitoring unencrypted wireless networks) or actively (e.g., by installing malicious software on the victim's computer). The adversary then simply clones the HTTP header containing the authentication cookies and can interact with the social network, unbeknownst to the SNS operator or user. The victim is unable to detect or prevent such attacks and the attacker is able to use the social network to its full extent from the victim's point of view. As with all HTTP session hijacking attacks, it becomes

possible to both retrieve information (*data acquisition from the social network*) as well as to insert malicious requests on the behalf of a user (*data publication into the social network*). However in the case of our FITM attack, further scenarios become available to attackers, which are specific to social networking sites:

- *Friend injection* to infiltrate a closed network
- *Application injection* to extract profile content
- *Social engineering* to exploit collected information

The rudimentary security and privacy protection measures of SNSs available to users are based on the notion of "friendship", which means that sensitive information is made available only to a limited set of accounts (friends) specified by the SNS user. Once an attacker is able to hijack a social networking session, she/he is able to add herself/himself as a friend on behalf of the victim and thus infiltrate the target's closed network. The *injected friend* could then be misused to access profile information or to post messages within the infiltrated network of friends.

By *installing* a custom third-party *application* [40], written and under the control by the attacker, it is possible to access the data in an automated fashion. Among other things, an application has access to sensitive information (birthday, email address, demographic information, pictures, interests) and in case of most SNSs to information of friends of the application user. Third-party applications such as online games have become a popular amusement within SNSs, and hiding a malicious application without any activity visible to the user is possible. Thus, the application is likely to remain undetected within a pool of installed third-party applications. This ultimately enables an attacker to extract profile content in a stealthy way as this retrieval method does not cause as much noise as a burst of separate HTTP requests. Even worse, the attacker might install the application, take all the data needed in an automated fashion and remove the application afterwards. This would be completely undetectable to the user and most likely to the SNSs providers as well.

Whereas *social engineers* traditionally relied upon context-information gathered through dumpster diving or quizzing people over the phone, with FITM attacks the context-information harvesting process becomes automated. We thus claim that FITM attacks allow sophisticated social engineering attacks. Two such social engineering attacks based on information extraction from social networking sites are context-aware spam and social phishing. These advanced versions of traditionally spam and phishing messages are described bellow as they are ultimately used to show the devastating effect a large-scale FITM attack might cause.

Context-Aware Spam. Context-aware spam can be generated from data harvested with FITM attacks, increasing the effectiveness of the spam. Brown et al. [12] identified three context-aware spam attacks which might be misused: relationship-based attacks, unshared-attribute attacks, as well as shared-attribute attacks. While the first attack is based on relationship information, the two remaining variations use content extracted from social networking sites such

as geographic information or a user's birthday. The social network itself might be used for sending the spam, e.g. by writing the message to other users' walls, or by sending it via private messages. If used on a large scale, the messages spam might get detected and removed by the SNSs providers. However, if used only on a small scale we believe that this would be feasible as well as effective. Out-of-bound spam from the SNS point of view would be another possible approach, whereas emails are used for sending spam messages. This traditional email spam is enabled through the availability of real email addresses users make available to their friends. Hence, if the spam attack is carried out over email instead of the SNS platform, these malicious messages cannot be detected by the SNSs providers.

Social-Phishing. Phishing is a common threat on the Internet where an attacker tries to lure victims into entering sensitive information like passwords or credit card numbers into a faked website under the control of the attacker. It has been shown [30] that social phishing, which includes some kind of "social" information specific to the victim, can be extremely effective compared to regular phishing. For example such information might be that the message appears to be sent from a person within the social environment of the victim, like a friend or a colleague from work. The social graph is therefore not only for the social network operator of value, but for an attacker too. Especially if it contains additional information like a valid email address or recent communication between the victim and the impersonated friend. With automated data extraction from social networks, a vast amount of further usable data becomes available to the spammers. Prior conversations within the social network like private messages, comments or wall posts could be used to deduce the language normally used for message exchange between the victim and the spam target. For example, a phishing target might find it very suspicious if the victim sends a message in English if they normally communicate in French. The message communication channel for sending the phishing link is of importance as well, as a phishing victim is more likely to click on a link that appears to match previous communication patterns. All of this information can be used to identify the subset of a victim's friends that are most likely to fall for such a phishing attack. It is not necessary to phish all the victim's friends but only a subset if the likelihood for success is high enough. Another novelty would be that the phisher could include authentic pictures, either of the victim or the phishing receiver. Extracted images could for example be used to send invitations to shared "photo albums", including a link which promises more pictures given that a user enters his social networking credentials.

Large-scale spam campaigns through FITM attacks. Figure 1 illustrates the outline of a spam campaign exploiting our novel FITM attack.
(1) In the first step, a network connection is monitored. Once the FITM application detects an active social networking session, it clones the complete HTTP header including the session cookie. **(2)** The cloned HTTP header serves then as a valid authentication token for the SNS provider and is used to temporarily hijack the SNS user's session. **(3)** In order to extract the profile content as

well as information on the target's friends, a custom third-party application is added to the target's profile. Once all information has been extracted the application is removed from the profile. Additional queries are used to fetch the email addresses of the target's friends in case they cannot be retrieved through the third-party application. **(4)** The extracted email addresses and account content are used to generate tailored spam and phishing emails. While the spam messages contain the actual payload of the attack, the phishing emails are used to steal credentials of the target's friends for further propagation (the FITM attack starts again from (3) with the phished SNS account credentials).

We decided to evaluate the impact of a large-scale spam campaign on basis of Facebook. FITM attacks based on Facebook serve in our opinion as a good example because it is the biggest SNS at the time of writing, HTTPS is only used to protect login credentials and Facebook supports custom applications. Furthermore, injections of third-party applications into Facebook profiles promise access to a plethora of personal information. Within the Facebook application framework, third-party applications can access the following information[1]:

- *Basic context information*: Full name, geographical location, birthday, affiliations, education, etc.
- *Likes and interests*: Favorite books, movies, tv-series, music, quotations, etc.
- *Private content*: Sent and received messages, photos, videos, etc.

In addition, third-party applications within Facebook are allowed to access the information of a user's friends as well. Thus an application injection in Facebook enables the extraction of a pool of valuable context information from the targeted user as well of his/her friends. Email addresses of users are not accessible through third-party applications and the addresses can be collected by using the hijacked user session. We created a proof-of-concept implementation of our novel FITM attack in the Python scripting language for Facebook. The proof-of-concept application uses the dpkt library [1] to gather network packets and the mechanize library [2] to interact with Facebook.

4 Methodology

In order to make assertions on the effectiveness of our FITM attack, an experiment which mimics a real large-scale attack would provide valuable insights on effectiveness, but raises also serious ethical concerns. Hence, we applied the following twofold approach: we made an empirical evaluation on the number of possible sessions that could have been hijacked, without collecting any data or injecting any malicious requests. We furthermore simulated the impact of our FITM attack on basis of established results from similar work in a model. In comparison to survey-based methods we avoid problems such as selection bias, refusal rates, telescoping, forgetting and exaggeration [25]. The methodology applied within our two experiments is explained in this section.

[1] [19] as well as [20] summarize all context information that can be accessed through Facebook third-party applications.

Finding attack seeds. To conduct the FITM attack, numerous attack vectors could be used: DNS poisoning, cross-site request forgery (*CSRF*), wireless networks without or with only weak encryption, malicious software like a trojan or a rootkit running on the victims computer, deep packet inspection from an ISP or other malicious entity that has access to the traffic between the client and the SNS, as well as modified software running on a users residential or company gateway. However, we used our proof-of-concept application to analyze HTTP cookies from Facebook sessions passing through a Tor exit node.

The Tor network [13] is a widely deployed anonymization network which hides the user's IP address on the Internet. It is expected to be used by hundreds of thousands of users every day and is believed to be the most heavily used open anonymization network today [51]. The Tor infrastructure relies on servers run by volunteers, hence anyone can support the Tor project by setting up a dedicated Tor server. For our experiment, we have set up a Tor exit node on a minimal GNU/Linux Debian server with a relay bandwidth rate of 5 Mbit. The server was furthermore configured to only allow HTTP traffic (TCP port 80) from the Tor network to the Internet, and the Tor daemon was restarted on a daily basis.

We then counted the number of Facebook sessions together with the Facebook locales that were observable to our Tor server and could have be used for the FITM attack. The Facebook locales were necessary as these would be needed information by an adversary when conducting further context-aware spam or social phishing attacks. To prevent counting the same user multiple times, we saved hash values of the static session information in an encrypted file container. Note that we only counted and saved the number of FITM injection possibilities, not the number of users that used our Tor node. Our attack is not restricted to Facebook but is also applicable to all SNSs that fail to secure the network layer and the transmitted session information. We decided to use Facebook as it the biggest social network at the moment, claiming more then 400 million users.

FITM attack simulation. We decided to perform a simulation in order to estimate the impact a large-scale FITM attack would have. The underlying model of our simulation is further explained in section 5 while the results of our simulation are outlined in section 6.

5 Simulation Model

In this section, we will define the model used to test the attack strategies defined in the above sections. We will also give a short insight into social graph theory and graph modeling. But of course, our main focus lies on our approach to model a social graph for the purpose of testing attack strategies.

Social network theory is an area of network science which itself is an area of graph theory. Therefore, we will use the common graph theory notation for the Facebook graph. We consider Facebook to be an undirected social graph $G = (V, E)$ with $v_i \in \mathbb{N}$ and $E \subseteq [V]^2$, where the nodes $v_i \in V$ are the users and the edges $\{v_i, v_j\}$, denoted as $v_i v_j \in E$, are the connections between two users v_i and v_j. It is undirected due to the fact that friendships in Facebook are mutual

friendships. If $V' \subseteq V$ and $E' \subseteq E$, then $G' = (V', E')$ is a *subgraph* of G, written $G' \subseteq G$. If $E' = \{v_i v_j \in E \mid v_i, v_j \in V'\}$, then G' is called an *induced subgraph*. A graph is called *simple* when it has no *self-loops* and no *multiple edges* between any pair of nodes. The *degree* $d_G(v)$ of a node v is the number of neighbours which is equal to the number of edges on v, denoted $|E(v)|$. Note that for an induced subgraph G' the degree of a node $v \in V' \subseteq V$ is $d_{G'}(v) \leq d_G(v)$.

We say that an *extended induced subgraph* is a simple graph with following enhancements of V' and E': Let $\widetilde{E'}$ be the set of edges with one endpoint being in V' : $\widetilde{E'} := \{v_i v_j \in E \backslash E' | v_i \in V'\}$ then $\widetilde{V'} := \{v_j \in V \backslash V' | v_i v_j \in \widetilde{E'}\}$. $\widetilde{V'}$ contains all nodes which have a direct connection to a node within the induced subgraph nodeset V'. With this extension we obtain a new graph $\widetilde{G} := (\widetilde{V}, \widetilde{E})$ with $\widetilde{V} = V' \cup \widetilde{V'}$ and $\widetilde{E} = E' \cup \widetilde{E'}$.

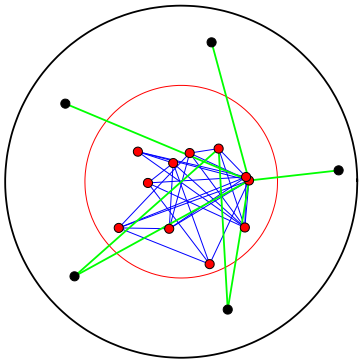

Fig. 2. extended induced subgraph $\widetilde{G}$.

Facebook has currently a total node amount of $|V| = 4 \cdot 10^8$. A model this exceeding is hardly possible to simulate due to its sheer enormity of nodes, edges etc.

We assume that the Facebook graph, denoted as $F = (V_F, E_F)$, consists of $n \in \mathbb{N}, n < \infty$ extended induced subgraphs
$\{G_1, G_2, ..., G_n\} \subseteq F$ with a *pairwise almost disjoint* property: $|G_i \cap G_j| < \varepsilon$ with $i, j = 1, ..., n \wedge i \neq j$ and $\varepsilon \in \mathbb{N}$ is "small enough".
Concerning the degree of a node, following proposition holds:[2]

$$\text{for } v \in V_i' \subseteq V_F \ : \ d_F(v) = d_{G_i}(v)$$

Note that it makes sense to use this subgraph assembly of Facebook. Imagine V_i' consists of nodes satisfying a certain property like *nationality, gender* or *affiliations of other kinds*. Handcock et al. [23] showed that such affiliations

[2] V_i' is the set of nodes from the induced subgraph G_i', whereas G_i is the extended induced subgraph where G_i' is the underlying induced subgraph

implicates clustering within the social network.
There are various publications which focus on the acquisition of actual data such as [21,34]. We do not claim to have developed a new way of modeling a social graph which applies better to Facebook than the various existing models introduced in the related work section, but we will use the very recent results of those publications to implement an almost accurate extended induced subgraph of a social network, such as Facebook.

Modeling a Facebook Subgraph. We will give a short insight into the world of *configuration models*. More rigorous disquisitions on configuration models can be found in [42,52]. We wish to construct a simple graph out of a degree distribution p_k, such that p_k is the fraction of nodes in the graph having degree k. Then choosing a degree sequence $\mathbf{d} = \{d_i \mid i = 1, ..., n\}$, w.o.l.g. we assume $d_i \geq 1$, from this distribution, which are the degrees of the n nodes $\{v_1, v_2, ..., v_n\}$. Since it is not always possible to construct a simple graph with a given degree sequence [52], we can definitely construct a *multigraph*. Obtaining a simple graph out of such a multigraph is easily achieved by erasing all loops and combing all multiple edges into one. The obtained simple graph has asymptotically the same degree distribution. It has been shown [42] that the chance of finding a loop goes as n^{-1}, therefore the probability is humble for large n. As an example for an applicable degree distribution we want to consider the much studied *power-law* distribution [43]. Distributions of the form $p(x) = Cx^{-\alpha}$ are said to follow a power law, where $\alpha > 0$ is called the *exponent* and C functions as a *normalizing constant*. C is given by the normalization requirement

$$1 = C \int_{x_{min}}^{\infty} x^{-\alpha} dx = \frac{C}{1-\alpha} \left[x^{-\alpha+1} \right]_{x_{min}}^{\infty} \tag{1}$$

Formula 1 shows that: 1) $\alpha > 1$ and 2) for a given $\alpha > 1$ and known limit x_{min} it is easy to compute the normalization constant C.

Attack Cycle. For an attacker the overall knowledge of the properties of the entire graph is unknown. What he does know is the degree of a node and also the degrees of its neighbours. Other useful knowlegde such as centrality, assortiativity, betweenness, clustering coefficient etc. is unknown to the attacker. The attack process behaves as follows: We choose a random node v_i, in the following called **user**, the user has a predetermined degree ($d(v_i) = d_i = k$) which is the amount of friends. We *spam* a fixed percentage p of the users friends and *propagate*[3] the remaining ones. This cycle then repeats itself for a given amount of iterations, e.g. 5 times ($it = m = 5$). Therefore we get probably more nodes to spread from in these iteration steps than in the first one. A user can either be *spammed* or *propagated* and no more than 1 time. Therefore it is possible that the attack cycle breaks after 1 iteration. This is when the starting user v_1 has only friends with degree $d_i = 1$, because the propagated user "has nowhere to go" then to go back to the staring point, which is not possible since this one is already *propagated*. Although this will not happen in a real network, due to the

[3] As the our propagation strategy is social-phishing and on basis of Jagatic et al. [30] we asume that propagations have a success rate of 72 %.

fact that a user with a high degree tends to have friends with high degree as well [21], a model is never immune to such a case. One may ask himself how to make the best attack strategy out of the above described attack cycle. We tested two strategies within our implemented model. **Strategy 1:** randomly choose a user, spam and propagate as mentioned above for $it = 1, ..., m$ times. **Strategy 2:** randomly choose a user, fix the number of iterations ($it = m$), after repeating the cycle m times jump to another randomly choosen user and repeat the cycle another m times. Jump for $jp = 1, ..., l$ times.

6 Results

Based on the model we needed to evaluate how many FITM attacks would be possible within a reasonable amount of time and effort. Our method for finding attacks seeds is just one out of many and presents a snapshot of this particular time interval, other methods might be far more successful. We did not compare different attack seeds discovery methods.

6.1 Finding Attack Seeds

During a period of 14 days, approximately 6.1 x 10^6 HTTP requests passed through our Tor exit node. Facebook was the most requested domain and was responsible for 7.68 % of the overall traffic. The second most frequent social networking site was Orkut which caused 0.49 % off all HTTP requests. We observed 4267 unique Facebook sessions throughout our experiment which could have been hijacked for friend-in-the-middle attacks. Furthermore our cookie analysis suggests that the majority (92.81 %) of observed unique Facebook sessions were persistent sessions.

One alternative source for attack seeds is eavesdropping on a WLAN. Indicative experiments on a university's WLAN showed that we could gather 60 seeds within seven hours. The main drawback of this method is that seeds are not dispersed as evenly over the entire social graph as many users are students are friends or share at least one common friend. Table 3 shows the distribution of the Facebook sessions in regard of the used language. 71.77 % of all users used English, followed by 6.19 % used an Italian, and 5.55 % of users used Spanish. We furthermore observed that in total 3.45 % of users might have originated from China and 1.28 from Iran, where in both countries Facebook is blocked by governmental authorities. The information of the different locales used could be exploited by attackers to adapt the language of their spam messages.

6.2 Simulation Results

We implemented a configuration model (see Section 5) with a power-law degree distribution. Gjoka et al. [21] presented a new degree distribution for Facebook which does not follow a power-law. Instead they found two regimes $1 \leq k < 300$ and $300 \leq k \leq 5000$, each following a power law with exponent $\alpha_{k<300} = 1.32$

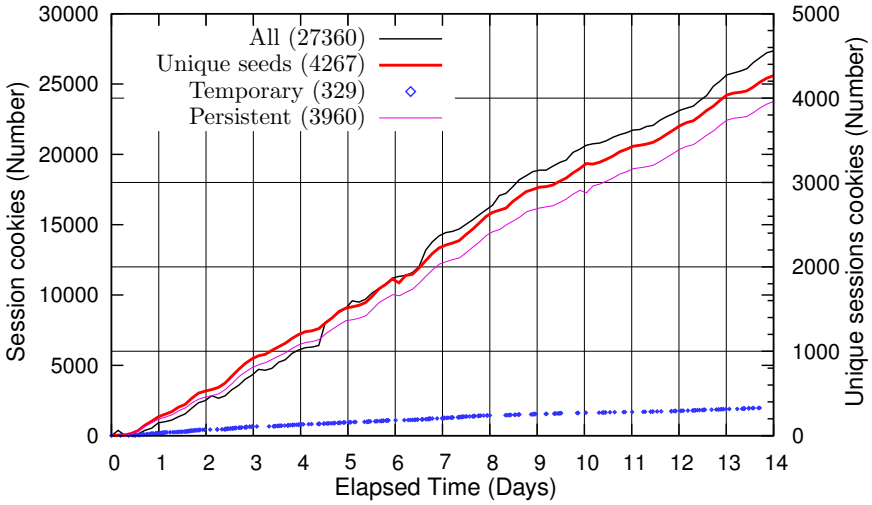

Fig. 3. Number of sessions found through our Tor exit node server within 14 days.

and $\alpha_{k\geq 300} = 3.38$. With this specific information it was possible to generate a accurate power law degree sequence for the two intervals $[1; 300[$ and $[300; 5000]$. We generated a model with a total amount of $1 \cdot 10^4$ nodes and computed C for each of the two intervals with formula 1.

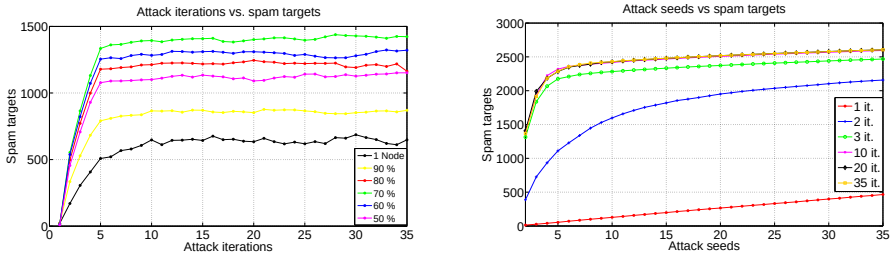

Fig. 4. Results of the FITM attack simulation. (Left) Strategy 1: Spam targets vs. Attack iterations. (Right) Strategy 2: Spam targets vs. Attack seeds (jumps).

Strategy 1. We ran our attack model for *strategy 1* for each number of iterations, from $it = 1, ..., 35$, 1000 times. For each iteration step we averaged over the 1000 results and got the mean value for the potential spam targets. For example, for $it = 5$, the mean number of spam targets is $sp = 1178$ corresponding to the red curve. The five colored curves belong to different percentages p of the

Facebook locale		
Language	ISO	%
English (US)	en_US	60.46
English (UK)	en_GB	11.31
Italian	it_IT	6.19
Spanish	es_LA	5.55
Indonesian	id_ID	3.21
Simplified Chinese (China)	zh_CN	2.41
French (France)	fr_FR	2.01
Persian	fa_IR	1.28
Traditional Chinese (Taiwan)	zh_TW	1.04
Others	-	6.54

Table 3. Facebook locales of analysed user sessions

above mentioned percentage of the amount of spam targets in each iteration step. The black curve describes the number of spam targets when only propagating 1 friend in each iteration step. In Figure 6.2 we can see, that (corresponding to the black curve) at the beginning ($it = 1, ..., 10$) the number of spammed nodes increases almost linear with a high slope. After that ($it = 11, ..., 35$) the curve slowly levels to $sp \approx 643$. The colored curves in fact behave the same way, but their linear growing area is reduced to 5 attack iterations. The slope is significantly higher for values of p between 50 and 80 percent. All curves nearly level to a final value within the first 5 iteration steps. By this results we also can try to find an optimal ratio between spam targets and propagating percentage. We see that propagating a fixed percentage $100 - p$ of a certain users friends yields to better results than in the case of only propagating 1 friend. One also sees that a too small percentage of spamming targets ($p \leq 60\%$ in our simulation) yields to a decrease of spam targets. The value $p = 70\%$ is in our simulation the best choice for p. The leveling of all 6 curves yields to the assumption that in a highly clustered structure it is not possible, with this strategy, to elude the cluster. That was something we expected, due to our hypothesis that Facebook assembles out of *extended induced subgraphs*. In the figure we limited the number of iterations to 35 because the slope for larger iterations converges to 0. Hence, we will get not significantly more spammed nodes even if we increase the number of iterations.

Strategy 2. We ran our attack model for *strategy 2* for each number of jumps, from $jp = 1, ..., 35$, 1000 times. Note that before jumping to another seed the attack strategy executes a specified number of iterations. In our Figure (6.2) the number of iterations would be $it = 1, 2, 3, 10, 20, 35$. *Strategy 2* can be seen as an extended algorithm of *strategy 1*. For each jumping step and number of iterations we averaged over the 1000 results and got the mean value for the spammed nodes, e.g. for $jp = 10,\ it = 3$ the mean number of spammed nodes is

$sp = 1260$. For this strategy we used the optimized value $p = 70\%$, determined in the above simulation. In Figure 6.2 we see that for one iteration ($it = 1$) the slope is almost linear. It represents the fact that in a large network we get almost the same number of *new attacked nodes* for each jump, hence the linear slope. We also see that there is an ample difference in the number of *reached* nodes between $it = 1$ and $it = 2$. This difference decreases with increasing iteration steps. While there is still a considerable difference between $it = 2$ and $it = 3$ and $it = 10$, there is almost none between $it = 20$ and $it = 35$. As in strategy 1, the beginning of our simulation ($jp = 1, ..., 10$) shows a rapid increase of spammed nodes. After $jp = 10$ the slope decreases. It seems that all curves converges to a limit of 3000 spammed nodes. This is a result of the fact that even with 35 iterations and jumping 35 times randomly in the network, the chance of finding "new" *unattacked* areas is negligible. There might be some attacks which are significantly more successful than spamming 3000 nodes, but overall it is to be expected to get something between 2000 and 3000 nodes, depending on how many jumps one executes. The fascinating thing is, that even with a moderate rate of iterations, say $it = 10$ per jump, we get almost the same amount of spammed nodes when executing $it = 35$ iterations per jump. That's why strategy 2 is not only better in performance but also more inconspicuously, as it is possible to get even more spammed nodes while not resting too long in one area or cluster.

6.3 Discussion

The FITM attack could be extremely effective on a large scale, as our results suggest. In a relatively short amount of time, an attacker would be able to collect information from thousands of users in an automated fashion, resulting in tens of thousands possible victims for context-aware spam. Even without using the social network itself, the adversary could achieve a high degree of success by using out-of-band attack vectors like email. The information could be further used for cloning user profiles to other SNSs and increasing the number of spam targets even more, as described in [8].

As simulation 2 shows with merely five different attack seeds and three iterations of our attack over 2000 email addresses for spamming as well as context-information on the spam targets could be collected. In case of the Tor exit node server we ran, these 5 attack seeds would have been collected in less than 25 minutes.

6.4 Mitigation strategies

In recent years numerous privacy protection schemes have been published with the intention to increase the privacy in currently deployed social networking services. The first class of mitigation strategies we discuss are privacy enhancing extensions to SNSs, which do not protect the network layer but may limit the amount of information that can be extracted from user profiles.
An interesting approach is the flyByNight application for Facebook [37], which

encrypts messages between users with strong cryptography. It is implemented in JavaScript and enables the secure transmission of messages. However, at the time of writing it seems that it is no longer maintained and became unusable. It furthermore might be unusable on mobile devices due to the limited performance of these devices. In the future the model of flyByNight might get ported to other social networks that support third-party application, as it is bound to Facebook. The major drawback of flyByNight is that it only protects from a curios social networking operator. It still is vulnerable on the network layer to an active attacker who could still perform the FITM attack or by replacing the flyByNight JavaScript routines to defeat the encryption. Another approach was chosen by the designers of FaceCloak [38], which is intended to hide sensitive information of a user by means of encryption and by providing fake information. It is implemented as a Firefox browserextension, and stores sensitive information on a separate server. One shortcoming that FaceCloak has in common with flyByNight is that only text-based content can be protected leaving out e.g. pictures that could be used for context-aware spam and social phishing. Yet another problem might arise when a vast number of users would start to use Facecloak, which would then require a huge infrastructure similar to Facebook's to store the encrypted FaceCloak content.

Hence, to protect against our friend-in-the-middle attack effectifly it would be necessary to secure the network layer between the SNS users and SNS providers. Thus, the second mitigation strategy available to the average user seems to be the use of browser extensions such as ForceHTTPS [29], which attempt to force HTTPS for requests that would have been normally transferred over HTTP. The Tor project announced [9] that the will include a special version of NoScript [45] within the Tor browser bundle which enforces HTTPS for a number of websites including some SNSs. In order to effectively mitigate FITM attacks, SNSs providers have to ultimately ensure that all communication between their users and their platform is done over HTTPS. At the time of writing only XING [3] fully supports HTTPS, which leaves SNS users with browser extensions as the only working mitigation strategy.

7 Conclusion

In this paper, we have introduced the new FITM attack against social networks. By stealing the user's authentication cookie which is transmitted in an unencrypted way, it becomes possible to completely impersonate the user on social networking sites. This can then be used to collect sensitive information in an automated fashion, possibly leading to large campaigns of context-aware spam and social phishing. Both are known to be highly successful in luring their receivers into revealing sensitive information like passwords and could be conducted abusing the social network itself or by using out-of-band communication like email.

Numerous attack vectors could be exploited by an adversary, such as unencrypted wireless networks. The FITM attack itself is applicable to most of the currently deployed SNSs, such as Facebook, Friendster, and Orkut. Our proof-

of-concept showed the possible effectiveness in the case of Facebook. Based on the FITM attack, following subsequent exploits are easily possible (1) Friend injection, (2) Application injection, and (3) Social engineering.

Our results suggest that finding possible FITM attack seeds for spam campaigns is cheap regarding time and hardware resources. We furthermore showed that a large-scale spam campaign on the basis of FITM attacks would have a severe impact on basis of a simulation. There a number of limited protection strategies available to social networking users. Hence social networking providers have to ultimately protect their users against FITM attacks by securing the communication with HTTPS. As there is no (monetary) incentive for them to do so, we believe that our attack remains applicable for the time being.

References

1. dpkt - python packet creation / parsing library. `http://code.google.com/p/dpkt/`.
2. Python mechanize library. `http://wwwsearch.sourceforge.net/mechanize/`.
3. Xing business network - social network for business professionals. `https://www.xing.com/`.
4. R. Albert and A. L. Barabási. Statistical mechanics of complex networks. *Review of Modern Physics*, 74:47–97, 2002.
5. L. A. N. Amaral, A. Scala, M. Barthélémy, and H. E. Stanley. Classes of small-world networks. In *Proceedings Of The National Academy Of Sciences Of The United States Of America*, volume 97, pages 11149–11152. National Acad Sciences, 2000.
6. A. Barabási and E. Bonabeau. Scale-free networks. *Scientific American*, 288:50–59, 2003.
7. A. L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286:509–512, 1999.
8. L. Bilge, T. Strufe, D. Balzarotti, and E. Kirda. All your contacts are belong to us: Automated identity theft attacks on social networks. In *18th International World Wide Web Conference*, April 2009.
9. T. T. Blog. Tor Browser Bundle for GNU/Linux. online, 2010. [Retrieved 2010-04-10].
10. B. Bollobás. *Random Graphs*. Cambridge University Press, 2nd edition, 2001.
11. J. Bonneau, J. Anderson, R. Anderson, and F. Stajano. Eight friends are enough: social graph approximation via public listings. In *Proceedings of the Second ACM EuroSys Workshop on Social Network Systems*, pages 13–18. ACM, 2009.
12. G. Brown, T. Howe, M. Ihbe, A. Prakash, and K. Borders. Social networks and context-aware spam. In *Proceedings of the ACM 2008 conference on Computer supported cooperative work*, pages 403–412. ACM New York, NY, USA, 2008.
13. R. Dingledine, N. Mathewson, and P. Syverson. Tor: The second-generation onion router. In *13th USENIX Security Symposium*, San Diego, CA, USA, August 2004.
14. P. Erdos and A. Rényi. On random graphs. *Publicationes Mathematicae*, 6:290–297, 1959.
15. Facebook. Facebook statistics, 2010. [Online; accessed 1-March-2010], `http://www.facebook.com/press/info.php?statistics`.
16. B. J. Fogg and D. Iizawa. Online Persuasion in Facebook and Mixi: A Cross-Cultural Comparison. In *PERSUASIVE*, pages 35–46, 2008.

17. L. C. Freeman. *The Development of Social Network Analysis: A Study in the Sociology of Science*. Empirical Press, 2004.
18. Friendster. About Friendster, 2010. [Online; accessed 20-January-2010], `http://www.friendster.com/info/index.php`.
19. Facebooks user.getinfo() documentation. `http://wiki.developers.facebook.com/index.php/Users.getInfo`.
20. Facebooks user.getinfo() documentation. `http://wiki.developers.facebook.com/index.php/Extended_permissions`.
21. M. Gjoka, M. Kurant, C. Butts, and A. Markopoulou. Walking in facebook: A case study of unbiased sampling of osns. to appear in Proc. of Infocom '10, June 2010.
22. R. Gross and A. Acquisti. Information revelation and privacy in online social networks (the Facebook case). In *Proceedings of the 2005 ACM workshop on Privacy in the electronic society*, pages 71–80, 2005.
23. M. S. Handcock, A. R. Raftery, and J. M. Tantrum. Model-based clustering for social networks. *Journal of the Royal Statistical Society*, 170:301–354, 2007.
24. X. He. A Performance Analysis of Secure HTTP Protocol. *STAR Lab Technical Report, Department of Electrical and Computer Engineering, Tennessee Tech University*, 2003.
25. C. Herley and D. Florêncio. A profitless endeavor: phishing as tragedy of the commons. In *NSPW '08: Proceedings of the 2008 workshop on New security paradigms*, pages 59–70, New York, NY, USA, 2008. ACM.
26. G. Hogben. Security Issues and Recommendations for Online Social Networks. *Position Paper. ENISA, European Network and Information Security Agency*, 2007.
27. T. Holz, C. Gorecki, K. Rieck, and F. Freiling. Measuring and detecting fast-flux service networks. In *Symposium on Network and Distributed System Security*. Citeseer, 2008.
28. M. Huber, S. Kowalski, M. Nohlberg, and S. Tjoa. Towards automating social engineering using social networking sites. *Computational Science and Engineering, IEEE International Conference on*, 3:117–124, 2009.
29. C. Jackson and A. Barth. ForceHTTPS: Protecting high-security web sites from network attacks. 2008.
30. T. Jagatic, N. Johnson, M. Jakobsson, and F. Menczer. Social phishing. *Communications of the ACM*, 50(10):94–100, 2007.
31. S. Jansen, T. Luczak, and A. Rucinski. *Random Graphs*. John Wiley & sons, Inc., 2000.
32. H. Jones and J. Soltren. Facebook: Threats to Privacy. *Project MAC: MIT Project on Mathematics and Computing*, 2005.
33. C. Kanich, C. Kreibich, K. Levchenko, B. Enright, G. Voelker, V. Paxson, and S. Savage. Spamalytics: An empirical analysis of spam marketing conversion. In *Proceedings of the 15th ACM conference on Computer and communications security*, pages 3–14. ACM New York, NY, USA, 2008.
34. A. Korolova, R. Motwani, S. U. Nabar, and Y. Xu. Link privacy in social networks. In *Proceeding of the 17th ACM conference on Information and knowledge management*, pages 289–298. ACM New York, NY, USA, 2008.
35. C. Kreibich, C. Kanich, K. Levchenko, B. Enright, G. Voelker, V. Paxson, and S. Savage. On the spam campaign trail. In *First USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET'08)*, 2008.
36. LinkedIn. LinkedIn About us, 2010. [Online; accessed 10-February-2010], `http://press.linkedin.com/about`.

37. M. Lucas and N. Borisov. flybynight: Mitigating the privacy risks of social networking. In *Proceedings of the 7th ACM workshop on Privacy in the electronic society*, pages 1–8. ACM, 2008.
38. W. Luo, Q. Xie, and U. Hengartner. FaceCloak: An Architecture for User Privacy on Social Networking Sites. In *Computational Science and Engineering, 2009. CSE'09. International Conference on*, volume 3, 2009.
39. S. Milgram. The small world problem. *Psychology Today*, 2:60–67, 1967.
40. A. Nazir, S. Raza, and C.-N. Chuah. Unveiling facebook: a measurement study of social network based applications. In *IMC '08: Proceedings of the 8th ACM SIGCOMM conference on Internet measurement*, pages 43–56, New York, NY, USA, 2008. ACM.
41. H. Networks. Hi5 Press, 2010. [Online; accessed 20-February-2010], `http://www.hi5networks.com/press/index.html`.
42. M. Newman. The structure and function of complex networks. *SIAM Review*, 45:167–256, 2003.
43. M. Newman. Power laws, pareto distributions and zipf's law. *Contemporary Physics*, 46(5):323–351, September 2005.
44. M. E. J. Newman. Models of the small world. *Journal of Statistical Physics*, 101(3-4):819–841, November 2000.
45. Noscript browser extension. `http://noscript.net/`.
46. A. Rapoport. Contribution to the theory of random and biased nets. *Bulletin of Mathematical Biophysics*, 19:257–277, 1957.
47. Spamassassin website. `http://spamassassin.org/`.
48. Spamhaus project. `https://www.spamhaus.org/`.
49. H. Stern. A survey of modern spam tools. *pro eedings of the Tth gonferen e on im il nd entiE pm@ gie A*, 2008.
50. I. Times. Google unveils new look for Orkut, 2010. [Online; accessed 20-February-2010], `http://economictimes.indiatimes.com/Google-unveils-new-look-for-Orkut/articleshow/5181314.cms`.
51. Tor: anonymity online. `https://www.torproject.org/`.
52. R. van der Hofstad. *Random Graphs and Complex Networks*. Lecture Notes, Departement of Mathematics and Computer Science, Eindhofen University of Technology, 2009.
53. D. J. Watts and S. H. Strogatz. Collective dynamics of "small-world" networks. *Nature*, 393:440–442, 1998.
54. M. Wong and W. Schlitt. Sender Policy Framework (SPF) for Authorizing Use of Domains in E-Mail,. RFC 4408, Apr. 2006.