iFriendU: Leveraging 3-Cliques to Enhance Infiltration Attacks in Online Social Networks

Rahul Potharaju†, Bogdan Carbunar‡, Cristina Nita-Rotaru†
† Department of Computer Science, Purdue University, IN, USA
‡ Pervasive Platforms and Architectures, Motorola Labs, IL, USA
Email: rpothara@purdue.edu, carbunar@motorola.com, crisn@cs.purdue.edu

1 Introduction

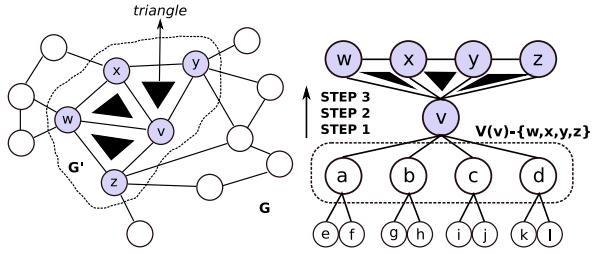
Online Social Networks (OSNs) such as Facebook have become ubiquitous in the past few years, counting hundreds of millions of people as members. OSNs allow users to form friendship relationships, join groups, communicate and share information with friends. The tremendous popularity of OSNs has naturally made them an appealing target for privacy compromising attacks. In this abstract we propose a novel attack against tightly knit OSN communities. Such (artificial) communities consist of users that know well each other and that are reluctant to accept other users as friends. Becoming a member of such a community may be only a first milestone for the attacker. Harvesting private information of members of such communities and following up with offline attacks may be the longer term benefit.

In a naïve approach, the attacker sends random friend invitations to users in the target community "*hoping*" that some of them will accept the request. However, by definition such communities are difficult to infiltrate using a direct invitation based approach. The attack we propose relies on a novel technique, which makes use of *3-cliques* [1, 2] to find the most vulnerable member of a targeted community. The attacker then sends invitations to all the friends of this member. After befriending its friends, the attacker's chances of befriending the weakest community member increase. Then, the attacker not only gains initial access to the community, but also increases its chances of befriending other, less accessible members. Our experiments, performed on a real-world social network, show that our attack can be 75% more efficient than the naïve attack. Using real social network data, we also propose and evaluate a solution that mitigates the problem.

2 Definitions

We define a social network to be an undirected, simple graph $G = (V, E)$ with a set of nodes V denoting users and a set of edges E denoting friendship relations between users. Let n denote the number of nodes and m denotes the number of edges. We use *degree* $d(v) := |u \in V : \exists \{v, u\} \in E|$ for a user v to denote the number of users in V that are friends of v. A *friendship 3-clique* $\Delta = (V_\Delta, E_\Delta)$ of a graph $G = (V, E)$ is a three node subgraph such that $V_\Delta = \{u, v, w\} \subset V$ and $E_\Delta = \{\{u, v\}, \{v, w\}, \{w, u\}\} \subset E$. Also let $\delta(v)$ denote the number of friendship 3-cliques of user v.

The quality of a friendship relationship between two users A and B, $\chi_{A,B}$, can range between acquaintance to close friendship. We propose to compute $\chi_{A,B}$, based on data publicly available: $\chi_{A,B} = \frac{|F_A \cap F_B|}{|F_A|}$ where F_A denotes the set of friends of user A and $|F_A|$ denotes the number of friends of A. We define then the *social closeness* between users A and B to be $\Psi_{AB} = \frac{\chi_{A,B} + \chi_{B,A}}{2}$. Intuitively, this is taking into account two factors: what proportion of A's friends are also B's friends and what proportion of B's friends are also A's friends.



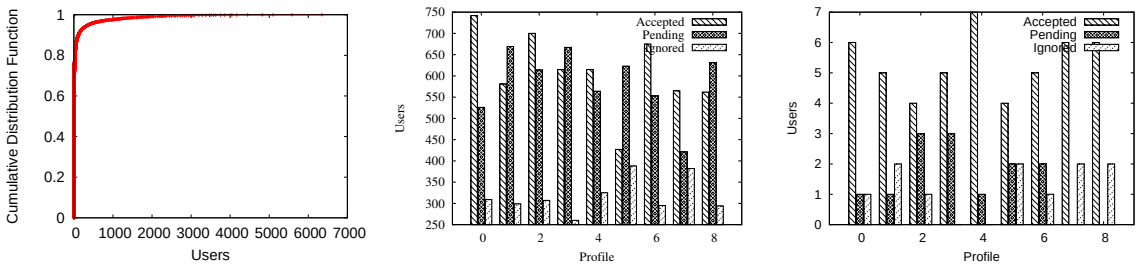
(a) (b)

Figure 1: (a) **Infiltrating a network**: Node v has the highest $\delta(v)$ in G', *i.e.*, the attacker's payoff is higher if it establishes a link with v. (b) Method for establishing a link with v (highest Δ value) - links are established with v's network and finally with its clique members.

3 Attack Description

We are now ready to describe our attack, *iFriendU*. Let $G' \in G$ denote the target community, a subset of the OSN. For each user $v \in G'$, the attacker computes $\delta(v)$, then picks the user with the highest δ value (refer to Alg. 1). A member with a high $\delta(v)$ is socially tied to a higher number of groups so establishing a link with this member is of high value to the attacker because he can earn the trust of more 3-cliques that the member is participating in.

The crux of our attack lies in the observations we made during our initial experimentation: a user may be more willing to accept an invitation if it shares a certain number of mutual friends with the inviter. Therefore, to establish a link with a member having a high $\delta(v)$, the attacker has to first establish links with $v's$ friends. Note that the same reasoning can be applied to $v's$ friends: having more links with the friends of $v's$ friends will help establish links with $v's$ friends. This is a recursive pattern and depending on the payback of infiltration, the depth can be set to a higher value. In our current attack, we set this value to 2 *i.e.*, to establish a link with v, the attacker has the patience and resources to establish links with the friends of $v's$ friends. The



(a) (b) (c)

Figure 2: (a) *3-clique* distribution for 339K links between 178K users (b) Establishing links with friends of 72 users using 9 fake accounts, (c) Establishing links with the 72 users

steps taken then are depicted in Fig. 1(b). In Step 1, the attacker's primary goal is to establish links with v's friends (excluding those already belonging to G'). Thus, he starts off by sending invitations to the friends of $v's$ friends and then $v's$ friends. In Step 2, the attacker invites v and finally in Step 3, it repeats this process, to establish links with the remaining clique members of v.

Algorithm 1 iFriendU's Enhanced Infiltration

```
1. G' = Sample(G)
2. Triangles = ListTriangles(G')
3. do
4.     v_max = extract_max(Triangles)
5.     f = V(v_max) − {x ∈ G', x ≠ v_max}
6.     u = getMaxIVS(f)
7.     RequestOrder = Sort(x ∈ V(u), Ψ_{x v_max})
8.     foreach node p in RequestOrder :
9.         Send request to p
10.    endfor
11.    FinalRequestOrder = Sort(x ∈ {w, x, y, z}, Ψ_{x v_max})
12.    Send requests according to FinalRequestOrder
13. while(len(Triangles) > 0)

14. Procedure ListTriangles(G' = (V, E)) :
15. Construct Adj(v) ∀ v ∈ V
16. foreach edge e in E :
17.    //e consists of v1, v2
18.    v_min = min{|v1|, |v2|}
19.    foreach node x in Adj(v_min) :
20.        if(x ∈ Adj(e − v_min))
21.            store {x, v1, v2}
22.    endfor
23. endfor
```

4 Infiltration Evaluation

We have investigated the effectiveness of iFriendU by conducting several experiments with Facebook. Specifically, we captured a relationship graph with 339K edges (friendships) between 178K nodes (users). We implemented an efficient version of Alg. 1 that found 679K 3-cliques in roughly under five minutes (had a similar performance on a dataset from [3]). Fig. 2(a) shows the 3-clique distribution for the nodes in the network. Note that the large number of 3-cliques follows immediately from the small world property of OSNs [2]. To test our attack, we randomly picked 72 profiles that reflected the profiles of an average user (who typically has 130 friends according to data from Facebook), such that every user is at most two hops away from every other user. Our G' then is the total number of 3-clique friends for these 72 users which totaled to 150. Our goal is to establish as many links as possible with these 150 users. We performed the following two experiments:

Experiment 1 - Naive Approach: Using a fake account, we sent invitations to the 150 users directly. 67 users accepted the invitation giving us an acceptance rate of *45%*.

Experiment 2 - iFriendU Approach: We associated one fake account to a set of 8 users out of the 72 users. Thus, we needed 9 fake accounts in total. In the first phase, each fake account was used to send invitations to the friends of each of the 8 users assigned to it (following the constraints outlined in Alg. 1). Fig. 2(b) shows the result of this phase, where for instance, the first set of bars indicate that roughly 750 users (out of 1550 users in total for one set of 8 users) accepted invitations from the first fake account. In the second phase, the 9 fake accounts attempted to befriend the 72 users. Fig. 2(c) shows the results of this intermediate phase where 48 out of the 72 established links with the 9 fake accounts. Note that this high number is due to the presence of a significant number of mutual friends. In addition, 12 users are yet to decide giving us an actual success rate (excluding the 12) of *80%* and an overall success rate (including the 12) of *66.55%*. As a final step, we befriended the 3-clique members of these 72 members (*i.e.*, the 150 members). The acceptance rate was *79%*.

Our approach clearly outperforms the naïve approach and is 75% more efficient. Note that we used the most basic form of attack *i.e.*, our fake accounts were anonymous profiles having some random names with no profile picture. By forging real profiles, an attacker can achieve a higher degree of success in establishing links with the 3-clique members.

5 Mitigation Approach

The solution we present in this section is part of our system called MORPH-x, which is designed to defend a user against the infiltration attack presented previously in addition to a variety of other attacks not discussed in this abstract. The goal of MORPH-x is to protect rational users, that lack the tools or time to analyze each decision against cyber stalkers, while minimally impacting the experience of honest social network users. Due to space limitations we only present a brief overview of our system internals.

Figure 4 shows the architecture of MORPH-x with its primary components: (i) client-side component (implemented as a Facebook application) that runs on a user's account and collects relevant data, (ii) a Firefox extension that runs inside the user's browser where it monitors its activity and intercepts (accepted and pending) friend requests and (iii) a server component that uses the information collected to detect safe inviters and report them back to the clients. MORPH-x works by inferring trust information between users and their friends. In our current design, we use $\chi_{A,B}$ defined in Section 2 to calculate the inferred trust between users A and B.

After installing the MORPH-x client, a user's newly accepted friends are confined to a probation list, where they

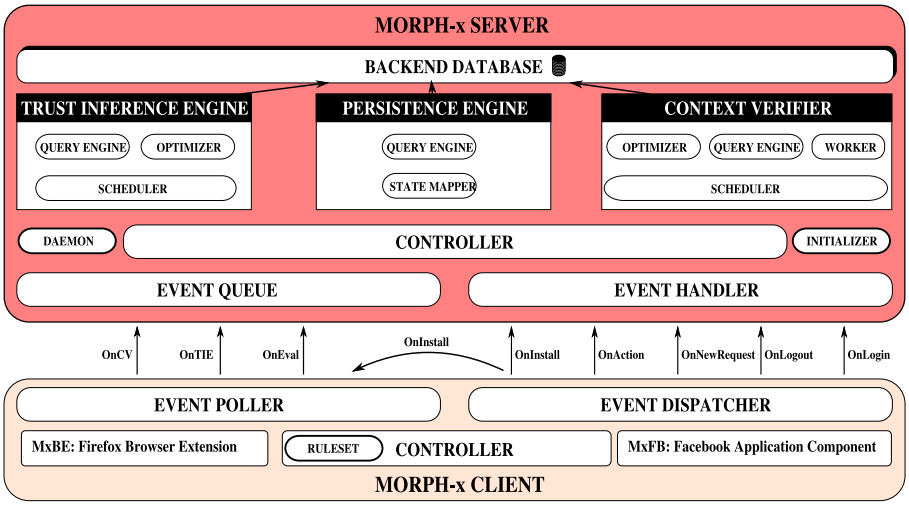
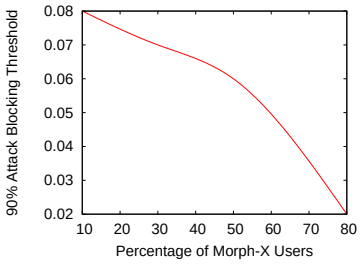


Figure 3: MORPH-x Architecture

are not provided access to the user's profile. A probation friend is promoted to a full friend status (and given profile access) based on the trust values previously built. That is, the promotion occurs only when the MORPH-x server is able to infer a trust value for the probation friend exceeding a given threshold value, *TR-Thr*.

6 Experimental Results

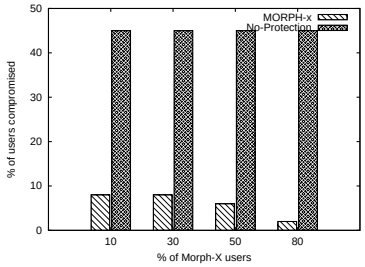
We have evaluated the ability of MORPH-x to block the attack using data collected from Facebook. While the attacks were evaluated in the real world settings, evaluating the defense requires the system to be under attack so we adopted a hybrid approach - we built a Java based simulator that uses the graph (with 179000 users and 380000 relationship edges) we previously crawled to evaluate the success rate of an attacker.



(a)

Figure 4: Evolution of the TR-Thr for MORPH-x users to block 90% of the attacks for various MORPH-x user densities. The decrease is sub-linear, with TR-Thr as small as 0.02 when 80% of the users run MORPH-x.

Figure 4 shows the evolution of the TR-Thr threshold that blocks 90% of the attacks, when launched against densities of MORPH-x users ranging from 10% to 80%. Note that when 80% of users run MORPH-x, a *TR-Thr* threshold value as small as 0.02 is sufficient to block 80% of the attacks. This is because higher densities of MORPH-x users among a user's friends are more efficient in blocking attacks. Note that MORPH-x users can detect which of their neighbors are also running MORPH-x. Then, knowledge of the density of MORPH-x users in its vicinity can be used to locally set a threshold value that provides a user desired level of protection against attacks. Finally, Figure 5 compares MORPH-x users (to be less conservative, we set TR-Thr to be 0.08) with users having no protection. Observe that our defense is clearly efficient: only up to 7% of the attacks succeed. This comes in contrast with users having no protection where around 45% of the attacks succeed. Note that as the MORPH-x user density increases, the attack success rate decreases significantly: when 80% of the users run MORPH-x, only 2% of the attacks succeed.



(a)

Figure 5: Comparison of the effectiveness of MORPH-x with non MORPH-x users in blocking the attack.

7 Conclusion and Future Work

In this paper we study an innovative attack based on 3-cliques in a real world social network setting. Through an extensive implementation we show that such an attack is 75% more efficient than a naïve approach and can be very valuable to spammers and phishers. We design and implement MORPH-x, a solution for mitigating this attack which we evaluated using a real data set. The experimental results demonstrate that our solution is clearly efficient in blocking the proposed attack.

References

[1] N. Alon, R. Yuster, and U. Zwick. Finding and counting given length cycles. *Algorithmica*, 17(3):209–223, 1997.

[2] D. Watts and S. Strogatz. Small world. *Nature*, 393:440–442, 1998.

[3] B. Viswanath, A. Mislove, M.Cha and K.P. Gummadi. On the Evolution of User Interaction in Facebook. *WSON*, 2009.