Thank You For Being A Friend: An Attacker View on Online-Social-Network-based Sybil Defenses

David Koll*, Martin Schwarzmaier*, Jun Li†, Xiang-Yang Li‡ and Xiaoming Fu*

*University of Goettingen, Germany
{dkoll,xfu}@cs.uni-goettingen.de, m.schwarzmaier@stud.uni-goettingen.de

†University of Oregon, USA
lijun@cs.uoregon.edu

‡University of Science and Technology of China, China
xiangyangli@ustc.edu.cn

***Abstract*—Online Social Networks (OSNs) have become a rewarding target for attackers. One particularly popular attack is the Sybil attack, in which the adversary creates many fake accounts called Sybils in order to, for instance, distribute spam or manipulate voting results. A first generation of defense systems tried to detect these Sybils by analyzing changes in the structure of the OSN graph—unfortunately with limited success. Based on these efforts a second generation of solutions enriches the graph-structural approaches with higher-level user features in order to detect Sybil nodes more efficiently. In this work we provide an in-depth analysis of these defenses. We describe their common design and working principles, analyze their vulnerabilities, and design simple yet effective attack strategies that an adversary could launch to circumvent these systems. In our evaluation we reveal that an miscreant can exploit the credulity of OSN users and follow a targeted attack strategy to successfully avoid detection by all existing approaches.**

I. Introduction

Due to their unprecedented success Online Social Networks (OSNs) have become a popular target for attackers, who seek to, for instance, crawl user data, manipulate voting schemes, or distribute spam, malware and fake news articles. For instance, the Locky ransomware was distributed in a spam campaign via the Facebook messenger application in late 2016 [1]. Regardless of their goal, adversaries often instrumentalize a multitude of *Sybil* (or fake) nodes to enable interaction with their targeted audience. These Sybils can easily be created or acquired from specialized underground markets for as little as $0.02 to $0.10 per node [2]. In this context, researchers have recently revealed a 350,000 node strong Twitter botnet [3].

As a consequence, detecting Sybil nodes in OSNs is a major challenge that has been addressed by a variety of solutions. A first generation of approaches tried to exploit solely the structure of the social graph at the core of each OSN (e.g., [4]–[6]). These approaches, however, were built on the over-simplified assumption that attackers can only create few *friendships* with honest users (also called *attack edges*). As soon as an attacker was able to establish a handful of such friendships these approaches were found ineffective [7], [8].

Reseachers have thus subsequently moved away from purely topological approaches in a second generation of Sybil defense solutions. These solutions can be categorized as (i) purely machine-learning (ML) based solutions [9]–[11] and (ii) hybrid solutions [12]–[14]. Here, pure ML systems try to distinguish Sybil accounts from honest users with a ML classifier trained on, for instance, the profile data of users. Unfortunately, trying to detect Sybil nodes solely by employing a ML classifier leads to an arms race between defenders and attackers, as attackers will try to avoid detection by mimicking honest user features [12], [14].

Hybrid approaches thus seem more promising. The key difference with respect to pure ML systems is that hybrid solutions employ feature-based classification *in addition* to graph-based ideas similar to those of first generation systems. Hence, the attacker needs to defy both the ML elements and the topological algorithms. In particular, defying the graph algorithms may be detected by a ML classifier, and vice versa. For instance, even if an attacker might be able to establish many attack edges, she may still experience a suspiciously high amount of rejected friend requests—which can be detected by the classifier. At the same time, trying to please the classifier will result in a small amount of attack edges, which results in a detection by the graph algorithms.

It is however unclear how well these hybrid solutions perform if the adversary attacks more intelligently rather than trying to randomly place attack edges. For instance, in the example above, an attacker could execute a two-staged attack. She could first sacrifice some of her attack power and identify users with a high likelihood of accepting friend requests, and then send friend requests to these users in a second stage with fresh Sybil accounts. Doing so would lead to a high ratio of request acceptance, thus evading the classifier.

In this work we take an attacker's view on hybrid Sybil defense solutions: we exploit observations on user behavior when receiving friend requests in order to find entry points into the OSN [15], [16]. Then, from these entry points, we let the attacker further infiltrate the OSN to extend the reach of the Sybil attack. Ultimately, our goal is to find whether or not current defensive systems can reliably detect Sybil nodes in different attack scenarios. Concretely, our contributions are:

- We thoroughly analyze three state-of-the-art hybrid detection methods [12]–[14]. We dissect their working principles and find that all systems use additional high-level features as input for topological graph algorithms. These algorithms

resemble those of the first generation solutions. In particular, all solutions try to distinguish between Sybils and honest nodes in a similar fashion by propagating a notion of trust through the network and then, based on that trust, apply a ranking scheme that (ideally) should rank all honest nodes higher than all Sybil nodes.
- We find vulnerabilities in all systems and subsequently design five different attack strategies against hybrid defenses. These attack strategies could easily be implemented by an adversary and exploit the curiosity of OSN users that leads to high acceptance rates for friend requests sent by Sybils.
- We implement the detection methods and evaluate their performance under attack. Our results show that even under conservative attack success assumptions, all Sybil defenses suffer from high false positive or false negative rates.

The remainder of this paper is structured as follows: In Sec. II we briefly review related work. We then analyze hybrid Sybil defenses for commonalities in Sec. III. Afterwards, we work out vulnerabilities in Sec. IV and subsequently design attack strategies exploiting these vulnerabilities in Sec. V. We evaluate the performance of Sybil defenses under attack in Sec. VI and finally conclude the paper in Sec. VII.

II. Related Work and Scope

Sybil Defenses: A plethora of approaches to defend an OSN against the Sybil attack has been proposed in the past ten years. A first generation of Sybil defenses approached the problem from a graph-theoretical perspective (e.g., [4]–[6]) by assuming that attackers can hardly befriend honest users and could thus be detected by their isolated position in the social graph. After several studies found this assumption to be over-simplified [7], [8], the second generation of Sybil defenses was built on more sophisticated grounds. These solutions typically add higher-level components (e.g., via machine learning) to the defense scheme [9]–[14]. In this work, we do not propose another defense solution, but rather investigate the state of the art for its success in Sybil defense.

Sybil Defense Surveys: Surveys on Sybil defenses [17]–[19] are intended to give an overview of the state of Sybil defense research and thus do not analyze Sybil defense solutions with respect to their effectiveness. On the contrary, our goal is to thoroughly analyze Sybil defenses in order to evaluate their strength when faced with more sophisticated attackers.

Sybil Defense Analysis: Viswanath et al. were the first to systematically analyze Sybil defenses for their working principles [20]. In a previous work, we have built on their discoveries and analyzed first-generation Sybil defense solutions with regards to their efficiency [7]. In this paper, to the best of our knowledge, we provide the first in-depth analysis of Sybil defenses for second-generation approaches.

Scope of this paper: Second-generation approaches can be divided into solutions (i) purely based on machine learning classifiers and (ii) hybrid approaches, which combine classifiers with graph-theoretical elements. Unfortunately, purely ML-based approaches can be tricked quite easily by the attacker, if she is able to mimic honest user behavior (and thereby features) [12], [14]. Thus, in this paper we focus on the more promising direction of hybrid Sybil defense solutions.

III. Hybrid Sybil-defenses: Overview

In general, defense solutions approach the problem of detecting Sybils in an OSN by investigating the *friendships* among the OSN users, as represented in the social graph of the OSN. First-generation approaches assumed that an attacker would be unable to establish a significant amount of friendships with honest users (*attack edges*). As a consequence, starting a *random walk* in the region of the social graph containing the Sybil accounts would only occasionally find a path into the connected region of honest nodes, and vice versa. In this way, the common abstraction of first-generation defensive solutions was to *rank* a suspect in the OSN based on some notion of the probability of a random walk (or variation thereof) starting from a *seed* reaching the suspect. The suspect was then deemed honest (in case of high probability) or Sybil (in case of low probability) [7], [20].

Recently however, Sybils were found to be able to establish a significant amount of friendships with honest users [8]. Thus, starting a random walk at an honest node in fact traverses Sybil nodes more often than tolerable by first-generation Sybil defenses [7], [12]. In order to mitigate this problem a second generation of *hybrid* Sybil defense solutions was proposed recently [12]–[14]. Here, the common assumption is that, while Sybils may be able to befriend honest users, their behavior in doing so can be detected. Thus, the joint principle among all hybrid solutions is to enrich the social graph with higher-level behavioral features that can help to differentiate between both classes of nodes. These features are then used as additional information when traversing the social graph. Here, the common goal is to lower the impact of potential attack edges in the system (e.g., by assigning a low traversal probability to them). Still, the final traversal of the graph results in a ranking of nodes in each solution and, similar to first-generation solutions, the ranking of a particular node decides its label (i.e., Sybil or honest). In the following we describe how each hybrid solution implements this process.

Integro [12]: Before traversing the social graph, Integro applies a ML classifier in order to detect *victims*, i.e., honest nodes that are likely to accept friend requests sent by Sybils. Features contributing to this classification are, e.g., related to the number of friends, or interaction frequency and volume of each user. Integro then uses the result of this classifier as additional information when traversing the social graph. In particular, all weights of the edges adjacent to a victim are reduced drastically in a first step. Then, starting from a seed node and with respect to edge weights, *trust* is propagated through the now modified social graph via power iterations (similar to SybilRank, see [6]). The rationale is that Sybil nodes will achieve lower trust values than honest nodes as their attack edges are typically connected to victims and thus propagate less trust due to their low weight. A nodes's rank is then determined by dividing its final amount of trust by its weight-adjusted degree.

Votetrust [13]: Different in its strategy from Integro, Votetrust tries to directly classify Sybils. Votetrust is based on the assumption that an attacker can often create many friendships with honest users but can be expected to gather an unusual amount of rejections in the process. Thus, the proportion of rejections is used as the main additional information to later modify the social graph. Votetrust then proceeds in two phases: First a limited amount of *votes* (a measure similar to trust in Integro) is propagated from seed nodes through the original network. Subsequently, each successful or unsuccessful friendship request results in a positive or negative vote to be cast onto the requester, weighted by the amount of votes available at the target of the request. Here, the weighting operation prevents colluding Sybils from up-voting each other without limits. A node's rank is then determined by the total of each node's weighted votes.

SybilFrame [14]: SybilFrame employs two ML classifiers to obtain prior information before traversing the graph. Here, one classifier is giving an estimate (the *node prior*) how likely each node is to be Sybil based on features like the clustering coefficient or the ratio of accepted incoming friend requests. The other classifier yields an estimate (the *edge prior*) of how likely each pair of adjacent nodes is equal (i.e., both nodes are Sybil or both nodes are honest) based on similarity measures like the Jaccard Index. This information is then used to modify each node (Sybil likelihood) and edge (equality likelihood) in the social graph. Afterwards, SybilFrame determines the rank of a node as the probability of each node to be Sybil by using Loopy Belief Propagation (LBP) [21] as message passing algorithm. LBP works by collecting for each node u its neighbors' belief. That is, each node v adjacent to u estimates whether u is honest or a Sybil. This knowledge is then integrated with u's node prior to reason on its label. Similarly, u sends messages to each neighbor v by combining the information of u's label with the edge prior of the edge (u, v). This process is repeated several times and results in a ranking of nodes indicating their likelihood of being Sybil.

In summary, as indicated above, all three hybrid solutions make use of additional information when trying to detect Sybil nodes with graph traversal techniques. While Votetrust and SybilFrame use this information to directly detect Sybils, Integro follows a different path and rather tries to predict the victims of a Sybil attack. After modifying the social graph with additional information (e.g., changing edge weights), all three solutions then propagate some notion of trust through the network. Although they differ in the details on how this trust is distributed (e.g., power iterations or vote propagation), all three solutions finally produce a ranking that should ideally rank all honest nodes higher than all Sybil nodes.

IV. Vulnerability Analysis

In the following, we investigate each of the solutions for possible entry points for an attacker. Here, we focus on conceptual issues rather than implementation details.

Integro: While Integro's low weighted victim edges address the problem of large and dense sybil regions the system is still susceptible to isolated Sybils with many attack edges, which is the prevalent attack pattern observed in literature [8]. Since each node's degree is computed as the sum of its incident edges' weights the effect of low weight edges in terms of their ability to conduct trust is counteracted when the nodes' trust is divided by their degree to determine their rank.

Votetrust: Votetrust's approach of reasoning on successful and rejected requests can be assumed to reliably detect unprepared attackers even when Sybil nodes are isolated and have many attack edges. However, we have identified two potential vulnerabilities. First, colluding Sybils might be able to acquire large amounts of votes by tricking a few honest nodes into sending a request to a Sybil node, which can easily be achieved [22]. These votes can then be used to up-vote other Sybils. Second, by sending requests to friends of already established victims, a Sybil node can improve the probability of its requests to be successful increasing its chance to a high rank. This attack option is based on the acceptance probability of a friend request drastically improving with an increasing number of mutual friends between the requester and her target [15]. In particular, while a request is successful in with probability $p = 0.2$ in cases without mutual friends, an attacker can more than double this probability with two mutual friends, and can overall increase the probability to $p = 0.7$. The adversary can also further improve her friend request acceptance ratio by first sacrificing some attack power (i.e., Sybil accounts) to identify accepting users, and then use the majority of her attack power to infiltrate these users' friends.

SybilFrame: Using its probability inference system SybilFrame is able to counteract a classifier's original assessment of a node by weighing in the beliefs of a node's neighbors regarding the node's label. However, its performance is fully dependent on its two classifiers predicting the nodes' and edges' labels based on features that are at least partially controlled by the attacker. In particular, these classifiers have already produced a 32% FN ratio (node label classifier) and a 80% FN ratio (edge label classifier) in [14]. Taken together, these misclassifications can result in a high FN ratio for Sybil detection, as the beliefs exchanged via LBP become inaccurate and the distinguishing ability of SybilFrame suffers. Consider for instance an isolated Sybil u that has managed to establish five attack edges. Then, this Sybil will trick the classifier in certain features (e.g., clustering coefficient, as it is not connected to a Sybil region) and thus has a good chance of being misclassified. Also, on average, four of its five attack edges could be classified as normal edges, again leading to a network belief of u being honest.

V. Attack Strategies

While OSN infiltration attacks can be arbitrarily complex, in this work we focus on attacks that are easy to implement. We define an attack strategy as the way an adversary (i) organizes Sybil nodes and (ii) tries to establish attack edges.

A. Organization of Sybils

Links between Sybil nodes can be set freely by the adversary. This is typically done one of two ways [7].

Community: An attacker can form a densely connected *Sybil region* (e.g., a complete graph), where fake profiles befriend each other to appear legitimate to honest users.

Peripheral: However, in typical real-world attacks, Sybils rather form the majority of their links with honest users [8], thereby evading the (community-)detection mechanisms of first-generation Sybil defenses [7]. Here, only 25% of a Sybil's links are to another Sybil [8].

B. Attack Edges

The adversary cannot control the placement of attack edges, as friend requests may be rejected. However, the attacker can increase success rates by following a variety of strategies.

Random: The attacker can naively send out friend requests to honest nodes at random.

Targeted: Our core strategy is to initially perform the random strategy until an attack edge to a particular user is established. Afterwards, the attacker sends requests to that user's friends. Here, we exploit the good will of OSN users that results in a higher request acceptance likelihood in the presence of mutual friends [15]. This strategy is then recursively applied in a breadth-first manner, further increasing the chance of acceptance for the attacker. The attack can be effective against both Votetrust and SybilFrame. In the former, it will increase the success ratio for friend requests, and in the latter it will reduce the classifier accuracy. Figure 1 shows a visualization of this attack. Here, the sybil (S) randomly sends out friend requests (the order in which requests are sent is indicated by the numbering of nodes). After she is not successful at node 1, node 2 accepts her request. The attacker thus sends requests to nodes 5 to 7 next, as she expects a higher probability of being accepted there. After recursively traversing the friend list of node 5 (node 4 accepts the request), the attacker continues to randomly send requests (node 3).

Boosted: Third, to undermine Votetrust, the attacker can trick a few honest nodes into sending requests to a small boosting circle consisting of three Sybil nodes. Vote capacity flowing from these honest nodes into the circle will accumulate there. The attacker then—from different nodes—sends friend requests to the boosting circle, all of which are accepted. Based on the voting scheme of Votetrust this results in a fixed amount of positive votes for each attacking node that can be used as a starting capital. The attacker then follows the targeted strategy.

C. Attack Combinations

An attacker can arbitrarily combine the two Sybil organization strategies with the three attack edge strategies. In this paper, we will use the combination of *community and random* as a base benchmark. Afterwards, we will increase the attacker strength by using the combination of *peripheral* with each of the three attack edge strategies.

VI. Evaluation

In order to properly evaluate the effectiveness of our attack strategies, we have implemented all three hybrid Sybil defenses. In this section, we describe our evaluation methodology and present our key results.

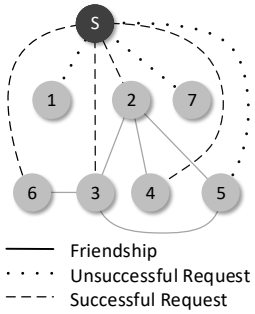

Fig. 1: Illustration of the targeted attack in chronological order. The attacker applies BFS after the successful second request.

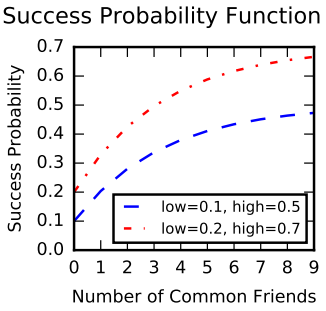

Fig. 2: Functions of friend request acceptance. The more mutual friends, the higher the likelihood of acceptance [15].

A. Methodology

We evaluate Integro, Votetrust and SybilFrame through repeated simulations using two real-world OSN graphs of different sizes from Facebook (4,039 nodes with 88,234 edges, [23]) and Slashdot (77,360 nodes with 905,468 edges [24]). In both graphs, we first add all Sybil nodes and their connections among each other (i.e., Sybil region or peripheral organization). Then, each Sybil node issues friend requests according to the respective attack edge strategy.

The success of establishing an attack edge is influenced by the friend request acceptance probability. While research has shown success probabilities of up to 0.9 for carefully crafted Sybils [16], we take a conservative approach as shown in Figure 2. We follow the model presented in [15] with an initial success probability of 0.2 that increases to up to 0.7 depending on the number of mutual friends between a Sybil and its target. We also evaluate a model in which the initial probability is set to 0.1, and only increases to 0.5.

Finally, as far as possible we have set the system parameters to the values used in [12]–[14]. Here, Integro and SybilFrame depend on ML classifiers based on ground truth data, to which we did not have access. We thus applied the same classification results (in terms of false positives and false negatives) as obtained with the original classifiers for our experiments.

To quantify our results, we use *false positives (FP)*, *false negatives*, and the *Area Under the ROC (AUC)* as our main metrics. The FP rate describes the ratio of honest users falsely accused of being Sybil. A high FP rate is undesirable as it can suspend a large number of honest accounts. The FN rate is a direct indicator of a system's capability to reliably detect Sybil nodes as it describes the ratio of Sybils remaining undetected by the defensive solution. The AUC, on a higher level, describes the overall capability of a system to accurately label nodes with low false positives and false negatives.

B. Results

Figure 3 shows our key results.[1] Here, we evaluate the AUC for all systems as Sybils are sending out an increasing amount

[1]For the sake of clarity we only show the results obtained from the Facebook graph. The Slashdot graph yields very similar results.

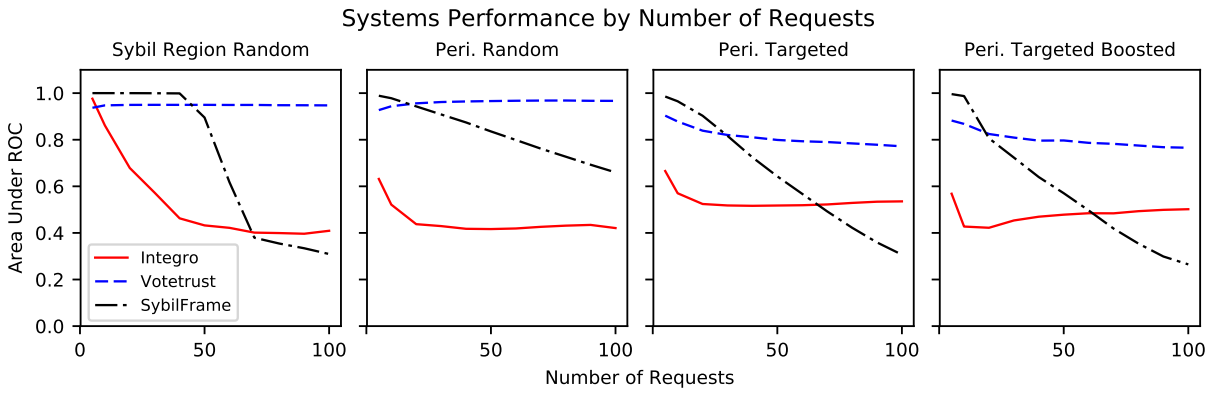

Fig. 3: The Performance of Integro, Votetrust and SybilFrame with regards to the AUC with an increasing number of friend requests sent by attackers.

of friend requests while employing different attack strategies.[2] The attack strength increases from left to right.

Votetrust works consistently well when faced with random friend requests originating from a Sybil region, and Integro works well as long as the number of requests issued per Sybil is below 20. SybilFrame exhibits an interesting behavior in this scenario, deteriorating quickly after performing perfectly for the first 40 requests. Here, the Sybil region acts as a catalyst that amplifies the current classification tendency. Before information flows into the Sybil region from outside the intra-Sybil edges reinforce the belief that all Sybil nodes are in fact Sybil. After enough attack edges have been established and enough messages claiming that the Sybil nodes are honest flow into the region, this belief is overturned. Then, the intra-Sybil edges reinforce the new tendency, resulting in a strong belief that the Sybil nodes take an honest label.

When we do not employ a Sybil region to send random requests but execute a peripheral attack, Votetrust continues to reliably detect Sybils. This is an improvement over first-generation solutions, where two attack edges per node were enough to degrade defense performance [7]. In this scenario, SybilFrame suffers from 20 friend requests onwards as attack edges are more often misclassified as honest. However, in the absence of a Sybil region, this misclassification can not act as a catalyst, leading to a better performance in this case.

However, as we move to the more advanced peripheral attack strategies that target friends of already infiltrated honest users, we observe that all systems consistently fail to reliably detect Sybil nodes. Either FP or FN increase, resulting in a smaller AUC. Note that Votetrust is closest to performing reasonably well (AUC > 0.8 in all cases), but still admits a significant amount of Sybils. Here, when the Sybils are looking for their entry point to the OSN, they also accumulate rejected requests, thereby reducing their rank. Once they find that entry point and are more successful in establishing friendships, Votetrust loses some of its predictive power. The boosting strategy only improves the attack effectiveness by nuances.

In Figure 4, we show the CDFs of normalized ranks

[2]Note that we use friend requests, and not attack edges as parameter, as distinguishing between both is important in, e.g., Votetrust's success ratio feature. The actual number of attack edges is probabilistically determined.

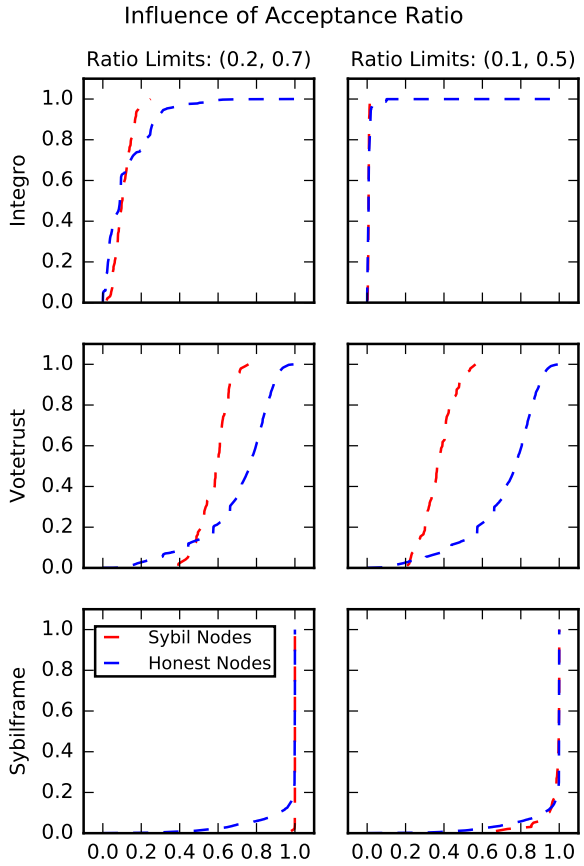

Fig. 4: The influence of the request acceptance ratio under the targeted peripheral attack.

assigned to Sybils and honest nodes, respectively. In an ideal system, the two CDF curves representing both node classes would be separated significantly, indicating a high capability of the systems to distinguish Sybils from honest users. However, in the left column of Figure 4 we observe that the two CDF curves largely overlap for Integro and SybilFrame, which means that both systems rank Sybils and honest nodes similarly. Votetrust performs better, but, if all Sybils should be detected, would also incur a $\approx 30\%$ FP rate.

In the right column of Figure 4 we show the influence of lower acceptance ratio probabilities for friend requests sent

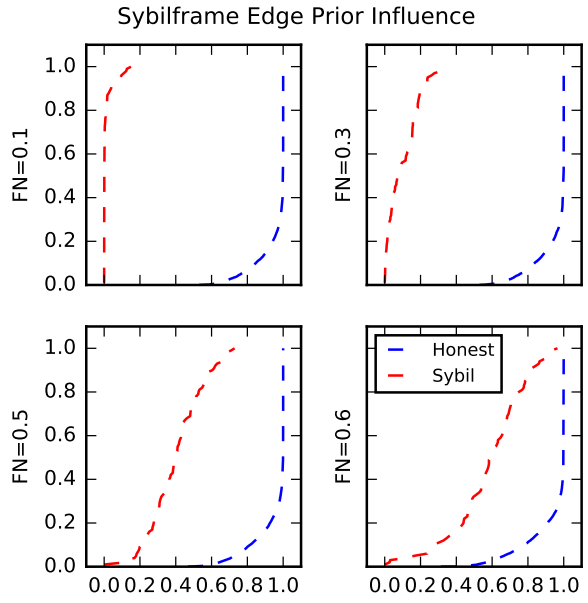

Fig. 5: With decreasing classifier sensitivity, SybilFrame has trouble in distinguishing all Sybils from all honest nodes (peripheral targeted attack). At the same time, a classifier with sensitivity of 90% yields close to perfect results (top left).

out by Sybils (cf. Figure 2). Even with our very conservative probability model, both Integro and SybilFrame do not perform well. Votetrust significantly improves in performance (the curves for Sybil and honest users overlap less), but will still incur a FP rate of approximately 10% if all Sybils would be excluded. On the scale of OSNs with hundreds of millions of users, this would result in a prohibitive manual overhead in verifying and re-establishing the accounts of falsely accused honest users—or worse, a loss of honest users to the OSN.

Finally, in our vulnerability analysis we noted that SybilFrame is dependent on the quality of its classifiers. As we performed our experiments with the same parameters as in [14], we also employed the edge-classifier with 20% sensitivity, i.e., missing out on 80% of the attack edges in the system (FN=0.8). Therefore, in a final step we evaluate the performance of SybilFrame in case a better classifier would be available. Figure 5 shows that the better the classifier, the better the performance of SybilFrame. A sensitivity of 70% for the edge prior classifier (FN=0.3) would indeed help SybilFrame in appropriately detecting Sybil nodes, while any sensitivity below still results in a degraded performance.

VII. Conclusion

In this work we have taken an attacker's view on state-of-the-art hybrid Sybil defense solutions. These solutions make use of higher level features as input for graph-traversal algorithms, which in turn calculate a ranking of nodes in the OSN. We have discussed the vulnerabilities of current approaches and designed a set of rather simple attack strategies that exploit these vulnerabilities and the credulity of OSN users. Evaluating the impact of these attack strategies in our evaluation based on two real-world OSN graphs shows that while hybrid systems are an improvement over first-generation Sybil defenses, they are still not able to reliably detect Sybil nodes. In fact, all approaches induce prohibitive false positive or false negative rates when faced with our attack strategies.

Acknowledgements

This material is partially based upon work supported by the Lindemann Foundation and the National Science Foundation under Grant No. 1564348.

References

[1] S. Khandelwal, "Spammers using facebook messenger to spread locky ransomware," http://thehackernews.com/2016/11/locky-ransomware-facebook.html (retrieved January 9th 2017), November 2016.
[2] K. Thomas, D. McCoy, C. Grier, A. Kolcz, and V. Paxson, "Trafficking Fraudulent Accounts: The Role of the Underground Market in Twitter Spam and Abuse," in *USENIX Security'13*.
[3] J. Echeverria and S. ZhouGao, "The 'star wars' botnet with >350k twitter bots," *arXiv preprint arXiv:1701.02405*, 2017.
[4] H. Yu, M. Kaminsky, P. B. Gibbons, and A. D. Flaxman, "SybilGuard: Defending Against Sybil Attacks via Social Networks," *IEEE/ACM Trans. Netw.*, vol. 16, no. 3, pp. 576–589, 2008.
[5] N. Tran, B. Min, J. Li, and L. Subramanian, "Sybil-resilient Online Content Voting," in *USENIX NSDI'09*.
[6] Q. Cao, M. Sirivianos, X. Yang, and T. Pregueiro, "Aiding the Detection of Fake Accounts in Large Scale Social Online Services," in *NSDI'12*.
[7] D. Koll, J. Li, J. Stein, and X. Fu, "On the state of osn-based sybil defenses," in *IFIP Networking'14*.
[8] Z. Yang, C. Wilson, X. Wang, T. Gao, B. Y. Zhao, and Y. Dai, "Uncovering social network sybils in the wild," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 8, no. 1, p. 2, 2014.
[9] B. Viswanath, M. A. Bashir, M. Crovella, S. Guha, K. P. Gummadi, B. Krishnamurthy, and A. Mislove, "Towards detecting anomalous user behavior in online social networks," in *USENIX Security'14*.
[10] C. Xiao, D. M. Freeman, and T. Hwa, "Detecting clusters of fake accounts in online social networks," in *ACM AISec'15*.
[11] Q. Cao, X. Yang, J. Yu, and C. Palow, "Uncovering large groups of active malicious accounts in online social networks," in *ACM CCS'14*.
[12] Y. Boshmaf, D. Logothetis, G. Siganos, J. Lería, J. Lorenzo, M. Ripeanu, and K. Beznosov, "Integro: Leveraging victim prediction for robust fake account detection in osns." in *NDSS'15*.
[13] Z. Yang, J. Xue, X. Yang, X. Wang, and Y. Dai, "Votetrust: Leveraging friend invitation graph to defend against social network sybils," *IEEE Transactions on Dependable and Secure Computing*, vol. 13, no. 4, pp. 488–501, July 2016.
[14] P. Gao, N. Z. Gong, S. Kulkarni, K. Thomas, and P. Mittal, "Sybilframe: A defense-in-depth framework for structure-based sybil detection," *arXiv preprint arXiv:1503.02985*, 2015.
[15] Y. Boshmaf, I. Muslukhov, K. Beznosov, and M. Ripeanu, "The Socialbot Network: When Bots Socialize for Fame and Money," in *ACSAC'11*.
[16] L. Bilge, T. Strufe, D. Balzarotti, and E. Kirda, "All Your Contacts are Belong to us: Automated Identity Theft Attacks on Social Networks," in *WWW'09*.
[17] H. Yu, "Sybil Defenses via Social Networks: A Tutorial and Survey," *SIGACT News*, vol. 42, no. 3, pp. 80–101, Oct. 2011.
[18] A. Mohaisen and J. Kim, "The sybil attacks and defenses: a survey," *arXiv preprint arXiv:1312.6349*, 2013.
[19] L. Alvisi, A. Clement, A. Epasto, S. Lattanzi, and A. Panconesi, "SoK: The Evolution of Sybil Defense via Social Networks," in *IEEE S&P'13*.
[20] B. Viswanath, A. Post, K. P. Gummadi, and A. Mislove, "An Analysis of Social Network-based Sybil Defenses," in *ACM SIGCOMM'10*.
[21] K. P. Murphy, Y. Weiss, and M. I. Jordan, "Loopy belief propagation for approximate inference: An empirical study," in *Uncertainty in Artificial Intelligence'99*.
[22] V. Sridharan, S. Vaibhav, and M. Gupta, "Twitter Games: How Successful Spammers Pick Targets," in *ACSAC'12*.
[23] J. McAuley and J. Leskovec, "Learning to discover social circles in ego networks," in *NIPS'12*.
[24] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney, "Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters," *Internet Mathematics*, vol. 6, no. 1, pp. 29–123, 2009.