Exploiting Temporal Dynamics in Sybil Defenses

Changchang Liu[1,2] Peng Gao[1,2] Matthew Wright[3] Prateek Mittal[2]
[1] Equal contribution joint first authors
[2] Department of Electrical Engineering, Princeton University
Email: {cl12, pgao, pmittal}@princeton.edu
[3] Department of Computer Science and Engineering, University of Texas at Arlington
Email: mwright@cse.uta.edu

Abstract

Sybil attacks present a significant threat to many Internet systems and applications, in which a single adversary inserts multiple colluding identities in the system to compromise its security and privacy. Recent work has advocated the use of social-network-based trust relationships to defend against Sybil attacks. However, most of the prior security analyses of such systems examine only the case of social networks at a single instant in time. In practice, social network connections change over time, and attackers can also cause limited changes to the networks. In this work, we focus on the *temporal* dynamics of a variety of social-network-based Sybil defenses. We describe and examine the effect of novel attacks based on: (a) the attacker's ability to modify Sybil-controlled parts of the social-network graph, (b) his ability to change the connections that his Sybil identities maintain to honest users, and (c) taking advantage of the regular dynamics of connections forming and breaking in the honest part of the social network. We find that against some defenses meant to be fully distributed, such as SybilLimit and Persea, the attacker can make dramatic gains over time and greatly undermine the security guarantees of the system. Even against centrally controlled Sybil defenses, the attacker can eventually evade detection (e.g. against SybilInfer and SybilRank) or create denial-of-service conditions (e.g. against Ostra and SumUp). After analysis and simulation of these attacks using both synthetic and real-world social network topologies, we describe possible defense strategies and the trade-offs that should be explored. It is clear from our findings that temporal dynamics need to be accounted for in Sybil defense or else the attacker will be able to undermine the system in unexpected and possibly dangerous ways.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General – Security and Protection; K.4.1 [**Computers and Society**]: Public Policy Issues – Abuse and Crime Involving Computers; K.6.5 [**Management of Computing and Information Systems**]: Security and Protection – Authentication

Keywords

Sybil attacks; temporal dynamics

1. INTRODUCTION

In December of 2011, a political protest by Russian citizens on the Twitter social network was swamped by spam from thousands of bot accounts [14, 22]. Similarly, the Tor anonymity network was infiltrated by a botnet in 2010 that resulted in 25% compromised relays [1, 2], and more recent attacks have been reported in which multiple relays, apparently controlled by a single entity, attempted man-in-the-middle attacks on users [32]. These incidents are examples of *Sybil attacks*, in which a single entity controls many different identities so as to overcome security mechanisms and attack the system and its users [9]. Sybil attacks are a particular concern for distributed systems, which lack a central authority to vet identities and perform admission control. Many of these designs assume a bound on the fraction of malicious identities in the system for correct operations. For example, byzantine consensus protocols, quorum-based systems, reputation systems and distributed hash tables are designed to tolerate only a bounded number of malicious identities. The Twitter example, however, along with many other cases of Sybil identities on online social networks [34, 23], shows that even centralized systems have serious challenges with such attacks.

An important thread of research proposes to defend against Sybil attacks using social-network-based trust relationships [36, 35, 8, 15, 18, 4, 19, 26, 27, 16]. The key insight in these defenses is that it is hard for an adversary to set up trust relationships with honest users (called attack edges), particularly when user interactions are used to infer strong social ties [11, 31] (see Section 2). Using a wide range of mechanisms, a bound on the number of attack edges is translated into a bound on the number of malicious Sybil identities that an adversary can insert in the system. For example, SybilLimit [35], a well-known distributed Sybil defense, offers a proof of security based on this assumption. Given g attack edges between honest users and an adversary, SybilLimit claims that an adversary cannot insert more than $g \cdot w$ Sybil identities, where w is the mixing time of the social network.

CCS'15, October 12–16, 2015, Denver, Colorado, USA.
ACM 978-1-4503-3832-5/15/10.
http://dx.doi.org/10.1145/2810103.2813693.

While most Sybil defense mechanisms in the literature have been analyzed theoretically or using experiments on synthetic or real data, they have all been developed and evaluated from the perspective of a *static* social network, in which the trust relationships are established and unchanging. Social networks, however, are constantly evolving as new relationships are formed and others fade out [13], particularly when using the more dynamic interaction-based networks that are critical for strong Sybil defense [31]. Further, attackers are not limited to attacking a system at a single point in time and may be able to improve an attack's effectiveness with persistent effort. These *temporal dynamics* of Sybil defense systems have received very little attention in the research community, and it is thus unclear whether Sybil defenses provide any security guarantees over time.

Contributions:

In this paper, we begin to address this issue by investigating the temporal dimension of Sybil defense systems and its impact on system security. In particular, we examine the following questions:

- What kinds of temporal dynamics do social networks and Sybil defense systems have?
- What are the possible attacks that leverage these temporal dynamics and how effective are they?
- What defense mechanisms that account for temporal dynamics, or even leverage them, could be used to prevent these attacks?

We identify three main temporal aspects of social networks that are relevant to Sybil defense: (a) churn among the Sybil identities, in which the attacker replaces some identities with others; (b) churn in attack edges, in which the attacker changes which honest users he targets for creating an attack edge; and (c) churn among the honest users in the social network.

We propose a general attack model that leverages these temporal dynamics, and identify a number of possible attacks on previously published Sybil defense systems. We first examine a new attack on SybilLimit [35] based on churn in Sybil identities–the *counter elevation attack*–that greatly reduces the effectiveness of a key defense mechanism, a set of counters used to track who is validating whom. We then examine attacks based on churn in attack edges, including attacks on SybilInfer [8], SybilRank [7], and a powerful attack that undermines the effectiveness of the Persea Sybil-resistant DHT [4]. Third, we examine how the Ostra [16] and SumUp [27] systems can be leveraged by attackers to conduct denial-of-service attacks when the attacker creates churn in the Sybil identities.

Finally, we discuss possible countermeasures to these attacks. While careful design may overcome some of the issues that we explore in this paper, developing theoretically backed defenses that explicitly address our temporal attacks remains an important open problem for future work.

2. BACKGROUND AND RELATED WORK

2.1 Sybil Attack Problem

Consider a social network topology $G = (V, E)$, comprising a set V of nodes with a set E of edges. In the network topology, a node $v \in V$ denotes an identity, and an edge $(u, v) \in E$ denotes a relationship between two identities u and v. We only consider mutual relationships; hence G is an undirected graph, and an edge exists in G only if both nodes on that edge trust each other (bidirectional trust). Every node $v \in V$ in the network represents either an honest (benign) user or a Sybil (malicious) identity. We denote n to be the number of *honest* nodes in G, and denote m to be the number of edges between honest nodes.

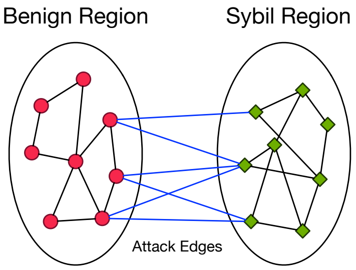

Figure 1: Sybil attack problem.

Figure 1 depicts the *Sybil attack* problem in social networks, in which an adversary can create an unlimited number of Sybil nodes and set up edges between them arbitrarily. We denote the subnetwork comprising all honest nodes to be the *honest* region, denote the subnetwork comprising all Sybil nodes to be the *Sybil* region, and denote the edges that connect the honest region and Sybil region to be the *attack edges*. We denote g to be the number of attack edges.

2.2 Social Sybil Defenses

An important thread of research has investigated the use of social trust relationships to mitigate the threat of Sybil attacks [36, 35, 8, 7, 4, 16, 27], which rely on the insight that it is costly for an adversary to set up trust relationships with honest users (i.e., attack edges). These defenses use a wide array of graph-theoretic techniques to bound the number or influence of the Sybil identities in proportion to the number of attack edges controlled by an adversary. For example, the SybilLimit protocol [35] guarantees that an adversary with g attack edges can insert about $g \cdot w$ Sybil identities, where w is the mixing time of the honest social network. Despite considerable differences between proposed mechanisms, researchers have shown that they rely on identifying local communities around a trust node [29].

Trust Assumptions: Social Sybil defenses often employ two key assumptions. First, the honest region is fast mixing, which presumes the existence of a well-connected, giant community structure of honest users. Second, the social network is a strong trust network, where the number of attack edges is relatively small [29].

Prior work has shown that these assumptions oversimplify reality. Mohaisen et al. [20] measured the mixing time of real-world social graphs and found that the actual mixing time is longer than the theoretical value anticipated by researchers who designed social Sybil defenses. This is due to the presence of multiple small communities in real-world social networks. Recent work has also questioned the assumption that it is costly for an attacker to set up trust relationships with honest users in the current online social networks [31, 6, 12]. Yang et al. showed that the number of attack edges may not be bounded if the underlying social network does not have strong trust: RenRen, the largest social networking platform in China, does not follow this assumption [34]. Ghosh et al. [10] showed that link farming is widespread on Twitter and that a majority of attack edges are farmed from a small fraction of Twitter users. These attacks highlight that relying on friendship links in current online social networks is not sufficient.

We note that large number of attack edges and a longer mixing time would degrade the security guarantees offered by these systems (e.g. linearly with mixing time for SybilLimit), but not render them inapplicable. Additionally, to strengthen defenses against weak social links, prior work has proposed (a) using user **interactions** to extract strong real-world trust relationships that cannot be manipulated at scale by an adversary [11, 31], or (b) explicitly asking users to identify their most trusted social contacts [30].

3. OVERVIEW OF TEMPORAL ATTACKS

Temporal attacks against Sybil defenses exploit the *dynamics* of the underlying social network. Such attacks are typically long-term attacks that incrementally exploit the vulnerabilities of Sybil defenses as the system evolves. In this section, we present a taxonomy of temporal attacks, and formalize our temporal attack model.

3.1 Temporal Attack Taxonomy

Based on the type of the exploited dynamic, we categorize temporal attacks into three categories: (1) attacks based on exploiting churn in the Sybil region, (2) attacks based on exploiting churn in the attack edges, and (3) attacks based on exploiting churn in the honest social region. The effects of these attacks can be compunded to effectively compromise system security.

3.1.1 Exploiting Churn in Sybil Region

The first category of attacks aim to exploit churn in the Sybil region. We observe that the attacker has complete control over identities in the Sybil region, as well the edges among the Sybil identities (which we term *Sybil edges*). Thus, the attacker can (1) artificially induce churn in the Sybil identities by deleting existing Sybil identities and introducing new Sybil identities, and (2) artificially induce churn among the edges connecting the Sybil identities by creating new Sybil edges and deleting existing Sybil edges. Note that such induced churn in the Sybil region does not violate the assumptions made in Sybil-defense mechanisms, since the number of attack edges remains bounded.

Next, we introduce a subclass of temporal attacks called *re-registration attacks* that exploit churn in the Sybil region while ensuring that the number of Sybil identities at any given time remains the same. In a re-registration attack, a strategic attacker can first ensure that the total number of Sybil identities are below the detection threshold for the corresponding Sybil defense mechanism. Next, the attacker can exploit vulnerabilities in the design of systems by changing its Sybil identities over time, while keeping the total number of Sybil identities at any instant of time below the detection threshold. This is easily achieved by deleting an existing Sybil identity and replacing it with a new Sybil identity.

For higher-level applications that leverage Sybil defenses, the re-registration attack has two immediate consequences.

- First, these applications cannot enforce secure *blacklisting* of Sybil identities that behave maliciously. This is because the re-registration attack can delete the blacklisted Sybil identity, and replace it with a new Sybil identity. Existing Sybil defenses do not defend against this attack as they are designed to provide a bound on the number of Sybil identities at an instant of time, while the re-registration attacks preserves this bound.
- Second, the re-registration attack allows the attacker to impact application resources/properties by changing Sybil identities over time. Let us consider the example of a voting system that validates user identity at the time of voting. By changing the registered Sybil identities over time, the attacker can insert a large number of malicious votes so as to subvert the voting results.

The above observations highlight how re-registration attacks can impact security properties of higher-layer applications. In this paper, we observe that such attacks have an impact on the security of the Sybil defense mechanism itself. In Section 4, we discuss how the re-registration attack can be used to disable a key security mechanism in the SybilLimit protocol, completely breaking its security and *allowing the attacker to insert an unbounded number of Sybil identities at a single instant in time.*

3.1.2 Exploiting Churn in Attack Edges

The second category of attacks aim to exploit churn in attack edges. Given the assumption that the number of attack edges is bounded, the designers of Sybil defenses have not considered the the issue of attack edge churn in the security analysis of their protocols. In a similar spirit to the re-registration attack discussed above, we propose to consider changes in attack edges over time, such that the total number of attack edges remains bounded at any given instant of time. However, in contrast to the re-registration attack which involved addition and deletion of Sybil identities, the attacker does not fully control the process of obtaining new attack edges, since an honest user must accept or interact with a Sybil identity to establish a new edge. Thus, careful attention is needed to model the capabilities of an attacker aiming to induce and exploit churn in attack edges (see Section 3.2 for a formal description of our *dynamic attack edge model*).

Since our churn model for attack edges ensures that the total number of attack edges are fixed at any instant of time, the security guarantees of social Sybil defenses should ideally hold. However, we show in this paper that an attacker can exploit the dynamic nature of attack edges to compromise the security of a number of Sybil defense mechanisms. We find that Sybil-resilient applications such as the Persea DHT completely fail against dynamic attack edges, because the system does not have the capability to revoke resources granted to an attacker based on an attack edge, even if the attack edge no longer exists. This allows an attacker with limited resources/attack edges to increase its influence in the system over time by changing its attack edges. We will also show that a number of protocols that rely on the knowledge of a trusted entity in the system (such as SybilInfer and SybilRank) are vulnerable to attacks where over time, the attacker can move its attack edges closer to the honest trust seed. Finally, such attacks also impact the design of Sybil-resilient messaging applications such as Ostra, in which an attacker can exploit system design and dynamic attack edges to deny service to honest users.

3.1.3 Exploiting Churn in Honest Region

The third category of attacks aim to exploit natural churn in the honest social network. Social networks are inherently dynamic, where new nodes and edges are formed frequently, and sometimes, existing nodes and edges get deleted. We note that most Sybil defenses should use interaction graphs, in which social trust edges are based on interactions among users. Relying only on binary friendship relationships is vulnerable to the attacker gaining many attack edges due to high rates of users accepting friendship requests from strangers [34, 6]. Prior work has observed, however, that the frequency of churn among existing edges is greater for interaction graphs than basic friendship graphs [31].

While the attacker may not control the rate or timings of churn in the honest social region, we find that system designers have not explicitly considered these issues in designing their protocols. Churn in the honest social region can lead to changes in the protocol state; such changes are often left unspecified by the system designers, and have serious consequences for system security. We will uncover such a vulnerability in the design of the SybilLimit protocol, allowing the attacker to compromise system security.

3.2 Temporal Attack Model

3.2.1 Attacker capabilities

Based on the above discussion, we note that the attacker **can actively** leverage temporal dynamics, by: (1) inducing churn in the Sybil identities by deleting existing ones and introducing new ones, and (2) inducing churn among the edges connecting the Sybil identies by deleting existing edges and creating new edges. Since the Sybil region is completely controlled by the attacker, we do not enforce any rate limit on this exploitation of churn in Sybil region. The attacker **can also passively** leverage temporal dynamics by exploiting churn in honest region, and monitoring changes in the protocol state. For the attack edges churn, the capability of the attacker is bounded by the basic assumption of social Sybil defense, i.e. a bounded number of attack edges, which is also inspired by the use of interaction graphs. Thus, we note that the attacker **cannot** create an arbitrary number of attack edges. Within the given bound, the attacker **can leverage churn** in attack edges, by intentionally deleting some existing edges (e.g. letting the interactions lapse) and creating new attack edges. In order to do this, the attacker would need to recollect and reuse its contrained resources for interactions with honest users, which takes certain amount of time. Thus, the attacker **cannot** regain new attack edges immediately after losing old ones.

3.2.2 Dynamic attack edges model

Motivated by the use of interaction graphs in prior work [31] and the idea of exploiting churn in attack edges, we formalize our dynamic attack edges model as follows.

Suppose that the attacker has a recurring budget of $\$R$ per unit time, and that it costs $\$E$ per unit time to maintain a trust relationship. An attack edge with a given target user can be maintained by posting messages that generate responses or comments from the target user, chatting with the target user, or otherwise inducing two-way communication that the system could use to label the edge as an active social relationship. The cost $\$E$ thus depends on the amount of interaction required for maintaining an attack edge, as well as the cost of human-based services or running intelligent chatbots for getting regular two-way communication with users. The attacker initially leverages his budget resources to obtain $g = \frac{R}{E}$ attack edges.

Under this model, the number of attack edges at any instant of time remains bounded (by R/E). Our temporal attacks exploit the fact that the interaction graph model allows the attacker to change the entities it interacts with over time, i.e., attack edges can be dynamic. The attacker can achieve this by utilizing its recurring budget. For example, in the next time instant, the attacker could allocate his budget to establish new trust relationships while foregoing previous trust relationships.

Studies of Sybil attacks in online social networks have shown that although many users do accept friendship requests from strangers, a significant fraction of users do not [6]. We thus further constrain the attacker by introducing a parameter δ to denote the fraction of users that never establish a trust relationship with an attacker. Further, attack edges might not form immediately upon demand. We use p to denote the maximum rate at which the attacker can obtain new attack edges (at the cost of previous attack edges).

4. EXPLOITING TEMPORAL DYNAMICS IN SYBILLIMIT

In this section, we first introduce the SybilLimit protocol [35], and then present our novel temporal attacks that allow an adversary to break SybilLimit's security guarantees. In particular, we show that an adversary can eventually register an unbounded number of Sybil identities in the SybilLimit protocol (at a single instant in time).

4.1 SybilLimit Background

SybilLimit is a decentralized protocol that defends against the Sybil attack. The goal of the protocol is for an honest verifier node v to determine whether a suspect node s is an honest node or a Sybil node.

Random routes and tails: SybilLimit defines a primitive called *random route* that operates as follows. Each user (node) in the social graph first constructs a *permutation map* of its edges, in which each edge is mapped to another edge pseudo-randomly. Then a random route of length l is constructed as a sequence of edges starting from a selected starting edge and iteratively applying the permutation map given by the current edge's terminating user. For example, a node A with neighbors B, C, D may have the following permutation map: $AB \to AC$, $AC \to AD$, and $AD \to AB$. If a random route reaches node A via edge AB, then according to the permutation map, the route traverses edge AC to node C, and the process continues with C's permutation map. The terminating edge of the random route is defined as the *tail* of the random route.

Protocol state: In SybilLimit, each user maintains $O(\sqrt{m})$ independent permutation maps (m is the number of edges in the honest region) and performs $r = O(\sqrt{m})$ random routes of length $w = O(\log n)$ (the mixing time of honest region); the i'th random route leverages the i'th permutation map for all nodes in the graph. Each user locally generates a public-private key pair and registers the public key at the terminal edges of the random routes (tails). SybilLimit does not assume knowledge of m; the value of $r = O(\sqrt{m})$ is estimated by the protocol using a benchmarking technique; please see [35] for more details.

Verification protocol: The key idea in SybilLimit is that the terminal edge of a random route starting from the honest region is more likely to be within the honest region than in the Sybil region, given that the number of attack edges g is bounded. This intuition motivates the following procedure used by a verifier node v to validate the identity of a suspect node s, if s wants to send some network traffic to v.

- *Intersection Condition:* The $r = O(\sqrt{m})$ tails of the random routes of the verifier and the suspect must have an intersection (using the Birthday paradox). Verifier nodes query the suspect for a list of its tails, contacts the tails to validate that the suspect is registered at those tails, and computes intersection with its own tails. If there is no intersection, the suspect is classified as a Sybil node. If there is an intersection, then the following Balance condition is checked.

- *Balance Condition:* A tail with a malicious terminal edge is known as a malicious tail. To prevent a malicious tail from validating an unbounded number of Sybil nodes, a verifier maintains a counter value for each of its tails that corresponds to how many suspect identities have been validated by that tail. The key idea is to keep the counter values for different tails of a user roughly uniform. If the acceptance of a suspect identity results in the counter values being unbalanced, then the suspect is classified as a Sybil.

Security claim: SybilLimit claims to provide the following security guarantee: given an adversary with $g = O(\frac{n}{\log n})$ attack edges to honest users, the number of Sybil identities in the system is bounded by $g \cdot w$ (i.e., the adversary can insert w Sybil identities per attack edge).

4.2 Temporal Attacks

We find that the complexity of decentralized Sybil defenses such as SybilLimit creates opportunities for temporal attacks. We now present our temporal attacks on SybilLimit: (a) a counter elevation attack that completely breaks SybilLimit's security guarantees, and (b) an induced social churn attack that also impacts SybilLimit security.

Re-registration attacks in SybilLimit: SybilLimit ensures that an adversary is limited in the number of edges (tails) where he can register its Sybil identities. The intersection condition ensures that the limited registration slots translates into a limited number of Sybil identities that can be validated by honest nodes. However, SybilLimit is vulnerable to the re-registration attack discussed previously, since it allows the initiator of the random route to overwrite the public key registered at its tail. Thus, while the number of Sybils is bounded at any *instant* of time, over time, the adversary can use these limited registration slots multiple times to insert an unbounded number of identities (by revoking previously registered Sybil identities and inserting new Sybil identities in its place).

Counter elevation attack: We now present the *counter elevation attack*, which leverages the inter-play between the re-registration attack and the SybilLimit balance condition. This attack allows an adversary to insert an unbounded number of identities at a particular instant in time. Thus, the consequences of our attack are devastating – in its current form, SybilLimit fails to provide any defense against Sybil attacks. We note that the counter elevation attack model is consistent with the SybilLimit threat model, and considers attack edges to be static.

We explain our attack using the following series of observations.

1) The SybilLimit balance condition is designed to prevent malicious tails in the network (i.e., tails where the terminal node is Sybil/malicious) from validating an unbounded number of Sybil identities (by claiming to have those identities be registered with itself). However, observe that in the re-registration attack, when an adversary revokes its Sybil identities and inserts new ones, the tails registering those Sybil identities are actually *honest*.

2) If the adversary is able to register its Sybil identities at a sufficient number ($O(n)$) of *honest* tails, then it can manipulate the balance condition by (a) registering its Sybil identities at honest tails, (b) getting its Sybil identities validated by honest nodes, thus increasing the counter values corresponding to the balance condition for its tails, and (c) repeating the previous steps. In other words, if the adversary has enough attack edges (O(n/wr)) to register its Sybil identities at a threshold number of honest tails, then the adversary can uniformly increase the counter values corresponding to honest tails. We note that the required number of attack edges are several orders of magnitude lower than the bound of $O(n/w)$ attack edges that SybilLimit claims to tolerate.

3) The above observations imply that an adversary with sufficient attack edges (O(n/wr)) can increase the counter values for honest tails (used by other nodes to check the balance condition). As the counter values corresponding to honest tails grow, an adversary can proportionally increase the number of Sybil identities that are validated by malicious tails, thus bypassing the balance condition and breaking the guarantees offered by SybilLimit. In this way, the adversary can insert an unbounded number of Sybil identities.

Induced social churn attack: Finally, we discuss another attack on SybilLimit that leverages churn in both the Sybil and honest regions. Recall that the balance condition in SybilLimit aims to limit the number of malicious identities than can be validated by a malicious tail. However, a strategic adversary can exploit protocol mechanisms to induce *artificial* churn in the social topology. The adversary can introduce new edges incident to intermediate nodes that are part of *escaping random walks*, random walks that start at an honest node and end in the Sybil region. This attack induces a change in the permutation tables of malicious users that are part of escaping random walks, resulting in the replacement of one malicious tail with another malicious tail. We note that churn in the honest region can also result in this behavior.

How does a change in the tails impact the balance condition? This behavior has not been specified in the SybilLimit protocol. Since the new tail has not participated in validating any identities, a natural interpretation would be to reset the counter value corresponding to the new tail to zero. This

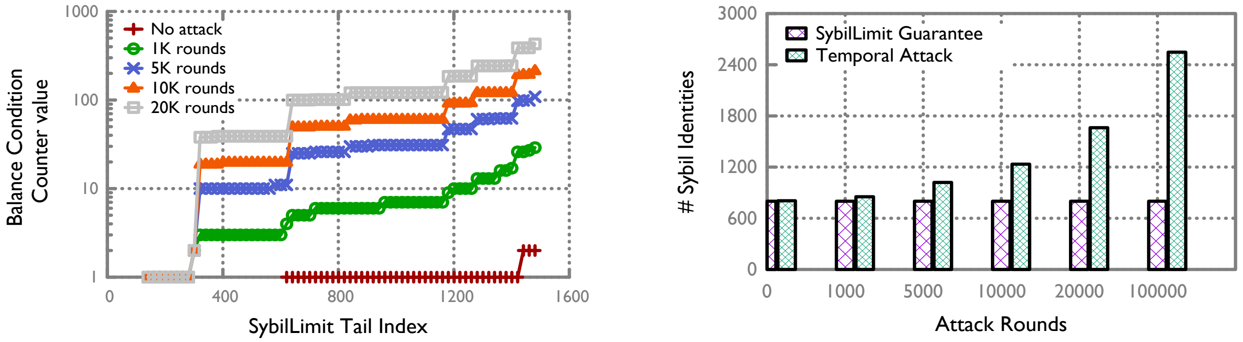

Figure 2: ***Counter value distribution corresponding to the SybilLimit balance condition (left), and the resulting Sybil identities that can be validated by an adversary (right), for varying attack rounds, using the Facebook interaction topology. We can see that over time, the attacker can increase the entire distribution of counter values, rendering the balance condition useless, and breaking SybilLimit security guarantees.***

behavior allows the adversary to validate additional Sybil identities via the new malicious tails. When counter values corresponding to these tails reaches a bound enforced by the balance condition, the adversary can repeat the attack.

4.3 Attack Demonstration

Here, we experimentally demonstrate the feasibility of our counter elevation attacks. Our goal is to show that an adversary can insert an unbounded number of Sybil identities in the SybilLimit protocol, breaking its security guarantees. For our evaluation, we consider a real-world Facebook interaction graph from the New Orleans regional network [28]. The dataset comprises of 46,952 nodes (users) connected by 876,993 edges.

We setup an initial configuration of the system by having 1000 honest users invoke the SybilLimit verification procedure to validate themselves to a verifier node. This helps initialize the counter values corresponding to the tails of the verifier node. Next, we considered 10 random nodes in the system to be malicious (representing a very low malicious node fraction of 0.0002 in the system, with only 80 attack edges), who perform the counter elevation attack described above. In each attack round, all of the attack edges are used to insert new Sybil identities, and these Sybil identities get validated by the honest verifier using the SybilLimit verification procedure. The results of the experiment are depicted in Figure 2, which shows the distribution of counter values corresponding to the tails of the verifier (left), and the resulting number of Sybil identities that can be validated by an adversary (right). As we expected, over time, the adversary is able to increase the counter values associated with the tails of the honest verifier, thus manipulating the floating bar corresponding to the SybilLimit balance condition. This validates our insight that the SybilLimit balance condition can be circumvented by an adversary, allowing it to register an unbounded number of Sybil identities at an instant in time (and breaking security guarantees offered by Theorem 3 in [35]).

5. EXPLOITING TEMPORAL DYNAMICS IN PERSEA, SYBILINFER, AND SYBILRANK

In this section, we present temporal attacks against SybilInfer, SybilRank and Persea. Our attacks rely on the insight that attack edges can change over time, even though the total number of attack edges remain fixed (at any time). We find that system designers have failed to consider such temporal dynamics in system design, leading to serious security vulnerabilities.

5.1 SybilInfer

SybilInfer [8] is a centralized algorithm for labeling nodes in a social network as honest users or Sybils controlled by an adversary. SybilInfer observes that a large Sybil attack results in the Sybil identities being separable from the honest identities via the minimum-quotient cut in the social graph, and directly aims to estimate the minimum-quotient cut. Towards this end, SybilInfer first constructs a probabilistic model of honest social networks based by performing special random walks over the social graph. The algorithm then leverages knowledge of a known honest user (trust seed) and uses Bayesian inference on the generated probabilistic model to output the set of detected Sybil identities (if any).

5.1.1 Temporal Attacks on SybilInfer

Next, we present a temporal attack against SybilInfer that exploits churn in attack edges to enhance connectivity between Sybil identities and the location of the trust seed over time. Let us suppose that an adversary has several Sybils and attack edges inserted in the system at time t. SybilInfer would detect some Sybil identities and block the detected Sybils. In the next timestamp $t+1$, the attacker could adjust its attack strategy to 1) preserve the survived attack edges, i.e., attack edges connected to Sybil identities that were not detected by SybilInfer, and 2) replace the detected attack edges with new attack edges that are closer to the trust seed. Observe that the total number of attack edges remains bounded in our attack model.

In our temporal attack, we aim to replace detected attack edges with new attack edges that connect to new benign nodes that the adversary is not already connected to. Connecting to new benign users has the advantage of moving the Sybil identities closer to the trust seed over time.

Therefore, our temporal attack lowers the probability that Sybils will be detected in timestamp $t+1$. As time evolves, Sybil identities become harder to be detected since the attack edges are moving closer to the trust seed and the survived Sybils are becoming stronger. Figure 3 depicts our experimental results, using a synthetic social network topology generated based on the Preferential Attachment model [5] (we use synthetic topologies for our SybilInfer experiments

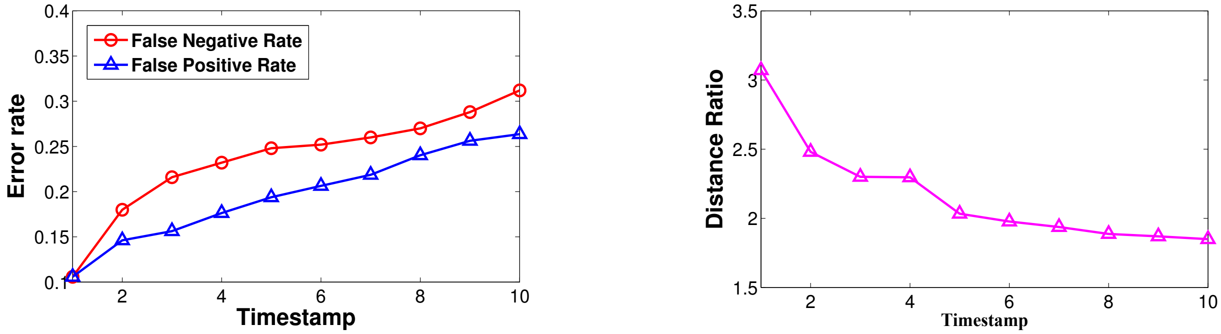

Figure 3: SybilInfer detection performance degrades with time under our attack edge churn model ($\delta = 0.5$). (a) is for the scale-free data and (b) is the ratio of the distance between trust seed and Sybils identities, and the distance between trust seed and other benign identities.

since it is difficult to scale the protocol to real-world datasets). The synthetic graph contains 1,000 benign nodes, 1,000 Sybil nodes, and 100 attack edges. We can see that SybilInfer detection performance degrades with time, validating our attack.

To shed insight behind the success of this attack, we evaluate the *distance* between the trust seed and the Sybil identities at each timestamp. Here, we utilize inverse of the affinity between two nodes to evaluate their distance. To normalize the distance, we further utilize the ratio of *the average distance between the trust seed and the Sybils* and *the average distance between the trust seed and the benign identities* as our distance metric, where

$$distance(t) = \frac{\mathbb{E}(dist(Trust\ node, Sybil))}{\mathbb{E}(dist(Trust\ node, Honest))} \qquad (1)$$

and the $dist(a, b)$ represents the inverse of the affinity between a and b, and the affinity is computed by random walk with restart method as in [25]. We can see that the success of our temporal attack is correlated with lower distance ratio.

5.2 SybilRank

SybilRank [7] is a centralized Sybil defense mechanism which is also based on the assumption that the Sybils have limited social connections to real users, i.e., the number of attack edges is bounded. The key insight is that it is easy for a short random walk starting from a set of honest users to quickly reach other honest users. On the other hand, it is relatively hard for these random walks to enter into the Sybil region because of a bound on the number of attack edges. Specifically, SybilRank performs a random walk starting from a set of honest users, i.e. a set of trust seeds, using power iterations. The length of the random walk is on the order of $log(|V|)$, i.e. the graph mixing time. When the random walk terminates, honest users tend to have a larger degree-normalized landing probability than Sybil identities.

Intuitively, SybilRank can be viewed as a mechanism that distribute trust scores via random walks starting from a set of trust seeds. This trust can only flow into the Sybil region via the limited number of attack edges. Thus, if we terminate the random walk early before it reaches stationary distribution (length less than the mixing time), honest users will have higher degree-normalized trust scores than Sybil identities. These trust scores can be used to produce a ranking list of each node.

5.2.1 Temporal Attacks on SybilRank

Similar to our attack on SybilInfer, we propose a temporal attack on SybilRank that (a) exploits dynamic nature of the connectivity between the benign region and the Sybil region, and (b) knowledge or inference of the honest trust seeds.

If the attacker already knows the identities of the honest trust seeds, then under the dynamic attack edge model (Section 3), the attacker can choose to preserve the attack edges that are close to the trust seeds, and replace the attack edges that are not. Thus, over time, the attacker can alter the location of its attack edges to move them closer to the honest trust seeds. Hence, the total trust that flows into the Sybil region from the trust seeds will be significantly larger than the theoretical anticipation.

Even in the absence of any prior knowledge about the trust seeds, an adversary can infer the identities of the trust seeds by using the Sybil defense mechanism as an oracle, as demonstrated in our experiments on SybilInfer.

5.3 Persea

Persea is a Sybil-resilient social DHT system [4]. The Persea DHT uses a circular identifier (ID) space, and hierarchically distributes the ID space among a set of bootstrap nodes in the network, by assigning each node its own ID chunk/region. Each bootstrap node in turn can invite other peers (based on trust relationships) and assigns them a node ID, and a subset of its own ID chunk/region. This process continues to form a bootstrap tree, that helps assign node identifiers in a Sybil resilient fashion. Persea ensures that a bound on the number of attack edges is translated into a bound on the size of the ID space controlled by an adversary.

Furthermore, Persea replicates (key, value) pairs over multiple ID locations that are evenly spaced over the circular ID space. Therefore, even if ID location is occupied by the malicious Sybils, redundant lookup operations can be used to retrieve the desired (key, value) pair from other honest ID regions.

5.3.1 Temporal Attacks on Persea

We present a novel attack against Persea that exploits churn in attack edges. Under our dynamic attack edge model (Section 3), the attacker can change attack edges over time. Specifically, for each timestamp $t + 1$, the attacker could replace a fraction p of its existing attack edges in the previous timestamp t with new attack edges. Note that at any instant of time, the total number of attack edges remains bounded.

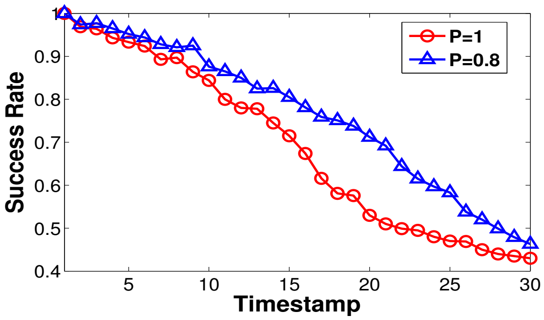

Figure 4: The lookup performance of Persea degrades with time (with Facebook interaction data $N_h = 46952, g = 2000$) under our attack edge churn model ($\delta = 0.5$). As time evolves, attackers gain access to a larger chunk of the ID space, resulting in lower lookup success rate.

We find that the security guarantees offered by Persea significantly degrade with time. This is because the system model in Persea does not allow honest entities to revoke the ID space they assigned to a trusted contact when the trust relationship is deleted.

Thus, as time evolves, the attacker would take control of an increasing fraction of the ID space in the system. This directly allows the adversary to exert greater control over the lookup process. In particular, as the fraction of the ID space controlled by an attacker increases, the probability than an attacker is able to intercept all of the redundant lookups in Persea also increases.

Figure 4 shows the lookup success rate in the Persea DHT over time, using the Facebook interaction dataset for $g = 2,000$. We can see the degradation in lookup performance over time. A lower value of p increases the required time duration to degrade lookup performance to a desired level. We used $\delta = 0.5$ to model the fraction of users that never establish attack edges with the adversary. These results validate our attack observations, and motivate our key message: system designers should explicitly consider system evolution in their design.

6. EXPLOITING TEMPORAL DYNAMICS IN OSTRA AND SUMUP

In this section, we present temporal attacks against Sybil-resilient applications such as Ostra and SumUp that allow an adversary to deny service to honest users.

6.1 Ostra

Ostra [16] is a Sybil-resilient messaging system that leverages trust relationships between users to thwart unwanted communication. For each user, Ostra is able to bound the rate of unwanted communication that a user can produce, based on the number of trust relationships that the user has. Ostra assumes that there is a trusted entity that observes all user actions and associate them with user identities. The key insight in Ostra is to associate the concept of *credit balances* with trust relationships (edges). Each edge in the network is associated with a credit balance B, and a balance range $[L, U]$. If the sender x wants to send some communication to a friend y, Ostra issues a specific token for this communication and the edge (x, y)'s L (from the sender's perspective) is raised by one. After y receives the communication, he/she needs to make a decision to mark this communication as wanted or unwanted. Previous adjustment of L is undone after y makes the decision. If y marks the communication as unwanted, the balance B of (x, y) is lowered by one (from the sender's perspective). If a sender x wants to communicate with a non-friend user z, Ostra finds a path from x to z. Then, the bounds and balance of all edges along the path will be adjusted accordingly.

To ensure legitimate users always able to communicate, credit balances in Ostra decay towards 0 at a constant rate d with $0 < d < 1$. Furthermore, if a user finds he/she has too much credit on all his/her links, he/she can forgive a small amount of debt from one of his/her friends. To mitigate communication failures, Ostra uses a timeout T and reset the credit bound adjustments if a communication has not been classified by the receiver after T.

However, we find that the path-based balance adjustment scheme makes Ostra vulnerable to a combination of re-regi--stration attack and attack edge churn. We will discuss this in the next two subsections.

6.1.1 *Temporal User Targeting Attacks on Ostra*

Attack scenario: Let us suppose that on the social network, a malicious node x is connected to a honest node y which is then connected to a honest node z. Also, let us suppose that x wants to send some unwanted communication to z and Ostra finds the path: $x \rightarrow y \rightarrow z$. If z marks the communication as unwanted, the credit balance of both edge (x, y) and edge (y, z) will be reduced by one. Thus, if the victim z has **less edges** compared to the number of attack edges owned by the attacker, a user targeting attack becomes practical. An attacker controlling a number of Sybils can successively send unwanted traffic to z. As a result, all edges of z will be exhausted within a short period of time, and z is no longer able to receive any communication from other honest users. We note that this attack is practical even if the system considers a decay factor d, since not only edges of the target, but also attack edges, will rejuvenate after some time. Also, even if the target choose to forgive the debt on one of his/her edges, since the attacker has more attack edges than the total number of edges owned by the victim user/target, the attacker is still able to send traffic to the target via an available attack edge, thus making the target unable to receive communication again.

Even if the target has **more edges** than the number of attack edges, this attack is still practical by letting the attacker leverage re-registration and attack edge churn. A typical attack model is that at the end of each day, the attacker could revoke certain fraction of invalid attack edges (with balance below a threshold), and replace with new attack edges, without changing the Sybil nodes. The attacker might also choose to revoke old Sybil identities associated with these invalid attack edges, and register new Sybil identities and establish new attack edges. The current design of Ostra simply assigns an initial balance $B = 0$ and the same L and U for all edges, including new attack edges. Thus, even if the number of attack edges at each time instant is bounded, the attacker is able to dynamically exploit attack edge churn to obtain new attack edge resources, and continuously send unwanted traffic to the target. As a result, all edges of the target will be exhausted over time.

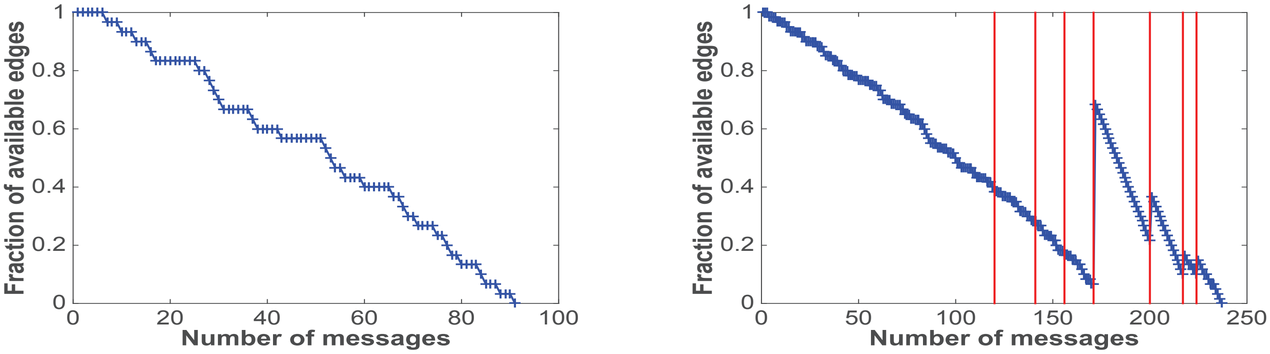

Figure 5: Temporal user targeting attacks on Ostra, (a) without attack edge churn, when the target has less edges than the number of attack edges, and (b) with attack edge churn, when the target has more edges than the number of attack edges. We observe that the target user is eventually unable to receive communication from other honest users.

Attack evaluation: We simulate Ostra protocol on a synthetic network structure generated by Preferential Attachment (PA) [5] model, and set $L = -3$ and $d = 10\%$ per day, as specified in Ostra. The synthetic network contains 1,000 honest nodes, 100 Sybil nodes, and 40 attack edges. We use synthetic graphs because we want to explore the behaviors of the attack under different parameter settings. For a randomly selected honest target, we randomly pick a Sybil node and let it send a message to the target. Figure 5 shows the fraction of available edges of the target versus the number of messages sent by the Sybils. Specifically, Figure 5 (a) shows the scenario when the target has less edges than the number of attack edges. We observe that as the Sybils send more messages, the fraction of available edges, i.e. capability to receive communication, decreases towards zero. In practice, this attack can be completed within a short period of time. Furthermore, the attacker can continuously monitor the credit of edges of the target, so that if one or some edges become valid again after several days due to the decay factor d, the attacker can immediately send traffic to the target and hence make these edges invalid again. We note that this type of attack can be completed without leveraging attack edge churn.

Figure 5 (b) shows the scenario when the target has more edges than the number of attack edges. In such scenario, the attacker is not able to exhaust all edges of the target within one round. Thus, the attacker needs to leverage attack edge churn w/o re-registration and continuously performs the attack for multiple rounds. In each round, the attacker randomly selects a Sybil and sends a message to the target, until all attack edges or all edges of the target become invalid. If all attack edges become invalid, the attacker leverages the attack edge churn and obtains some new attack edges.

In our experiments, we set $\delta = 50\%$ and $p = 0.1$. We also model the case that if the target finds all of his/her edges become invalid, he/she will randomly pick a neighbor and forgive the debt on the corresponding edge. The attack ends if all edges of the target become invalid. In Figure 5 (b), we observe that by leveraging attack edge churn, the attacker is able to continuously exhaust edge resources of the target via multiple rounds (separated by red vertical lines), even if some edges of the target become valid again due to decay factor (the jumps on the figure). Thus, the attacker can significantly reduce the rate of communication between honest users by perform this type of temporal attacks.

6.1.2 Temporal Edge Targeting Attacks on Ostra

Attack scenario: In addition to attacking a randomly selected target, the attacker can also target at certain edges in the honest region, by continuously sending unwanted traffic across these edges and eventually exhausting the credit on them. As a result, the honest region will be partitioned into two communities, such that users in one community will not be able to send traffic to users in the other community. Researchers have found that honest users tend to form multiple small communities [17] driven by different purposes (e.g., geographical location, education and career). This multi-community structure prohibits the existence of a giant community component and hence makes the honest region vulnerable to temporal edge targeting attacks.

Like the temporal user targeting attacks, if the count of targeted edges is **less** than the number of attack edges owned by the attacker, the attacker can quickly exhaust the credit on them thus making these edges invalid. Even if the count of targeted edges is **greater** than the number of attack edges, the attacker is still able to leverage attack edge churn w/o re-registration to exhaust the credit on these edges over time.

Attack evaluation: To evaluate this attack, we synthesize the network structure by generating two honest regions (500 nodes for each) from PA model and connect them with certain number of edges. We then generate two Sybil regions (100 nodes for each) and connect each to a honest region with the same number of attack edges. We set the parameters in Ostra the same as in 6.1.1. Denote the two Sybil regions as $Sybil\_region\_A$ and $Sybil\_region\_B$. To perform the attack, we randomly pick a Sybil in each Sybil region, and send a message from $Sybil\_region\_A$ to $Sybil\_region\_B$. In our experiments, we vary the number of internal edges that link the two honest regions, and the total number of attack edges owned by the attacker, to understand different behaviors. Figure 5 shows the fraction of available internal edges versus the number of messages passed across the network. Specifically, Figure 5 (a) shows the scenario when the number of internal edges is less than the number of attack edges divided by two. We observe that as more messages pass between Sybils in the two regions, the fraction of available (valid) internal edges decreases towards zero.

Figure 6 (b) shows the scenario when the number of internal edges is greater than the number of attack edges divided

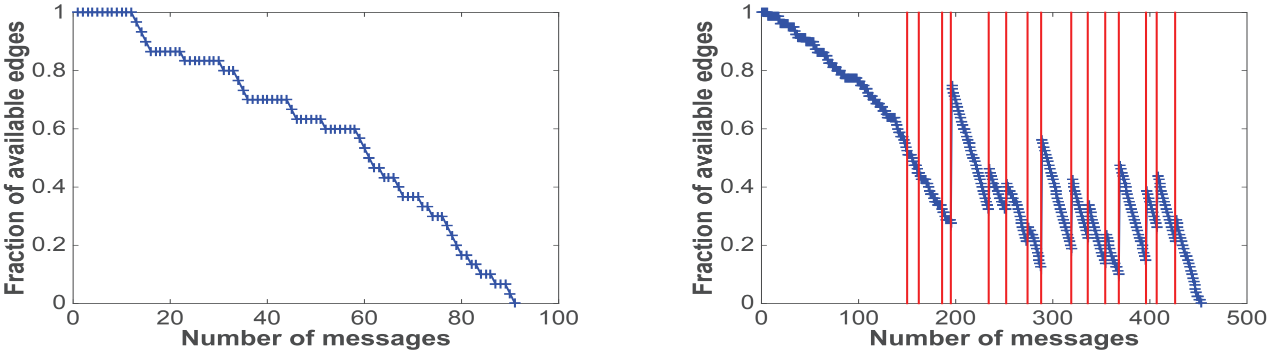

Figure 6: Temporal edge targeting attacks on Ostra, (a) without attack edge churn, when the number of targeted edges is less than the number of attack edges, and (b) with attack edge churn, when the number of targeted edges is greater than the number of attack edges. We observe that honest users are eventually unable to communicate across the targeted edges.

by two. Similarly to user targeting scenario, the attacker is able to exhaust the resources of the internal edges over time by leveraging attack edge churn.

6.2 SumUp

SumUp [27] is a Sybil-resilient voting system that leverages trust network among users. SumUp assumes that there is one trusted vote collector that is far from the Sybil region. In the ticket distribution process, the vote collector distributes C_max tickets across the network in a breadth-first manner. Each internal node aggregates the tickets it receives, and consumes one ticket, and distributes the remaining tickets evenly to the neighbors at the next level. The capacity of an edge is the number of tickets transferred on this edge plus one. When the ticket distribution completes, nodes that consumed a ticket before become entry points. In the vote collection process, each voter needs to vote for the product via an entry point. SumUp computes the set of max-flow paths from the vote collector to all voters, and the votes will be collected back to the vote collector via the computed paths. Hence, each object can receive C_max votes in maximum.

To limit the number of bogus votes, the vote collector assigns a penalty value for each link. Once the vote collector identifies a bogus vote, all the links on the path to the voter will be penalized. The penalty grows if a voter continuously sends bogus votes, and the link will be eliminated if the penalty grows above a threshold. In the further ticket distribution process, links with high penalty will be distributed fewer tickets.

6.2.1 *Temporal Attacks on SumUp*

SumUp assumes that the number of the attack edges is bounded. However, after examination, we find that the security guarantee of SumUp heavily relies on another underlying assumption, which is that the vote collector is placed far away from the attack edges. The system does not make it clear how to select the location of the vote collector. The attacker can exploit natural churn of the honest region and place the attack edges close to the vote collector. If this happens, a large number of Sybil nodes will become entry points. Since each user needs to vote for a product via an entry point, a large fraction of votes collected will be bogus votes, which breaks the system design goal.

Even if the vote collector is placed far away from attack edges, like Ostra, the attacker can still perform a combination of re-registration and attack edge churn. The observation is that over time, the attacker can re-register Sybil identities and make bogus votes. Since honest edges are mostly located in the lower level of graph, i.e. close to the vote collector, after those bogus votes are successfully marked by the vote collector , the penalty of honest edges on the path will also be raised. Thus, over time, the penalty of targeted honest edges will exceed the threshold value, eliminating the edges. This combined attack can target a specific voter, eventually exhausting the available edges of this voter and makes him/her unable to vote.

In this section, we discussed and demonstrated temporal vulnerabilities of two Sybil resilient applications: Ostra and SumUp. Our attacks highlight the importance of considering temporal system dynamics in the design and security analysis of social Sybil defense mechanisms.

7. COUNTERMEASURES AND DISCUSSION

In this section, we first discuss how economic considerations could impact the attacks described in this paper. We then consider possible countermeasures for temporal attacks on different Sybil defense systems. Finally, we briefly sketch out a general method that aims to detect anomalous churn in the Sybil region.

7.1 Economic Considerations

Our attack model in Section 3.2 assumes that an adversary that can induce churn in the Sybil region via creation and deletion of Sybil identities. This is consistent with the threat model of social Sybil defenses, which typically assume that such operations have no cost for an adversary [35, 8]. In practice, social networks employ a range of defense mechanisms that raise the bar for an adversary, including account registration barriers such as CAPTCHAs, email and phone confirmation, and IP blacklists [3, 24]. Thus creation and deletion of Sybil identities has an economic impact on the adversary, in terms of the resources required to circumvent registration barriers.

In recent years, however, there has been an emergence of an underground market that specializes in bypassing registration barriers such as email, phone, and CAPTCHA confirmation [33, 23, 21]. In fact, a number of websites and Internet forums have emerged that allow adversaries to easily obtain a large network of Sybil accounts (or followers)[1,2,3]. For example, Hotmail and Yahoo accounts are available on blackhatworld.com for $6 per thousand, while Twitter accounts from the same forum are $40 per thousand. Thus, we expect that the combination of such ad hoc mechanisms with social Sybil defenses should increase the cost of performing temporal attacks, much as CAPTCHAs increase the cost of spam [21]. They do not fundamentally mitigate our attacks, however, motivating the search for improved defenses.

7.2 System-specific Countermeasures

Batch mode enforcement for SybilLimit: To rate limit the re-registration of new Sybils, a simple countermeasure is for honest nodes to process protocol messages (such as incoming random route requests) in a batch mode, say every x time units. This means that if an attacker tries to introduce multiple Sybil identities using the registration slot (tail), then it can do so only once every x time units.

This countermeasure introduces a trade-off between usability and security. Clearly, as the time period x is increased, the security of the system improves as the attacker has to wait longer before being able to replace its existing Sybil identities. On the other hand, increasing the time period x adversely impacts the usability of the system, since new honest nodes have to wait longer before being able to set up their random routes and get validated by the system.

Note that this defense only slows down the rate of attack. It does not fundamentally mitigate our observations that (a) over time, an adversary can insert different Sybil identities in the system (bounded), and (b) given enough time, an adversary can insert an unbounded number of Sybil identities at a single instant of time.

Bound the variance for SybilLimit: The second countermeasure is for honest nodes to bound the variance in the number of new random routes, terminating at itself, corresponding to its $O(\sqrt{m})$ entries. For example, if in time period x, a particular public key entry registered at an honest node A is overwritten a large number of times, as compared to other public keys registered at node A, then this is an indication of attack. The parameters for the bound on the variance can be learned using models of honest social network evolution in real world datasets. A new incoming random route message that violates this condition is ignored.

Note that our second countermeasure constrains the effects of an adversary by rate limiting new random route setups based on models of honest social network evolution. Similar to before, this attack also only slows down an adversary, but does not fundamentally mitigate our observations.

Moving-target defense for SybilInfer and SybilRank: The idea of moving-target defense (MTD) is to impose asymmetric uncertainty for the attacker by making systems dynamic and harder to predict. By adding randomness in the system, the attacker has to use lots of resource to study the system, identify its vulnerabilities, and deploy the attacks. Specifically for systems like SybilInfer and SybilRank, which rely on performing random walks from honest trust seeds, the idea of MTD can be leveraged by first selecting multiple random seeds, and then regularly changing them after some time T. Thus, the attacker is not able to estimate the location of the trust seeds once and use this location information to perform effective attacks forever.

Ephemeral attacker resource for Persea: The identified problem with Persea is that the resources corresponding to deleted edges can not be revoked. Thus, the attacker is able to change attack edges over time and obtain more system resources. To deal with this, a natural way is to introduce the concept of "ephemeral resources", such that the obtained system resources eventually time out (unless renewed). Thus, the attacker would not be able to increase its share of system resouces by changing attack edges over time. For Persea, a possible solution is to enforce a timeout T for the ID space of an edge, so that this ID space will eventually not be valid once the edge has been deleted.

Asymmetric penalty for Ostra and SumUp: The design of Ostra penalizes each edge along the path evenly once the receiver marks the communication traffic as unwanted. To mitigate the previously discussed user targeting and edge targeting attacks, we may adopt an asymmetric penalty approach, by penalizing edges close to the sender more and penalizing edges close to the receiver less. Thus, the attacker would lose more attack edges to attack a target/set of edges, comparing to the previous symmetric penalty approach. For SumUp, a similar approach could be adopted by penalizing edges close to the vote collector (i.e., in the lower level) less and penalizing edges far from the vote collector more.

Generic Defense via Detecting Anomalous Churn: We now briefly discuss a possible approach to detect anomalous churn in the social graph that could work on a variety of systems. The key insight is that temporal attacks often rely on the attacker inducing a high rate of churn in the Sybil region, which can be used as a point of detection. Thus, we propose to observe the graph evolution to distinguish the Sybil region from the honest region. For example, one can quantify change in the neighborhood structure for each user in a time series of graphs, using statistical distance metrics, and use them as a feature for detection. Once the Sybil region is detected, Sybil identities and their attack edges can then be blocked from the social network (and the process is repeated).

8. CONCLUSION

In this paper, we explored temporal dynamics of social Sybil defenses: churn in the Sybil region, churn in attack edges, and churn in the honest region. We proposed temporal attacks that exploit these system dynamics and investigated the vulnerabilities of a variety of social Sybil defenses. We find that temporal attacks can have devastating consequences for system security, specially for distributed Sybil defenses such as SybilLimit and Persea. We also discussed proposed possible countermeasures that could be used to prevent these attacks, though carefully designing and evaluating robust countermeasures remains for future work. Our work motivates the importance of explicitly considering temporal dynamics in both system design and system security evaluation.

[1] https://devumi.com/twitter-followers/

[2] https://www.fastfollowerz.com/

[3] http://twitterboost.co/

Acknowledgments

We would like to thank the anonymous reviewers at CCS 2015 for helpful feedback, and we are especially grateful to Ting Yu for his guidance as our shepherd. This work was supported in part by NSF awards number CNS-1423139, CNS-1409415, CNS-1423163 and CNS-0954133.

9. REFERENCES

[1] Known bad relays in Tor. `https://trac.torproject.org/projects/tor/wiki/doc/badRelays`.

[2] Trotsky IP addresses in Tor. `https://trac.torproject.org/projects/tor/wiki/doc/badRelays/trotskyIps`.

[3] Ahn, L. V., Blum, M., Hopper, N. J., and Langford, J. Captcha: Using hard ai problems for security. In *EUROCRYPT* (2003).

[4] Al-Ameen, M. N., and Wright, M. Design and evaluation of Persea, a Sybil-resistant DHT. In *ACM ASIACCS* (2014), pp. 75–86.

[5] Barabási, A.-L., and Albert, R. Emergence of scaling in random networks. *science 286*, 5439 (1999), 509–512.

[6] Bilge, L., Strufe, T., Balzarotti, D., and Kirda, E. All your contacts are belong to us: automated identity theft attacks on social networks. In *WWW* (2009).

[7] Cao, Q., Sirivianos, M., Yang, X., and Pregueiro, T. Aiding the detection of fake accounts in large scale social online services. In *NSDI* (2012).

[8] Danezis, G., and Mittal, P. Sybilinfer: Detecting Sybil nodes using social networks. In *NDSS* (2009).

[9] Douceur, J. The Sybil Attack. In *IPTPS* (2002).

[10] Ghosh, S., Viswanath, B., Kooti, F., Sharma, N. K., Korlam, G., Benevenuto, F., Ganguly, N., and Gummadi, K. P. Understanding and combating link farming in the twitter social network. In *WWW* (2012).

[11] Gilbert, E., and Karahalios, K. Predicting tie strength with social media. In *CHI* (2009).

[12] Irani, D., Balduzzi, M., Balzarotti, D., Kirda, E., and Pu, C. Reverse social engineering attacks in online social networks. In *DIMVA* (2011).

[13] Kossinets, G., and Watts, D. J. Empirical analysis of an evolving social network. *Science 311*, 5757 (2006), 88–90.

[14] Krebs, B. Twitter bots drown out anti-kremlin tweets, Dec. 2011. `https://krebsonsecurity.com/2011/12/twitter-bots-drown-out-anti-kremlin-tweets/`.

[15] Lesniewski-Laas, C., and Kaashoek, M. F. Whanaungatanga: A Sybil-proof distributed hash table. In *NSDI* (2010).

[16] Mislove, A., Post, A., Druschel, P., and Gummadi, K. P. Ostra: leveraging trust to thwart unwanted communication. In *NSDI* (2008).

[17] Mislove, A., Viswanath, B., Gummadi, K. P., and Druschel, P. You are who you know: inferring user profiles in online social networks. In *Proceedings of the third ACM international conference on Web search and data mining* (2010), ACM, pp. 251–260.

[18] Mittal, P., Caesar, M., and Borisov, N. X-vine: Secure and pseudonymous routing using social networks. In *NDSS* (2012).

[19] Mohaisen, A., Hopper, N., and Kim, Y. Keep your friends close: Incorporating trust into social network-based Sybil defenses. In *INFOCOM* (2011).

[20] Mohaisen, A., Yun, A., and Kim, Y. Measuring the mixing time of social graphs. In *IMC* (2010).

[21] Motoyama, M., Levchenko, K., Kanich, C., McCoy, D., Voelker, G. M., and Savage, S. Re: Captchas-understanding captcha-solving services in an economic context. In *USENIX Security Symposium* (2010), vol. 10, p. 3.

[22] Thomas, K., Grier, C., and Paxson, V. Adapting social spam infrastructure for political censorship. In *LEET* (2012).

[23] Thomas, K., Grier, C., Song, D., and Paxson, V. Suspended accounts in retrospect: an analysis of twitter spam. In *ACM IMC* (2011), ACM, pp. 243–258.

[24] Thomas, K., Iatskiv, D., Bursztein, E., Pietraszek, T., Grier, C., and McCoy, D. Dialing back abuse on phone verified accounts. In *ACM CCS* (2014).

[25] Tong, H., Faloutsos, C., and Pan, J.-Y. Fast random walk with restart and its applications. In *ICDM* (2006).

[26] Tran, N., Li, J., Subramanian, L., and Chow, S. Optimal Sybil-resilient node admission control. In *INFOCOM* (2011).

[27] Tran, N., Min, B., Li, J., and Subramanian, L. Sybil-resilient online content voting. In *NSDI* (2009).

[28] Viswanath, B., Mislove, A., Cha, M., and Gummadi, K. P. On the evolution of user interaction in Facebook. In *WOSN* (2009).

[29] Viswanath, B., Post, A., Gummadi, K. P., and Mislove, A. An analysis of social network-based Sybil defenses. In *SIGCOMM* (2010).

[30] Wei, W., Xu, F., Tan, C., and Li, Q. Sybildefender: Defend against Sybil attacks in large social networks. In *INFOCOM* (2012).

[31] Wilson, C., Boe, B., Sala, A., Puttaswamy, K. P., and Zhao, B. Y. User interactions in social networks and their implications. In *Eurosys* (2009).

[32] Winter, P., Köwer, R., Mulazzani, M., Huber, M., Schrittwieser, S., Lindskog, S., and Weippl, E. Spoiled onions: Exposing malicious Tor exit relays. In *PETS* (2014).

[33] Yang, C., Harkreader, R., Zhang, J., Shin, S., and Gu, G. Analyzing spammers' social networks for fun and profit: a case study of cyber criminal ecosystem on twitter. In *WWW* (2012), ACM, pp. 71–80.

[34] Yang, Z., Wilson, C., Wang, X., Gao, T., Zhao, B. Y., and Dai, Y. Uncovering social network sybils in the wild. *ACM Transactions on Knowledge Discovery from Data (TKDD) 8*, 1 (2014), 2.

[35] Yu, H., Gibbons, P. B., Kaminsky, M., and Xiao, F. Sybillimit: A near-optimal social network defense against Sybil attacks. In *IEEE S&P* (2008).

[36] Yu, H., Kaminsky, M., Gibbons, P., and Flaxman, A. SybilGuard: Defending against Sybil attacks via social networks. In *SIGCOMM* (2006).