Consequences of Connectivity: Characterizing Account Hijacking on Twitter

Kurt Thomas[†] Frank Li[†] Chris Grier[†*] Vern Paxson[†*]
[†]University of California, Berkeley [*]International Computer Science Institute
{kthomas, frankli, grier, vern}@cs.berkeley.edu

ABSTRACT

In this study we expose the serious large-scale threat of criminal account hijacking and the resulting damage incurred by users and web services. We develop a system for detecting large-scale attacks on Twitter that identifies 14 million victims of compromise. We examine these accounts to track how attacks spread within social networks and to determine how criminals ultimately realize a profit from hijacked credentials. We find that compromise is a systemic threat, with victims spanning nascent, casual, and core users. Even brief compromises correlate with 21% of victims never returning to Twitter after the service wrests control of a victim's account from criminals. Infections are dominated by *social contagions*—phishing and malware campaigns that spread along the social graph. These contagions mirror information diffusion and biological diseases, growing in virulence with the number of neighboring infections. Based on the severity of our findings, we argue that early outbreak detection that stems the spread of compromise in 24 hours can spare 70% of victims.

Categories and Subject Descriptors

K.4.1 [**Public Policy Issues**]: Abuse and crime involving computers

Keywords

Account hijacking; compromise; social networks

1. INTRODUCTION

In this paper, we expose the serious large-scale threat of criminal account hijacking and the resulting damage incurred by users and web services. To conduct our study, we develop a systematic approach for detecting large-scale attacks on Twitter that we leverage to identify victims of compromise, track how compromise spreads within the social network, and evaluate how criminals ultimately realize a profit from hijacked credentials. We retrospectively apply our detection scheme on a dataset consisting of 8.7 billion tweets generated by 168 million Twitter users during a 10-month period between January, 2013–October, 2013. In total, we identify 14 million users that fell victim to hijacking in addition to nearly 5 million fraudulent accounts used to fuel spam campaigns. While it is possible to repurpose our detection techniques to operate in real time and to serve as a defense, our key contribution with this research is to bring to light the systemic risks legitimate users face on social networks.

We find that criminals succeed in hijacking accounts from users around the globe, irrespective of user savviness. Nascent, casual, and core users with hundreds to thousands of followers all fall victim to attacks. Even brief compromises—we find a median duration of 1 day in our dataset—correlate with 21% of victims never returning to Twitter after the service wrests control of a victim's account from criminals. Furthermore, 57% of victims lose friends post-compromise in response to spam the victim's account sends. These results illustrate that compromise is not a simple threat easily solved by password resets; instead, social networks incur lasting damage after each attack with respect to core success metrics such as user retention and engagement.

Our results suggest that criminals rely heavily on *social contagions*—phishing and malware campaigns that spread along the Twitter social graph and exploit a user's friends. Of over 2,600 distinct outbreaks of compromise we identify, 88% exhibit graph connectivity between the victims. These contagions grow in virulence with the number of neighboring victims, where a user with 20 compromised neighbors is 10x more likely to become compromised compared to a user with one compromised neighbor. We find that the social process driving compromise mirrors that of information diffusion [2, 5] and biological infections [21]. A direct consequence is that compromise in social networks spreads much slower compared to Internet worms [24], with only 30% of victims emerging in the first day of a contagion's outbreak. This opens up the possibility for the early detection of hijacking attacks, where stopping contagions in 24 hours would spare 70% of victims.

Finally, as compromise is a financial endeavor, we examine how criminals monetize hijacked accounts. We identify three dominant strategies: the sale of nutraceutical weight loss supplements, fake follower (or retweet and favorite) programs, and lead-generation scams. Combined, 68% of compromised victims hawk these money-making schemes, ac-

CCS'14, November 3–7, 2014, Scottsdale, Arizona, USA.

ACM 978-1-4503-2957-6/14/11 15.00.
http://dx.doi.org/10.1145/2660267.2660282.

counting for 69% of all spam tweets generated by hijacked credentials. These schemes are similar to previous Twitter spam campaigns that rely on fraudulent accounts [16, 28], though the challenge of reaching an audience is vastly simplified by account hijacking.

In summary, we frame our contributions as follows:

- We demonstrate the hijacking by criminals of more than 13 million Twitter accounts; this threat represents one of the single largest challenges facing web services.
- We find that even when Twitter wrests control of accounts from criminals, 21% of victims never return to the service and 57% lose friends.
- We show that criminals rely on phishing and malware campaigns that exploit social connections to spread in a similar fashion to memes and biological diseases, making them increasingly virulent.
- We characterize how criminals ultimately profit by spamming a victim's followers with weight loss supplements, fake follower programs, and lead generation scams.

2. BACKGROUND AND RELATED WORK

While the prevalence of compromised users amongst spamming accounts has been identified by a number of prior studies, the process driving account hijacking and the damage that ensues has never been explored. We provide an overview of the potential mechanisms criminals use to hijack accounts as well as outline previous approaches for detecting unwarranted behavior on a victim's account. Whenever applicable, we highlight how our research fits into the broader context of abuse targeting social networks.

2.1 Account Hijacking Techniques

Database Dumps: Sophisticated attacks on companies are emerging as a regular threat, where breaches have resulted in the exposure of millions of usernames and passwords at Adobe, LinkedIn, and Twitter [17, 22, 26]. When breaches are directly linked to a social network, criminals can take control of a victim's account. Alternatively, break-ins at unrelated services can still prove lucrative to criminals due to 43% of users reusing passwords across services [8].

Password Guessing: Weak passwords face a significant risk from brute-force guessing. For example, miscreants launched an attack targeting GitHub from 40,000 addresses, affecting an unknown number of victims [4]. Provided enough time and resources, criminals can effectively mine the credentials of multiple victims to access their accounts.

Social Contagion: Rather than target weaknesses in a web service, criminals can target a service's users by disseminating phishing pages and malware via social engineering like the Koobface botnet [29] or by drive-by exploits [15]. We refer to attacks that spread within a social network along graph edges as a *social contagion*.

External Contagion: Malware and phishing attacks spreading externally to a social network can still result in criminals stealing a victim's social credentials. We refer to such attacks as an *external contagion*.

2.2 Detecting Hijacked Accounts

Our approach for detecting hijacked accounts builds on a large body of prior work for characterizing spam and abuse in social networks. In particular, we iterate on previous approaches by Gao et al. [12, 13] and Grier et al. [16] for clustering social network content into spam campaigns based on text and URL features. One limitation of these approaches is they fail to distinguish between *fraudulent accounts* used solely to disseminate spam and *compromised accounts* exhibiting symptoms of an infection. While the authors of both works conclude that compromised accounts are responsible for a substantial fraction of spam, this conclusion was based purely on manual analysis, rather than devising an automated framework for detecting hijacking. Furthermore, the authors focused their analysis on spam campaigns, omitting any discussion of how victims were hijacked.

One existing approach for explicitly detecting hijacked victims in social networks is COMPA [10]. This system builds a historical model of a user's activities such as application usage, language, and posting frequency. When an anomalous message appears on a victim's account that violates the constructed usage profile, the user is considered hijacked. This signal is boosted by identifying clusters of users that all post similar content. Our approach, while similar in that we cluster hijacked victims, makes no assumptions on the stability of user behavior (which may change due to travel, installing new apps, or varying engagement levels) and does not require an historical model per account, which is expensive to maintain at scale. Furthermore, while we believe our detection framework can be deployed as a proactive defense, its purpose in this paper is a means for generating a sample of hijacked victims—our primary contribution in this work is examining the impact and spread of compromise.

3. METHODOLOGY

Our strategy for identifying the hijacked accounts in social networks consists of five components, outlined in Figure 1. Over a 10 month period we collect 61% of all tweets containing URLs, amounting to roughly 40M tweets per day (❶). We organize these tweets into clusters (❷), classifying each cluster as either a benign meme, an infection spreading via compromised accounts, or spam campaigns produced by fraudulent accounts (❸). We then crawl the social graph of the accounts involved in each cluster (❹), allowing us to measure the connectivity of victims and associated graph properties. Finally, we label each account in our dataset as benign, compromised, or fraudulent (❺). The entire process uses a combination of Hadoop, Pig, and Spark.

3.1 Data Collection

Our dataset consists of 8.7 billion tweets posted by 168 million users between January 7, 2013 through October 21, 2013. We collect tweets directly from Twitter's streaming API[1] using the `statuses/filter` method which we configure to return a privileged sample of all tweets containing URLs. Our feed does not provide a fixed sample rate; instead, we receive a reduced sample size during peak strain on network routes between our collection point and Twitter's API infrastructure, a phenomenon previously documented by Morstatter et al [20].

[1]https://dev.twitter.com/docs/streaming-apis

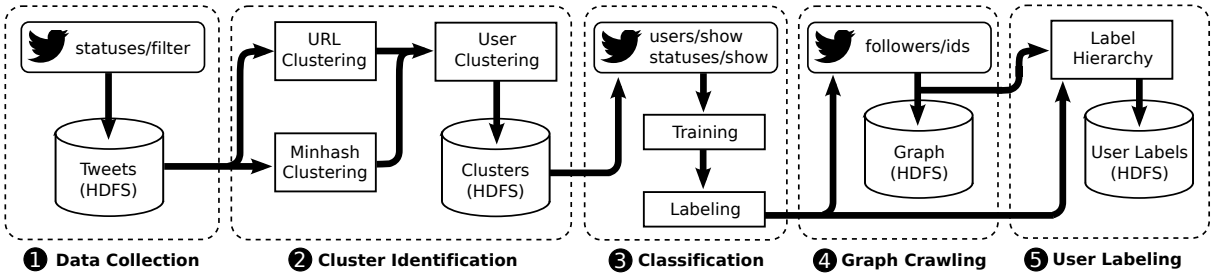

Figure 1: Data processing pipeline. We receive a stream of roughly 40M tweets per day, which we store into HDFS. We then group these tweets into clusters, labeling them as memes, spam from fraudulent accounts, or infections based on whether Twitter has since deleted or disabled the tweets or accounts. Once classified, we crawl the social graph of all accounts and finally label each account.

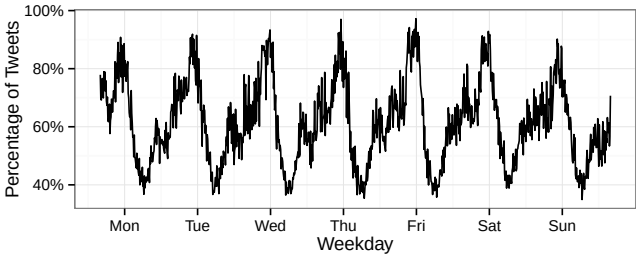

Figure 2: Average daily sample rate throughout our collection period. We receive nearly 100% of all tweets with URLs around midnight PST, while our sample rate drops to roughly 40% during peak hours of network strain.

To understand how the network bottleneck impacts our sample rate, we compare the overlap of our sample with the statuses/sample feed (collected over the same period and unaffected by network bottlenecks) which Twitter advertises as a real-time truly random sample fixed at 1% of all tweets. We find that we receive roughly 40% of all tweets with URLs during peak hours of Twitter activity and nearly 100% of all tweets with URLs during low periods of activity, as shown in Figure 2. In total, we estimate we receive 61% of all tweets with URLs averaged across all days and hours.

3.2 Identifying Tweet Clusters

We organize the billions of tweets in our dataset into clusters through a combination of text, URL, and user clustering, similar to previous approaches for identifying memes and spam clusters on Twitter [12, 13, 16, 27, 28]. Clustering is a two step process, as shown in Figure 1 (❷): we first group all of the tweets in our dataset based on an approximation of their content represented by a minhash [3]. We also employ a secondary clustering strategy, whereby we group any tweets with the same URL. The clusters resulting from both of these steps are then fed into a final phase where we merge clusters with overlapping users, storing the final collections of tweets and users.

3.2.1 Clustering on Similar Content

Apart from retweet chains where the provenance of a cluster is explicit, identifying tweets that all discuss a similar topic is typically a problem associated with near duplicate detection and topic modeling. In order to measure the semantic similarity between two tweets, there is a noteworthy distinction between *exact duplicates* and *near duplicates* [27].

Exact Duplicates are any pair of tweets where every character is identical. Legitimate exact duplicates result from retweet chains, users sharing news stories, and auto-generated messages from applications, while spam exact duplicates appear because a single spammer posts an identical message distributed via multiple fraudulent or compromised accounts.

Near Duplicates are any pair of tweets with a strong degree of content overlap (as developed below). Cosmetic differences occur because legitimate users rephrase a story or news item, while spammers frequently permute spam templates in order to evade rudimentary text clustering. An example from our dataset:

$m1$: *Aweesomeee! I made $171.50 this week so far taking a couple of surveys. http://t.co/cwG67lh4*

$m2$: *Awesome! I made $106.03 this week so far just filling out a couple of surveys. http://t.co/PoHBayLz*

In order to detect whether a pair of tweets are near duplicate, we first strip each tweet of Twitter specific nomenclature such as mentions (@user) and retweets (RT @user), any URLs in the tweet, and any remaining non-alpha characters (thus removing digits, punctuation, and whitespace). We then compute the set of all character n-grams from the message using a rolling window, where n is a tunable parameter. We consider two messages to be equal if their set of n-grams M_i and M_j have a Jaccard similarity coefficient greater than τ, where we calculate the Jaccard metric as:

$$J(M_i, M_j) = \frac{|M_i \cap M_j|}{|M_i \cup M_j|}$$

To avoid a pairwise $O(N^2)$ comparison of billions of tweets, we rely on minhashing to serve as an estimator for $J(M_i, M_j)$. For each element e in the set of character n-grams M_i we compute the hash $H(e)$. We then sort the resulting set of hashes and select the k minimum hashes from the ordered set, where k is a tunable parameter. The likelihood that two near duplicate messages share the same k hashes is proportional to the Jaccard similarity coefficient [3]. We ignore messages with fewer than k hashes, preventing us from treating simple tweets such as *lol http://...* as part of the same cluster.

In order to determine the optimal parameters for n and k given our text corpus, we perform a grid search over multiple variants of the minhash algorithm on a sample dataset of 19M tweets. We calculate the average Jaccard coefficient for all pairs of messages with the same minhash, excluding messages that are exact duplicates. We show our results

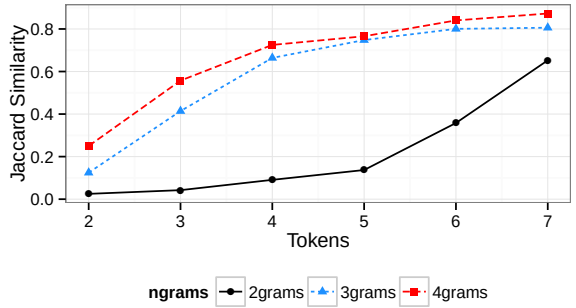

Figure 3: Grid search of the relation between the Jaccard similarity coefficient and pairs of tweets sharing the same minhash. We search over k, the number of hashed tokens to use, and n, the size of n-grams.

in Figure 3. We ultimately elect a minhashing algorithm set to $n = 3$ and $k = 7$ for a Jaccard similarity threshold of $\tau = 0.8$, which we find strikes the best balance between capturing wide variants of spam templates without colliding with benign, unrelated messages.

Given our final minhashing algorithm $H_{(k,n)}$, we transform every tweet in our corpus into a key-value pair $(H_{(k,n)}(m),\ tweet)$. We then group all tweets with the same minhash and treat them as a single cluster. Once grouped, we filter out any minhash clusters with fewer than a thousand tweets to reduce the volume of clusters that we must process and label. In total, this clustering approach yields 29,687 clusters.

3.2.2 Clustering on Duplicate URLs

Another canonical method for identifying the spread of ideas or spam campaigns in social networks is by clustering messages based on the URL they contain [13, 14, 28]. We follow previous approaches and transform each tweet into a key-value pair *(URL, tweet)*, subsequently grouping all tweets by their URL key. In the event a tweet contains multiple URLs, we create a key-value pair for each URL, allowing a tweet to be a member of multiple clusters. Once grouped, we again filter out any URL clusters with fewer than a thousand tweets. In total, this step yields 36,994 clusters.

3.2.3 Merging Clusters of Overlapping Users

In our final clustering step, we merge pairs of clusters with overlapping sets of users. This step is necessary to combine clusters of spam accounts that post multiple distinct messages and to merge duplicate minhash and URL clusters. Similar to how we merge near duplicate tweets, given the set of users for two clusters C_i and C_j, we merge the two clusters if $J(C_i, C_j) > \tau$. Again, because pairwise similarity is expensive to compute, we rely on minhashing the set of users for each cluster, performing a similar grid search to the one presented in Figure 3 to determine the optimal parameters. We ultimately elect $k = 2$ for a Jaccard similarity threshold of $\tau = 0.5$, which we determine via manual analysis best captures evolving stockpiles of spam accounts controlled by a single criminal without combining benign clusters with spam campaigns or merging distinct memes. Using these parameters, we combine clusters with the same user minhash and afterwards filter out clusters with fewer than two thousand distinct users. This step provides us with our final set of 16,206 clusters containing 254,366,938 tweets and 35,869,312 users.

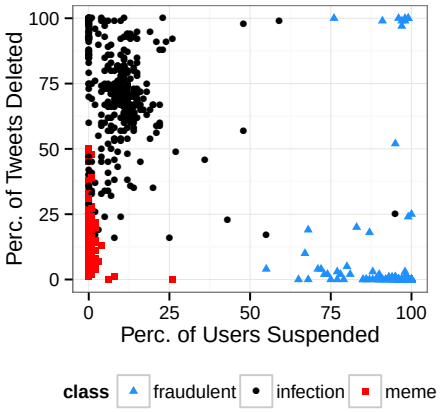

Figure 4: Scatter plot of actions taken by Twitter (and its users) and their relation to memes, infections, and fraudulent account behavior. We find that retroactive labeling of clusters provides a powerful tool for distinguishing the three classes of clusters.

3.3 Classification

In order to distinguish legitimate memes from infections or spam produced via fraudulent accounts, we develop a multi-class classifier based on observing retroactive actions taken by Twitter and its users, shown in Figure 1 (❸). As discussed in Section 2, there are no existing classifiers that both scale and accurately distinguish between compromised and fraudulent accounts, forcing us to develop a new technique. Our classifier hinges on the observation that compromised users—or Twitter acting on their behalf—frequently delete spam tweets posted via their account upon recognizing criminals have hijacked their credentials. Similarly, as previously observed [28], Twitter proactively suspends abusive accounts that disseminate spam or form an excessive number of relationships.

To capture both of these behaviors, we select a random sample of a thousand tweets and a thousand users from each cluster in our dataset and query Twitter's `users/show` and `statuses/show` API respectively. The response code returned by Twitter allows us to label users and tweets as *valid*, *deleted*, *suspended*, or now *private*. Due to delays in actions, we waited until one month after the final day of data collection (October 21, 2013) before querying the API endpoints. To verify the power of these features, we manually analyze a sample of 1,699 clusters and label each as either a meme (1,038 samples), infection (265 samples), or spam from fraudulent accounts (365 samples). We plot each labeled cluster as a function of the percentage of suspended users versus the percentage of deleted tweets, shown in Figure 4. The three non-overlapping clusters indicate that our retroactive feature set provides a strong starting point for segmenting the three classes of clusters.

3.3.1 Training

To determine an optimal hyperplane that separates our three classes, we train a multi-class logistic regression using 10-fold cross validation seeded with our manually labeled dataset of 1,699 clusters. We represent each cluster as a

feature vector that includes the ratio of sampled tweets and users that are valid, deleted, suspended, and private. Our feature vectors also include the fraction of tweets in each cluster that are retweets, the average tweets in the cluster per user, the number of distinct sources used to generate all tweets within the cluster (e.g., web, TweetDeck, Android, etc.), and finally the number of distinct languages appearing in the cluster, as determined by users self-reporting their language via their Twitter profile. Twitter embeds these latter features within our streaming dataset, requiring us to perform no additional API calls.

The resulting classifier has a 99.4% accuracy and a weighted average false positive rate of 0.5%, with a bias towards considering infections and fraudulent account activity as memes when the classifier is uncertain.[2] The most important features from the classifier's perspective are the ratio of suspended users for detecting clusters generated from fraudulent accounts, while the ratio of deleted tweets and the number of distinct languages are the best features for detecting large-scale compromise.

3.3.2 Labeling

Once trained, we apply our classifier to every cluster in our dataset. In total, we identify 10,792 benign memes containing 129 million tweets, 2,661 infections containing 80 million tweets, and 2,753 spam campaigns produced by fraudulent accounts containing 43 million tweets. The relatively small number of tweets in memes (1.4%) compared to the size of our initial dataset is consistent with previous results by Goel et al. [14] which found that the majority of content posted on social networks is never re-shared. As such, our analysis is biased towards only successful memes that reach thousands of users as well as large-scale spam campaigns.

3.4 Graph Crawling

The penultimate step in our data pipeline fetches the social graph for all of the accounts that belong to a cluster. While relationships in Twitter are directed, we are only interested in egress pathways that allow information to flow out from users to their followers. We enumerate these pathways by querying the followers/ids API endpoint for every userid in our dataset, collecting a total of 18,860,823,344 edges. We note that Twitter prevents any access to the social graph of suspended accounts. When this occurs, we flag the account as suspended and omit the account for graph measurements, passing the label along to the final stage (❺) where we label individual users. As a result, we restrict our graph analysis in Section 5 to compromised and uninfected (legitimate) accounts.

Because we delay graph crawling until a month after we cease collecting the tweet stream, there is a 1–11 month period during which an account's social graph may have evolved. To understand any bias this introduces, we select a random sample of 100,000 users appearing in clusters and compare changes in their follower graph over a 4 month period. We find that a median user loses 6 of their original followers while gaining 14 new followers (growing 17%) during this period, adhering to previous findings that social networks become more connected over time [18]. As such, our *post facto* graph collection will overestimate the number of followers who may have been exposed to a meme or spam tweet.

[2]We believe this is optimal as to prevent our analysis from overestimating the number of compromised or fraudulent accounts.

Measurement	Value
Meme clusters	10,792
Compromise clusters	2,661
Fraudulent account clusters	2,753
Meme participants	17,312,989
Compromised victims	13,899,907
Fraudulent accounts	4,656,416
Meme tweets	129,812,284
Spam tweets via compromised accounts	80,898,061
Spam tweets via fraudulent accounts	43,656,593

Table 1: Summary of our dataset after clustering and labeling.

3.5 User Labeling

The final stage in our pipeline labels individual users as benign, compromised, or fraudulent for the purposes of account-based measurements. We derive user labels based on each of the clusters a user participates in, selecting the maximum label from the following cluster label ordering:

$$\text{meme} \prec \text{infection} \prec \text{fraudulent}$$

This ordering captures the possibility that compromised users post tweets belonging to both memes as well as infections, while fraudulent accounts can inject content into popular memes, generate their own spam campaigns, or seed infection chains. Our final user labeling approach considers an account to be fraudulent if it is either suspended—as determined by a graph API call (❹)—or if it ever participates in a cluster classified as fraudulent. Similarly, if a legitimate user is ever compromised, when we compare uninfected users to compromised users, we treat the user as strictly compromised. All said, our dataset contains 4,656,416 fraudulent accounts and 13,899,907 compromised accounts. A final summary of our dataset can be found in Table 1. We caution that these are only lower bounds and do not encompass all possible abusive behavior on Twitter (e.g., follow and favorite spam, which will not appear in Twitter's streaming API; small scale spam clusters filtered by our thresholds; or compromise campaigns propagating and monetizing solely through direct messages).

3.6 Sampling Error

Our collection methodology introduces two forms of error. First, when we discuss the size of clusters or their rate of growth, we will likely underestimate their true values due to our sampling only about 61% of all tweets with URLs. Similarly, because sampling omits users and tweets that should be part of a cluster, any graph-based measurements we conduct that treat clusters as information diffusion processes may exhibit skew [9]. In particular, if information (such as retweets spreading) diffuses from user $u_a \rightarrow u_b \rightarrow u_c$, if u_b is omitted from our sample, we will incorrectly associate both u_a and u_c as progenitors of the process as opposed to the correct observation that u_c was influenced by u_a. As such, we restrict ourselves to comparing relative differences between diffusion processes (where any errors introduced by sampling should be consistent) rather than speaking in terms of absolute values.

4. ANALYZING HIJACKED ACCOUNTS

In this section we explore which populations of users are most vulnerable, characterize the impact of large-scale outbreaks on the Twitter ecosystem, and examine the mechanisms that criminals use to puppet compromised accounts.

4.1 Vulnerable Populations

Compromise is a systemic threat to all users, irrespective of savviness or geographic distribution. To illustrate this point, we examine five basic metrics of users: an account's *maturity*, *followers*, *followings*, *tweet count*, and self-reported *language*. We compare each of these properties against legitimate users participating in memes as well as with a random sample of 500,000 users selected uniformly throughout our collection period.

4.1.1 Maturity

We measure an account's maturity as the time between an account's creation up to its first tweet appearing in our dataset, effectively measuring how long an account exists prior to our analysis or its first tweet. Our results, shown in Figure 5(a), indicate that compromised accounts follow the same age distribution as uninfected users, having existed for a median of 1.5 years before we start logging their activity. In contrast, 50% of fraudulent accounts are less than a month old before we begin monitoring their activity—likely due to the heavy churn rate of fraudulent accounts due to regular suspension by Twitter and account pre-aging performed by criminals [28].

4.1.2 Followers, Followings, and Tweet Count

Twitter embeds a user's follower count, following count, and total statuses posted thus far inside every new tweet. As we receive multiple tweets over time, we measure a users follower count as the maximum value appearing in any of our clustered tweets, repeating the process again for followings and tweets. We show our results in Figure 5(b)–(d) respectively. We find that 50% of fraudulent accounts have fewer than 10 followers (users who receive an account's content) and 80% have fewer than 10 followings (users they receive content from). In contrast, compromised users have a median of 100 followers and 58 followings, which is slightly fewer compared to a random sample of users. Similarly, we find that compromised users are also less active at tweeting, with 50% of compromised accounts having fewer than 200 statuses compared to other legitimate users who have a median of 1,000 tweets. Paired with fewer followers and followings, compromised users appear to be less emphatic in their Twitter usage. Nevertheless, criminals are able to hijack accounts belonging to nascent, casual, and core users.

4.1.3 Global Diversity of Compromised Users

Language barriers and the absence of attackers targeting victims within certain geographic regions may cause localized infections as opposed to systemic outbreaks. To understand whether compromised accounts are uniformly distributed throughout Twitter, we aggregate the self-reported language of each account[3] and then compare the popularity of languages between compromised users and a random sample of 500,000 Twitter users who serve as a baseline for the language distribution of Twitter. Our results, presented in Table 2, show that compromise is a global phenomenon. English users are far and away the largest source of victims, accounting for 64% of all compromised accounts. This represents a 22% increase over the general frequency of English speakers as derived from our random sample. Turkish, Indonesian, and Italian are the most overrepresented languages in our ranking, while all other languages exhibit lower than expected compromise rates.

[3]Geolocation data is not available on a per-user basis, so we use language as a proxy metric for geographic distribution.

Rank	Language	Popularity	Divergence
1	English	64.4%	22%
2	Spanish	7.7%	-45%
3	Japanese	7.2%	-37%
4	Turkish	4.7%	76%
5	Indonesian	4.2%	94%
6	Arabic	2.0%	-49%
7	French	1.7%	-19%
8	Russian	1.6%	-29%
9	Italian	1.6%	50%
10	Portuguese	1.5%	-60%

Table 2: Top 10 languages spoken by compromised users, their overall popularity, and their divergence from the expected value given Twitter's underlying language distribution.

4.2 Impact of Compromise

Compromise is more than just a threat to users. Infections also impact web services as a whole, degrading core metrics such as user retention and engagement. We examine three facets of the damage incurred by compromise: the duration a victim loses control over their account, the likelihood a user continues using Twitter after becoming infected, and finally whether a user's social connections disengage from a victim. Our findings indicate that even brief compromises correlate with users abandoning their Twitter account and losing friends.

4.2.1 Compromise Duration

We measure the duration of compromise as the number of distinct dates a user posts any tweets falling into cluster labeled as an infection. Given that criminals may control a victim's account for long periods, but choose to stockpile credentials until the time of a spam campaign, this measurement is strictly a lower bound. We find that 60% of compromises last only a single day, while 90% last fewer than five days.

To understand how quickly users react to unwarranted activity in their timeline, we measure the delay between a criminal posting a spam tweet to a victim's account and that tweet's deletion. We determine the fine-grained timestamps of a tweet's removal based on a `delete` events appearing in the `statuses/sample` stream over a 10-month period, whereby Twitter notifies API consumers to strike a tweet from public display. Due to sampling, we are limited to 187,133 delete events associated with spam tweets posted to compromised accounts and 46,169 delete events tied to non-spam content. We find the median reaction time of victims (or Twitter) that delete spam tweets is under one hour, while 90% of spam tweets are deleted within 3.5 days. In contrast, when user's opt to erase their participation in a meme, they do so in a median of 5 days. This demonstrates that users (or

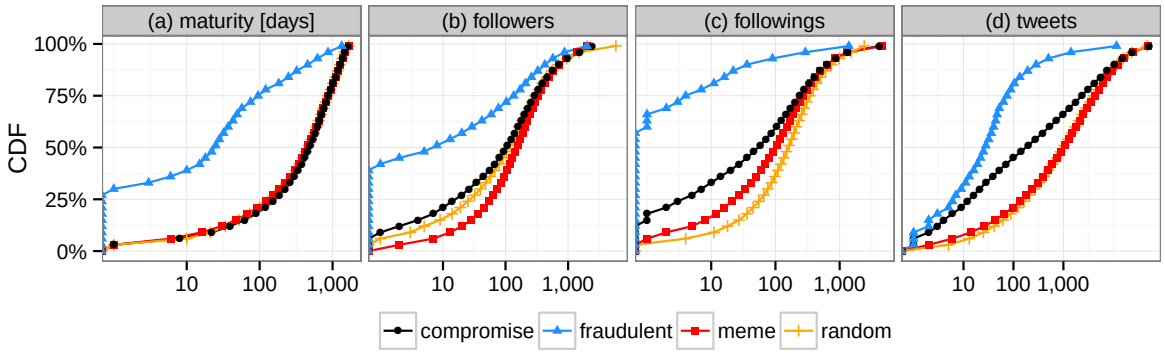

Figure 5: Basic properties of fraudulent accounts, compromised accounts, users participating in memes, and a random sample of 500,000 Twitter accounts. Compromised users are less active than other legitimate or meme users, but nevertheless distinct from fraudulent accounts.

their friends, or Twitter) are quick at policing unwarranted activity, minimizing the duration that criminals have access to a victim's account.

4.2.2 *User Retention*

While compromise may be brief, it strongly correlates with whether a user returns to Twitter after an action—such as a password reset—is taken to wrest control of the account from criminals. To measure this effect, we fetch the latest tweet for every compromised user two months after our collection concludes, repeating the process for a random sample of 500,000 users selected uniformly throughout our collection period. We then measure the time between each account's last (possibly spam) tweet up to the current time, the results of which we show in Figure 6. We find that only 60% of compromised users were active in the last 30 days compared to 83% of random users.

Given that compromised users are more casual than a random sample of Twitter users—as explored in Section 4.1—we refine our analysis one step further. We find that 21% of compromised users never tweet again after we observe their last spam tweet, compared to only 3% of random users. If we broaden this restriction slightly, 40% of compromised users tweet fewer than five times after their infection concludes, compared to only 7% of random users. While we cannot draw definitive conclusions as these results are only correlations, it is possible that users abandon their accounts due to lack of understanding of the account recovery process; not having a valid email or phone number to send recovery codes to; frustration with Twitter; or embarrassment.

4.2.3 *Stymied Engagement*

Once criminals compromise a victim's account, they can expose all of the victim's followers to a range of spam and abuse. We measure how a victim's followers react to compromise, comparing the number of followers a victim has at the onset of an infection versus their current follower count, as determined by an API call for each victim's latest follower count. Again, we compare this to a random sample of 500,000 Twitter users, where we measure the difference in followers at the time the user appeared in our data collection pipeline versus an updated count retrieved from Twitter's API.

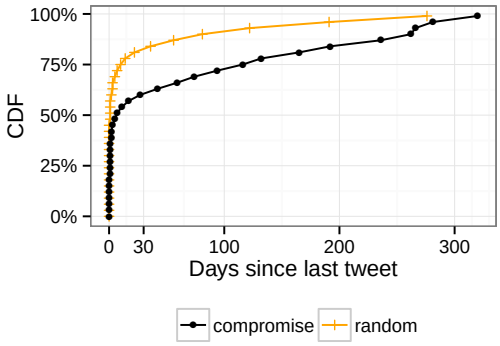

Figure 6: User retention of victims post-compromise compared to a random sample of Twitter users. We find 40% of compromised accounts were not active in the last month compared to 17% of random accounts.

Our results, shown in Figure 7, indicate that 57% of compromised users lose followers post-compromise compared to only 18% of random users. Again, we cannot conclusively determine whether this is a direct result of compromise. One alternative explanation is that compromised victims were participants in a fake follower scheme [25], after which other victims cleaned up their social connections upon becoming uninfected (or voluntarily leaving the program), thus reducing the follower counts of all parties involved. Whichever the conclusion, it is clear that compromised users have a higher likelihood of becoming more isolated from the rest of Twitter, stymieing their future engagement.

4.3 Controlling Hijacked Accounts

Criminals author spam tweets from hijacked accounts in one of two ways: *directly* with control of a victim's username and password, browser, or cookie, or alternatively via an *application* with a valid OAuth token (approved by either the victim or the criminal; we cannot distinguish which). We observe that 30% of spam tweets sent via a victim's account originate from the web and mobile sites where criminals require direct access. Miscreants generate the remaining 70% of tweets through long tail of over 9,900 applications. We explore some of the most popular applications used to control compromised accounts further in Section 6.

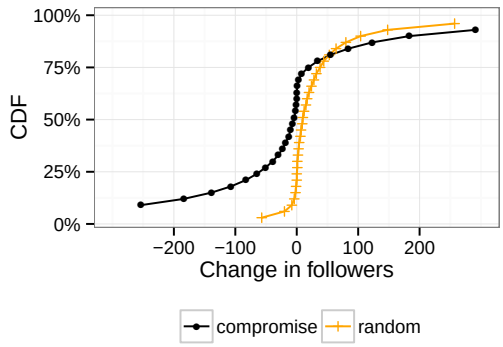

Figure 7: Change in follower counts of victims post-compromise compared to a random sample of Twitter users. We find 57% of compromised accounts lose relationships compared to only 18% of random users.

In contrast, spammers operating fraudulent accounts eschew any requirement of installing an application, authoring 94% of spam tweets via the web and mobile sites. Users post legitimate content on the other hand—measured in terms of both memes as well as a random sample of tweets selected uniformly throughout our collection period—65% of the time via clients owned and operated by Twitter (e.g., Twitter for Android, Twitter for iOS, TweetDeck) and other sanctioned cross-posting platforms such as Tumblr and Facebook.

Our findings indicate that platform abuse via the API contributes substantially to the control of compromised accounts. Assuming these applications are unwittingly installed by victims (as opposed to criminals controlling their credentials), improved API safeguards such as detecting anomalous fluxes in application installs as well as near-duplicate content being posted by an application can reduce the spread of compromise.

5. SOCIAL NATURE OF COMPROMISE

A critical question in relation to account hijacking is how criminals obtain access to a victim's account. We find evidence that infections are dominated by *social contagions*—phishing and malware campaigns that propagate along the social graph, abusing the trust that users place in their friends. We examine how this trust influences the rapid spread of compromise and how criminals bootstrap contagions that fan out to millions of users.

5.1 Social Contagions

We measure the connectivity of infected users and meme participants to understand whether compromise, like memes, spreads along the social graph. Given our crawl of the Twitter graph $G(V, E)$, we denote a victim as a *singleton* if no edge exists between the victim and another user infected by the same contagion (e.g., another user in the same cluster).

We find that 88% of contagions exhibit connectivity between victims, where an average of 56% of compromised users have at least one neighbor that is also compromised. These contagions account for 95% of all spam tweets sent by compromised accounts. As we discussed in Section 3.6, our down-sampled dataset may result in users (and thus their relationships) being omitted, leading to a higher estimate of singleton infections. To understand this bias, we compare the connectivity of retweeted memes, where we find 76% of participants share an edge with a participating friend. Assuming all retweets spread via the social graph and abide the same sample rate as contagions, this would indicate an average of 0.56/0.76, or 74% of compromised users share at least one relation with another victim.

The remaining 12% of compromised clusters are composed entirely of singleton infections. These compromises may be tied to password guessing, database dumps, or external contagions, as discussed in Section 2. We caution that it may be possible for external contagions, such as those spread through email social graphs, to reflect as though they spread along the Twitter social graph. Similarly, natural homophily [19] in friends using the same web services may result in database dumps exhibiting social connectivity. As such, we cannot definitively say whether the majority of compromise spreads within Twitter, but there is a strong tendency for victims to be connected.

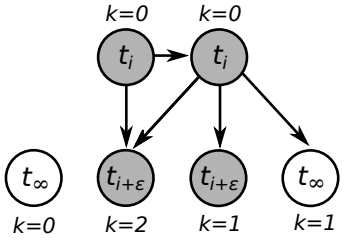

Figure 8: Example of calculating the number of incoming edges through which an infection could propagate for a synthetic graph.

5.2 Influence of Compromised Neighbors

Neighbors connected to a user—either in real life or in an online social network—influence that user's decision-making. This influence manifests in *information cascades* including the adoption of online memes [1, 2, 5, 23], health and lifestyles choices [6], and the spread of biological infections [21]. We observe that compromise in online social networks abides by the same process, where users with infected friends appear more likely to fall prey to malware and phishing scams as a result of the trust they place in their social connections.

To gauge the influence of compromised neighbors, we measure the probability $p(i|k)$ that a user becomes infected given they have k previously infected neighbors. Figure 8 shows a sketch of our approach—adopted from social science techniques for measuring the virulence of memes [2, 7]. To start, we label all of the vertices V in our graph crawl $G(V, E)$ as either *infected* (grey in our example) or *uninfected* (white). We treat each cascade independently, running this process once per each cluster in our dataset. Next, we mark each node with its infection time t_i—the timestamp of the first spam tweet a victim posts belonging to the contagion in question—or ∞ if the node is not infected. Given a directed edge $E(s, d)$ between a source s and destination d, we consider the source to be an *influencing neighbor* if s is infected and $t_{(i,s)} < t_{(i,d)}$. This metric captures whether a neighboring infection could spread to the destination or whether that destination was already infected. Finally, we count the total number of influencing neighbors k for each

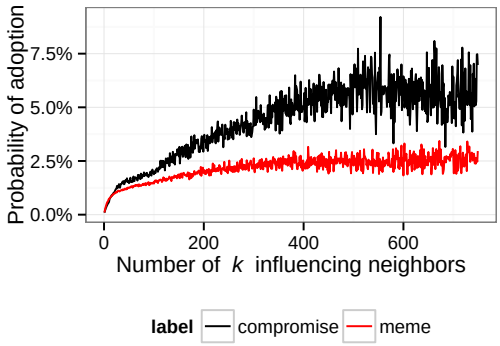

Figure 9: Likelihood of a user joining a cascade given k-neighbors already participate in the cascade. Both compromise and memes spread via a social process where friends influence a user's decisions.

user in V and calculate $p(i|k)$ for all possible k. In our example, $p(i|k=1)=0.5$, while $p(i|k=2)=1$. To analyze all contagions at once, we calculate the average $p(i|k)$ across contagions, weighting each contagion equally independent of size. To serve as a comparison, we repeat this same process for all of the memes in our dataset.

Our results, shown in Figure 9, demonstrate that compromise is more effective at spreading as more of a user's neighbors fall victim to attacks. We find that the probability of a victim becoming compromised increases from 0.1% with only one neighboring infection to 1% when a user has 20 neighboring infections. This behavior is nearly identical to memes in the early stages, indicating that compromised victims, like meme participants, are influenced by their peers. In both cases, the influence of friends eventually tapers off to a constant likelihood—a phenomenon previously observed by social scientists [2, 23]—but we find that compromised peers have stronger lasting power.

Our results highlight that compromise occurs as a social process where users in social networks are vulnerable to the bad decision-making of their neighbors. In contrast, if large-scale compromise was more frequently related to database breaches effecting millions of users or password guessing, we would expect the likelihood of a victim's compromise to be independent of their number of compromised peers. Consequently, we argue that early outbreak detection in social networks is critical as it both prevents neighbors from spreading their infection as well as restricts infections to their nascent stage before they become 10–100 times more effective at spreading.

5.3 Seeding Compromise Diffusions

If compromise is a social contagion, there remains a question as to how criminals bootstrap the initial cascade effect. We find that 35% of compromise campaigns rely on more than 100 fake accounts to start the infection process. Of these accounts, 25% tweet within 24 hours of the onset of the compromise campaign. (The remaining accounts join at a later date, presumably to re-seed new infection chains.) Our dataset cannot elucidate how the attacker started the remaining 65% of compromise campaigns. One hypothesis is that criminals obtain a small number of compromised accounts via targeted attacks or by directly purchasing them from the underground, in turn compromising the victim's friends to start a cascade.

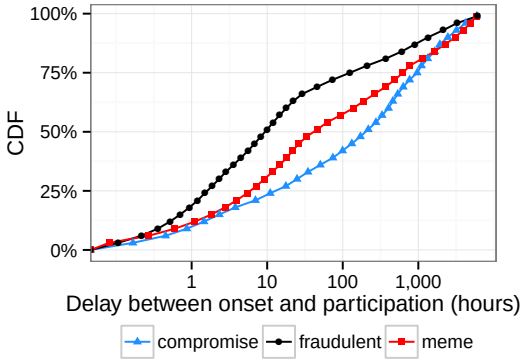

Figure 10: Delay between the onset t_0 of a cluster and each user's participation at t_i.

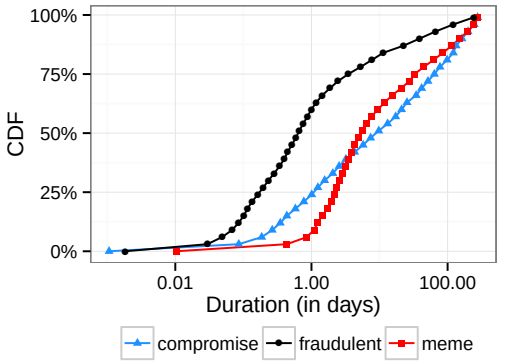

Figure 11: Duration of clusters measured from the first clustered tweet t_0 to the last tweet t_n.

5.4 Rate of Spread

To understand how rapidly contagions spread, we compare the delay between each compromised user's first spam tweet t_i to the onset of the contagion t_0, estimated as the first tweet in the cluster. For comparison, we repeat this process for every meme participant and fraudulent account. Our results are shown in Figure 10. We find that compromise is the slowest process, with 30% of victims posting in the first day compared to 44% of meme participants. Spam campaigns reliant on fake accounts are the most condensed, with 65% of accounts posting within a day of the campaign's onset. The long tail for fraudulent accounts results from fresh accounts joining campaigns over time, with multiple campaigns being merged due to overlapping participants. Our findings indicate that, contrary to Internet worms [24], social contagions are slow moving due to the requirement that victim's interact with harmful content.

5.5 Campaign Duration

Despite the quick reaction time of compromised users to unwarranted content in their feeds (discussed in Section 4.2), contagions are nevertheless able to spread and last for multiple days. The median compromise cluster—shown in Figure 11—measured from its first tweet t_0 to its last tweet t_n, is 9 days in duration. For comparison, memes last a median of 5 days and spam campaigns conducted via fraudulent accounts only last 1 day. The long duration of compromise campaigns indicates that even batch jobs for detecting compromise contagions can reduce the number of victims im-

Scam Type	Contagions	Victims	Spam Tweets	Duration	API Sources	Distinct URLs
Weight Loss Supplements	221	4,758,207	11,552,045	229 days	439	2,158,837
Gain Followers	779	3,704,314	42,082,699	234 days	4,344	3,270,720
Survey Leads	1	994,563	2,512,330	31 days	1	1,219

Table 3: Summary of the most prevalent monetization techniques spammed by criminals via compromised accounts.

pacted. If Twitter detected and remedied social contagions within 24 hours, it would result in 70% fewer compromises.

6. MONETIZING HIJACKED ACCOUNTS

Profit is the ultimate goal of account hijacking. We explore the three dominant monetization strategies that criminals rely on once they gain access to compromised credentials. Table 3 shows a summary of each strategy and its impact on Twitter. These monetization schemes are similar to previous Twitter spam campaigns that rely on fraudulent accounts [28], though the challenge of reaching an audience is vastly simplified by account hijacking.

6.1 Weight Loss

Easy weight loss nutraceuticals were the most prolific ploys that criminals used to monetize compromised accounts. We identify 221 campaigns in our dataset containing roughly 4.7 million unique victims (34% of all compromised accounts). Despite the short lifetime of individual campaigns (an average of 6 days), weight loss schemes persisted for 229 days—nearly the entirety of our data collection period. We find that criminals relied on stolen credentials to author spam tweets; miscreants composed 98% of the advertisements for weight loss via Twitter owned and operated clients where a username and password is required. The largest single contagion in this set used 1.1 million compromised accounts to advertise "its been 2 weeks and i lost 20 lbs thanks to garcinia, try it for free...", linking to nearly 70,000 distinct URLs over a 23 day period.

Such scams generate a profit through visitors voluntarily providing criminals their credit card details and subsequently purchasing garcinia, green coffee, acai berry, raspberry ketone, or some other nutraceuticals (often advertised with a misappropriation of the "Dr Oz" brand). Merchants fulfilling these orders have recently come under target by the FTC for deceptive practices [31], while graymarket advertisers on Facebook have also had their ads pulled for running afoul of advertisement rules [11]. The reliance of criminals on compromised accounts to reach a wide audience can thus be viewed as merely an evolution in the long battle against weight loss scams.

6.2 Gain Followers & Retweets

Fake follower schemes where victims unwittingly (or willingly) provide their credentials to criminals in return for purportedly gaining more followers (or alternatively retweets) are the second largest source of compromises on Twitter. We identify 779 of these schemes that netted roughly 3.7 million users (27% of all compromised accounts). Victims advertise "easy way to get free followers..." and "ücretsiz takipçi kazan" (Earn free followers), with the most popular advertisements appearing in English (47%), Turkish (22%), and Indonesian (19%). Contrary to weight loss, 88% of spam tweets were authored via a long tail of 4,343 OAuth applications and the remaining 12% via the web. These applications include Retweetlr, BestFollowers App, and a slew of throw away OAuth credentials that criminals automatically generate to withstand Twitter disabling their app.

Fake follower schemes both spread and generate a profit by victims installing their application. Criminals use compromised accounts to advertise services such as *http://followrush.org/* where miscreants can buy 20,000 followers for $40 and 5,000 retweets and 5,000 favorites for $40—all of which are sourced from the compromised accounts. Alternative monetization strategies rely on tiered pricing between *free membership* and *premium membership* to fake follower rings [25]. One example of this appearing in our dataset is PlusFollower, where free members must follow all premium users and send a promotional tweet as frequently as every 4 hours, in return receiving an unspecified number of followers. Premium members on the other hand pay a fee (starting at £10) to gain followers, in turn avoiding the requirement to follow other users or advertise the service.

Fake follower rings on Twitter have persisted since 2010, with criminals advertising both "real accounts" and fraudulent accounts as the source of follows [16, 25, 30]. While perhaps some victims willingly participate in these scams, we argue that they should nevertheless be considered compromised. In particular, users lose control of their accounts and have no oversight capability over the subsequent spam advertisements, retweets, favorites, and follows that occur. Even if victims wish to leave the service they must go through the same mechanism to restore account control as compromised victims. Finally, the monetization of these participants is clearly criminal and in violation of Twitter's Terms of Service.

6.3 Lead Generation

The final monetization strategy we highlight is lead generation, where criminals entice victims into filling out surveys for a nominal payment or trick victims into paying a "one-time fee" before they can be paid for their work. We find only one contagion relying on this approach, but it alone consisted of nearly 1 million victims and lasted 31 days. Criminals used hijacked credentials to tweet an automated template loosely matching "Sweeeet!! I earned $157.18 this week filling out a couple of surveys", with all posts originating from the web where a username and password is required. Surprisingly, all of the URLs that criminals used to monetize clicks lead to Facebook applications that are no longer operational, hinting the contagion may have existed both on Twitter and Facebook. While sleuthing through the broken Facebook applications, we found an embedded iFrame that linked back to at least one live site, *getcashforsurveys.com*, which advertises a quick cash program through completing surveys with an upfront enrollment fee of $74.

7. SUMMARY

Our work illuminates the threat of large-scale compromise in social networks, and in concrete terms we identify 13 mil-

lion hijacked victims on Twitter. Our measurements capture the underlying social component of compromise and how it operates at scale. This includes: the human cost of 21% of victims losing access to their account and 57% of victims becoming more isolated from friends; the ability of miscreants to generate social cascades that propagate as virulently as media sensations, with the single largest contagion infecting 1.1 million users; and the finding that user vulnerability to compromise appears independent of user savviness.

Our results indicate that compromise is dominated by *social contagions*—phishing and malware campaigns that prey on user trust—as opposed to weak passwords or lax site security. The underlying social process that drives compromise mirrors that of information diffusion for benign memes and viral content. Consequently, a user with 20 compromised neighbors is 10x more likely to become compromised compared to a user with one compromised neighbor.

To combat the threat of wide-spread hijacking, we observe that in addition to better account-hijacking detection signals, existing victims can serve as early detectors for attacks. We find that a median user deletes an errant spam tweet posted by a hijacker within one hour of its appearance. If Twitter detected social contagions within 24 hours of their outbreak via correlated deletion events (similar to the detection scheme outlined in this paper), they could protect 70% of future potential victims. In particular, we emphasize that the centralized control afforded to online social networks provides an opportunity to inoculate victims to arrest the spread of contagions in a way that has never been possible for Internet worms.

8. ACKNOWLEDGMENTS

We would like to thank James Fowler for his helpful comments on our research and its development. This work was supported in part by the National Science Foundation under grant 1237265, by the Office of Naval Research under MURI grant N000140911081, and by a gift from Google. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors.

9. REFERENCES

[1] Eytan Bakshy, Brian Karrer, and Lada A Adamic. Social influence and the diffusion of user-created content. In *Proceedings of the 10th ACM conference on Electronic commerce*, 2009.

[2] Eytan Bakshy, Itamar Rosenn, Cameron Marlow, and Lada Adamic. The role of social networks in information diffusion. In *Proceedings of the 21st international conference on World Wide Web*, 2012.

[3] Andrei Z Broder. On the resemblance and containment of documents. In *Compression and Complexity of Sequences 1997. Proceedings*, 1997.

[4] Chris Brook. Github resets users' passwords following brute force attack. *http://threatpost.com/github-resets-users-passwords-following-brute-force-attack/102983*, 2013.

[5] M. Cha, H. Haddadi, F. Benevenuto, and K.P. Gummadi. Measuring User Influence in Twitter: The Million Follower Fallacy. In *Proceedings of the 4th International Conference on Weblogs and Social Media*, 2010.

[6] Nicholas A Christakis and James H Fowler. The spread of obesity in a large social network over 32 years. *New England Journal of Medicine*, 2007.

[7] Dan Cosley, Daniel P Huttenlocher, Jon M Kleinberg, Xiangyang Lan, and Siddharth Suri. Sequential influence models in social networks. In *Proceedings of the International Conference of Weblogs and Social Media*, 2010.

[8] Anupam Das, Joseph Bonneau, Matthew Caesar, Nikita Borisov, and XiaoFeng Wang. The tangled web of password reuse. In *Symposium on Network and Distributed System Security (NDSS)*, 2014.

[9] Munmun De Choudhury, Yu-Ru Lin, Hari Sundaram, K Selcuk Candan, Lexing Xie, and Aisling Kelliher. How does the data sampling strategy impact the discovery of information diffusion in social media? In *Proceedings of the International Conference of Weblogs and Social Media*, 2010.

[10] Manuel Egele, Gianluca Stringhini, Christopher Kruegel, and Giovanni Vigna. COMPA: Detecting Compromised Accounts on Social Networks. In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2013.

[11] Facebook. Guidelines for advertised products & services. *https://www.facebook.com/help/399392800124391/*, 2014.

[12] Hongyu Gao, Yan Chen, Kathy Lee, Diana Palsetia, and Alok Choudhary. Towards online spam filtering in social networks. In *Symposium on Network and Distributed System Security (NDSS)*, 2012.

[13] Hongyu Gao, Jun Hu, Christo Wilson, Zhichun Li, Yan Chen, and Ben Y Zhao. Detecting and characterizing social spam campaigns. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*. ACM, 2010.

[14] Sharad Goel, Duncan J Watts, and Daniel G Goldstein. The structure of online diffusion networks. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, 2012.

[15] C. Grier, L. Ballard, J. Caballero, N. Chachra, C.J. Dietrich, K. Levchenko, P. Mavrommatis, D. McCoy, A. Nappa, A. Pitsillidis, et al. Manufacturing compromise: The emergence of exploit-as-a-service. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, 2012.

[16] C. Grier, K. Thomas, V. Paxson, and M. Zhang. @spam: The Underground on 140 Characters or Less. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, 2010.

[17] Brian Krebs. Adobe breach impacted at least 38 million users. *http://krebsonsecurity.com/2013/10/adobe-breach-impacted-at-least-38-million-users/*, 2013.

[18] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, 2005.

[19] Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a feather: Homophily in social networks. *Annual review of sociology*, 2001.

[20] Fred Morstatter, Jurgen Pfeffer, Huan Liu, and Kathleen M Carley. Is the Sample Good Enough? Comparing Data from Twitter's Streaming API with Twitter's Firehose. In *Proceedings of the International Conference of Weblogs and Social Media*, 2013.

[21] Mark EJ Newman. Spread of epidemic disease on networks. *Physical review E*, 2002.

[22] Nicole Perlroth. Lax Security at LinkedIn Is Laid Bare. *http://nyti.ms/1fRQIl4*, 2012.

[23] Daniel M Romero, Brendan Meeder, and Jon Kleinberg. Differences in the mechanics of information diffusion across topics: Idioms, political hashtags, and complex contagion on Twitter. In *Proceedings of the 20th international conference on World wide web*, 2011.

[24] Stuart Staniford, Vern Paxson, and Nicholas Weaver. How to 0wn the Internet in Your Spare Time. In *USENIX Security Symposium*, 2002.

[25] Gianluca Stringhini, Gang Wang, Manuel Egele, Christopher Kruegel, Giovanni Vigna, Haitao Zheng, and Ben Y Zhao. Follow the Green: Growth and Dynamics in Twitter Follower Markets. In *Proceedings of the 2013 conference on Internet measurement conference*, 2013.

[26] Fred Tanneau. Twitter hacked! 250,000 user accounts breached. *http://www.cnbc.com/id/100343530*, 2013.

[27] Ke Tao, Fabian Abel, Claudia Hauff, Geert-Jan Houben, and Ujwal Gadiraju. Groundhog day: Near-duplicate detection on Twitter. In *Proceedings of the 22nd international conference on World Wide Web*, 2013.

[28] K. Thomas, C. Grier, V. Paxson, and D. Song. Suspended Accounts In Retrospect: An Analysis of Twitter Spam. In *Proceedings of the Internet Measurement Conference*, November 2011.

[29] Kurt Thomas and David M. Nicol. The Koobface botnet and the rise of social malware. In *Proceedings of The 5th International Conference on Malicious and Unwanted Software (Malware 2010)*, 2010.

[30] C. Yang, R. Harkreader, J. Zhang, S. Shin, and G. Gu. Analyzing Spammers' Social Networks for Fun and Profit: a Case Study of Cyber Criminal Ecosystem on Twitter. In *Proceedings of the 21st International Conference on World Wide Web*, 2012.

[31] Alison Young. FTC takes action against deceptive weight-loss products. *http://www.usatoday.com/story/news/nation/2014/01/07/ftc-charges-deceptive-weight-loss-products/4354669/*, 2014.