A Rice Cooker wants to be my Friend on Twitter

Miranda Mowbray

HP Laboratories
HPL-2011-175

Keyword(s):
Twitter; security; pervasive computing

Abstract:
Pervasive computing devices are already using Twitter as a communication channel. In the future you may receive unwanted friend requests from inanimate objects. Even if you refuse them all, receiving frequent friend requests can be annoying, and may lead you to mistakenly refuse some welcome friend requests. There are some Twitter validation services which assist and partially automate the decision of which requests to accept; but these have limitations. Without improved validation services, the rise of pervasive computing devices on Twitter or similar networks may degrade the experience of the human users of these networks.
A RICE COOKER WANTS TO BE MY FRIEND ON TWITTER

By Miranda Mowbray, HP Labs Bristol (miranda.mowbray @ hp.com)


Abstract

Pervasive computing devices are already using Twitter as a communication channel. In the future you may receive unwanted friend requests from inanimate objects. Even if you refuse them all, receiving frequent friend requests can be annoying, and may lead you to mistakenly refuse some welcome friend requests. There are some Twitter validation services which assist and partially automate the decision of which requests to accept; but these have limitations. Without improved validation services, the rise of pervasive computing devices on Twitter or similar networks may degrade the experience of the human users of these networks.

Introduction

This paper discusses an issue that may arise from the combination of the accelerating growth in the number of pervasive computing devices, and the continued rise of social computing, in particular microblogging. The issue is that in the future you may receive unwanted friend requests from inanimate objects.

A Twitter user who wants to be sent in real time the tweets published by another Twitter user’s account, or to read private messages sent by that account, must first send a friend request to the account. The act of sending a friend request to a user’s account is also known as “adding” the user. A tweet sent in 2009 by Eugene Huo said [Tan, 2009]:

*Man, I gotta watch what I talk about on twitter... as soon as you mention something they find you... a rice cooker just added me.*

It was probably a company advertising rice cookers rather than an actual rice cooker that added Eugene Huo, but in the future it might well be a rice cooker. This paper will give some examples of pervasive computing devices that are already using Twitter as a communication channel. This raises the spectre of human users being bombarded with friend requests from inanimate objects with embedded computing systems.

In fact, unwanted friend requests from automated accounts are already a problem on Twitter, because many Twitter spammers use software that sends friend requests to a large number of users [Mowbray, 2010]. Users who accepted such a request might find themselves in unwanted communication with an automated account — whether this was the account of a pervasive computing
device or a spammer, or even both at once. Or their private tweets might be automatically harvested and used in ways that they did not wish. (One thing that authors of private tweets presumably do not wish their recipients to do is to copy the content of a private tweet, which is visible only to recipients whose friend requests the author has accepted, and re-send it as a public tweet visible to everyone. Meeder et al. have identified over 4.4 million occasions on which this has happened [Meeder et al., 2010]. However is not clear how many, if any, of the offending public tweets were sent by automated accounts rather than by human recipients.)

Even if users refuse every automated friend request, receiving frequent friend requests can be annoying in itself. It may also lead to users mistakenly refusing requests from some people (or objects) that they would in fact like to be friends with on Twitter.

To address this, some form of validation service is needed to help users to decide whether a requesting account is likely to be one that they wish to communicate with, and enable them to automatically refuse or accept requests from some classes of accounts. Unfortunately, the Twitter validation services currently available have some limitations. It follows that better identification tools are needed so that Twitter users have a better idea of who or what they are really communicating with. If we do not solve this problem there is a risk that the rise of pervasive computing devices on Twitter or similar networks will significantly degrade the experience of the human users of these networks.

The rest of the paper is structured as follows. Section 2 gives examples of pervasive computing devices currently using Twitter, and argues why such devices may interact with other Twitter users (either human or automated) rather than only publishing tweets. Section 3 gives more detail on Twitter spamming software that makes large numbers of automated friend requests. Section 4 briefly discusses limitations of some current Twitter validation methods. The conclusion suggests some lessons from this discussion for pervasive computing systems in general.

1. Pervasive Computing and Twitter

Cheng and Evans estimated in August 2009, based on a survey of 11 and a half million Twitter accounts, that 24% of all tweets were sent by rapidly-tweeting automated accounts [Cheng & Evans, 2009]. Much of this is marketing messages and spam, but Cheng and Evans’ report also mentions local weather stations tweeting meteorological information, and radio stations tweeting their playlists.

As observed by Kevin Slavin, sensor devices typically produce short bursts of data, either at regular intervals or when triggered by some event, and this - together with Twitter’s ease of use and widespread adoption - makes Twitter the obvious communication medium for these devices [Slavin, 2009].

Everyday objects that tweet include a toaster (@mytoaster), shoes (@ramblershoes), ovens (@bakertweet), and a cat flap (@GusAndPenny); the cat flap detects which cat is going out or in, and publishes a tweet with this information and a link to a photo taken by a camera triggered by the flap’s movement. Mattel Inc. sells dog tags that sense when your pooch moves or barks, and
translate this into tweets [Mattel, 2010]. An electronics company sells kits that enable your plants to tweet personalized messages when they need watering [SparkFun Electronics, 2011]. A house may tweet about its power consumption, water usage or room temperatures ([Torrone, 2009; Cellan-Jones, 2009], @ckhome). There is a home security system that sends public tweets about which doors are open – this is perhaps not a good idea.

The examples given so far are mostly of devices that send out data on Twitter but do not receive it. However, Twitter offers an easy-to-use channel through which pervasive computing devices can receive data as well as transmit it. It seems natural to use Twitter communication to allow the devices with Twitter accounts to interoperate. For example, the home security systems already on Twitter might be programmed to react to announcements from the police incident reporting systems already on Twitter (e.g. @shawneePD), or the plants already on Twitter might take information from the local meteorological services already on Twitter (e.g. @crondallweather) into account, in addition to their own moisture levels, when determining whether they need to tweet that they need watering.

Pervasive computing devices may also be able to gather useful information from human tweeters, including tweeters who may be strangers to the owner of the device. Information from human tweeters can form a rapid and flexible service in emergencies [Hughes & Palen, 2009]. Company service representatives are already using Twitter by following customers and other users who tweet words connected to their brands, and responding to any complaints [Torrone, 2009]; in the future these companies’ products themselves might use Twitter to automatically mine real-time information from customers that could be used to enhance the service provided by the product or proactively solve problems. The richness of human-generated tweets as a potential information source can be illustrated by the fact that measures based on Twitter activity have been found to predict the result of the most recent UK general election [Kiss, 2010] and box-office takings for Hollywood films [Asur & Huberman, 2010] better than the previous best prediction methods.

Moreover, the Twitter interface to an embedded system offers a potentially intuitive way for human users (especially remote users) to interact with a service. A playful example of this is @tweet_tree, the Twitter account of a Christmas tree whose lights change colour according to commands sent to it by Twitter users. Remote human users have also used Twitter to switch on and off a coffee machine [InstructablesTV, 2011], control a humanoid robot [ogutti, 2009], draw pictures in light on an LED-covered table [Mace, 2010], and to water plants which tweet when they are thirsty [Fahner, 2009]. The ideal way for a user to control an automated account is by sending it direct messages, which are private one-to-one messages that can be sent by a Twitter user to any of the user’s followers. It is possible to use public tweets instead (and in fact the examples just cited use this method), but this offers less privacy and also adds the communications to the Twitter search results, possibly distorting them.

Tweets are public by default, but individual Twitter users have the option of making all their tweets private, that is, only visible to their followers. Public tweets can be viewed using Twitter’s search tools. However, only followers of a Twitter account are sent public tweets from this account in real time, or can receive direct messages or private tweets sent from the account. To become a follower, a user sends a friend request to the owner of the account in question. If the owner accepts this request
rather than blocking the requester, the requester becomes one of the account’s followers. Thus, automated Twitter accounts that interact with other automated or human-operated accounts via direct messages need to send friend requests to these accounts.

An additional advantage of using direct messages rather than public tweets for the interaction is that this limits the Twitter accounts that can send commands to the embedded system. If the hosepipe in your garden is controlled by public tweets, then (in the absence of additional controls) any stranger with a Twitter account anywhere in the world could turn the hosepipe on until your garden is flooded. If instead the hosepipe is controlled by direct messages, it can only be turned on by accounts to which the hosepipe account has sent a friend request. Moreover, if any of these accounts misbehave, you can prevent them from sending any further direct-message commands to the hosepipe simply by logging into your hosepipe’s Twitter account and unfollowing them.

2. Follow-spamming

Data from 13,447 Twitter users collected in April-May 2010 showed that automated accounts using the API to access Twitter followed more users than human-operated accounts did [Mowbray & Andrade, 2010]. Some automated accounts follow over 10,000 users. Although some of these may be following back all the users that follow them, either as a symbolic politeness or to allow interaction via direct messages, the evidence from the collected data suggests that most of the automated accounts that follow large numbers of other accounts are follow-spammers.

Follow-spamming is a method of spamming on Twitter (and other social networks) which begins by sending friend requests to many users, hoping that some will accept the request and follow the spamming account back. If they follow back, the spammer automatically sends them a spam message, either in a direct message or a tweet. This message may for example be a marketing message, or entice the recipient to visit a malware-infected web page. Some spamming accounts target users to send friend requests to, based on the presence of key words in the users’ public tweets. The account that added Eugene Huo may have been an automated spamming account targeting users who tweeted the word rice. It appears from an investigation of all the spammers reported to Twitter in 7 days (reported in [Mowbray, 2010] ) that follow-spamming is the most common spamming method used on Twitter.

The adoption of follow-spamming by many spammers has resulted in some human Twitter users being sent annoyingly large numbers of friend requests. Users who accept a spamming account’s request and follow it back can unfollow it as soon as they realize that it is a spamming account – but by that time they will have received the spam message anyway. It is not necessarily easy to tell that a friend request is from a spammer, because the interface for most Twitter clients only supplies limited information about the requester at the time of the request. Moreover, human curiosity and the human tendency for social reciprocation mean that it is relatively common for Twitter users to accept all friend requests and follow back all the requesters. Indeed, to save effort, and precisely to avoid the annoyance of having to deal with large numbers of requests, many users set their Twitter clients to follow back automatically. In an experiment, about half of all the Twitter users that were sent a friend request by a Twitter account created especially for the experiment followed back the requesting account [Cosoi & Cosoi, 2009].
3. Twitter Validation Services

Because of the presence of spammers on Twitter, there is a need to validate accounts that make friend requests, to assist users to decide whether to block them or follow back. The expected massive increase of pervasive computing in the next few years, together with the possibility that many of the pervasive computing devices will wish to follow human (and automated) Twitter users in order to receive data from them, may exacerbate this problem.

1.1 Some Twitter Validation Methods from Academic Papers

Most academic papers on identifying Twitter spammers look at spammers who use methods other than follow-spamming - for example, including trending topics in their tweets [Benvenuto et al., 2010], [Yardi et al., 2010] - or look at spammers identified just from their tweets, followers and friends [Wang, 2010], which may not detect follow-spammers. However, Lee et al. and Stringhini et al. used honeypots to collect follow-spammers, and investigated which features might be used to detect them [Lee et al., 2010; Stringhini et al., 2010]. The features that Lee et al. found to have better than low discrimination power were the average number of unique URLs per tweet, the average number of unique usernames per tweet, the average content similarity between a user’s tweets, text-based features extracted from tweets, the average number of URLs per tweet, and the account age (with young accounts being more likely to be spammers). Like Lee et al., Stringhini et al. considered the average number of URLs per tweet and a measure of tweet similarity; they also considered the total number of tweets sent, and two measures based on the number of followers of the account and the number of accounts followed. Stringhini et al. used these measures to detect 15,857 spamming accounts on Twitter, and co-operated with Twitter to get these accounts deleted.

Unfortunately, the effectiveness of most of these features in identifying spammers is vulnerable to changes in popular spamming software. Follow-spammers who use direct messages can send any pattern of tweets, for example some just copy the tweets of non-spammers. As Lee et al. observe, some follow-spammers only start spamming a while after setting up their accounts. Although poorly designed follow-spammers follow many accounts and gain few followers, follow-spamming accounts can (and many do) attain a high number of followers and a followers/followed ratio close to 1 just by making very large numbers of friend requests, counting on the fact that a percentage of users will accept the request and follow them back, and automatically unfollowing the users who do not follow them back. The follower and followed numbers for such spamming accounts are similar to those for popular non-spamming accounts that follow back their followers. Automated unfollowing is prohibited under Twitter’s automation rules [Twitter, 2011] because of its abuse by spammers.

1.2 Some Current Twitter Validation Services

The Twitter anti-spam apps TwitSweeper™ [Emerge2 Digital Inc., 2009-2010], StopTweet [Joi Company, 2009-2010], TwitChuck [TwitChuck, 2011], and the Clean Tweets Firefox extension [BLVD Status, 2009] identify spamming accounts as accounts that have tweets with certain properties, and/or a small account age (typically less than a day), and/or a small ratio of followers to followed accounts. As just noted, these features are vulnerable to changes in spamming software.
The average age of the Twitter spam accounts identified by Stringhini et al. was not one day but 31 days.

The Friendfilter service [Peri, 2009] scores Twitter accounts based on the number of mutual social contacts that the account has with the Friendfilter user, the recent tweet rate, the followers/followed ratio, “and a few other things”. New Twitter users may get similar scores to spammers. The number of mutual social contacts may be a helpful measure in the detection of spammers, however it might be defeated by spammers selecting Twitter users to send friend requests to by spidering over following relationships. A large-scale experiment on Facebook, sending friend requests to friends of users who have previously accepted a request, has shown that this could be a highly effective way for spammers to gain followers on social networks [Boshmaf et al., 2011].

The Tweepi app [Tweepi, 2011] is intended to give Twitter users information about their followers in general, rather than specifically to detect spammers. (It also incorporates other features to help Twitter users manage their accounts.) Although users can customize the statistics presented, the preset categories suggested by Tweepi for finding “quality” followers identify followers that tweet with high frequency, that retweet and are retweeted frequently, that have a large ratio of replies, that have a large ratio of followers to followed accounts, and that have URLs in a high percentage of their tweets. A spammer with more than one Twitter account could easily ensure that the accounts were in all these categories, and some of these categories (the high URL percentage, frequent retweets, and high tweeting frequency) actually favour spammers using certain spamming techniques over non-spammers.

The TwitBlock app identifies spammers by the number of TwitBlock users that have blocked them, together with some heuristics [Whitlock, 2009]. The shared blacklist looks promising, but TwitBlock’s heuristics are vulnerable to changes in spamming software, and have identified some users with more passive use of Twitter as spammy. Another app operating a shared blacklist is SocialOomph [SocialOomph.com, 2008-2011]. SocialOomph also uses some of the measures reported above, and lets users specify filters on the direct messages that they see.

One particularly interesting validation service, TrueTwit Basic [TrueTwit LLC, 2010], is challenge-based. When a user who has signed up to TrueTwit Basic gains a new follower, a direct message is automatically sent through the user’s account to the follower, asking the follower to solve a CAPTCHA [von Ahn et al., 2003]. If the follower successfully solves the challenge, or has also signed up for the TrueTwit service, TrueTwit Basic sends the user an email indicating that the follower is validated. (TrueTwit Basic is free. The programmers of TrueTwit Basic also offer the non-challenge-based service TrueTwit Premium for $20 a year: the details of TrueTwit Premium’s method have not been published.) The great advantage of TrueTwit Basic is that it does not use features of the tweets or direct messages sent by accounts to identify potential spammers, and is thus completely robust to any changes in spamming software changing the patterns of tweets or direct messages. This service does, indeed, reduce the annoyance caused by spammers, but it does have some drawbacks.

A drawback of any challenge-based validation is that it puts a burden - if only a small one - on benign followers. A more fundamental issue with TrueTwit Basic’s approach, in the context of
pervasive computing devices, is that it aims to detect which followers are automated and which human. There is an assumption that human users do generally not wish to grant a friend request from an automated Twitter account. But this is not necessarily the case, and may become increasingly the case as useful pervasive computing services move to Twitter. (The same issue applies to Grier et al.’s ingenious idea of testing for regularities in the distribution of the hour and minute part of tweet timestamps [Grier et al., 2010]. Grier et al. checked this test on dozens of Twitter accounts and found that although it could not always identify compromised accounts used to send some spam, it was highly accurate in identifying career spammers. However this test is unlikely to be able to distinguish between career spamming accounts and benign automated accounts.)

This issue might be addressed by a “don’t speak until you’re spoken to” convention for automated accounts, stipulating that a well-behaved automated account would not send a friend request to any user unless the user had followed the automated account first. However, this would mean that two automated accounts could not follow each other without direct human intervention, and would rule out some possible useful applications of automated Twitter use - for example, proactive support of users by pervasive computing devices in emergency situations.

SocialToo’s service [Stay, 2010] scans direct messages sent to SocialToo customers, filters out spammy ones, and shows the clean direct messages in the customers’ SocialToo inboxes. SocialToo users can mark a direct message as spammy, and this information is used to update the system-wide filter. This is a useful approach, but it might be circumventable by spammers who automatically send a slightly-different “personalized” message to each recipient. SocialToo also enables their customers to opt out of receiving automated direct messages from all services with an opt-out option. Unfortunately, not all Twitter spammers offer such an option.

Both the TrueTwit Basic and the SocialToo services have a centralized design (as does TrueTwit Premium) resulting in a potential single point of failure. Indeed, both of these services have occasionally been suspended for maintenance or because of server problems [Latko, 2010; @SocialToo, 2010].

Another problem common to TrueTwit Basic and SocialToo, and to other services that validate or filter messages from followers (i.e. accounts whose friend requests have already been accepted) rather than validating accounts that have made a friend request before the decision whether or not to accept the request has been made, is the potential exposure of private messages. Although a user may block a follower as a result of information received from the service, there will be a time interval during which the follower will be able to read the user’s private tweets.

1.3 Suggestions for an Improved Validation Service

The best defence against Twitter spam would probably use a combination of features from several of these services. Indicators based on tweets (along the lines of TwitSweeper, StopTweet, TwitChuck and Clean Tweets) could be used to detect spammers who include trending topics or who send spam messages in their tweets. A shared blacklist and a measure of mutual social contacts (ideas from TwitBlock, SocialOomph and TwittFilter) would harness information from other users. A measure of mutual social contacts uses richer information from the social graph than simply the numbers of
followed and following accounts; two other such measures that might be useful in combination with other tests are the TunkRank [Findable, 2010], and the passivity score which [Romero et al, 2011] suggests as an indicator for bots and spammers. Since timing regularities appear to be an accurate indicator of career spamming accounts [Grier et al., 2010], it makes sense to check for these, but to allow this indicator to be overruled by whitelisting or independent certification of Twitter apps, if the user wishes not to block certain classes of useful automated Twitter accounts. Finally, an analysis of direct messages based on SocialToo, using Bayesian analysis to circumvent personalization of messages, might be effective against accounts sending spam in direct messages.

The problem of centralization might be mitigated to a certain extent by having a number of local validation sites that periodically synchronized with eachother; thus if communication with one of the sites was lost, some validations could still be made locally based on less recent information. To reduce potential exposure of private tweets, the service should be able to provide a spamminess score for each account that makes a friend request, rather than requiring the friend request to be accepted before validation takes place. Requests from accounts with high spamminess scores might be automatically rejected by a user’s Twitter client to minimize annoyance to the user.

4. Conclusion

There are several lessons from this discussion which may be applicable to pervasive computing systems using communication channels other than Twitter. Any pervasive computing network open to the public, which allows automated messages to be sent to human users at low or zero cost, will almost certainly be used to send spam. An important lesson from Twitter is that it is not enough to ensure that messages are only sent to recipients who opt-in. Human users also should be protected from being pestered by large numbers of requests to opt in, and will in general need to be able to automate some of their decisions as to whether or not to accept such requests. It follows that a validation method is necessary to assist with these decisions.

A validation method to assist with these decisions should be reasonably accurate in identifying spammers, should be as robust as possible to changes in spamming software, should not assume that all automated messages will be unwelcome to humans, should be designed not to have a single point of failure, and should protect users’ privacy during the validation period. We need to have good methods of limiting the annoyance that spammers may cause on such networks, or we may find ourselves befriended by hundreds of rice cookers.

References

Tan, G. (2009), The power of Twitter search: a rice cooker just added you, online at garry.posterous.com/the-power-of-twitter-search-a accessed 24.03.2011


Slavin, K. (2009), Things that twitter, presentation at 140 Character Conference, New York, 16 June 2009, online at www.blip.tv/file/2267845 accessed 24.03.2011

Mattel, Inc. (2010), Puppy Tweets™ website, online at puppytweet.com/ accessed 24.03.2011

SparkFun Electronics (2011), Botanicalls Kit, online at www.sparkfun.com/products/10334 accessed 24.03.2011


King, R. (2009), Brands that tweet: customer service via Twitter, BusinessWeek, 26 Aug 2009, online at images.businessweek.com/ss/08/09/0908_microblog/index.htm accessed 24.03.2011

Kiss, J. (2010), Twitter election predictions are more accurate than YouGov, PDA digital content blog, Guardian, 13 May 2010, online at www.guardian.co.uk/media/pda/2010/may/13/twitter-tweetminster-election accessed 24.03.2011

Asur, S. and Huberman, B. (2010), Predicting the future with social media, arXiv.org e-print archive: 1003.5699, online at arxiv.org/abs/1003.5699 accessed 24.03.2011


ogutti (2009), Humanoid robot controlled by Twitter, YouTube video uploaded 3 Dec 2009, online at http://www.youtube.com/watch?v=0cJPkIWR31w accessed 25.03.2011
Mace, G. (2010), Live Twitter table using new Bluetooth shield, online at http://macetech.com/blog/node/93 accessed 25.03.2011

Fahner, C. (2009), #garden, Vimeo video uploaded 12 Oct 2009, online at vimeo.com/7036697 accessed 25.03.2011


Twitter (2011), Automation rules and best practices, online at support.twitter.com/entries/76915 accessed 24.03.2011

Emerge2 Digital Inc. (2009-2010), TwitSweeper™ web page, online at www.twitsweeper.com accessed 24.03.2011

Joi Company (2009-2010), StopTweet web page, online at stoptweet.com accessed 24.03.2011

TwitChuck (2011), TwitChuck – still talking about Twitter spam, online at www.twitchuck.com/blog/ accessed 24.03.2011

BLVD Status (2009), Clean Tweets: Get Rid of Twitter Trend Spam, online at www.seoq.com/blvdstatus/clean-tweets.html accessed 24.03.2011

Boshmaf, Y., Muslukhov, I., Beznosov, K., and Ripeanu, M. (2011), On the Penetration of Online Social Networks: Exploiting the Social Structure, online at lersse-dl.ece.ubc.ca/record/258 accessed 05.05.2011

Tweepi (2011), Tweepi v2.0 web page, online at tweepi.com accessed 24.03.2011

Whitlock, T. (2009), TwitBlock spam ratings explained, Web 2.1 blog, 3 August 2009, online at web.2point1.com/2009/08/03/twitblock-spam-ratings-explained/ accessed 24.03.2011

SocialOoomph.com (2008-2001), SocialOomph site, online at www.socialoomph.com accessed 24.03.2011

TrueTwit, LLC (2010), TrueTwit website, online at www.truetwit.com/truetwit/welcome/index accessed 24.03.2011


Grier, C., Thomas, K., Paxson, V. and Zhang, M. (2010), @spam: the underground on 140 characters or less, Proc. CCS 2010, ACM, October 2010, doi: 10.1145/1866307.1866311


Latko, C. (2010), TrueTwit is the devil, Dreaming of Beetles blog, 18 Jan 2010, online at http://www.latko.org/2010/01/18/truetwit-is-the-devil/ accessed 24.03.2011

@SocialToo (2010), SocialToo: in case you were wondering…,Twitter post, 10 Oct 2010, online at http://twitter.com/socialtoo/status/26917377375 accessed 26.04.2011

Findable, LLC (2010), TunkRank website, online at www.tunkrank.com accessed 26.04.2011