Spam and spam accounts in Web 2.0 applications

Manuel Tremmel

***Abstract*—Web 2.0 applications such as social networks rise and fall with the quality of content of their platform. The main contributor of the content is normally the user. Keeping web portals free of spam is essential for the operator of a web application in order not to lose users and visitors.**

To reduce spam in web applications, several approaches are used to keep automated scripts off the website. Spam detection algorithms can be used to identify spam by the content, suspicious links, by typical spam bot behaviour or other features. Also, the user can be asked to respond to a challenge (known as "Completely Automated Public Turing test to tell Computers and Humans Apart" or in short CAPTCHA) such as identifying characters in an image or verify his/her identity via SMS. Especially character identification CAPTCHAs are widely used, but do not provide sufficient security from spammers and inconveniences users. Novel CAPTCHAs have been suggested that may become the successor of traditional character based CAPTCHAs. This paper will discuss existing and novel anti-spam measures in the context of the underground economy and characteristics of the specific web application.

I. Introduction

Web 2.0 is different from classical static websites. Let us consider a social network, a social bookmarking system, a guest book, a dating website or similar platforms: The user does not have the passive role of a reader, but is able to discuss, collaborate, share his opinion to blog posts and share news, photos or videos in social networks [1]. The new role of the user ranges between being a member of the discussion about posts up to the role of the main content contributor in a collaborative wiki or social network. In social networks, the provider of the web application is merely providing the infrastructure for people to communicate. Being open for human interaction, however, also opens the doors for automated spamming.

Spamming can be defined as "the act of spreading unsolicited and unrelated content" [2]. The term content should be interpreted as any form of visible user interaction to include votes without text (e.g. Facebook likes), which are also a form of spam. Due to the wide range of possibilities for interaction on web 2.0 platforms, spam in web 2.0 applications (spam 2.0) is significantly different from traditional email spam [3]. The limits between spam and ham (i.e. non-spam) are fluent and in some cases subjective and hard to detect (e.g. when used to manipulate opinions). Moreover, spammers may contribute unique and interesting content regularly to deceit anti-spam mechanisms [3].

Countermeasures against spam can be classified into [4]:

- prevention based techniques, which place entry barriers, e.g. by using CAPTCHAs or by limiting user actions,
- demotion based actions to rank potential spam down in sorted lists within the platform, without eliminating it,
- detection based mechanisms may use metrics to remove identified spam or reduce the visibility of potential spam.

Spamming has negative effects for the operator of a web 2.0 application. Here are some examples:

- Loss of users and popularity due to spam content [2],
- deteriorated ranking of legitimate websites in search engines [2], e.g. due to comment spam,
- blacklisting of the website [1],
- unproductive use of server and network resources such as bandwidth and disk space [2].

Thus, providers of these platforms have a legitimate interest to reduce spamming activities.

The current importance of spam 2.0 appears to get ahead of email spam. However, there is still need for comprehensive research about the occurrence of spam 2.0 [2].

Spamming is often done by automated scripts (spam bots), which use the environment of web 2.0 applications to pretend to be a human user in order to spread spam content [1].

In most cases, spamming can be done only when signed up. To make account generation harder for bots, CAPTCHA challenges are often required to create user accounts. However, machine learning enables bots to circumvent CAPTCHAs. For example, Google applied its character recognition against their own CAPTCHAs [5] and outperformed even human users [6]. Also, humans solve CAPTCHA challenges easily and effectively, especially as knowledge of the context where the CAPTCHA is placed is traditionally not required [6]. At the same time, the overhead of the CAPTCHA is inconvenient, distracts the user from doing his intended tasks and reduces accessibility for people with disabilities.

As a spam account is a requirement for spamming web applications, spam account generation is an important constituent of the underground economy. A good understanding of the underground economy and its established structures is essential to develop methods to increase the effort for spammers and thus to reduce spam.

Having introduced spam 2.0 in this section, this paper will explain the motives for spamming and what makes spam 2.0 more effective than other forms of spam in section II. Then we elaborate on the underground services for spam account creation in section III and different types of spam (e.g. social spam, blog spam) in section IV. With this foundation, this paper will then discuss a selection of existing spam counteractions such as spam detection and CAPTCHAs (section VI), covering both traditional and novel approaches. Finally, I will present some open research challenges in section VII.

II. Motives of Spammers

There are various motives for spamming web 2.0 applications. One prominent motive is for search engine optimization (SEO) purposes [7], due to the fact that well-positioned websites with a natural good ranking in search results are of high

economic value. Alternatively, it would be very costly to pay for advertisement to be listed in a comparable position. Also, web spam can be done to sell products or distribute malware. This is a phenomenon that is similar to mail spam or scam where the sheer size of the audience increases the likelihood that somebody falls for the trick. Also, the importance of social networks has lead to opinion spam to manipulate opinions [8]. This is not only interesting for companies to improve the perception of its brand by people, but also helps political or other interest groups to wield influence. A full elaboration of the motives for web spam goes far beyond the scope of this paper, but this brief discussion can help to understand what kind of spammers web applications should be able to cope with.

The advantage of web 2.0 spam is that with a single spam submission a lot of users can be reached, in contrast to email spam. On top of that users cannot easily use spam filters on web pages or delete spam, as they can do with email spam. In most cases, only the provider or administrators of the websites have the privileges to remove spam content. Also, the effort of finding a web application that is suitable for spamming is minimal, as websites are public, whilst email addresses are not so easy to find. Therefore, spam 2.0 is more effective than traditional email spam [2].

III. The role of web spam and spam accounts in the underground economy

To understand the driving factors of spamming, the underground economy structures will be discussed in the following. Web spam and spam accounts are an important constituent of the underground economy. Web spam accounts are required to be able to distribute spam through social networks or other log-in based web 2.0 applications such as social bookmarking services. Therefore, web spam accounts are the base of other frauds and are required as a mass product for the underground market. Revenue is generated through the activities triggered by spam 2.0 such as selling products. In the following, I am going to present and discuss research done about Twitter accounts as a case study [9].

Thomas et al. [9] have conducted a study of fake twitter.com accounts. Twitter accounts are sold relatively cheaply not only because they just form a base of other frauds, but also because fake account generation has only small barriers for spammers. Due to the fact that only email verification as well as a CAPTCHA is used, it is relatively easy to automate the process of creating spam accounts. At the time of their study, no SMS verification was used, the overall cost of creating spam accounts is relatively low compared to Google accounts which require phone or SMS verification in order to be able to fully use the account. Email accounts are sold relatively cheaply and just increase the cost of the process to create fake twitter accounts negligibly. Again, the prices of email accounts vary depending on the kind of human verification mechanism that is employed. The automated CAPTCHA solvers have the advantage of being able to repeatedly guess the CAPTCHA and using different servers from different countries, these actions cannot be prevented or detected by Twitter easily.

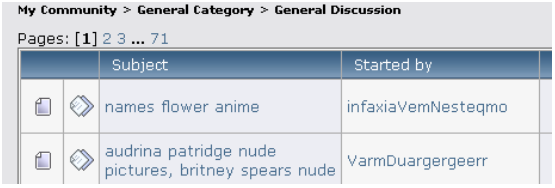

Figure 1. Screenshot of online forum spam (source: [7]).

Therefore, even though the failure rate of CAPTCHA solvers is significantly higher than human solvers, with a wide range of proxy hosts located world-wide, a trial-and-error approach will eventually still create millions of accounts. They suggest to increase the at-registrations defence as this will increase the price for bulk accounts and therefore criminal activities will stop using Twitter and start looking alternative ways to do fraud. A huge take-down activity conducted by the researches actually affected the market for a short time, leading to unavailability of fake accounts for a short time.

At the moment, different email accounts are treated equally by Twitter. I suggest to increase the difficulty of challenges for Twitter accounts which use email accounts for email verification that are cheap in the underground. As the total price of a Twitter account depends on the fake email account price plus circumventing Twitter's CAPTCHA, one might employ additional CAPTCHA verification for hotmail.com email accounts, as little spam prevention mechanisms are used for registering hotmail.com accounts. Then, the weakest link in a chain effect cannot be used to lower fake account registration costs in the underground economy.

To conclude, fake account market prices can be influenced by using adequate human verification mechanisms. In combination with regular take-down activities, a platform operator can create barriers for spammers that might result in a considerable reduction of spam. This improves content quality and user satisfaction. Also, Motoyama et al. [10] actually suggest evaluating CAPTCHAs economically: when the expected revenue is not high enough to justify the spendings to circumvent anti-spam mechanisms, spamming is likely to become less important.

IV. Types of Web 2.0 application spam

There are different kinds of web 2.0 application with different possibilities for user interaction. Due to the different nature of each type of web 2.0 application, there are different methods which spammers can use to attract attention and/or place spam content. The following sections will cover spam in online forums, blogs, wikis, social networks, opinion spam or comment spam. In order to limit the number of web 2.0 spam types, I am going interpret social spam more widely than done by Hayati and Potdar [7], including social bookmarking sites and dating platforms.

A. Forum spam

Just as humans, also bots can sign up in online discussion boards and create discussions or reply to existing discussions. When the forum is public, the links can improve the ranking

of the website which the spammer wants to improve. On discussion boards which are invisible for search engines as they require to be logged in, spamming can be also profitable. Through the use of false keywords, visitors of the forum may be tricked into opening the discussion and subsequently into visiting the website. There they might perform actions to generate revenue for the spammer, such as generating sales or collecting malware from that website [7].

Apart from posts and links in posts and comments, uploading attachments to posts or the profile of the forum user is another possibility to spread spam [7].

B. Blog spam

Spam blogs which are also called splogs attract users often through an illegally acquired good ranking on search engines or by links from legitimate web 2.0 applications. The creation of these blogs involves little effort, as web 2.0 tools and free hosting services exist [7]. Additionally, to this phenomenon mentioned in [7], existing websites can be compromised and filled with spam content. That is especially useful for SEO. Search engine algorithms may be trained to identify computer-generated text that does not make sense for a human reader. Therefore, websites which exist of spam content exclusively may be identified by algorithms and through devotion mechanisms, they are ranked down in search engines and social networks. In this situation, spammers can use human made websites and hijack them, e.g. due to poor configuration or missing security updates. Once spammers achieved that, they might also compromise the webhosting account itself. As blogging software makes it easy for everyone to create blogs, also people with little IT skills can have blogs and they might not be aware of the dangers of running a website and how to mitigate those risks, for example, by installing security updates for the blogging software. The advantage of using other people's web 2.0 applications is that the human generated unique content can be used to improve the ranking in search engines and adding spam content to a website makes spam even more effective. Exact information about how search engines treat spam on websites are hard to get, as this information is normally confidential [11].

a) Webhosting companies and compromised web applications: In the context of blog spam, it is interesting to discuss a variant of spamming web applications, especially blogs. Blogs, forums and wikis can be privately hosted. Whereas non-expert blog owners may not have the experience to evaluate risks and look for countermeasures, their webhosters should be able to warn them and help to ensure (or restore) the integrity of the website. In this context, black hat SEO approaches do not back away from compromising web applications e.g. by abusing known weaknesses in popular blogging software such as Wordpress [12]. When access over a webhosting account is gained, also email spam can be sent. When an email address is used for sending spam, the server gets blacklisted, which consequently affects email communication of other clients in a shared hosting environment badly. Therefore, webhosters should have an interest to detect compromised accounts and web 2.0 applications. However, Canali et al. [13] have shown that most webhosters are not even able to detect the most obvious signs of compromised websites and thus can be of little help for layman clients.

C. Wiki spam

Wikis [7] allow the collaboratively editing of wiki articles. Whereas in blogs the roles of editors and visitors leaving comments are normally clearly defined, all users may equally edit wiki pages and thus add reference links or inject invisible HTML code. Therefore, wiki users may actively detect spam and remove it from the wiki.

Wikipedia is a prominent example for a wiki system [14]. Anybody may edit articles (with few exceptions) and those changes are visible to all visitors. Other authors review those changes and may roll back, change or delete the article if the quality standards are not fulfilled. New users need to prove themselves trustworthy by contributing quality content to Wikipedia. Only then they may rise in the Wikipedia hierarchy and thus get more responsibilities within the Wikipedia ecosystem.

D. Comment and opinion spam

Blogs, newspaper articles, review platforms and many more web applications get "alive" by giving visitors the possibility to comment the posts and share their opinion (thus called "opinion spam" [7]). Also, by rating or recommending (e.g. Facebook Likes) users can share their opinion about things of interest. Spammers can comment blog posts or articles and include links to websites where they can sell the product, or they can spread opinions to manipulate readers. Opinion spam, (as mentioned briefly in section II), is done by placing many reviews or comments etc. that are for or against certain products, services or other items [8] with the intention to influence the readers of these user contributions and possibly adapt these opinions themselves. Hayati and Potdar [7] see opinion spam in the context of products and services, however also political interest groups may use spam to manipulate people and their point of view, in my opinion.

Many blogs by default allow trackback. When a blog post refers to another post of a different website, a trackback is used to inform the author of the original post about the new post. This trackback is displayed as a comment and therefore may count as a form of comment spam. The goal of trackback spam is to get quality links to improve search engine ranking and potentially to trick visitors to visit the own website [7].

Blogs may be configured to refuse links in comments and therefore can eliminate spamming for SEO purposes. However, spamming for manipulating opinions cannot be avoided like that. Also, blog owners may decide to allow comments only after prior approval, which gives extra work to the blog owner. It should be noted, however, that based on my own experience, the default configuration of Wordpress, the most commonly used blogging system [12] is to allow user creation, commenting without approval and trackback spam. Layman users might not be aware of these settings and spammers might manipulate sections to place their content.

E. Social spam

Social networks have become an important place for connecting and networking with other people. Also, variants such as social bookmarking sites or dating sites exist. Social spam is well suited to manipulate opinions about entities. Increasing Twitter follower numbers or buying Facebook likes create the impression that the business is interesting and features a network effect which includes continuous growth by attracting "friends of friends". However, social networks normally do not allow paid likes or followers in their terms of use. Fake accounts in social networks can trick people to visit fraudulent websites or distribute malware through this channel [7]. Also, video sharing platforms such as YouTube have developed to a variant of social networks with a rating and commenting functionality and user channels, which exposes these platforms to the same dangers.

From a usability point of view, the barriers for creating social content should be as low as possible to reduce barriers for sharing [3]. In this context, using demotion and detection based approaches to mitigate spamming can be a better option.

Social spam in the context of social bookmarking services is also popular. Social bookmarking platforms are specialized platforms to share links to favourite websites. Evidently, these platforms are also attractive to generate links to the own website for SEO purposes. Through the excessive use of keywords, which are potentially misleading, spammers may attract attention and direct viewers to their page.

Spam on specialized platforms such as dating websites is different from classical social network spam. For example, on a dating website, a fake profile might be used to advertise for a non-existing woman, tricking people to do a paid contract with a partnership agency. In this case, one account is sufficient for this fraud and only if the operator of the platform verifies the profile, it can be ensured that users of the platform do not fall for fraudsters.

V. Black hat SEO

Search result spamming can be done by using the types of spam described in the previous section, both on self-hosted web 2.0 applications and by abusing existing web 2.0 applications. Sometimes, the final goal of spamming web 2.0 applications is to improve the ranking on search engines. This is especially true for smaller and less popular platforms, in my opinion. In the case of abusing web 2.0 applications, search engines use anti-spam mechanism which are the next "barrier" that web spam needs to pass in order to reach the end user.

An Experiment by [11] shows that numerous small modifications to the website can - in total - affect the ranking of the website in search results considerably. This is called "active SEO 2.0" by Boutet, Quoniam and Smith [15]. As mentioned above, a natural position as one of the first results is very valuable, as visitors normally select one of the first search results and often do not continue browsing when they have found what they were looking for. The term white hat or black hat SEO describes search engine optimization mechanisms that are either in compliance with search engine regulations ("white hat") or black hat, when they are not. White hat is a SEO technique which is time-consuming and hardly automatable as search engines demand quality first and foremost. The goal of black hat SEO is to trick the search engine into improving the ranking of a website more than it "deserves". As search engines implement mechanisms to detect these attempts, black hat SEO needs to use methods that are not obvious in order to be successful. As links to a website, when considered relevant, generally improve the ranking of a website, spammers may post comments on wikis or blogs containing a link to the website. Those links may count as votes, improving the ranking - or destroying the ranking when excessive use has been identified by the search engine. The exact procedure how search engines do their indexing is not public and the business secret of each search provider [11].

VI. Anti-spam mechanisms

Different classifications for anti-spam mechanisms have been proposed such as content specific countermeasures and source specific countermeasures by Hayati et al. [2]. I have chosen the classification of Gadhvi et al. [4] for this paper (which has been briefly introduced in section I), as their classification describes the goal of each countermeasure better. The classification differentiates between prevention, detection and demotion based mechanisms.

Prevention based approaches (or **interface or limit-based counteractions** [3]) use CAPTCHAs, which present some form of challenge to the user such as recognizing characters in an image or solving a mathematical problem. CAPTCHAs are a typical interface based anti-spam mechanism. Another prevention based mechanism is the limitation of actions. For example, hacked Facebook accounts may be detected by excessive friendship requests. To avoid further spamming, the user is blocked to send friendship requests for a specific time.

Demotion based mechanisms (also known as rank-based [3]) rank the content also by the likelihood that the content is spam. For each search within a web application, only a subset of the result is shown. Therefore, the algorithm can retrieve results first which are certainly no spam. With this approach, spam may still reside inside a system. However, it will not be visible in prominent positions of the website. Therefore, spam is less likely to bother users. Heymann et al. call this approach identification based countermeasure [3]. This approach works well with popularity sorting, (in contrast to recent first order), as content generated by a diverse set of users (e.g. geographical) can be favoured over potentially auto-generated content [3].

Detection based mechanisms use supervised or unsupervised machine learning, statistical analysis of the content or user behaviour [3] associated with actions such as removing the content or tagging that content as spam. Detected spam can be removed or flagged as potential spam in order to rank it down in sorted listings (demotion). Other authors call this approach **identification based countermeasure** [3].

The three groups of countermeasures are not mutually exclusive, but a combination of these is recommendable. Whilst prevention can be used in an initial phase against both web spam and web spam accounts, detection can be used when

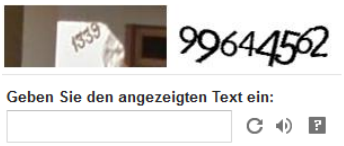

Figure 2. Screenshot of a reCAPTCHA challenge (source: https://accounts.google.com, 01/06/2014) [5]).

spam content already exists. Demotion, on the other hand, tries to reduce the visibility of potential spam by appropriate ranking.

A. CAPTCHAs (prevention based mechanism)

A simple form of a spam prevention mechanism is to limit the number of content submissions or account generations or require a payment if the submission exceeds a certain limit [3]. For example, one can impose a limit of accounts that can be generated from a specific IP over a certain time. Within user accounts, such limitations may be the number of content submissions that are allowed over a defined time period. A combination with source specific countermeasures can be used to further reduce the limits adaptively if machine-like behaviour is detected, such as high frequency of content submission.

1) Character based CAPTCHAs: The "classical" CAPTCHA technology is character recognition based. Implementation is relatively easy as a combination of characters is displayed in the form of an obfuscated image. Like this, the human challenge consists of recognizing the characters, which used to be difficult to do in the past and therefore prevented automated spamming.

An example for such a CAPTCHA system is reCAPTCHA [16], which is used by many web applications (see Fig. 2 and 7). Apart from to the obfuscated characters, reCAPTCHA displays scanned words from books to the right of the characters. The user therefore does not only authenticate himself but also helps to digitalize books in a collaborative approach.

a) Improvements on simple character CAPTCHAs: There have been minor improvements to the traditional obfuscated character CAPTCHA technology such as using colour inversion [17]. These technologies are used to increase the complexity of solving CAPTCHAs for bots, while keeping the challenge for humans at a similar level. Colour inversion, for instance, utilizes the fact that gradients of colours are harder to detect for scripts.

Recent algorithms of Google which are used in large scale to recognize the house numbers in Google Street View images can be applied for solving CAPTCHAs automatically and work with an accuracy of 99.8% in this context. This accuracy is even higher than human accuracy which is less than 93% [6]. This means that with the right tools, spammers can be even more productive than humans in solving CAPTCHAs. In combination with automation, an excessive number of spam or spam accounts can be placed in web applications. Of course, the algorithm of Google is confidential and inaccessible for spammers for now, but webmasters should start implementing alternative ways of securing the websites from spammers. In my opinion, it may be simply a matter of time until character recognition algorithms with a similar accuracy are widely available.

Figure 3. Screenshot of ASIRRA, a CAPTCHA system where users are asked to select all cat photos (source: [10]).

2) Image based CAPTCHAs: The problems with the character based CAPTCHAs described in the previous section have lead to developments to use different approaches to keep robots out of the web application.

ASIRRA is a system by Microsoft which shows images of cats and dogs to the user [18]. An example of ASIRRA is shown in Fig 3. The images come from an image database for pets. This task is very hard to solve for automated scripts, and "much more enjoyable than a text-based CAPTCHA that provides equal security" [18]. Due to an abundant number of animal photos in different settings, it is unlikely that solving the challenge can be automated. Making automatic solving harder for bots and more enjoyable for humans through gamification is certainly highly desirable. However, outsourcing of the challenge is not excluded with this approach. For now, pattern recognition may not be able to automatically solve the challenge, but improvements in technology may also remove the barrier, in my opinion. Also, I see the problem that the data set can be used for training potential automated CAPTCHA solvers. In response to that, one can choose images that ambiguous (i.e. a cat that looks similar to a dog), at the cost of user experience, increasing user frustration. Also, one should consider herd effects: many webmasters follow trends and tend to use the most widely used or subjectively "the best" CAPTCHA technology. However, it is more profitable to circumvent a frequently used CAPTCHA technology as circumventing it gives access to all websites using that CAPTCHA system.

Humans are used and trained to face recognition and excel in that more than in recognizing other objects. In particular, this skill can be used to give humans an advantage over bots. Goswami et al. suggest a CAPTCHA called FaceDCAPTCHA that uses rendered images where parts of the image is cut away: "Face detection algorithms are not able to detect genuine faces due to noise, rotation, background and hidden facial features". Another approach by Goswami et al. is to show faces of the same person from different perspectives. To pass the CAPTCHA, the user has to select all images of the same person. This CAPTCHA has a 94% human success rate [19]. In my opinion, technology to recognize faces will become more and more widely available in the future, as there are

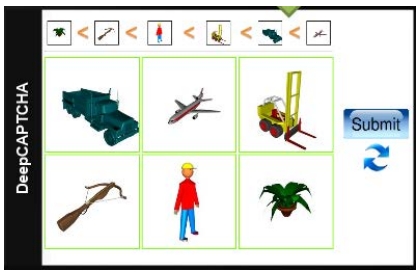

Figure 4. Screenshot of DeepCAPTCHA, where the challenge is to sort the 3D real-world items by size (source: [20]).

numerous real-world applications that require face recognition. In this context, it is worth to mention the face recognition algorithm of Facebook that has the advantage of having a huge training set provided by the users themselves.

Similar to face recognition, algorithms are still not as good as humans in depth recognition of 3D images. Using this fact, Nejati et al. use depth perception of humans to sort 3D images by size which is a task that cannot yet be completed by scripts [20]. This project is called DeepCAPTCHA (see Fig. 4).

3) Audio based CAPTCHAs: Web application accessibility is important to enable people with disabilities to enter and use a website. All major websites have a sincere interest to provide accessibility and some websites are obliged to implement accessibility by law. Audio based CAPTCHAs are already used to enable visually impaired people to get access to pages that are secured by an image CAPTCHA. For example, reCAPTCHA uses an alternative audio based CAPTCHA for accessibility.

4) Video based CAPTCHAs: Videos can communicate complex messages that may be describe through annotating tags. When those tags are assigned understandably, a user challenge might consist of assigning some tags. Kluever and Zanibbi [21] have developed a CAPTCHA using YouTube videos, with the bonus of a more enjoyable user experience with similar security and usability as traditional CAPTCHAs. The CAPTCHA can be used in two different configurations: either a 90% user success rate with 13% attack rate or with a 70% success rate with a 2% attack rate, depending on how hard the challenge for the user is chosen to be.

In my opinion, due to the multimedial nature of videos, there is still a high potential for utilizing videos, animations or simple graphical games as CAPTCHA challenges. Especially the combination of auto-generated videos with click-based and context-dependent challenges as presented in the subsequent section. Context-dependent means that the CAPTCHA can only be solved in the context of the website where the CAPTCHA is used.

5) Click based CAPTCHAs: Some of previous CAPTCHAs work by paraphrasing characters, words or tags into a text field. Click-based CAPTCHAs are different, as an image is presented and the click positions are sent back to the CAPTCHA verification mechanism. This can take several forms. The advantage of click based CAPTCHAs is that the search space for those positions is more complex than the characters of a keyboard.

Figure 5. Screenshot of the graphical password keyboard. The user clicks his password and the click positions are used to check if the password is correct. This alone can be seen as a prevention based mechanism. For a classical click-based CAPTCHA, apart from the "keyboard" also the "password" would be provided (source: [22]).

Figure 6. A screenshot of an implicit CAPTCHA that allows to solve it with a single click. The goal is to click on the word Submit (source: [23]).

Zhu et al. suggest to present obfuscated images of characters to the user. By click at the desired characters, the user can enter his password (see Fig. 5). The CAPTCHA records the clicked character positions, which is advantageous also on touch devices without keyboard [22]. As only the click positions are transmitted to the server, access for spam bots gets very hard. A variation of this approach is to provide a text to a user that is to be entered using the obfuscated character "keyboard". This concept appears to be very versatile.

6) Implicit and dependent CAPTCHAs: The main problem of many CAPTCHAs is that solving them can be outsourced, which is known as relay attack. As suggested by Thomas et al. [9], solving CAPTCHAs by systems like Amazon Mechanical Turk is very cheap and in some cases even more efficient than automated solvers. The technologies presented in section VI may prevent bots (such as click based or audio based CAPTCHAs) if sufficiently complex challenges are presented. However, this may be at the cost of the user who has to try several times and might give up frustrated. The spammer can outsource the task easily and even give the challenge to other users to see some premium content with little or no cost to circumvent the CAPTCHA. The solution to this is to use implicit or dependent CAPTCHAs.

Implicit CAPTCHAs (see Fig. 6) are easy and ideally require one click only. For example, the instructing text and the options to click on may not be in the same image, but in different images. In Fig. 6, the CAPTCHA image is presented instead of the "submit" button. Therefore, the process of answering the CAPTCHA is meaningful and does not to inconvenience the user unnecessarily. They are hard for bots to solve or to farm out, as one image is not sufficient to solve the challenge [23].

Figure 7. Screenshot of a typical reCAPTCHA challenge, which is also hard to solve for humans (source: [5]).

Related to the concept of implicit CAPTCHAs are dependent CAPTCHAs [24], where usability is not as important as for implicit CAPTCHAs. The goal is to include context information in the challenge, such as information on other pages on the same website. A challenge might be to ask the user to replace the second character of a character-based CAPTCHA by the first letter of the second paragraph of an article on the same website. Relay attacks are avoided, as information has to be provided on which URL that CAPTCHA has been found.

In my opinion, implicit and dependent CAPTCHAs have a high potential. However, they can also be circumvented by disclosing context information about the current page with the human solver or by storing patterns of solving a challenge. For example, the spammer may provide a browser extension to the human solver. The browser extension automatically fills out some input fields, so that the human solver only needs to solve the CAPTCHA. However, this is increases the effort of solving services considerably. Also, it is avoided that CAPTCHA solving service providers farming-out CAPTCHAs so that other people willingly or without noticing solve the CAPTCHAs. This is especially true for the case of dependent CAPTCHAs, when navigating to other pages of the website is required to solve the challenge. Another problem of these CAPTCHAs is that solving patterns can be memorized. For example, the position of the mouse pointer needs to be stored and can be sent to the server if the same challenge appears again. Open source web applications, however, are deployed on many websites with little modifications and can only contain a finite set of these challenges.

7) Human Factors and CAPTCHAs: CAPTCHAs are required to ensure integrity of web applications. With the focus being on prevention of automated bots, user experience and human interests are often neglected. However, the actual clienteel are humans, therefore the benefit of customer has to be a central concern. Ideally, the overhead of solving CAPTCHA challenges should not lead to a decline of activity of legitimate users. A survey conducted by Bursztein et al. [6] shows to what extent people struggle when trying to solve CAPTCHAs. Only 93% of eBay CAPTCHAs are solved correctly, and there are harder CAPTCHA challenges than the eBay CAPTCHA such as reCAPTCHA. Interestingly, current audio CAPTCHAs seem to be especially hard for humans and should be improved. This may be due to the fact as alternative CAPTCHA challenges should not be a weaker link in the chain than the "main" CAPTCHA.

Figure 2 shows an example of the reCAPTCHA challenge, which is hard for users to interpret and may be frustrating. On the other hand, Fig. 2 shows a more recent example of the reCAPTCHA challenge, which is easier for humans to recognize.

8) CAPTCHA solvers: CAPTCHA solvers are humans or bots that illegally or for other reasons solve CAPTCHAs to circumvent the CAPTCHA protection. The underground economy provides CAPTCHA solving services in a large scale. The understanding for CAPTCHA solving mechanisms is fundamental for adjusting existing CAPTCHAs and developing new CAPTCHA systems. Therefore, the reaction of CAPTCHA solvers to improved CAPTCHAs can be used for evaluation of anti-spam methods. The goal is to increase the difficulty of CAPTCHA solving for CAPTCHA solvers and easy and possibly worthwhile for legitimate users. Automatic solvers are especially suited for character based CAPTCHAs, as character recognition is already well understood.

A very far-reaching impact has an experiment conducted by Google, which has already been mentioned briefly as part as the evaluation in section VI-A1. Google uses the algorithms for Street View house number detection in order to provide panorama images of streets next to other the traditional map or satellite view. In their experiment, they have used the algorithm trained for house number recognition against their own CAPTCHAs (i.e. reCAPTCHA) as an automatic solver and it turned out that the accuracy was 99.8% [5]. This makes character based CAPTCHAs highly insecure for "sophisticated" spammers.

Additionally, it is shown by [6] that underground services are not as effective as Amazon Mechanical Turk, which is an interesting finding.

There are automatic and human solvers used by spammers to circumvent CAPTCHAs. Automatic solvers are especially suited for character based CAPTCHAs.

To give an impression of the market prices, 1000 CAPTCHAs are solved for the price less than 1.50 US-Dollar [25].

Human solvers (also called CAPTCHA farms) solve CAPTCHAs externally. The client's CAPTCHA is forwarded to the CAPTCHA farm and the result is returned to the client. For example, there exists a Firefox add-on that allows clients with a paid subscription to have their CAPTCHAs solved through such an CAPTCHA solving service. The CAPTCHAs can be detected and solved automatically when configured to do so, which is ideal for spammers. On average, it takes 8.5 seconds to solve a CAPTCHA. Originally, that add-on was intended for dyslexic people [25].

Most users do not notice, if they are asked to solve more CAPTCHAs than required by the web application. CAPTCHA solving services take advantage of that fact by displaying slightly more CAPTCHAs than in a normal web application operation and displays as excess CAPTCHAs that are to be solved for a paying client by a CAPTCHA solving service. This phenomenon is accomplished through malware that manipulates the website on the client side and is called CAPTCHA Smuggling [25].

Humans can also be tricked into solving explicit CAPTCHAs, sometimes without realizing the fact that one's effort is used actually for spamming activities. This may be solved by dependent CAPTCHAs [24]. Dependent CAPTCHAs have been mentioned already in section VI-A6.

B. Wasting spammers' resources (prevention based mechanism)

CAPTCHAs in their various forms are certainly the most important prevention based mechanism. An example of a prevention based counter-measure against spamming that is not a CAPTCHA is presented by Nelson et al. in the form of a toolkit called Spamalot. To reduce the amount of spam to handle for real web forms, a system named Patsy tries to engage spam bots into filling the fake web forms. This unproductive interaction of the spam bot can similarly to CAPTCHAs increase the cost of spamming or spam account creation and thus increase market prices to an extent so that some fraud might not pay out any more [26].

In my opinion, what makes this approach interesting is that it is, ideally, invisible for the average user and thus spam attacks can be resolved without increasing the challenge for the user at the cost of user experience (e.g. in the form of hardly human readable character based CAPTCHAs as discussed in section VI-A7). This should not tempt webmasters to do without CAPTCHA.

An attempt that does not really prevent spamming, but slows it down is Hashcash [2]. The client has to perform a complex calculation that can be cheaply verified on the server side. For example, to create a user account, the browser is given a computational task which requires a considerable amount of CPU power. CPU power becomes the bottleneck of account generation for spamming purposes and as such the spamming or spam account generation process can be slowed down by a factor of millions, which increases the costs for the spammer and thus reduces spam.

C. Spam and spam account detection mechanisms (detection based mechanism)

Apart from the spam prevention mechanisms, spam detection should also be used by the web application to further reduce spam. Spam detection can be done in two ways [2], or ideally in a combination of both:

- **Content specific countermeasures** try to recognize spam by content features. Automatically generated spam content is easy to recognize for humans as a meaningless combination of words of non-matching topics.
- **Source specific countermeasures** detect bots by evaluating the interaction of the human/machine user with the system, such as behavioural patterns.

Detection mechanisms are clearly the favoured approach, as the user is not inconvenienced. Spam detection algorithm can borrow from email spam filtering, however web 2.0 applications have characteristics that when understood and applied properly can improve the performance of detection algorithms [3].

The characteristics of a web 2.0 application are, as pointed out by Heymann [3]:

- Only administrators can remove all sort of spam, whilst users can sometimes report it only. However, this allows for centralized decisions to avoid spam.
- Interactions are constrained through the web 2.0 application, e.g. by adding friends to social networks or add links to social bookmarking systems.
- Actions within the web 2.0 application can be traced back to the originating spamming account, thus making it easy to block spammers.
- Content is potentially displayed multiple times in different views, e.g. tag clouds. This increases the attractiveness of spam 2.0 for spammers, as a single interaction is visible on several pages.

Spam can be detected based on its content or its source (e.g. bot behaviour), or by combining the strength of both approaches, which is the best option in my opinion:

1) Content-based spam detection: Spam content can be generated by partially copying content from other websites or using natural language processing. This characteristic can be used for spam detection. Often, spam generated by natural language processing is a meaningless combination of words of different topics in the form of sentences. As such they do not make sense to a human reader. The spam detection tries to identify these unnatural topic transitions to identify spam. Learning is done by supervised machine learning with human-labelled data sets which contain both spam and ham (i.e. non-spam). Suhara et al. suggest using sentence-level topic information as features [27]. The approach of Suhara et al. was novel: Instead of using topic models, they are working on a sentence-level basis, which allows for a fine-granular detection of transitions. Therefore, ham (i.e. non-spam) texts that legitimately have words from several topics in one sentence or in the whole text are not falsely classified as spam. However, this approach does not solve the problem of partially copying text from other websites.

An approach of Wang, Irani and Pu is to build a framework to detect spam on social networks, which propagates spam findings to other platforms. The idea is to reduce the overhead of duplicate spam detection on different platforms. The spam detection is done in three steps: initially, the profiles, messages and web pages of the social network are converted to a format understandable by the framework. In a pre-filter stage, content is tested against existing simple spam patterns, e.g. by using a blacklist, or by similarity tests. The objective is to remove the most obvious spam in this step. In the subsequent classification step, a supervised machine learning algorithm is used to classify the submitted content as well as the context, e.g. incoming or outgoing links [28].

2) Source specific spam detection: Apart from detecting spam by assessing its content, it is also possible to detect spam by the spambot behaviour as shown by [29]. Behaviour can be measured in several features, such as action time and frequency. Action time is the time required to perform an action such as a login. Action frequency describes the number of actions that are performed, e.g. how many accounts are registered. Typically, there are longer idle periods between page accesses when humans navigate on a website compared to bots. Bots might submit several forms (action frequency) within a few milliseconds (action time). This characteristic can be used by source specific spam detection algorithms. The evaluation based only on behaviour gave a 94.7% accuracy

of spam bot detection when tested against an online forum. There exist numerous other features that might be employed for future source specific spam detection algorithms.

3) Spam link detection: URL shortening services are widely used for Twitter tweets, as tweets are length-constrained. Also, on other social networks, shortened URLs are used. However, shortened URLs are problematic because the target URL is not visible from the short link and therefore, in doubt links are opened, possibly installing malware on the computer even before the user can fully evaluate the URL. The providers of the web application can monitor those short URLs and display warnings in case of doubt, remove the links or identify the spamming account and initiate further actions, ideally near real-time. Such a monitoring system has been suggested for the case of Twitter by [30]. The authors point out that click traffic can be used to classify spam and non-spam URLs.

VII. Open research challenges

Researchers still see a high potential for spam 2.0 detection methods, because the emphasis appears to have been on email spam classification so far. Especially the field of spam bot detection asks for new solutions in order to be not dependent on CAPTCHAs which can be circumvented and reduce usability [1].

Currently, there is not sufficient research about the economic cost of spam 2.0 despite its huge impact on the web [2]. Also, there is little research about opinion spam, which is a form of spam that may be very hard to detect with current anti-spam methods. This spam may be inserted manually and written manually also, e.g. to rate a hotel positively.

Character based CAPTCHAs are popular and widely used by web 2.0 applications. However, automated character recognition is already better than human solvers. The example of Google character recognition algorithm with a precision of 99.8% has been mentioned in section VI-A8 [5], whereas humans would solve similar CAPTCHAs with an accuracy of less than 93% [6]. That means that these CAPTCHAs are not a challenge to tell humans and machines apart any more. Therefore, new CAPTCHAs need to be found to replace existing solutions. Many interesting approaches have been suggested, some of which have been presented in section VI.

When considering new types of anti-spam mechanisms, these aspects should be considered:

- usability of the web 2.0 application should be a key factor,
- community specific characteristics should be considered when designing anti-spam mechanisms [3] (e.g. for wikis one might prefer manual spam selection in combination with CAPTCHAs, whereas social networks might work better with spam detection algorithms and limits only),
- spam detection mechanisms, which are invisible for the user, should be favoured over CAPTCHAs,
- more features should be collected and weighted for spam detection to improve existing spam detection algorithms [29],
- accessibility should be provided with a similar difficulty of CAPTCHA alternatives [6],
- automated cooperation of different web 2.0 applications can help to improve the performance of spam detection through exchange of spam findings [28],
- real-world interaction such as SMS or phone verification should constitute a component of the human verification process,
- smart adaptation techniques of web 2.0 application to reduce visible barriers (e.g. CAPTCHAs) for unsuspicious users and increased barriers or barrier cascades when suspicious behaviour is detected, in my opinion,
- the challenges presented to the users may be personalized, in my opinion. Users who have email accounts that requires a sophisticated verification process may skip some visible barriers or CAPTCHAs. The goal is to keep the overall challenge similar for all users. Another example is to reduce CAPTCHAs for users who chose a strong password as hackers are less likely to guess it using brute force.

When taking the definition of spam 2.0 provided by Hayati, Potdar and Talevski strictly, spam 2.0 is limited to web applications, that are not self hosted [2]. In my opinion, widening the definition to all web 2.0 applications could be reasonable. Web 2.0 applications hosted by spammers can be gentrified by user content, who mistake the website for a legitimate website and thus post to that website. An example could be a social bookmarking platform that mainly hosts links to spam pages, but is also open for users to post their own links. To constrain user generated bookmarks to contain only unique descriptions can improve the overall ranking of the spam website, as search engines normally try to list relevant pages with unique content first. This is also a field that may be explored in future works.

VIII. Conclusion

Spam 2.0 has an importance that is estimated to exceed the importance of email spam. Spamming can be done e.g. for SEO, manipulating opinions and, analogue to email spam, to get people buy or do some other form of interaction for the spammers' benefit. Due to natural propagation of content throughout the web application via different views, one spamming "entry" may become visible in several views, which increases the media penetration.

There has always been a competition between webmasters trying to increase the challenge and "spammers" trying to automatically solve the CAPTCHAs. As the spammers abilities increased, CAPTCHAs had to become more difficult. However, as CAPTCHAs became more difficult, humans started to struggle more and more with solving CAPTCHAs, reducing the user experience.

Prevention mechanisms are essential for imposing a barrier to spammers. However, recent research suggests that relying on traditional CAPTCHAs which are character based will not be sufficient as character recognition algorithms have arrived at a point where automatic solvers are more accurate than human solvers. On top of that, the underground economy is used to the established CAPTCHA technologies and has created an infrastructure to solve CAPTCHA challenges cheaply and

in a large scale, using both bots and human solvers. Also, CAPTCHA solving services can abuse real visitors to solve CAPTCHAs. Thus, CAPTCHAs are required that do not suffer of the farming-out problem. Click-based CAPTCHAs are hard to farm-out, such as an obfuscated image of characters which are to be clicked to enter the password. Implicit CAPTCHAs are designed to be solvable easily by humans, ideally with one click. They are hard for machines as click positions are dependent on the composition of the current website.

An understanding of the motives and backgrounds of spammers helps to decide for the right approach. Apart from spamming for SEO, triggering users to perform desired (e.g. sales) actions, there is also spamming to manipulate the perception of brands or other entities in the community. Due to budget differences between underground CAPTCHA solvers and governmental agencies to manipulate opinions, a wide variety of attacks is to be expected. Concerning commercial spammers, the overall goal is to make spamming expensive and therefore economically inefficient to reduce spam in web 2.0 applications.

Spam detection is an evolving technology with high potential, as user behaviour can be monitored and based on behavioural data, spam bots can be identified. Thus, both spam content detection and spam bot detection is possible.

Each anti-spam mechanism should weigh carefully between the users' expectations and anti-spam. When counteractions hinder the user too much, web 2.0 applications are at the risk of loosing users and thus the web 2.0 applications become irrelevant themselves.

REFERENCES

[1] I. A. Adegbola and R. G. Jimoh, "Article: Spambot detection: A review of techniques and trends," *International Journal of Applied Information Systems*, vol. 6, no. 9, pp. 7–10, March 2014, published by Foundation of Computer Science, New York, USA.

[2] P. Hayati, V. Potdar, A. Talevski, N. Firoozeh, S. Sarenche, and E. Yeganeh, "Definition of spam 2.0: New spamming boom," in *Digital Ecosystems and Technologies (DEST), 2010 4th IEEE International Conference on*, April 2010, pp. 580–584.

[3] P. Heymann, G. Koutrika, and H. Garcia-Molina, "Fighting spam on social web sites: A survey of approaches and future challenges," *IEEE Internet Computing*, vol. 11, no. 6, pp. 36–45, Nov. 2007. [Online]. Available: http://dx.doi.org/10.1109/MIC.2007.125

[4] M. H. Gadhvi and M. M. Shukla, "Comparative study of classification algorithms for web spam detection," *International Journal of Engineering*, vol. 2, no. 12, 2013.

[5] I. J. Goodfellow, Y. Bulatov, J. Ibarz, S. Arnoud, and V. Shet, "Multi-digit number recognition from street view imagery using deep convolutional neural networks," *CoRR*, vol. abs/1312.6082, 2013.

[6] E. Bursztein, S. Bethard, C. Fabry, J. Mitchell, and D. Jurafsky, "How good are humans at solving captchas? a large scale evaluation," in *Security and Privacy (SP), 2010 IEEE Symposium on*, May 2010, pp. 399–413.

[7] P. Hayati and V. Potdar, "Toward spam 2.0: An evaluation of web 2.0 anti-spam methods," in *Industrial Informatics, 2009. INDIN 2009. 7th IEEE International Conference on*, June 2009, pp. 875–880.

[8] B. Mehta, "Lies and propaganda:detecting spam users in collaborative filtering," in *In Proceedings of IUI07*, 2007.

[9] K. Thomas, D. McCoy, C. Grier, A. Kolcz, and V. Paxson, "Trafficking fraudulent accounts: The role of the underground market in twitter spam and abuse," in *Presented as part of the 22nd USENIX Security Symposium (USENIX Security 13)*. Washington, D.C.: USENIX, 2013, pp. 195–210.

[10] M. Motoyama, K. Levchenko, C. Kanich, D. McCoy, G. M. Voelker, and S. Savage, "Re: Captchas: Understanding captcha-solving services in an economic context," in *Proceedings of the 19th USENIX Conference on Security*, ser. USENIX Security'10. Berkeley, CA, USA: USENIX Association, 2010, pp. 28–28.

[11] S. Chavan, A. Chitre, and H. Bhala, "Search engine optimization (seo)," in *International Journal of Engineering Research and Technology*, vol. 2, no. 10 (October-2013). ESRSA Publications, 2013.

[12] Q-Success. (2014, May) Usage of content management systems for websites. [Online]. Available: http://w3techs.com/technologies/overview/content_management/all

[13] D. Canali, D. Balzarotti, and A. Francillon, "The role of web hosting providers in detecting compromised websites," in *Proceedings of the 22Nd International Conference on World Wide Web*, ser. WWW '13. Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee, 2013, pp. 177–188.

[14] Wikipedia Foundation. (2014, May) Wikipedia:about. [Online]. Available: http://en.wikipedia.org/wiki/Wikipedia:About

[15] C.-V. Boutet, L. Quoniam, and W. S. R. Smith, "Towards active seo (search engine optimization) 2.0," *JISTEM - Journal of Information Systems and Technology Management*, vol. 9, pp. 443 – 458, 12 2012.

[16] L. von Ahn, B. Maurer, C. McMillen, D. Abraham, and M. Blum, "recaptcha: Human-based character recognition via web security measures," *Science*, vol. 321, no. 5895, pp. 1465–1468, 2008.

[17] T. Men, Y. Sun, D. Wang, and M. Wang, "A novel dynamic captcha based on inverted colors," in *Instrumentation and Measurement, Sensor Network and Automation (IMSNA), 2013 2nd International Symposium on*, Dec 2013, pp. 796–799.

[18] J. Elson, J. R. Douceur, J. Howell, and J. Saul, "Asirra: a captcha that exploits interest-aligned manual image categorization," in *In Proceedings of ACM CCS 2007*, 2007, pp. 366–374.

[19] G. Goswami, B. M. Powell, M. Vatsa, A. Noore, and R. Singh, "Fr-captcha: Captcha based on recognizing human faces," *PLoS ONE*, vol. 9, no. 4, p. e91708, 04 2014.

[20] H. Nejati, N.-M. Cheung, R. Sosa, and D. C. I. Koh, "Deepcaptcha: An image captcha based on depth perception," in *Proceedings of the 5th ACM Multimedia Systems Conference*, ser. MMSys '14. New York, NY, USA: ACM, 2014, pp. 81–90.

[21] K. A. Kluever and R. Zanibbi, "Balancing usability and security in a video captcha," in *Proceedings of the 5th Symposium on Usable Privacy and Security*, ser. SOUPS '09. New York, NY, USA: ACM, 2009, pp. 14:1–14:11.

[22] B. Zhu, J. Yan, G. Bao, M. Yang, and N. Xu, "Captcha as graphical passwords: A new security primitive based on hard ai problems," *Information Forensics and Security, IEEE Transactions on*, vol. 9, no. 6, pp. 891–904, June 2014.

[23] H. S. Baird and J. L. Bentley, "Implicit captchas," in *Electronic Imaging 2005*. International Society for Optics and Photonics, 2005, pp. 191–196.

[24] R. Halprin, "Dependent captchas: Preventing the relay attack," *Weizmann Institute of Science*, 2009.

[25] M. Serrao, S. Salunke, and A. Mathur, "Cracking captchas for cash: A review of captcha crackers," *International Journal of Engineering*, vol. 2, no. 1, 2013.

[26] P. C. Nelson, K. P. Dallmeyer, L. M. Szybalski, T. P. Palarz, and M. Wieher, "Spamalot: A toolkit for consuming spammers resources," in *In Proc. of CEAS*, 2006.

[27] Y. Suhara, H. Toda, S. Nishioka, and S. Susaki, "Automatically generated spam detection based on sentence-level topic information." in *WWW (Companion Volume)*, L. Carr, A. H. F. Laender, B. F. Lscio, I. King, M. Fontoura, D. Vrandecic, L. Aroyo, J. P. M. de Oliveira, F. Lima, and E. Wilde, Eds. International World Wide Web Conferences Steering Committee / ACM, 2013, pp. 1157–1160.

[28] D. Wang, D. Irani, and C. Pu, "A social-spam detection framework," in *Proceedings of the 8th Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference*. ACM, 2011, pp. 46–54.

[29] P. Hayati, K. Chai, V. Potdar, and A. Talevski, "Behaviour-based web spambot detection by utilising action time and action frequency." in *ICCSA (2)*, ser. Lecture Notes in Computer Science, D. Taniar, O. Gervasi, B. Murgante, E. Pardede, and B. O. Apduhan, Eds., vol. 6017. Springer, 2010, pp. 351–360.

[30] D. Wang, S. Navathe, L. Liu, D. Irani, A. Tamersoy, and C. Pu, "Click traffic analysis of short url spam on twitter," in *Collaborative Computing: Networking, Applications and Worksharing (Collaboratecom), 2013 9th International Conference Conference on*, Oct 2013, pp. 250–259.