Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Theoretical Computer Science 197 (2008) 149–157

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

Avoiding Massive Automated Voting in Internet Polls

Alessandro Basso[1] Michele Miraglia[2]

Computer Science Department
University of Torino
Torino, Italy

Abstract

Internet polls are becoming nowadays more and more important as they are being used on a large scale. Their security aspects are probably the most significant and hardest issue to be solved when we consider web surveys. In this paper we present an innovative solution which considerably increases the security of Internet polls and the reliability of their results by discriminating between human-cast and machine-cast votes. We define the basic idea of the method used to achieve our goal, its security properties and the testing results on a working prototype. In order to better test the performance and the effective robustness of the solution from the security point of view, we also propose a public challenge.

Keywords: Internet polls, web surveys, security, visual CAPTCHAs.

1 Introduction

Online polls can be defined as online surveys of public (or a sample of public) opinion to acquire information about a specific topic. Being a web application, a poll can suffer from a variety of attacks typical of this kind of programs. In particular, the most relevant one has to be considered the explicit alteration of poll results by means of automatic voting programs (bots), which act like human users in order to vote repeatedly.

There are many examples where the results of some polls were dramatically influenced by bots that were able to vote for a big number of times although security measures were taken, like the famous episode involving a *Slashdot* Internet poll about the best American graduate school in computer science [4].

Several factors make this security issue particularly complex to solve: the multitude of browsers used, inconveniences to distribute certificates or secrets like user-

[1] Email: alessandro.basso@di.unito.it

[2] Email: michele.miraglia@di.unito.it

1571-0661 © 2008 Elsevier B.V. Open access under CC BY-NC-ND license.

doi:10.1016/j.entcs.2007.12.024

names and passwords, the fact that users connect from different workstations, usually behind firewalls, proxies and address translation boxes. Moreover, the structure of a typical web application easily allows computer programs to imitate human behavior, permitting them to iteratively cast votes in order to obtain arbitrary modifications of the final results.

In this paper we present an innovative solution to solve the massive automated voting problem which does not require any user registration or authentication and is able to accept votes from computers connected to Internet with the same IP address. Then, we discuss about the advantages of using the security method that we have developed and we show some test results obtained from a working prototype which implements our idea. Finally, we invite Internet users to find weaknesses in the proposed method by undertaking an open challenge.

2 Related work

According to [3], currently there are three different categories of protection techniques which can offer some level of security to a poll system, preventing multiple and automated voting. However, their effectiveness cannot be considered high and, from a security perspective, they can be classified as medium or weak methods.

The simplest way to protect a poll from automated scripts is the *cookie-based* scheme. I works by setting a cookie containing the poll ID on the client, after a user has correctly voted. This cookie is then sent in each HTTP call to the voting script, thus stopping from being accepted subsequent voting requests from the same client. The main issue with this method is the possibility to delete the cookie before voting again, therefore allowing multiple voting.

A very widespread protection method is the *IP locking* scheme, which uses the IP address of voters to discriminate among different voting sessions. That is, the IP address of voters is stored on the server after the first vote and subsequent attempts of voting from the same address are considered invalid. This approach guarantees a high level of security, since IP address spoofing is generally considered impossible on Internet. However it also prevents from voting potentially allowed voters, like users who use the same proxy server or NAT box to connect to Internet, those who share multi-user workstations or are connected through a dynamically assigned IP address. Therefore, the IP locking method cannot be considered a suitable protection scheme for Internet polls, since it excludes a considerably large number of potential voters.

A further approach to poll security is *based on CAPTCHA* [3] and relies on inability of computers in recognizing textual contents embedded inside pictures, characterized by low quality and quite strong degradation. These images are generally easily readable for human beings, but their content is usually illegible even to the best OCR softwares [8].

[3] Completely Automated Public Turing Test to Tell Computers and Humans Apart, also known as *reverse Turing test*. Unlikely the traditional Turing test [13], the problem is not proving that one is a human to another human, rather a computer has to decide whether one is human or not. See [4] for a more detailed explanation and examples.

Many different types of CAPTCHAs have been proposed [1], based on hard-to-solve Artificial Intelligence problems: some exploit computer programs inability to read extremely distorted and corrupted text; some work by asking the user to solve a visual pattern recognition problem, while others are based on machines difficulties of understanding spoken language or concepts expressed by images.

Currently, there are two ways of using a CAPTCHA to protect the access to a poll: one requires the user to pass the test before obtaining the poll's main page [2]. However, this approach is not considered suitable for general applications, since it does not allow the poll page to be directly inserted inside a portal or a web site.

The second application of CAPTCHAs to the poll context is known as *order-based method* [3]. The main idea behind this protection scheme is to insert each of the poll voting options into a runtime-generated and heavily degraded image, in different rows. The order-based method posses a higher level of security compared to the cookie-based method and it is less restrictive than the IP locking scheme, since every user is allowed to vote. However, security weaknesses has been found in textual CAPTCHAs [5,6,11,12] and it is possible to bypass the protection scheme only by focusing on the different length of choice strings, rather than extracting the full text from the image.

All the above considerations clearly show that currently there is no protection scheme which can be considered both highly secure and fair enough to let all the potential voters express their personal opinions about a specific topic.

3 Avoiding automatic voting

The main problem in CAPTCHA-based protection schemes has been shown to be the type of test used, which cannot be considered anymore completely effective to tell humans and computers apart. Due to this reason, we devised a new technique for preventing alteration of Internet polls results which is based on the human ability to recognize a generic object displayed by a picture never seen before. Using a test based on this ability, we can discriminate between humans and computers, thus preventing automated programs to arbitrarily modify the distribution of poll results.

We focused our attention on an evolution of the well-known CAPTCHA named *ESP-Pix* [4]. This test requires the user to recognize a specific object from a set of pictures, then select the correct option representing that concept from a drop-down menu. By introducing this idea in the order-based method, we devised a more secure protection scheme named *picture-based method*, which does not suffer from the weaknesses affecting the other solutions.

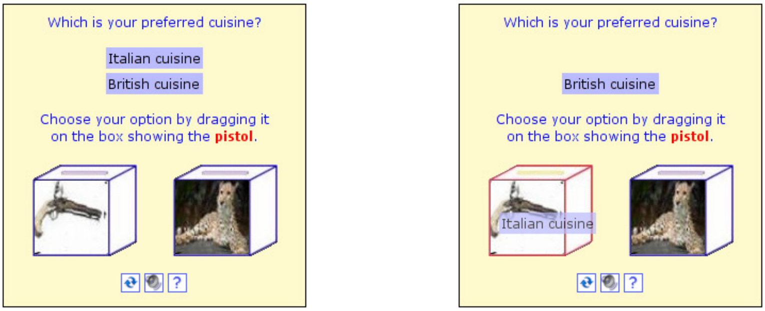
The new solution shares with the order-based method some features, above all, it *embeds* the security mechanism inside the poll, thus allowing a simple and direct insertion of the poll within portals and web sites. However, in order to properly integrate the evolved CAPTCHA into the poll page, we should consider that an Internet poll is a web application with some specific characteristics:

- it should be easily understandable by voters;

- the voting procedure should be quick and clear to learn and should not take more than few second to be brought to a conclusion;
- it should occupy a little portion of the screen;
- it should "encourage" a web site user to spend some time reading the poll and voting for it.

Considering all these constraints, we opted for a *drag-and-drop* approach: voters have to drag the desired choice, expressed in form of text, and drop it on the box with the picture suggested from the poll. This requires the user to recognize from two different pictures, chosen properly from a database, both representing a specific concept. When the user drops a choice on the indicated image, a vote related to that choice is submitted and correctly counted. Otherwise, a new test (with different images) is presented to the voter. If a voter fails the test for more than two times, he or she is prevented from accessing the poll page again. This is possible by locking his or her IP address for a variable amount of time. The locking time increases with the number of subsequent failed attempts of voting, until a maximum value is reached.

In figure 1 is shown an example of the explained method. Note that the picture-based scheme contains a CAPTCHA which is easier to solve if compared to the ESP-Pix. Indeed, the correct category is not listed within a drop-down menu, but it is clearly indicated by the test itself, thus improving the usability of the CAPTCHA.



Which is your preferred cuisine?

Italian cuisine

British cuisine

Choose your option by dragging it on the box showing the **pistol**.

Which is your preferred cuisine?

British cuisine

Choose your option by dragging it on the box showing the **pistol**.

(a) poll page

(b) voting for Italian cuisine

Fig. 1. Examples of the picture-based method

Being a visual-CAPTCHA, it inherits an issue typical of these tests: it poses problems for blind and visually impaired people and, in general, for those users who have cognitive and learning disability [10]. Due to this reason, the picture-based method has to be integrated by means of an audio CAPTCHA [9], which is, on the contrary, a suitable alternative in these specific circumstances. However, an approach to security of Internet polls exclusively based on such a test is not recommended, since the difficulty of undertaking the test would be too high and the level of accessibility of the poll considerably low.

4 Properties of the proposed method

As correctly stated in [3], an Internet poll is *fair* and *secure* against *massive falsifications* if:

(i) Internet users are always allowed to participate in the poll, without any restriction due to protection schemes based on IP address filtering or other discriminating parameter which is not unique for each user;

(ii) programming a bot for massive falsification is a difficult and time-consuming task to achieve.

Therefore, we must obviously rule out any protection technique based on the IP locking scheme and the cookie-based method. Regarding the picture-based method, we can claim that it satisfies both the above requisites, due to its security properties, described below.

Fairness

The proposed scheme can be considered *fair* in the sense that it does not prevent any human user from participating to the poll, even if multiple voters share the same IP address. Indeed, every voting session is separate from the others so that two or more users can vote to the same poll, at the same time, even using the same workstation. The only exception is the IP address locking which happens when a voter repeatedly fails the test. In that specific case, if different voters share the same IP address of the possible attacker, they are prevented from accessing the poll. Moreover, the suggested scheme can be considered fair even towards visually impaired people, since it allows the voters to switch to an audio security challenge.

Classification resistance

We claim that the proposed protection scheme is *classification resistant* since it is able to prevent automatic pictures recognition by means of classification of all images stored in our database. We achieve this goal by using the following techniques:

- Pictures are extracted from a database that contains a very large number of different images belonging to a considerably high number of categories. Thus, the probability to get the same picture twice is low, making the automatic classification of the database content much harder to achieve. Regarding the creation of the database, possible approaches are shown in [7].
- The system applies a number of transformations on every selected image, such as *resizing* it by a random percentage of the original size, *rotating* it by a randomly chosen angle or *flipping* it on its vertical axis. More complex operations like random *distortion* can also be applied, in order to make harder performing image comparisons by means of boundary detection. Furthermore, the system dynamically modifies the *shade* of a number of pixels with a randomly chosen percentage of its original color, in order to increase the difficulty of performing an attack based on colors recognition.

- *Multiple categories* are associated to a single picture, rather than only one. This way, if we ask a user to recognize the picture representing a *feline*, we can provide an image with either a *tiger* or a *cat*. The algorithm for selecting categories and pictures is based on the simple idea that, given an image a belonging to the category c, a second image b will be chosen from a category that has no pictures in common with c. With this scheme, there is no chance that the user might be asked to recognize an object that is displayed in both two pictures.
- The system uses the *pool of category* technique for improving classification resistance of the pictures database. With this term, we refer to a method for limiting the number of categories (and related images) that a single IP address can access. The idea is to create a subset of all categories, called *pool*, and associate it to a specific IP address. Each request for a poll page from that address would result in returning two images randomly chosen from two of the pool's categories. This way, a single IP address is bound to a fixed subset of categories until a valid vote is cast. Then, a new pool is created in function of the voter IP address and the number of valid votes (from that IP address). This technique thwarts both the manual and automatic classification process, since, each time a correct vote is casted, the classification task must be repeated.

Robustness

In order to be considered *robust*, the picture-based method must be able to stop automatic voting programs from modifying the result of a poll through proper attacks. From our tests and the Internet open challenge proposed in section 5, we have been able to detect two possible types of attacks. They are described below, along with the countermeasures used to prevent them from being successful:

- *blind-voting* attack, which executes repeated attempts to vote by subsequently dragging the chosen option on the first of the two pictures. Repeating this voting procedure for n times would allow the attacker to successfully cast an average of $n/2$ votes and fail the other half. All these wrong votes would lead to an increasing temporary lock of the attacker's IP address, making the blind-voting attack too slow and not effective in massively altering the results of a poll.
- *similarity-based* attack, which tries to guess the correct binding between the indicated category and one of the displayed pictures by exploiting visual image similarities. The attack is a two-step procedure: in the first one, the bot builds up a repertoire of reference images for each category, retrieving several pictures from the poll page, which are then categorized by a human being. The second step is the real attack and works by repeatedly comparing the two candidate images against previous instances of pictures that fit the category indicated in the poll. If a match is found, that image is used as the guess and a vote is attempted. In case the vote is unsuccessful, the other image is added to the collection associated with the category and the process restarts.

  The proposed protection method can be considered robust also against attacks based on images similarities due to the usage of the "pool of categories" technique, which forces a new classification phase each time a vote is correctly casted.

As a last consideration, note that the proposed method still allows multiple votes being manually cast from a human voter. Although this behavior might be considered inadequate under some circumstances, it does not affects the security of the protection scheme. Considering all the above premises, we can thus state that the proposed security method is highly resistant against automatic attacks.

5 Tests and results

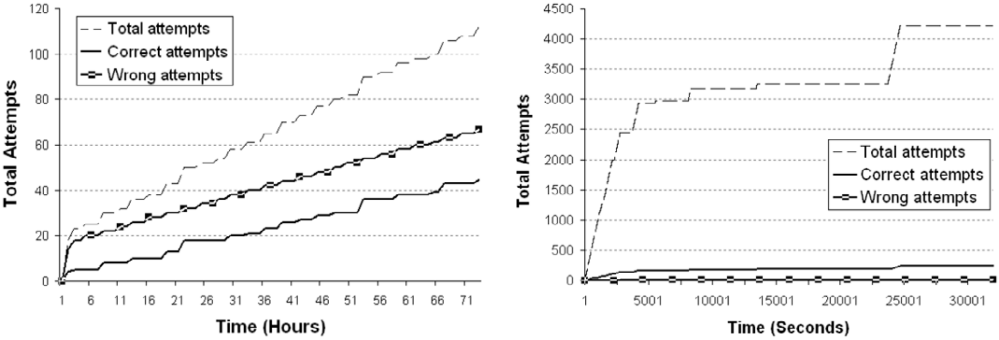
In order to verify the correctness of our proposed solution, we developed a prototype implementing the main idea and its security properties. We then performed a series of tests by simulating the usual behavior of a voting bot implementing both the blind-voting attack and the similarity-based attack. We also set up a wider test through an Internet open challenge[4] .

The test session on the blind-voting attack has been conducted by implementing a simple bot which is able to vote automatically by repeatedly selecting the same voting option and dragging it always to the first picture. The attack lasted for 72 hours and the total number of correct and wrong votes was recorded. By plotting those data on a graphic, we can analyze the distribution of votes when the picture-based scheme is used to protect the poll. As we can observe from figure 2(a), a total number of 113 votes was attempted (continuous line) and only 45 can be counted as effective increment of the poll results (square-marked line). The other 68 attempts are wrong (cross-hatched line) and led to a temporary locks of the bot's IP address. Since it has not being able to massively modify the final result of a poll in a relatively long period of time, the blind-voting attack cannot be considered successful.

The second series of tests was performed on the similarity-based attack. However, in this simulation, we decided to disable the "pool of categories" protection, in order to simplify the attack. Then, we ran a bot to automatically download 100 images from the poll and we manually classified them. With this small subset of pictures, we started the bot implementing the similarity-based attack, which tried to vote for several hours. The final results of the simulation are shown in figure 2(b). As we can observe from the graphic, a total number of 253 votes (continuous line) has been attempted, while the total number of accesses to the poll is 4221 (cross-hatched line). This fact clearly shows that the bot tried to vote only the 6% of times, that is only when the images were rather similar. However, for 240 times the vote was counted positively, with only 13 errors (square-marked line) which temporarily locked the bot's IP address.

In this second simulation, the bot was more effective compared to the blind-voting attack. However, in a real situation, with the "pool of categories" protection enabled, a human intervention for manually categorize an average of 100 pictures would be required for 240 times, one for each correctly cast vote. It is therefore reasonable to state that it is undoubtedly more convenient to vote manually for 240

[4] `http://secg.di.unito.it/ipollchallenge/`



(a) blind-voting attack

(b) similarity-based attack

Fig. 2. Tests

times than attempting an automatic, similarity-based attack.

6 Conclusions

In this paper we presented the picture-based scheme, a new solution for preventing automatic and massive alteration of Internet polls results, developed to address security issues arose with an existing protection method. The novelty of this solution is the direct embedding of the CAPTCHA-based protection inside the poll page, whilst its security aspects rely on the inability of modern computer to correctly recognize concepts expressed by pictures and sounds contained into audio streams. It can be defined as (1) fair, since it does not discriminate users, (2) robust, due to its strong CAPTCHA properties and (3) resistant to classification of all pictures constituting its database. In addition, the proposed scheme allows a simple and immediate integration of the poll inside Internet portals and web pages, which is a feature particularly important for granting high visibility and accessibility to online surveys.

References

[1] Ahn, L. V., M. Blum, N. J. Hopper and J. Langford, *Captcha: Using hard ai problems for security*, in: *EUROCRYPT*, Lecture Notes in Computer Science **2656** (2003), pp. 294–311.

[2] Ahn, L. V., M. Blum and J. Langford, *Telling humans and computers apart automatically.*, Commun. ACM **47** (2004), pp. 56–60.

[3] Basso, A., F. Bergadano, I. Coradazzi and P. D. Checco, *Lightweight security for internet polls.*, in: *EGCDMAS* (2004), pp. 46–55.

[4] Blum, M., L. V. Ahn and J. Langford, *The captcha project: Completely automatic public turing test to tell computers and humans apart* (2000), http://www.captcha.net.

[5] Chellapilla, K., P. Simard and M. Czerwinski, *Computers beat humans at single character recognition in reading-based human interaction proofs (hips)*, in: *In Proceedings of the Second Conference on Email and Anti-Spam (CEAS)*, Palo Alto, CA, 2005.

[6] Chellapilla, K. and P. Y. Simard, *Using machine learning to break visual human interaction proofs (hips)*, in: L. K. Saul, Y. Weiss and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, MIT Press, Cambridge, MA, 2005 pp. 265–272.

[7] Chew, M. and J. D. Tygar, *Image recognition CAPTCHAs*, Technical Report UCB/CSD-04-1333, EECS Department, University of California, Berkeley (2004).
URL http://www.eecs.berkeley.edu/Pubs/TechRpts/2004/5256.html

[8] Coates, A., H. Baird and R. Fateman, *Pessimal print: A reverse turing test*, in: *Proc. of the Sixth Intl. Conf. on Document Analysis and Recognition*, Seattle, WA, 2001, pp. 1154–1158.

[9] Kochanski, G., D. Lopresti and C. Shih, *A reverse turing test using speech*, in: *In Proceedings of the International Conferences on Spoken Language Processing*, Denver, Colorado, 2002, pp. 1357–1360.

[10] May, M., *Inaccessibility of captcha: Alternatives to visual turing tests on the web*, W3C Working Group Note (2005), http://www.w3.org/TR/turingtest/.

[11] Mori, G. and J. Malik, *Recognizing objects in adversarial clutter: Breaking a visual captcha*, in: *Proc. Conf. Computer Vision and Pattern Recognition*, Madison, USA, 2003.

[12] Robinson, S., *Up to the challenge: Computer scientists crack a set of ai-based puzzles*, SIAM News **35** (2002), http://www.siam.org/siamnews/11-02/gimpy.htm.

[13] Turing, A. M., *Computing machinery and intelligence*, Mind LIX **59** (1950), pp. 433–460.