

# Dependent CAPTCHAs: Preventing the Relay Attack

Ran Halprin

Weizmann Institute of Science, Rehovot 76100, Israel

`ran.halprin@weizmann.ac.il`

Dec 16, 2007 Last Revision May 27, 2009

## Abstract

CAPTCHAs are designed to be difficult for machines to solve yet easy for humans, allowing web sites to recognize software agents impersonating as humans and attempted to abuse the system. A basic flaw in this framework is that humans may be unwillingly solving CAPTCHAs for an attacker, a concept commonly referred to as a *relay attack*. In this essay I define the notion of **dependent CAPTCHA** and show how it prevents most types of relay attacks.

## 1 Introduction

### 1.1 I, Human

Since the beginning of the internet age, the phenomenon of computerized agents that create a massive number of email accounts for spam, mine information or influence online votes has been prominent. A general solution of “reverse Turing tests” was introduced by Moni Naor [2] and later popularized as “Completely Automated Public Turing test to tell Computers and Humans Apart” (CAPTCHA) by von-Ahn et al. [4]. While there are many versions of CAPTCHAs, the most common consist of a set of alphanumeric characters (random and/or dictionary words) that are given to the user as a visually garbled image. In order to pass the challenge, the user should recognize the characters. A delicate balance is attempted by CAPTCHA creators to make the images clear enough to be read by humans, yet garbled enough as to prevent automatic recognition by Optical Character Recognition (OCR) software. A relatively easy CAPTCHA challenge is given in Figure 1.

### 1.2 The relay attack

A relay attack is an attack on the core vulnerability of CAPTCHAs. CAPTCHAs are designed to be hard for machines yet simple for humans. Thus, all that is needed to break



Figure 1: a CAPTCHA generated by the EZ-Gimpy program

the CAPTCHA is to find a human willing to solve it for them, or in other words, relay it to humans.

Spammers have been known to use different methods to convince humans to solve CAPTCHAs for them. One is CAPTCHA sweatshops that pay workers a small fee per successful CAPTCHA. Such malicious behavior is obviously very hard and perhaps even impossible to combat, though it is relatively uncommon, as the benefit of a solved CAPTCHAs is not worth so much as to justify actual payment. We will therefore concentrate on a different type of attack, the *secret relay attack*.

Consider a non-malicious user who wants to use a service on the Internet. Given a CAPTCHA by this service, the user will solve it and move on. Now consider an attacker who sets up a web site designed to attract users, such as a free pornography site. When users visit this page, they are requested to solve a CAPTCHA to proceed. These CAPTCHAs are not generated by the attacker's site, but rather relayed from the attacked site by a script which then proceeds to relay the users' answers to the attacked site. The process is summarized in Figure 2. Note that the relay attack is not a theoretic discussion, as some incident were actually reported [1] [3].

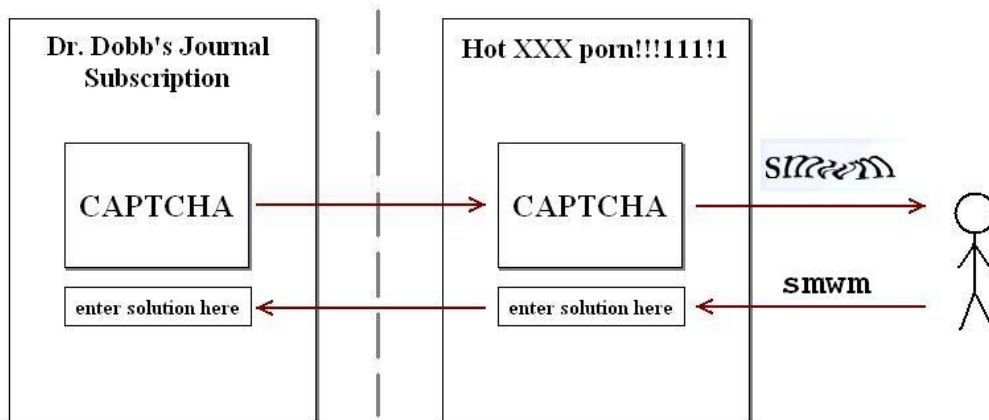


Figure 2: Schematic view of the relay attack

## 2 Dependent CAPTCHAs

### 2.1 Definition

We define the notion of Dependent captcha:

**Definition** A CAPTCHA is dependent if its solution requires data that exists on the page in which that CAPTCHA is given, but outside the CAPTCHA image itself.

We immediately notice that common CAPTCHAs are not dependent (but rather independent), i.e. solving the CAPTCHA requires nothing beyond the information supplied on the image itself. Note that according to this definition, assuming common knowledge of humans is not dependent. This means that CAPTCHAs such as “What is the color of grass” or “What is the air-speed velocity of an unladen swallow?” are independent.

### 2.2 Concrete Example

Consider for example a CAPTCHA that is constructed of two parts: a classic independent CAPTCHA, e.g. the one in Figure 1, and in addition the sentence (also visually garbled) “replace s with the first word in the third row of this paragraph”<sup>1</sup>. When placed in the context of this article, any human can easily detect this word to be “replace”, giving us the answer “replacemwn” to the CAPTCHA. Similarly, in the context of a web page, this dependent CAPTCHA is easy to solve and does not require much more trouble (perhaps sometimes even less) than a classic independent CAPTCHA.

### 2.3 Relay prevention

The dependent CAPTCHA in the above example is attached to the data of its web page, and positioning it in some other web site will not work, unless the original web page (or at least a large fraction of its text) is placed in the attacker’s site. Assuming the users are not interested in helping attackers spread scams and spam, they will avoid solving the CAPTCHA and working with the attacker’s site or service, as it becomes obvious what their intent is. Moreover, even if the attacker’s site contains information (such as pornography) that the user is interested in, their interest will drop if they have to decipher through a lot of irrelevant text in order to proceed.

## 3 Summary

This short essay introduced the notion of dependent CAPTCHA and their advantage over independent CAPTCHAs that are susceptible to relay attacks. My main conclusion is that dependent CAPTCHAs would be inherently safer than currently used CAPTCHAs and should therefore be put into use promptly.

---

<sup>1</sup>Of course such a CAPTCHA will randomly choose a different easily locatable word in the page in each run.

### 3.1 Future Work

A comprehensive study could attempt to optimize the dependent CAPTCHA both in security and in usability. By presenting users with different dependent CAPTCHAs, we could gain some insight as to which are easier and clearer for humans.

Another direction is to devise entirely different methods of human-impersonation prevention. Such methods could be developed that would not have the same basic flaws as the CAPTCHA method, either as a replacement or as an addition to CAPTCHAs.

## References

- [1] Cory Doctorow. Solving and creating captchas with free porn. <http://www.boingboing.net/2004/01/27/solving-and-creating.html>.
- [2] Moni Naor. Verification of a human in the loop, or identification via the turing test. <http://www.wisdom.weizmann.ac.il/naor/PAPERS/human.ps>, 1996.
- [3] BBC news. Pc stripper helps spam to spread. <http://news.bbc.co.uk/1/hi/technology/7067962.stm>.
- [4] L. von Ahn, M. Blum, N. Hopper, and J. Langford. CAPTCHA: Using hard AI problems for security. *Proceedings of Eurocrypt*, 2003.