Understanding of the Cyber Security and the Development of CAPTCHA

Yu Yang1,*

1School of Computer, Wuhan University, Wuhan, Hubei Province, P.R.China 430072

*562536365@qq.com

Abstract. CAPTCHA is the abbreviation of “Completely Automated Public Turing Test to Tell Computers and Humans Apart”, which is a program algorithm for distinguishing between computers and humans. It is able to generate and evaluate tests that are easy for human to pass yet are not possible for computers to. Common CAPTCHA generally contains symbols, text, pictures, and even videos, which is mainly used for human-computer verification. With the popularization of the Internet and its related applications, many malicious attacks against websites, systems and servers gradually appear. Therefore, the research on CAPTCHA is especially important. This article will briefly summarize and introduce the existing CAPTCHA technology, and summarizes the common problems of network attacks and information security. After listing the common type of CAPTCHA, it will finally propose feasible suggestions for the development of CAPTCHA.

1. Introduction

With the popularization and development of information technology, the issue of information security has drawn more and more attention. Especially the importance of cyber security has become a hot issue in many countries. One of the many solutions to common cyber security problems is to provide protection against network attacks on the client side, of which the CAPTCHA technology is the most widely used.

Completely Automated Public Turing Test to Tell Computers and Humans Apart (CAPTCHA) is a program that generates and evaluates solvable tests yet which exceed the capabilities of current computer programs [1]. This technology is now almost a standard security mechanism for defense. A good human-machine identification system must not only be humanized, but powerful enough to automatically pass (or challenge) the CAPTCHA test so as to prevent the computer program written by an attacker.

The most widely used CAPTCHA is the so-called text-based approach which relies on the complex distortions of textual images with the aim of rendering them unable to be recognized by the current pattern recognition methods. The popularity of such schemes is due to the many advantages they have, such as being intuitive to global users (user tasks are character recognition only), few localization problems (people from different countries recognize Roman characters), and have good potential to provide strong security (for example, if there is a perfect strategy for anti-attack, the computing amount of violent attacks will be huge) [2].

Earlier studies show that even though these characters are highly distorted, computers are very good at recognizing individual characters [3]. If CAPTCHA produces a known position of a character in the challenge image, this is a purely recognition problem for computers. Standard machine learning techniques, such as neural networks, can achieve high recognition success rates [4]. However, existing techniques (including machine learning) methods do not perform well in locating roles when the
character position in the CAPTCHA challenge is unknown. The problem of identifying a character's position in the correct order or segment is still a challenging issue in areas such as handwriting recognition and computer vision. In general, segmentation is computationally expensive and is usually a comprehensive issue [2].

This article discusses common cyber security issues and CAPTCHA, including a description of common types of malicious cyber attacks, common types of CAPTCHAs, and challenges that CAPTCHA has encountered and is about to encounter.

2. Background
CAPTCHA systems are sometimes referred to as "reverse Turing tests," for whose goal is to let the computer determine whether a remote client is a human. Although they are extremely important and widely used, there is no systematic way to design or evaluate them [5, 6, 7]. In fact, as we've demonstrated through in-depth research, many popular Web sites still rely on solutions that are vulnerable to match automated attacks. For instance, our program of automated DECAPTCHA tool breaking the Wikipedia. Among the 15 most widely used current scenarios, 13 are equally vulnerable to automated attacks by our tools. Therefore, a complete set of design and test principles will result in a significant need for more powerful CAPTCHAs. Previous work [5] showed that the security of the CAPTCHA depends on preventing the splitting, and we found that in our research, relying on segmentation alone does not provide a reliable defensive attack auto-attack.

Based on the techniques described in reference [8, 9], machine learning techniques [10, 11, 12] and efficient visual algorithms [10, 13, 14] can be used. We divide the automatic verification code solving process into five steps: Preprocessing, segmentation, general post-segmentation, recognition, and post-processing. While segmenting and separating a sequence of characters into single characters, identifying and recognizing these features is intuitive and generally understood, there are good reasons to consider extra pre-processing and post-processing steps as a standard procedure. For example, the preprocessing may remove the background pattern or eliminate other additional images that may interfere with the segmentation, while the segmented steps may "clean" the segmented output to regulate the size of each image, otherwise performing a different segmentation step. Recognition and post-processing can improve accuracy, for example, using spell checking for any verification code based on actual words (such as Slashdot).

3. Common attacks
In this section, the article will mainly introduce the main types of existing cyber attacks, and its attack principle. In general, there are four broad categories: read attacks, operational attacks, cheating attacks, and flood attacks.

3.1. Read Attacks
Read attacks primarily involve all relevant attacks that get information from the victim. Such attacks get the IP address of the organizational structure, perform port scanning and vulnerabilities scans in those address ranges, and finally gain access to compromised hosts.

3.1.1. Reconnaissance Attacks. Designed primarily to give attackers access to more information about victims, reconnaissance attacks can be proactive and reactive, and in almost all cases successful reconnaissance attacks greatly increase the likelihood of subsequent attacks succeeding since the attacker gets more information about the victim.

3.1.2. Sniffing attack. This can be referred to as some form of sniffing attack when an attacker catches a packet from the cable, or a packet passes through an attacker's system. The purpose of sniffing attacks is to read the information to obtain intelligence so that attackers understand the target system, so that the sniffed protocol information must be sent in plaintext rather than cipher text, then the
sniffing is successful. The main information is as follows: Authentication information, network management information, confidential affairs and so on.

3.2. Operational Attacks

3.2.1. Network manipulation attacks. The most common is IP fragmentation, where attackers deliberately fragment traffic flow in an attempt to bypass network-based (IDS or firewall) or application-based security controls. In addition to IP fragmentation, attackers can also perform source routing attacks, where an attacker can use source routing to choose an attack path in the network. Source routing has almost no legal application and is off by default on most routers. There are many attackers who can exploit the IP, TCP, and UDP protocols to attack.

3.2.2 Application manipulation attacks. Application manipulation attacks are attacks that take place at the application level. Its main use of application design or implementation of the program flaws, the most famous application manipulation attack is a buffer overflow attacks. Such as web application attacks, and insecure Common Gateway Interface (CGI).

3.3. Cheating Attacks

A cheating attack is that an attacker could cause a user or a device in the system to believe that the information came from a source that did not actually send the message, which can happen in any virtually location that is not certified for weakness or network traffic.

3.3.1. MAC cheat. It is a very straightforward attack. Attack systems use this attack to change their MAC address to a trusted system address. In an Ethernet environment, the content addressable memory (CAM) table on the switch tracks MAC addresses, VLANs, and MAC addresses. When an attacker changes the MAC address to the address of another system connected to the switch, the CAM table will be updated and the switch master considers a machine to move from one location to another.

3.3.2. IP cheat. An attacker simply enters the system's original packet driver, and an attacker can send a packet containing the IP header.

3.3.3. Transmission cheat. Transmission cheat means that communication spoofing is successfully implemented at the transport layer, such as UDP spoofing, TCP spoofing, and identity spoofing.

3.4. Flood Attacks

The flood attack is when an attacker sends too much data to some network resources. These network devices can be routers, switches, applications, hosts, or network links.

3.4.1. MAC Flooding. MAC flooding refers to the fake source MAC address and destination address of the packet from the attacker's system sent to the Ethernet link for MAC address occupied on the exchange in the CAM position, the CAM table capacity is limited. When CAM is full, other hosts can only flood the local LAN, allowing attackers to sniff these frames.

3.4.2. Network flooding. Network flooding is generally designed to consume the available bandwidth of the network link. This attack sends legitimate traffic to cables flooded with fraudulent traffic, greatly reducing the amount of legitimate traffic on the available broadband, often targeting internet links on the network.

4. Common types of CAPTCHA

4.1. Graphic CAPTCHA
The Graphic CAPTCHA is the earliest and most widely used CAPTCHA. Common graphic CAPTCHA codes often contain symbols, numbers, letters, etc., and then the verification code can be enhanced by aided means to improve the recognition difficulty. Common aids as shown in Table 1 and Table 2 below.

**Table 1.** Common CAPTCHA reinforcement method

| Reinforcement method       | Noise, fonts distortion | Different styles, alphabetic shading, alphabetical adhesion, background color interference | Main interference line, background color interference, interference background letters, font distortion, alphabetical adhesion | Background interference line, background color interference, background letter interference, font distortion, alphabetical adhesion, fonts hollow |
|---------------------------|-------------------------|------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|
| Fonts distortion          |                         |                                                                                          |                                                                                                  |                                                                                                  |
| Fonts Adhesion            |                         |                                                                                          |                                                                                                  |                                                                                                  |
| Fonts hollow              |                         |                                                                                          |                                                                                                  |                                                                                                  |
| Mix of fonts              |                         |                                                                                          |                                                                                                  |                                                                                                  |
| Main interference line    |                         |                                                                                          |                                                                                                  |                                                                                                  |

**Table 2.** Example of CAPTCHA hardening method

4.2. **SMS verification code**

SMS verification code is a valid verification code system by sending a verification code to the phone. SMS verification code is commonly seen in registration, the stage of forgetting the password, confirming the order and other stages, especially when it comes to the user's personal sensitive behavior. In order to confirm the operation is conducted by the user, the SMS authentication code is usually used for secondary verification. At the same time, SMS verification code is also one of the most common types of CAPTCHA.

4.3. **Voice verification code**

Compared with the SMS verification code, the voice verification code has the following advantages: 1) the voice verification is more effective on the anti-brush single effect; 2) the voice verification code has a higher arrival rate; and 3) the user experience is more friendly

4.4. **Sliding verification code**
Now there are a lot of Internet companies begin to use the sliding verification code for verification. Seemingly a simple sliding operation, the wind control engine and the corresponding rules are behind as a guarantee, the specific verification process is shown in Figure 1.

![Sliding verification process](image)

**Figure 1.** Sliding verification process

First, the user slides the CAPTCHA to the specified location. Upon completion, it will send back various encrypted information to the server. In order to judge whether there is any abnormality, the rules include user IP, operation path, UA, COOKIE, device fingerprint, etc. If the rule is not achieved, it will release verification, if achieved, it will pop up a secondary calibration, only after passing the verification can be released.

5. **Main challenges**

5.1. **Database limitations**

No matter what form of verification code is, it has a database to store pictures, text, voice and other information, if the database is limited or updated too slowly, malicious attackers will easily traverse the entire database, making them easily to master the crack code verification method.

5.2. **Poor experience**

Many users encounter this type of problem when entering a passcode, such as too obscured images, illegible numbers or letters, such as "0" and "O," or images that are too small. These problems directly increase the difficulty of the user identification code, when the user fails to verify the output of multiple answers and the system does not prompt, the user is likely to cease operation, so that the site will lose potential users.

5.3. **New technology brings challenges**

As mentioned before, the development of new technologies will bring new challenges to CAPTCHA technology. Machine learning, especially deep learning, greatly improves the computing power of the computer, reduces the computing time, and greatly reduces the difficulty of cracking the verification code.

6. **Conclusion**

With the development and popularization of information technology, the problem of cyber security has become increasingly prominent. This article focuses on the forms of network attacks that are common today, and discusses the types and features of CAPTCHAs according to one of the solutions. The CAPTCHA design is an interdisciplinary topic where the expertise from multiple domains plays an important role. The design of CAPTCHAs with good robustness and usability has been the research topic for a long time. The current collective understanding of this topic is still in its infancy. In order to develop the CAPTCHA design from the arts to science still need a lot of research. Our experience shows that CAPTCHA will lead to the development of more robust systems through similar evolutionary processes, such as cryptography and digital watermarking, and through successful iterations.

**References**
[1] L von Ahn, M Blum and J Langford. “Telling Humans and Computer Apart Automatically”, CACM, V47, No2, 2004.
[2] K Chellapilla, K Larson, P Simard and M Czerwinski, “Building Segmentation Based Human-friendly Human Interaction Proofs”, 2nd Int’l Workshop on Human Interaction Proofs, Springer-Verlag, LNCS 3517, 2005.
[3] K Chellapilla, K Larson, P Simard, M Czerwinski, “Computers beat humans at single character recognition in reading-based Human Interaction Proofs”, 2nd Conference on Email and Anti-Spam (CEAS), 2005.
[4] P Simard, D Steinkraus, J Platt. “Best Practice for Convolutional Neural Networks Applied to Visual Document Analysis”, International Conference on Document Analysis and Recognition (ICDAR), IEEE Computer Society, Los Alamitos, pp.958-962, 2003.
[5] K. Chellapilla, K. Larson, P.Y. Simard, and M. Czerwinski. Computers beat humans at single character recognition in reading based human interaction proofs (hips). In CEAS, 2005.
[6] K Chellapilla and P Simard. Using machine learning to break visual human interaction proofs. In MIT Press, editor, Neural Information Processing Systems (NIPS), 2004.
[7] J. Yan and A.S.E. Ahmad. Breaking visual captchas with naive pattern recognition algorithms. In ACSAC 2007, 2007.
[8] K. Chellapilla and P.Y. Simard. Using machine learning to break visual hips. In Conf. on Neural Information Processing Systems, NIPS 2004, 2004.
[9] J. Yan and A.S. El Ahmad. A Low-cost Attack on a Microsoft CAPTCHA. In Proceedings of the 15th ACM conference on Computer and communications security, pages 543–554. ACM, 2008.
[10] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. Computer Vision–ECCV 2006, pages 404–417, 2006.
[11] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, 1998.
[12] R. Quinlan. Machine Learning. Morgan Kaufmann Pub, 1992.
[13] Duda R O, Hart P E. Use of the Hough transformation to detect lines and curves in pictures. Communications of the Acm, 15(1):11-15, 1972.
[14] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images*. Journal of Applied Statistics, 20(5):25–62, 1993.