Bot, Cyborg and Automated Turing Test
(Or "Putting the Humanoid in the Protocol")

Jeff Yan

School of Computing Science
University of Newcastle upon Tyne, UK
Jeff.Yan@ncl.ac.uk

Abstract. The Automated Turing test (ATT) is almost a standard security technique for addressing the threat of undesirable or malicious bot programs. In this paper, we motivate an interesting adversary model, cyborgs, which are either humans assisted by bots or bots assisted by humans. Since there is always a human behind these bots, or a human can always be available on demand, ATT fails to differentiate such cyborgs from humans. The notion of "telling humans and cyborgs apart" is novel, and it can be of practical relevance in network security. Although it is a challenging task, we have had some success in telling cyborgs and humans apart automatically.

1 Introduction

Nowadays, CAPTCHA (Completely Automated Turing Test to Tell Computers and Humans Apart) [2] is almost a standard security mechanism for addressing undesirable or malicious bot programs such as:

- Voting bots, which could cast thousands of votes as masquerading humans in online polls [2];
- Email account registration bots, which could sign up for thousands of accounts every minute with free email service providers [2];
- Email spam bots, which could automatically send out thousands of spam messages every minute [2];
- Weblog bots, which could post comments in weblogs pointing both readers and search engines to irrelevant sites; and
- Search engine bots, which could automatically register web pages to raise their rankings in a search engine.

The basic idea of CAPTCHA is to force there to be a human in the loop – it works as a simple two-round authentication protocol as follows.

S(ervice) → C(lient): a CAPTCHA *challenge*
C → S: *response*

A CAPTCHA *challenge* is a test that most humans can pass but current computer programs cannot pass. Such a test is often based on a hard, open problem

B. Christianson et al. (Eds.): Security Protocols 2006, LNCS 5087, pp. 190–197, 2009.

in AI, *e.g.* automatic recognition of distorted text, or of human speech against a noisy background. Differing from the original Turing test [1], CAPTCHA challenges are automatically generated and graded by a computer. Since only humans are able to return a sensible *response*, an automated Turing test embedded in the above protocol can verify whether there is a human behind the challenged computer.

Although the original Turing test was designed as a measure of progress for AI, CAPTCHA is a security mechanism. It is straightforward to apply CAPTCHA to prevent a bot say from distributing spam emails: an email is not forwarded until it is confirmed using a CAPTCHA challenge that there is a human behind the computer sending the message.

However, in this paper, we show that CAPTCHA is not a panacea for dealing with the threat of undesirable or malicious bots. We motivate an interesting adversary model, cyborgs. A cyborg, according to the *Oxford English Dictionary*, is a creature that is part human and part machine. In reality, it can be either a human assisted by bots, or a bot assisted by humans. Since there is always a human behind the bot(s), or a human can always appear behind the bot(s) on demand, CAPTCHA will fail to differentiate such cyborgs from humans. We also show the novel notion of "telling humans and cyborgs apart" can be of practical relevance in network security.

The rest of this paper is organised as follows. Section 2 looks into two popular bots and shows why CAPTCHA is doomed to fail to deal with such bots. Section 3 presents a simple model explaining when CAPTCHA works in dealing with bots, and when it does not. Section 4 briefly introduces some of our work in automatically telling humans and cyborgs apart, and Section 5 concludes.

2 CAPTCHA: Not a Panacea!

In this section, we use two popular cheating bots in online multiplayer games to show that CAPTCHA is not a panacea for dealing with the threat of undesirable or malicious bots.

2.1 MMORPG Bots

MMORPG stands for "Massive Multiplayer Online Role-Playing Game". This genre of games runs an evolving virtual world on a server. Thousands of human players can play on the server over the Internet at the same time, each playing the role of a virtual character, which can for example be a medieval knight or an ancient Chinese kung-fu warrior depending on the game settings.

An MMORPG bot is a program that automatically plays the game on behalf of a human player. Such a program is usually game-specific, and driven by predefined scripts. MMORPG bots can be easily configured to perform many activities that a human player would do in the game, such as move, kill, cast spells and collect items. Some such bots also allow a human player to input instructions to directly control his character at any time. That is, MMORPG bots can be both script-driven and interactive.

With the help of MMORPG bots, which never get tired, cheaters can reap rewards with much less effort than their honest counterparts. The use of such bots is usually forbidden by game operators. The common practice for MMORPG game operators fighting against bots relies on human policing: a game master patrols game zones, recognises and questions suspicious players. Employing CAPTCHA would appear to be a good solution for keeping bots out of MMORPGs at a first glance, since MMORPG bots are in fact usually left unattended once they are connected to a game server.

Before attempting to reach any conclusion regarding the use of CAPTCHA in MMORPGs, we perform a simple security analysis as follows. We make the standard assumptions that the game server is under the control of a game operator, and secured by standard technologies, and that a game client can be under total control of a cheater (since it is run on his own computer, which is hard or expensive to be made tamper-proof). We also assume that the CAPTCHA facility is properly implemented by qualified security engineers, having addressed the following security issues.

- All critical things such as randomness used for generating CAPTCHA challenges are implemented in the game server. In addition to supporting game play, a client merely displays CAPTCHA challenges, accepts responses from a player, and returns them to the server for verification.
- An adversary should not be able to gain, by intercepting network traffic, anything that helps solve a CAPTCHA challenge. A naive mistake made by one student of mine was as follows. Each random string used for CAPTCHA challenges was generated on the server side, and it was transmitted in plaintext to a client. The client then distorted the string, embedded the distorted text, together with some noise information, into an image, and finally displayed the image. This implementation would allow a bot to easily pick up the right answers from the traffic! Encryption would not help, since the client is assumed to know everything. Rather, each CAPTCHA challenge should be prepared by the server, and then transmitted in an image format across the network to the client.
- Dictionary based replay attack. Adversaries should not be able to benefit by collecting old challenges and their answers into a dictionary, and then replaying old correct answers.

Unfortunately, such a CAPTCHA based bot defence is still vulnerable to the following attacks:

- **Man in the middle (MITM) attack.** This is not new. It was alleged that a spammer could make use of the MITM attack, shifting the load of solving CATPCHA challenges to porn site visitors [6]. This kind of attack is certainly doable, and could be exploited by MMORPG bot users.
- **Outsourcing attack.** Bot users can outsource both their game play and the task of solving CAPTCHA challenges to people in low-paying countries.
- **Housewife attack.** A key observation is that it is often possible for a bot to differentiate each CAPTCHA challenge from other game events. Thus,

a human can be alerted by the bot to answer the challenge in time. Bot attending in MMORPGs could then become an attractive profession for housewives, who would make money by attending their bots occasionally (*i.e.*, upon each alert), while looking after their household business as usual.
- **Collusion attack.** Bots can be made to communicate with each other, and then each alert can be propagated across the bot network. Therefore, a CAPTCHA challenge can be attended by either a cheater or one of his friends, whoever is available.

There is also a usability concern (beyond accessibility) when integrating CAPTCHA into MMORPG games: you cannot issue CAPTCHA challenges very often unless you want to ruin the fun of game play! Such a concern makes it even easier for all the above attacks to succeed.

2.2 Aiming Robots

Aiming robots (or aimbots for short) have frequently been used as a cheating tool in online multiplayer shooting games, *i.e.*, first-person shooter (FPS) games. An aimbot works as either a client hook or a proxy sitting between a client and a game server. In both cases, the bot tracks the movements and locations of players. Whenever a cheater issues a Fire command, his aimbot could automatically pick a target, and point the cheater's weapon straight at the selected target. An advanced aimbot can also pretend to act like an ordinary human being, either by switching itself on and off periodically, or by intentionally missing targets from time to time.

CAPTCHA does *not* appear to be applicable to prevent aimbots, since

- First, a human player is always behind his aimbot. Such a bot largely does aiming (as well as target tracking) only, and it cannot perform other in-game tasks on behalf of a player, which usually require human involvement due to characteristics of such games. Thus, the player could always be available to respond to CAPTCHA challenges, if any.
- Second, FPS games are usually fast paced. A CAPTCHA challenge would be too disruptive for game play. Due to this usability concern, few game designers and players, if any, would consider CAPTCHA as an acceptable solution. (Such a usability concern is not so serious in MMORPGs, which are usually slower paced.)

3 A Simple Explanation

Apparently, the following three notions are different:

- "Breaking CAPTCHA"

  "Breaking CAPTCHA" usually means "breaking CAPTCHA challenges", and it is as hard as solving the underlying AI problem.

– "Breaking CAPTCHA based authentication"

  "Breaking CAPTCHA based authentication" is to fail the CAPTCHA protocol's authentication purpose. Surely, either "breaking CAPTCHA" or the MITM attack will lead to such a protocol-level failure, alone. But the MITM attack does not need to solve the underlying AI problem.

– "Defeating CAPTCHA based bot defence"

  It is true that unless the underlying AI problem is solved, CAPTCHA challenges cannot be answered without having forced there to be a human behind the challenged computer. This, however, does not guarantee that the CAPTCHA mechanism is a good defence against all bots. First of all, there are circumstances in which arranging such human involvement is quite feasible. Thus there is the possibility that all CAPTCHA challenges to a bot could be properly answered, so bots would not be stopped – instead, they would remain undetected.

  On the other hand, as the earlier discussion of the MITM attack, the outsourcing attack and the collusion attack have revealed, the CAPTCHA mechanism could ensure that there was a human behind a computer, but it might not be the "right" human!

  Moreover, usability concerns can also render the CAPTCHA mechanism less effective as expected or even useless in defending against some bots.

The CAPTCHA mechanism can be refined to some extend to address the above "not the right human" problem. For example, when challenges that demand a *specialised skill* or a certain *cognitive ability* to solve are used, they will force there to be a human with the required skill or capability in the loop. For instance, to use "life and death" puzzles in Go games as CAPTCHA challenges can provide a guarantee that people behind a computer are humans who understand Go. In addition, a biometric signature such as keystroke dynamics might be embedded in one's response to a CAPTCHA challenge. All this might be sometimes sufficient for authentication purposes. But none can be a generic way of defeating undesirable or malicious bots due to the obvious disadvantages.

To explain when the CAPTCHA mechanism works to defeat bots and when it does not, we define a simple function as follows to describe the cost, denoted by C, that an adversary has to bear to beat such a bot defence.

$C = f \times t \times c$
f: The frequency of CAPTCHA challenges, *i.e.*, the number of challenges occurring per unit time.
t: Time period during which CAPTCHA challenges will occur.
c: Average cost for an adversary to solve a single CAPTCHA challenge.

The number of challenges an adversary has to solve in the given t time is defined by $f \times t$. To simplify our discussion, we assume that t is constant.

Thus, the CAPTCHA mechanism is effective at defeating bots when $f \times c$ is prohibitively high. The authors of [2] seem to have implicitly assumed that this

condition always holds in various bot defence scenarios, but this assumption is not necessarily true. Apparently, all the bots listed in Section 1, as well as others not discussed in this paper, are similar to MMORPG bots in that a CAPTCHA based defence will be vulnerable to all four attacks discussed in Section 2.1.

The condition $C = 0$ or $C \approx 0$ depicts scenarios where the CAPTCHA mechanism fails to prevent bot attacks. Such scenarios include the following cases.

- When the hard AI problem exploited for CAPTCHA tests becomes tractable by a program, $c = 0$ and thus $C = 0$. (We ignore the cost that an adversary has to bear for writing such a program. For example, the program might be downloaded from the Internet, free of charge.)
- When the adversary can make use of the MITM attacks, *e.g.*, shifting the load of solving CATPCHA challenges to porn site visitors, $c = 0$ and thus $C = 0$. (Similarly, we assume that the cost of implementing such MITM attacks are negligible.)
- In the aimbot case, $f = 0$ or $f \approx 0$ (otherwise it would be disruptive for game play), and thus C is negligible.

When f is large and $c \neq 0$, an adversary can defeat the CAPTCHA mechanism by lowering his average cost c, *e.g.* by outsourcing the task of solving each challenge.

In the MMORPG bot case, f must be small, and c can be lowered by the adversary through either the outsourcing attack or the housewife attack. That is, an adversary can also work around the CAPTCHA based bot defence.

4 Tell Humans and Cyborgs Apart

An aimbot, together with the human player using the bot, is a good example of cyborg. So is the combination of an MMORPG bot and a human behind it (a housewife, a porn site visitor or an outsourcing worker). Since there is always a human behind the bots, or a human can always appear behind them on demand, the CAPTCHA mechanism cannot tell humans and such cyborgs apart at all.

People might wonder: given that an adversary has to arrange that his bot is attended, why does he bother to use it in the first place? This can be simply explained as follows.

- First, bots can be superior to ordinary human beings in some aspects. For example, an aimbot can achieve better aiming accuracy; an MMORPG bot can be more patient at repeating tedious tasks. Thus, while bots do things that they are good at, the adversary just has to attend CAPTCHA challenges, and probably some other easy tasks as well. This is a good analogy of "Render unto Caesar what is Caesar's and render to God what is God's".
- Furthermore, while f or c is reduced due to usability concerns in some scenarios, the amount of human labour required to attend CAPTCHA challenges will be reduced, making the adversary's life even easier!

To tell humans and cyborgs apart automatically is a real security concern in online games, one of the most popular Internet applications, as well as in other contexts. For example, with the help of an aimbot, a cheater can easily achieve a high aiming accuracy and beat his opponents, gaining himself an unfair advantage over honest players. A cheater can also run bots in MMORPG games, collecting virtual items without real play and then selling them, for example on eBay, for real money – thus achieving an unfair financial gain. Botting (*i.e.* running bots) violates fair play, an important security concern in such settings [3,4]. Now that first-person shooters are emerging as spectator-games[1], fairness enforcement becomes an even more serious issue in these competitive games. In addition, botting in MMORPGs can also undermine the delicate balance of the game world, a critical factor affecting the success of such games.

Although the task is challenging, we have had some success in automatically telling cyborgs and humans apart. In the following, we briefly discuss our work on differentiating honest game players from cheaters assisted with bots (*i.e.* cyborgs). Our discussion is focused on the aimbot case, but our method, in principle, can be applied to identify MMORPG bots as well.

4.1 Aimbot Detection

We have developed a Dynamic Bayesian Network (DBN) based statistical inference approach to distinguish honest players from aimbot cheaters in FPS games. In our DBN model [5], the probability distribution of a player's aiming accuracy is dependent on the following random variables:

- Whether the player is cheating or not, denoted by the random variable $\mathcal{C}$;
- Whether the player is moving or not;
- Whether the aimed target is moving or not;
- Whether the player is changing the aiming direction;
- The distance between the player and the aimed target.

By learning the conditional probabilities from a set of training data, our algorithm can infer the probability of the missing variable $\mathcal{C}$ in other sets of data. In our initial experiments, this algorithm has differentiated all aimbot cheaters from honest players with no false positives.

Our ongoing work is to refine our algorithm so that we can distinguish between aimbot cheaters and professional players, who may achieve an outstanding aim accuracy in most cases. We rely on a simple intuitive observation: while being able to achieve a high aiming accuracy, a super human player is (by definition) also good at other aspects of the game play. But this is not necessarily the case with aimbot cheaters. (Indeed there might be some all-around aimbot cheaters who perform well in all these aspects, not much could be done to detect them automatically however.)

[1] Like traditional sports such as football and basketball, FPS tournaments now attract lots of spectators. Top professional game players (or the so-called "cyber athletes") can make a lucrative living with commercial sponsorships just like traditional sports stars.

So given that super players and aimbot cheaters both pass the threshold used in our previous DBN model [5] to identify cheaters, an additional correlation test will help: if there is a positive correlation between a player's high aiming accuracy and his other "performance factors" such as movement agility, mistake counts/frequencies, appropriateness of weapon choices and efficiency of ammunition use, we are more comfortable (than before) assuming that he is a super human player; otherwise we are more confident that we are dealing with a cyborg, *i.e.*, the suspect is an aimbot cheater.

5 Conclusion

CAPTCHA is not a panacea in dealing with the threat of undesirable or malicious bots. Often, it is insufficient to simply verify that there is a human behind a computer. Usability concerns beyond accessibility can also render CAPTCHA inappropriate for some application contexts. Instead, "telling humans and cyborgs apart automatically" can be a relevant and generic security problem, to which CAPTCHA fails to provide a solution. However, it appears to be promising to automatically differentiate cyborgs from humans by recognising the characteristics of both the machine and human facets of cyborgs.

Acknowledgement

The author thanks Brian Randell for valuable discussions and comments, Peter Ryan for suggesting this paper's subtitle, and Richard Clayton for pointing us to his investigation of the authenticity of the MITM attack story in [6] at this workshop.

References

1. Turing, A.M.: Computer machinery and intelligence. Mind 59(236), 433–460 (1950)
2. von Ahn, L., Blum, M., Langford, J.: Telling Humans and Computers Apart Automatically. Communications of the ACM 47(2) (February 2004)
3. Yan, J.: Security Design in Online Games. In: Proc. of the 19th Annual Computer Security Applications Conference (ACSAC 2003), Las Vegas, U.S.A, December 2003. IEEE Computer Society, Los Alamitos (2003)
4. Yan, J., Randell, B.: A Systematic Classification of Cheating in Online Games. In: Proc. of the 4th ACM SIGCOMM Workshop on Network & System Support for Games (NetGames 2005), October 10-11. ACM Press, New York (2005)
5. Fung, Y.S., Lui, J., Liu, J., Yan, J.: Detecting Cheaters for Multiplayer Games: Theory, Design and Implementation. In: The second IEEE International Workshop on Networking Issues in Multimedia Entertainment, Las Vegas, USA (January 2006)
6. Clayton, R.: Automating responses to email challenges (February 2006), `http://www.cl.cam.ac.uk/~rnc1/cr/index.html`