A Systematic Classification of Cheating in Online Games

Jeff Yan
School of Computing Science
University of Newcastle
NE1 7RU, UK
Jeff.Yan@ncl.ac.uk

Brian Randell
School of Computing Science
University of Newcastle
NE1 7RU, UK
Brian.Randell@ncl.ac.uk

ABSTRACT

Cheating is rampant in current game play on the Internet. However, it is not as well understood as one might expect. In this paper, we summarize the various known methods of cheating, and we define a taxonomy of online game cheating with respect to the underlying vulnerability (what is exploited?), consequence (what type of failure can be achieved?) and the cheating principal (who is cheating?). This taxonomy provides a systematic introduction to the characteristics of cheats in online games and how they can arise. It is intended to be comprehensible and useful not only to security specialists, but also to game developers, operators and players who are less knowledgeable and experienced in security. One of our findings is that although cheating in online games is largely due to various security failures, the four traditional aspects of security – confidentiality, integrity, availability and authenticity – are insufficient to explain it. Instead, fairness becomes a vital additional aspect, and its enforcement provides a convincing perspective for understanding the role of security techniques in developing and operating online games.

Categories and Subject Descriptors

D.2.0 [**Software Engineering**]: General—*protection mechanisms*; K.6.5 [**Management of Computing and Information Systems**]: Security and Protection; K.8.0 [**Personal Computing**]: General—*games*; J.m [**Computer Applications**]: Miscellaneous; K.4.4 [**Computers And Society**]: Electronic Commerce—*Security*; D.4.6 [**Operating Systems**]: Security and Protection; C.2.0 [**Computer–Communication Networks**]: General—*security and protection*

General Terms

Security, Design

NetGames'05, October 10–11, 2005, Hawthorne, New York, USA.

Keywords

Online computer games, security, cheating, taxonomy

1. INTRODUCTION

While online games are fast becoming one of the most popular applications on the Internet [15], cheating has emerged as a notable phenomenon in current game play on the Internet. Recent research has suggested that cheating is in fact a new, major security concern for online computer games [19, 22, 23]. Therefore, a careful investigation of online cheating can benefit the study of security in this representative Internet application.

However, cheating has not been studied as thoroughly as one might expect. For instance, although online cheating is rampant in games, there is no generally accepted definition for it.

Three reasons may explain this fact. First of all, online game cheating is a relatively new topic for security researchers, although many game players have been familiar with it for a considerable time. Second, the variety of online games now in existence has made cheating a complicated phenomenon. For example, there are a number of entirely different game genres, and each may give rise to varied forms of cheating. Third, many novel cheats have been invented that are different from but often entangled with ordinary security attacks.

In this paper, we systematically examine cheating in online games while adopting the following definition for it, which is a refined version of a previous definition used in [22].

> *Any behaviour that a player uses to gain an advantage over his peer players or achieve a target in an online game is cheating if, according to the game rules or at the discretion of the game operator (i.e. the game service provider, who is not necessarily the developer of the game), the advantage or the target is one that he is not supposed to have achieved.*[1]

Specifically, we present a classification scheme for online game cheating, in the expectation that by categorizing various cheats, our understanding of this phenomenon will be extended, and useful patterns and conclusions can be identified, and that it will be possible to protect online game systems against cheating using these knowledge.

[1]At present the preponderance of cheating in online games is carried out by male game players, so for linguistic convenience in the rest of this paper we will appear to imply that all cheaters are male.

Our classification scheme provides a three dimensional taxonomy for online cheating, in which the classification is made with respect to the underlying vulnerability (what is exploited?), cheating consequence (what type of failure can be achieved?) and cheating principal (who is cheating?), respectively.

Our taxonomy provides a systematic view of the characteristics of cheats in these games and how they can arise, and it is aimed at being comprehensible and useful not only to security specialists, but also to game developers, operators and players. For example, game developers can learn how previous online game systems have failed to prevent cheating. Game operators and players can learn to recognize various cheats and manage the risks of encountering cheaters. For security specialists, especially those who are new to the topic of online game security, our taxonomy is a good starting point to understand the cheating phenomenon in online games.

We have found that many cheats in online games are in fact due to poor or non-existent security designs in these systems – in other words, they are insecure by design. This indicates that many game developers may lack the relevant security expertise that is essential for defending against some cheats. While our taxonomy organises common cheating forms in a way that is not only structured, but also understandable to both security and game specialists, it may constitute a shared framework in which both sides can better communicate about and understand game cheating as well as their defence. Thus, it may encourage a better collaboration between both sides, which not only can from the beginning promote good system designs that eliminate or minimize the possibility of being exploited by online cheaters, but also helps identify residual cheats during system evaluation.

There is a legitimate concern that this taxonomy could assist those who would cheat in online games. Partly for this reason, where possible, we will try to discuss cheating cases at a level of detail that illustrates the underlying principles without giving a "cheater's cookbook".

This paper extends our previous work in [22, 23], and is organized as follows. Section 2 reviews the related work in this field. Section 3 lists common cheating forms as they have occurred or might occur in online games. Section 4 describes our three dimensional taxonomy. All common cheating forms identified in the previous section are classified using this taxonomy. Section 5 presents some results deduced from our taxonomy. Finally, Section 6 concludes.

2. RELATED WORK

A number of researchers [21, 2, 13] have investigated some interesting cheats in online games and their defence, but their approach has largely been on a case by case basis.

Several authors have attempted to define a framework for classifying and understanding online game cheating. For example, Davis [9] categorized traditional forms of casino cheating and discussed their potential counterparts in online games. However, a casino is not representative enough to reflect all forms of online game settings, in which cheating may occur with differing characteristics.

Pritchard [19] reported many real cases of online cheating that have occurred in various games, and classified them into a framework of six categories. However, his classification is ad hoc and not comprehensive. Indeed, a lot of online game cheats do not readily fit into any of his categories.

Yan and Choi [22] reported a more thorough effort identifying eleven common cheating forms in online games. In addition, Yan [23] examined, using a simple classification scheme, cheats that have occurred or might occur in online contract bridge communities, and discussed how these cheats would impact the system design of online bridge.

There is also a large amount of literature investigating the definition of taxonomies for security vulnerabilities, attacks or intrusions in a general setting. For example, Landwehr et al [11] constructed a classification of security flaws in software with respect to genesis (how did the flaw enter the system?), time of introduction (when did it enter the system?) and location (where in the system is it manifested?). But this classification largely focused on flaws in operating systems. Neumann et al [18] gave a taxonomy of attacks with respect to techniques used to launch the attacks. The MAFTIA project [20] proposed a taxonomy for intrusion detection systems and attacks. Lindqvist et al [14] discussed the desired properties for a taxonomy, and defined a taxonomy of intrusions with respect to intrusion techniques and results. However, as will be discussed below, while a game player can cheat by launching an "attack" or "intrusion", cheating in online games can also have some unique manifestations.

3. CHEATING IN ONLINE GAMES

Before defining our taxonomy, we identify all cheating forms known to us, as they have occurred or might occur in online games. We also briefly discuss some general properties of these cheats. Readers are encouraged to communicate additional cheating forms and their manifestation cases to the authors so that we can better understand online game cheating, and further refine our taxonomy.

3.1 Common Cheating Forms

We identified eleven common cheating forms in [22]. While continuing our study on game cheating, however, we have seen the need of expanding and refining this list, since new cheating forms have been identified and our understanding about game cheating has also increased. In the following, we present a revised list, which classifies cheats into 15 categories. Those that are new, or are significantly revised versions of the categories listed in [22], are marked with asterisks.

A:* Cheating by Exploiting Misplaced Trust. Many cheats involve tampering with game code, configuration data, or both, on the client side [19]. A cheater can modify his game client program, data, or both, and then replace the old copy with the revised one for future use. Alternatively, the modification or replacement of code and data can be done on the fly.

A cheater can also tamper with his game client program on the fly to access sensitive game states which are otherwise unavailable to the game player. Typical examples include the hacker program which displayed all army formation information in the popular oriental *Si Guo* game (or "Four States" in English) [23], and the "map hack" [19] which has been often used to reveal unexplored map areas on the display in real time strategy games.

This form of cheating is really due to misplaced trust. Too much trust is placed on the client side, which in reality cannot be trusted at all because a cheating player can have the total control over his game client. Countermeasures based on security by obscurity approaches such as program obfuscation will eventually fail in the fight against this form of cheating, because they try to protect the wrong thing.

B: Cheating by Collusion. People can collude with each other to gain unfair advantages over their honest opponents in online games. For example, the so-called "win-trading" was a collusion cheat widely seen in the popular *StarCraft* [6] game, in which two cheaters colluded with each other as follows. Each lost to the other alternately in the ladder competition. The loss that one took would give the other a victory point, raising his ladder rank, and vice versa. Thus, both of them could climb to top positions in the ladder without playing a legitimate game.

Collusion cheating has also been widely seen in online bridge, which facilitates the play of contract bridge over the Internet. Contract bridge is a four-person card game played between two pairs of partners. Unlike chess, in which all pieces are on the board and known to each side, bridge is a game with hidden information. Each player knows only a subset of 52 cards in the course of game play. However, by illicitly exchanging card information over the telephone, instant messenger or the like, collusive cheaters can gain huge advantages over honest bridge players. We have discussed this collusion cheat and various other collusion scenarios as well as their countermeasures in [23].

C:* Cheating by Abusing the Game Procedure. This form of cheating may be carried out without any technical sophistication, and a cheater simply abuses the operating procedure of a game. One common case that we have observed in many online games is *escaping*: a cheater disconnects himself from the game system when he is going to lose.

Another example is *scoring cheating* that we have personally observed in a popular online Go community. It abuses the scoring procedure as follows. When a game is finished in Go, "dead" stones must be identified and then removed, both by hand, before the system can determine which side wins this game. During this scoring process, however, a cheating player may stealthily remove "alive" stones of his opponent, and then "overturn" the game result. (When the size of territory occupied by each side is similar, this cheating may easily escape the awareness of the cheated player, especially when he is not a strong player.)

D: Cheating Related to Virtual Assets. Virtual characters and items acquired in online games can be traded for real money. A cheater might offer a virtual item, receive real money for the item but never deliver it as agreed. Such kind of cheating has been reported in Korea [10].

E:* Cheating by Exploiting Machine Intelligence. Artificial intelligence techniques can also be exploited by a cheating player in some online games. For example, the advancement of computer chess research has produced many programs that can compete with human players at the master level. When playing chess online, a cheater can look for the best candidates for his next move by stealthily running a strong computer chess program.

This is in fact cheating due to the superiority, in this particular situation, of machine intelligence over that of an ordinary human being. It can happen in many other online games (including online versions of many traditional board and card games), depending on two factors: 1) the properties of the game: whether the game can be modelled as a computable problem, and 2) the maturity of AI research into such games. For example, online Go players do not worry about this form of cheating, since the state of the art of AI research can produce only very weak computer Go programs – the strongest one at present can be easily beaten by an amateur human player [17].

F:* Cheating by Modifying Client Infrastructure. Without modifying game programs, configurations or data on the client side, a player can cheat by modifying the client infrastructure such as device drivers in his operating system. For example, he can modify a graphics driver to make a wall transparent so that he can see through the wall, locating other players who are supposed to be hidden behind the wall [7]. This is the so-called "wall hack", a popular cheat in some online games.

G: Cheating by Denying Service to Peer Players. A cheater can gain advantages by denying service to his peer players. For example, we have personally observed that a cheater could delay the responses from his opponent by flooding his network connection. Other peer players would then be cheated into believing that there was something wrong with the network connection of the victim, and agree to kick him out from the game in order to avoid the game session being stalled.

H:* Timing Cheating. In some real-time online games, a cheating player can delay his own move until he knows all the opponents' moves, and thus gain a huge advantage [2]. This *look-ahead cheat* is one kind of *timing cheating*.

Other timing cheating includes the *suppress-correct cheat*, which allows a cheater to gain an advantage by purposefully dropping update messages at the "right" time [2].

I: Cheating by Compromising Passwords. A password is often the key to much of or all the data and authorization that a player has in an online game system. By compromising a password, a cheater can have access to the data and authorization that the victim has in the game system. Some online game operators, e.g. [5], have provided detailed guidelines on password selection and protection to their users.

J: Cheating by Exploiting Lack of Secrecy. When communication packets are exchanged in plain text format, one can cheat by eavesdropping on packets and

inserting, deleting or modifying game events or commands transmitted over the network. This form of cheating can also cause passwords to be compromised, if user passwords are sent to the server in plain text, as in many text-based MUD games [3], where *telnet* was the main interface to the game server.

K: Cheating by Exploiting Lack of Authentication. If there is no proper mechanism for authenticating a game server to clients, a cheater can collect many ID-password pairs of legitimate players by setting up a bogus game server. Similarly, if there is not a proper mechanism authenticating a client, a cheater can also exploit this to gain advantages. For example, it is critical to re-authenticate a player before any password change is executed for him. Otherwise, when a player leaves his computer temporarily unattended and his game session unclosed – we have personally observed that in countries such as China and Korea, many people play online games in internet cafes – a cheater who can physically access the player's machine may stealthily change his password, and exploit the changed password afterwards.

L:* Cheating by Exploiting a Bug or Loophole. This form of cheating exploits a bug or loophole in game programs or the game design itself, without involving any modification of game code or data. Once discovered, such a bug/loophole will give knowledgeable players a major advantage. An early case of such cheating can be traced back to an incident, that occurred in Lucasfilm's *Habitat*, one of the first multi-user virtual environments. Due to an inadvertent pricing error, people in the game could sell virtual items to a pawn shop at a higher price than they paid to get them from a vending machine. By shuttling back and forth between the vending machine and the pawn shop, some players become millionaires overnight [16].

M:* Cheating by Compromising Game Servers. A cheater can tamper with game server programs or change their configurations once he has obtained access to the game host systems. Various such cheating cases can be found in [19].

N: Cheating Related to Internal Misuse. A game operator usually has the privileges of a system administrator. It is easy for an insider – an employee of the game operator – to abuse this privilege. For example, he can generate super characters by modifying the game database on the server side [8].

O: Cheating by Social Engineering. Often cheaters attempt to trick a player into believing something attractive or annoying has happened to him and that as a result his ID and password are needed. Blizzard has published guidelines on avoiding such scams on its popular Battle.net [4], indicating that this kind of cheating has been a real problem.

The major revisions we have introduced to the listing above compared to that given in [22] include the following:

- New category. Three new cheating forms are added, namely *cheating by exploiting machine intelligence, by modifying client infrastructure* and *timing cheating*.

- Replaced category. *Cheating by modifying game software or data* defined in the previous framework could accommodate a wide range of cheating of different characteristics. It is replaced with two mutually exclusive and more specific categories, *cheating by exploiting misplaced trust* and *cheating by compromising game servers*.

- Revised category. The category *cheating by exploiting a bug or design flaw* as defined in the previous framework is ambiguous and inaccurate, since many cheats, such as cheating by exploiting lack of secrecy or authentication, can be eventually accused as exploiting a design flaw. We revise it to *cheating by exploiting a bug or loophole*, and redefine the properties that distinguish it from other types of cheating. Namely this form of cheating exploits a bug or loophole in game programs, but involves no modification of game code or data. If a player has to modify the game program or data in order to exploit a bug or design loophole, his cheating behaviour will not fall into this category, but into *cheating by exploiting misplaced trust* or *cheating by compromising game servers*.

  In addition, the category *cheating by abusing procedure or policy* as previously defined is renamed to *cheating by abusing the game procedure*, since the term "policy" can have special meanings in the context of security and thus the previous name can be sometimes misleading.

3.2 Nature of Cheats: Atomic vs. Complex

The list given above attempts to be comprehensive but not necessarily disjoint. Therefore, a given cheat might fall into more than one category. It would be ideal to define a list of common cheating forms that is disjoint, but unfortunately this has proved to be a very challenging task.

Although each listed form can be an independent cheat, an actual case of cheating may be complex and involves multiple forms of cheating. For example, the Pogo cheat discussed in [23] involved two dishonest players who collusively abused a voting protocol to gain advantages. It is in fact a cheat due to collusion, which abuses the game procedure, and at the same time also exploits a loophole in the game system design.

Another example is the *hit-then-run* cheat that we have personally observed in online Go games. Go is a time-critical game played between two people. The Go server counts the time spent by each player in a game, and the player who runs out of time will automatically lose the game. Many online players choose to play 25 moves in 10 minutes or less, and it is usual for one to play 5 stones in the last 10 seconds. Therefore, a cheating player can easily defeat an opponent by timing him out with a well timed flooding attack. This is a form of cheating by denying service to peer players.

The above *timeout* cheat can be used together with cheating by abusing the game procedure. Some Internet Go services implemented a penalty rule to fight against the *escaping* cheat: players who disconnect themselves will lose their unfinished game unless they return to finish it within a limited period. A *hit-then-run* cheater can take advantage of this rule in the following way. He floods one opponent so that the game is recorded as disconnected by the opponent. Then he does not log on until the penalty period has passed.

Type	*Label*	*Cheating Form*
Of special relevance to online games	A	Cheating by Exploiting Misplaced Trust
	B	Cheating by Collusion
	C	Cheating by Abusing the Game Procedure
	D	Cheating Related to Virtual Assets
	E	Cheating by Exploiting Machine Intelligence
	F	Cheating by Modifying Client Infrastructure
	H	Timing Cheating
Generic	G	Cheating by Denying Service to Peer Players
	I	Cheating by Compromising Passwords
	J	Cheating by Exploiting Lack of Secrecy
	K	Cheating by Exploiting Lack of Authentication
	L	Cheating by Exploiting a Bug or Design Loophole
	M	Cheating by Compromising Game Servers
	N	Cheating Related to Internal Misuse
	O	Cheating by Social Engineering

Table 1: Common cheating forms in online games

The game cannot be finished in time, and the opponent will automatically lose points for it.

3.3 Generic vs. Specific Cheats

Table 1 classifies all the above fifteen cheating forms into two divisions. The "generic" division includes eight forms of common cheating in online games, which are also generic to all network applications but may be given different names such as "attacks" or "intrusions" in different contexts. The "of special relevance" division includes both cheating specific to online games, and cheating that may also occur under different names in other network applications but has some interesting features or implications in the context of online games.

In fact, some cheating forms are specific to game genres. For example, *cheating related to virtual assets* has been widely seen in multiplayer role-playing games, driving games (where players can upgrade and trade vehicles), among others. But it has of course not occurred in first-person shooting games and online board/card games (such as bridge and Go) that do not have virtual assets.

4. A TAXONOMY OF ONLINE CHEATING

In this section, we define a taxonomy for online game cheating. This is a three dimensional taxonomy, and online cheating is classified by the underlying vulnerability (what is exploited?), the cheating consequence (what type of failure can be caused?) and the cheating principal (who is cheating?). Our classification is intentionally reminiscent of the dependability taxonomy provided in [12, 1] and the conceptual model described in [20].

Table 2 shows the details of the taxonomy by vulnerability, possible failure and exploiter (i.e. cheating principal), respectively. Note that the same cheating form will appear at least once in each of these categories. Divisions and, where appropriate, subdivisions are provided within the categories; these and their motivations are described in detail below.

4.1 By Vulnerability

Some cheats in online games exploit design inadequacies in the systems, and some others do not. For example, *cheating by exploiting a bug or loophole* takes advantage of inadequacies in the game design, implementation or both. However, social engineering does not exploit any technical design inadequacies. We classify the vulnerabilities exploited by online cheating to two divisions: *system design inadequacy* which concerns a technical design flaw arising in the process of system development, and vulnerability of various *people* involved in operating or playing online games.

There are two subdivisions in system design inadequacy: *inadequacy in the game system* and *inadequacy in the underlying systems*. Online games are applications running on top of an underlying networking and operating system. A cheater can exploit a flaw in a game system, a flaw in its underlying networking or operating system, or both.

Cheating by exploiting misplaced trust, lack of secrecy or authentication, timing cheating and *cheating by exploiting a bug or design loophole* take advantage of technical inadequacies in the game system, and they belong to the first subdivision. *Cheating by collusion, by abusing the game procedure*, and *by exploiting machine intelligence* can all be ultimately a technical design failure in the game system: they arise due to "the inability to foresee all the situations of the system will be faced with during its operational life, or the refusal to consider some of them" [12] for reasons such as a concern for time-to-market. Therefore, they also belong to the first division.

Two common cheating forms, namely *cheating by modifying client infrastructure* and *cheating by compromising game servers*, belong to the second subdivision. Specifically, the first of these occurs on the game client side. However, rather than exploit the game application itself, it modifies the system infrastructure, e.g. a device driver that is part of the operating system. Similarly, a cheater compromising a game server usually breaks into the server by exploiting a flaw in the operating system or network protocols on the server side[2].

Cheating by denying service to peer players usually exploits some inherent weakness of the network layer. However, it can also be committed by exploiting a design in-

[2]A game server program may have flaws that can be remotely exploited by a cheater, but we have not yet seen such cases in real life.

Classification of the various types of cheating	Vulnerability (-ies)				Possible Failure(s)					Exploiter(s)			
	System Design Inadequacy	People			Fairness Violation	Masquerade	Integrity Violation	Service Denial	Theft of Information or Possessions	Independent		Cooperative	
	In Underlying Systems	In Game System	Player	Game Operator						Single Player	Game Operator	Multiple players	Operator and Player
A) Cheating by Exploiting Misplaced Trust		●					●		●	●			
B) Cheating by Collusion		●			●				●			●	
C) Cheating by Abusing the Game Procedure		●			●					●			
D) Cheating Related to Virtual Assets			●		●					●			
E) Cheating by Exploiting Machine Intelligence		●			●					●			
F) Cheating by Modifying Client Infrastructure	●						●			●			
G) Cheating by Denying Service to Peer Players	●	●						●		●			
H) Timing Cheating		●			●		●			●			
I) Cheating by Compromising Passwords			●						●	●			
J) Cheating by Exploiting Lack of Secrecy		●					●		●	●			
K) Cheating by Exploiting Lack of Authentication		●				●				●			
L) Cheating by Exploiting a Bug or Design Loophole		●			●					●			
M) Cheating by Compromising Game Servers	●						●			●			
N) Cheating Related to Internal Misuse				●			●				●		●
O) Cheating by Social Engineering			●						●	●			

Table 2: Classification of online game cheating

adequacy in the game system alone. For example, a cheat that occurred in the *Firestorm* game [19] exploited a buffer-overflow condition in the game program to disconnect all players. Another example is the so-called "spawn point camping" cheat that is popular in some real-time shooting games. A spawn point is a place in the game where a player creates his avatar at the beginning of his gameplay (i.e. spawning) and recreates the avatar immediately after its death (i.e. respawning). Waiting around the spawn point, cheaters could easily kill players as they spawn or respawn, and thus prevent them from being able to join into the action. This is indeed a denial of service that exploits a design inadequacy in the game system. A more cautious design could, for instance, protect the newly (re)spawned players from being killed by granting them an immunity to ammunition for a short while, so that they could move themselves to a safe place. (During this period, they would not be allowed to do any harm to other players either.) Therefore, this form of cheating is included in both subdivisions.

Other cheating forms, such as social engineering, password compromising, cheating related to virtual assets and cheating related to internal misuse, are only marginally related to any technical design inadequacy. Instead, the first three forms largely exploit vulnerabilities of innocent *players*, and the fourth form is involved with vulnerability of insiders employed by the *game operator*.

4.2 By Consequence

We largely base our classification of cheating consequences on the four traditional aspects of computer security: confidentiality (prevention of unauthorized disclosure of information), integrity (prevention of unauthorized modification of information), availability (prevention of unauthorized withholding of information) and authenticity (the ability to assure the identity of a remote user regardless of the user's host). A breach of confidentiality results in *theft of information or possessions*, a breach of integrity results in *improper modification of game characteristics*, i.e. *integrity violation*, a breach of availability results in *service denial* and a breach of authenticity results in a *masquerade*.

Cheating by collusion, compromising passwords or *social engineering* can result in the theft of information or possessions in a game. *Cheating by exploiting lack of authentication* results in a *masquerade*. *Cheating by denying service to peer players* involves selective service denials, but *cheating by compromising game servers, by modifying client infrastructure or related to internal misuse* usually involves improper modification of game characteristics, i.e. integrity failure.

When exploited only to eavesdrop communications, *cheating by exploiting lack of secrecy* results in the theft of information. But when a cheater also try to make use of the theft information by inserting, deleting or modifying game events or commands, this form of cheating will also cause integrity violation. Moreover, as shown in the previous section, *cheating by exploiting misplaced trust* can lead to the theft of information, improper modification of game characteristics, or both.

However, these traditional aspects of computer security are insufficient to cover all the consequences of online game cheating. For example, both the cheat exploiting the erroneous pricing bug in *Habitat* and the win-trading collusion in *StarCraft* violated none of the issues of confidentiality, availability, integrity or authenticity. And the list goes on.

We have therefore introduced *fairness* between peer players as an additional aspect for understanding online game cheating, and a breach of fairness results in a *fairness violation*. Either *cheating by abusing the game procedure, cheating by exploiting a bug or design loophole*, *cheating by exploiting machine intelligence*, or *cheating related to virtual assets* can result in a fairness violation. As evidenced by the win-trading case, *cheating by collusion* can also result in a fairness violation. Moreover, *timing cheating* is usually involved with improper modification of game characteristics, and it also directly leads to a fairness violation.

4.3 By Cheating Principal

A player can cheat single-handedly in either single-player or multi-player online games, whereas in multi-player games two or more players can collaborate to cheat. Furthermore, a player can also collude with an insider employed by the game operator to cheat. The identity of the cheating principal is used as the third dimension in our classifications, and it provides a way of distinguishing *cooperative* cheats from their *independent* counterparts.

The *cooperative* division includes two subdivisions: the category of *multiple players* cover cheats, such as *cheating by collusion*, that have to be committed by two or more players cooperatively, and the category of *operator and player* accommodates cheating committed through the cooperation of at least a player and an insider, which typically involves collusion as well as internal misuse that are specific to the game.

The *independent* division also includes two subdivision. The category of *game operator* accommodates cheating related to internal misuse, where no collusion between player and insider is involved, however. One example is that of an insider who is also a player. As discussed in [23], house cheating orchestrated by a game operator alone is likely to occur. However, it is beyond the definition of online cheating used in this paper. The category of *single player* accommodates all cheating forms that can be committed by a player single-handedly.

5. DISCUSSION

Our taxonomy brings out a systematic view of online cheating, from which a number of observations can be made.

First, online cheaters have exploited both vulnerabilities in the computer systems and those of people that are involved in the games. Both design inadequacies in game systems and weaknesses in their underlying system infrastructure can be exploited to cheat. So can vulnerabilities of both innocent players and corrupt insiders.

Second, the classification by cheating principal shows that most current game cheating can be committed by one player independently, but that some other cases involve collusion between one and his peer player(s) or an insider. Security breaches caused by a single user or through cooperation of a user and an insider have been commonly seen in many contexts. However, it appears that collusion between peer users in other contexts is not such a serious problem as in online games, which seem to provide more opportunities or incentives for people to collude.

Third, as shown by the classification by consequences, cheating is in fact largely due to various security failures. It makes this observation more clear and specific to examine the distribution of each common cheating form in

	Info Theft	Service Denial	Integrity Failure	Masquerade	Fairness Violation
Design inadequacy in the game system	A, B, J	G	A	K	C, E, H, L
Design inadequacy in the underlying systems		G	F, M		
Vulnerability in player	I, O				D
Vulnerability in insider			N		

Table 3: Distribution of observed cheating forms in the vulnerability-consequence matrix

the two orthogonal dimensions of vulnerabilities and consequences. Table 3 constructs such a distribution matrix, where the vulnerability and consequence are displayed in rows and columns respectively, and cheating forms in the cells are represented with their labels assigned in Section 3. The matrix clearly shows that most types of online game cheats have been about information theft, improper modification of game characteristics, or fairness violation, and they largely exploit flaws in the game systems.

However, the distribution of cheating forms in the vulnerability-consequence matrix may not remain stationary as online games and the cheating phenomenon co-evolve. Therefore, any observation based exclusively on this matrix may have to remain tentative. For example, it is not yet clear whether cheats exploiting flaws in the underlying networking and operating systems will fast increase in the future.

It is interesting to note that as a result of taxonomic analysis using Table 3, we have corrected a mistake in a previous version of this paper. Namely, we found that we carelessly missed a type of cheating by denying service to peer players, which involves exploitation of design inadequacies in the game system only.

It appears that we can also use Table 3 to suggest novel additional forms of cheating that will likely occur in the future. For example, it is very likely that cheats which lead to masquerade, information theft or fairness violation and which are due to design inadequacies in the underlying systems, will occur in the future, although it is not yet clear in which forms they will manifest themselves.

Furthermore, it is worthwhile asking what implications online cheating has for security in such a representative Internet application as online games, since it is well known that security can mean different things in a different context. Re-examining the classification by possible failures in Table 2 tells us that no matter whether a cheating form results in either information theft, service denial, improper modification of game characteristics, or masquerade, a fairness violation in fact will be caused, gaining a cheater some advantages over his peer players in the game. Thus, fairness and its enforcement appear to be a proper perspective for understanding the role of security in applications like online games. This echoes the result of [23] and can be easily explained as follows. On the one hand, fair play is essential to any game. Online gaming is not an exception, and fairness should be an inherent concern in its design. On the other hand, online players usually do not know each other, and they are often scattered across different physical locations. Therefore, the social structures preventing or discouraging cheating in the non-electronic world are no longer in place for online games. It is security that can become an alternative mechanism for fairness enforcement.

Nonetheless, security techniques alone cannot provide a complete solution to cheating prevention or mitigation in online games. For example, as shown in our previous work [23], security mechanisms that usually can mitigate collusion in one way or another do not work well in online bridge, in which cheaters can collude via the telephone, instant messenger or other means. Instead, a collusion detection approach based on artificial intelligence techniques appears to be essential in mitigating this devastating threat. Therefore, security plays an important but non-exclusive role in enforcing the fair play in online games.

6. CONCLUSIONS

Online computer games open themselves to a wide spectrum of cheating. We have presented a classification scheme for cheating in online games, in which the classification is made with respect to the underlying vulnerability, consequences and the cheating principals.

We have found that cheating in online games is largely due to various security failures. However, the four traditional aspects of security – confidentiality, integrity, availability and authenticity – are insufficient to explain cheating and its consequences. But fairness becomes a vital additional aspect, and its enforcement a perspective for understanding the role of security in developing and operating online games.

7. ACKNOWLEDGEMENT

We are grateful to Ross Anderson who has read an earlier version of this paper and made valuable comments and suggestions. Comments from an anonymous reviewer also improved this paper.

8. REFERENCES

[1] A Avizienis, JC Laprie, B Randell and C Landwehr, "Basic Concepts and Taxonomy of Dependable and Secure Computing", IEEE Transactions on Dependable and Secure Computing, Vol. 1, No. 1, 2004, pp 11-33.

[2] N Baughman and B Levine. "Cheat-proof Playout for Centralized and Distributed Online Games", in *Proc. of the Twentieth IEEE INFOCOM Conference*, Apr. 2001.

[3] R Bartle, "Interactive Multi-User Computer Games", Report commissioned by British Telecom Research Laboratories, December, 1990. Available at `http://www.mud.co.uk/richard/imucg.htm`.

[4] Blizzard, "Battle.net Scams", available at http://www.blizzard.com/support/?id=asi0461p.

[5] Blizzard, "Creating a Secure Password", available at http://www.blizzard.com/support/?id=asi0505p.

[6] Blizzard, Starcraft official site, http://www.blizzard.com/starcraft/.

[7] C Choo, "Understanding Cheating in Counterstrike", Nov. 2001. Available at http://www.fragnetics.com/articles/cscheat/print.html.

[8] Chosun Ilbo, "Rayegard system developer is punished", June 27, 2001 (in Korean). Also available at http://www.chosun.com/w21data/html/news/200106/200106270431.html.

[9] SB Davis, "Why Cheating Matters: Cheating, Game Security, and the Future of Global On-line Gaming Business", in *Proc. of Game Developer Conference 2001*, 2001.

[10] Hankyoreh, "Online cheating is ubiquitous", May 9, 2001 (in Korean). Also available at http://www.hani.co.kr/section-005100025/2001/05/005100025200105091907004.html.

[11] CE Landwehr, AR Bull, JP McDermott and WS Choi, "A taxonomy of computer program security flaws", ACM Computing Surveys, Vol.26 No.3, Sept. 1994. pp211-254.

[12] JC Laprie (ed.), Dependability: Basic Concepts and Terminology, Springer-Verlag, Vienna, 1992.

[13] K Li, S Ding and D McCreary, "Analysis of State Exposure Control to Prevent Cheating in Online Games", The 14th ACM International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV), 2004.

[14] U Lindqvist and E Jonsson, "How to Systematically Classify Computer Security Intrusions", in *Proceedings of the 1997 IEEE Symposium on Security & Privacy*, Oakland, California, May 4-7, 1997. IEEE Computer Society Press. pp154-163.

[15] S McCreary and K Claffy, "Trends in Wide Area IP Traffic Patterns: A View from Ames Internet Exchange", in *Proceedings of the ITC Specialist Seminar on IP Traffic Modeling, Measurement and Management*, Monterey, CA, USA, Sept. 2000.

[16] C Morningstar and FR Farmer, "The Lessons of Lucasfilm's Habitat", in *Cyberspace: First Steps*, M Benedikt (ed.), MIT Press, Cambridge, 1990.

[17] M Müller, "Computer Go", Artificial Intelligence, Vol.134, No.1-2 (Special issue on Games, Computers and AI), January 2002, pp145-179.

[18] PG Neumann and DB Parker, "A Summary of Computer Misuse Techniques", in *Proc. of the 12th National Computer Security Conference*, Baltimore, MD, 1989, pp. 396–407.

[19] M Pritchard, "How to Hurt the Hackers: The Scoop on Internet Cheating and How You Can Combat It", Information Security Bulletin, February 2001.

[20] D Powell and RJ Stroud (Editors), "Conceptual Model and Architecture of MAFTIA", MAFTA Project Deliverable D21, February 2003 (available from http://www.maftia.org).

[21] E Raymond, "The Case of the Quake Cheats", unpublished manuscript, 1999. Available at http://www.catb.org/~esr/writings/quake-cheats.html.

[22] J Yan and HJ Choi, "Security Issues in Online Games", *The Electronic Library*, Vol. 20, No.2, 2002. A previous version appears in *Proc. of International Conference on Application and Development of Computer Games*, City University of Hong Kong, Nov. 2001.

[23] J Yan, "Security Design in Online Games", in *Proc. of the 19th Annual Computer Security Applications Conference*, IEEE Computer Society, December, 2003.