Taxonomy on Online Game Security

Junbaek Ki
Inae Steal Cooperation, kijun21@hotmail.com
Jung Hee Cheon
Department of Mathematics, Seoul National University (SNU), jhcheon@math.snu.ac.kr
Jeong-Uk Kang
Korea Advanced Institute of Science and Technology (KAIST), ux@calab.kaist.ac.kr
Dogyun Kim
Cemtlo Media, dgkim100@shinbiro.com

Abstract

We present classification of known attacks in online games and provide security solutions against them. While previous works just presented attacks and solutions in case by case, we newly classify attacks by objectives and methods. Moreover, we present attacks in each of four layers: client, server, network, and environment. Through this systematic classification, solutions can be provided more efficiently against even unknown future attacks.
Keywords: Online Games, Security, Network Security, Hacking, Cryptography, Digital Rights Management (DRM)

1. Introduction

Online games are being rapidly developed with the advancement of the Internet. However, with the growth of online games, the presence of malicious users is becoming more evident. They use illegal programs to hide and manipulate the powers of their characters and abuse the game regulations. Also, in the case of item exchange, users illegally copy online game items, and exchange or sell them. Nevertheless, not enough security has been developed with respect to online games. The reason is that game developers have not seriously considered the malicious attacks. Since previous researches have begun to devote themselves in classifying these attacks case by case, they couldn't provide appropriate and organized security solutions to the known attacks. In this paper, we analyze known attacks and newly classify them by their "objectives" and "methods". We classify known attacks into ID/password theft, hidden information exposure, data modification, and advantage from cheating by their objectives. We also categorize attacks into four layers such as client, server, network, and environment. Finally, we review the security tools and provide possible security solutions against known attacks according to the classification.

The remainder of this paper is organized as follows: Section 2 provides classification of online games into three types by the structure of the system, Section 3 describes the classification of attacks by objectives and methods, Section 4

presents solutions, and Section 5 describes security solutions according to classification. Brief concluding remarks appear in Section 6.

2. Classification of Online Games

Online games are classified into three types by the structure of the system [12]. We briefly review them and discuss their security aspects.

2.1 Distributed Client Model

There are precisely two types of games in the distributed client model. The first would be peer-to-peer games such as *Doom* and *Duke Nukem* in which each player's machine run parallel with the game engine, and the machines precede in lockstep. The second would be peers-to-server games such as *Diablo I* and *Starcraft Battle.net* where the server provides only the connections and environment to peers. Most of the process in this case is operated at the client layer. The advantages of this model are that it can reduce server load and will not always need a guarantee of connection between client and server [12].

Most security problems in this model come from client’s sides. Because each personal computer stores important data and perform almost all operations in it, malicious clients may easily modify data and software to get illegal advantage. If these security problems were settled, this model would be the ideal one in the near future.

2.2 Centralized Server Model

In the centralized server model, a client device has only the ability of input and output (I/O). It is basically nothing more than a specialized terminal. Each client gathers up keyboard, mouse, and joystick input for each frame and transmits it to the server. On the other side, the server works all operations. The server receives the input from all clients, runs the game for a fixed time slice, and sends the results to the clients. Last the clients display the results during the next frame after they are received [1]. This model includes *Ultra Online*, *Diablo I* and *Ever Quest*, which are familiar to many game players.

Since all the operations are performed at the server in this model, the server may be overload. It becomes a serious problem in this model because of many delays occurring in games being played. In addition, this model also has security problems including server environment weaknesses, security holes in operating systems.

2.3 Client/Server Model

In the client/server model, the client has an ability of operation like *Mu*. The client and server have separated processes running on different machines, so this model can simplify the synchronization problem between various players.

This model can overcome a little latency problem between player's actions. Latency can make the game difficult to play.

This problem raises the possibility of running some game logic on the client in parallel with server or other clients at a peer-to-peer architecture [1]. Also, this model can provide a solution to some server overload problems. However, since all security problems in the previous two models can occur in this model, we need more it requires more secure design.

3. Classification of Attacks

Since previous works have been devoted to classifying attacks of cracker by cases, they could not provide systematic classification of security issues on online games. In this chapter, we classify attacks by objectives and methods of attacks.

3.1 Classification of Attacks by Objectives

User ID and Password Theft. User IDs and passwords are used to confirm, ensure his/her account and check whether he/she has an authority to access a system. Therefore, user IDs and passwords have been always the first target of attackers.

Hidden Information Exposure. Attackers prefer to play games on the advanced position, so they use hacking programs such as a *map hack* to find the location of hidden items or enemies and look over the map which cannot be shown before the players get an authority. For example, realm hack for *Diablo II v1.09d* that reveals auto map of entire act, including all quest features, position of many shrines and their type. Also hostile players and special monsters, runes and charms are all in different colors along with many settings configurable [10].

Data Modification. Attackers may modify data or software in a system in order to generate and duplicate items and modify the number of victories. For example, *trade hack* can steal items being exchanged on game, but after a game off, when player logs on the game again, the items disappear from sight.

Advantage from Cheating. Attackers may disturb the opponent's process through speed hack, collusion, and denial of service (DoS). Attackers may break down game balance and damage the opponent with a *speed hack* which can raise the running speed of their system. Two players may collude to kill an opponent’s character and take his items. DoS attack may cause an opponent's network connection flood, so the opponent cannot normally receive and send packets or access an online game server.

3.2 Classification of Attacks by Methods

A structure of online game can be divided into four layers: client, server, network, and environment. We present attacks in each layer.

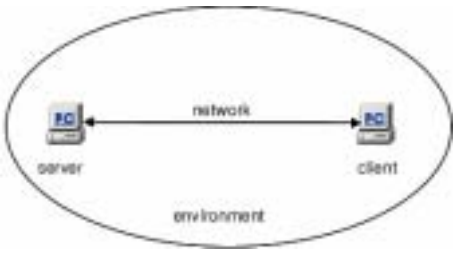

Figure 1. Four layers

1) Client Layer

Memory Scan and Modification. The memory scanning tools such as a *game buster* helps attackers look for critical variables when online games are running in the memory [13]. The attackers may modify the variables in the memory, not the game files.

Software modification. The players install game software to play online games and connect with the online game servers. However, Attackers abuse the software. They may remove validating routines, modify configuration parameters or rewrite some parts of game software [13].

Bug or Flaw in Software. Online game software becomes complicated. Therefore, bug or flaw in software is continuously found out. A typical attack may exploit a bug in the overflow logic.

Malicious Code. Malicious code typically includes a virus, worm, and Trojan horse. A virus can break the computer system and delay or interrupt system works. A worm resides in active memory and duplicates itself. It is commonly noticed only when their uncontrolled replication consumes system resources, slowing or halting other tasks. A Trojan horse makes several modifications to the system and allows the attackers to send commands to the system. The attackers can upload and download files, delete and create files to the system.

Backdoor. Once backdoor is installed in a system, an intruder can gain remote access to a computer system without process of authentication and any traces such as log. It allows attackers to access and control the computer system.

2) Server Layer

Server Environment Weakness. The online game server is opened over the insecure network so that it has been a target of the attackers. However, only firewall has been used to protect it against a thousand of illegal intrusion. Firewall is not smart enough as would be desirable.

ID and Password Dictionary Attack. A good password for users such as the birth date, phone number or favorite sports

is what users can easily remember. However, these passwords are very easy to guess. A secure password should be very difficult to guess and therefore may be more difficult to remember. Attackers have special cracker programs that will automatically try every word in the dictionary as well as common names. These programs can crack a single dictionary word password [5].

Security Holes of OS. Operating system comes with installation programs or installation scripts to be convenient for the user to install OS. However, it creates many of the most dangerous security holes [11]. Moreover, default installations nearly always include extraneous services and corresponding open ports. Attackers may break into systems via these ports.

Internal Misuse. Since an operator has the authority to approach the internal system and to rewrite or delete data, he/she may misuse of it. For example, the developer of the online game, *Rain Guard* once manipulated the database and created an undefeatable character.

3) Network Layer

Packet Sniffing. Packet sniffer is a program or device that monitors data traveling over a network. Attackers may use the sniffer to eavesdrop on valuable information on a network. Many online game services transfer their password in plain text so that it is easily captured by the sniffer. After the attackers retrieve them, they can intrude a system and damage it.

Packet Inserting, Deleting, and Modifying. Packets including command and information are transferred via the open network so that the attackers can insert, delete or modify the packets to illegally get an advantage. For online game *Fortress*, attackers may insert a packet including the location of target. Finally, the attackers' bullets always hit it.

Pretending Game Web Page. Attackers can create a game Web page pretending to be a real one. However, users don’t doubt it, because they don’t have a process to authenticate the Web page. It would let attackers either read files from the user local drive or read information such as ID and password from pages visited by the user [3].

Denial of Service (DoS). DoS attacks can disable a user from accessing the online game server. These attacks exhaust the limited resource in the system, destruct the configuration information, or break down the physical components on the network [7] so that users can’t receive any services.

4) Environment

Hack. Attackers want to play games on an advanced position, so they use the hacking programs such as *speed hack* and *map hack*. A *speed hack* enables attackers to run at incredible speeds in realm games. A *map hack* can find the location of hidden items or enemies and look over the map which cannot be shown before the players receive an authority.

Abusing Procedure or Policy. Collusion is a kind of attack that abuses the legitimate process and policy. Two attackers are on the same part. They attack one opponent. Although this attack does not use any malicious programs or intrude a system illegally, it damages players.

Social Engineering. Social engineering attack is used for ID and password impersonation. Attackers attempt users to disclose the ID and password by telephone, email or chatting. For instance, users receive false messages such as “Could you please send your ID and password to draw lots? We will send some presents to you!” or “There are some problems in your account, so you should send your ID and password.” However it's not true. This is just for ID and password impersonation.

4. Solutions

Through this chapter, we will review the security tools for online games.

4.1 Security Tools

Encryption. Encryption provides confidentiality, which is a service used to keep information secret from all but those who are authorized to see it. There are two types of encryption by using keys: symmetric and asymmetric key encryption. Symmetric key encryption is relatively fast, but the key has to be agreed in advance. Asymmetric key encryption is used to establish secure communications with any individual, but it is slow and computationally intensive. For asymmetric key encryption, a computer generates a pair of keys, a public key and a private key. When people wish to send a secure transmission to someone, use the public key to encrypt. When the receiver obtains the message, he decrypts it using his private key that he keeps secret.

Message Authentication Codes (MAC). MAC provides integrity of information. Typically, message authentication codes are used between two parties that share a secret key in order to validate information transmitted between these parties. As design method, MAC is divided into dedicated MAC and HMAC based on cryptographic hash functions such as MD5 and SHA-1 [8].

Digital Signature. A digital signature is used to authenticate the origin of a message and to check its integrity. Digital signature schemes are based on public-key cryptography that uses different two keys such as public and private keys. The signer makes a signature with the private key, and the receiver verifies the signature with the signer’s public key. As well as integrity and authentication, digital signatures provide non-repudiation of the origin of the signature, that is, the verifier can prove to third party that the signature was made by the origin signer.

Key Management. For the most system, ID and password are used for authentication as the first. If the ID and password are exposed to attackers, they can get an authority to access the system. Therefore, for solving this problem, users should

change a password regularly, not reuse an old password and not use passwords easily guessed. Moreover, there are security tools to manage keys. One-time password system generates a sequence of single use passwords, so it is secure against passive attacks based on replay attacks [6]. A smart card can protect the intrusion into the devices so that no one can tamper keys.

Antivirus Program. Virus protection software works by scanning the lines of code in each file and comparing them to the code from known viruses. The faster method to check virus is to find the common characters in a virus. It is a favorable method to estimate infection of virus. Other methods are to examine the boot sector and the amount of memory being used and variation of file size [5]. However, new viruses come on every day, so antivirus program can't protect all new viruses. Finally users should install the latest upgraded antivirus program.

Tamper Resistant Module (TRM). Tamper resistant module can be used to prevent the software cracking, verify the integrity of it, and hide the private key in the software or memory to insure the security. Especially, TRM is necessary for preventing the duplicating the digital contents, the software modification and the removal of functions. However, it’s not easy to figure out them in software, so reverse engineering protection tools such as code obfuscator can be used, while hardware TRM can easily implement tamper resistant.

Firewall. A firewall system is a fundamental component for the defense from unauthorized access. Firewall is usually not a single system. It can be a router, a personal computer, a host, or a collection of hosts. Firewall is usually placed between two or more networks. The main purpose of firewall is to control access to a protected system. They have a function to defense against external attacks to the computer systems, networks, and critical information. Firewall can also be used to reduce the internal misuse.

Intrusion Detection System (IDS). Intrusion detection system inspects all internal and external network activity and identifies suspicious patterns that intruder may destroy a system or legitimate user may misuse system resources. Moreover, new IDSs can identify attacks in progress, generate real-time alerts, and even launch countermeasures or reconfigure routers or firewalls to counter an attack [4].

Secure OS. Secure OS is the operating system that graft secure kernel on the existing OS to protect a variety hacking available. Secure OS provides identification and authentication (IA), discretionary access control and mandatory access control. Also, it supplies a diverse security functions such as authentication between file and process, audit, log, record and network filtering [9].

4.2 Security Solutions

We provide possible security solutions against known attacks according to the classification.

1) Client Layer

Data Scan. The memory scanning tools such as a *game buster* are to help attackers look for the critical variables in the memory. The attackers may modify the variables in the memory. To prevent the data scanning, we propose encryption. Encryption can encrypt critical values in the memory so that the attackers can't recognize the location of the elements and amend them.

Software Modification. The attackers may modify their own software to get the unfair advantage. They can remove validating routines, modify configuration parameters or rewrite some parts of game software. To prevent the software modification, we propose TRM and obfuscator. TRM can verify the integrity of software whether the modification occurs in it. An obfuscator can protect decompiling Java source that removes the unnecessary or unused methods and inserts the false information.

Bug and Flaw in Software. As online games come to be complicated, the number of system bugs increase. The attackers misuse these game bugs. To prevent exploiting bug and flaw in software, there is only bug patching. Because the game developers can't thoroughly find the bugs in the system, these problems continuously happen. They give a question to the software engineering.

Malicious Codes, Backdoor. Malicious codes and backdoors may be established in a system by email, unauthenticated file download, or open ports. Malicious codes can harm the system and backdoors can provide an access to unauthorized users. To prevent the malicious codes and backdoor, we propose a digital signature, firewall, IDS and antivirus. A digital signature can be authenticated as the legitimate users so that they can rely on a received data. A firewall can control a system from external or internal access, IDS can detect and respond the intrusion from unauthorized users and antivirus can detect and delete malicious codes in a system.

2) Server Layer

Server Environment Weakness. Since a game server is accessed through the Internet, it becomes a target of the malicious attackers. To prevent using the server environment weakness, we propose firewall and IDS. A firewall can perform packet filtering against the DoS attack and protect against inside or outside attacks by the access control. The IDS can identify attacks in progress, generate real-time alerts, and even launch countermeasures or reconfigure routers or firewalls to counter an attack.

ID and Password Dictionary Attack. Attackers may use the ID and password dictionary attack to impersonate ID and password. To prevent the ID and password dictionary attack, we propose key management. Tamper resistant methods such as smart cards can make it impossible to write or read system inside information. Moreover, one time password system can protect the reuse of exposed ID and password against the replay attack. In addition, in order to manage ID

and password, they employ a security policy; the password has expiring periods, the used password can't be reused on the fixed duration, system restricts the input of the wrong password and it does not allow using related user's information such as birthday and telephone numbers.

Security Hole of OS. Malicious users abuse the security holes exposed in the operating system. To prevent exploiting the security hole of OS, we propose secure OS. It provides identification and authentication, discretionary access control, and mandatory access control by using the secure kernel. Moreover, to protect game server from attackers, turn off unneeded services and close extraneous ports, and apply standard installation guidelines for all operating systems. These guidelines include installation of only the minimal features needed for the system to function effectively [11].

Internal Misuse. Insider operators may misuse of the authority to access the inner system, and delete and modify the data. To prevent the internal misuse, we propose audit and log. An audit analyzes fact related in security and records it in the traces of audit. Log supplies enough information to support accountability and traceability for all privilege system commands and includes a record of user-initiated, security-relevant activities [2].

3) Network Layer

Packet Sniffing. Attackers eavesdrop on the user ID and password or sensitive information on the network by packet sniffing. After that, they may intrude and harm the system. To prevent the packet sniffing, we propose encryption and key management like one-time passwords. First, the server and client send the encrypted packets on the interaction between them, so no one can sniff any pockets and cannot understand them unless they are authorized. For key management, one-time password system can protect from replay attack when the password is exposed.

Packet Inserting, Deleting, and Modifying. Data is transmitted in plain text, not ciphertext, over the TCP/IP protocol, so it is not secure. The data always has a risk of attackers inserting, deleting, and modifying packets. To prevent them, we propose a digital signature and MAC. The digital signature is used to authenticate the origin of a message and to check the integrity of the information by a public key cryptography. Moreover, MAC is used to validate information transmitted between two parties that share a secret key. Therefore, MAC can authenticate both the source of a message and its integrity.

Pretending Game Web Page. Attackers can get user IDs and passwords or intrude the user's local drive to read files and information by using false game Web page. To prevent pretending game Web page, we propose a digital signature. If users could verify the server authentication, they would assure of the Web page. For instance, certificate authority (CA) issues a digital certification so that users can strongly have proof of server authority and verify the integrity of data.

Denial of Service (DoS). Attackers may interrupt users' access to online games with DoS attacks, which destroy the specific system service and get the network overflowed to block their own tasks so that the service doesn't work and user

can't receive any accurate service. To prevent the DoS attack, we propose firewall and IDS. First, firewall can perform packet filtering against the DoS attack. Most DoS attack uses informal IP address or inner IP address not used in a group. In addition, the IDS can detect and cope with the abnormal intrusion. There are statistical control methods and extractable methods of specific intrusion pattern in the IDS. However, distributed denial of service (DDoS) attacks have recently emerged as one of the most serious attacks. There is no current prospect of preventing DDoS attacks in the near future.

4) Environment

Hack. Attackers may play games on an advanced position by using hacking programs. To prevent the hack, we propose TRM. A TRM has a function to recognize and protect execution of the malicious program such as hack in game software. Once the software runs, TRM checks integrity itself whether software is modified by hacking program or not. If there are some problems, TRM makes an online game stopped.

Abusing Procedure or Policy, Social Engineering. Abusing the legal procedure or policy and social engineering are different from using the malicious program or illegally attacking a system, but they are more intelligent attacks. To prevent the abusing procedure or policy and social engineering, we propose continuous education and strict management of ID and password. A technical means can't figure out all security problems in online games. Game service provider should establish a security policy and educate users to keep in mind.

5. Security Solutions by Classification

We generally classify computer game into three types such as PC game, console game and arcade game. Also, each one can be divided into three types such as distributed client model, centralized server model and client/server model by the structure of the system. We provide known attacks in each section and security solutions for each one. We summarize them in The Table I. The rows are classified by game configuration and the columns are classified by system structure in online games.

Table I. Security solutions by classification.

	Distributed Client Model	Centralized Server Model	Client/Server Model
Arcade Environment	Client Layer		
Console Environment	Client Layer (Passive attack) Network Layer	Server Layer Network Layer	Client Layer (Passive attack) Server Layer Network Layer
PC Environment	Client Layer Network Layer	Server Layer Network Layer	Client Layer Server Layer Network Layer

6. Conclusion

Online games are rapidly developing along with the Internet's development. These developments make a new culture through online games and allow people to have experiences in the virtual society that could not be possible in the real world. However, online games have given a new terrain for crimes to appear and stimulate hackers to try their newly acquired knowledge and techniques. Through this paper, we reviewed the security problems occurring on online games and gave a new classification of online games based on attacks, objectives and methods. Moreover, a review of the security tools was given to allow an opportunity to search for a security solution.

However, even at this very moment, malicious users are becoming smarter and using high technology that threatens online games. In order to provide better services to users, there is a need for game developers and security experts to continuously involve themselves in research and development concerning this matter. Moreover, what must not be overlooked is that complex security tools can cause an "inconvenience" to the users whishing to enjoy the game. The reason is that these security tools might demand more procedures in order to log in to the program or an additional program installation. Also, the use of encryption in order to prevent data modification would also cause a "latency" problem in the network. Therefore, these problems could cause a drop in online game "efficiency". But if security issues were to be disregarded on online games, the quality of service would be at risk or even more, be impossible to provide.

References

[1] Abrash, M. (2000), *Quake's 3-D Engine: The Big Picture*, GameDev.net. Avalable http://www.gamedev.net/reference/articles/article987.asp

[2] Barman, S. (2002), *Writing Information Security Policies*, New Riders Publishing

[3] Broersma, M. (2002), *MS warns of 'critical' flaws*. Available http://zdnet.com.com/2100-1105-844318.html

[4] Canavan, J.E. (2001), *Fundamentals of Network Security*, Artech House, Inc.

[5] Hagen, R.D. (2001), *A User's Guide to Security Threats on the Desktop.* Available http://rr.sans.org/securitybasics/user-guide.php/

[6] Haller, N., Metz, C. (1996), "A One-Time Password System," Obsoleted by RFC2289, RFC pages, IETF. Available http://www.ietf.org/rfc/rfc2289.txt?number=2289

[7] Householder, A., Manion, A., Pesante, L., and Weaver, M. (1991), *Managing the Threat of Denial-of-Service Attacks*, CERT Coordination Center. Available http://www.cert.org/archive/pdf/Managing_DoS.pdf

[8] Krawczyk, H., Bellare, and M., Canetti, R. (1997), "HMAC: Keyed-Hashing for Message Authentication Network Working Group", Request for Comments, the Internet Engineering Task Force (IETF). Available http://www.ietf.org/rfc/rfc2104.txt

[9] Korea Information Security Agency (KISA) (2001), *Information Security Technology, System Protection Technology, Secure Operating System Development Technology*. Available http://www.kisa.or.kr/

[10] Mroms.com (2001), *4 Dimensional Web Design, Maphack v4.3d*, Available http://4dwd.com/mroms/diablo2.htm

[11] SANS Institute resources (2002), *The Twenty Most Critical Internet Security Vulnerabilities (Updated), The Experts' Consensus*,

version 2.504. Available http://www.sans.org/top20.htm/

[12] Sweeney, T. (1999), *Unreal Networking Architecture*, Epic MegaGames, Inc. Available http://unreal.epicgames.com/Network.htm

[13] Yan, J.J. and Choi, H.J. (2002), “Security Issues in Online Games,” The Electronic Library, EmeraldMCB University Press, UK.

Junbaek Ki took an electronic commerce course at Information and Communications University (ICU) in Daejeon, Korea from 2001 to 2003. During these times, he worked for an industrial-educational cooperation project on "Development of Secure Electronic Trading System for Online Game Items" in Applied Cryptography Lab. He took his master's degree in computer science from ICU. His title of master thesis is "Online Game Security and Secure Electronic Item Trading Protocol". Now he is working for Inae Steel co. in Seoul, Korea.

Jung Hee Cheon received his B.S., M.S., and Ph.D. degrees in mathematics from Korea Advanced Institute of Science and Technology (KAIST) in 1991, 1993, and 1997, respectively. For three years from 1997, he worked for Electronics and Telecommunications Research Institute (ETRI). In 2000 he held postdoc position in Brown university working with J. Silverman. After working for Information and Communications University (ICU), he joined to Seoul National University as an assistant professor. His research interests include computational mathematics, cryptography, and applied cryptography.

Jeong-Uk Kang received his B.S., and M.S. degrees in Computer Science Division, Dept. of EECS from Korea Advanced Institute of Science and Technology (KAIST) in 1998, and 2000, respectively. Currently, he is enrolled in the PhD program in Computer Science Division, KAIST. His research interests include Operating System, and applied cryptography.

Dogyun Kim received his B.S. and M.S. degrees in mathematics from Korea Advanced Institute of Science and Technology (KAIST) in 1990 and 1992 respectively. For 11 years from 1992, he worked for Samsung Electronics co., Ltd (SEC), Samsung SDS co., Ltd, Cemtlomedia Corporation and SNLAB Corporation.