The Security Research of Massively Multiplayer Online Role Playing Games

Yameng BAN
Department of Software Engineering
Shijiazhuang Information Engineering Vocational College
Shijiazhuang, China
e-mail: banyameng@163.com

Yuepeng ZHAO
Department of Software Engineering
Shijiazhuang Information Engineering Vocational College
Shijiazhuang, China
e-mail: zypplayks@yahoo.com.cn

***Abstract*—The security problem is independent from online game itself, but it is the basic guarantee to the survival and the development of online games. This article discusses the server architecture security, user input security, data transmission security, login verified security, game logic security and transaction security on MMORPG in detail, and provides a feasible solution to the design and realization of some key technology.**

Keywords- MMORPG; game security; server architecture

I. INTRODUCTION

The software of Massively Multiplayer Online Role Playing Games (MMORPG) is consisted of the client side and the server side. Players play the role of virtual word and the server is responsible for the management and maintenance. The server is provided by the game operator who stores the data of players and offers services for them. The whole process of game is the interactivity between players who play different roles in the virtual cyberspace.

This article analyzes designs and realizes several security technology of MMORPG in detail, such as the server architecture, data transmission security, logging on validation security, game logic security and transaction security, etc.

II. SERVER ARCHITECTURE SECURITY

The strategy of server architecture security is MMORPG network topology security based on C/S structure.

This article analyzes the foundation of the server architecture security strategy in terms of MMORPG server and client network topology structure and builds the network topology in the thoughts of layered defense.

The game client side is located in the public network in layered structure. The public network is trustless area, so the client connects to the server through Internet and the external firewall is deployed between them. The server which is in the so-called Demilitarized Zone (DMZ) communicates with database through the internal firewall.

Firewalls are deployed in every side of DMZ. The external firewall refuses some data streams enter into DMZ, such as FTP, SNMP, Telnet while the internal firewall refuses SSL and SSH.

To ensure the security of server, the types and amounts of the server in DMZ should be reduced as far as possible. In the network topology structure adopted in this article, game client side separates from logic server (including world server and ground server) and database server (logging on database and game database). The data request from trustless client side in public network is verified by the server which is in DMZ.

The network topology structure of MMORPG is as shown in Fig. 1.

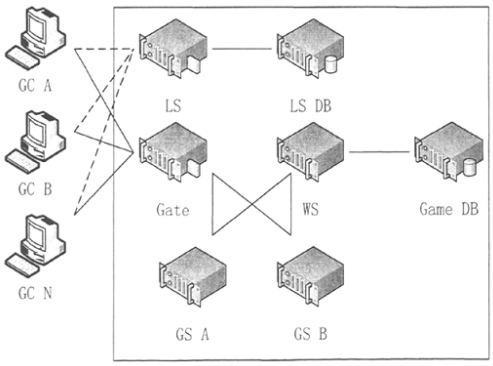

Fig. 1. The Network Topology Structure of MMORPG

The network topology structure of MMORPG is as shown in Fig. 1.

The game client (GC) sends the account information to logging on server (LS) for verification. The verified database linking with LS (LSDB) is in LAN which is transparent to GC.

After the verification, the connection between GC and top gate server (Gate) is created. GC interact the game data with ground server (GS) via Gate. Similar to LSDB, GS, WS and Game DB are invisible to GC, which are located in the area of Gate.

In the development of applications, in order to support more connections of GC, there is many-many mapping relationship between Gate and GS. When one of the Gates is attacked or closed, the others can run normally and the GCs connected with those can be played as usual. Meanwhile, Gate can share part of the work of message verification and security management to ease the stress of GS. For instance, if a GC does not send data packet over a period of time, Gate can disconnect the network to this GC for saving net resource.

III. PERIPHERALS INPUT SECURITY

After the connecting of the client and server, a player can input an account and password through peripherals (keyboard, mouse, USB card, cell phone, etc) and send them to server for verification to acquire detailed game data. The server can validate the legality of the player through account and password.

978-0-7695-4719-0/12 $26.00 © 2012 IEEE
DOI 10.1109/CSSS.2012.473

Identifying code, security control, soft keyboard and secret card are technologies to enhance the client logging on security, which are widely used in MMORPG.

A. Identifying Code

The implement of CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) is: a group of words or numbers (mixture of Chinese, English and numbers) generated by server at random and sent to client as a distortional and blurred image. Players need to input the words or numbers displayed in the image and send to server to validate. The distorting and blurring to image can avoid to be recognized by the programs with image recognition technology.

Despite the graphics identifying codes, voice identifying technology is widely researched and used.

Identifying code is used to avoid being brute forced and anti-add-ons. With the development of image and voice recognition technology, it cannot be avoided to be recognized automatically for identifying code but mainly used to prevent cheating.

B. Secuity Control

The client users usually input account and password in an edit box window. In Windows operating system, most is a standard Edit control. One application can send WM_GETTEXT or EM_GETLINE messages to obtain the account and password of the client side.

1) Improved EditEx control

The problem of standard Edit control is that the WM_GETTEXT and EM_GETLINE messages sent by other applications are not validated. EditEx limit WM_GETTEXT and EM_GETLINE messages only can respond current application itself.

2) Security control which can prevent keyboard hook

Windows applications are based on message-driven mood, so once the user inputs the message into input devices (keyboard and mouse), the message is delivered in system queue to wait being acquired and processed. In a same application, messages are mostly acquired and processed by message processing mechanism itself. Windows messages can be acquired and processed by other applications via Hook, which is the implement basis of both most remote assistant program and monitoring program and many Trojan horses.

The implement principle of security controls is preventing other programs from loading hook or encrypting to keyboard directly. Implement methods of different security controls are different, most of those carry on level ring0 of operating system, whose common features are forbidden the operations of selecting, copying, pasting text and inserting characters.

C. Soft Keyboard Inputing

Soft keyboard is that software simulates all characters visible in keyboard and users click them by mouse, for forbidding Trojan horses record the inputs of hard keyboard. Compared with security control, the realization of soft keyboard is relatively simple and cost-effective.

The implement of soft keyboard is as follows:

- The random keyboard is generated and displayed;
- Responds the clicks of user;
- Acquires the key assignments of user;
- Treats with the user inputs.

D. Engyption of One Time Passord (OTP)

One Time Password (OTP) is an application of Security Studies, which changes the password dynamically every time to ensure the safety of user's password. The forms of OTP mainly are matrix card, secret card, cell phone OTP, etc.

1) Matrix Card

The technology of matrix card is a reduced form of OTP, whose basic theory is generating groups of numbers at random and printing them on a card or send to users as a graphic. Under the Challenge/Response mechanism, when the user logs on, the server brings a query message through random algorithms and asks the user input specified groups of numbers. Once the client side receives the query message, it inputs the numbers printed on card and feeds back to server for validation, which is the whole process of implement of OTP.

2) Time Token Card

Generally time token card is hard device of USB card form, which is an implementation form of OTO and mostly uses dynamic password validation of time synchronous. The password is changed every interval (generally 60 seconds) and synchronized with the server.

The key point of time token ring is the synchronization of the server and client, who use the same random algorithms to generate a group of random password according to time factor.

3) Cell Phone OTP

The theory of cell phone OTP generally is a 6-8 bits random password that sent to user's cell phone as a message, which is similar to the theory of time token card. This password is timelines as same as time token card password.

Cell phone OTP and network static password construct a relatively safe two-path validation.

IV. NETWORK TRANSMISSION SECUITY

Besides the input security, transmission security is the second essential element. The process of implementation of data transmission is:

(1) Encrypting to something important (accounts and passwords) and message itself with encryption algorithm;

(2) Transmitting the encrypted message with secure channel.

Hash function is commonly used in the algorithm in handling account and password messages. A good hash function should be with features of One Way, collision resistance, hash distribution uniformity and difference distribution uniformity, etc.

SHA and MD are two series of frequently used hash algorithms. MD5 is adopted in this article to mange the accounts message.

There is an assumed input M whose length is n ($n \geqslant 0$) bit(s), the algorithm MD is needed to fill M with M' ($b_1b_2\cdots b_nb_{n+1}b_{n+2}\cdots b_Lb_{L+1}b_{L+2}\cdots b_{L+64}$).

$L \equiv 448 \bmod 512$ ($n+1 \leqslant L \leqslant n+512$);

b_{n+1} fills in 1; when $L \geqslant n+2$, $b_{n+2}\cdots b_L$ fills in 0, the filler in $b_{L+1}b_{L+2}\cdots b_{L+64}$ is $n \bmod 264$ (n is length of input M, when n>264, the filler is low 64bit of n).

Four 32bit variables are used in message abstract:

_int32A = 0x01234567;

_int32B = 0x89ABCDEF;

_int32C = 0xFEDCBA98;

_int32D = 0x76543210;

The 128bit hash result is computed several times by MD5, which uses the following four functions to break the filled M' into fragments (512bit):

F(X, Y, Z) = (X&Y) | (~X&Z);

G(X, Y, Z) = (X&Z) | (Y&~Z);

H(X, Y, Z) = X^Y^Z ;

F(X, Y, Z) = Y^(X | ~Z).

&, |, ^, ~ are logic operators in C/C++ of AND, OR, XOR and NOT.

The account information computed with MD5 is assumed to be safe (because of the hash collision, this assume is of a risk). Account information is sent to logging on server as a message to validate through network. In order to pretending the account information from monitoring and stealing to log on invalid, the matrix card and other OTP technology should be cooperated in practice to ensure the safety of account information.

After the validation of user account information, logging on server informs the game client to connect with the game server and enter the game.

Only after the validation of user account information, GS allows GC to establish long-connection with it. Once the connection is built, GS firmly believes that GC is valid user and communicates with it.

In the process of game, GS confirms identification of GC according to connecting socket ID. Therefore, GC cannot simulate other GCs to communicate with GS. The attacks to GS suffered from GCs (even valid users) are mainly aiming at game logic security.

V. GAME LOGIC SECURITY

Client side logic validation is that the legality of all operations on data which game client sends to server must be validated by server.

Observing from the angle of the security of server design, the data from client are considered to be trustless. The client requests needed to be validated include some important game data such as AI, skills, articles, trading, etc.

To release the burden of logic processing to game server, it is customarily that moving AI, state AI and action AI are processed by game client. These AIs include: displacements, direction altering, speed changing, entering the ground, leaving the ground, state modifying, action processing, etc.

The processing of client mainly is the AI verified request sent to server after the calculating. The server which stores the game data such as players and NPC validates the legality of AI request. To take the player displacement request from client as an example, this article explains the processing of AI logic validation from server.

The server calculates every player's displacement (StateDistance) and the times of displacement (MoveTimes). When the StatDistance or MoveTimes accumulates until the validating limitation or the player moving speed (MoveSpeed) has changed, the server validates the legality of StatDistance according to the player's current MoveSpeed and MoveTimes:

$$| \text{MoveSpeed} \times \text{MoveTimes} - \text{StatDistance} | < \Delta d$$

Among this, Δd is the tolerance.

After the request sent from client that the player's current position (Xc, Yc) moving to target position (Xt, Yt), the server firstly determined if validate the legality to StatDistance. If there is no need to validate or the validation succeeds, the player's moving is considered to be valid; if the validation fails, the player's position is enforced as the start position (Xs, Ys).

In every validation, the server resets the play's StatDistance and MoveTimes as 0, and reset the start position (Xs, Ys) as the current position (Xc, Yc), no matter the validation is valid or not.

The server also can calculate the play's invalid moving times (InvalidTimes). Once the times is up to the warning times, the server will confirm that the player has malicious attempt to move and disconnect the network with the client, even close down the player's account.

VI. TRANSACTION SECURITY

Transaction is one of notable features of MMORPG virtual world. Similar with the effect of real world on the social economy, transactions in the game world influence the economic system in game world, which are both important method and result of players interacting with others.

The broadly-defined transaction of MMORPG includes goods-goods transaction and money-goods transaction, while the narrowly-defined transaction usually does not include the goods-goods transaction between players and NPC (like the virtual properties or experience NPC rewards to players when they finish the tasks NPC gives). The transaction logic in this article is narrowed-defined.

According the classification of the hosts between both sides, transactions in game world can be sorted out into two categories:

(1) Money-goods transactions between players and NPC (purchase in stores, sales in stores);

(2) Goods-goods transactions and money-goods transactions between players.

According the terms of electronic business, the transactions between players and NPC is called B2B (Business to Consumer) and the transactions between players is called C2C (Consumer to Consumer).

The operation in transaction process includes: request for transaction, affirming transaction and canceling transaction.

To takes a relatively complex money-goods transaction between two players as an example, this article illustrates the validation process of the transaction logic between the server and the players.

A. Transaction Session

Session is a container with lifecycle which stores and manages all terminals that join into the session. In the development of MMORPG, session is used to manage many kinds of the interactive process with Request/Response mechanism in a period of time, such as team management, transaction management, auction sale and consignment management, etc.

Transaction session is used to mange one transaction between players. When a player request for transaction, the

server firstly validate if the two parts have the qualification of transacting:

(1) Both of the players coexist (online, not death) and the transaction distance meets the limitation;
(2) There is no other operation from the both sides relating to the transaction goods: death, purchase, other sessions, etc.
(3) There is enough space to store the transaction goods in both sides' backpacks (the virtual container used to store goods);
(4) The limitation of other transaction logic (ranking, gender, profession, race, labor union and country, etc).

When the both sides agree on the transaction and pass the validation from server, the transaction session and session terminal are created and the session will be managed in transaction session.

Because some operations such as articles moving, transaction affirming and canceling transaction and so on involve in the process of transaction, the whole process will last for a short while and the server will create a session for every transaction and create a session terminal for both sides separately.

B. Transaction Session Terminal

A terminal associates with a player who has joined in session, and takes charge of the corresponding logic process of the session.

The main function of terminal is to manage several shadow containers. Different from the physical containers which store articles in practice such as players' backpacks and storehouse, the shadow container actually is an image of the article in physical container, which is convenient for players moving the articles in physical container and being showed the effect after the moving. Once the player cancels the operation, these articles are still in the original physical container; just when the player affirms the moving operation at last, the article will be deleted from the physical container.

It is convenient for the players to operate and lay the articles with shadow containers in transaction session terminals. If the transaction is canceled, the shadow container will be released along with session terminal. Only when the transaction succeeds, the server will operate the players' physical containers.

C. Management to Transation Exception

Some events or operations may lead to the transaction cannot be carried out and ended. These exceptions are:

(1) Either side disconnects the network no matter actively or passively;
(2) Either side is dead;
(3) Either side cannot be validated (any matters and operations cause the players objects be deleted or can not be affirmed);r
(4) Either side cancels the transaction;
(5) There is no enough space of either side's backpack.

When the transaction session is over, the server will release the transaction session and session terminal resource and inform both sides to terminate the transaction.

VII. CONLUSIONS

The security problem is independent from online game itself, but it is the basic guarantee to the survival and the development of online games. This article not only discusses the security mechanism comprehensively from aspects of network topology structure, users' input, data transmission, game logic and so forth, but provides a feasible solution to the design and realization of some key technology.

ACKNOWLEDGMENT

This article acquired Hebei Province Higher Education Natural Science Research Guidance project — " High Performance Concurrent Design on Online Games Server Architecture" (Project NO. Z2010311).

REFERENCES

[1] C.Zhan, W.Li, E Safaei, P.Ogunbona. Face to Face Communications in Multiplayer Online Games: A Real-Time System. Human-Computer Interaction, Part Iv, HCH 2007, pp. 40l–410.

[2] Bjom Knutsson, Honghui Lu, Wei Xu and Bryan Hopkins, "Peer-to-Peer Support for Massively Multi-player Games". INFCOM 2004, March 2004, Hong Kong, China.

[3] Thor Alexander, "Massively multiplayer online game". 2006, Beijing, China.

[4] M.Assiotis, V.Tzanov. A distributed architecture for MMORPG Proceedings of 5th ACMSIGCOMM workshop on Network and system support for games, Singapore, 2006, Article4.

[5] S.Hu, G.Liao, Scalable peer-to-peer networked virtual environment, Proceedings of 3rd ACM SIGCOMM workshop on Network and system support for games, 2004, pp. 129–133.