УДК 004.42

M. Błaszczak, student (Lublin Catholic University, Poland);
P.P. Urbanovich, prof. (BSTU, Minsk, Belarus)

SERVER SECURITY OF THE MULTIPLAYER GAME «PROJECT I.G.I. 2: COVERT STRIKE»

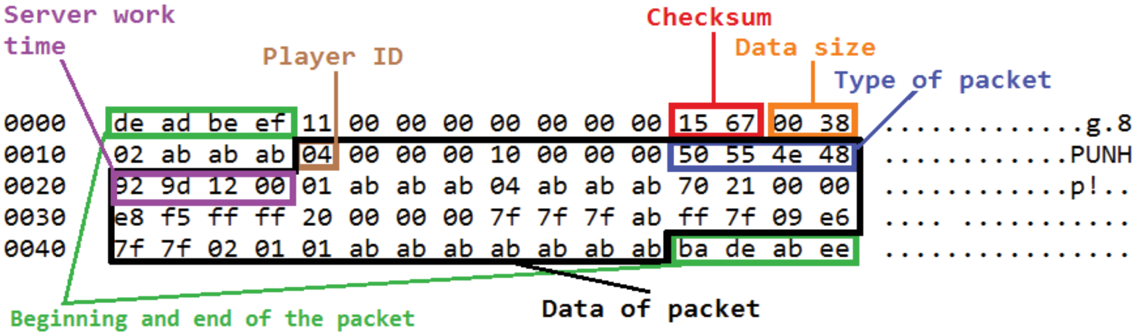
Secure of applications for multiplayer games is very important aspect for creators. The more expanded system requires a greater security level.

Often you can find systems of virtual money, payment. This is undoubtedly great opportunity for cybercriminals [1-2].

The modern games have really well protection of data and operations while elder games, which are still of interest to players, are exposed for various attack. The great part of games like those aren't supported by creators. Therefore users acquire control of security and create applications which support protection. The example of a game like that is "Project I.G.I. 2: Covert Strike". Creators stopped support servers of the game and because of that there have been many attacks and modifications which destroy the game.

One of the main ways to attack the server is sending false packet [3-4]. After analysis of a network traffic during the game you can observe the pattern of communication between the server and the gamer.

The following figure (fig. 1) shows an example of the most common packet format. After deciphering the contents of the packet you can easily compile your own data which may be dangerous for the server.



```
0000  de ad be ef 11 00 00 00 00 00 00 00 15 67 00 38  ..............g.8
0010  02 ab ab ab 04 00 00 00 10 00 00 00 50 55 4e 48  ............PUNH
0020  92 9d 12 00 01 ab ab ab 04 ab ab ab 70 21 00 00  ............p!..
0030  e8 f5 ff ff 20 00 00 00 7f 7f 7f ab ff 7f 09 e6  .... ...........
0040  7f 7f 02 01 01 ab ab ab ab ab ab ab ba de ab ee  ................
```

Fig. 1 – An exemplary format of a packet of communication between the server and the player

The most popular way to attack is overflow the buffer of a player name. In the game application user has restricted ability to set the name to 32 characters. However it isn't secure for server application. During sending the packet which is responsible for create the player, you can include

1000 characters long name. The server applications will try save that name in memory but specified character string is going to be too long what will result in a program error.

The next form of attack is “Format string attack”. It is based on misuse of arguments delegated to the following functions in language *C* which formats characters strings.

It’s hard to secure attacks which have been shown [3-5]. One of the ways to do that is creating system and accurate protection the server by the firewall. One of the elements of the system is application for players which is responsible for registrations, logging in and questions for access to the server.

The next element is application to the server which gives permission for the access, adds an exception to the firewall for a particular gamer and when he ends the game, the exception is going to be deleted. The firewall set to reject all of the packets from the outside is the crucial element.

Activity of the whole system may be show in that way:

1) registration with the e-mail address;
2) logging in to the application;
3) sending questions for the access to the server;
4) the application on the server checks players authenticity and adds the exception to the firewall with giving IP address and port which the player connects from;
5) after leaving the gave application deletes the exception from the firewall;

Thanks to this solution, the player is not able to send any crafted packets because he has no access. To protect the server more, you can set a server password that changes before each entry to the game. A player does not know it so there is no way to start an attack, because without the first packet in which the server password is sent, other packages will not be analyzed by the program. The extra solution is allocation the unique keys for every single player [4]. If any attack will be detected, you can lock adding access to a server for a player with the particular key. If a player registers again system will recognizes the previously used key and won’t allowed finish registration.

To secure server better you can use the algorithm which can detect the attack and lock it. A program like that one is able to work with servers log, that is a text file which includes register of the servers operations with exact date and hour. When a player enters a game the following line shows in the log:

```
"[20:30:42] Server info sent to 192.168.1.1:26015".
```

We download time and wait for the next line:

"[20:30:42] NETWORKPACKET_TYPE_CLIENTCONNECT [192.168.1.1:26015]".

If the time of saving in log is the same or there is one second difference it means sending packets by the attacking program. In a regular screenplay of entry to the game the difference of the time between those lines is minimum three seconds but mostly over 5 seconds.

The next proof for sending packets by the assailant are destination ports. If in both lines ports are different, it means that attacking packets were sending. This is due to the fact that the program created to attacks like that after each sending the packets is closed by the nest, this results in a change of port. The player who enters a game by the application sends all packets through the same port. Similar algorithm you can use during the analysis of network traffic which works faster.

In conclusion it must be said that to secure a server it is necessary to stock up on a system to control players. As an extra function it may be checking originality of the files to the game because there are a lot of modifications which disturb during playing the game.

REFERENCES

1. Pieprzyk, J. Teoria bezpieczeństwa systemów komputerowych/ J. Pieprzyk, T. Hardjano, J. Seberry. – Wydawnictwo Helion, 2003. – 595 s.

2. Paweł Urbanowicz, Marek Smarzewski. Bezpieczeństwo w cyberprzestrzeni a prawo karne/ Księga pomiątkowa ku czci Księdza Profesora Andrzeja Szostka MIC. – Lublin: Wydawnictwo KUL, 2016. – S. 489-496.

3. Urbanovich, P. P. Zashchita informatsii metodami kriptografii, steganografii i obfuskatsii/ P.P. Urbanovich: ucheb.-metod. posobiye dlya stud. – Minsk: BGTU, 2016. –220 s.

4. Ochrona informacji w sieciach komputerowych / pod red. prof. P. Urbanowicza. – Lublin: KUL, 2004. – 150 s.

5. Makas, S. B. License Protection of a component of web-applications on .Net framework / S. B. Makas, P. P. Urbanovich // New Electrical and Electronic Technologies and their Industrial Implementation: proc. of the 5-th Intern. Conf., Zakopane, Poland.– Lublin. 2007. – P. 99.