He *et al. EURASIP Journal on Wireless Communications and Networking* (2020) 2020:63
https://doi.org/10.1186/s13638-020-1660-5

EURASIP Journal on Wireless Communications and Networking

RESEARCH Open Access

Analysis of computer network attack based on the virus propagation model

Yanshan He*, Ting Wang, Jianli Xie and Ming Zhang

Abstract

The conventional method can make a reasonable analysis of common network attacks, but the reliability of the analysis is low under the virus propagation model. This paper proposes a new research method of computer network attack analysis based on the virus propagation model. Based on the relationship between the framework and daemon, the framework of the model for computer network attack analysis is set up, the attack analysis technology of computer network is determined, and the construction of the model for computer network attack analysis is completed. Computer attack objects and computer attack process are analyzed, and computer network attack analysis is carried out. Using the coverage test and the uncertainty test, the parameters of the reliability calculation variables are measured and the reliability calculation formula is replaced. It is concluded that the designed method of computer network attack analysis is 47.15% more reliable than the conventional analysis method, and is suitable for the network attack analysis under the virus propagation model.

Keywords: Virus propagation, Network attack, Interface construction, Analysis technology, ARP analysis

1 Introduction

In the conventional method of computer network attack analysis, the data end analysis method is used, which can make a reasonable analysis of the common network attack, but for the network attack under the virus propagation model, because of the model limitation, there is a low analysis reliability [1, 2]. Therefore, an analysis of computer network attack based on the virus propagation model is proposed. Depending on the framework daemon relationship, the frame analysis unit is defined, and the mutual instructions and file relationships are clarified. The framework of the model for computer network attack analysis is set up to determine password attack analysis, denial of service attack analysis, buffer overflow attack analysis, data-driven attack analysis, forged information attack analysis, and address resolution protocol (ARP) attack analysis technology so that the analysis of computer attack objects and the process of computer attack are implemented. The analysis and research of computer network attack based on the virus propagation model are completed. In order to ensure the effectiveness of the designed method of computer network attack analysis, the network attack test environment under the virus propagation model is simulated, and two different methods of computer network attack analysis are used to carry out the reliability simulation test of computer network attack analysis. The experimental results show that the method of computer network attack analysis is very effective.

The rest of this paper is organized as follows. Section 2 discusses the methods, followed by implementation of computer network attack analysis in Section 3. Experimental simulation is discussed in Section 4. Section 5 concludes the paper with summary and future research directions.

2 Methods

2.1 To build the framework of the model for computer network attack analysis

The framework of the model for the computer network attack analysis is to analyze the network attacks in the virus propagation model by establishing the relationship

* Correspondence: heyanshan@mail.lzjtu.cn
Electronics and Information Engineering, Lanzhou Jiaotong University, Lanzhou 730070, China

function of the framework daemon, so as to determine the structure of the framework analysis unit, and to clarify the mutual instructions and file relations of each analysis unit.

Computer network attack under the virus propagation model adopts client/server mode to initiate computer network attack. The principle of network client/service mode is that a host provides service (server side) and another host receives service (client). A server as a host usually calls a default port and listens. If a client has a connection request on the port of the server, the corresponding program on the server will run automatically to respond to the client's request. This program, called the daemon (which was originally a term for UNIX, has been transplanted to the Microsoft Corp system) [3], and after the two machines were connected successfully, the network began to control and be controlled. The end to be controlled becomes a server, and the control end is a client. Its daemon relationship function is shown in formula 1.

$$P_{ok} = {(\Sigma \gamma_i h_i + q_k)} \Big/ {K_o} \tag{1}$$

In which P_{ok} is the daemon of the program, γ_i is the state coefficient of program's sub-process, h_i is the new meeting words in the sub-process, q_k is the reset file to create the mask, and K_o is the security factor.

The network attack under the virus propagation model usually needs to install the server program on the controlled computer and install the client program on the master computer [4]. The entire control process is generally first to execute the client program on the master computer, and then the client program will send a connection request to the server program in the controlled end computer and establish a network connection. Then the client program can control the computer to be controlled through the server-side program and can carry out various kinds of network attacks and control. It mainly includes modifying the setup of the controlled terminal system, obtaining the list of the target computer process, closing or restarting the operating system in the controlled terminal computer, activating the goods to discontinue various processes on the controlled end, recording and extracting the remote keyboard events, and managing the files and folders of the controlled computer.

Therefore, the analysis unit in the framework can be divided into two parts: one is the server analysis unit, and the other is the client analysis unit. The analysis unit in the framework mainly analyzes the path information among the client, the communication side and the server side [5]. The analysis unit structure includes analysis unit instructions and analysis files.

Analysis unit instruction refers to a string of characters sent by the analysis unit on the client side to the server to perform some function. The return instruction includes acknowledgement information and error message. It is a string of characters sent back from the server to the client. An analysis file is a file that analyzes the transmission between the client and the server, that is, the files to upload and download at both ends. In order not to confuse instructions with files, we define an internal protocol to distinguish them. The format of the instruction we define is as follows: a multi-bit string type, in which the first three bits are numeric labeled bits to indicate the function represented by this command, from the fourth bits, it represents different structures from the first three bits, which can be represented as the path of the disk or folder, the structure of the file (file path, file name and file size), process name, specific string, etc. The instructions and files have the following relationship [6]:

$$Z_p = \sigma_o \gamma_i + \frac{\partial^2 q_k dx}{h_i} W_0 \tag{2}$$

in which Z_p is the program instruction, σ_o is the string mask law, γ_i is the state coefficient program's sub-process, h_i is the new meeting words in the sub-process, q_k is the reset file to create the mask, and W_0 represents the file attributes which are transmitted between the client and the server.

Since the client is connected to the server side, the server side initiatively transfers the list of drivers of server side computers to the client, so we do not define to view the command format of the list of drivers [7]. For file transfer, we do not define the corresponding format, but only transmit it from the first byte in turn after receiving the corresponding instruction. The following table lists the file interfaces that we define, and the corresponding path relations between the analysis instructions and the files are shown in Table 1.

The file path in the file structure of the upload file is the path for the client to place the file on the server. The name of the file is the file name established on the server side [8]. The file path in the file structure of the downloaded file is the location of the file that the client wants to download on the server side

2.2 Analysis of computer network attack

The computer network attacks under the virus propagation model include password attack analysis, denial of service attack analysis, buffer overflow attack analysis, data-driven attack analysis, forged information attack analysis, and ARP attack analysis technology. The password attack analysis technology is the simplest and most direct attack analysis technology under the virus

Table 1 Analysis of the relationship between instructions and files

Instruction set	Format	Interface
View the file list	001+ The spec if ic path	001c:\ 001c:\windows\system32
Upload a file	002+ file structure (file path, file name, file size)	002e:\film\milan.rmvb70000000
The download file	003+ file structure (file path, file name, file size)	003e:\music\better.man.mp3
Delete the file	004+ file structure (file path, file name)	004d:\study\l. txt
Extract process list	005	005
Close the process	006+ the name of the process to be closed	006winword.exe
Running processes	007+ file structure (file path, file name)	007d:\hello world.exe
Remote shutdown/restart	008+ a spec if ic string (poweroff/reset)	008poweroff 008reset
The bug report	009+ error instruction	009005 009007reset

propagation model. Password attack analysis is an attack technique used by an attacker in view of other population orders, and the attacker often attacks the user's password as the start of the attack when it attacks the target. As long as an attacker can guess or determine the user's password, he can get access to a machine or network and can access any resource that the user can access. If this user has domain administrator or root user permissions, this is extremely dangerous [9].

Password attacks mainly include dictionary generation, password interception and deception, and non-technical means. The dictionary generation attack uses a word library to generate passwords, which contain a number of word roots that can form a password to generate a guessing password under the rules. Because the choice of roots is based on people's habit of making passwords, they are obtained after a lot of statistics. The non-technical means in password attack mainly refers to the compilation rules of passwords obtained by non-information means, which satisfies the following rule [10]:

$$k = \frac{\sum K_o + \left(R'_{kk}/S_{x,k}\right)}{\sum\left(q_j b_j + \Delta G_j\right)} \tag{3}$$

in which k is the non-technical means coefficient in the password attack, K_o is the security factor, q_j is the query recognition rate of finger command of the target host, and $S_{x,\ k}$ is the number of bytes in the directory query service within the k time period. R'_{kk} is the effective byte rate obtained by k time password attack. b_j stands for the difficulty of setting password. ΔG_j represents password complexity. Password attack objects mainly include Linksys, MikroTik, NETGEAR, and TP-Link routers used in the small and home office (SOHO) and QNAP network additional storage (NAS) devices, which have not been found to be infected by other network equipment suppliers [11].

A denial of service attack is a simulation of a denial of service attack. First, an attacker wants to stop the target machine from providing service. It is one of the hacker's commonly used attacks. In fact, the consumption attack on the network bandwidth is only a small part of the denial of service attack. As long as it can cause trouble to the target, some services are suspended or even the host is dead, all of which belong to the denial of service attack [12]. The problem of the denial of service attacks has not been properly solved. The reason is the security defects of the network protocol itself, and the denial of service attack has also become the ultimate technique of the attacker. The attacker performs a denial of service attack; in fact, it enables the server to achieve two effects: one is to force the server's buffer to be full and not to receive a new request; another is to use IP deception to force the server to reset the connection of the illegal user and to affect the connection of the legitimate user.

Connectivity attack refers to the use of a large number of connection requests to impact the computer, making all available operating system resources depleted, and eventually the computer cannot rehandle the request of the legitimate user. Common attack means are synchronous flood, WinNuke, PNG of death, Echl attack, ICMP/SMURF, Finger bomb, Land attack, Ping flood, Rwhod, tearDrop, TARGA3, UDP attack, OOB, and so on [13].

Buffer overflow attacks mean that when the computer fills the buffer with the number of bits more than the capacity of the buffer itself, the overflow data is covered on the legitimate data. Ideally, the program will check the length of the data and do not allow the input of characters that exceed the buffer length. But most programs assume that the length of data is always matched with the allocated storage space, which is a hidden danger for buffer overflow. The buffer zone used by the operating system, also known as the "stack". Between operations, instructions are temporarily stored in the stack, and the stack also has buffer overflow [14].

By writing the content beyond its length to the buffer of the program, it causes the overflow of the buffer, which destroys the stack of the program, and makes the program to execute other instructions, so as to achieve the purpose of the attack. The reason for the buffer overflow is that the parameters entered by the user are not carefully checked in the program. For example, it is as the following program [15]:

```
void function(char *str)
{
char buffe r[16]; strcpy(buffer,str);
}
```

The strcpy () above will directly transform the content from str to copy into the buffer. So long as the length of str is greater than 16, it will cause buffer overflow and cause the program to run wrong. There are standard functions for problems like strcpy, strcat (), sprintf (), vsprintf (), gets (), scanf (), and so on.

Of course, filling anything in the buffer zone will cause it to overflow. Generally, there will only be segmentation fault, which cannot achieve the purpose of attack. The most common way is to create a buffer overflow to enable the program to run a user shell and execute other commands through shell. If the program belongs to root and has suid permissions, the attacker gets a shell with root permissions and can operate any on the system [16].

The reason why buffer overflow attacks become a common security attack is that overflow vulnerabilities in buffer zone are common and easy to implement. Moreover, the main reason for a buffer overflow to become a remote attack is that the overflow vulnerability gives the attacker everything he wants: implants and executes the attack code. The embedded attack code runs a program having overflow vulnerabilities with certain permissions to get the control of the attacked host.

Overflow vulnerabilities and attacks in buffer zone take many forms. Accordingly, defense means vary with different attack methods, including effective defense measures for each type of attack.

The counterfeit user attacks the deception gateway, and host A imitates host B to send a forged ARP message to the gateway, causing the gateway's ARP table to record the wrong address mapping relationship of host B, so that the normal data message cannot be correctly received by host B. The false ARP message is sent to host C by cheating other host A and phishing host B, causing the host C's ARP table to record the wrong address mapping relationship of host B, so that the normal data message cannot be correctly received by host B [17].

The attack of ARP is to achieve ARP deception by forging IP addresses and MAC addresses, which can generate a large amount of ARP traffic in the network to block the network. An attacker can change the IP-MAC entry in the target host's ARP cache by making a continuous delivery of a forged ARP response packet, resulting in a network interruption or a middleman attack [18].

ARP attacks mainly exist in the LAN network. If a computer is infected with an ARP Trojan in the LAN, the system that infects the ARP Trojan will try to intercept the communication information of other computers in the network by means of "ARP deception" and thus cause the communication failure of the other computers in the network [19].

The attacker sent a forged ARP response to computer A and tell computer A that the MAC address corresponding to computer B's IP address 192.168.0.2 is 00-aa-00-62-c6-03, and computer A believes and writes the corresponding relationship into its own ARP caching table. When the data is sent later, the data that should have been sent to computer B are sent to the attacker. Similarly, the attacker sends a fake ARP response to computer B and tells computer B that the MAC address corresponding to computer B's IP address 192.168.0.2 is 00-aa-00-62-c6-03, and then computer B will also send the data to the attacker [20].

At this point, the attacker controls the traffic between computer A and computer B. He can choose to monitor the traffic passively, get the password and other secret information, and can also forge the data and change the communication content between computer A and computer B [21, 22].

In order to solve the problem of ARP attack, 802.1x protocol can be configured on the switches in the network.

IEEE 802.1x is a port-based access control protocol, which authenticates and authorizes users connected to switches. After configuring the 802.1x protocol on a switch, an attacker needs to authenticate when connecting a switch (combined with MAC, port, account, VLAN, password, etc.), and only by authentication, it can be sent to the network. The attacker cannot send forged ARP messages to the network without authentication [23].

Based on the analysis of the computer network attack principle under the virus propagation model, the computer network attack interface is built, and the computer network attacks under the virus propagation model include password attack, denial of service attack, buffer overflow attack, data-driven attack, forged information attack, and ARP attack, to implement the construction of the computer network attack analysis model [24].

3 Implementation of computer network attack analysis

3.1 Analysis of computer network attack objects

Based on the construction of the computer network attack analysis model, we divide the network attack into three modules: the client module, the server module, and the communication module. Therefore, it first establishes three classes: client, server, and message. To define the class needed for development, and instantiate the class, the object of computer network attack can be gotten, which mainly includes client, server, message, list, disk, folder, file, process, HTTP packet, server request message, and client response message [25].

3.2 Analysis of the process of computer network attack

The unusual complexity of the invasion process leads to various kinds of intrusion, and the characteristics of the intrusion are different. It is obvious that it is difficult to use a unified formal model to describe the intrusion. The finite state automata are used to describe some typical intrusion processes and try to find out their characteristics in order to seek a formal description of the various intrusion processes.

For the automaton model M = (Q, sigma, F, S, Z) corresponding to different intrusion processes, the system state Q may be described with different objects, which can be made up of the state of one or a few monitored hosts and can also be described by the process state running within the host. Transformation condition set, also known as transformation function, is a function cluster, including attack function class, communication function class, and feature judgment function class. In the process of attack, each function is instantiated gradually, and the system changes from one state to another. For different intrusion processes, the specific meaning of each element in M may be different. In the process of using the model, through the analysis and audit of all the captured data packets and log data, the characteristic parameters are obtained to determine whether there is a system anomaly or an intrusion behavior. In either case, a state transition diagram will be obtained from the initial state to the end state. The state of the system can be known by judging the state of termination.

The TCP protocol in the Internet is a connected protocol. When two network nodes communicate, they first need to connect through the three handshake signals. When host A wants to access the resources of server B, host A first establishes a connection with server B. Firstly, host A sends a connection request with a SYI\ flag to server B. The packet contains the initial serial number of host A. After receiving the SYI\ package, server B changes the state to SYN RCVL1 and assigns the required data structure for the connection. Then server B sends the confirmation packet with the SYN/ACK flag to the host A. It contains the initial serial number of server B and clearly confirms that the serial number ACK is x + 1, which is in the so-called semi connection state. Host A receives the SYN/ACK packet and then sends the ACK packet to server B, at this time the ACK confirmation number is y + 1; server B receives the confirmation packet after the state turns to be established, and the connection is completed. In this way, host A establishes a connection with server B, and then they can communicate through the link.

The above is the case of the normal establishment of a connection for the TCP protocol. However, if server B sends the SYN/ACK packets to host A without the response of host A for a long time, server B will have to wait for quite a long time. If such a half connection is too large, it is likely to consume the resources (such as buffer) that server B is used to establish a connection. Once the system resources are exhausted, the normal connection request to server B will not respond.

The specific process of network attack is as follows: attacker/intruder forges one or more nonexistent host C and sends a large number of connection requests to server B. Because the forged host does not exist, each connection request server B is waiting for a period of time because of receiving no confirmation information from the connection; a large number of connection requests in a semi connected state appear in a short time, which quickly depletes the related system resources of the server B, making the normal connection request unable to respond. It leads to a denial of service attack. Next, a finite state automaton is used to describe the network attack process.

Set M = (Q, sigma, F, S, Z). Supposing M =(Q, Σ, F, S, Z), where, $q \epsilon Q$, and q = (intruder status serves status system status). Intruder status is the state of the attacker, and its fetching range is {listen, faked, SYN SENT, ACK., SENT, failed established}; semen status is the state of server B, and its value range is {listen, SYN, RCVD, SYIN, ACK, SENT, ACK, RCVD, blocked, established}. System status indicates whether the system intrusion occurs. Its value range is {false, true}, and when system status = hue, it indicates that intrusion occurs. Σ is a set of transformation functions, including attack function, communication function, and test function. Specifically,

{

E0: fake()

E1: Communication(Res host Des host SYN-ISN, 0)

F2: Communication (Res host Des host SYN-ISN, ACK-ISN)

E3: Tcp_resource_used_out()

}

where function *E0* is used to forge a nonexistent host randomly. Function *E1* is used to issue a SYN request packet to the server and SYN-ISN is the serial number sent. Function *E2* is used to send the SYN-ACK reply packet to the server sending the connection request, and SYN-ISN and ACK-1SN are the sending and confirming serial number. The function *E* is used to determine whether the TCP connection resource on the server is running out. If it is used up, it will return to “true”; otherwise, it will return to “false.” The states of the figure are as follows:

S0=(listen, listen, false), *S1*=(&ked, listen, false), *S2*=(SYN, SENT, SYN, RCVLI false), *S3*=(failed, SYN-ACK, SENT, false), *S4*=(listen, blocked, true);

S0 is the initial state of the model. The attacker enters the *S1* state by forging a nonexistent host, and the attacker is faked. Then the host tries to establish a connection with server B, and the model enters state *S2*. The server responds to a request to establish a connection, and the model enters state *S3*, but the attacker is failed since the attacker cannot receive the SYN-ACK packet because of forgery of the nonexisting host. Finally, the model determines whether the TCP connection resources of the system are depleted. If there is no exhaustion, the model returns to the initial state; otherwise, it will enter the termination state *S4* to achieve the analysis of the attack process.

4 Experimental simulation

In order to ensure the effectiveness of the computer network attack analysis and research based on the virus propagation model, simulation analysis is carried out. In the process of the experiment, the network attack under different virus propagation models is taken as the test object, and the reliability simulation test for computer network attack analysis is carried out. The different virus types and modes of network attack under the virus propagation model are simulated. In order to ensure the validity of the experiment, the conventional method of computer network attack analysis is used as the comparison object. The results of the two simulation experiments are compared, the experimental data are presented in the same data chart, and the conclusions of the test are obtained by analyzing the reliability calculation.

4.1 Preparation of the experimental data

In order to ensure the accuracy of the simulation test process, the test parameters are set firstly. In this paper, the experiment process is simulated, and the network attack under different virus propagation models is used as the test object. Using the two different methods of computer network attack analysis, the reliability simulation test of computer network attack analysis is carried out, and the simulation test results are analyzed. Because the analytical results obtained by the different methods are different from those of analysis, it is necessary to ensure the consistency of test environment parameters in the test process. The results of the test data set in this article are shown in Table 2.

4.2 Design of the test process

In the virus propagation model, the client of the network attack uses the active port, and the server uses the passive port. When the connection is to be established, the server opens a default port and enters the monitoring state. The client puts forward connection requests to the port to the server. The server regularly reads requests from HTTP protocol and initiatively connects them. The client’s listening port is generally open at 80, and the 80 port is a port dedicated to the HTTP protocol. In order not to let the server’s firewall find, we can also use a port number greater than 1024.

Because of our testing environment and some of the limitations of the software, it cannot be widely tested on the Internet or in the military network, but only through several computers using a hub to form a LAN for testing. On this LAN, firstly, a fully shared folder (files can be readable and written and deleted) is built in the FTP server as a springboard computer.

Then, when the client is open, the program takes the active IP address of the client and the open port number, and then sets up a file in the shared folder of the FTP server. The file is named ip, the type is.txt, and the format content is as follows:

IP:10.131.1.130
Port:5656

Finally, after the server is started, it first connects to the computer as a springboard, and then reads the shared folders on the FTP server. In this folder, keep looking for the existence of a file ip.txt, if there is a file ip.txt, read the content of the file, get the client’s IP and port number; if it does not exist, it will be searched periodically at a certain interval until the file is found or the server is closed.

When the server reads the IP and port numbers of the client, the server initializes and then connects the client according to the read IP and port numbers.

For a communication link between two network nodes, link encryption can provide security for the data transmitted on the Internet and can also intercept the commands and data transmitted on the network through the firewall. For link encryption, all messages are encrypted before being transmitted, and the received messages are decrypted at the destination node. A more common des encryption algorithm is used.

The client on the client machine is opened. The client program takes the active ip (10. 131. 1.130), and the available port number (5656), writes the native ip address and port number in the ip.txt file, and then uploads the ip.txt file to the FTP server of the springboard. The server first connects to the FTP server that is a springboard computer, reads the shared folders on the FTP server, and downloads the ip. txt to the computer of the server. Then the server program opens and reads

Table 2 Test parameter setting

Project	Type parameters	Version parameter
The operating system	Windows	7Ultimate
Central processing unit	AMD Ryzen 7 2700X	CPU main frequency 3.7 GHz Maximum frequency 4.3 GHz
The sound card	Asus Xonar D-Kara	88.2 kHz, 24 bit
The graphics card	NVIDIA Rainbow iGame GTX 1080Ti Vulcan X OC	Core frequency 1480/1733 MHz Display frequency 11000 MHz Video memory capacity 11 GB Display memory bit width 352 bit
Memory chips	Astrazeneca X3 16 GB DDR4 2400	Memory type DDR4 Main memory frequency 2400 MHz Memory capacity 16 GB

the ip. txt file to get the IP and port number of the client's computer. With the obtained IP address and port number, the server and the client program successfully establish a connection.

The attack methods of password attack analysis, denial of service attack analysis, buffer overflow attack analysis, data-driven attack analysis, and forged information attack and ARP attack analysis are selected, and the analysis coverage and uncertainty of analytical method are obtained by using two kinds of methods for computer network attack analysis. The reliability of the two methods is determined by the calculation method.

4.3 Analysis of coverage test results

In the course of the experiment, two different methods of computer network attack analysis are used in the simulation environment to analyze the changes of computer network attacks. At the same time, because of two different methods for computer network attack analysis, the analysis results cannot be directly compared. The third party analysis record software is used to record and analyze the test process and results, and the results are displayed in the comparison result curve of this experiment. In the simulation test result curve, the third party analysis and recording software function is used to eliminate the uncertainty of the personnel operation and computer equipment factors in the simulation test, which only aims at the network attack under different virus propagation models and the different methods of computer network attack analysis, and the analysis coverage test model is carried out. The test was made. The comparison curve of the analysis coverage test is shown in Fig. 1.

4.4 Analysis of the test results of uncertainty

At the same time, different network propagation models and different computer network attack analysis methods

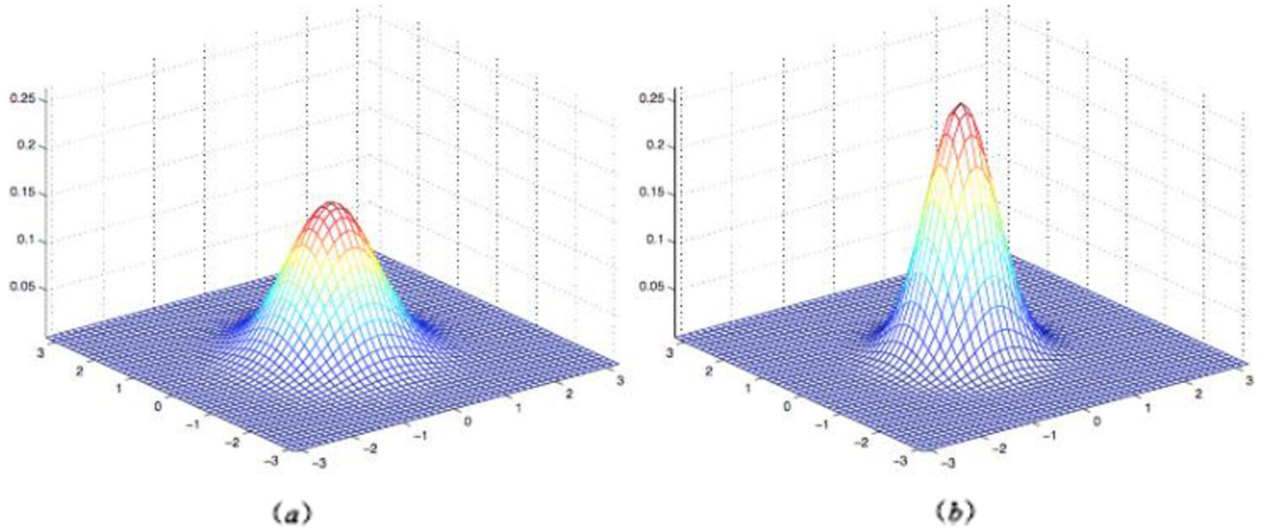

Fig. 1 Analysis of comparison result curve of coverage test. In the results of the test curve, **a** represents a conventional method of computer network attack analysis and **b** represents the proposed method of computer network attack analysis. The third party software is used to analyze and record. The average analysis coverage of the conventional method is 86.42%, and that of the method designed in this paper is 99.89%

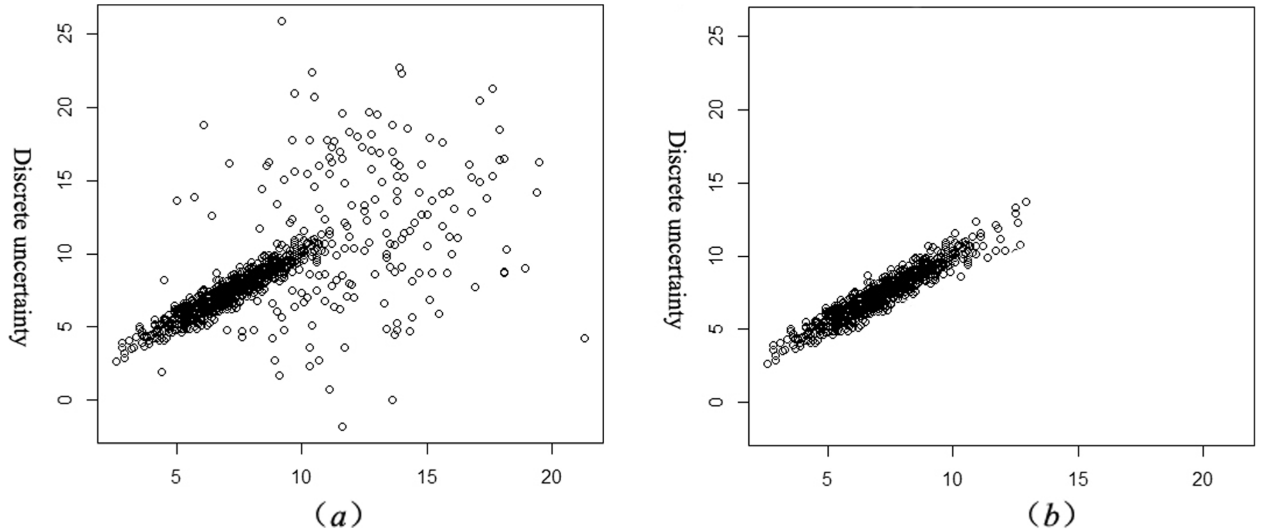

Fig. 2 Analysis of the comparison result curve of the uncertainty test. In the test curve results, **a** represents the conventional method of computer network attack analysis and **b** represents the designed method of computer network attack analysis. Using third party analysis to record software analysis records, the average analysis uncertainty of conventional method for computer network attack analysis is 15.12%, and that of the method in this paper is 2.70%

are used to analyze the uncertainty simulation test. The comparison curve of the test results is shown in Fig. 2.

4.5 Analysis of reliability analysis

According to the analysis of coverage and uncertainty by the above analysis methods, the reliability analysis is obtained by formula 4.

$$s = f(s) = \frac{1}{\theta}\int_{0}^{r} \xi\lambda\mu dx \tag{4}$$

in which ξ is the coverage of analysis, λ is the uncertainty of analysis, μ is the test influence parameter, and θ is the influence of attack effectiveness. The method proposed in this paper is S_1, and the conventional method is S_2. $\Delta S = S_1 - S_2$ is the positive number to represent the reliability improvement, and $\Delta S = S_1 - S_2$ is the negative number to represent the reliability reduction. It is substituted to formula 4 to obtain $\triangle S$:

$$\begin{aligned}\Delta S &= S_1 - S_2\\ &= \frac{1}{\theta}\int_{0}^{r} \xi_1\lambda_1\mu_1 dx - \frac{1}{\theta}\int_{0}^{r} \xi_2\lambda_2\mu_2 dx\\ &= 0.471523\end{aligned} \tag{5}$$

It can be seen that the proposed method of computer network attack analysis is better than the conventional method, and the reliability is improved by 47.15%. It is suitable for the analysis of network attacks under the virus propagation model.

5 Conclusion

In this paper, an analysis of computer network attack based on the virus propagation model is proposed. Based on the construction of the computer network attack analysis model, the object and process of computer network attack are analyzed, and the research in this paper is completed. The experimental data show that the method designed in this paper is very effective. It is hoped that the research in this paper can provide a theoretical basis for computer network attack analysis method under the virus propagation model.

Abbreviation

ARP: Address resolution protocol

Acknowledgements

National Natural Science Foundation of China, Research on Joint algorithm of Network selection and Cognitive Spectrum allocation in High-Speed Rail Communication Environment, 61661026.

About the authors

Yanshan He (1983-), female. She graduated from Lanzhou University in China in 2010 as a Master of Computer Science. She is currently a lecturer in the school of Electronic and Information Engineering, Lanzhou JiaoTong University. Her research interests include data mining and the technology of Internet of Things.

Ting Wang (1981-), female. She graduated from Lanzhou Jiaotong University in China in 2008 as a Master of Computer Science. She is currently an associate professor in the school of Electronic and Information Engineering, Lanzhou JiaoTong University. Her research interests include computer networking and the technology of Internet of Things.

Jianli Xie (1972-), male. He got his PhD of Intelligence Traffic from the Lanzhou Jiaotong University in China in 2014. He is currently a professor in the school of Electronic and Information Engineering, Lanzhou JiaoTong University. His research interests include communication engineering, intelligence traffic, and the technology of Internet of Things.

Ming Zhang (1982-), male. He got his PhD of Engineering from the Pukyong National University in Korea in 2015. He is currently an associate professor in the school of Electronic and Information Engineering, Lanzhou JiaoTong

University. His research interests include machine intelligence and the technology of Internet of Things.

Authors' contributions

YH wrote the entire article. TW is responsible for data preprocessing. JX was responsible for the simulation part of the experiment. MZ is responsible for the analysis of the results of the article. All authors read and approved the final manuscript.

Funding

National Natural Science Foundation of China, Research on Joint algorithm of Network selection and Cognitive Spectrum allocation in High-Speed Rail Communication Environment, 61661026.

Availability of data and materials

The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

Ethics approval and consent to participate

This article does not contain any studies with human participants or animals performed by any of the authors.
All authors agree to submit this version and claim that no part of this manuscript has been published or submitted elsewhere.

Competing interests

The authors declare that they have no competing interests.

Received: 20 November 2019 Accepted: 30 January 2020
Published online: 12 March 2020

References

1. L. Wenjie, H. Lihua, Y. Xinfeng, Simulation analysis of optimal network port communication selection model. Computer Simulation **33**(8), 248–251 (2016)
2. M. Yangyang, H. Gaofeng, Z. Bo, Simulation study on optimal identification of attack information in power resource network. Computer Simulation **34**(6), 104–107 (2017)
3. S. Lazfi, S. Lamzabi, A. Rachadi, et al., The impact of neighboring infection on the computer virus spread in packets on scale-free networks. Int J Modern Physics B **31**(30), 1750228 (2017)
4. Z. Frontistis, C. Drosou, K. Tyrovola, et al., Experimental and modeling studies of the degradation of estrogen hormones in aqueous TiO2 suspensions under simulated solar radiation. Ind. Eng. Chem. Res. **51**(51), 16552–16563 (2017)
5. H. Rahbari, M. Krunz, L. Lazos, Swift Jamming Attack on Frequency Offset Estimation: The Achilles Heel of OFDM Systems. IEEE Trans. Mob. Comput. **15**(5), 1264–1278 (2016)
6. A. Souyah, K.M. Faraoun, Fast and efficient randomized encryption scheme for digital images based on Quadtree decomposition and reversible memory cellular automata. Nonlinear Dynamics **84**(2), 715–732 (2016)
7. B. Yao, X. Li, L. Shi, et al., A Multiscale Model of Reentry Plasma Sheath and Its Nonstationary Effects on Electromagnetic Wave Propagation. IEEE Transactions on Plasma Science **PP**(99), 1–8 (2017)
8. K. Jung, H. Hu, L.J. Saif, Porcine deltacoronavirus infection: etiology, cell culture for virus isolation and propagation, molecular epidemiology and pathogenesis. Virus Res. **226**, 50–59 (2016)
9. B. Hilary, J. Sembia, M.S. Bangura, et al., Exposure-specific and age-specific attack rates for Ebola virus disease in Ebola-affected households, Sierra Leone. Emerg. Infect. Dis. **22**(8), 1403–1411 (2016)
10. C. Sva, N. Rls, M.P. Papa, et al., Development of standard methods for Zika virus propagation, titration, and purification. J. Virol. Methods **246**, 65–74 (2017)
11. P. Zhu, L. Liang, X. Shao, et al., Host cellular protein TRAPPC6AΔ interacts with influenza A virus M2 protein and regulates viral propagation by modulating M2 trafficking. J. Virol. **91**(1), JVI.01757–JVI.01716 (2016)
12. J.N. Conde, E.M.D. Silva, D. Allonso, et al., Inhibition of the membrane attack complex by dengue virus NS1 through interaction with vitronectin and terminal complement proteins. J. Virol. **90**(21), JVI.00912–JVI.00916 (2016)
13. M. Ge, Y. Zhang, Y. Liu, et al., Propagation of field highly pathogenic porcine reproductive and respiratory syndrome virus in MARC-145 cells is promoted by cell apoptosis. Virus Res. **213**(1), 322–331 (2016)
14. A. Sayaka, O. Toru, S. Yukari, et al., TRC8-dependent degradation of hepatitis C virus immature core protein regulates viral propagation and pathogenesis. Nat. Commun. **7**, 11379 (2016)
15. R. Suzuki, K. Saito, M. Matsuda, et al., Single-domain intrabodies against hepatitis C virus core inhibit viral propagation and core-induced NFÎ°B activation. J. Gen. Virol. **97**(4), 887–892 (2016)
16. S.M. Soubies, C. Courtillon, M. Abed, et al., Propagation and titration of infectious bursal disease virus, including non-cell-culture-adapted strains, using ex vivo-stimulated chicken bursal cells. Avian Pathology, 1–10 (2017)
17. J.K. Stodola, G. Dubois, A.L. Coupanec, et al., The OC43 human coronavirus envelope protein is critical for infectious virus production and propagation in neuronal cells and is a determinant of neurovirulence and CNS pathology. Virology **515**, 134–149 (2018)
18. J.R. Glynn, H. Bower, S. Johnson, et al., Variability in intra-household transmission of Ebola virus, and estimation of the household secondary attack rate. J. Infect. Dis. **217**, 2 (2017)
19. C.J. Schweitzer, F. Zhang, A. Boyer, et al., N-Myc downstream-regulated gene 1 restricts hepatitis C virus propagation by regulating lipid droplet biogenesis and viral assembly. J. Virol. **92**(2), JVI.01166–JVI.01117 (2017)
20. Wu, X., Narasimha Reddy, A.L.: SCMFS: a file system for storage class memory. In: Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis, SC '11, New York, NY, USA, pp. 39:1–39:11. ACM (2011)Google Scholar
21. Xu, J., Swanson, S.: NOVA: A Log-Structured File System for Hybrid Volatile/Non-Volatile Main Memories. In: FAST, pp. 323–338 (2016).
22. Liu, Z., Sha, E.H.-M., Chen, X., Jiang, W., Zhuge, Q.: Performance Optimization for In-Memory File Systems on NUMA Machines. In: 2016 17th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT), pp. 7–12. IEEE (2016)
23. Zhenhua Huang, Xin Xu, Juan Ni, Honghao Zhu, and Cheng Wang. Multimodal representation learning for recommendation in Internet of Things. IEEE Internet of Things Journal (2019).
24. Z. Chen, H. Cai, Y. Zhang, C. Wu, M. Mu, Z. Li, M.A. Sotelo, A novel sparse representation model for pedestrian abnormal trajectory understanding. Expert Syst. Appl. **138**, 112753 (2019)
25. K.L. Schierhorn, F. Jolmes, J. Bespalowa, et al., Influenza A virus virulence depends on two amino acids in the N-terminal domain of its NS1 protein to facilitate inhibition of the RNA-dependent protein kinase PKR. J. Virol. **91**(10), JVI.00198–JVI.00117 (2017)

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.