A Novel Protocol Fuzz Testing Approach

Li Haifeng, Shuai Bo, Wang Jian and Tang Chaojing

School of Electronic Science and Engineering, National University of Defense Technology, Changsha, Hunan, P. R. China, 410073

lihaifeng_nudt@163.com

Keywords: Fuzz; genetic algorithm; DAT; fitness function

Abstract. In this paper we are discussing about the fuzz testing of network protocol. Compared with the general software vulnerability mining, the difficulty of using fuzz method to detect the protocol vulnerabilities is that the network protocol is a state machine, and the correctness of the input message has a strong constraint. In order to solve the problems of test message being rejected by the network protocol, a novel method is proposed by introducing the genetic algorithm into the test message generation process. Meanwhile, an improved AC algorithm is applied in the process of packet format identification. Experiments show that the proposed fuzz testing method could achieve effective results.

Introduction

Recently from both solution vendors in network industry and research communities, protocol security testing has attracted increasing interest. Fuzz testing [1] has become one of the important methods for the analysis of security of the protocol. In the process of fuzzy testing of network protocols, many different information are used as input for a given specification network protocol. When the test error occurs, through the analysis of the test case, the analysis of the network protocol processing process is achieved [2, 3]. An effective test case, with different effective test data, can trigger various problems in network applications.

The network protocol analysis based on Fuzz testing originated from a fault based test case [4-6], through developing a test set, using the "mutation" to introduce the code, so that a small change can lead to the effective change of the test case. The difficulty in detecting the flaws of the protocol is that the network protocol state machine and the input information are correct. If an exception is not in agreement with the status of the protocol requirements, it is easy to be discarded by the network protocol, so there is no opportunity to trigger the vulnerability. Therefore, an effective fuzzy testing method should have a certain ability to be intelligent, which can give priority to the test cases of the basic state of the protocol. We believe that genetic algorithm (GA) [7] model is very suitable to take the job. Genetic algorithm model is a kind of adaptive search method based on natural selection, to find the optimal or near optimal solution. GA model is able to filter out the input data which is in accordance with the protocol state, so as to improve the test efficiency.

Packet Format Identification Using AC Algorithm

In this paper, the Trace Network method is utilized to solve the problem of the network protocol identification. Aho-Corasick(AC) algorithm[8] is a global comparison algorithm, which has outstanding advantages of reducing the amount of computation.

Let $K = \{k_1,k_2,\ldots,k_n\}$ be a finite set of strings keywords and let *TS* be an arbitrary string which called the *text string*. The AC machine is a program which takes as input the *TS* and produces as output the identify in *TS* at which the keywords k_i appear as substrings. The pattern matching machine consists of a set of states. Each state is represented by a number. The main functions of the AC automaton are three main processes, namely, the design of the transfer function, the design of the failure function and the design of the output function. In the design of the transfer function, we propose one or two tuple, which is composed of the state and the character of the character which is mapped to other states. When the transfer function fails, the failure function will be triggered. The

© 2016. The authors - Published by Atlantis Press

output function is about the key words of each state, that is, the state of the output. These three functions completely define an AC automaton, which is no longer an automaton or a state machine in the classic sense.

When the text string *TS* is processed, at first we attempt to use the *goto* function G traverse the Tries *(function G)*. For a given character *si*, if no appropriate child exists in the current node, the *failure* links are traversed instead *(function F)*, until we find such a node or reach the root. When the *output (function O)* is not empty, an effective matching case is considered to be found. The process is shown in algorithm 1.

Algorithm 1: The AC machine algorithm
Input : *The input text string TS =(ts1,ts2,…tsl),pattern matching machine M , goto function G and failure function F ,output function O ,the keywords set K ={k1,k2,…kn}.*
Output : *the Locations arry Loc at which kx occur in the text string TS.*
Def *AC(TS)*
State←0
for *tsi in TS* ***do***
  while *G(state ,tsi) = fail* ***do*** *state ← F(state)* ***end while***
  state ← G (state ,tsi)
  if *O (state) !=NULL Loc.push(i , O(state))* ***end if***
end for

There are many advantages of the traditional AC algorithm, we can find out all the patterns by scan the text string only one times, and the time complexity of the AC algorithm is O(n)[8]. The time complexity is only related to the length of the text, and is not influenced by the so-called pattern set *K*. However, the disadvantage is that the required storage space is very large. There are an improved method to reduce the requirement for storage space by applying the Double Array Trie (DAT) [9].

DAT algorithm is proposed to reduce the waste of shortage memory and ensure the query efficiency of the trie, it leads to a more compact representation. DAT is a kind of tree structure which belongs to the deformation. This kind of data structure can improve the efficiency and speed of trie search. In this way, the search process is related to the length of the data that needs to be searched, and the amount of data needed to be stored is not needed.In essence, DAT is a deterministic finite automaton. When DAT reach an end state or not for state transitions, which means that the query process end. DAT uses two one-dimensional arrays called BC and TAIL. The BC include two linear matrices for operations, namely *base* and *check* matrices. These two matrices are used to check the state of the transfer and ensure the validity of the search process.

A new method which is called DAT-AC[14] was proposed by Miao Hou recently. They used the DAT to improve the communication algorithm, which is to reduce the requirement of the data storage space. Specifically, two linear matrix, namely *base* and *check* matrices are introduced into the design of the transfer function. The ASCII of input character is added into the *base* value of the current state, the father state information of the current state is stored in the table *check*. Transfer function constructs the failure function, while the output function is still constructed by the classical AC algorithm. Table *next* represents transfer function and the subscript of *next* is the position offset while the output is the status value. The subscript of the *base* is status value and the output is the *base* value.

Algorithm 2: The DAT algorithm
Input : *The input character* ***tsi*** *in the set TS ={ts1,ts2…tsl}, the current status of state* $\boldsymbol{S_{in}}$.
Output : *The next state* $\boldsymbol{S_{out.}}$ *the table* ***base[]****and* ***check[]***
Def *DAT (*S_{in}*,tsi)*
Init *base[] ,check[],next[]*
for *tsi in TS* ***do***
  State S_t*←0*
  loc =next[S_{in} *].loc +base[*S_{in}*]+tsi ;* S_t *←Next[loc];*

if *check[S_t]=S_{in}* ***then*** *S_{out}= S_t*
else return *false* ***end if***
end for

In this paper, the DAT-AC algorithm is applied in the process of packet format identification. We set the keywords from the RFC as the *K*, we take the FTP protocol for example K_{ftp} *= {PASS, USER, MDTM…}* and set the messages from the network trace as the text string *TS.* We locate the position of the pattern in the set *K[i]* for each message *TS*, and calculate the frequency of the key word, the *frequency* table *F(TS, K)={F_{k1},F_{k2},… F_{kn}}*.we take the text string TS_{ftp}={USER hello PASS yes },for example the *F(TS, K_{ftp})={1,1,0}*.And then we can cluster the protocol messages by the pattern matching result.

Algorithm 3: Protocol Packet Format Identification algorithm
Input : *The input text string TS =(ts1,ts2,…sl…),the keywords set K ={k1,k2,…kn} from the RFC.*
Output : *the frequency* table *Frq={Frq_{k1},Frq_{k2},… Frq_{kn}} and the locate table Loc={(i,kj)…}*
Def *PPFI (TS ,K)*
state←0
for *tsi in TS* ***do***
while *DAT(state ,tsi) = fail* ***do*** *state ← F(state)* ***end while***
state ←DAT (state ,tsi)
if *O (state) !=NULL L.push(i , O(state))* ***end if***
end for
while L!=NULL do
Loc(i,kj) ←L.pull(i,O(state)) ***and*** *Frq[j]=1*
end while

As the packet is clustered, the rest of the message could be considered similar. Hence the length of the packet determines the size of the change. On the basis of above, packet format identification technology is used to carry out multiple sequence alignment for each group of similar messages, and obtain the corresponding packet structure, the constant and variable domains are separated. Then we can test the protocol by using the fuzzing method.

Protocol Fuzz Testing Using Genetic Algorithm

Genetic algorithm (GA) [7] is a kind of adaptive search method based on natural selection, which is a kind of adaptive search for optimal or near optimal solution. The basic concept of genetic algorithm is to simulate the process of evolution of natural systems, in particular those who follow the principle of Charles Darwin, in the survival of the fittest. Thus, they represent a random search for intelligent development in a defined search space in order to solve the problem.

Effective fuzz testing method, not only need to be able to trigger the vulnerability effectively, but also need to in accordance with the agreement of protocol, so as to pass the examination of the server. On the other hand, we observe that, in general, the difference between the test cases that can cause the vulnerability of the test cases and the normal message will not be great. In most cases, only a few characters are not the same. Therefore, we propose to improve the efficiency of fuzz testing by using the genetic algorithm and the design of fitness function to improve the similarity between the test message and normal message.

In general, the process of GA is as follows. First, genetic algorithm is needed to generate the initial population randomly. In the genetic algorithm, the population represents a set of solutions, and the chromosome represents a specific solution. Chromosomes are usually in binary form, and all the parameters are on the chromosome of encoding. Secondly, the genetic algorithm is designed according to the fitness function, and the corresponding values are calculated for each chromosome. The so-called fitness function is defined by the user, which is closely related to the search problem.

The evaluation criteria for each chromosome were calculated according to the calculated fitness function. Thirdly, genetic operators such as reproduction, crossover and mutation are used for the evolutionary process. Then the next generation of population is produced. Finally, set up a stopping criterion, that is, the end of the search process, and then leave the population that is the problem solution.

The core problem of genetic algorithm is to construct the fitness function. In the process of the application of fuzzy testing of the network protocol, we can set the similarity between the test message and the normal message as fitness function. In this way, the genetic algorithm will be able to filter out the most close to the normal message. In this method, Euclidean distance [10] is used to measure the similarity between the test message and the normal message. The author proposes a novel method shown in (1).

By the formula (1), if the test message msg_i is more far from the normal message msg_0, the fitness value is lower. To avoid the fitness value of msg_0 equals to zero, the fitness function is modified in (2):

Thus, the fitness value of normal message msg_0 equals to 1, other test massages msg_i is less than 1. The goal of our proposed algorithm is to keep the difference between the test message and the normal message in a certain range.

Typically, the GA first need to be binary coded. Let's set the aim function to [m2, m1], solving accuracy to ε, then the code length L can be calculated as (3).

By the reproduction operator, solutions with higher fitness values are reproduced with a higher probability. Here, we choose the roulette wheel selection method and the selection probability is calculated in (4).

Crossover means exchanging substrings from pairs of chromosomes to form new pairs of chromosomes. Mutation involves generating mutations of the chromosomes. Mutation prevents the search process from falling into local maxima, but a mutation rate that is too high may cause great fluctuation. So, the mutation rate is generally set to a low value.

Experiments and Evaluations

FTP protocol is a kind of open network protocol, which is widely used. The FTP protocol is used to establish a control connection and a data connection, and the FTP protocol also has two different modes: PASV and PORT. In our experiment, we choose the PORT method as the test mode. The experiment condition contains a PC with Intel(R) Core(TM) i5-3450 3.10 GHZ CPU and 4.0GB memory, using Windows XP SP3 and ServU 5.0 FTP server.

FTP is a protocol that can be viewed as a text type, except for 0x0D0A, which does not have any other non-printing characters. Part of achieved message structure of packet format identification by using the Algorithm 3 is shown in Table 1.

Table 1. Results of packet format identification

Packet Field	Field Attribute	Field Value
0	String - const	PASS
0-1	String - const	
1	String - variable	1Q2W3E
1-1	Binary - const	x0dx0a

For example, the message contains the string type keyword “PASS”, separator symbol “ ”, string type variable “1Q2W3E”, and the binary constant “x0dx0a”. Data statistics show that the current implementation of the proposed protocol reverse engineering technology can accurately identify the FTP protocol packet format information.

The ultimate goal of this paper is to exploit the vulnerability detection of network protocol, an automatic generation of Fuzzer is realized on the SPIKE [11] framework.FTP protocol is widely used in network data transmission, and its security is also studied. For example, the network has announced the existence of one remote buffer overflow vulnerabilities in ServU version 5.0. That is the MDTM command [13], the format is MDTM time+timezone remote-filename, when the parameter of timezone is a long string, it will lead to overflow. The fuzz testing script of MDTM command is generated by the proposed approach as shown in Table 2.

Table 2. The fuzz testing script of MDTM command

No.	*Content*
0	s_string("MDTM ");
1	s_string_variable("20151217112133");
2	s_string("+");
3	s_string_variable("AAAAAAAAAAA...AAAAAAAAAAAAAAAAA ");
4	s_string(" ");
5	s_string("/");
6	s_string_variable("test.txt");
7	s_binary("0d 0a");

Conclusion

The problem of generating more effective test messages is well motivated by introducing the genetic algorithm. In the process of packet format identification, an improved AC algorithm using DAT is proposed to improve the alignment effectiveness. In the process of test messages generation, Euclidean distance is introduced in the fitness function design to raise the similarity between test message and normal message. Experiments show that the proposed fuzz testing method could achieve effective results. The further step unfolding study may be the improvement of identification result of the packet format.

Acknowledgment

This work is supported by the National Natural Science Foundation of China under Grant No. 61302091.

References

[1] Glenford J. Myers, “The Art of Software Testing,” John Wiley and Sons, ISBN 0-471-04328-1 (1979)

[2] P. Oehlert, “Violating Assumptions with Fuzzing”, IEEE Security & Privacy, pp. 58-62 (2005)

[3] Guoqiang Shu, Yating Hsu, and David Lee, “Detecting Communication Protocol Security Flaws by Formal Fuzz Testing and Machine Learning”, IFIP International Federation for Information Processing 2008, LNCS 5048, pp. 299-304 (2008)

[4] A.T. Acree, T.A. Budd, R.A. Demillo, R.J. Lipton, and F.G.Saywared,“Mutation analysis,” Technical report, School of Information and Computer Science, Geogia Inst. of Technology, Atlanta, Ga., September (1979)

[5] T. Tsuchiya, T. Kikuno, “On fault classes and error detection capability of specification-based testing,” ACM Transactions on Software Engineering and Methodology, Vol.11, No.1, January, pp. 58-62 (2002)

[6] Jin-hua Li, Geng-xin Dai, Huan-huan Li, “Mutation Analysis for Testing Finite State Machines,” in Proceedings of the 2009 Second International Symposium on Electronic Commerce and Security (ISECS'09), Vol. 01 (2009)

[7] David E. Goldberg, John H. Holland, “Genetic Algorithms and Machine Learning,” Machine Learning, vol 3, pp. 95-99 (1988)

[8] Aho, A., Corasick, M.: Efficient string matching: An aid to bibliographic search. CACM 18(6), pp. 333–340 (1975)

[9] Aoe J. An efficient digital search algorithm by using a double-array structure. Software Engineering, IEEE Transactions on, 15(9), pp. 1066-1077 (1989)

[10] Per-Erik Danielsson. Euclidean distance mapping. Computer Graphics and Image Processing Volume 14, Issue 3, November, pp. 227-248 (1980)

[11] Dave Aitel. MSRPC Fuzzing with SPIKE 2006. Technology Report, Immunity Inc, August (2006)

[12] Win FTP Server 'LIST' FTP Operation remote buffer overflow vulnerability. URL: http://www.venustech.com.cn/NewsInfo/124/3557.Html

[13] URL: MDTM Operation remote buffer overflow vulnerability of Serv-U Sever. URL: http://www.jiangmin.com/News/jiangmin/networksafe/syssafty/200433111945.htm

[14] Ai-Fen Sui, Wen Tang, Jian Jun Hu, Ming Zhu Li Corporate Technology, Siemens Ltd. China Wangjing Zhonghuan Nanlu, Chao yang District, P.O.Box 8543, Beijing 100102, P.R. China. An Effective Fuzz Input Generation Method for Protocol Testing. IEEE Beijing Section.Proceedings of 2011 IEEE 13th International Conference on Communication Technology(ICCT 2011) IEEE Beijing Section (2011)

[15] Jinfu Chen,Huanhuan Wang,Dave Towey,Chengying Mao,Rubing Huang,Yongzhao Zhan. Worst-Input Mutation Approach to Web Services Vulnerability Testing Based on SOAP Messages. Tsinghua Science and Technology, pp. 429-441 (2014)

[16] Genetic Structure and Diversity of Parental Cultivars Involved in China Mainland Sugarcane Breeding Programs as Inferred from DNA Microsatellites. Journal of Integrative Agriculture, pp. 1794-1803 (2012)