Scholar Publishing Group
Distributed Processing System
https://doi.org/10.38007/DPS.2021.020401
ISSN 2790-0916 Vol. 2, Issue 4: 1-9

Distributed System Vulnerability Model Based on Reliability Theory

Patil Rajendra[*]

Royal Marsden NHS Fdn Trust, London SW3 6JJ, England

[*]*corresponding author*

Keywords: Reliability Theory, Distributed System, Vulnerability Model, Reliability Model

Abstract: One of the key issues of information security is software vulnerabilities in computer systems. Malicious attackers can exploit security vulnerabilities to gain privileges, access unauthorized system resources, and even change sensitive data. The main purpose of this paper is the vulnerability model of distributed systems based on reliability theory. This paper directly analyzes the machine instruction, formulates the corresponding standard model, and studies the corresponding model control method. This paper presents an analytical method that uses reliability theory to introduce the Fuzzing random test method. The method uses static analysis techniques to gather information about the structure of the system, its interfaces, and useful code regions, and then develops test problems for executing the useful code regions—for which this paper uses genetic algorithms to understand. It guides the generation of test data and overcomes the shortcomings of random fuzzing methods in terms of large and unpredictable test data sets. The method also analyzes files that can be executed directly without the source code of the program. Experiments show that the vulnerability model system constructed in this paper is more realistic. The parallel login response time of the system in this paper is about 0.12s, and the parallel use response time is about 0.2s.

1. Introduction

A distributed system is a network-based software and hardware system. A group of independent devices work together in the network to present a unified whole to users. The normal operation of the distribution system depends on the normal operation of each key switch in the network. Therefore, the stability and reliability of the network equipment provide a guarantee for the normal operation of the distribution system. If these key nodes fail in a certain area, if it cannot be found and repaired at the first time, it will bring irreparable losses to network communication [1-2].

In the research on the vulnerability model of distributed systems based on reliability theory, many scholars have studied it and achieved good results, for example : Lujano-Rojas J has proposed many software debugging techniques for monitoring and modeling to measure and Predict the reliability of software systems [3]. Gautam P used the opportunity graph to describe the attacker's opportunistic advancement process, represented the different paths to the attacker's goal as the attacker's different attack strategies, and used a dynamic algorithm to calculate the average attack cost of behavioral attacks. It is not difficult to see that this method It is "attack-centric", and the method of calculating the average attack cost is obtained empirically, without scientific basis, so the quantitative results cannot well demonstrate the security of the system [4].

This paper directly analyzes the machine instruction, formulates the corresponding standard model, and studies the corresponding model control method. This paper presents an analytical method that uses reliability theory to introduce the Fuzzing random test method. The approach uses static analysis techniques to gather information about the structure of the system, its interfaces, and useful code regions, and then develops test problems for executing the useful code regions—for which this paper uses genetic algorithms to understand. It guides the generation of test data and overcomes the shortcomings of random fuzzing methods in terms of large and unpredictable test data sets. The method also analyzes files that can be executed directly without the source code of the program.

2. Distributed System Vulnerability Model Based on Reliability Theory

2.1. Software Vulnerability Analysis Method

(1) Manual analysis

Human analysis is a method used by many security researchers today. For open source software, human analysts generally use source code reading tools such as source code anonymization to speed up source code retrieval and search. Modified strcpy in the system call library to further control the use of loops [5-6].

(2) Software Vulnerability Static Analysis Technology

This type of equipment has two important index parameters: Object, indicating that the weakness detected by static analysis is a real weakness, if it is not a weakness, it will produce a false ideal. Find the real weakness as much as possible. If static analysis does not find a real vulnerability, a false negative will be displayed[13-14]. In addition, the efficiency and scalability of such tools are also two important indicators, so they can be better applied to the inspection of large software applications or operating system kernels. The static analysis method can quickly complete the inspection of the source code, the inspector does not need to know the implementation method of the system, nor does it need to check the operation and operation status, which is very beneficial to the automatic inspection of the source code [7-8].

2.2. Vulnerability Comprehensive Analysis Model

The process of vulnerability analysis method is as follows: First, it is necessary to determine the security points that a secure IMS network must have and identify system components, and then analyze system vulnerabilities to analyze that assets can be subject to some threat behaviors, damage, and cause adverse consequences. The specific analysis steps are as follows[9-10]:

Step 1: Determine the security objectives that the IMS network should have.

Step 2: Identify all the components that make up the IMS network.

Step 3: Analyze IMS network vulnerabilities. Step 4: Classify threat behaviors to identify specific threat behaviors.

Step 5: Identify each network element and reference point in the IMS network.

Step 6: Identify unexpected results. Identify unintended consequences based on damage to network assets after threat actors exploit system vulnerabilities.

Step 7: Identify IMS network vulnerabilities. Based on the analysis of the previous steps, modify the characteristics of risk assets, vulnerabilities, threats, consequences, conditions, etc., to complete the vulnerability description [11-12].

2.3. Features of Vulnerability State Diagram

Compared with strengths and conflict diagrams, the use of vulnerability state diagrams to describe the vulnerability of distributed systems has the following advantages:

(1) Using system locations as nodes is more prevalent than attack-centric attack-and-dominant architectures, because the vulnerability of the system is not only due to attacks, but also users that may be caused by system failures. Operation and hardware damage[13-14].

(2) Compared with the collision image, it can save more position space.

A Vulnerability State Diagram (VSG) describes the many possible ways a system can reach an unsafe state. In this paper, the property that the system will not reach an unsafe state is denoted as AG (-safe). When the property is false, there is a safe state that can be reached from the initial state, the exact definition of safe depends on the specific application. The state diagram is shown in Figure 1 [15-16].

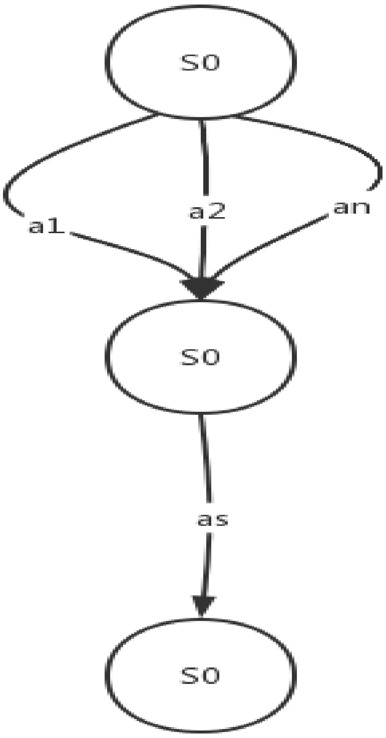

Figure 1. Vulnerability state diagram of an instance

Using the vulnerability analysis method based on reliability theory and according to the above vulnerability state diagram, the reliability function of the database system can be calculated as[17-18]:

$$R_s(c)=\sum_{i=1}^{5}\frac{\prod_{j\neq 1}\lambda_j}{\prod_{j\neq i}(\lambda_j-\lambda_i)}e^{-\lambda_i c} \tag{1}$$

The corresponding average attack cost is

$$E(C)=\int_0^{+\infty}R_s(c)dc=\sum_{i=1}^{5}\frac{1}{\lambda_i} \tag{2}$$

3. Research and Design Experiment of Distributed System Vulnerability Model Based on Reliability Theory

3.1. System Test

Unlike integration testing, system testing has more application scenarios, and not all of them are presented in the form of test cases.

This article only lists some representative system tests as follows.

(1) Basic feature service verification.

a) Included in the upgraded environment, to verify that each service is running properly.

b) The data created on the old version can be read and written normally on the new version.

c) Life test, test on the old version for a period of time, and continue to test on the new version for a period of time.

(2) Upgrade the framework test.

a) Validation of the upgrade process.

b) Upgrade verification with/without data nodes.

c) Upgrade verification with load.

d) Upgrade path test.

(3) Data migration framework test.

a) Use specific tools to verify that the results of the data migration are complete and accurate.

b) Data migration path test.

c) Active testing, data migration testing with/without data nodes.

d) Negative testing, extreme cases occur during testing, points where data migration is in progress, service stops, power outages, etc.

e) Passive testing, at the point where data is not being migrated, service stops, power outages, etc.

f) Negative testing, the entire cluster service stops, or directly crashes, etc.

(4) Performance test
a) Test of upgrade and data migration under empty data volume.
b) Test of upgrade and data migration under normal data volume.
c) Test of upgrade and data migration under large data volume.

Similar to the integration test, there are some tests in the system test that are very representative and enlightening. The following will take the test cases in the path test of data migration and the test cases in the upgrade and data migration under large data volume as examples., introduces the system test cases of this system[17-18].

3.2. Experimental Design

This paper mainly conducts two experimental tests for the system in this paper. The first is for the vulnerability model system in this paper, and for the research on the fault content of different modules, to analyze whether the model system in this paper is in line with reality. The second is to analyze the performance of the system in this paper, mainly through multiple parallel login and multiple parallel use to judge.

4. Experimental Analysis of Distributed System Vulnerability Model Research Based on Reliability Theory

4.1. Comparison of Fault Content

This paper compares the content of module faults for distributed systems, using the model in this paper and the traditional fault model respectively, and divides them into reused modules and new development modules for different modules, of which a1a2 is a reused module, a3a4 is a new development module, the specific fault content is as follows shown in Table 1.

Table 1. Failure difficulty comparison of different modules

	A1	A2	A3	A4
This article model	61.50	11.45	198.97	92.81
Conventional model	18.91	18.91	170.15	170.15

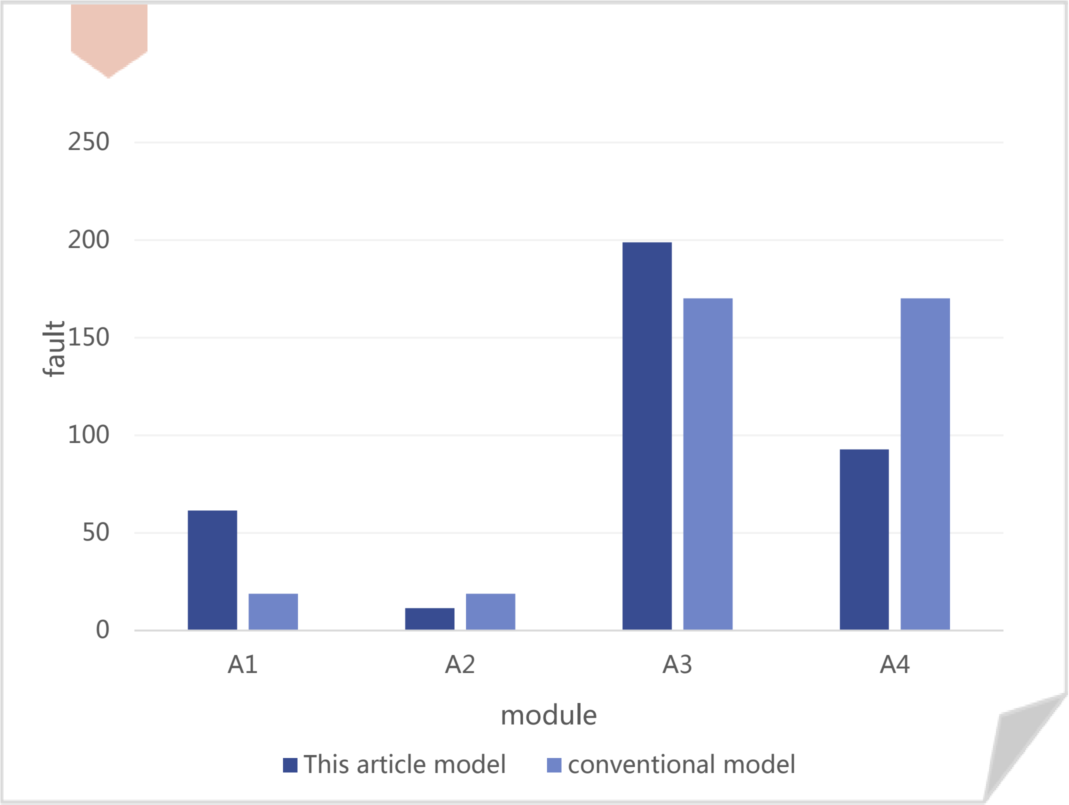

Figure 2. Fault content in both models

It can be seen from Figure 2 that the fault contents in different traditional modules are completely consistent, which is obviously not in line with the actual situation. Compared with this, the fault conditions of different modules in this model are different, so different response situations can be determined, which is more in line with reality.

4.2. Performance Test

Due to the large number of modules in the distributed system, the performance requirements of the system are relatively high. Therefore, this paper conducts concurrent operation and concurrent login tests for the vulnerability model system designed in this paper, and records the response time of the system. The data is shown in Table 2.

Table 2. System parallel login and use response time

	50	100	200	400	800
Parallel login	0.12	0.14	0.12	0.13	0.12
Parallel use	0.21	0.19	0.19	0.22	0.20

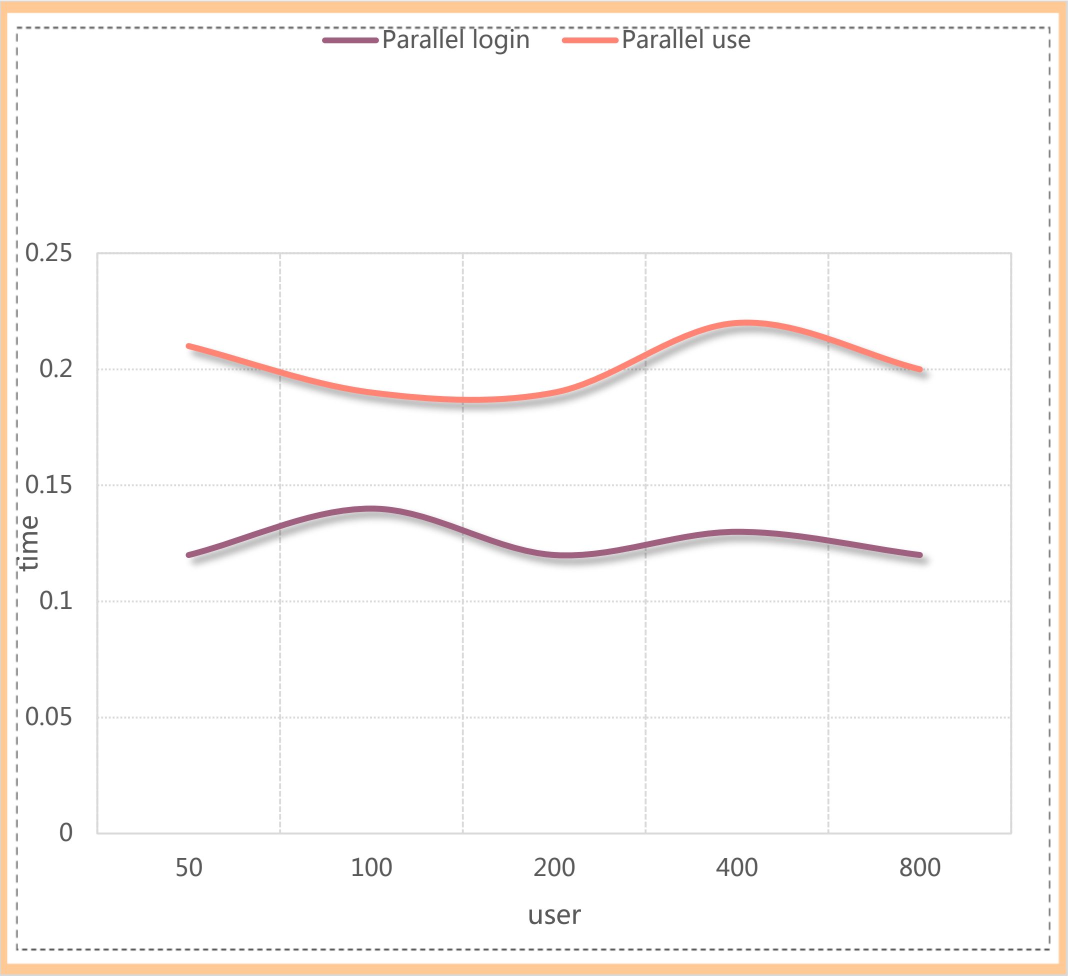

Figure 3. System performance test

As can be seen from Figure 3, the response time of the system in this paper is relatively stable. In the case of multiple people using it in parallel, the response time curve does not change much. The average parallel login time is about 0.12s, and the average time for parallel use About 0.2s, the response speed is fast, and the system performance is stable.

5. Conclusion

Through the research and study of related activities at home and abroad in recent years, this paper finds that there is no comprehensive model and evaluation method for distributed systems. In this paper, the vulnerability model of distribution system is proposed for the first time, and a vulnerability model is established for various factors that affect the security of distribution system. The state of the system indicates that the state space cannot be reduced, which is not in the interest of the person attacking the graph or method. It is more representative and general, while providing a good representation of the health of many different combinations of states in a distributed system. The model analysis method is used to show the complete process of exploiting system vulnerabilities in the form of vulnerability state diagrams when the system crashes or fails. Distributed systems offer theoretical opportunities to improve system security. The analysis of network problems further verifies the vulnerability model and related statistical methods proposed in this paper.

Funding

This article is not supported by any foundation.

Data Availability

Data sharing is not applicable to this article as no new data were created or analysed in this study.

Conflict of Interest

The author states that this article has no conflict of interest.

References

[1] Barbosa K, Fernandes M. Elderly vulnerability: concept development.. Revista brasileira de enfermagem, 2020, 73(suppl 3):e20190897.

[2] Ncube A, Tawodzera M. Communities' perceptions of health hazards induced by climate change in Mount Darwin district, Zimbabwe.. Jamba (Potchefstroom, South Africa), 2019, 11(1):748. https://doi.org/10.4102/jamba.v11i1.748

[3] Zang T, Gao S, Liu B, et al. Integrated fault propagation model based vulnerability assessment of the electrical cyber-physical system under cyber attacks. Reliability Engineering & System Safety, 2019, 189(SEP.):232-241.

[4] Lujano-Rojas J, José Yusta, José Domínguez-Navarro. Mitigating Energy System Vulnerability by Implementing a Microgrid with a Distributed Management Algorithm. Energies, 2019, 12(4):616. https://doi.org/10.3390/en12040616

[5] Gautam P, Piya P, Karki R. Development and Integration of Momentary Event Models in Active Distribution System Reliability Assessment. IEEE Transactions on Power Systems, 2019, PP(99):1-1.

[6] Xin, Gai, Chunyu, et al. Type the National Vulnerability Evaluation System Research Based on Fuzzy Comprehensive Evaluation. IOP Conference Series: Materials Science and Engineering, 2019, 490(6):62039-62039.

[7] Taylor R, Mitchell G, Andrade J, et al. Expression of Collagen Types I, II, IX, and X in the Mineralizing Turkey Gastrocnemius Tendon.. Anatomical record (Hoboken, N.J. : 2007), 2020, 303(6):1664-1669. https://doi.org/10.1002/ar.24091

[8] Arslan S S. A Reliability Model for Dependent and Distributed MDS Disk Array Units. IEEE Transactions on Reliability, 2019, 68(1):133-148.

[9] Zareie S, Hazbavi Z, Mostafazadeh R, et al. Vulnerability Comparison of Samian Sub-watersheds based on Climate Change Components. Physical Geography, 2021, 52(2):217-236.

[10] Karngala A K, Singh C. Reliability Assessment Framework for the Distribution System Including Distributed Energy Resources. IEEE Transactions on Sustainable Energy, 2021,

PP(99):1-1.

[11] Zhang Y, Wang Y, Wang L, et al. An Extended Object-Oriented Petri Net Model for Vulnerability Evaluation of Communication-Based Train Control System. Symmetry, 2020, 12(9):1474. https://doi.org/10.3390/sym12091474

[12] Choi B, Sohn J Y, Yoon S W, et al. Secure Clustered Distributed Storage Against Eavesdropping. IEEE Transactions on Information Theory, 2019, 65(11):7646-7668.

[13] Kalla H, Farid A. Task allocation based modified bacterial foraging algorithm for maximising reliability of a heterogeneous distributed system. International Journal of Communication Networks and Distributed Systems, 2019, 23(1):1.

[14] Jagannathan J, Parvees M. Vulnerability Recognition and Resurgence in Network based on Prediction Model and Cognitive based Elucidation. Journal of Physics: Conference Series, 2021, 2070(1):012122-.

[15] Ciminello M. Reliability of structural health monitoring system based on distributed fiber optic and autocorrelation of the first order derivative of strain. IEEE Sensors Journal, 2019, PP(99):1-1.

[16] Ustun, Ayyubi. Automated Network Topology Extraction Based on Graph Theory for Distributed Microgrid Protection in Dynamic Power Systems. Electronics, 2019, 8(6):655.

[17] Fichera A, Marrasso E, Sasso M, et al. Energy, Environmental and Economic Performance of an Urban Community Hybrid Distributed Energy System. Energies, 2020, 13(10):2545. https://doi.org/10.3390/en13102545

[18] Bogdanovich V I, Giorbelidze M G. Ion-Plasma Treated Parts Quality Improvement Analysis Based on the Reliability Theory Criteria. IOP Conference Series Materials Science and Engineering, 2021, 1118(1):012004. https://doi.org/10.1088/1757-899X/1118/1/012004