Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org
© 2023 SCPE. Volume 24, Issues 4, pp. 651–661, DOI 10.12694/scpe.v24i4.2067

GENETIC ALGORITHM SERVICE VULNERABILITY MINING TECHNOLOGY OF ANDROID SYSTEM

XIAOYAN GUO [*]AND YANFENG SUN [†]

Abstract. The inland ships energy efficiency is significantly influenced by navigational environment, including speed and direction of wind and water depth. In order to solve the problem of low efficiency of conventional fuzzy test mining, a research method of Android system Service Vulnerability mining technology based on Genetic Algorithm (GA) is proposed. An efficient genetic selection operator model based on probability ranking and combination is also presented to improve the sample coverage and fuzzy test efficiency. Through the framework testing on different systems of mobile phones, multiple system service vulnerabilities are excavated. The execution results guide the generation of test cases, which reduces the proportion of invalid parameters in the test process to improve the efficiency of fuzzy testing. It is observe that the fuzzy test based on GA is much better than the conventional fuzzy test method in the vulnerability mining of system services, and has certain effectiveness and superiority. In addition, after using the two-point crossover algorithm to recombine the gene strings of two individuals, the phenotype of the newly generated individual gene string may become meaningless. It is observed that the selection algorithm factor has a very low p-value, while the ANOVA test confirms at least two groups that have statistically-significant difference.

Key words: System service; Vulnerability mining; Binder Genetic algorithm; Android system; Fuzzy test

1. Introduction. With the rapid growth of mobile Internet technology, mobile devices have greatly improved people's life and entertainment. At present, the mainstream operating systems in the market are mainly Android IOS and Windowphoneo. Android is an intelligent operating system released by Google [1]. Android has a large number of applications and developers. Due to the low threshold of Google's application developers, they can easily get the official developer signature of Android applications from Google, and Google has not adopted a strict security review system, resulting in increasingly serious security problems for Android applications. Android system services play an important role in the whole Android system. While Android system services provide functions for mobile phone users, there are also some threats and risks. These security vulnerabilities can cause serious consequences [2]. For example, if an application obtains the SMS system service in the system service, it will get the user's SMS message, and the user's privacy is likely to be exposed. In addition, if special external data is used in the process of using external system services, Android system services may crash, and even serious consequences such as remote code execution and memory damage may occur. Therefore, the security of Android system services needs our attention. These system services are provided by systems or system applications running in the background. These system services encapsulate the basic functions of Android system, and they open the call interface to ordinary applications [3]. These basic functions include Bluetooth, call and so on.

The system service code occupies the main part of the Android framework. In this sense, this is also an important difference between Android system and traditional desktop PC operating system, so the traditional vulnerability mining tools for desktop operating system are not applicable to Android operating system. In the test process, according to the feedback of the results, guide the genetic algorithm to continuously mutate the test parameters, and propose an efficient genetic selection operator model based on probability ranking and combination, so as to improve the sample coverage and fuzzy test efficiency [4]. Through the testing of the framework on different system versions of mobile phones, multiple system service vulnerabilities are excavated.

According to the query of CNVD (China national vulnerability database) and other well-known vulnerability submission platforms, the previously submitted an droid vulnerabilities are concentrated in the application layer, mostly in the types of component exposure, information disclosure, secondary repackaging, privilege promotion

[*]Yellow River Conservancy Technical Institute, Kaifeng, Henan 475003, China (xiaoyanguo10098@gmail.com).
[†] China Radio and television Henan Network Co., Ltd. Kaifeng branch, Kaifeng, Henan 475003, China.

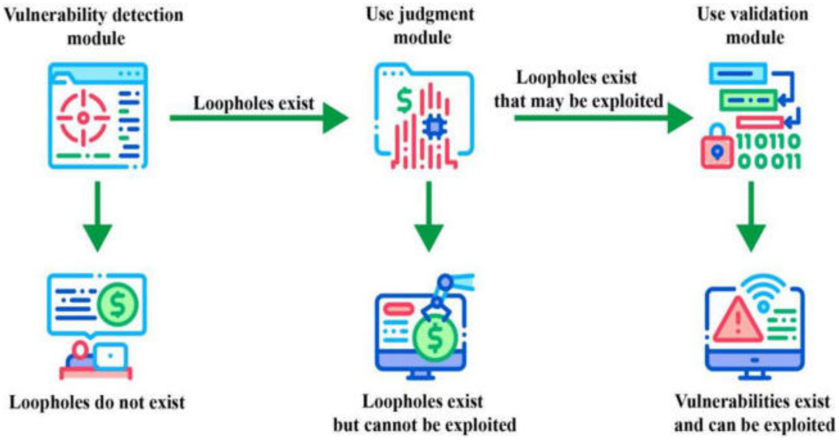
Vulnerability detection module
Use judgment module
Use validation module
Loopholes exist
Loopholes exist that may be exploited
Loopholes do not exist
Loopholes exist but cannot be exploited
Vulnerabilities exist and can be exploited

Fig. 1.1: An Android system Service Vulnerability verification method, device and process [7]

and so on, but there is little research and Exploration on the system service level. As the core process running at the bottom of the system, Android system services may lead to privacy disclosure once they are obtained by malicious programs.

In addition, if the service receives and uses the incoming illegal parameters during operation, it may cause unknown results such as system restart, denial of service and even memory damage. At present, the relevant research mainly focuses on the mining of Linux or windows driver vulnerabilities. The driver runs in the system kernel state, and it is difficult to analyze the interaction between device driver and kernel by static analysis; Dynamic analysis needs to run relevant hardware and provide unconventional input, which makes it more difficult to mine driver vulnerabilities. The fuzzy testing technology based on genetic algorithm maps the input data to the gene space through coding, obtains the path conditions by using the white box test method, calculates the fitness of the test cases based on the path coverage, and obtains the test cases that meet the conditions through genetic algorithm [5]. Without the source code, the execution path conditions of the program cannot be known, so this method is inadequate when the source code of Android driver cannot be obtained. How to make the vulnerability mining system generate more effective test cases without source code and dig out more unknown vulnerabilities in less time is the research difficulty of Android driven vulnerability mining technology [6, 7]. Figure 1.1 shows an Android system Service Vulnerability verification method, device and process.

In Android system, application is composed of four components, among which activity and service are two important components. Components may be in the same process or in different processes. When activities or services are in different processes, the cross process communication between them is realized through binder. Android's cross process communication mechanism is based on binder mechanism, not any of the mechanisms mentioned above. Binder cross process communication mechanism is not a communication mechanism created by Android system from 0 to 1.

1.1. Contribution.

1. This paper focuses on a Service Vulnerability mining technology of Android system based on genetic algorithm, which combines feedback mechanism and data optimization through genetic algorithm.
2. In order to solve the problem of low efficiency of conventional fuzzy test mining Android system service vulnerabilities, a research method of Android system Service Vulnerability mining technology based on genetic algorithm is proposed.
3. This paper proposes a research on Service Vulnerability mining technology of Android system based on genetic algorithm. Combining genetic algorithm with fuzzy testing technology, the variation of

parameters is guided by fitness function to ensure the diversity of parameters, and the corresponding combined variation operation is carried out for different parameters according to the variation priority table of data type.

4. This method reduces the influence of combination explosion on parameter genetic variation to a certain extent, and improves the coverage of test cases.

The rest of the paper is organized as follows. The related work is reviewed and discussed in Section 2 followed by the section 3 which explains the research methodology utilized in this work. Section 4 gives the result analysis and Section 5 concludes the paper.

2. Literature Review. At present, many scholars have proposed some new vulnerability mining methods: vulnerability mining methods based on symbolic execution. These methods have the problem of low degree of automation [8].Authors used symbolic execution method to study driver vulnerability mining, and used hardware virtualization technology to solve the problem of requiring specific equipment to test driver vulnerabilities. However, there is a path explosion problem in symbol execution, and the mining type is single. What's more, there is no mature Android full symbolic system, and this technology can not be effectively transplanted to Android driven vulnerability mining. In terms of vulnerability mining, fuzzy testing is one of the fast and simple mining tools, but the traditional fuzzy testing has the shortcomings of lack of understanding of the target program, random and blind testing. How to overcome these shortcomings has always been the focus of fuzzy testing research [9].

Palazzolo, N. and others used the fuzzy test method in the process of Android system Service Vulnerability mining, but the test data type was single and did not construct the complex parameter types in the communication process, which led to the incomplete coverage of parameter use cases to some extent [10]. Zhang, S. Q. and others used fuzzy testing to mine vulnerabilities in system services, but did not reasonably control the variation of multidimensional parameters in system services, which may lead to problems such as combination explosion. How to solve the problems of low use case coverage and multi-dimensional parameter variation in the process of fuzzy testing has also been a loophole [11].

Peng, D. and others proposed a method to test Android system based on binder mechanism adopted by Android system. This method uses a third-party application to destroy the Android system kernel memory by passing abnormal numbers into the system service, so as to obtain the permission to manipulate the kernel space. Then import the shared library into this space, bypass a series of Android system security mechanisms such as SELinux, and achieve the purpose of improving application permissions. This method takes advantage of the negligence of Android system service in parameter checking, tests on Android system, successfully obtains the permission of Android system server through media player system service, and rebounds shell successfully [12].

Martowibowo, S. Y., and others proposed another test method. They analyze the special input verification of Android system services at the system framework layer to find input verification vulnerabilities. They first analyzed the exceptions thrown by the service interface when the Android system service performs input verification on different inputs, looked for the exploitable interface, analyzed the interface, and finally designed the application to scan the similar interfaces that may exist in all Android devices. For the Android system, the vulnerability detection method was used to detect the vulnerability of [90] of the 13 service parameters of the Android system, and finally sent to the Android system. Zhang, B. and others believe that the smooth operation of Android terminal is inseparable from the support of system services. For example, SMS manager is required for receiving and sending SMS, and win dowmanager is required for opening and closing windows. Various service managers provide access interfaces to the bottom layer, which facilitates the call of upper applications [14].

Meng, Y. and others first carried out systematic research on the security of Android customized content and invented the attached system. By dynamically analyzing the relevant files of sensitive operation of the device and comparing them with the relevant files of the native system, we can see the difference between its security protection and the native system. After several platform tests, it is found that the customized systems of different manufacturers have different degrees of security problems. Although some customization related driver vulnerabilities have been tested, there is no specific research on Android driver vulnerability mining [15].

Author evaluates the K-Nearest Neighbor (K-NN) supervised algorithm performance in determining students' learning styles [16]. Edeh Author presented the entrepreneurship education across the globe. Author examines cyber-security awareness among undergraduate students from crime on the cyber-space [17].

Author in this paper discuss the problem of firmware vulnerability mining and the traditional method of vulnerability mining research based on fuzzing test which is not efficient. A noval mining vulnerabilities method in industrial firmware is proposed. This method constructs test cases for the variables for triggering the vulnerabilities [18]. The presented method can reduce about 23% of test cases and can effectively improve test efficiency.

Author in this paper presents a novel solution for detecting rare malware programs and provides the scarcity of datasets for modeling these malware [19]. Author's analysis system includes an internet simulator and a human emulator to successfully execute them and prevent system halting. An objective function is used to optimize the vital indicators and tracking rate with a linear time complexity. Real-world malware samples were used for the performance evaluation and comprehensive scenarios were involved to evaluate the proposed strategy performance. The results demonstrate the improvement in detection accuracy and the results also demonstrated an enhancement in true positive rate for the presented deep-learning algorithm. Author in this work presented an Android malware detection framework GA-Stacking which employs stacking to compose five different base classifiers [20]. The GA is applied to optimize the hyper parameters of the framework and experiments show that stacking could improve malware detection accuracy as compared with single classifier. The presented technique achieves accuracies of 98.43% and 98.66% on CIC-And Mal and CIC MalDroid datasets, which shows the efficiency and feasibility of the presented method. Author in this paper presented a machine learning-based detection approach by utilizing hybrid analysis-based particle swarm optimization (PSO) and an adaptive genetic algorithm (AGA) [21]. The feature selection is performed by applying PSO in the dataset. Further, the XGBoost and random forest (RF) machine learning classifiers performance is optimized utilizing the AGA. With the random forest classifier, an accuracy of 98.72% and F-score were achieved. Our results present that the PSO application and an AGA greatly increases the classification performance of the information obtained from the hybrid analysis. Author in this paper details GA-based feature selection which helps Android malware detection [22]. The machine learning algorithms with GA-based feature selection for 1104 static features included in the Andro-AutoPsy dataset is used. The comparative analysis is done by the author and showed that the GA performed better than the information gain-based method, which is generally used as a feature selection method. Moreover, machine learning using the presented GA-based feature selection has an absolute advantage in terms of time as compared to ML without feature selection. Further, it is useful to apply GA-based feature selection to improve malware detection performance.

2.1. Research Gaps. The inland ships energy efficiency is significantly influenced by navigational environment, including speed and direction of wind and water depth. The inland navigational environment complexity makes it difficult for determining the optimal speeds under different environmental conditions for the best energy effectiveness. In order to solve the problem of low efficiency of conventional fuzzy test mining, an efficient research method of is needed.

3. Research Methodology.

3.1. Research on Android vulnerability mining. Security vulnerabilities refer to some problems and defects in some security schemes, which are embodied in the detailed implementation of hardware, software and protocols. The damage to the system can be completed by the attacker without authorization. These security vulnerabilities are usually some vulnerable people left by carelessness. These entries may exist in computer hardware and computer components, applications or some online resources. Binder is an inter process communication mechanism in Android system. In Android system, generally speaking, different applications run in different processes. For the same application, different system components can also run in different processes [23]. When a process wants to provide services for other processes, it needs to provide services through inter process communication. In Android system, application is composed of four components, among which activity and service are two important components. Components may be in the same process or in different processes. When activities or services are in different processes, the cross process communication between them is realized through binder. Android's cross process communication mechanism is based on binder mechanism. Binder cross process communication mechanism is not a communication mechanism created by Android system from 0 to 1. It is developed on the basis of open binder project. As an IPC mechanism, binder's architecture is a distributed

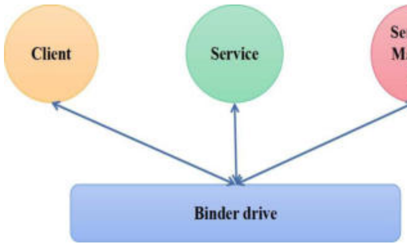

Fig. 3.1: Binder architecture

component architecture, which can provide remote calling functions. Binder mechanism is composed of four parts, including user space and kernel space. The four components are client, server, binder driver and service manager. The client, server and service manager run in user space and the driver runs in kernel space, as shown in Figure 3.1. Binder mechanism can effectively combine the above four parts. Binder driver component is the core member, client server interaction in user space is completed through driver, and service manager is responsible for auxiliary management of system services. The interaction between client and server is completed through the underlying driver. The Android system has helped us realize the driver and service manager, and the rest of the client and server need to be implemented by Android developers themselves [24].

3.2. Genetic Algorithm.

3.2.1. Basic flow of genetic algorithm. The search of genetic algorithm begins with a potential solution set of the problem, which is also the parameter space of the actual problem. For different problems, there are many methods to generate the parameter space of practical problems. In fuzzy testing, random algorithm is usually used to generate the parameter space of practical problems. Gene code each input parameter in the parameter space according to the predetermined coding rules to generate the initial population, and then perform the following iterative process until the predetermined iterative threshold is reached or the required optimal solution has been found: The fitness value of each individual is calculated through the fitness function, and then according to the fitness value of the individual, the better individual is selected according to a certain selection algorithm to be inherited to the next generation [25]. The selected individuals are paired in pairs, and the paired two individuals are cross operated according to a certain cross probability to exchange some genes. Then, an individual is randomly selected from these individuals to randomly change the value of one or some loci in the individual gene coding string with a certain mutation probability. Usually, the value of the locus is replaced by alleles. In the iterative process, individuals with poor fitness can be appropriately eliminated according to specific conditions. The basic flow of genetic algorithm is shown in Figure 3.2. After the iteration of genetic algorithm, the individual encoded by gene needs to be decoded to obtain the optimized solution of the actual problem.

3.2.2. Gene Coding. When using genetic algorithm to solve specific problems, we first need to solve the problem of parameter coding and decoding in the actual problem solution space. The parameters in the solution space of practical problems can only be processed by genetic algorithm after they are transformed into individuals (also known as chromosomes) in the genetic space represented by gene strings using certain coding rules [26]. This process is called the coding process from individual phenotype to gene in genetic algorithm. On the contrary, the conversion process from individual genotype to phenotype is called decoding process. The conversion process is shown in Figure 3.3.

The common gene coding methods of genetic algorithm include binary coding, gray coding and floating point coding. Binary coding is the most commonly used coding method. The gene encoded by it is a binary string composed of 0 or 1. Using binary coding, encoding and decoding operations and genetic operations such as mutation and crossover are easy to implement, and also comply with the coding principle of minimum character set. However, due to the randomness of binary coding, its local search ability is poor. After mutation, the individual's phenotype changes greatly and is easy to be far away from the optimal solution. Gray code

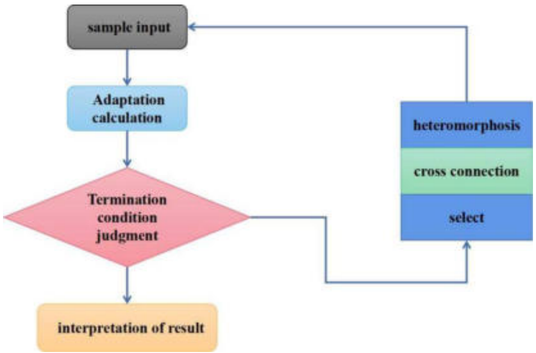

Fig. 3.2: Basic flow chart of genetic algorithm

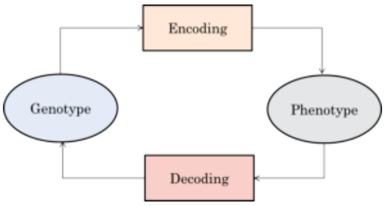

Fig. 3.3: Schematic diagram of genetic algorithm encoding and decoding

is an improvement of binary code. Only one code point is different between the codes corresponding to two adjacent integers encoded by gray code, and the other code points are exactly the same. Assuming that there is binary code , and its corresponding gray code is , the binary code can be converted into the corresponding gray code using formulas (1) and (2), and the gray code can be converted into the corresponding binary code using formulas (3) and (4), where i=1,2......n-1. For the individual encoded by gray code, the phenotype before mutation and the phenotype after mutation are continuous, so it has better local search ability. At the same time, because gray code is still a binary code in essence, gray code still has the advantages of easy implementation of genetic operations such as crossover and mutation, and conforms to the coding principle of minimum character set. Floating point number coding method is to represent each gene of chromosome with a real number. The coding length of individual depends on the number of decision variables. Floating point coding is often used to solve the continuous function optimization problem with multi-dimensional and high precision requirements. When dealing with individuals with such problems, binary coding will produce gene coding individuals with large length, which will lead to a sharp increase in search space. Floating point coding method has the advantages of improving the accuracy of genetic algorithm, improving the computational complexity of genetic algorithm, and facilitating genetic search in a large space.

$$g_n = b_n \tag{3.1}$$

$$g_i = b_i * \oplus * b_{i+1} \tag{3.2}$$

$$b_n = g_n \tag{3.3}$$

$$b_i = b_{i+1} * \oplus * b_i \tag{3.4}$$

3.3. Genetic manipulation. In order to make the individuals in the population approach the optimal solution in the process of generation by generation evolution, in the process of genetic iteration, it is necessary to perform certain genetic operations on the individuals according to the fitness value. There are three common genetic operations: selection, crossover and variation. These three genetic operations are also called genetic operators in genetic algorithms.

3.3.1. Selection operator. Selection, also known as replication, is the first step of genetic operation. It eliminates the fittest according to the fitness value of each individual calculated by the fitness function: the higher the probability that the individual with higher fitness will be inherited to the next generation, and the lower the probability that the individual with lower fitness will be inherited to the next generation. In this way, the individuals in the population can continuously approach the optimal solution. The main function of selection operation is to avoid losing useful genetic information in genetic iteration and improve global convergence and computational efficiency. Therefore, the choice of algorithm design will affect the final result of the algorithm. Common selection operators include roulette selection, random competition selection, best reservation selection and random selection without playback.

3.3.2. Crossover operator. The crossover operation of genetic algorithm simulates the process of two homologous chromosomes forming new chromosomes through mating and recombination. Crossover is also called recombination. Its basic operation is to select two parent individuals according to a certain probability and form two new offspring individuals by exchanging some gene strings of the two parent individuals. The two offspring individuals inherit some genes of the parent individuals. Through the exchange of gene strings, new chromosomes are generated, which improves the diversity of the population. The key to the design of crossover operator of genetic algorithm mainly lies in: 1. The determination of intersection position; 2. How to carry out partial gene exchange. Crossover operator is the main method of generating new individuals in genetic algorithm, which plays an important role in the correct implementation of genetic algorithm. Before the crossover operation, the individuals in the population need to be paired in pairs. The more common pairing strategy is random pairing, that is, randomly assign n individuals in the population to [n / 2] pairing groups. The crossover operation is completed on two individuals in each paired group.

3.3.3. Mutation operator. In genetic algorithm, the mutation operation is completed by replacing some gene values in the individual gene coding string with their alleles. For example, in binary coding, change ”0” to ”1” or ”1” to ”0”. Using mutation operation in genetic algorithm can avoid the loss of some information caused by selection and crossover operation. Crossover operation is the main method of genetic algorithm to generate new individuals, which determines the global search ability of genetic algorithm. Although mutation operation is only an auxiliary method to generate new individuals in genetic algorithm, it can avoid premature phenomenon and improve the local search ability of genetic algorithm, which determines the local search ability of genetic algorithm. Cross operation and genetic operation cooperate with each other, so that genetic algorithm can obtain better search performance when solving optimization problems. There are two main purposes of using mutation operator in genetic algorithm: 1. Improve the local search ability of genetic algorithm; 2. Maintain the diversity of the population and prevent the occurrence of precocity.

4. Results and Discussion. For the initial test data set, the traditional simple genetic algorithm and the optimized genetic algorithm are used for optimization. The primary optimization result of a population is shown in Figure 4.1. In Figure 4.1, the horizontal axis represents the population evolution algebra, and the vertical axis represents the average fitness value of individuals in each generation of population.

As can be seen from Figure 4.1, although the traditional simple genetic algorithm converges rapidly and the population has achieved high fitness value in the early stage of evolution, the average fitness value after population convergence is lower than that obtained by the genetic algorithm optimized in this paper. In order to verify the performance of ASFuzzer in actual vulnerability mining, this paper uses ASFuzzer to mine vulnerabilities in practical applications in the Internet, and compares the mining results with those of WFBGA and spike. This experiment digs the official website of a university and the official website of an organ. The comparison of vulnerability mining results of a university is shown in Figure 4.2 .

From the analysis of experimental results, the number of vulnerabilities mined by ASFuzzer framework is more than the number of system service vulnerabilities mined by conventional fuzzy test under the same system version, and some results have been achieved in the test on the customized system of third-party manufacturers. Genetic algorithm mutation generates highly diversified test cases, which greatly improves the possibility of triggering vulnerabilities. By analyzing the interface and its parameters that generate exceptions, 15 of the 20 vulnerabilities are caused by the variation of multi parameter combination, that is, exceptions will be triggered only when specific data is filled.The experimental results show that the fuzzy testing based

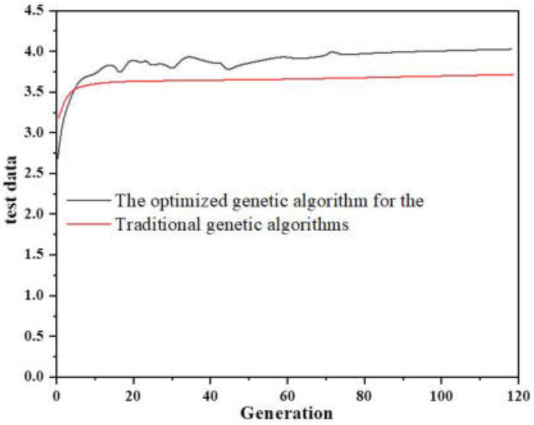

Fig. 4.1: Genetic optimization results

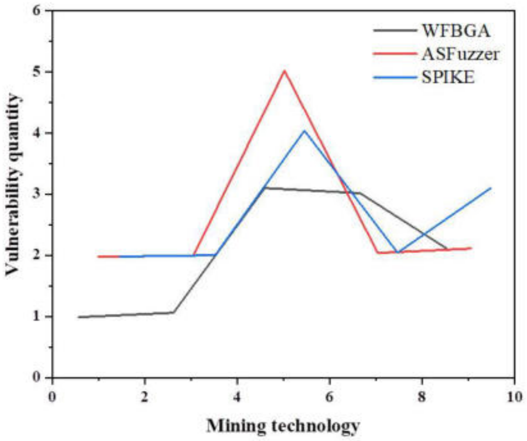

Fig. 4.2: Comparison of vulnerability mining results

on genetic algorithm is much better than the conventional fuzzy testing methods in the vulnerability mining of system services, and has certain efficiency and superiority. In the mining strategy of genetic algorithm, the individual in the current population is selected randomly in the selection operator stage. As individuals about to enter the next round, random selection may miss excellent individuals, mislead the direction of testing and affect the mining efficiency. This paper makes full use of the variability range of a single parameter and the quantitative relationship of parameters, and combines the arrangement and combination method to guide the selection operation in the process of genetic algorithm, so as to ensure the timely input of excellent individuals, so as to guide the development of mining testing in the direction of high efficiency.

4.1. Comparison among evolutionary configurations. To determine the performance of the configuration in the evolutionary, every combination of the 3 fitness functions are considered and applied these combinations on each of the system services, for evolutionary fuzzing campaigns. Number of covered basic

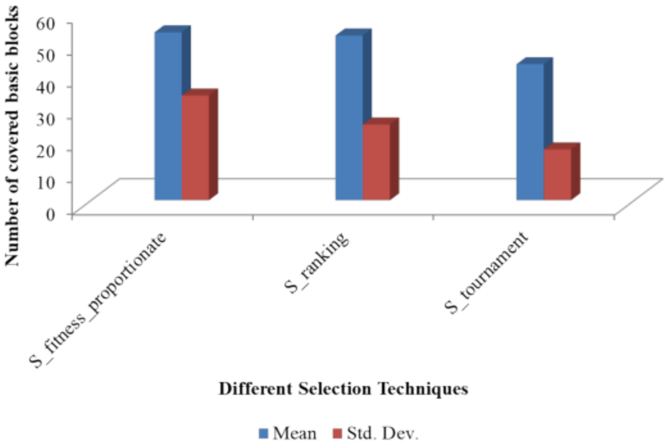

Fig. 4.3: Number of covered basic blocks, with respect to different selection algorithms

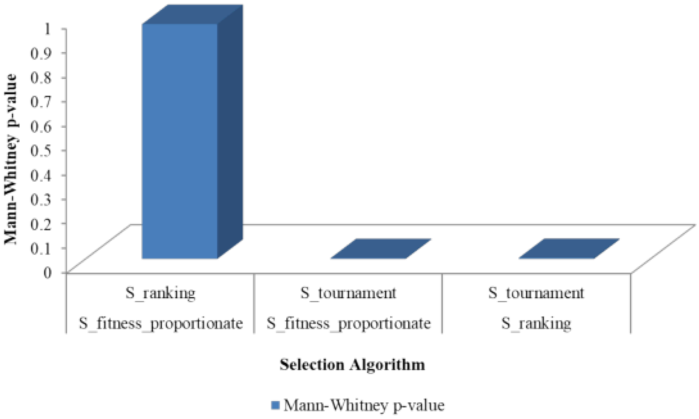

Fig. 4.4: Pairwise analysis of selection algorithms

blocks w.r.t different selection algorithms are shown in Figure 4.3.

Figure 4.3 show the number of basic blocks grouped by the fitness functions and the selection algorithms. The selection algorithm factor has a very low p-value: the selection algorithm has a significant effect, however, while the ANOVA test confirms at least two groups that have statistically-significant difference, it does not point out the specific group to detect the best selection algorithm for fuzzing. The pairwise tests and measure effect size are performed in each case of the results are in Figure 4.4.

The null hypotheses are considered that fitness function choices and the selection algorithm have no effect on the testing, and the non-parametric Kruskal-Wallis tests are performed since data are not distributed normally. The selection algorithm factor has the choice of a selection algorithm having significant effect. The best selection algorithm detection is performed for evolutionary fuzzing. The pairwise tests are performed and the effect size is measured.

5. Conclusion. This paper introduces a research on Service Vulnerability mining technology of Android system based on genetic algorithm, which combines feedback mechanism and data optimization through genetic algorithm. The execution results are used to guide the generation of test cases, which reduces the proportion of invalid parameters in the test process and improves the efficiency of fuzzy testing. After being applied to the Android driver vulnerability mining practice, several unpublished driver vulnerabilities are found, such as binder, camera and other exploitable denial of service attacks. Through the continuous optimization of use cases by genetic algorithm, it can generate more comprehensive and diversified use cases to test the objective function, and then trigger the vulnerability. By analyzing the interface and its parameters that generate exceptions, 15 of the 20 vulnerabilities are caused by the variation of multi parameter combination, that is, exceptions will be triggered only when specific data is filled. The test results on different systems show that ASFuzzer test framework can effectively mine the vulnerabilities of system services, and then find some potential security problems; At the same time, it also shows the efficiency superiority of this scheme compared with the conventional fuzzy test mining method. The current research work still has some shortcomings as follows: in the multiple execution process of genetic optimization algorithm, it still converges to the local optimal solution. This may be due to the insufficient continuity of gray code encoding for string data. In addition, after using the two-point crossover algorithm to recombine the gene strings of two individuals, the phenotype of the newly generated individual gene string may become meaningless. These may lead to poor optimization results of genetic algorithm. In future, this work can be continuing on a project to determine the mobile technology influence on the menace of cybercrimes in Nigeria.The selection algorithm factor has the choice of a selection algorithm having significant effect. The best selection algorithm detection is performed for evolutionary fuzzing. The pairwise tests are performed and the effect size is measured. The future research will focus on the present energy efficiency optimization to provide the foundation for energy efficiency.

REFERENCES

[1] Guo-Hong, S., Application development research based on android platform, In 2014 7th International Conference on Intelligent Computation Technology and Automation, IEEE, 579-582, 2014.

[2] Arabo, A., & Pranggono, B., Mobile malware and smart device security: Trends, challenges and solutions, IAIC Transactions on Sustainable Digital Innovation, 1(2), 526-531, 2013.

[3] Alam, T, Cloud Computing and its role in the Information Technology, IAIC Transactions on Sustainable Digital Innovation, 1(2), 108-115, 2020.

[4] Atabati, A., Alizadeh, M., Schuh, H., & Tsai, L. C., Ionospheric scintillation prediction on s4 and roti parameters using artificial neural network and genetic algorithm, Remote Sensing, 13(11), 2092-2096, 2021.

[5] Ponticelli, G. S., Lambiase, F., Leone, C., & Genna, S., Combined fuzzy and genetic algorithm for the optimisation of hybrid composite-polymer joints obtained by two-step laser joining process, Materials, 13(2), 283-288, 2020.

[6] [4] Duan, Li, Ruizheng, Shi, Ni, & Yao, et al., Real-time patient-specific ecg arrhythmia detection by quantum genetic algorithm of least squares twin svm, Journal of Beijing Institute of Technology, v.29(01), 32-40, 2020.

[7] Hraiech, S. E., Chebbi, A. H., Affi, Z., & Romdhane, L., Genetic algorithm coupled with the krawczyk method for multi-objective design parameters optimization of the 3-upu manipulator, Robotica, 38(6), 1138-1154, 2020.

[8] Zameni, M., Rezaei, A., & Farzinvash, L., Two-phase node deployment for target coverage in rechargeable wsns using genetic algorithm and integer linear programming, The Journal of Supercomputing, 77(12), 1-29, 2021.

[9] Atabati, A., Alizadeh, M., Schuh, H., & Tsai, L. C., Ionospheric scintillation prediction on s4 and roti parameters using artificial neural network and genetic algorithm, Remote Sensing, 13(11), 2092-2096, 2021.

[10] Palazzolo, N., Peres, D. J., Bordoni, M., Meisina, C., & Cancelliere, A., Improving spatial landslide prediction with 3d slope stability analysis and genetic algorithm optimization: application to the oltrepò pavese,Water, 13(6), 801-806, 2021.

[11] Zhang, S. Q., & Zhang, Y., Harmonic detection method based on permutation entropy and variational modal decomposition optimized by genetic algorithm, Review of Scientific Instruments, 92(2), 025118-025122, 2021.

[12] Peng, D., Tan, G., Fang, K., Chen, L., & Zhang, Y., Multiobjective optimization of an off-road vehicle suspension parameter through a genetic algorithm based on the particle swarm optimization, Mathematical Problems in Engineering, 2021(9), 1-14, 2021.

[13] Martowibowo, S. Y., & Damanik, B. K., Optimization of material removal rate and surface roughness of aisi 316l under dry turning process using genetic algorithm, Manufacturing Technology, 21(3), 373-380, 2021.

[14] Zhang, B., Zhou, X., Liu, Y., Yang, B., & Zhang, Y., ACombining application of wavelet analysis and genetic algorithm in wind tunnel simulation of unidirectional natural wind field near a sand ground surface, Review of Scientific Instruments, 92(1), 015123-015126, 2021.

[15] Meng, Y., Liang, Y., Zhao, Q., & J Qin., Research on torsional property of body-in-white based on square box model and multiobjective genetic algorithm, Mathematical Problems in Engineering, 2021(41), 1-13, 2021.

[16] Onyema, E.M; Elhaj, M.A.E; Bashir, S.G; Abdullahi, I; Hauwa, A.A; Hayatu, A.S., Evaluation of the Performance of K-Nearest Neighbor Algorithm in Determining Student Learning Styles, Int. J. of Innovative Sci., Eng. & Techn., 7 (1), 91-102, (2020).

[17] Onyema, E.M.., Edeh, C. D.., Gregory, U.S..,. Edmond, V.U., Charles, A.C. and Richard-Nnabu, N.E. Cybersecurity Awareness Among Undergraduate Students in Enugu Nigeria, International Journal of Information Security, Privacy and Digital Forensic (Nigeria Computer Society), 5 (1), 34 -42, (2021).

[18] Li, Y., Liu, X., Tian, H., & Luo, C. (2018, November). Research of Industrial Control System Device Firmware Vulnerability Mining Technology Based on Taint Analysis. In 2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS) (pp. 607-610). IEEE.

[19] Javaheri, D., Lalbakhsh, P., & Hosseinzadeh, M., A novel method for detecting future generations of targeted and metamorphic malware based on genetic algorithm IEEE Access, 9, 69951-69970, 2021

[20] Xie, N., Qin, Z., & Di, X., GA-StackingMD: Android Malware Detection Method Based on Genetic Algorithm Optimized Stacking Applied Sciences, 13(4), 2629, 2023.

[21] Hammood, L., Doğru, İ. A., & Kiliç, K., Machine Learning-Based Adaptive Genetic Algorithm for Android Malware Detection Applied Sciences, 13(9), 5403, 2023.

[22] Lee, J., Jang, H., Ha, S., & Yoon, Y., Android malware detection using machine learning with feature selection based on the genetic algorithm Mathematics, 9(21), 2813, 2023.

[23] Senanayake, J., Kalutarage, H., Al-Kadri, M. O., Petrovski, A., & Piras, L., Android source code vulnerability detection: a systematic literature review ACM Computing Surveys, 55(9), 1-37, 2023.

[24] Li, G., Source Code Vulnerability Mining Method Based on Graph Neural Network International Journal of Frontiers in Engineering Technology, 4(4).

[25] Sarker, I. H., Ai-based modeling: Techniques, applications and research issues towards automation, intelligent and smart systems SN Computer Science, 3(2), 158, 2022.

[26] Zhang, H., Song, R., Yang, J., Dan, W. U., & Wang, Y., Connection damage detection of double beam system under moving load with genetic algorithm, Mechanika, 27(1), 80-87, 2021.

Edited by: Pradeep Kumar Singh
Special issue on: Intelligent Cloud Technologies Enabled Solutions for Next Generation Smart Cities
Received: Nov 13, 2022
Accepted: Sep 9, 2023