International Journal of Advanced Science and Research
www.allsciencejournal.com
ISSN: 2455-4227
Received: 20-02-2022, Accepted: 07-03-2022, Published: 22-03-2022
Volume 7, Issue 1, 2022, Page No. 21-27

A mechanism to reduce the probability of trojan horse attacks on computers

Huakun Wu[1], Yahui Meng[1*], Z Y Chen[1], Wanting Chen[1], Timothy Chen[2]
[1] Guangdong University of Petrochemical Technology, Sch Sci, Maoming, Peoples R China
[2] Division of Engineering and Applied Science, California Institute of Technology, Pasadena, CA, US
*Corresponding Author: Yahui Meng

Abstract
Purpose: By dividing the information in the information system into different security levels, it ensures the isolation of different levels of information, and those with different permissions Each user can access different levels of confidential information.
Design/methodology/approach: The main function of access control is to allow legitimate users to access and use the protected resources and services of the system, and to prevent illegal users from accessing and obtaining related resources and services. Trust management is a common method used to explain security policies, it provides a standard and common mechanism for security policies and review the corresponding security of important operations, and directly authorize them after the review is passed.
Findings: Considering the relationship between different levels of defense capability and security, this paper discusses the data path that trojan attack network must rely on, and attack methods from a single source node to an intermediate node or from a single source node to a tail node or from a single source node to attack both the intermediate node and the tail node at the same time. Three probability models are established to discuss the relationship between attack probability of Trojan horse among nodes.
Originality/value: Considering the relationship between different levels of defense capability and security, this paper discusses the data path that trojan attack network must rely on, and attack methods from a single source node to an intermediate node or from a single source node to a tail node or from a single source node to attack both the intermediate node and the tail node at the same time. Three probability models are established to discuss the relationship between attack probability of Trojan horse among nodes. Finally, the model is applied to an enterprise information system, and the result is reasonable.

Keywords: trojan attacks, network security based on multilevel security policy, probability model, data path, computer network security, network node

Introduction

In modern society, with the rapid development of computer network and communication technology, using open network to realize global communication has become the development trend of information management. However, providing shared network resources also brings various risks. Therefore, in order to protect their own interests, many organizations and institutions have established their own communication subnet and resource subnet to protect internal nodes from illegal access, so as to protect your data from being illegally accessed and causing leakage (Bao, D and Srivastava, 2018) [1]. In these resource subnets and communication subnets, the network based on multi-level security policy is adopted by many organizations, such as some large Internet companies, senior government agencies and the military. At the same time, because many networks have adopted multi-level security policies, in this network mode, more and more Trojan horses are actively seeking and attacking vulnerabilities by targeting network nodes under the temptation of confidential information (Hasegawa *et al.* 2018) [4]. In recent years, they have launched round after round of attacks. A small enough Trojan can be masked by process change and noise on the side channel fingerprint of logic and circuit. In advanced nodes, the scale of integration is gradually expanding, which makes the situation worse. From the above analysis, We can't help asking whether the network based on multi-level security policy can effectively resist Trojan horse attacks? Once the Trojan horse invades the nodes in the network, how likely are the nodes with different security levels attacked by Trojan horse? How likely is the node to connect with the intrusion node infected by Trojan horse? These important issues involve the interests of relevant organizations, but there are no strong measures to prevent them. So far, there are few studies on this prevention.
Therefore, this paper studies the relationship between defense capability and security level (Ruo,2018) [9], analyzes the possibility of Trojan attacking nodes through data path, and establishes three probability models to provide reference for network Trojan prevention based on multi-level security strategy.

Previous works and their Problems

Based on the multi-level security strategy, the data path of computer network security is analyzed, and several probability models to resist Trojan horse attacks on high-level nodes are established. Finally, this paper selects a

specific enterprise information network to test our model. The model provides a powerful reference for the computer network Trojan defense and related security evaluation based on multi-level security strategy. Because the relationship between security level and defense capability is not clear, only a general formula can be given in this paper. Therefore, we need to further study this relationship. Practice is the only standard to test truth. We should apply it in combination with the actual situation, and improve the relevant contents constantly, so as to enrich it constantly (Zhang *et al*, 2015) [5]. We can find out the dependence between them through Trojan horse attack test.

Here we set the node to Vi (i=1,2,3...), and the initial attack node is VS. Suppose Vs = {V1}, then Trojan can propagate along the data to other nodes: {V2, V3, V4, V5, V5.1, V5.2, V6, V6.1, V6.2, V7, V7.1, V7.1.1, V7.2}. Here, taking V7.2 as an example, started by Trojan horse attack characteristics starting from VS = {V1}. Here we set the length of the path to access a node to 1, and the identifier of the access length is $\varphi(i)(i=1,2,3,....)$,The path length between the attacked node at the head and the attacked node at the tail can be expressed as $l_i(i=1,2,3...)$, the path length between each adjacent node is 1(Liu *et al*,2010).From the calculated result $\varphi(1)=1$, $\varphi(7,2)=6$, we get: $l_{7.2-1}=\varphi(7.2)-\varphi(1)=5$, The average degree (defense factor) of nodes can be calculated according to Figure 1: $k=\frac{0+1+2+3+3+3+4+4+4+4+4+5+5+6}{14}=\frac{24}{7}$. Take these results into equation (Xiang *et al*, 2019) $Z^{(j)}=\sum_{V_i\in V_s}[\frac{1}{k^{l_{i,j}}}a_{i,j}^{(l_{i,j})}\prod_{t=P_i+1}^{P_i+l_{i,j}}\frac{1}{t}]$, we obtain the corresponding probability: $P^{(7,2)}=10.42\%$ .By the same way, we can obtain the probability of other nodes attacked by Trojan horses and the results are shown in Table 1(Mathure *et al*, 2020).

Table 1: The probability of nodes attacked by Trojan horse (Zhong *et al*,2015)

Sources nodes	Target nodes	Probility
V1	V_2	53.50%
	V_3	42.45%
	V_4	31.30%
	V_5	31.30%
	$V_{5.1}$	20.90%
	$V_{5.2}$	20.90%
	V_6	31.30%
	$V_{6.1}$	20.90%
	$V_{6.2}$	20.90%
	V_7	20.90%
	$V_{7.1}$	10.45%
	$V_{7.1.1}$	6.15%
	$V_{7.1.2}$	3.08%

In the former test cases, through the analysis, the predecessors put forward their own views: first, the transmission of information, some nodes are attacked by Trojans, Trojans continue to extend through the transmission of data path (Malik and Sharad, 2015) [7], information companies should strengthen their intrusion monitoring to protect the company's information from leakage. The data path is connected by these nodes, so we should increase the intensity of killing Trojans. In addition, we should simplify the network topology and output redundant paths to reduce the possible spread of Trojan horse. On this basis, I will analyze the situation I put forward to make the test more comprehensive (Doukim *et al*, 2016).

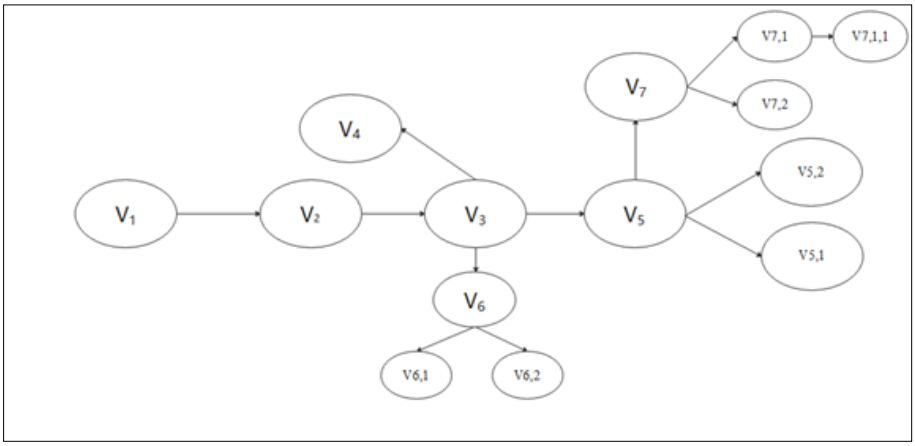


Fig 1: The information system network topology of an Internet company

Modeling and analysis of three Cases

In the network based on multi-level security policy, the most important thing is the confidentiality of information (Guha *et al*, 2017) [3]. The difference between different confidential information is that it is placed on nodes of different security levels.

A. basic concepts

Concept 1: entity refers to network resources (such as files, processes, devices, etc.) and legitimate users;

Concept 2: topic refers to the action that an active entity performs on its entity. The topic here refers to the host (client / server) participating in the communication, and each host has a security level;

Concept 3: an object is a passive entity operated by an active entity. The object here is the data of communication between hosts, such as datagram. Each object has a secret level;

Concept 4: security level is the security attribute of an entity in a network. Security level consists of sensitive level and category set. The security level is used to determine whether topics are allowed to access objects. The relationship between the security level set and the level determined on it constitutes a lattice;

Concept 5: security level refers to the security information assigned to an object, reflecting the importance of the object. The security level is also composed of sensitive level and category;

Concept 6: visa reflects the degree of trust of the corresponding subject. It also has the same level of security. The user's visa is assigned according to the background investigation of the user by the network security officer. The process visa is determined by the user visa it represents.

B. Signature based detection technology detects the situation at the starting point

Signature based detection technology is mainly based on the idea of pattern matching, which generates a unique signature signature signature token for each known malicious code to create malicious code base. These signatures include many different properties, such as file name, content string or byte, and also discusses the protection of system security from the perspective of eliminating the security vulnerabilities caused by these malicious code (Subramani *et al*, 2020) [11]. Compare the signature characteristics of unknown code with malicious code base, and there are matching malicious code signatures when searching for malicious code base. If there is any match, it will be judged as malicious code; otherwise, it will be judged as normal code. These signatures are manually found by experts or generated by automatic methods. A signature is extracted to mark the characteristic properties of a specific malicious code. The implementation steps based on signature method are as follows:

Step 1: Collect known malicious code samples;

Step 2: In malicious code samples, malicious code signature (feature) is extracted;

Step 3: Include the signature in the malicious code database;

Step 4: Test the file. If the detected file contains the signature in the malicious code library, it is judged that the file is malicious code or has been infected by malicious code.

The equation of straight line described by oblique section: y = kx + b, which is not suitable for straight lines parallel to y axis. Suppose the measured data pairs are n groups (n > 2), (X1, Y1), (X2, Y2), (X3, Y3),..., (Xn,, Yn), n>2 that is to say, in order to solve this linear equation, only two data pairs are needed, that is, two points are used to determine a straight line (Zhao *et al*, 2019). However, such a line is inaccurate for the actual measurement. When n>2, the solution theoretically is called the over determined equation because the condition number exceeds the unknown number of the equation, so it is called the over determined equation. Since the true value (theoretical value) is (y = kxi+ b), the data pair is substituted. The residual is yi-y = yi-kxi-b, which is obtained by the least square equation:

$$\min f = \sum_{i=1}^{n} (y_i - kx_i - b)^2$$

Here, there are two variables k, b. the following equation is obtained by solving partial derivatives:

$$\begin{cases} \dfrac{\partial f}{\partial k} = \sum_{i=1}^{n} [(y_i - kx_i - b)g(-x_i)] = 0 \\ \dfrac{\partial f}{\partial b} = \sum_{i=1}^{n} [(y_i - kx_i - b)g(-1)] = 0 \end{cases}$$

By expanding the above formula, we can get the following results:

$$\begin{cases} \sum_{i=1}^{n} (x_i y_i) - k\sum_{i=1}^{n} (x_i^2) - b\sum_{i=1}^{n} (x_i) = 0 \\ \sum_{i=1}^{n} (y_i) - k\sum_{i=1}^{n} (x_i) - nb = 0 \end{cases}$$

Make $A = \sum_{i=1}^{n} (x_i^2), B = \sum_{i=1}^{n} (x_i), C = \sum_{i=1}^{n} (x_i y_i), D = \sum_{i=1}^{n} (y_i)$, get

$$\begin{cases} Ak + bB = C \\ Bk + nb = D \end{cases}$$

$$\begin{cases} k = \dfrac{Cn - BD}{An - B^2} \\ b = \dfrac{AD - CB}{An - B^2} \end{cases}$$

Through the above method, the following figures are obtained after matlab calculation:

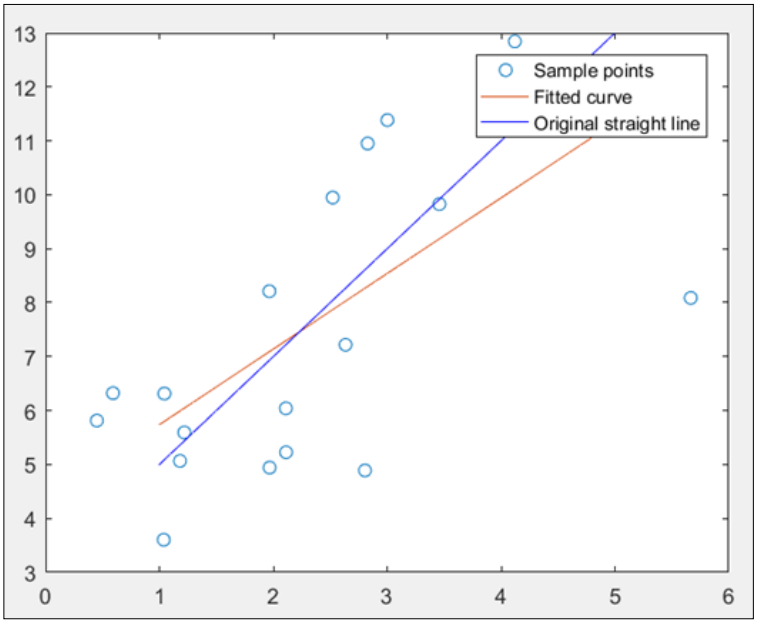
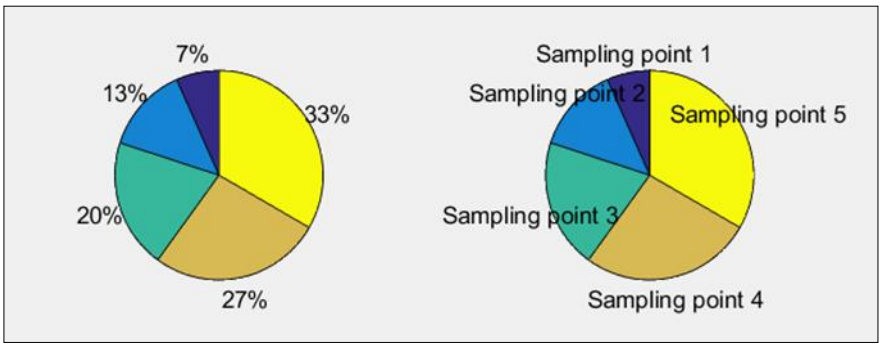


Fig 2: The situation of starting point attack



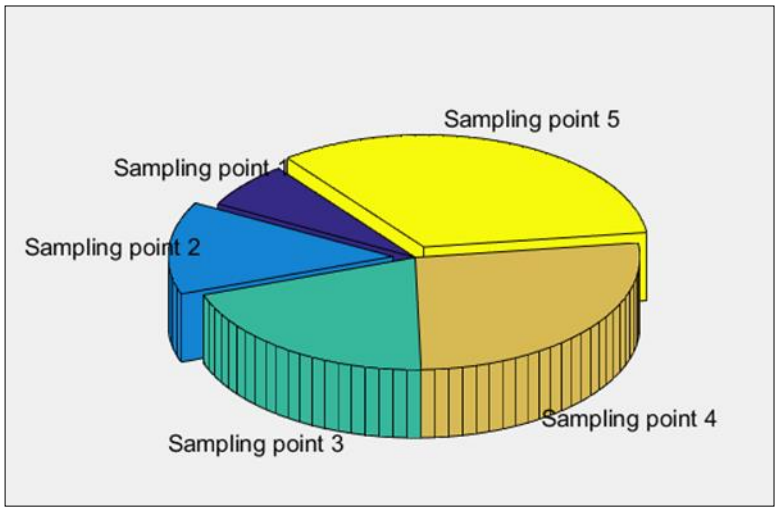


Fig 3, 4: The probability of the starting point being broken

From the above figure, we can draw the following conclusion: the farther away from the starting point, the lower the probability of successful attack. When the point is between nodes, the probability will change to a certain extent. Let me continue the analysis.

C. Detection of attack probability in intermediate path based on logistic regression algorithm

Logistic regression is a powerful statistical method, which can use one or more explanatory variables to express a binomial result. It estimates the probability by using logic function, so as to measure the relationship between class dependent variables and one or more independent variables, which obey cumulative logic distribution. The essence of logistic regression is to assume that the data obey this distribution (He *et al*, 2015) [5], and then use the maximum likelihood estimation to estimate the parameters. The l-distribution is a continuous probability distribution. Its distribution function and density function are as follows:

$$F(x) = P(X \le x) = \frac{1}{1+e^{-(x-\mu)/\gamma}}$$

$$f(x) = F'(X \le x) = \frac{e^{-(x-\mu)/\gamma}}{\gamma(1+e^{-(x-\mu)/\gamma})^2}$$

Where μ is the position parameter and γ $(\gamma > 0)$ is the shape parameter. The decision boundary can be expressed as $\omega_1 x_1 + \omega_2 x_2 + b = 0$. assuming a sample point $h_\omega(x) = \omega_1 x_1 + \omega_2 x_2 + b > 0$, it can be judged that its category is 1. This process is actually a perceptron. Logistic regression also needs to add a layer, it needs to find the direct relationship between the classification probability p (y = 1) and the input vector x, and then judge the category by comparing the probability values. In this paper, we consider the binary classification problem: $D = (x_1, y_1), (x_2, y_2), ..., (x_N, y_N), x_i \subseteq R^n, y_i \in 0,1, i = 1,2,...,N$, Considering that the value of $\omega^T x + b$ is continuous, it can not fit discrete variables. It can be used to fit conditional probability $p(Y = 1 | x)$, because the value of probability is continuous. But for $\omega \neq 0$ (if it is equal to zero vector, there is no value to solve), $\omega^T x + b$ is R, and the probability is 0 to 1, so the generalized linear model is considered. The most ideal is the unit step function:

$$p(y = 1 | x) = \begin{cases} 0, & z < 0 \\ 0.5, & z = 0, z = \omega^T x + b \\ 1, & z > 0 \end{cases}$$

However, the step function is not differentiable, and the logarithmic probability function is a commonly used substitute function: $y = \frac{1}{1+e^{-(\omega^T x+b)}}$, So there are: $\ln\frac{y}{1-y} = \omega^T x + b$ .We regard y as the probability that x is a positive example, then 1-y is the probability that x is its counterexample. The ratio of the two is called probability, which refers to the probability ratio of the occurrence and nonoccurrence of the event, if the probability of the occurrence of the event is p. Then the logarithm probability: $\ln(odds) = \ln\frac{y}{1-y}$. In this paper, y is regarded as the class a posteriori probability estimate:

$$\omega^T x + b = \ln\frac{P(Y = 1 | x)}{1 - P(Y = 1 | x)}$$

$$P(Y = 1 | x) = \frac{1}{1+e^{-(\omega^T x+b)}}$$

That is to say, the logarithm probability of output y = 1 is expressed by the linear function of input x, which is the logistic regression model. When the value of $\omega^T x + b$ is closer to positive infinity, the probability value of $P(Y = 1 | x)$ is closer to 1. Therefore, the idea of logistic regression is to first fit the decision boundary (not limited to linear, but also polynomial), and then establish the probability relationship between the boundary and the classification, so as to obtain the probability in the case of two classification. After the mathematical form of the logistic regression model is determined, the rest is how to solve the parameters in the model. In statistics, the maximum likelihood estimation method is often used to solve the problem, that is, to find a group of parameters, so that under this group of parameters, the likelihood (probability) of our data is the maximum, Hypothesis:

$$P(Y = 1 | x) = p(x)$$
$$P(Y = 0 | x) = 1 - p(x)$$

Likelihood function:

$$L(\omega) = \prod [p(x_i)]^{y_i} [1 - p(x_i)]^{1-y_i}$$

In order to solve the problem more conveniently, we take logarithm on both sides of the equation and write it as log likelihood function:

$$L(\omega) = \sum [y_i \ln p(x_i) + (1 - y_i) \ln(1 - p(x_i))] = \sum [y_i \ln \frac{p(x_i)}{1 - p(x_i)} + \ln(1 - p(x_i))] = \sum [y_i (\omega \cdot x_i) - \ln(1 + e^{w \cdot x_i}))]$$

In machine learning, we have the concept of loss function, which measures the degree of model prediction error. If we take the average log likelihood loss of the whole data set, we can get the following results: $J(\omega) = -\frac{1}{N}\ln L(\omega)$. That is to say, in the logistic regression model, the maximum likelihood function and the minimum loss function are actually equivalent. There are many ways to solve logistic regression. Here we mainly talk about gradient descent and Newton method. The main objective of optimization is to find a direction. The value of loss function can be reduced when the parameter moves in this direction. This direction is often obtained by various combinations of first-order partial derivatives or second-order partial derivatives. The loss function of logistic regression is as follows:

$$J(\omega) = -\frac{1}{n}(\sum_{i=1}^{n}(y_i \ln p(x_i) + (1-y_i)\ln(1-p(x_i)))$$

.

Method 1 : Gradient descent is to find the descent direction through the first derivative of J (W) to W, and update the parameters iteratively:

$$g_i = \frac{\partial J(\omega)}{\partial \omega_i} = (p(x_i) - y_i)x_i$$

$$\omega_i^{k+1} = \omega_i^k - \alpha g_i$$

Where k is the number of iterations. Each time the parameters are updated, the iteration can be stopped by comparing that $|J(\omega^{k+1}) - J(\omega^k)|$ is less than the threshold or reaches the maximum number of iterations.
Method 2 : The basic idea of Newton's method is to do the second-order Taylor expansion of F (x) near the existing minimum estimate, and then find the next estimate of the minimum. Suppose ω^k a is the current minimum estimate, then there are: $\varphi(\omega) = J(\omega^k) + J'(\omega^k)(\omega - \omega^k) + \frac{1}{2}J''(\omega^k)(\omega - \omega^k)^2$, And then let $\varphi'(\omega) = 0$ get $\omega^{k+1} = \omega^k - \frac{J'(\omega^k)}{J''(\omega^k)}$. Therefore, there is an iterative update formula:

$$\omega^{k+1} = \omega^k - \frac{J'(\omega^k)}{J''(\omega^k)} = \omega^k - H_k^{-1} \cdot g_k$$

Where H_k^{-1} is Hessian matrix: $H_{mn} = \frac{\partial^2 J(\omega)}{\partial \omega_m \partial \omega_n} = h_\omega(x^{(i)})(1 - p_\omega(x^{(i)}))x_m^{(i)}x_n^{(i)}$
In addition, this method needs the objective function to be second order continuous differentiable, and the J (W) in this paper meets the requirements. The final results are as follows:

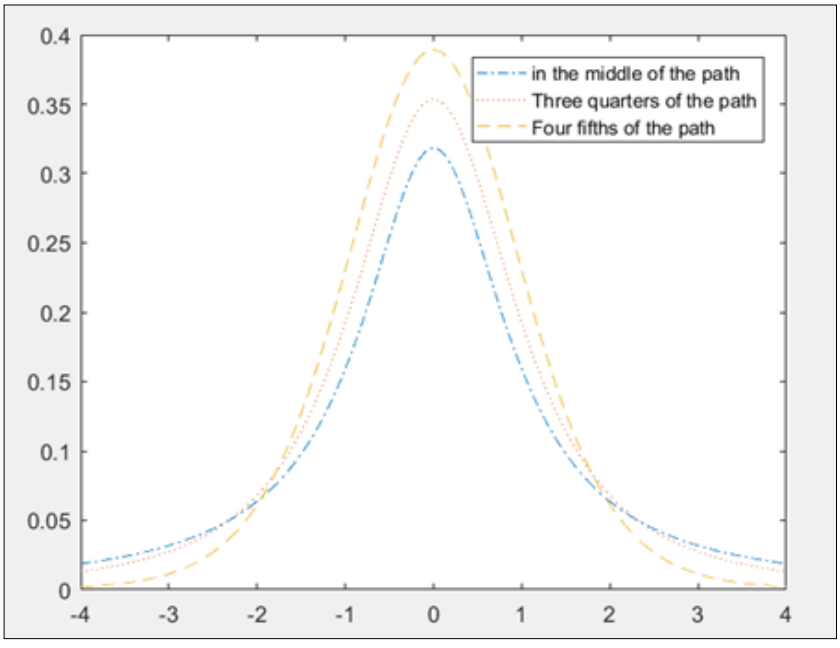


Fig 5: attack probability in the middle path

Conclusion and Future Work

This paper presents an effective simulation method for Trojan horse attack, and gives the probability of successful attack. Different from the way of predecessors, it studies the way of attack rather than the path. Heuristic algorithm and logistic regression algorithm are used to calculate the attack probability of Trojan horse in different ways, and the approximate results are obtained. Simulation results show the effectiveness and availability of the method. On this basis, I intend to continue to study the attack probability of other attack methods.

References

1. Bao C, D Forte, Srivastava A. On reverse engineering-based hardware trojan detection. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,2018:35(1):49-57.
2. Doukim CA, Dargham JA, Chekima A, Omatu S. Combining neural networks for skin detection. Signal & Image Processing, 2011, 1(2).
3. Guha K, Saha D, Chakrabarti A. Self Aware SoC Security to Counteract Delay Inducing Hardware Trojans at Runtime. International Conference on Vlsi Design & International Conference on Embedded Systems. IEEE, 2017.
4. Hasegawa K, Yanagisawa M, Togawa N. [ieee 2018 ieee international conference on consumer electronics (icce) - las vegas, nv, usa (2018.1.12-2018.1.14)] 2018 ieee international conference on consumer electronics (icce) - a hardware-trojan classification method utilizing boundary net structure, 2018, 1-4.
5. He L, Qiang L, Zhang J, Lv YQ. A Survey of Hardware Trojan Detection, Diagnosis and Prevention. CAD/Graphics 2015. IEEE., 2015.
6. Liu YF, Zhang LW, Liang J, Qu S, Ni ZQ. Detecting Trojan horses based on system behavior using machine learning method. International Conference on Machine Learning & Cybernetics. IEEE, 2010.
7. Malik, Sharad. [ieee 2015 formal methods in computer-aided design (fmcad) - austin, tx, usa (2015.9.27-2015.9.30)] 2015 formal methods in computer-aided design (fmcad) - detecting hardware trojans: a tale of two techniques, 2015, 6-6.
8. Mathure N, Srinivasan SK, Ponugoti KK, Malik A, Quanbeck S. A Formal Verification Approach for Detecting Opcode Trojans. 2020 27th IEEE International Conference on Electronics, Circuits and Systems (ICECS). IEEE, 2020.
9. Ruo-Lei Z. Detection of attack by hidden process using anti-hook technology. Journal of Nanchang University (Engineering & Technology), 2008.
10. Shila DM, Venugopal V. [ieee icc 2014 - 2014 ieee international conference on communications - sydney, australia (2014.6.10-2014.6.14)] 2014 ieee international conference on communications (icc) - design, implementation and security analysis of hardware trojan threats in fpga, 2014, 719-724.
11. Subramani KS, Helal N, Antonopoulos A, Nosratinia A, Makris Y. Amplitude-modulating analog/rf hardware trojans in wireless networks: risks and remedies. IEEE Transactions on Information Forensics and Security,2020:(99)1-1.
12. Xiang B, Hao YJ, Zhang Y, Liu HY. A Novel Anti-Trojan Approach using Behavioral Analysis. 2008 International Conference on Apperceiving Computing and Intelligence Analysis. IEEE, 2009.
13. Zhang JL, Fang L, Li L, Zhang ZX. A Novel Approach to Detecting Hardware Trojan Horses. 2015 8th International Symposium on Computational Intelligence and Design (ISCID). IEEE, 2015.
14. Zhao Y, Hu X, Li S, Ye J, Deng L, Ji Y, *et al*. Memory Trojan Attack on Neural Network Accelerators. 2019 Design, Automation & Test in Europe Conference & Exhibition (DATE). IEEE, 2019.
15. Zhong M, Fan Y, Huanzhou LI, Tang Z, Zhang J. Heuristic detection system of trojan based on trajectory analysis. Journal of Computer Applications, 2015.