Command Execution Bypassing Evasion Based on Win32Net Module

Shengke Ye[a], Weiqiang Qin[b], Tinghua Chen[c], Dewei Ou[d*], Genhai Zhang[e]

School of Information Technology, City College of Huizhou, Huizhou, China Ac

[a]yeshengke@tm.hzc.edu.cn, [b]qinweiqiang@tm.hzc.edu.cn,
[c]chentinghua@tm.hzc.edu.cn, [d*]2873366260@qq.com*,
[e]zhanggenhai@tm.hzc.edu.cn

Abstract. Bottom layer API interface evasion technology is one of the core technologies in current evasion technology and is also one of the leading technologies in network security. In order to further promote the development of evasion technology, propose a method of evading antivirus detection by calling WindowsAPI interfaces at the Bottom layer in the Win32NET module. This method first utilizes the low-level interfaces of the Windows system to execute the required commands and bypasses common antivirus software detection during the calling process. Through Bottom layer invocation, we avoid antivirus software's regular expression matching and monitoring, thus enhancing the concealment of Trojan software. conducted evasion tests in the Win10 system environment and successfully bypassed Chinese security software with the evasion technology of the Win32NET module. employed various evasion methods such as code obfuscation, code encryption, and file bundling for comparative testing and validation. The results indicate that the Bottom layer API interface evasion technology of the Win32NET module has a higher probability of successful evasion. Compared to other evasion methods, Win32NET evasion technology is more practical with simpler code and superior evasion effectiveness.

Keywords: kill-free technology: underlying calls: command execution: antivirus software kill-free probability statistics:

1 INTRODUCTION

In today's cybersecurity landscape, malicious software continues to evolve, rendering traditional defense mechanisms inadequate in combating new threats. Therefore, evasion techniques, as an emerging security defense approach, have garnered significant attention. This paper aims to review the current status and challenges of evasion techniques in cybersecurity, and explore future directions for technological development.

The continuous evolution of malware has surpassed the capabilities of traditional defense mechanisms[1]. brought unpredictable and severe challenges to the development of cybersecurity.[2] Therefore evasion techniques, as an emerging form of security

A. Haldorai et al. (eds.), *Proceedings of the 2024 3rd International Conference on Artificial Intelligence, Internet and Digital Economy (ICAID 2024)*, Atlantis Highlights in Intelligent Systems 11,
https://doi.org/10.2991/978-94-6463-490-7_39

defense, have attracted considerable attention. Accurate experimental analysis, prediction, and statistical analysis of various evasion techniques and indicators are of significant importance for network security evasion technologies.[3] This paper aims to comprehensively review the current application status of evasion techniques in the field of cybersecurity. The continuous updates and iterations of evasion techniques have filled the gaps in traditional defense measures. For a long time, industry professionals have been concerned about the evasion of Trojan detection methods.[7,8] Extensive research has been carried out [3,4] And discussion[5,6] and has achieved significant research results[9,10], Roughly categorized into decomposition set methods[13] and camouflaged methods[14] The method represented by the decomposition set method [11,12] There are characteristics of the decomposition, and code command decomposition, any kind of Trojan virus itself contains certain characteristics, a variety of detection and killing techniques are used to extract features, analyze the characteristics of the method to achieve the purpose of identifying the Trojan virus. Currently, the mainstream check and kill technology is ultimately based on feature matching and identification algorithms, analyzing malicious program code, matching antivirus software malicious rule feature templates, identifying Trojan horse viruses, and then perform check and kill operations. The vast majority of antivirus software products for the program malicious behavior of the statistical scope is incomplete, rarely on the malicious behavior between the program correlation mining analysis, resulting in the use of inter-program correlation behavior can not be protected and check the use of Trojan horse viruses to carry out malicious attacks. For example, a single program is not a Trojan virus, but multiple programs can establish a collaborative relationship based on certain association rules, thus realizing the purpose of malicious attacks. This method effectively eliminates a Trojan with concealed features through the decomposition of Trojan characteristics [15] This type of method, which better reflects the typical behavior of antivirus software, is to perform antivirus by recognizing features. The advantage of such methods lies in their ability to target the characteristics of Trojans and execute corresponding evasive strategies, based on deep-level analysis. However, the drawback is that they appear relatively ineffective against antivirus software that does not rely solely on feature recognition. These methods are more suitable for targeting antivirus software that primarily relies on feature recognition for security. Represented by obfuscation methods, such as file bundling, packing, file compression, file recombination, etc., these methods aim to generate evasive Trojans based on obfuscation features. They can evade detection by conventional antivirus software but are limited to modifying, packing, bundling, and other operations on pre-generated Trojans. However, they have a disadvantage compared to feature decomposition methods and are more suitable for simple Trojan evasion testing.

This paper proposes a method for evasion testing based on the Win32NET module's low-level interface calls. Initially, data analysis and probability assessment are conducted on Precision as the primary layer, followed by secondary analysis and probability assessment on Recall data. Through the multifaceted data interfaces, F1 scores are derived, and subsequently, Accurate probability, Extinction probability, and Escape probability are calculated using a formulated equation. The comparative and diversified evasion testing methods are analyzed, summarized, and judged. Results indicate that

the proposed method demonstrates superior effectiveness in bypassing the majority of antivirus software in China, with a higher success rate in evasion. The data obtained from experiments, combined with calculated formulas, provide direct confrontation data against Chinese antivirus software for the exemplified evasion methods, yielding bimodal probabilities of victory. The superiority of each method is intuitively revealed through the probability data analysis.

2 WIN32NET BYPASSING PRINCIPLE AND METHOD

2.1 Execute CMD to add users normally

In the research, a novel method has been discovered for adding users using CMD commands, along with an alternative execution approach, which bypasses antivirus detection. Typically, users can be added through user management in computer management or by executing CMD commands. A commonly used CMD command is as follows:

net user username password /add

Under normal circumstances, the process of adding a user via CMD involves the following steps:1. Open CMD.2. Construct the command to add the user.3. Execute the command.

During this process, installed antivirus software will continue to monitor for any suspicious activities. If the antivirus software detects a command executed by CMD that poses a high risk or contains malicious code, it will pause and intercept the command. It will then display a prompt window to the user, alerting them to the potential danger and asking for permission to proceed with the execution of the command. And show in figure 1 to 2.

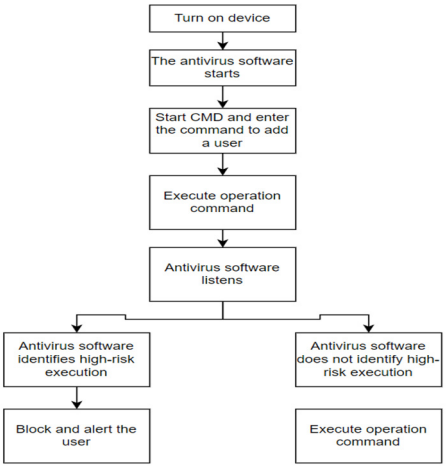

Fig. 1. Flowchart of CMD user creation execution

2.2 Win32NET module call to add user

The Win32NET module call is an operation on the underlying API of the Windows system. When it starts to execute, it first detects the code. When the code is executed, instead of performing the add operation through the CMD window, the user is added through the Win32NET module's call to the Windows system's underlying API service. In this way, the antivirus software will only listen to the commands of the CMD window, but not to other interfaces called by the Win32NET module. Therefore, the bypass effect is successfully realized and the operation of adding users is directly executed. After execution, an object is constructed that the attacker wants to create, i.e. the user to be added. This object is created by the Win32NET module calling the underlying service interface of the system API to add the user the attacker wants to create to the target computer. This process is executed on a computer that has a checking function and successfully bypasses the detection of the antivirus program and creates the desired user object on the target computer.

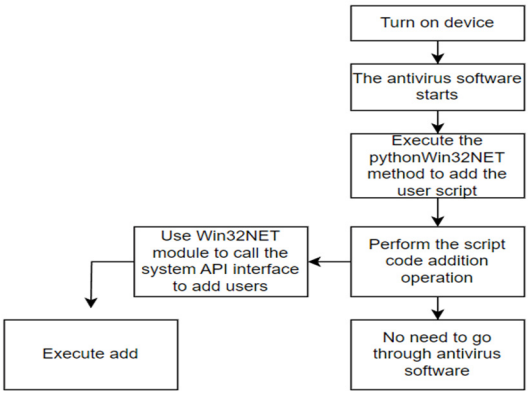

Fig. 2. Win32NET module creates user execution flow chart

When writing code, it is necessary to call both the win32net module and the win32netcon module. While the win32net module contains many functions for creating and managing user accounts, groups, shared resources, network connections, etc., the reason for also calling the win32netcon module lies in its inclusion of various constants and error codes. These constants play a crucial role in the function calls within the win32net module.

$$import\ win32net$$

$$import\ win32netcon$$

These constants define a lot of information related to network operations, such as privilege levels, error codes, and so on. By using these constants, it is possible to write code that is more readable and maintainable, and that is more compatible across different versions of Windows. This has the advantage of avoiding hard-coded constants, making the code easier to understand and maintain. The use of these constants also improves the portability of the code by ensuring that the behavior of the code remains consistent across different Windows environments.

3 RELATED WORK

To assess the evasion capabilities of anti-detection techniques within commonly used antivirus software in China, a series of experiments were designed. Six representative domestic antivirus software were tested, including 360 Total Security version 13.0.0.2006, Huorong Security Software version 5.0.75.3, Tencent PC Manager version 16.9.24712.211, 2345 Security Guard version 8.12, Rising Antivirus Software version 25.00.09.95, and Jiangmin Antivirus Software version 13.00.900. In the experiments, the latest versions of each antivirus software were initially collected to ensure the accuracy and effectiveness of the testing. Subsequently, a selection of classic anti-detection techniques was chosen, and the detection capabilities of each antivirus software were evaluated. These anti-detection techniques include but are not limited to code obfuscation, encryption. To ensure the accuracy and comparability of the experiments, a standardized set of environmental factors was utilized. All experiments were conducted on a Windows 10 Professional Workstation Edition, version number 22H2. The operating system employed was Windows 10 Professional Workstation Edition, version 22H2. The CPU utilized was a 13th Gen Intel(R) Core(TM) i7-13700KF 3.42 GHz (2 processors), employing identical processor models. The system was equipped with 20.0GB of RAM and operated on a 64-bit operating system. The local hard disk capacity was 200GB.

Additionally, Python version 3.8.5 was uniformly selected for the experiments. In order to ensure rigor and fairness in the experiments, Python was chosen as the programming language for implementing experimental code and proposing methods. Various evasion techniques commonly found in the market, including code obfuscation, encryption, packing, file bundling, and software signing, were implemented using Python. This standardization not only enhances the comparability of experimental results but also demonstrates the optimization and rigor of the experiments. By utilizing Python to develop evasion techniques, it becomes easier to comprehend and compare the effectiveness of different evasion techniques. Moreover, conducting experiments in the same environment helps to minimize biases introduced by language discrepancies. The consistent use of Python for experimental code writing further underscores the rigor and optimization of the experiments, ensuring the reliability and accuracy of the experimental results.

By standardizing environmental factors, it is possible to eliminate interference factors caused by differences in operating systems, processor models, or Python versions, thereby ensuring the reliability and effectiveness of experimental results. This approach

allows the experimental results to more accurately reflect the performance of the studied evasion techniques in specific hardware and software environments, providing a more reliable basis for subsequent analysis and applications.

During the experiments, the same number of detection trials (100 times)x2 were conducted, and multiple evaluation metrics were employed, including Precision, Recall, F1 score, Accurate probability, Extinction probability, Escape probability, etc., to comprehensively assess the detection capabilities of each antivirus software against evasion techniques. By simulating real-world malicious software attack scenarios and observing the reactions and performances of antivirus software under different conditions.

Through the analysis of the experimental results, we can get the comprehensive evaluation of the detection and defense capability of each antivirus software for the kill-free technology, and further compare their influence ability in the domestic security field. These evaluation results will help to provide suggestions for technical improvement and optimization of domestic antivirus software, so as to enhance the overall level and competitiveness of domestic security antivirus software.

4 CALCULATION FORMULA

The elements to be calculated are, respectively, precision rate, recall rate, F1 score, accuracy rate, kill rate and escape rate. The formulas for each are as follows:

$$\text{Precision} = TP / (TP + FP)$$

(1) when it is necessary to make statistics on the probability of the kill-free technology, many calculations need to be carried out in many aspects. First of all, the first calculation is accuracy, which will affect the following data calculation, so the value of calculation accuracy should be particularly rigorous. In the formula, TP and FP are the corresponding antivirus testing software for the first 100 times, the number of times they were checked and killed, and the number of times they escaped.

$$\text{Recall} = TP / (TP + FN)$$

(2) Recall is the calculation of the second 100 times the second retrospective test, to strengthen the rigor and accuracy of the results of the calculations, the formula in the TP and FN means, respectively, antivirus software, the number of times the second check and the number of escapes.

$$F1 = 2 (\text{Precision Recall}) / (\text{Precision} + \text{Recall})$$

(3) F1 takes into account the comprehensive indicators of Precision accuracy and Recall review rate, that is, the flat average of Precision and Recall, to comprehensively evaluate the performance of the model.

By calculating F1 scores, the performance of the model can be evaluated more comprehensively, especially when dealing with unbalanced data sets. F1 scores can better measure the overall performance of the model.

The Precision and Recall of the formula are the numerical results of Precision accuracy and Recall review rate, respectively.

Accurate probability = Precision + Recall + F1 / three

(4) Given how well the no-kill technique correctly predicts overall, the accuracy, recall and F1 scores are averaged. This is also a way to combine these metrics. In the formula Precision and Recall, F1,three are the precision rate value, recall rate value, and F1 score value, respectively, and three represents the three calculated items, which increase according to the number of calculated items.

Extinction probability = Accurate probability

(5) In the formula Extinction probability = Accurate probability It means that the probability of being killed and wiped out is equal to the accuracy.

Escape probability = (100% - extinction probability)

(6) simplified formula name

Escape=ESC | Precision=PRE

Recall=REC | F1=F1

Accurate probability = ACP

Extinction probability = EXP

Escape probability = ESP

5 COMPARISON TEST OF ANTIVIRUS SOFTWARE FREE DIVERSITY

To fulfill the same execution requirement, we adopted commonly seen evasion techniques in the market to add a user on the current experimental computer. We wrote code to meet this requirement and applied prevalent evasion techniques in the market to evade detection. We then conducted testing experiments using commonly used antivirus software in China. Each testing session comprised 100 initial tests followed by 100 subsequent reviews. And show in figure 5.

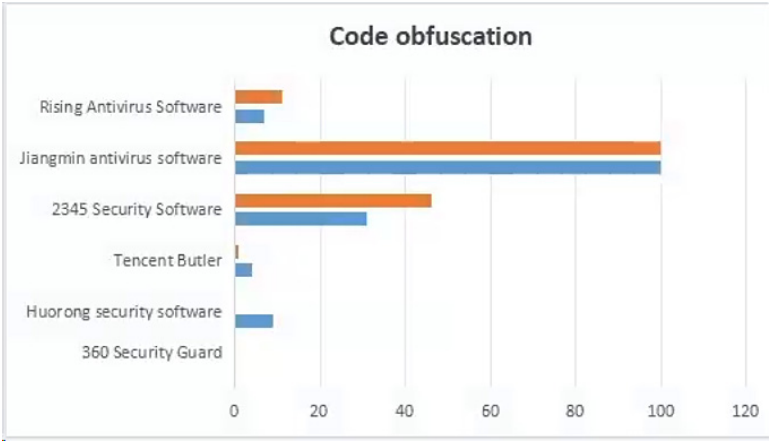

Fig. 3. (Statistical Chart of ode obfuscation)

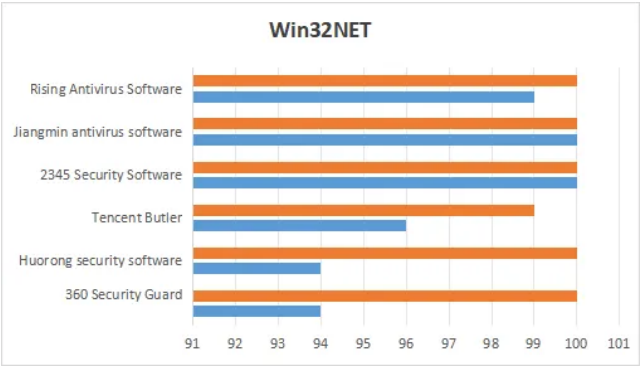

Fig. 4. (Win32NET Test data)

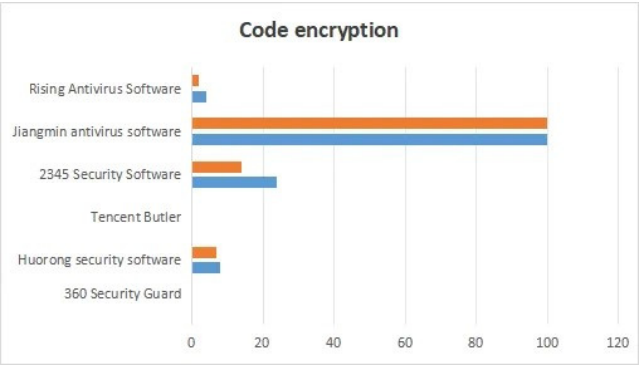

Fig. 5. (Code encryption Test data)

5.1 Code obfuscation

Table 1. Code obfuscation test kill-free bypass success data

	ESC	ESC(2)	PRE	REC	F1	AC-P	EX-P	ES-P
360 Security Guard	0	0	1	1	1	1	1	0
Huorong Internet Security	9	0	0.91	1	0.95	0.95	0.95	0.05
Tencent Butler	4	1	0.96	0.99	0.97	0.97	0.97	0.03
2345 Security Software	31	46	0.69	0.54	0.6	0.61	0.61	0.39
Jiangmin antivi-rus software	100	100	0	0	0	0	0	1
Rising Antivirus Software	7	11	0.93	0.89	0.91	0.91	0.91	0.09

And show in table 1. Figure 3 presents the statistical chart for bypassing tests of code obfuscation, where code obfuscation techniques are employed to evade detection of the Trojan. From the graphical data, it can be observed that the test results for Jiangmin antivirus software and 360 Security Guard show a value of 0 for 360 Security Guard twice. Meanwhile, the value for Jiangmin antivirus software is 100 twice, indicating that the code obfuscation technique did not effectively evade detection by these antivirus programs. However, in the case of Rising Antivirus Software, the effectiveness of the code obfuscation technique is significantly demonstrated. For Huorong Internet Security, Tencent Butler, and Rising Antivirus Software, their evasion performance in anti-detection tests is relatively low, with escape rates only within the range of 15%. However, 2345 Security Software has an escape rate of 50%, indicating a higher evasion capability. In the face of code obfuscation techniques employed by 2345 Security Software, there exists a better anti-detection performance.

From the code confusion test data graph,360 Security GuardThe probability of finding dangers such as Trojans is the highest. HuorongInternetSecurity and Tencent ButlerThere are a small number of successful circumvention cases.

5.2 Code encryption

Table 2. Code encryption test no-kill bypass success data

	ESC	ESC (2)	PRE	REC	F1	AC-P	EX-P	ES-P
360 Security Guard	0	0	1	1	1	1	1	0
HuorongInternetSecurity	8	7	0.92	0.93	0.92	0.92	0.92	0.08
Tencent Butler	0	0	1	1	1	1	1	0
2345 Security Software	24	14	0.76	0.96	0.85	0.86	0.86	0.14
Jiangmin antivirus software	100	100	0	0	0	0	0	1
Rising Antivirus Software	4	2	0.96	0.98	0.96	0.96	0.96	0.04

Statistical Analysis of Successful Evasion of Anti-Malware Detection Through Code Encryption: Test Results of Figure 3 and Table 2, Commonly used code encryption techniques face Tencent Butler and HuorongInternetSecurity and 360 Security GuardHis anti-killing effect is not so good, corresponding to the corresponding test bypass the probability of success is as follows 0,0.08,0,However, if you use code encryption and kill-free means to face the effect of general antivirus software, it will have a certain effect, The bypassed data from the test gives 2345 Security Software, Jiangmin antivirus software Rising Antivirus SoftwareThe probability of success of the corresponding test bypass is 0.14,1,0.04,From the general detection rate of software, it can

be observed that code encryption plays a certain role in evading detection. The success rate of evasion through code encryption depends largely on the complexity of the encryption algorithm employed. Utilizing complex encryption algorithms, such as symmetric encryption algorithms like AES or asymmetric encryption algorithms like RSA, can increase the difficulty of decryption. However, employing more complex algorithms does not necessarily guarantee higher security; in fact, they may be more easily detected by antivirus software.

During evasion testing, when a code's encryption level is excessively high or overly complex, antivirus software may find it difficult to determine or comprehend the intent of the software. As a result, the software may be flagged for further inspection or intense monitoring. This is why the complexity of an algorithm does not necessarily equate to higher security.

5.3 Win32NET module call kill-free bypass

Table 3. Win32NET test kill-free bypass success data

	ESC	ESC (2)	PRE	REC	F1	AC-P	EX-P	ES-P
360 Security Guard	94	100	0.94	0	0	0.94	0.06	0.94
HuorongInternetSecurity	96	99	0.96	0.99	0.97	0.97	0.03	0.97
Tencent Butler	99	99	0.99	0.99	0.99	0.99	0.01	0.99
2345 Security Software	100	100	0	0	0	0	0	1
Jiangmin antivirus software	100	100	0	0	0	0	0	1
Rising Antivirus Software	96	100	0.96	0	0	0.96	0.04	0.96

Win32NET module free kill bypass test, through the free kill bypass test statistics Figure 4 Win32NET chart and Table 3 Win32NET test free kill bypass success data can be seen, the proposed method for the domestic six common antivirus software, the test to get the probability of success of the antivirus have reached more than ninety percent, which concludes that the Win32NET module for the domestic This shows that the Win32NET module for the domestic antivirus software to show the performance of anti-killer than the five examples of anti-killer techniques.

The kill-free method adopted by the Win32NET module is mainly based on the underlying call interface of the system, rather than the traditional methods such as code characteristics, encryption or confusion. The following are the features and advantages of Win32NET module calling API to avoid killing:

1. Based on the underlying system call interface: The Win32NET module's kill-free approach is based on direct calls to the underlying system interface to perform the required operations. By directly interacting with the underlying system interface, it is possible to bypass some of the detection based on code characterization and behavioral analysis, as it does not directly focus on the code itself, but directly interacts with the operating system.

$$win32net.NetUserAdd(None, 1, user\_info)$$

2. Bypassing code characterization: Since the Win32NET module does not use code obfuscation, encryption, or feature elimination techniques, it does not arouse security software's suspicion of the code itself. Security software usually analyzes the code features, but because these features have not been changed in the Win32NET module, so it is easier to bypass detection.
3. Use of the underlying system interface: direct calls to the underlying system interface makes the malicious operation more covert. Interfaces are usually standard functions provided by the operating system, security software is difficult to identify them as malicious behavior. At the same time, because these interfaces are system-level, with higher privileges, can perform more potentially malicious operations.

6 SUMMARY

To address the unstable situation of cybersecurity and the evolving challenges posed by various threats such as Trojan viruses, this paper proposes a method for bypassing antivirus detection based on low-level Win32NET calls. This approach aims to enhance antivirus evasion capabilities in response to the development of anti-detection techniques. Experimental data and illustrative examples further demonstrate:

1) Employing the low-level Win32NET call method for antivirus evasion operations proves to be more effective in circumventing the detection of commonly used domestic antivirus software, thereby reducing the risk of detection for the evasion target.

2) Comparative experiments on multiple evasion techniques based on the Win32NET low-level call method indicate the superiority and advanced nature of this approach through tested data and calculated probabilities. The proposed method effectively utilizes Win32NET's low-level interface for executing commands and code, code obfuscation, code encryption, and other evasion techniques, demonstrating a high success rate in evasion.

At this stage, relying on the common kill-free methods in the success of the kill-free method, there is still room for improvement. Combined with the Win32NET underlying call method proposed in this paper, the kill-free bypass of the bottom layer of the code will help to further strengthen the kill-free method.

However, with the continuous development of information technology, the kill-free technology based on the bottom layer of the code system is not strong and advanced enough, the hardware Trojan horse, and the hardware Trojan horse, how to combine with the call-free method of the Win32NET module, how to combine the kill-free bypass method based on the Win32NET module call at the bottom of the code and the system bottom with the hardware Trojan horse.

This will also become the next research goal.

ACKNOWLEDGMENT

Huizhou Education Science Research Project, 2022hzzjkt36; Special Topic on Scientific Research in Higher Education, 2023GXJK938.

BIBLIOGRAPHY

1. Jiang Xiaojing, Wei Yifei. Research and exploration on detection and prevention of unknown Trojan virus [J]. China Financial computer, 2023 (8): 88-90.
2. Liao Danzi. Pluralistic non-traditional security threats: network security challenges and governance [J]. International Security Research, 2014 (3): 15.DOI:10.3969/j.issn.1004-3489.2014.03.002.
3. Ji Tiantian, Fang Binxing, Cui Xiang, Wang Zhongru, Gan Ruiling, Han Yu, Yu Weiqiang. Research progress on attack and defense of malicious code empowered by deep learning. Journal of computer Science. 2021: 44 (4): 669-95.
4. Kanaker H, Karim NA, Awwad SA, Ismail NH, Zraqou J. Trojan Horse Infection Detection in Cloud Based Environment Using Machine Learning. International Journal of Interactive Mobile Technologies. 2022 Dec 15;16(24).
5. Li ZH, Wang L, Xu J, Yang Y, Al-Amri M, Zubairy MS. Counterfactual trojan horse attack. Physical Review A. 2020 Feb 27;101(2):022336.
6. Labafniya M, Picek S, Borujeni SE, Mentens N. On the feasibility of using evolvable hardware for hardware Trojan detection and prevention. Applied Soft Computing. 2020 Jun 1;91:106247.
7. Pan Z, Mishra P. Design of ai trojans for evading machine learning-based detection of hardware trojans. In2022 Design, Automation & Test in Europe Conference & Exhibition (DATE) 2022 Mar 14 (pp. 682-687). IEEE.
8. Pan Z, Mishra P. Ai trojan attack for evading machine learning-based detection of hardware trojans. IEEE Transactions on Computers. 2023 Mar 2.
9. Doan BG, Abbasnejad E, Ranasinghe DC. Februus: Input purification defense against trojan attacks on deep neural network systems. InProceedings of the 36th Annual Computer Security Applications Conference 2020 Dec 7 (pp. 897-912).
10. Afianian A, Niksefat S, Sadeghiyan B, Baptiste D. Malware dynamic analysis evasion techniques: A survey. ACM Computing Surveys (CSUR). 2019 Nov 14;52(6):1-28.
11. Xu X, Wang Q, Li H, Borisov N, Gunter CA, Li B. Detecting ai trojans using meta neural analysis. In2021 IEEE Symposium on Security and Privacy (SP) 2021 May 24 (pp. 103-120). IEEE.
12. Gao Y, Xu C, Wang D, Chen S, Ranasinghe DC, Nepal S. Strip: A defence against trojan attacks on deep neural networks. InProceedings of the 35th annual computer security applications conference 2019 Dec 9 (pp. 113-125).
13. Yao Xiaoyu, Ma Hui, Lian Zhe. Research on kill-free method based on feature decomposition [J]. Information Network Security, 2012 (4): 4.DOI:10.3969/j.issn.1671-1122.2012.04.012.
14. Lin Cong, Hei Xiali. Analysis of camouflage and anti-killing technology of Trojans [J]. Computer and Modernization, 2009 (1): 3.DOI:10.3969/j.issn.1006-2475.2009.01.008.
15. Ren Hao, Liu Minchao. Research on hiding and Discovery Technology of Trojan Horse virus [J]. Chinese Digital Medicine, 2019, 14 (6): 76-78.