Transactions on Computer Science and Intelligent Systems Research
ISSN: 2960-1800, eISSN: 2960-2238 | Volume 3
CECS 2023

Exploring the intersection of network security and database communication: a PostgreSQL Socket Connection case study

Shengke Ye, Kaiye Dai, Guoli Fan *, Ling Zhang, Zhihao Liang

City College of HuiZhou, HuiZhou, China

*Corresponding author e-mail: 3040221437@qq.com

Abstract. In this study, the network security of PostgreSQL database using Socket connection is deeply analyzed. By exploring Socket connections established by PostgreSQL over TCP, we find potential security threats and vulnerabilities during data transmission, which may expose database systems to network attacks such as unauthorized access and data leakage. In order to assess these security risks, this study simulated a variety of network attack scenarios, especially the implantation and detection of Webshell, to reveal the vulnerability of PostgreSQL to such network threats. Especially in defending against complex and changeable cyber threats such as Webshell attacks, this research also uses machine learning and artificial intelligence techniques to improve the automation level of security threat detection and response. These technologies can help identify complex attack patterns and improve resilience to emerging threats, thereby enhancing the overall security of PostgreSQL databases.

Keywords: Network Security Vulnerabilities; PostgreSQL and Socket Connections; Webshell Detection and Attacks.

1. Introduction

In 2022, the global cybersecurity situation is still at a high operating state, and the severity is far greater than before. Two years into the deadly and devastating global epidemic, we have not only seen more and more attacks, but also more and more data breaches. The report pointed out that in 2022, Rising's "cloud security" system intercepted a total of 73.55 million virus samples, down 62.19% from the same period in 2021, and the number of virus infections was 124 million. There were 45.15 million new Trojan viruses, the largest type of virus, accounting for 61.39% of the total number. Among them, 579,200 ransomware samples were infected 194,900 times; The total number of mining virus samples was 2.61 million, and the number of infections was 797,500. In addition, 1,520,500 mobile phone virus samples were intercepted, and the virus types were mainly information theft, remote control, malicious charge deduction, and tariff consumption, as shown in Figure 1.

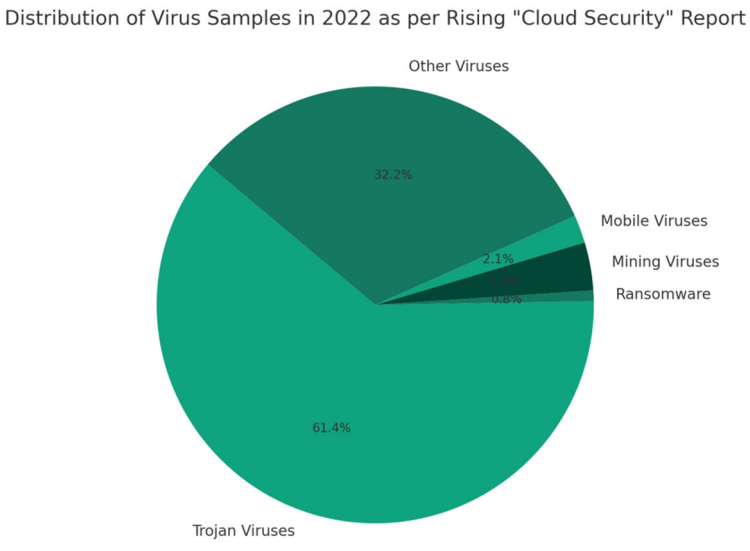

Figure 1. Data analysis

From this, we can conclude that the grim trend of cyber security, the battle between attackers and defenders will be further worsened by increasing complexity, showing a stronger academic game, and the confrontation between technologies

In March 2019, Microsoft discovered that its Exchange Server was infected by a webshell called China Chopper, which allowed an attacker to take control of the server remotely and execute arbitrary commands. Microsoft said that this is a hacker organization supported by Iran, the purpose is to steal sensitive data and emails, you can find this is a very serious topic, here is a brief summary of the characteristics of Webshell, Webshell is through the server open port to obtain some permissions of the server. Put malicious files into the web server, that is, often said "back door", after which you can carry out file management, database management, remote command execution, rights and other malicious operations.

This study will discuss the idea and process of using convolutional neural network model for Webshell detection. In this process, the most important thing is the quantity and quality of samples, without high-quality samples and continuous threat intelligence, it will be difficult to keep up with the rapid development of the high era [1]. It also extends a security analysis problem of network traffic, PostgreSQL database uses a bypass of native communication Socket connection, and exploits the original vulnerability to perform a combination escape, providing a new perspective for analyzing network traffic offline analysis traffic. [2]

The previous part introduces some ideas of attack detection, how to quickly understand and block an attack means, if the interception is not effective, the attack is successful, and a large range of horizontal diffusion, what methods will be used to locate and trace the entire attack process? At this point, various logs should be reviewed to determine whether the webshell file generated by the attacker is a program based on which data layer and which user permissions (e.g. : MySQL, Apache, etc.), and linkage network traffic to determine the time of abnormal traffic fluctuations, such as SQL injection attacks and Web Shell use left traces on the Web server. In this case there are some very important log files that can be used as forensic material [3-4].

The academic significance of this paper lies in the study of webshell Trojan [3] in the field of network security in information technology, the escape thinking in the combination environment mentioned in this paper and the construction of PostgreSQL native communication. In practical sense, this paper can improve network security defense capabilities, design and propose experimental defense measures more effectively, and protect users' privacy and security more reliably

2. Methodology

2.1. Weaknesses of static analysis

When using regular expressions, the biggest advantage of this approach is its simplicity. This command is used to delete files such as images in the web directory. However, the biggest drawback of regular expressions is that they only check for critical hazard functions. As shown in Figure 2, the common Webshell execution process is as follows: external data is passed into the PHP execution framework, the data is transferred, the corresponding dangerous functions such as (exec(), shell_exec(), system(), passthru()) are found, and then the command is executed. Finally view the corresponding permissions of the website. In the case of regular expressions, if we transform the key function slightly in some way, the regular expression will not be able to kill. Moreover, the false alarm rate of this method is also very high, so the detection and killing effect of this method is not ideal. If it is necessary to improve the detection accuracy, the novel Webshell detection system MSDetector[3] can be used, or a pollution function should be judged in the process of pollution tracking [5].

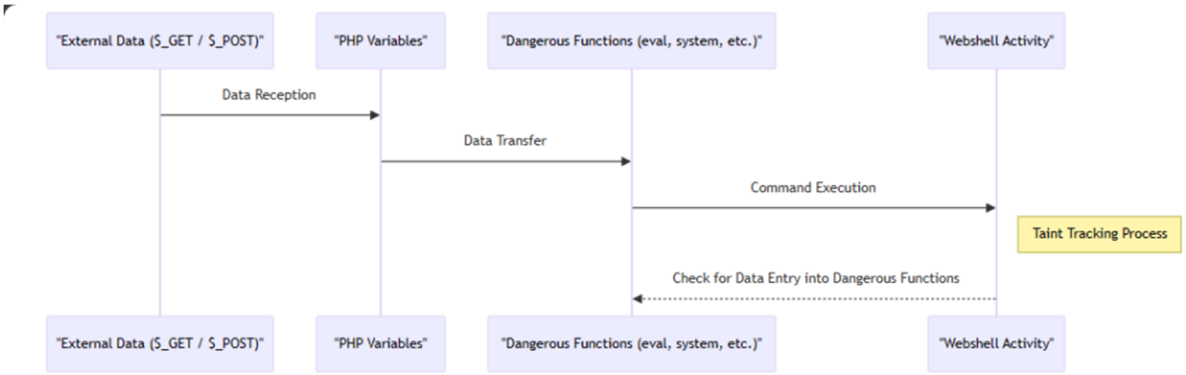

Figure 2. PHP Webshell execution flow

2.2. Weaknesses in sandbox execution

Dynamic sandbox and static analysis have little difference in process. The main difference is that, with spot-tracking technology, the sandbox enables the traceability of user input and the actual execution of object code. At the same time, the sandbox can also collect real-time behaviors of samples and label them, such as command execution backdoor (HEUR.WebShell.Exec), Chinese kitchen knife (HEUR.WebShell.Chopper), etc., so as to make more accurate judgments [6].

Obviously, the sandbox bypass method is far less than static analysis, the main reason is that the sandbox really performs the parts of the static analysis that are not analyzed in place, and the code is really executed, making the spread of the stain extremely difficult. But this sandbox uncontrollable behavior is too much, if there is a specific vulnerability in the sandbox, or after a function that is not disabled is Fuzzing out, the consequences are extremely serious, this will be a nuclear bomb level vulnerability, the virtual environment escapes to the real environment

As shown in Figure 3, Sandboxing technology is a security mechanism used to isolate running programs. Its purpose is to restrict the access rights of untrusted processes or untrusted code running. The detected code is put into a sandbox in a virtual environment, and the virus attacks are changing with each passing day. It solves this kind of problem. The sandbox can isolate suspicious files or programs inside the sandbox and induce suspicious files or programs to run in the sandbox environment. As it runs, the sandbox checks the status of each command under activation and looks for any suspicious behavior, such as self-copying, overwriting files, and other actions that are common in viruses. If a high-risk action is detected, it can be judged as a dangerous procedure

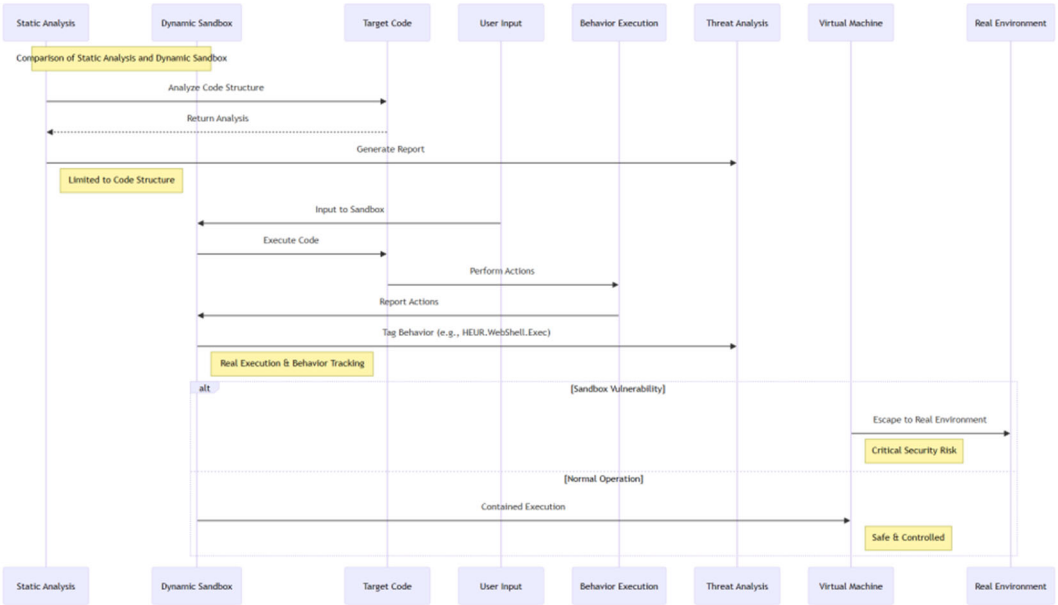

Figure 3. Sandbox execution process

2.3. PostgreSQL Socket communication

One of PostgreSQL's communications is socket communication, which uses the unix_socket_directories parameter to specify the directory where the server listens to Unix domain sockets connected by client applications, in addition to the socket file itself named.s.pgsql.nnnn. Where nnnn is the port number of the server, a normal file named.s.pgQl.nnn.lock will be created in each unix_socket_directories directory. By using its native socket communication features, it can bypass some WAF of network detection [3-4], which shows that the native socket communication features can make PostgreSQL executive users not subject to the constraints of traditional network security policies. In common scenarios, attackers can obtain the permission of a rebound Shell by means of combination, from which they can evade traditional network security monitoring and common security measures to a large extent [5-6]. With this technology, they can control the system by crossing the data layer and using PostgreSQL database users to execute commands as a medium.

At the same time, an attacker may adopt a local Socket (Unix Socket) as an inter-process communication tool. This communication method realizes direct communication between different processes by pointing to a file on the host (essentially a virtual device), and its speed is better than that of conventional network communication. Such communication mechanisms may not be detected by standard network monitoring tools, increasing the risk of data theft or system control. Therefore, the research on database security and interprocess communication is particularly important in preventing such complex network attacks. The key strategy to prevent such attacks is to strengthen the monitoring of Unix Socket communication mechanism and the management of database user permissions.

In actual operations, you can use the psql -h /var/run/postgresql/ -U postgres command to implement Socket communication with postgresql, as shown in Figure 4. This point in the academic discussion and research, for understanding and preventing complex network attacks to give a new perspective, especially in the strengthening of database interaction and inter-process communication security.

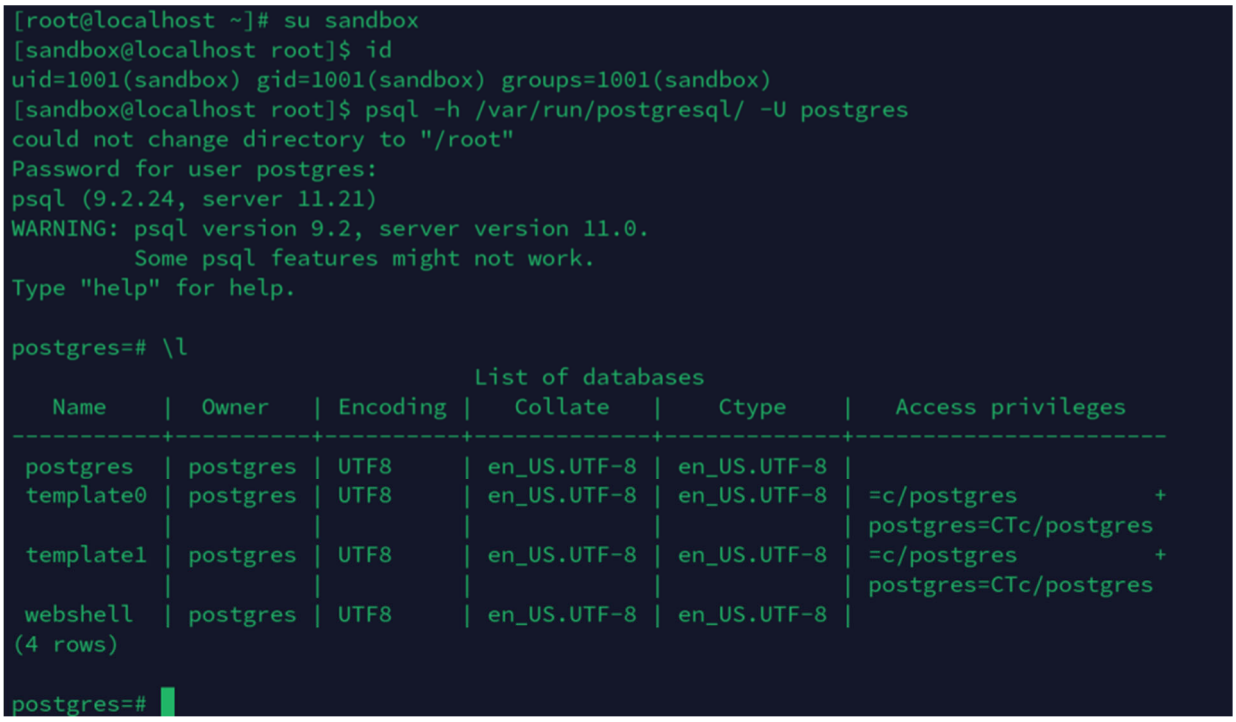

Figure 4. Successful communication

3. Related work.

3.1. Webshell data set

webshell usually refers to the use of asp, jsp, php, py, pl scripting languages to manage the web server tool, also called webadmin. webshell can be used to upload and download files, view databases, and call system commands, so it is often used by hackers and carries out a series of intrusion operations on the server, which has the characteristics of great threat and strong concealment.

This project collected 160 webshell black samples of Github projects and a large number of open source php, jsp, asp and java projects as white samples, 2944 black samples and 11945 white samples after reloading. CountVectorizer and TfidfTransformer are used for eigenvector processing of n-gram samples, and multi-layer neural network, XGBoost and naive Bayes are respectively used for training. MLPClassifier model performs better. MLPClassifier model performs better.

For practical application and further verification of the effectiveness of the model, only a set of white samples from phpcms and 2000 black samples were used for training and testing. This experimental design helps to focus on the model's performance on a specific data set as shown in Figure 5.

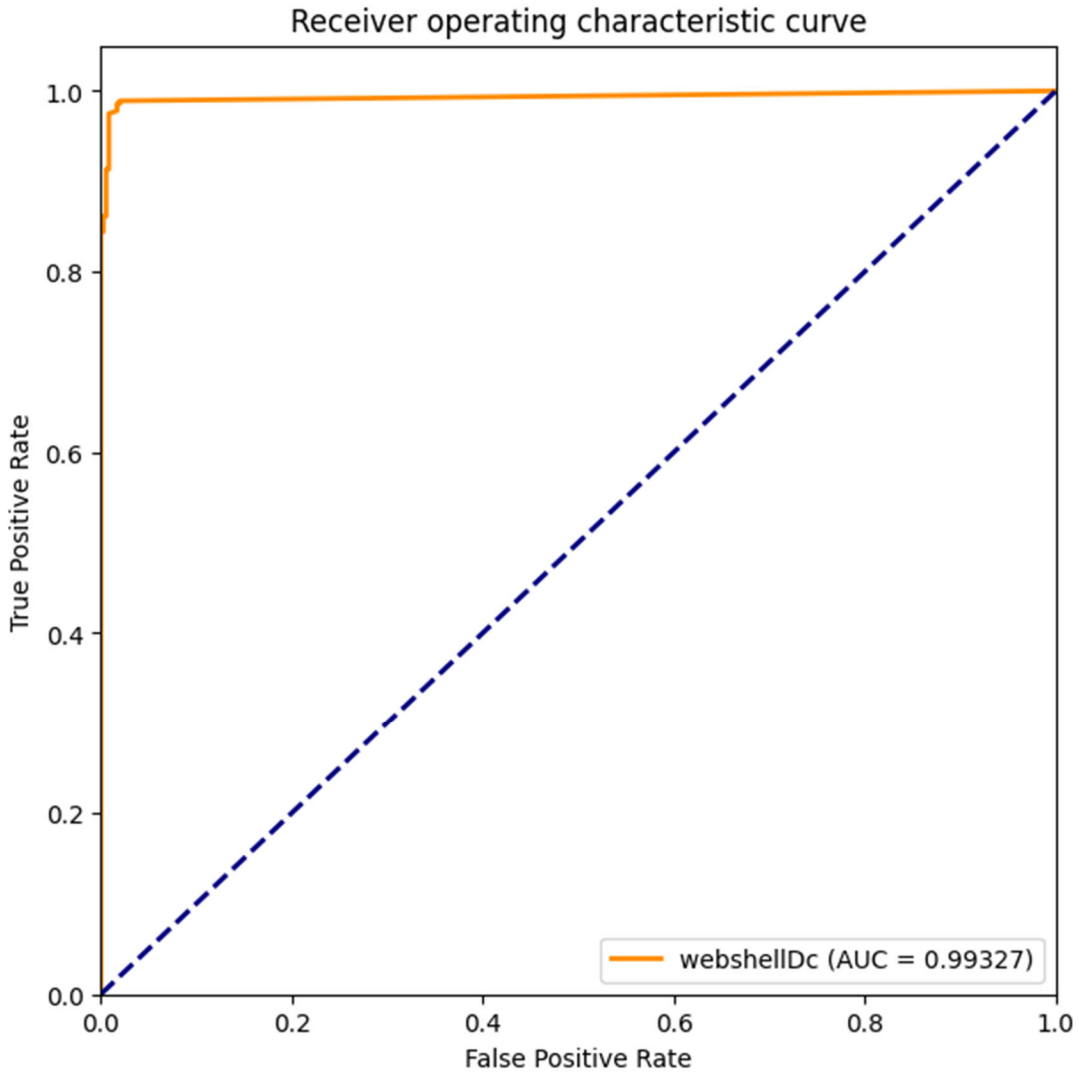

Figure 5. Performance analysis

According to the above model (refer with: Table 1), the detection effect is good. A single detection can detect webshell, but the risk of webshell harm is still very large. However, even the error of 0.00673 cannot be ignored, and other Webshells may avoid detection from it as a springboard for further attack

Table 1. Data analysis

Item	Description
Model Name	webshellDc
AUC Value	0.99327
Meaning	Area Under the Curve
ROC Curve	Depicts the model's True Positive Rate (TPR) and False Positive Rate (FPR) at various thresholds
True Positive Rate (TPR)	The model's ability to correctly identify positives (Webshell)
False Positive Rate (FPR)	The frequency at which the model incorrectly identifies negatives (non-Webshell) as positives
Performance Evaluation	Higher AUC values indicate better model performance
Ideal AUC Value	1 (Perfect Classifier)
Random Classifier AUC Value	0.5
Model Performance	An AUC value of 0.99327 indicates excellent performance of webshellDc in distinguishing between Webshell and non-Webshell

3.2. PHP and PostgreSQL communication and escape

In this study, an escape method [8] combining CVE vulnerability [7] and PostgreSQL native communication Socket is proposed to avoid traditional interception and permission promotion. The core of this method is to use the communication capability of PostgreSQL database, as shown in Figure 6. PostgreSQL is a powerful object relational database management system (ORDBMS) that uses communication escape technology to hide interactions with the database, thereby bypassing traditional network and host-level monitoring interception systems at the underlying level. The COPY TO/FROM PROGRAM function is added in 9.3. This feature allows the superuser of the database, as well as any user in the pg_read_server_files group, to execute operating system commands. In this way, the malicious activity is carried out at the database level rather than through standard HTTP or Web requests, greatly reducing the likelihood of detection. It allows an attacker to execute arbitrary operating system commands without triggering a regular security alert as shown in Figure 7.

Therefore, our approach emphasizes the importance of integrated understanding and defense against attack means in cybersecurity. By deeply analyzing and exploiting the interaction between database communication patterns and network monitoring strategies, we can more effectively identify and block such sophisticated and covert attack strategies. This approach not only provides a complement to existing security measures, but also provides a new perspective and strategy for defending against advanced persistent threats (APTs) in the field of cybersecurity.

```php
<?php
  $input = "\0\0\0T\0\3\0\0user\0postgres\0database\0postgres\0application_name\0psql\0client_encoding\0UTF8\0\0";

$fp = stream_socket_client("unix:///tmp/.s.PGSQL.5432", $errno, $errstr, 30);
fwrite($fp, $input);
$res = fread($fp,1024);
$salt = substr($res, 9);
$input = "p\0\0\0(md5".md5(md5('a'.'aa').$salt)."\0";
fwrite($fp, $input);
$res = fread($fp,1024);
$input = "Q\0\0\0\244DROP TABLE IF EXISTS cmd_exec;CREATE TABLE cmd_exec(cmd_output text);
COPY cmd_exec FROM PROGRAM 'sh -i >& /dev/tcp/xxxx/xxxx 0>&1';SELECT * FROM cmd_exec;\0";
fwrite($fp, $input);
$res = fread($fp,1024);
var_dump($res);
$fp = stream_socket_client("unix:///tmp/.s.PGSQL.5432", $errno, $errstr, 30);
fwrite($fp, $input);
$res = fread($fp,1024);
$salt = substr($res, 9);
```

Figure 6. Performance analysis

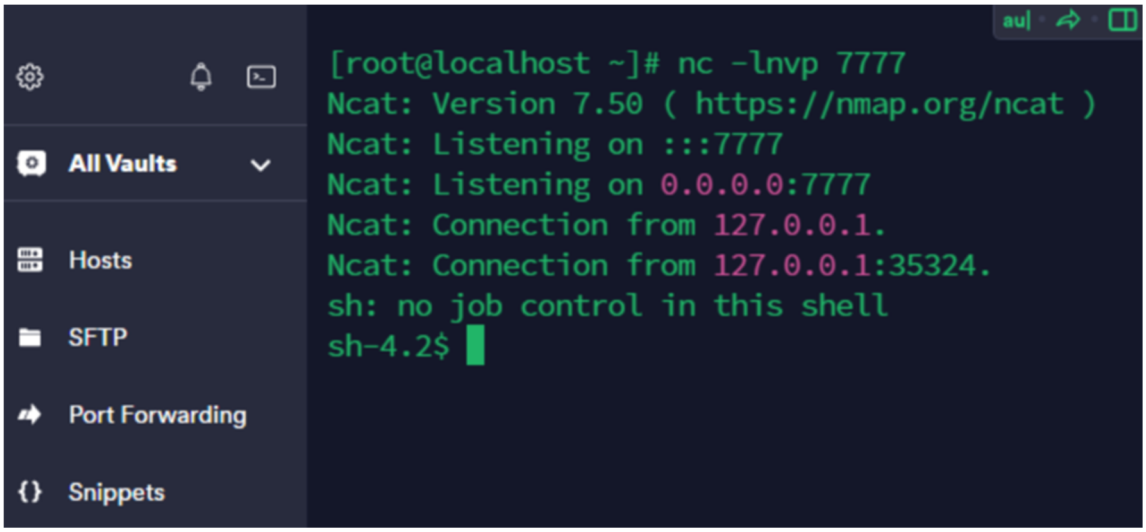

Figure 7. Results

At present, the detection on the market is too simple and requires cumbersome data to support a judgment mechanism and the underlying detection. The new perspective found in this paper is shown in Figure 8. PHP first upload the configuration file and the payload of CVE-2019-9193 to carry out a combination and binding. As shown in Figure 8, it can be seen from the machine learning webshell experiment results that the above attack methods can bypass detection. On top of that, traditional endpoint security tools cannot detect or eliminate advanced threats that slip past them. As a result, these threats can lurk and roam the network for months, gathering data and identifying vulnerabilities in preparation for launching ransomware attacks, zero-day exploits, or other large-scale cyberattacks.

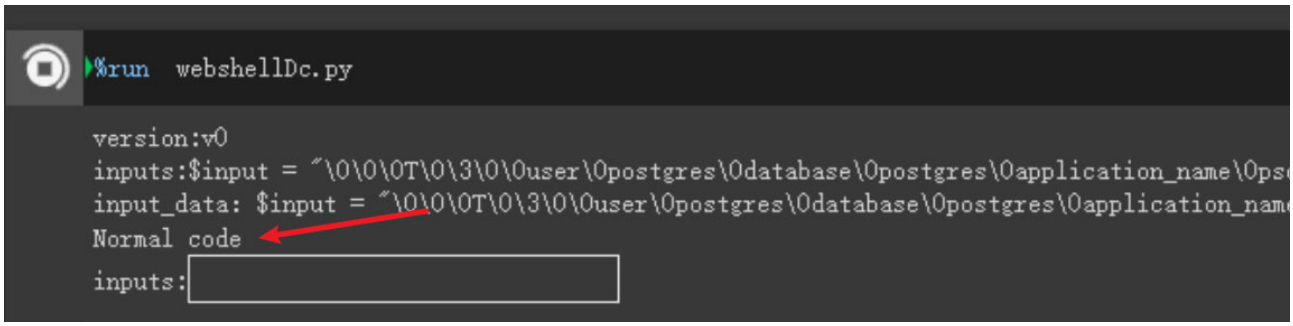

Figure 8. Bypasses static detection

As shown in Figure 9, we need to jump out of the limited thinking of PHP first. In the whole scene, only one structural point needs to be used, which is actually a client that can communicate with PostgreSQL. At the level of network communication, as long as it can establish communication with uninx socks and transmit data, it is enough. Back to the structure of PHP, socks access can be established and data can be transmitted according to the PostgreSQL communication protocol. Figure 6 shows a legitimate PostgreSQL client. Therefore, this study constructs the PostgreSQL communication method according to its features. After identity verification, the attacker can interact with PostgreSQL.

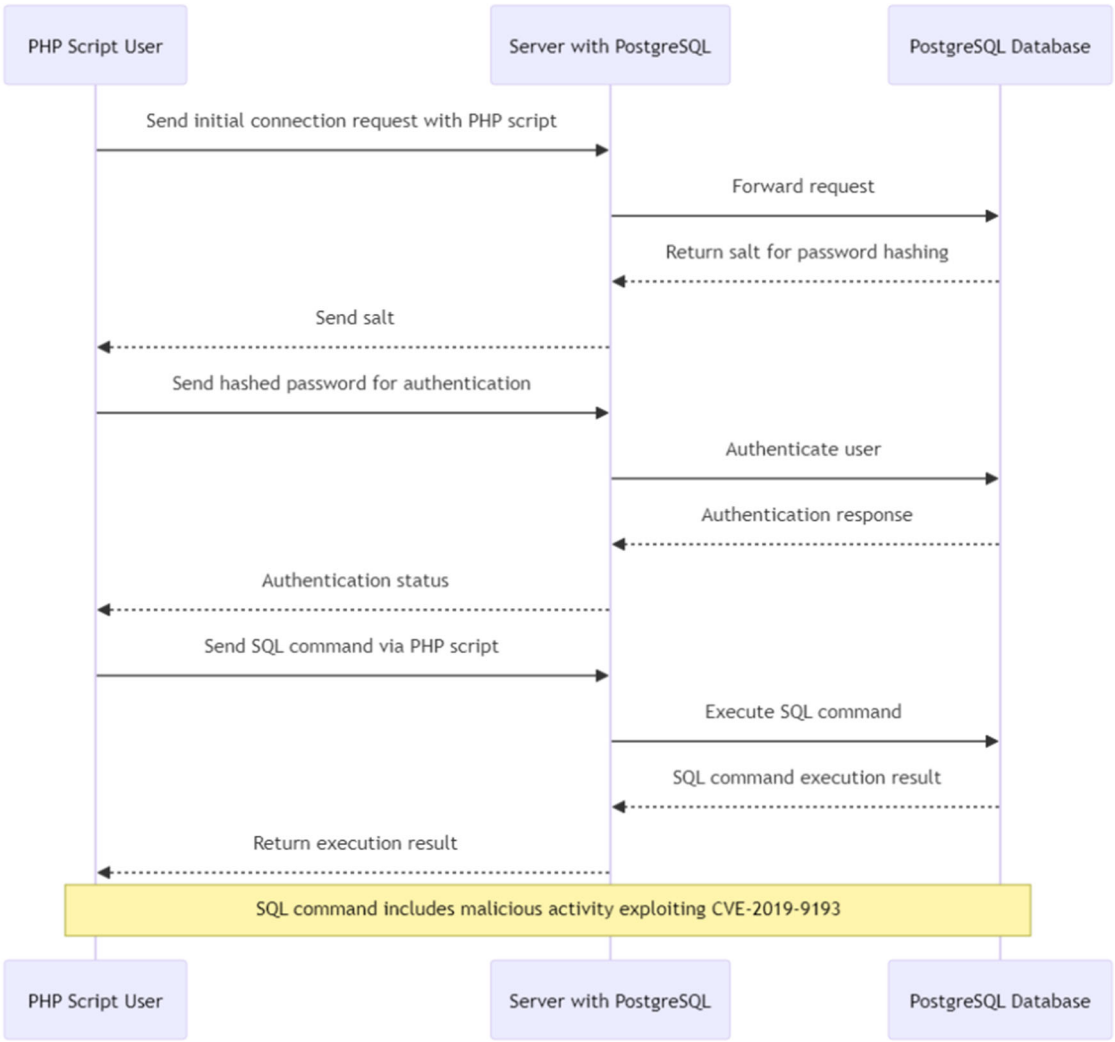

Figure 9. Escape execution flow

4. Conclusion

This paper studies webshell escape. By modifying configuration files, this research method can execute malicious operations without triggering network monitoring alarms under common interception conditions. By using PostgreSQL's CVE vulnerability and native socket communication method, this research can execute malicious codes in combination with configuration files. It also bypasses network-based monitoring, as well as common webshell interception, as shown in Figure 8. The communication method in this paper, by using the Unix domain socket communication protocol and characteristics, can hide the malicious interaction with the database, thus avoiding detection.

In this study, through literature research and data set model testing, it is found that static analysis and sandbox testing have certain defects, especially in the scenario involving socket communication, there are obvious defects:

1. Disadvantages of static analysis: Very limited context understanding Static analysis is usually only able to detect known patterns or specific features in the code, and it is difficult to understand and deal with complex behavioral patterns. Static analysis fails to detect dynamic behavior at runtime, such as in-memory operations and real-time data interactions. Through code obfuscation and encryption, malicious code can easily bypass static analysis detection

2. Disadvantages of sandbox detection: Environment restrictions The sandbox environment may be different from the actual operating environment, resulting in some malicious behaviors that cannot

be triggered in the sandbox. Attackers can exploit specific weaknesses in sandbox environments to design malicious code that can escape sandbox detection. To attack more stealthily

According to the characteristics of socket communication escape, in order to better improve the accuracy of security detection, it is recommended to adopt comprehensive security policies, and combine the code logic analysis of deep behavior, machine learning technology, multi-level security defense system and the method of constantly updating to adapt to the attacker's means. The behavior of socket communication is analyzed quantitatively by monitoring network flow in real time, so that the complex attack mode can be identified more accurately. Attention is paid to the continuous update and real-time optimization of security systems, as well as to adapt to new threat means and attack methods, to ensure that security measures keep pace with The Times and can effectively protect network security.

The main defects of this paper focus on several key aspects. First, the proposed escape method relies heavily on specific environmental configurations and conditions, such as specific versions of PostgreSQL database and server Settings, which may limit the general applicability and adaptability of the method. In addition, if the data sets used are not comprehensive enough or are too specific, such data set limitations can cause studies to face the risk of false positives and under-positives in practical applications, that is, to incorrectly flag normal activity as malicious or to fail to detect truly malicious activity. In addition, research may be too focused on specific escape techniques and not sufficiently consider integrated security strategies, such as multi-layer defenses and behavioral analysis, which may affect the effectiveness and comprehensiveness of methods in real-world Settings

References

[1] Z. H. Lv, H. B. Yan, and R. Mei, "Automatic and Accurate Detection of Webshell Based on Convolutional Neural Network," in Cyber Security. CNCERT 2018. Communications in Computer and Information Science, vol 970, X. Yun, Ed. Singapore: Springer, 2019. [Online].

[2] Q. Chen, X. Zhang, Y. Wang, Z. Zhai, and F. Yang, "Applying a Random Forest Approach to Imbalanced Dataset on Network Monitoring Analysis," in Cyber Security. CNCERT 2022. Communications in Computer and Information Science, vol 1699, W. Lu, Y. Zhang, W. Wen, H. Yan, and C. Li, Eds. Singapore: Springer, 2022.

[3] P. M. R. Gumilang and D. W. Chandra, "Implementasi dan modifikasi WebShell untuk monitoring serangan berbasis website," AITI, 2021.

[4] G. Supriyatno, "Searching for Forensic Evidence in a Compromised Virtual Web Server against SQL Injection Attacks and PHP Web Shell," 2018.

[5] B. Cheng et al., "MSDetector: A Static PHP Webshell Detection System Based on Deep-Learning," in International Symposium on Theoretical Aspects of Software Engineering, Cham: Springer International Publishing, 2022.

[6] B. R. Dawadi, B. Adhikari, and D. K. Srivastava, "Deep Learning Technique-Enabled Web Application Firewall for the Detection of Web Attacks," Sensors, vol. 23, no. 4, p. 2073, 2023.

[7] "Protocol Overview," PostgreSQL 8.3 Documentation.

[8] "CVE-2019-9193 Detail," National Vulnerability Database, NIST.