Exploiting Bitcoin Mining Pool for Stealthy and Flexible Botnet Channels

Pengyu Pan[†‡], Xiaobo Ma[†‡], Huafeng Bian[†‡]

[†]MOE Key Lab for Intelligent Networks and Network Security, Xi'an Jiaotong University, Xi'an, China
[‡]Faculty of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an, China
Email: panda5345@stu.xjtu.edu.cn, xma.cs@xjtu.edu.cn, t1071680825@stu.xjtu.edu.cn

***Abstract*—Botnets are used by hackers to conduct cyber attacks and pose a huge threat to Internet users. The key of botnets is the command and control (C&C) channels. Security researchers can keep track of a botnet by capturing and analyzing the communication traffic between C&C servers and bots. Hence, the botmaster is constantly seeking more covert C&C channels to stealthily control the botnet. This paper designs a new botnet dubbed mp-botnet wherein bots communicate with each other based on the Stratum mining pool protocol. The mp-botnet botnet completes information transmission according to the communication method of the Stratum protocol. The communication traffic in the botnet is disguised as the traffic between the mining pool and the miners in a Bitcoin network, thereby achieving better stealthiness and flexibility.**

I. Introduction

A botnet refers to a network composed of infected Internet devices. The infected devices are called bots and are managed by a botmaster. The purpose of this paper is to further improve the stealthy, real-time and flexibility of botnet communication by connecting the mining pool communication protocol Stratum in the Bitcoin network. This paper refers to it as a mining pool botnet. By hiding the botnet in the mining information in the Bitcoin network, the Stratum protocol and the Bitcoin network are used to complete the information dissemination between the botmaster and the bot.

II. METHODOLOGY

This section will introduce the C&C communication and workflow of the mining pool botnet in detail. Including the communication and authentication details of botmaster and bot nodes, as well as how to manage large-scale botnets and ensure the real-time transmission of attack instructions.

A. System Design

Mining pool botnet (mp-botnet) is designed to disguise botnets as mining pools and communicate through the mining pool protocol Stratum. All nodes contained in it are divided into two types: bullet-proof bot and client bot. Bullet-proof bot is a host with a static public IP address can receive and communicate with a connection request initiated by a remote client to act as a server in the botnet and client bot refers to a host that has an internal IP but cannot accept access from outside to inside. Bullet-proof bots form a P2P network with each other. The client bots only connect to a bullet-proof bot for information exchange. They use Stratum protocol for communication. When a host is infected, its identity must first be judged to determine whether it is suitable as a bullet-proof bot or a client bot. The working flow chart of the mining pool botnet is shown in Figure 1. And the detailed workflow is as follows:

Step1: prepare a C&C server and some infected devices, and select multiple ones as bullet-proof bots to form the P2P part of the botnet, and the rest as client bot nodes.

Step2: The client bot will continuously infect other devices as new bots and register the new bots. The registration information will be sent to the upper-level bullet-proof bot and the new bot, and the new bot can connect to the bullet-proof bot.

Step3: The bullet-proof bot carries a peer list and stores the addresses of other bullet-proof bots to obtain the latest information. The botmaster only needs to propagate the instructions to any bullet-proof bot to send information to the botnet.

Step4: The botmaster can disguise the C&C server as a bullet-proof bot and hide in the P2P network composed of bullet-proof bots to collect the information spread in the botnet.

Step5: When the botmaster needs to launch an attack on the target, it can send the IP address of the C&C server to any bullet-proof bot. The bullet-proof bot uses the client.reconnect method to gather some of its subordinate bots to the specified IP to receive malicious instructions.

Step6: After completing the attack, the botmaster uses the same method to connect the client bot to different server bots, and continues to lurking in the Bitcoin network.

Step7: If the client bot cannot obtain information from the server bot for a long time, it is considered to be in an unavailable state. At this time, the client bot will select a new server bot from the peerlist to connect and update the peerlist in time.

B. Command and Control

The C&C system of the mining pool botnet is shown in the figure below. The upper layer is the backbone of the P2P botnet composed of bullet-proof bots, and the lower layer is a large number of client bots. The bullet-proof bot constitutes the communication layer of the mining pool botnet, that is, the communication between the botmaster and client bot nodes depends on this. The bullet-proof bot is shown in the following figure. The peer list of bullet-proof bot3 contains

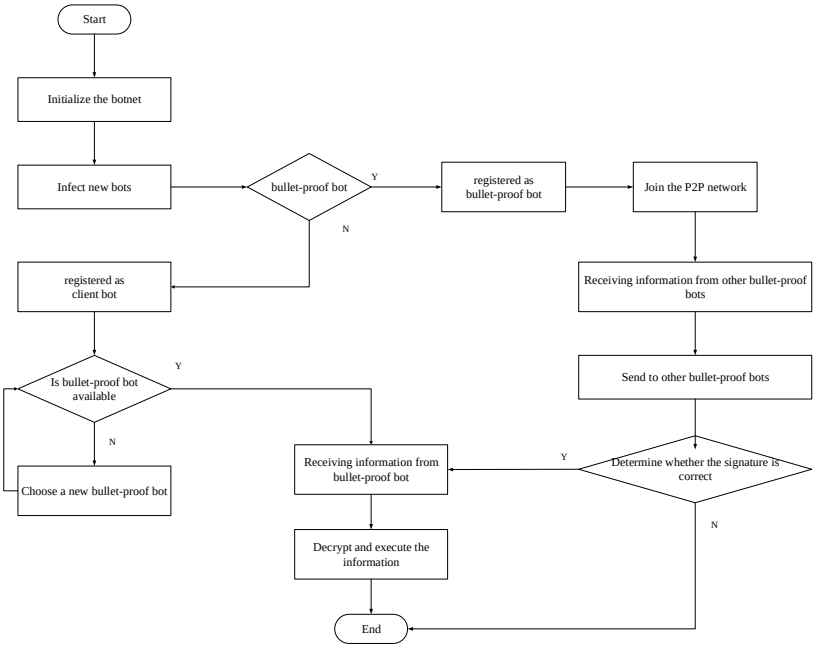

Fig. 1. Work flow chart of mining pool botnet

the IP addresses of bullet-proof bot1, bullet-proof bot2, and bullet-proof bot4, and its subordinate client bot2 also has the same peer list. When the botmaster sends a message to the client bot node, it can send the information to the bullet-proof bot, and the bullet-proof bot will propagate it to other bullet-proof bots through the peerlist and pass it to the client bots of its subordinates. When the client bot sends information to the botmaster, it will send the information to the upper-layer connected bullet-proof bot, which will be propagated by the bullet-proof bot. The botmaster only needs to pretend to be a bullet-proof bot to receive the information sent by the bot. C&C architecture is shown in Figure 2.

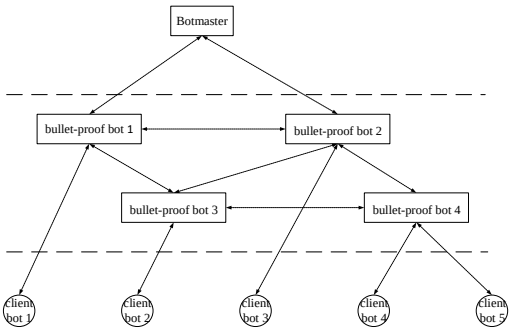

Fig. 2. C&C architecture diagram of the mining pool botnet

III. CONCLUSION

This paper mainly discusses the use of Stratum protocol to build a new type of botnet to improve the concealment of botnets. In future work, we will focus on relevant experimental tests to further optimize and improve the methods proposed in this paper.

ACKNOWLEDGMENT

This work was supported in part by National Natural Science Foundation (61972313), Postdoctoral Science Foundation (2019M663725, 2021T140543), the Fundamental Research Funds for the Central Universities, and CCF-NSFOCUS Kun-Peng Research Fund, of China. Xiaobo Ma is also an XJTU Tang Scholar supported by Cyrus Tang Foundation.

REFERENCES

[1] M. Baden, C. F. Torres, B. B. F. Pontiveros, and R. State, "Whispering botnet command and control instructions," in *2019 Crypto Valley Conference on Blockchain Technology (CVCBT)*. IEEE, 2019, pp. 77–81.

[2] S.-C. Su, Y.-R. Chen, S.-C. Tsai, and Y.-B. Lin, "Detecting p2p botnet in software defined networks," *Security and Communication Networks*, 2018.

[3] C. O. Kumar and P. R. S. Bhama, "Detecting and confronting flash attacks from iot botnets," *The Journal of Supercomputing*, pp. 8312–8338, 2019.

[4] S. T. Ali, P. McCorry, P. H.-J. Lee, and F. Hao, "Zombiecoin: Powering next-generation botnets with bitcoin," in *International Conference on Financial Cryptography and Data Security*. Springer, 2015, pp. 34–48.

[5] J. Zhang, R. Perdisci, W. Lee, X. Luo, and U. Sarfraz, "Building a scalable system for stealthy p2p-botnet detection," *IEEE transactions on information forensics and security*, pp. 27–38, 2013.