Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 2795004, 11 pages
https://doi.org/10.1155/2022/2795004

*Research Article*

# Detectable, Traceable, and Manageable Blockchain Technologies BHE: An Attack Scheme against Bitcoin P2P Network

**Jiale Yang**,[1] **Guozi Sun**,[1,2] **Rongyu Xiao**,[1] **and Hansen He**[3]

[1]*School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210023, China*
[2]*Key Laboratory of Urban Land Resources Monitoring and Simulation, MNR, Shenzhen 518000, China*
[3]*Nanjing Jiangshipin Data Technology co. LTD, Nanjing 210019, China*

Correspondence should be addressed to Guozi Sun; sun@njupt.edu.cn

Received 27 March 2022; Accepted 11 July 2022; Published 2 August 2022

Academic Editor: Yin Zhang

As the most successful cryptocurrency, bitcoin has become the primary target of attackers. The security risks existing in bitcoin network (P2P networks) may pose serious threats to itself. It has been proved that network attackers of the autonomous system level could isolate a specific set of bitcoin nodes using prefix hijacking attacks; since this attack achieves bitcoin partition by deleting all data packets of the victim node, it is easy to be discovered by the victim node, and cannot maintain a long-term connection (the partition will disappear after canceling the BGP hijacking) (Apostolaki M et al. (2017)). This paper proposes a new attack scheme—eclipse attack method based on BGP hijacking (BHE). The attack can occupy the network connection of the victim node, and only need to delete part of the TCP handshaking packets of the victim node during the attack, and it makes the attack more hidden and can occupy the network connection of the victim node for a long time. The innovation of the BHE attack is that it can control the peering decision of the victim node by controlling the victim node's internal peer database (new table and tried table) and preventing the victim node from establishing a good connection. It enables the attacker to occupy all network connections of the victim node and become its natural network middleman. We verify the feasibility of the BHE attack through experimental evaluation and demonstrate that an attacker who can launch BGP hijacking may occupy all connections of the victim node within 20 minutes (ignoring the time of traffic diversion). To reduce the attack's impact, the paper provides some countermeasures that can use in practice according to the basic characteristics of the attack.

## 1. Introduction

The essence of the bitcoin system is a decentralized ledger based on the Internet, and the blockchain is the name of this ledger. The bitcoin system does not depend on a centralized entity. All nodes in the system have equal identities and are connected to form a huge p2p network and share the mission of providing network services. Anyone can join this decentralized network through a bitcoin client [1]. Satoshi Nakamoto pointed out that the bitcoin system uses a proof-of-work mechanism to ensure the consistency of the entire blockchain state [2]. Any attacker who attempts to destroy the bitcoin system needs to control more than half of the computing power in the entire network to break this consistency, which undoubtedly guarantees the security of the bitcoin. However, this security is based on the consistency of information. All Bitcoin nodes have the same view of the blockchain and can always receive the same blocks and transactions within a certain period, which requires the bitcoin network to be safe and reliable.

The bitcoin network is a typical P2P network. Each node in the network maintains a long-term connection with multiple peer nodes. Through these connections, nodes exchange blockchain views to synchronize information and maintain the consistency of the blockchain state. The purpose of bitcoin network-level attacks is to control the network connection of the victim node as much as possible so that the victim node cannot receive the latest or even receive the wrong blockchain view [3]. The bitcoin partition attack showed that an autonomous system (AS) with a large number of IP sources can intercept Bitcoin traffic by hijacking interdomain routing and then isolate the selected victim node set from the bitcoin network [4]. Due to the characteristics of BGP hijacking and the way the attack is implemented (dropping the traffic of the victim node), the attack has the defects of not being able to maintain a long-term connection with the victim node and being easy to detect.

Based on the BGP hijacking mechanism and the Bitcoin network mechanism, this paper proposes a new attack method against the bitcoin network, which can occupy the peer-to-peer connection of the victim node, and only needs to delete some TCP handshake packets of the victim node during the attack. Therefore, compared with the Bitcoin hijacking attack, this attack is more difficult to detect and can maintain a long-term connection with the victim node (even if the BGP hijacking is canceled, the partition will not disappear).

Figure 1 provides a general overview of BHE, mainly describing how the attack occupies the peering connections of the victim node (in AS A) (only two connections are shown here). After hijacking the traffic of the victim node, the attacker (AS D) does not indiscriminately delete all packets of the victim node but monitors the network activity of the victim node. When the victim node establishes a new connection (blue dotted line), the attacker blocks the formation of a good connection (A to E) and forces the victim node to the bitcoin nodes (in AS D and F) (the path from the victim node to these nodes contains attacker D) and establishes the connection (solid red line). Eventually, the attacker will occupy all peer connections of the victim node. In this way, D can control the network view of the victim node through ordinary traffic interception and Bitcoin message forgery.

The rest of the paper is organized as follows. The second chapter gives an overview of the bitcoin P2P network and typical Bitcoin network attacks. The third chapter describes the current research status at home and abroad. The paper detailed introduces the BHE attack in the fourth chapter and evaluates BHE attacks in chapter 5. Then, the sixth chapter puts forward some countermeasures to BHE in a targeted manner. Finally, the seventh chapter concludes this article.

## 2. Background

This section first introduces the bitcoin network and then reviews two typical Bitcoin network attacks.

*2.1. Bitcoin Peer-to-Peer Network.* The bitcoin network is a vast p2p network mounted on the Internet, and all nodes in the network are identified by IP addresses. Each Bitcoin node can select up to 8 remote nodes to connect. If the node has a public IP, it can also accept up to 117 incoming connections [5, 6].

During the running of the node, it will store and broadcast the verified node information to maintain the stability of the network. Below, we describe the most noteworthy part of this section: how a node obtains and stores network information and how to select a node to connect.

*2.1.1. Node Information Propagation.* Bitcoin provides two ways to spread node information: DNS seed and addr message. DNS seed is a DNS server that can return the address information of full nodes on the bitcoin network, and it is hardcoded into the source code to help the node joining the network for the first time find the full node [7]. Addr message is a list containing no more than 1000 node information used to relay node information on the network.

*2.1.2. Node Information Storage.* The IP address of the public node is stored in the tried table and new table of the bitcoin node.

The tired table contains 64 buckets. Each bucket stores the nodes that have established outgoing connections and can store up to 64 IP addresses. New table contains 1024 buckets. Each bucket stores the IP addresses that have not successfully established a connection and can store up to 64 IP addresses.

*2.1.3. Peer-to-Peer Connection.* Bitcoin nodes will maintain at most 8 outgoing connections by default. When the node restarts or an outgoing connection is disconnected, the node will establish a new connection. When establishing a new connection, the node will select the new table or tried table with a probability of 1/2 and then randomly select an IP whose group is distinct from other outgoing connections from the selected table to connect.

*2.2. BGP.* BGP (border gateway protocol) is a network protocol used to exchange routing information between networks on the Internet [8, 9]. Generally, it is used to determine the best path to route data between independent networks or autonomous systems. Different autonomous systems need to exchange routing information and inform each other of their IP prefix. When an autonomous system obtains a new IP prefix, it needs to broadcast the routing information to its neighbors, who will further spread the information until the whole network receives and stores the routing information.

In BGP protocol, the authenticity of routing information will not be checked, which means that any autonomous system can publish false routing information, causing other autonomous systems to send traffic to the wrong location [10]. This attack is called BGP hijacking.

*2.3. Bitcoin Eclipse Attack.* The purpose of the eclipse attack is to monopolize the peer-to-peer connection of the victim node and partition the victim node [11]. The early Bitcoin address manager had some vulnerabilities in the new table and the tried table. For example, tried table stores the IP address of the incoming connection, and the node does not check the validity of the IP before inserting the IP into the new table. It allows the attacker to control a botnet or the basic organization to fill the new table and tried table of the victim node with invalid IP and malicious IP, respectively, and finally, occupy all connections of the victim node when the victim node restarts. At present, the bitcoin community has fixed these vulnerabilities.

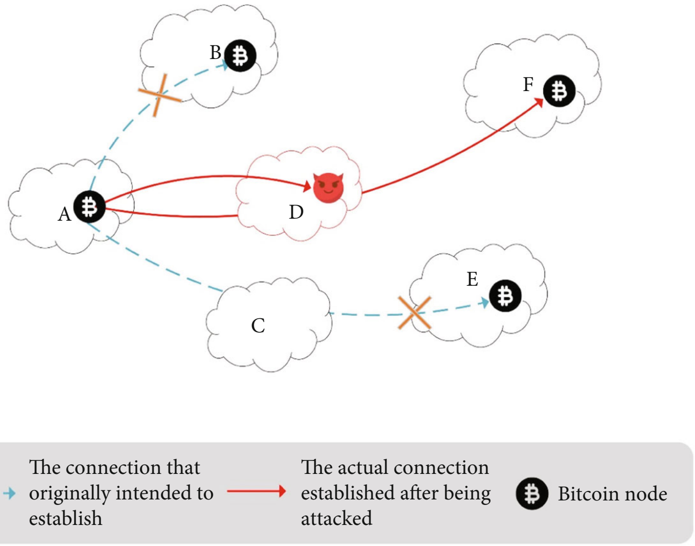
B
F
A
D
E
C
The connection that originally intended to establish
The actual connection established after being attacked
Bitcoin node

Figure 1: Adversary D used the bitcoin partition to prevent the A's legal network connections from forming and then force A to establish connections to D or through D.

*2.4. Bitcoin Partition Attack.* The principle of partition attack is to use the well-known BGP prefix hijacking vulnerability to redirect the outgoing and incoming connections of the victim node set to the attacker's autonomous system (AS), and then control the communication between the victim node set and its neighbor [4]. Attackers generally take the country or nation as the background and rely on their network topology advantages to hijack the traffic of the victim node. If the attacker can hijack all the connections of the victim node, he can isolate the victim node from the bitcoin network.

## 3. Related Work

*3.1. Attacks on Bitcoin Peer-to-Peer Networks.* Chapter 2 describes the bitcoin eclipse attack [11] and the bitcoin partition attack [4], which are most similar to BHE. Among them, the BHE attack and the eclipse attack have similar attack effects (both can monopolize the network connection of the victim node, making the attacker become the natural middleman of the victim node) [11], but BHE does not take advantage of Bitcoin's own vulnerabilities but relies on the attacker's network topology advantage to hijack the connection of victim nodes, so there is currently no patch that can be applied to BHE attacks. Although both BHE and partition attacks utilize the BGP prefix hijacking mechanism, their attack strategies are different. The bitcoin partition attack achieves the partition by deleting the data packets of the victim node, so the attack will form a black hole, which is easy to be discovered by the victim node, and the partition will disappear when the attacker cancels the BGP hijacking; that is, the attack cannot maintain a long-term connection [4]. BHE implements partitioning by controlling the peering decision of the victim node. During this process, only a part of the TCP handshake packets are deleted, which has higher concealment than the partition attack [4], and after the attack is successful, the partitioning will not disappear even if the BGP hijacking is canceled.

Gervais et al. took advantage of the vulnerability that bitcoin nodes only request blocks from the same neighbor each time and successfully delayed the time for the victim node to receive blocks by 20 minutes; the vulnerability exploited by the attack has now been patched [12]. Walck et al. successfully used one full node and two light nodes to carry out delay attacks on the victim node by taking advantage of the vulnerability of the new Bitcoin propagation protocol [13]; compared with BHE and eclipse attacks, the delay attack can only prevent the victim from receiving the latest block and does not allow the attacker to send the wrong block to the victim node. Yves Christian et al. used the defects of the bitcoin's behavior mechanism to realize an eclipse attack on the victim node with a small amount of IP [14], and the attack is based on an eclipse attack, so this attack is difficult to implement in the latest Bitcoin. Muoi tran et al. proposed an attack on the data level of the bitcoin network [15]. The attack does not apply any routing operation. It uses the attacker's network topology advantage to fill the malicious IP to the victim node, slowly affecting the routing table of the victim node. Although the attack can influence the peering decision of the target node, it requires the routing table of the victim node to be heavily populated with malicious IPs, which often takes weeks.

*3.2. Defensive Measures for the Bitcoin Network Attack.* At present, the research results at home and abroad mainly resist network attacks by optimizing the bitcoin network structure. A more efficient network structure means higher network connectivity, which makes it difficult for attackers to control the network view of nodes.

Marcal et al. proposed an adaptive network mechanism that can reduce bandwidth [16]. And they showed that this mechanism would reduce bandwidth consumption by 10.2%, reduce the number of exchanged messages by 41.5%, and not harm transaction submission. Gleb Naumenko et al. proposed a more effective bitcoin transaction relay protocol [17]. The protocol abandoned the original messages flooding transmission mode and adopted the collective coordination method. Otsuki el at. reduce the degree of data redundancy and improve network connectivity by adjusting the ratio of relay nodes in the bitcoin [18]. Bin Zhang proposes an eclipse attack traffic detection method based in a custom combination of features and deep learning [19] and a distributed DDoS-attack traffic detection method based on a cross multilayer convolutional neural network model in the blockchain network layer [20].

## 4. The BHE Attack

In this section, we first introduce the attack model of BHE based on the threat model considered in this paper. Then, we describe the attack process of BHE in detail; in this part, we propose an attack strategy that can control the peering decision of the victim node. The strategy is to control the internal routing table of the victim node and prevent the victim node from establishing a good connection. Finally, we analyze the possible harm caused by BHE.

*4.1. Threat Model.* Similar to the bitcoin partition attack [4], our attacker is a network adversary that controls a single AS, and the attacker's goal is to control all network connections of the victim node. Our victim node is a Bitcoin node with a public IP. Besides, we assume that during the attack, the attacker can hijack all the traffic of the victim node through the BGP prefix hijacking attack and leak point deletion algorithm [4].

*4.2. Attack Model.* Aiming at the problem that Bitcoin hijacking attacks are easy to be detected and cannot maintain long-term malicious connections [4], we propose a new Bitcoin network attack method—BHE based on BGP prefix hijacking mechanism and Bitcoin network mechanism, which can control the peering decision of the victim node, allowing the attacker to occupy all the peering connections of the victim node in a short time, becoming the natural middleman of the victim node.

Our attacker is a malicious AS (AS D in Figure 1), and the victim node is a bitcoin node with a public IP (node in AS A). Since the original peer-to-peer connection of the victim node does not necessarily pass through the attacker, the attacker's attack goal is to force the victim node to establish peer-to-peer connections to some special nodes (nodes in D or F) in order to manipulate these connections.

BHE attacks are mainly divided into two phases: attack preparation and attack execution.

*4.2.1. Attack Preparation.* During the attack preparation phase, the attacker's goal is to collect IP addresses that can be used for the attack, and these IPs have special IP prefixes, and when the victim node establishes outgoing connections to these IPs, the route passes through the attacker. We call these IPs as malicious IPs.

*4.2.2. Attack Execution.* During the attack execution phase, the attacker's goal is to force the victim node to establish an outgoing connection to the malicious IP. The traditional eclipse attack has been proved to be unable to affect the peering decision of the node [11], and the attacker of BHE takes advantage of its network topology, that is, our attacker attackers can simulate different malicious IPs to slowly fill up the victim node's internal peer database, and more importantly, our attackers capture the relevant data packets of the victim node through the BGP hijacking attack and perform operations such as deletion and modification, which can prevent the victim node from establishing a good connection and learning a good IP. As shown in Figure 2, this stage consists of 3 steps: (1) Hijack the victim node's traffic. (2) Fill the victim node's internal peer database with malicious IPs. This step is similar to the Bitcoin eclipse attack [11, 15], and the purpose is to increase the probability of the victim node establishing an outgoing connection to malicious IPs. (3) Observe the network activity of the victim node through hijacked packets, wait for it to establish a new connection, and prevent it from establishing a good outgoing connection (e.g., AS1 to AS2), and to speed up the process, the attacker may force the victim node to restart.

*4.2.3. Attack Properties.* BHE attacks have the following three properties:

(1) Stealthiness: Unlike the Bitcoin partition attack, which indiscriminately discards the data packets of the victim node [4], the attacker of BHE will only delete part of the TCP handshake packets of the victim node to prevent it from establishing a good connection. During the attack, the victim node can communicate normally, so it is difficult to detect that it is being attacked in time

(2) Persistence: The attacker of the BHE attack will eventually occupy all network connections of the victim node (similar to the eclipse attack). After the attack is successful, even if the BGP hijacking attack is canceled, the attacker can still operate the network view of the victim node. And if the attacker has enough computing power, he can launch n confirmed double-spend attacks to the victim node

(3) Efficiency: BHE attackers mainly control the peering decision of the victim node through route hijacking, without waiting for the internal routing table of the victim node to be filled with malicious IP in a large area. Therefore, compared with traditional eclipse attacks, BHE allows attackers to occupy the network connection of the victim node for a relatively short time

*4.3. Attack Process*

*4.3.1. Attack Preparation.* During the attack preparation phase, the attacker needs to collect malicious IPs for the selected victim nodes. The malicious IP is determined by the topological relationship between the attacker and the

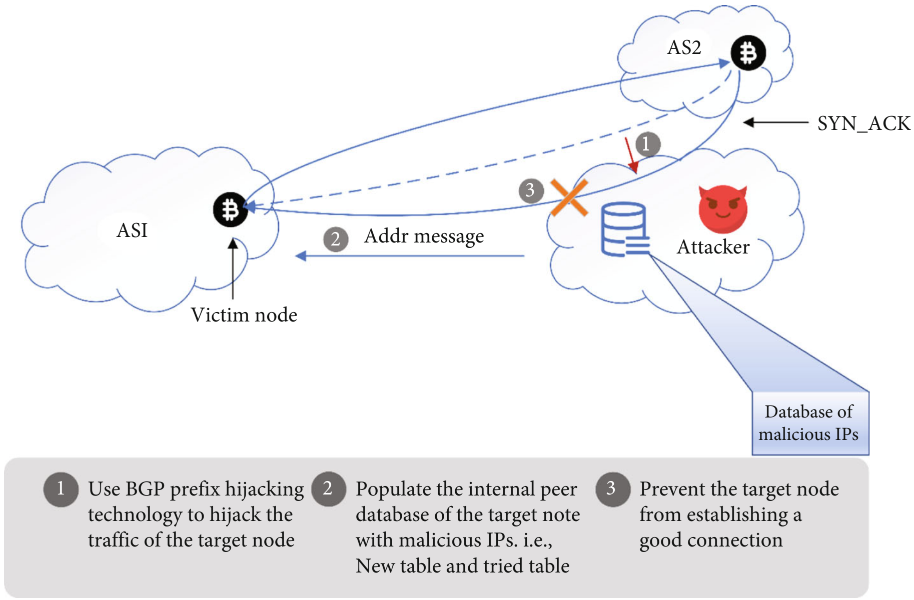
AS2
SYN_ACK
1
3
ASI
2 Addr message
Attacker
Victim node
Database of malicious IPs
1 Use BGP prefix hijacking technology to hijack the traffic of the target node
2 Populate the internal peer database of the target note with malicious IPs. i.e., New table and tried table
3 Prevent the target node from establishing a good connection

Figure 2: Describe the execution flow of a BHE attack, which consists of 3 steps and is able to control the peering decision of the victim node.

victim node, and the corresponding malicious IP can be enumerated by evaluating the interdomain routing state. Research has proven that most AS can easily enumerate a large number of malicious IPs (several million or more) [15]. Please note that the malicious IPs enumerated by the attacker are all valid IPs. Validity here is not meant to represent a real Bitcoin node, but it means that these IPs are legitimate (the network segment is correct).

*4.3.2. Attack Execution.* In the attack execution phase, the attacker will control the peering decision of the victim node based on traffic hijacking, which is embodied by controlling the internal routing table (new table and tried table) of the victim node and preventing the victim node from establishing a good connection. Eventually, the attacker will occupy all network connections of the victim node. Below, we describe our attack strategy in detail.

*(1) Hijack Traffic.* Similar to the Bitcoin hijacking attack, the attacker first transfers the network traffic of the victim node through the BGP hijacking attack, then deletes the leak point in the victim node (if the target is a set) [4], and finally hijacks all the network traffic of the victim node. Note that, unlike the Bitcoin hijacking attack, this step only hijacks the network packets of the victim node, and the manipulation of these packets will take place in the next two steps.

*(2) Dominate the Internal Peer Database of Victim Node. (2)1. How to Dominate New Table.* The running node mainly obtains the IP addresses through the addr messages. The node stores the learned IP addresses in the new table. Due to the characteristics of bitcoin's plaintext transmission, after hijacking the traffic of the victim node, the attacker can capture the addr message sent to the victim node and modify the IP entry in it. The above operations do not modify the structure of the data packet and therefore do not attract the attention of both communicating parties.

When a Bitcoin node inserts an IP address into its new table, it hashes the IP prefix group (IP group) (i.e., the/16 of IPv4 addresses or/32 IPv6 addresses) and the prefix group of the peer relayed that IP (peer group) to determine the bucket for the IP among 1,024 buckets in total; i.e.

$$\begin{aligned} h_1 &= \mathrm{H}(\mathrm{SK}, \mathrm{ip\_group}, \mathrm{peer\_group}), \\ h_2 &= H(\mathrm{SK}, \mathrm{peer\_group}, h_1\%64), \\ \mathrm{bucket\_new} &= h_2\%1024, \end{aligned} \tag{1}$$

where H(·) is the SHA-256 hash function and SK is a secret key of the node. The exact slot for IP in the bucket (which contains 64 slots) is determined by hashing the bucket index and the entire IP address; i.e.,

$$\mathrm{solt\_new} = \mathrm{H}\left(\mathrm{SK}, '\mathrm{N}', \mathrm{bucket\_new}, \mathrm{ip}\right). \tag{2}$$

If the slot is already occupied, a validity check is performed on the existing IP (for example, if the existing IP is more than 30 days old, or if the connection fails several times). If the existing IP has expired, it will be replaced by the new IP being inserted; otherwise, the IP being inserted is ignored. Note that the IP address is stored with the timestamp. If the IP is already in the new table, its timestamp will be updated.

The eclipse attack populates a new table of victim nodes by sending malicious IPs quickly. In an eclipse attack, the attacker will continuously establish connections to the victim node and send addr messages containing a large number of malicious IPs to the victim node through these connections and then slowly wait for the victim node's new table to be filled with malicious IPs [11]. The disadvantage of this method is that it cannot prevent the victim node from receiving good IPs; that is, good IPs compete with malicious IPs.

We propose a new strategy to fill up the new table of the victim node based on BGP hijacking. The strategy makes the victim node unable to learn good IPs, which makes up for the defect that traditional eclipse attack cannot prevent good IP insertion when filling new table [11, 15].

The strategy consists of the following two parts:

(1) The first part is similar to the eclipse attack. The attacker simulates different malicious IPs to repeatedly establish incoming connections to the victim node and then sends addr messages containing 1000 malicious IPs to the victim node

(2) The attacker prevents the victim node from learning a good IP through BGP hijacking. The specific process is described in Algorithm 1; the attacker first filters out the addr messages sent by the ordinary node to the victim node by identifying the destination IP, source IP, and message type of the data packet, then replaces the IP addresses in messages with the malicious IPs in sequence, and finally sends it to the victim node

*(2)2. How to Dominate Tried Table*. In the new version of bitcoin, the IP address in the tried table can only come from the new table, and the tried table cannot be accessed directly from the outside, which is to prevent attackers from inserting malicious IP addresses directly into the tried table. Therefore, ordinary eclipse attack methods are difficult to impact the tried table directly.

The bitcoin node will migrate the IP address in the new table to the tried table in two cases. First, when the node successfully establishes an outgoing connection to an IP in the new table, the IP address will be transferred to the tried table. Second, the bitcoin node will randomly select an IP from the new table every two minutes to establish an outgoing connection called a probe connection. If the probe connection establishes successfully, the selected IP will be transferred to the tried table.

When a Bitcoin node inserts an IP address into the tried table, it needs to perform a series of hash operations on the IP to obtain the index of its bucket and slot.

$$\begin{aligned} h_1 &= H(\mathrm{SK}, \mathrm{ip}), \\ h_2 &= H(\mathrm{SK}, \mathrm{ip\_group}, \mathrm{h}_1\%8), \\ \mathrm{bucket\_tried} &= h_2\%256, \\ \mathrm{solt\_tried} &= H\left(\mathrm{SK}, '\mathrm{K}', \mathrm{bucket\_tried}, \mathrm{ip}\right)\%64. \end{aligned} \tag{3}$$

The strategy of BHE filling the tried table is similar to trickle-down attack; that is, by filling the new table, it slowly affects the tried table [15]. In addition, we have further optimized the attack process based on BGP hijacking, so that the victim node cannot establish a connection to a good IP, which accelerates the rate of IP transfer. This method is similar to preventing the victim node from establishing a good connection and is described in detail in the next subsection.

*4.3.3. Prevent the Victim Node from Establishing a Good Connection.* When a Bitcoin node establishes an outgoing connection to an IP, it needs to perform a TCP three-way handshake. Because the attacker hijacks all network traffic of the victim node, the data packets (SYN or SYN_ACK) used by the victim node to establish an outgoing connection can be captured (e.g., the data packet sent by AS2 to AS1 in Figure 2). If the peer of the victim node is not a malicious IP, the attacker will delete the corresponding packet to prevent the connection from being established.

When the victim node establishes a new outgoing connection due to disconnection or restart, the attacker can prevent it from establishing an outgoing connection to a good IP through the following steps (as shown in Algorithm 2):

(1) According to the source IP, port number (Bitcoin default port number is 8333), and data packet type (SYN data packet) of the data packet, determine whether the data packet is a TCP handshaking packet used by the victim node to establish a new connection, if so, enter the second step. If not, go to step 3 for judgment

(2) If the destination address of the handshaking packet is a malicious IP, then the purpose of the attack is achieved. The attacker will pretend to be that IP to communicate with the victim node. Otherwise, the data packet will be discarded, and the attack will enter the next cycle

(3) The attacker may fail to intercept the SYN packet sent by the victim node, but the attacker may intercept the response packet sent by the legitimate node to the victim node. The identification method is similar to that of 1. If the packet is identified as a response packet (SYN-ACK), drop the packet. Otherwise, forward the packet along the original path to avoid forming a black hole and enter the next cycle

During the attack, the victim node will establish an outgoing connection to malicious IP (such as A to D and A to F in Figure 1), and the attacker will intercept the connection and communicate with the victim node by disguising the malicious IP through source IP spoofing. Ultimately, the attacker will occupy all network connections of the victim node in this way and then partition it. Because the route from the victim node to the malicious IP contains the attacker, even if the attacker cancels the BGP hijacking, he can still intercept the network packet of the victim node and occupy its network connection. In other words, the attacker can occupy the network connection of the victim

```
Input: S=[pkt1…]: hijacked network packets. M: the set of malicious IPs. dp: the ip of victim node
for pkt ∈ S do
    if pkt.ipDst=dp and pkt.ipSrc not in M and pkt.payload=Addr then
        for index in len(pkt.payload) do
        pkt.payload[index].ip = M[index]
        end for
    end if
    send(pkt)
end for
```

Algorithm 1: Modify the IPs in the addr message to malicious IPs to prevent the victim node from learning a good IP.

```
Input: S = [pkt1…]: hijacked network packets. P: the set of malicious IP prefixes. dp: the ip of victim node
for pkt ∈ S do
    if pkt.ipSrc=dp and pkt.dport=8333 and pkt.payload=SYC then
        ipStr ←‘ ’
        ipStr ← Prefix(pkt.ipDst)
        if ipStr in P then
        success(pkt)
    else
        drop(pkt)
        end if
      else if pkt.ipDst=dp and pkt.sport=8333 and pkt.payload=SYC_ACK then
      drop(pkt)
  else
      send(pkt)
  end if
end for
```

Algorithm 2: Selectively deleting TCP handshaking packets forces the victim node to establish an outgoing connection to the victim node.

node for a long time, and even if the BGP hijacking is canceled, the formed partition will not disappear.

Please note that it is relatively hidden to prohibit the victim node from establishing a new connection with a legal IP by discarding the TCP handshaking packet. Because the connection has not formed, the victim node will think that the remote node does not exist and then randomly select an IP from the routing table to connect.

*4.3.4. Occupy Incoming Connections.* Since Bitcoin nodes receive unsolicited incoming connections, and in most cases, incoming connections are very short-lived (e.g., a couple of minutes) [15], the attacker can easily occupy all incoming connections of the victim node; e.g., the attacker can simulate malicious IPs to repeatedly establish a connection to the victim node.

*4.4. Implications of BHE Attack.* BHE is the strengthened eclipse attacks, so the damage for the bitcoin with the eclipse attacks is similar. It destroys the information consistency of bitcoin network, which allow an attacker to control the network view of the victim node and easier to launch the traditional attacks to the victim node, such as the double-spending attack [21–23] and selfish mining [24–27]. In particular, compared to the partition attack [4], BHE supports the n-confirmation double spend; in the scenario of a n-confirmation double spend, the victim node has to wait for the confirmation of n blocks before accepting the current transaction; and since BHE can forge the blockchain view of the victim node by sending fake blocks, an attacker with sufficient computing power can achieve the n-confirmation double spend attack.

## 5. Experiment

We simulate the attack scenario of BHE in the laboratory environment and evaluate the effectiveness of BHE. The experiment configures 3 Ubuntu 18 machines with public IP to simulate the attack environment. Two machines equipped with Bitcoin core (v0.19.1) act as the attack node and the victim node, and the remaining one acts as the proxy of the attacked node.

*5.1. Attack Preparation.* Since BGP hijacking in the public network environment requires a lot of network resources, it is difficult to implement. So we simulate traffic hijacking by the proxy host. And in the experimental environment, we can evaluate and compare different combinations of attack strategies.

Our attack script is configured in the proxy host and implements the ability to flood the victim node routing table with malicious IPs and prevents the victim from establishing a good connection (the attack process in chapter 4). In addition, we implement some additional functions in the script to simulate the execution of the attack more realistically: (1) The traffic sent to the malicious IP is directed to the attacking host through the address translation function (NAT) of the firewall (iptables) to ensure that the malicious IP is always reachable [28, 29], and it is convenient to record the attack result. (2) When the time of flooding the malicious IP address reaches the set attack duration, restart the victim node and prevent it from establishing a good connection. When the attacking host occupies all the victim's outgoing connections, the victim node state is rolled back to before restarting and repeated the above restart operation to record multiple sets of experimental data.

*5.2. Attack Setup.* We evaluate the attack effect of BHE under different attack configurations: (1) The number of malicious IPs, which refers to how many malicious IP prefixes does the attacker collected to fill the routing table of the victim node. (2) The age of the victim node, which refers to the number of running days since the victim first ran.

The validity period of IP addresses in the internal peer database of the bitcoin node is 30 days, and we selected the 4 most representative nodes for experiments (running for 0 days, 10 days, 20 days, and 30 days). We found that by running the node on the public network many times when the node runs for some time (more than a week), its internal database will save IP addresses with various prefixes; that is, for the latter three nodes, it can also make the attack successful without flooding the victim with malicious IPs (experimentally proved this). Research has proved that one hundred IP prefixes can control the victim's internal routing table very well [15]. Therefore, in the experiment, the number of malicious IP prefixes ranges from 0 to 100, and finally, four representative groups of data were selected for analysis.

### 5.3. Experimental Results and Analysis

*5.3.1. The Impact of Node Age on Attack Efficiency.* Figure 3 shows some experimental data under 100 malicious IP prefix configurations, which describes the attack efficiency of the attack against nodes of different ages under different attack duration. The attack duration is the duration of filling the malicious IP to the victim node before the victim node restarts. The attack efficiency refers to the time required to monopolize all outgoing connections of the victim node after the victim node restarts. The data in Figure 3 shows that with the increase of attack duration, the attack efficiency on all nodes is increasing, but under all attack duration, the attack effect on nodes running for 0 days is the best (except when the attack duration is 0, because in this case the node does not store the IP and cannot complete the connection), followed by nodes running 30 days (our script intercepts and deletes the IPs returned by the DNS seed so that the node can only receive malicious IPs in the attack). This is because more malicious IPs will be inserted into the peer database of these two nodes under the same attack duration. Furthermore, since the peer databases of nodes running 0 days only store malicious IPs (our experiments show that malicious IPs account for nearly 100%), the attack works best on nodes running 0 days with little variation over time (except the case where the attack duration is 0).

We experimentally find that for a node running for 0 days, an attacker can always monopolize all its connections with any number of malicious IP prefixes (less than 0.5 minutes) in a very short period of time. This is because, relative to other nodes, the database of 0-day nodes lacks legitimate IPs to compete with malicious IPs. In fact, when the node runs for about 10 days, the IP (legal IP) in the database will tend to a stable value, which we call a strong node. Our subsequent experimental analysis is mainly performed on strong nodes (10, 20, and 30 days nodes), because non-strong nodes (0-day nodes) can always be easily attacked (not dependent on the number of malicious IPs), so it is difficult to show the experimental law under different attack conditions (for example, the more malicious IP prefixes, the higher the attack efficiency).

To better understand the impact of node age on attack efficiency, this article demonstrates the process of filling the routing table of each node with malicious IP in Figure 4. In general, the number of malicious IPs in the routing table of each node is increasing. However, the insertion rate of malicious IP is very slow for nodes running for 10 and 20 days, while nodes running for 30 days are easy to be filled by malicious IP. This is because many IP addresses in the routing table of nodes running for about 30 days are marked as terrible, making it easier for malicious IP to insert into the routing table.

*5.3.2. The Impact of the Number of Malicious IP Prefixes on Attack Efficiency.* Table 1 describes the relationship between the optimal attack efficiency and the number of malicious IP prefixes. The optimal attack efficiency refers to the time threshold at which the time required to monopolize all connections no longer decreases significantly as the attack duration increases. Taking the data in Figure 3 as an example, when the attack duration reaches about 25 minutes, the attack reaches the optimal efficiency, and the optimal attack efficiency of the attack on a node that has been running for 30 days is within 1 minute. The data in Table 1 shows that with the reduction of the number of malicious IPs, the attack efficiency on all types of nodes will reduce accordingly. This is because the smaller the number of malicious IP prefixes, the fewer malicious IPs will be inserted eventually. Among them, nodes running for 30 days are least affected by this rule, because this type of node is more likely to be populated by malicious IPs, so the nodes can also be well-populated while the malicious IP prefix decreases.

*5.3.3. Best Attack Strategy.* We call the attack duration that reaches the optimal attack efficiency for the first time for the optimal attack duration. After reaching the optimal attack duration, the attack efficiency will not change significantly as the attack duration increases. It is not difficult to see from the data in Figure 3 that 25 minutes is the optimal attack duration for that configuration.

The optimal attack strategy refers to the optimal attack duration and attack efficiency combination. When the attack

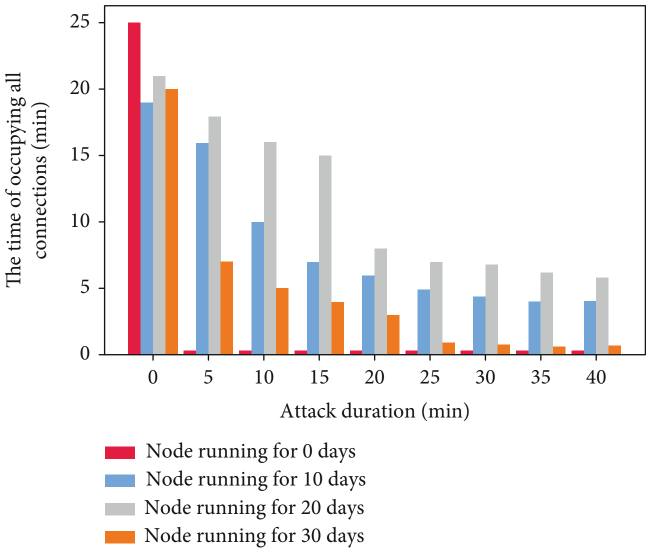

Figure 3: The relationship between attack duration and attack efficiency. Under the same attack time, the attack has the best effect on nodes that have been running for about 30 days.

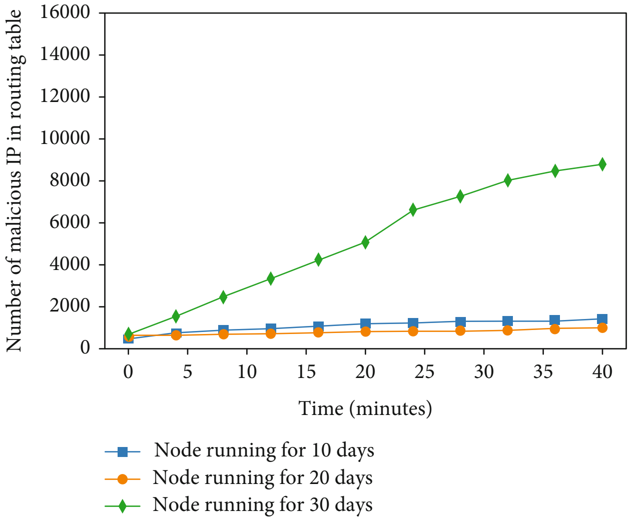

Figure 4: The number of malicious IP addresses in the routing table of nodes running for 30 days increased faster than those running for 10 days and 20 days.

Table 1: For the same type of node, the more malicious IP prefixes, the higher the optimal attack efficiency (the shorter the required attack time). The nodes running for 30 days are the least affected by this rule.

| Node type | Number of malicious IP prefixes | Optimal attack efficiency(min) |
|---|---|---|
| Node running for 30 days | 25 | <6 min |
| | 50 | <5.5 min |
| | 75 | <2.5 min |
| | 100 | <1 min |
| Node running for 20 days | 25 | <17 min |
| | 50 | <14 min |
| | 75 | <7 min |
| | 100 | <6 min |
| Node running for 10 days | 25 | <13 min |
| | 50 | <9.5 min |
| | 75 | <5.5 min |
| | 100 | <5 min |

achieves the optimal attack efficiency, it can monopolize all the network connections of the victim node in the shortest time. At this time, it is most difficult to be found by the victim node. The optimal attack efficiency will be obtained when the attack duration reaches the optimal attack duration. Even if the attack duration increases, the attack efficiency will not significantly improve. Therefore, the optimal attack strategy = optimal attack duration + optimal attack efficiency, when filling the victim node with malicious IP reaches the optimal attack duration of the current configuration, restart the node and occupy its outgoing connection. Take the data in Figure 3 as an example. When the attack duration reaches 25 minutes, stop filling malicious IP, launch a denial-of-service attack to restart the victim node, and occupy all its connections.

## 6. Countermeasure

In this section, we introduce some countermeasures against the BHE attack.

*6.1. Add Connection within AS.* The attack is based on the fact that BGP hijacking can intercept all network connections of the victim node, but BGP hijacking cannot intercept the traffic inside the AS to which the victim node belongs. Therefore, Bitcoin nodes can add an additional connection, which actively connects to the IP in the target AS and can give the node a view of the blockchain first. In this way, even if the attacker controls all other connections, it is not easy to control the network view of the victim node.

*6.2. Diverse Connections.* The nodes inside the mining pool use private protocols to connect, and it is difficult for attackers to capture this traffic. Therefore, when Bitcoin nodes join the bitcoin network, they can join some organizations (such as mining pools) and use private protocols for information exchange, which can effectively resist BHE attacks.

*6.3. Replace the Port.* The attacker can accurately identify the bitcoin traffic of the victim node in a large amount of traffic because most of the bitcoin traffic selects port 8333 by default. Therefore, the bitcoin node can set up several more ports for other nodes to connect, significantly increasing the difficulty for attackers to filter traffic.

*6.4. Record the Time Interval between the Arrival of Adjacent Blocks.* The purpose of the BHE attack is the same as the eclipse attack, which is to prevent the victim node from receiving the latest block or receiving the wrong block. It takes longer than normal for the victim to receive the next block [30]. Research has proved that under normal circumstances, the node will receive the next block within 40 minutes, but in the case of an attack, it is significantly higher than this value [30]. Therefore, Bitcoin can detect the attack by recording the time interval between the arrival of adjacent blocks.

*6.5. Observe the Restart Time of Node.* When a node is restarted after being attacked by BHE, it takes longer to establish 8 outgoing connections than a normal node because the node will experience many failures. Our experimental data shows that under normal circumstances, the node establishes all its outgoing connections within 5 minutes, but in the case of being attacked, most of time it exceeds this value (such as the nodes running for 10 days and 20 days in Figure 3). Therefore, we can judge whether the node is attacked by observing the time required for the node to establish eight outgoing connections after a restart. If the restart time significantly exceeds the average restart time of the node, the node is considered to be under attack. However, if the attacker collects enough malicious IPs and adopts the optimal attack strategy, this method will not apply.

## 7. Conclusion

This paper presents a new attack on the bitcoin network—BHE. This attack can control all network connections of the victim node in a short time. An attacker can use this attack to increase his mining advantage or launch a traditional blockchain attack on the victim node or target mining pool, which destroys the original intention of bitcoin design. The paper implemented the attack model in our experimental environment and proved its powerful destructive power. At the end of this paper, the paper gives some practical measures to mitigate the harm of attack.

## Data Availability

The experimental data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The author declares that there are no conflicts of interest.

## Acknowledgments

The authors would like to thank the anonymous reviewers for their elaborate reviews and feedback. This work was supported by the National Natural Science Foundation of China (No. 61906099) and the Open Fund of Key Laboratory of Urban Land Resources Monitoring and Simulation, Ministry of Natural Resources (No. KF-2019-04-065).

## References

[1] C. G. Akcora, Y. R. Gel, and M. Kantarcioglu, “Blockchain networks: data structures of Bitcoin, Monero, Zcash, Ethereum, Ripple, and Iota,” *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 12, no. 1, p. e1436, 2022.

[2] S. Nakamoto, “Bitcoin: a peer-to-peer electronic cash system,” *Decentralized Business Review*, no. article 21260, 2008.

[3] X. Han, Y. Yuan, and F. Y. Wang, “Security problems on blockchain: the state of the art and future trends,” *Acta Automatica Sinica*, vol. 45, no. 1, pp. 206–225, 2019.

[4] M. Apostolaki, A. Zohar, and L. Vanbever, “Hijacking bitcoin: routing attacks on cryptocurrencies,” in *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 375–392, San Jose, CA, USA, 2017.

[5] F. Franzoni, X. Salleras, and V. Daza, “AToM: active topology monitoring for the bitcoin peer-to-peer network,” *Peer-to-Peer Networking and Applications*, vol. 15, no. 1, pp. 408–425, 2022.

[6] L. Zhang, T. Wang, and S. C. Liew, “Speeding up block propagation in Bitcoin network: uncoded and coded designs,” *Computer Networks*, vol. 206, article 108791, 2022.

[7] W. Yue and L. Junxiang, “The evolution process of blockchain P2P network protocol,” *Computer Application Research*, vol. 36, no. 10, pp. 2881–2886, 2019.

[8] S. Secci, J. L. Rougier, A. Pattavina, F. Patrone, and G. Maier, “Peering equilibrium multipath routing: a game theory framework for internet peering settlements,” *IEEE/ACM Transactions on Networking*, vol. 19, no. 2, pp. 419–432, 2011.

[9] H. S. Alotaibi, M. A. Gregory, and S. Li, “Multidomain SDN-based gateways and border gateway protocol,” *Journal of Computer Networks and Communications*, vol. 2022, 23 pages, 2022.

[10] H. Ballani, P. Francis, and X. Zhang, “A study of prefix hijacking and interception in the internet,” *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 4, pp. 265–276, 2007.

[11] E. Heilman, A. Kendler, A. Zohar, and S. Goldberg, “Eclipse attacks on bitcoin’s peer-to-peer network,” in *24th {USENIX} Security Symposium ({USENIX} Security 15)*, pp. 129–144, Washington,D.C., August 2015.

[12] A. Gervais, H. Ritzdorf, G. O. Karame, and S. Capkun, “Tampering with the delivery of blocks and transactions in bitcoin,”

in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pp. 692–705, Denver, Colorado, USA, October 2015.

[13] M. Walck, K. Wang, and H. S. Kim, "Tendril staller: block delay attack in Bitcoin," in *2019 IEEE international conference on Blockchain (Blockchain)*, pp. 1–9, Atlanta, GA, USA, 2019.

[14] A. E. Yves-Christian, B. Hammi, A. Serhrouchni, and H. Labiod, "Total eclipse: how to completely isolate a bitcoin peer," in *2018 Third International Conference on Security of Smart Cities, Industrial Control System and Communications (SSIC)*, pp. 1–7, Shanghai, China, 2018.

[15] M. Tran, I. Choi, G. J. Moon, A. V. Vu, and M. S. Kang, "A stealthier partitioning attack against bitcoin peer-to-peer network," in *2020 IEEE symposium on security and privacy (SP)*, pp. 894–909, San Francisco, CA, USA, 2020.

[16] J. Marçal, L. Rodrigues, and M. Matos, "Adaptive information dissemination in the bitcoin network," in *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, pp. 276–283, St. Raphael Resort, Limassol, Cyprus, April 2019.

[17] G. Naumenko, G. Maxwell, P. Wuille, A. Fedorova, and I. Beschastnikh, "Erlay: efficient transaction relay for bitcoin," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pp. 817–831, London,-United Kingdom, November 2019.

[18] K. Otsuki, Y. Aoki, R. Banno, and K. Shudo, "Effects of a simple relay network on the bitcoin network," in *Proceedings of the Asian Internet Engineering Conference*, pp. 41–46, Andaman Cannacia Resort and Spa Hotel, Kata Beach, Phuket, Thailand., August 2019.

[19] Q. Dai, B. Zhang, and S. Dong, "Eclipse attack detection for blockchain network layer based on deep feature extraction," *Wireless Communications and Mobile Computing*, vol. 2022, 19 pages, 2022.

[20] Q. Dai, B. Zhang, and S. Dong, "A DDoS-attack detection method oriented to the blockchain network layer," *Security and Communication Networks*, vol. 2022, 18 pages, 2022.

[21] C. Pinzón and C. Rocha, "Double-spend attack models with time advantange for bitcoin," *Electronic Notes in Theoretical Computer Science*, vol. 329, pp. 79–103, 2016.

[22] G. O. Karame, E. Androulaki, and S. Capkun, "Double-spending fast payments in bitcoin," in *Proceedings of the 2012 ACM conference on Computer and communications security*, pp. 906–917, Raleigh, NC, USA, October 2012.

[23] G. O. Karame, E. Androulaki, M. Roeschlin, A. Gervais, and S. Čapkun, "Misbehavior in bitcoin," *ACM Transactions on Information and System Security (TISSEC)*, vol. 18, no. 1, pp. 1–32, 2015.

[24] Q. Bai, X. Zhou, X. Wang, Y. Xu, X. Wang, and Q. Kong, "A deep dive into blockchain selfish mining ICC," in *2019-2019 IEEE International Conference on Communications (ICC)*, pp. 1–6, Shanghai, China, 2019.

[25] A. Sapirshtein, Y. Sompolinsky, and A. Zohar, "Optimal selfish mining strategies in bitcoin," in *International Conference on Financial Cryptography and Data Security*, pp. 515–532, Berlin, Heidelberg, 2017.

[26] Z. Wang, Q. Lv, Z. Lu, Y. Wang, and S. Yue, "ForkDec: accurate detection for selfish mining attacks," *Security and Communication Networks*, vol. 2021, 8 pages, 2021.

[27] S. Solat and M. Potop-Butucaru, "Zeroblock: preventing selfish mining in bitcoin," 2016, https://arxiv.org/abs/1605.02435.

[28] M. S. Rahman, M. Y. Uddin, T. Hasan, M. S. Rahman, and M. Kaykobad, "Using adaptive heartbeat rate on long-lived TCP connections," *IEEE/ACM Transactions on Networking*, vol. 26, no. 1, pp. 203–216, 2018.

[29] Y. Xiang and D. Loker, "Trans-causalizing NAT-modeled Bayesian networks," *IEEE Transactions on Cybernetics*, vol. 52, no. 5, 2020.

[30] B. Alangot, D. Reijsbergen, S. Venugopalan, and P. Szalachowski, "Decentralized lightweight detection of eclipse attacks on bitcoin clients," in *2020 IEEE international conference on Blockchain (Blockchain)*, pp. 337–342, Rhodes, Greece, 2020.