2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS) | 978-1-6654-7177-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICDCS54860.2022.00027

The Security Investigation of Ban Score and Misbehavior Tracking in Bitcoin Network

Wenjun Fan*§, Simeon Wuthier†, Hsiang-Jen Hong†, Xiaobo Zhou†, Yan Bai‡ and Sang-Yoon Chang†
*Department of Communications and Networking, School of Advanced Technology
Xi'an Jiaotong-Liverpool University, Suzhou, Jiangsu, P. R. China, 215123
Email: Wenjun.Fan@xjtlu.edu.cn
†Computer Science Department, College of Engineering and Applied Science
University of Colorado Colorado Springs, Colorado Springs, United States, CO 80918
Email: {swuthier, hhong, xzhou, schang2}@uccs.edu
‡School of Engineering and Technology
University of Washington Tacoma, Tacoma, United States, WA 98402
Email: {yanb}@uw.edu

***Abstract*—Bitcoin P2P networking is especially vulnerable to networking threats because it is permissionless and does not have the security protections based on the trust in identities, which enables the attackers to manipulate the identities for Sybil and spoofing attacks. The Bitcoin node keeps track of its peer's networking misbehaviors through ban scores. In this paper, we investigate the security problems of the ban-score mechanism and discover that the ban score is not only ineffective against the Bitcoin Message-based DoS (BM-DoS) attacks but also vulnerable to the Defamation attack as the network adversary can exploit the ban score to defame innocent peers. To defend against these threats, we design an anomaly detection approach that is effective, lightweight, and tailored to the networking threats exploiting Bitcoin's ban-score mechanism. We prototype our threat discoveries against a real-world Bitcoin node connected to the Bitcoin Mainnet and conduct experiments based on the prototype implementation. The experimental results show that the attacks have devastating impacts on the targeted victim while being cost-effective on the attacker side. For example, an attacker can ban a peer in two milliseconds and reduce the victim's mining rate by hundreds of thousands of hash computations per second. Furthermore, to counter the threats, we empirically validate our detection countermeasure's effectiveness and performances against the BM-DoS and Defamation attacks.**

***Index Terms*—Bitcoin, Ban Score, Misbehavior, P2P Networking, Denial of Service, Sybil, Spoofing**

I. Introduction

In the decentralized cryptocurrency economy, since Satoshi Nakamoto published the white paper in 2008 [1], Bitcoin as the most typical cryptocurrency has gained great popularity and has millions of wallet users. Bitcoin and cryptocurrency have also drawn great attention of the security community since it secures integrity and non-repudiation of the transactions while supporting permissionless operations for decentralization and anonymity. Bitcoin's peer-to-peer (P2P) networking does not use identity-/credential-based cryptographic protections, all the Bitcoin messages ride on plain-text TCP connections, and thus security threats including denial of service (DoS) are prevalent [2]–[4]. For instance, the real-world DoS attacks on the Bitcoin exchange platforms were studied in [5], [6], and the DoS threat using the Bitcoin Core's vulnerability (CVE-2018-17144) [7] enables the malicious miner to validate a block that contains a transaction attempting to spend same input twice to crash the Bitcoin infrastructure.

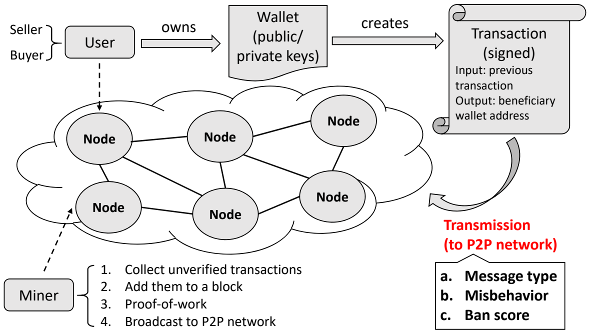
Seller
Buyer
User
owns
Wallet (public/ private keys)
creates
Transaction (signed)
Input: previous transaction
Output: beneficiary wallet address
Node
Node
Node
Node
Node
Node
Node
Transmission (to P2P network)
a. Message type
b. Misbehavior
c. Ban score
Miner
1. Collect unverified transactions
2. Add them to a block
3. Proof-of-work
4. Broadcast to P2P network

Fig. 1. A high-level overview of the Bitcoin P2P network: our work focuses on the transmission to the P2P network

Figure 1 illustrates an overview of the Bitcoin P2P network. In particular, Our work focuses on the ban-score mechanism applicable to any Bitcoin message types, including those for delivering transactions and blocks. The present ban-score mechanism of Bitcoin Core [8] was originally designed for resisting against the DoS attack (though the ban-score mechanism was considered for preventing any other potential network threats). With the ban-score mechanism, a Bitcoin node can use the ban score to keep track of its peer's misbehaviors by increasing the ban score, and once the ban score reaches the threshold (at 100), the peer will get banned for 24 hours. However, we find that the current ban-score mechanism is not only ineffective but also vulnerable. In this paper, therefore, we are motivated to investigate the security threats on Bitcoin networking nodes to discover the attack vectors that can reveal and exploit the ineffectiveness and vulnerability of the ban-score mechanism.

To this end, we investigate the interplay between the threats and defenses in terms of the current Bitcoin practice. First, we sort out that the *Bitcoin-Message-based DoS* (BM-DoS) attack reveals that the ban score is ineffective and deficient. Though the ban score provides some resistance against Sybil, because the attacker requires the bulky handshaking-based reconnection process, which costs the attacker up to hundreds

[§] This work was done while Dr. Wenjun Fan was a Postdoctoral Research Associate at University of Colorado Colorado Springs.

2575-8411/22/$31.00 ©2022 IEEE
DOI 10.1109/ICDCS54860.2022.00027

of packet transmissions, it still suffers multiple DoS attack vectors which are discovered and described in Section III-B. Second, we come up with the *Defamation* attack that aims to exploit the vulnerability of the ban-score mechanism. More specifically, the attacker spoofs the innocent peer to send misbehaving messages to the target node in order to fool the target node to ban the innocent peer. Using this attack, the network adversary can easily disconnect the peer connections with the purpose of decreasing the diversity of the peer connections and disturbing the Bitcoin operations.

Therefore, we analyze the Bitcoin's ban-score mechanism and discover that it is ineffective against the BM-DoS attack and vulnerable against the Defamation attack. To counter such threats exploiting the ban-score mechanism, we present an anomaly detection approach, including the identification of the detection features which are especially effective against such anomalies, and discuss some potential countermeasures which involve modification on the ban-score mechanism. Our paper aims to inform the Bitcoin and blockchain R&D communities of the vectors and the severity of the attacks against the current ban-score mechanism, and demonstrate the potential countermeasure to them for further security research and development to secure the networking.

In this paper, we take an empirical study to achieve an astute observation of the attacks against the misbehavor tracking in Bitcoin network. Hence, we first build an active Bitcoin node connected to the Bitcoin Mainnet. Second, we prototype and measure the attacks against the active Bitcoin node to investigate how the attacks impact the node's networking and operations. Third, we analyze the system cost on both the attacker's side and the victim's side, i.e., use the impact-cost ratio to measure the attack. Fourth, we build our defense measures and consolidate the effectiveness to address the security problem that the ban-score mechanism exacerbates.

Responsible Disclosure We disclosed our research findings, including the vulnerabilities, threats, and countermeasures, to the Bitcoin Core team on March 29th, 2021 and received their feedback on April 1st, 2021. The Bitcoin Core team confirmed that the ban-score mechanism doesn't accomplish much in terms of protection from the attackers considered in this paper. Our discoveries and security concerns about ban score and misbehavior tracking are based on Bitcoin Core 0.20.0. The disclosure to the Bitcoin Core Team affected the following developments as the team changed and improved parts of the ban-score mechanism. These changes include: updating the rules with `VERSION` in the later Bitcoin Core versions since we reported the Defamation attack using the `VERSION` message (see Section VI-D) and an anomaly detection method against the `BLOCK` message traffic in the following Bitcoin Core versions. We are continually working with them to analyze the benefits and costs/vulnerabilities of ban score, to explore the proposed countermeasures and how they fit into the Bitcoin system and other existing mechanisms, and to improve the Bitcoin design and implementation.

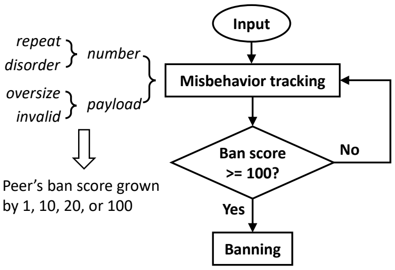

Fig. 2. The Bitcoin's ban-score mechanism logic

II. A Primer of the Bitcoin's Ban-score Mechanism

Bitcoin network is designed as an open P2P network, whereby any node can join the network without permission. This openness, however, makes it possible for malicious nodes to join and attack the P2P network. Thus, on the one hand, the Bitcoin P2P networking requires availability and resistance against denial of service (DoS) threats, otherwise, a node will be disabled from the cryptocurrency operations. On the other hand, a Bitcoin node should resist against the Eclipse attacks [9], [10]. The principle of Eclipse attack is that the attacker would occupy all the peer-connection slots of the victim node and then filter the victim node's view of the blockchain network. With Eclipse, the attacker can take down a victim node with low cost, and it could even control the victim's mining power for its nefarious purposes or cheat the victim to launch a double-spending attack.

Therefore, a DoS prevention framework of Bitcoin (in Pull 517 [8]) was introduced to protect the Bitcoin's P2P networking nodes. Though the framework was originated for defending against DoS attacks, it was informed for responding to other potential attacks, e.g., Eclipse attack. The framework essentially provided a ban-score mechanism. Figure 2 illustrates a brief of the ban-score mechanism of Bitcoin Core. We define that "misbehavior tracking" refers to the score keeping of ban score of node's peers. If a peer is sending wrong information to a node, the node can keep track of the misbehavior in terms of certain ban-score rules and punish the peer (that is a connection identifier denoted by a pair of IP address and Port number, i.e.,`[IP:Port]`) by dropping the peer connection and banning the connection identifier when the ban score reaches the threshold (100 by default) for a banning period (24 hours by default). Thereby, the peer is disconnected and added to the banning filter so that it cannot immediately reconnect using the same connection identifier. The ban-score rules mainly focus on the number (repeat or disorder) and the payload (oversize or invalid) of the message, which will result in an increment (1, 10, 20, or 100) of the ban score to the peer. Further, we found that the misbehavior tracking information is only stored in the node's memory, and the node never broadcasts the misbehaving peer's information. Peers that get banned are just disconnected and can not reconnect back immediately, but there is no warning or reason sent to the peer.

We have noticed and studied the ban-score mechanism since Bitcoin Core 0.20.0 till the current 0.22.0, and we found that

TABLE I
THE BAN-SCORE RULES OF BITCOIN CORE (0.20.0 VS.0.21.0 VS. 0.22.0)

Message Type	Message Misbehavior	Ban Score'20	Ban Score'21	Ban Score'22	Object of Ban	Misbehavior Type
BLOCK	Block data was mutated	100	100	100	Any peer	Invalid
	Block was cached as invalid	100	100	100	Outbound peer	Invalid
	Previous block is invalid	100	100	100	Any peer	Invalid
	Previous block is missing	10	10	10	Any peer	Invalid
TX	Invalid by consensus rules of SegWit	100	100	100	Any peer	Invalid
GETBLOCKTXN	Out-of-bounds transaction indices	100	100	100	Any peer	Oversize
HEADERS	10 non-connecting headers	20	20	20	Any peer	Disorder
	Non-continuous headers sequence	20	20	20	Any peer	Disorder
	More than 2000 headers	20	20	20	Any peer	Oversize
ADDR	More than 1000 addresses	20	20	20	Any peer	Oversize
INV	More than 50000 inventory entries	20	20	20	Any peer	Oversize
GETDATA	More than 50000 inventory entries	20	20	20	Any peer	Oversize
CMPCTBLOCK	Invalid compact block data	100	100	100	Any peer	Invalid
FILTERLOAD	Bloom filter size $>$ 36000 bytes	100	100	100	Any peer	Oversize
	Protocol version number $>=$ 70011	100	-	-	Any peer	Invalid
FILTERADD	Data item $>$ 520 bytes	100	100	100	Any peer	Oversize
	Protocol version number $>=$ 70011	100	-	-	Any peer	Invalid
VERSION	Duplicate VERSION	1	1	-	Inbound peer	Repeat
	Message before VERSION	1	1	-	Inbound peer	Disorder
VERACK	Message (other than VERSION) before VERACK	1	-	-	Inbound peer	Disorder

the mechanism evolves time to time (updated by the Bitcoin development community). However, the ban-score mechanism is not formally studied until now, and there is no well-documented report about it. Though some work [11], [12] mentioned it, neither the Bitcoin's developer guide nor the literature showed the comprehensive ban-score rules. Therefore, we were motivated to look into the Bitcoin Core's source code to reveal the ban-score rules. We summarize the ban-score rules in Table I, which includes and compares the rules amongst Bitcoin Core 0.20.0, 0.21.0, and 0.22.0. With this table, we have got some interesting discoveries:

1) We found that not all of the Bitcoin's P2P networking message types (the complete description of all message types can refer to the Bitcoin's developer-reference [13]) have ban-score rules.
2) Among the ban-score rules, though the punishment is supposed to be directly proportional to the severity of the misbehavior, some individual rules remain to debate. For example, the rule of increasing the ban score of 10 points when a previous block is missing is considered to be too arbitrary.
3) One node's peer can be identified as inbound peer (which initiates the TCP connection to the node in question) or outbound peer (that is requested to be connected by the node). One ban-score rule only affects outbound peer, i.e., block was cached as invalid, while some ban-score rules can only be used to ban inbound peer, such as the rules of VERSION and VERACK.
4) We can see that the Bitcoin Core team is increasingly refraining from applying the ban-score rules to the message misbehaviors, since several rules have been deprecated in the recent Bitcoin Core versions, e.g., the rules with VERSION and VERACK.

III. INEFFECTIVENESS: BAN SCORE FAILS TO DEFEND TRICKY BM-DOS VECTORS

In this section, we study the interplay between the *Bitcoin-Message-based DoS* (BM-DoS) attack and the ban-score mechanism. We reveal multiple attack vectors to exploit the ineffectiveness of the ban score.

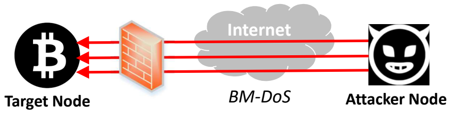

Fig. 3. Threat model of the BM-DoS attack

A. Threat Model against Ineffectiveness

The attacker is capable of injecting networking packets on the victim for DoS and is aware of the Bitcoin protocol. Figure 3 shows the threat model for the BM-DoS attack. First, the network adversary needs to connect to the public internet and knows the target Bitcoin node's IP address, and the target node should be reachable. Second, to launch the application-layer BM-DoS, the attacker node needs to create Bitcoin session to the target node, which means the attacker node needs to establish TCP connection with the target node first. Thus, BM-DoS is a connection-based attack. Third, the attacker has enough computing/networking resources (e.g., botnet) to overwhelm the victim. For instance, every bot builds a connection to the target node (that can maintain up to 117 inbound peer connections out of the overall 128 connections). Fourth, we assume that the real world Bitcoin node could be deployed behind a perimeter firewall. A public node is publicly reachable by definition (if not, it is a private node) and the firewall- or perimeter-based filtering is not applicable in cryptocurrency networking and for our work. A public node can physically be located behind the network perimeter firewall, but the firewall uses an open port 8333. Therefore, the traditional application-layer or circuit-layer firewall mechanisms are not applicable for cryptocurrency contexts.

B. Attack Vectors

We discover that the ban-score mechanism is ineffective and deficient against multiple tricky attack vectors, which are described specifically as follows.

*1) **Nullifying ban score by using messages never getting banned:*** We can see that not all of the Bitcoin message types are equipped with ban score as stated in Section II. According to Table I, only 12 out of 26 message types [13] possess corresponding ban-score rules in Bitcoin Core 0.20.0. Thus,

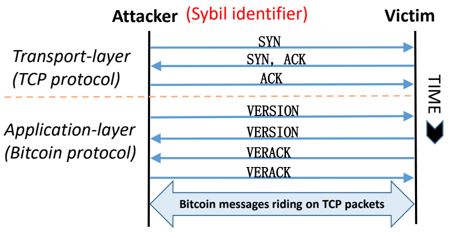

Fig. 4. Sybil identifier to establish Bitcoin sessions (built on TCP connections)

the adversary can still use those message having no ban score (e.g., Bitcoin `PING`) to launch the BM-DoS attack.

*2) **Forgoing ban score by constructing bogus messages**:* We discover that the adversary can still use the messages which are protected by ban score to attack the target node as long as the adversary can construct the bogus message payload to bypass the misbehavior tracking to avoid the ban score increasing. For instance, despite the fact that a Bitcoin `BLOCK` message has ban-score rule protection, an adversary can still construct and send a bogus payload with an invalid Proof-of-Work (PoW) hash value and an incorrect checksum. Usually, the target node will check an arrival packet's header information on the transport layer first, i.e., the TCP `seqnum`, `acknum`, `flags`, and `checksum` field. If the packet passes the check on the transport layer, the message payload will be decapsulated from the segment of the transport layer, passed to the application layer, and processed by the misbehavior tracking. In our attack case, the target node will process and drop the packet when it finds the incorrect checksum on the transport layer, which will occur before the misbehavior checking on the application layer, whereby the adversary's connection will not be banned. This gives an option to the adversary to forgo the ban score.

*3) **Defeating ban-score mechanism by creating serial and multiple Sybil connections**:* Further, even if the adversary is not able to construct the message payload to trick the ban-score-based protection, the adversary can still generate serial and multiple Sybil identifiers to connect to the target node and transmit misbehaving messages to it. Actually, in the context of permissionless Bitcoin P2P networking, one entity/node could have multiple identifiers. Hence, an attacker node can use multiple identifiers to establish Bitcoin sessions to the target node. Figure 4 illustrates how Sybil identifier works for this attack vector. The attacker node (A) using Sybil socket pairs (picking up a free `[IP:Port]`) initiates the TCP three-way handshake to the victim node (V), which is the target node listening on port 8333. Once the TCP three-way handshake is done, they have connected on the transport layer, and then A needs to build the Bitcoin application-layer session with V by exchanging `VERSION` and `VERACK` (which ride on TCP packets) named Version Handshake. If A succeeds, it can transmit Bitcoin messages with V directly. Thus, A can send misbehaving messages to V. When a prior identifier gets banned, A can use another un-banned identifier (e.g., another port with the same IP address) to create the next connection in serial to keep sending misbehaving messages to V.

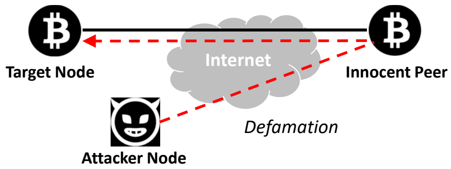

Fig. 5. Threat model of the Defamation attack

IV. Vulnerability: Ban Score Enables Unexpected Defamation Attack Vectors on Innocent Peer

In this section, we expose the vulnerability and a major threat of the Bitcoin's ban-score mechanism. We propose the *Defamation* attack to exploit and leverage the ban score to make the innocent peer get banned by the target node.

A. Threat Model against Vulnerability

We build on the threat model in Section III-A. However, the threat model for launching the ban-score-exploiting threats requires greater capabilities than the one against ineffectiveness. Figure 5 presents the threat model of Defamation. There are two victims involved in this attack, i.e., the target node and the innocent peer. Assuming that there is a connection between the target node and the innocent peer, the attacker node sends the misbehaving Bitcoin messages to the target node pretending to be the innocent peer in order to make target node ban the innocent peer. To inject such messages into the TCP connection, the attacker should know the 4-tuple `<source IP, source port, destination IP, destination port>` and the real-time TCP state of the connection, i.e., `seqnum` and `acknum`. Although, some network switches/routers may prevent sniffing, and Internet service providers (ISPs) and autonomous systems (ASes) could use network access control to block IP spoofing, there are a number of cases that can satisfy the requirement in practice. For example, we can sniff and spoof when the attacker node and the target node are in the same network using the promiscuous mode (as is the case for Bitcoin public nodes) (e.g., [14]), or when they are in the same 802.11 wireless network, or with the help of a compromised ISP/AS (e.g., [10]).

Our threat model is different from the network connection based man-in-the-middle (MitM) threat model (where the attacker covertly relays and possibly alters the communications between two parties who believe that they are directly communicating with each other). We only assume that the adversary can eavesdrop on the connection and inject data into the TCP connection, but we do not assume that the adversary must have the capability to perform route manipulation to make itself have a privileged position in the middle of the target node and the Bitcoin Mainnet (as done in previous cryptocurrency research, e.g., [10], [15]).

To compare the TCP reset attack [14] from the Defamation threat model. Although a TCP reset attack also needs the real-time TCP connection's state, such an attack is feasible regardless of the existence of the ban-score mechanism. There is not much can be done for defending against the TCP reset attack by the public P2P network. Nevertheless, using TCP

Algorithm 1 Post-connection Defamation

```
Require: $A$ can sniff $i$'s inbound peer connection from $j$
Ensure: $j$ gets banned by $i$
$A$ gets 4-tuple [$i$'s IP, $i$'s port, $j$'s IP, $j$'s port]
while $A$ performs real-time eavesdropping do
    $A$ learns the current TCP seqnum and acknum
    $A$ crafts misbehaving message with the packet header using the 4-tuple and the expected seqnum and acknum
    $A$ injects the misbehaving message to $i$
    $i$ increases ban score against $j$
end while
```

reset attack can only terminate a connection but can not ban a peer identifier for 24 hours.

B. Attack Vectors

We unveil the Defamation attack vectors to exploit and show the vulnerability of the ban-score mechanism as follows.

*1) **Pre-connection Defamation**:* In this case, the attacker only needs to know the existence of the innocent peer identifier j and the target node identifier i, and only to preemptively make j get banned by i before j attempts to connect to i. In other words, the attacker node performs IP spoofing by using j to connect and transmit misbehaving messages to i. This attack will work if it occurs before the innocent peer uses j to connect to i, i.e., there is no TCP connection between the original innocent peer and the target node, and so the attacker node only needs to perform IP spoofing rather than real-time eavesdropping and TCP data injection.

*2) **Post-connection Defamation**:* In this case, the target node identifier i and the innocent peer identifier j have already established a TCP connection, where assuming j is an inbound peer identifier towards i. To defame j, the attacker A needs to perform not only spoofing but also sniffing and learning the real-time TCP connection state in order to inject data into the connection. In brief, A should be capable of spoofing j and inject misbehaving messages to i to make i ban j. The attack procedures can be described by Algorithm 1.

V. Implementation

This section presents the attack prototyping and the testbed setup for conducting experiments.

A. Attack Prototyping

Regarding the attack prototyping, the attacker is required to be able to establish a Bitcoin session with the target node and construct certain Bitcoin messages for data transmission/injection. The attacker is not necessary to be a full Bitcoin node, but it should use the Bitcoin library like python-bitcoinlib [16] to satisfy the requirements. Moreover, to facilitate the post-connection Defamation attack, we use Scapy to sniff the target connection, craft misbehaving message packets, and inject the spoofed TCP packet into the connection heading to the target node.

B. Testbed Setup

We implement the real-world Bitcoin nodes on machines (using Ubuntu 18.10 64-bit operation system, Intel Core i7 4GHz CPU, 4GB memory, and Intel PRO/1000 MT Desktop network adapter). In addition to the node labeled "target node", we implement another two nodes: "attacker node" and "innocent peer", but all the three nodes have equal machine specifications. Both target node and innocent peer install and run Bitcoin Core (software version Satoshi 0.20.0 and protocol version 70015) with the default configuration.

TABLE II
MEASUREMENT OF BITCOIN MESSAGE TYPES PER QUERY

Bitcoin Message	Attacker's cost (clocks)	Victim's impact (clocks)	Impact-Cost ratio
VERSION	60.71	129.5	2.13
VERACK	48.57	241.375	4.97
ADDR	5743.68	42.981	0.0075
INV	47112.62	77.83	0.0017
GETDATA	41270.62	238.905	0.0058
GETHEADERS	50.8	38.875	0.77
TX	54.55	609.016	11.16
HEADERS	7220.95	16.394	0.0023
BLOCK	23.45	617282.101	26323.33
PING	21.33	95.582	4.48
PONG	20.68	9.797	0.47
NOTFOUND	16.75	10.232	0.61
SENDHEADERS	12.89	7.125	0.55
FEEFILTER	15.37	8.714	0.57
SENDCMPCT	15.85	4.889	0.31
CMPCTBLOCK	14.48	46225.182	3192.35
GETBLOCKTXN	422.32	874	2.07
BLOCKTXN	16.66	97445.452	5849.07

The Bitcoin configuration file for the target node and innocent peer includes the adapter attribute so that the innocent peer can automatically connect to the target node as an inbound peer when they start up. Other than the innocent peer, the target node also connects to the other peers from Internet. Our attacker node only connects to the target node and never connects to the Bitcoin Mainnet due to ethical concerns.

We note that i) for the BM-DoS vectors and the preemptive/pre-connection Defamation attack, we only need the attacker node to directly connect to the target node to carry out desired attacks; ii) for the post-connection Defamation attack, all the three nodes are involved, where the attacker node needs to sniff the connection of the target node and the innocent peer node, and spoofs the innocent peer and transmits misbehaving messages to the target node.

VI. Attack Measurements and Analyses

In this section, we measure and analyze the impact as well as the cost of using different attacks.

A. Measurement of Message Types: Impact-Cost Ratio

The *impact-cost ratio* of DoS indicates the ratio of the impact for processing the received message on the target node, over the cost for sending the message on the attacker node. To this end, we measure the clock cycles (i.e., CPU time) for processing different Bitcoin message types (used by default) on both the attacker and the target nodes respectively. Thus, from the attacker node's perspective, the mean clock cycles indicate how much CPU time is needed to generate a certain message, while from the target node's perspective, the average clock cycles denote how long it has to take to process a certain arrival message.

Table II presents a measurement of processing every message per query from both sides. We calculate the impact-cost ratio which is shown in the fourth column of the table to find out which message type can make the adversary have

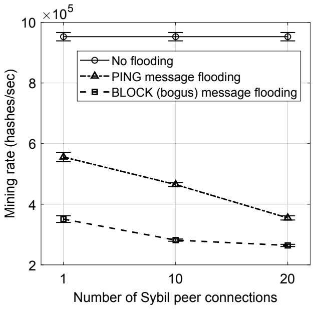

Fig. 6. BM-DoS impacts mining rate

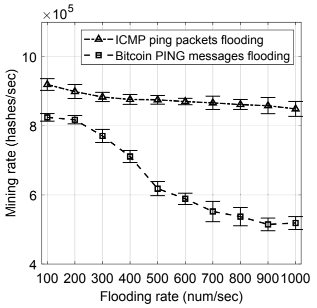

Fig. 7. Mining rate impact

the greatest gain - the highest value of impact-cost ratio. We found that using BLOCK would gain the highest impact-cost ratio, which is 26323.33 on average and is more than 4 times as much as the second-highest BLOCKTXN that spends 5849.07 in mean. Also, it is countable that one BLOCK message has the same impact as 6 BLOCKTXN messages in the victim's computational resource cost. Thus, sending BLOCK messages is the best option for flooding, which forces the victim to execute resource-consuming operations that are disproportionate to the attack effort.

B. BM-DoS Impacting Mining Rate

As stated in Section III, the bogus BLOCK message can bypass the misbehavior checking, and the previous subsection also shows that it leads the target node to cost the greatest number of clock cycles[1]. Hence, we use bogus BLOCK BM-DoS to flood the target node to impact the mining rate. Figure 6 shows the results which also include the result under PING messages flooding for comparison. In this case, the attacker performs flooding as fast as possible which means the attacker setting no interval/delay between two consecutive messages. The figure shows average values (with 95% confidence level) given by using the Bitcoin node to do 100 mining samples, and each sample performs 10^7 hashes. We find that the mining rate under no flooding is as high as $9.5 \cdot 10^5$ h/s, which is much higher than the one of $3.5 \cdot 10^5$ h/s under invalid BLOCK BM-DoS with a single connection. Further, we enable Sybil identifiers to handle 10 sockets (with 10 threads running in parallel, whereas running them across different bots will yield even more impact) and 20 sockets respectively to contact the target node and keep flooding the target node using bogus BLOCK messages concurrently. With this, an increasing number of Sybil connections is applied to DoS the victim and measure the change of its mining rate. That does impact the mining rate, which results in $2.8 \cdot 10^5$ h/s and $2.6 \cdot 10^5$ h/s on average under 10 and 20 Sybil connections separately. By contrast, using PING BM-DoS leads to the mining rates of $5.5 \cdot 10^5$ h/s, $4.6 \cdot 10^5$ h/s and $3.5 \cdot 10^5$ h/s under 1, 10 and 20 Sybil peer connections respectively. It indicates that the BLOCK BM-DoS has a higher impact than the PING BM-DoS, and the attacker's cost is directly proportional to the impact on the victim.

TABLE III
DoS attack Impact-to-Cost comparison

		Cost (Attacker)		**Impact (Victim)**	
Layer	**Rate (num/sec)**	**CPU (%)**	**MEM (MB)**	**Bandwidth DoSed (kBits/s)**	**Mining Rate (times/sec)**
Bitcoin PING	10^2	1.3	14.34	96.48	824564.81
	10^3	4.7	14.34	482.31	518954.34
ICMP ping	10^2	2.7	2.048	107.72	919619.71
	10^3	14	2.048	383.14	841188.46
	10^4	62.9	2.048	1853.44	639356.67
	10^5	88.7	2.048	3491.84	505638.85
	10^6	98.3	2.048	5990.40	359115.99

C. BM-DoS vs. Network-layer Traffic Flooding

We also measure and compare the attack efficiency between the network-layer traffic flooding and the application-layer BM-DoS. To facilitate a fair comparison, we use Bitcoin's PING message for the BM-DoS attack and ICMP protocol ping packet for the network-layer traffic flooding. We measure the attacker‘s cost including CPU and memory usage, and the target node's impact including the bandwidth and the mining rate in terms of applying different flooding rates. Table III presents the measurement results.

Our attack implementation in python reveals that the BM-DoS has the flooding rate limitation to 10^3 messages per second, which denotes that if the attacker node increases the rate beyond that value, the flooding only lasts several seconds then the network socket becomes inundated and the pipeline breaks. By contrast, the network-layer using flooding tools like *hping* can grow the flooding rate up to 10^6 packets per second. However, comparing the same flooding rate, we observe that the Bitcoin PING BM-DoS costs less attacker node's CPU (and more memory) resource than the ICMP ping flooding.

Further, we reveal that BM-DoS can impact the mining rate more than the network-layer traffic flooding (see Figure 7), although the network-layer traffic flooding can eat more bandwidth than BM-DoS. That is because the arrival Bitcoin PING message will cause the Bitcoin node to process it in the application-layer, which consumes CPU resource, while the arrival ICMP ping packet will only be processed by the network-layer that is conducted by the operating system.

D. Innocent Peer Ban Rate by Defamation

As stated, the adversary can even leverage the ban score to attack the target node by recreating multiple socket connections in serial if the prior connection gets banned. Figure 8 presents the target node's ban score tracking results when it receives misbehaving VERSION messages. We know that each VERSION message (except for the first one) arriving to the target node will result in an increment of the ban score. Thus, the attacker node can carry out a loop-attacking by continuously establishing new connection after the prior connection getting banned and use such serial Sybil connections to keep sending VERSION messages to the target node.

We see that if there is no delay (sending the message as fast as possible), one peer identifier will get banned in 0.1 seconds

[1] The bogus BLOCK message (without application-layer process at the victim's side) still has the highest impact-cost ratio which is 2132.79 amongst all the message types using bogus payloads.

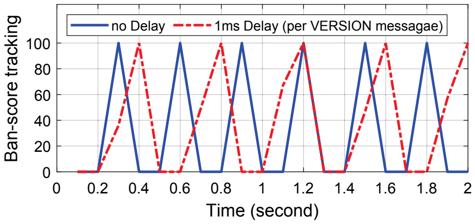

Fig. 8. Defamation attack using `Version` message, where the attacker bans another peer node if the ban score reaches 100

in mean, while if we set 1 millisecond delay between every two consecutive messages for comparison, the Sybil identifier will be banned slower, in 0.2 seconds on average. The quicker the attacker gets banned, the more continuous Sybil connections it can launch within a certain time interval. However, the lower flooding rate saves the networking/transmission resource and is generally more efficient. Also, we find that the Sybil attack program used by the attacker node has latency to establish every new socket pair connecting to the target node, which is approximately 0.2 seconds. That is bulky since it costs the attacker up to hundreds of packet injections.

Moreover, if one attacker would like to perform the preemptive Defamation attack to fully defame an IP address, which means that it needs to defame $65536 - 49152 = 16384$ ports in theory (TCP ports with numbers from 49152 to 65535 are used as dynamic, private or ephemeral ports), which will take approximately $16384 \cdot (0.1 + 0.2)/60 = 81.92\ mins$. With that attack, this IP address will not be able to create any connection to the target node for 24 hours, because all the potentially available identifiers had been banned. Therefore, the peer-table's diversity of the target node is decreased, and if the attacker is powerful enough, the greater number of peers can be defamed within an even shorter time.

VII. Our Detection Countermeasure

In this section, we present the anomaly detection to build the intelligence to defend against the BM-DoS attack and the Defamation attack. The proposed detection approach does not need any changes to the current Bitcoin Core, thus, the anomaly detection still makes use of the existing ban-score mechanism. In the next section, we discuss other potential countermeasures which include changes to the Bitcoin Core.

A. Anomaly Detection for Bitcoin

In general, anomaly detection is helpful for monitoring and detecting anomalous behavior, in particular, for the networking traffic where DoS attack's impact can be exposed. As opposed to using identifier-based detection that is ineffective in the permissionless environment as the spoofing and Sybil attacks can often violate peer identifiers in such network environment, we introduce an identifier-oblivious detection approach to detect anomalies like DoS attacks. That is typically effective against our attacks for the permissionless Bitcoin P2P network, because the presented BM-DoS attack and Defamation attack involving Sybil and spoofing which make the identifier-based detection approach ineffective. In this regard, this proposed detection method mainly uses the analysis of the message traffic information rather than the peer connection identifiers to detect the anomalies. The novelty of this approach is not because of the traffic information analysis but the specific features of the Bitcoin messages used to detect the anomalies. In addition, this approach uses statistical analysis rather than machine learning, and so there is no computational resource requirements for implementing the detection engine.

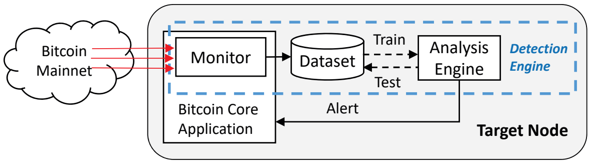

Fig. 9. Anomaly detection engine using statistical analysis and involving no Bitcoin Core change

The detection engine builds on the real-world Bitcoin node implementation connecting to the Bitcoin Mainnet. Figure 9 presents the target node prototype where the anomaly detection engine builds. The detection engine consists of three main components: Monitor, Dataset and Analysis Engine. The Monitor component (which actually is a functional module of the Bitcoin Core application) is used to collect the arrival Bitcoin messages transmitted from the Bitcoin Mainnet. The Dataset component is used to store the collected data (messages) in a certain format. The Analysis Engine component uses the data to train its reference profile and applies the built model to detect the testing data. When it detects anomaly, it can send alert to the Bitcoin Core application to inform it to take reaction, e.g., to drop and rebuild peer connections. The machine connects to the Bitcoin Mainnet so that it can collect the real-world data for training and testing. The outlier data is generated by an attacker node that only connects to the target node so that the misbehaving message traffic will not spill out to the Bitcoin Mainnet. The generated anomaly traffic is mixed with the normal real-world data when we generate the abnormal dataset.

1) Key Features for Detection: This subsection presents the features of the Bitcoin messages used to perform anomaly detection. The following feature is novel and specific to the Defamation attack:

Outbound Peer Reconnection Rate (c) This feature denotes the peer reconnection count per minute (rate). It is specifically selected to detect the Defamation attack, because we know that once an innocent outbound peer gets banned, the target node will rebuild a new outbound peer connection, and this operation will lead the target node to have an abnormal peer reconnection rate during a certain time window (e.g., 10 minutes), which can be used to detect the anomaly.

There are also other features that are generally adopted for anomaly detection although they are Bitcoin specific:

Overall Message Rate (n) This feature indicates the count rate over all the arrival Bitcoin messages (num/min). It is typically selected to detect the BM-DoS attack, because if there is a message flooding traffic, the overall message rate will deviate from the normal message rate. That can be used to detect the anomaly.

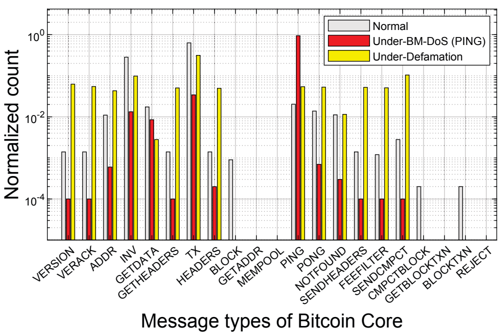

Fig. 10. Anomaly detection by comparing the count of messages (Λ): the normal case and the under-BM-DoS case have the correlation $\rho = 0.05$; the normal case and the under-Defamation case have the correlation $\rho = 0.88$

Message Count Distribution (Λ) This feature represents the relative count distribution among all messages. It is used for detecting both the BM-DoS and the Defamation attack, since both attacks using certain Bitcoin messages for carrying out the attack will change the relative message count distribution from the normal distribution.

2) Anomaly Detection Performance: After training the model using the normally collected data for approximate 35 hours, we figure out that the threshold of c is $\tau_c = [0, 2.1]$ reconnections per minute during a time window of 10 minutes, the threshold of n (i.e., the range of the message arrival rate) is $\tau_n = [252, 390]$ messages per minute, and the threshold of Λ (i.e., the similarity using correlation coefficient) is $\tau_\Lambda = 0.993$.

Detection Accuracy With the reference profile and the fixed thresholds, we can further use the model to perform anomaly detection. Figure 10 shows the comparison of the normalized count of messages (in the vertical axis in the logarithmic scale) between the normal case, the under-BM-DoS case and the under-Defamation case.

On the one hand, the under-BM-DoS case represents the one when the attacker takes the BM-DoS attack through sending numerous `PING` messages to the target node. We can find that the under-BM-DoS case has the `PING` message dominating the message count distribution, whereby the `PING` message takes 94.16% of the normalized count of the overall messages, it is 45.38 times greater than the `PING`'s normalized count in the normal case, and it is also 26.45 times greater than the `TX`'s normalized count in the under-BM-DoS case. Thus, the similarity of the normal case and the under-BM-DoS case becomes very low, i.e., the correlation $\rho = 0.05$ which is largely lower than $\tau_\Lambda = 0.993$. Also, the flooding `PING` messages increase the n, resulting in approx. 15,000 messages per minute, which is much greater than the upper bound of τ_n that is 390 messages per minute.

On the other hand, the under-Defamation case denotes the one when the attacker keeps defaming the innocent outbound peers of the target node so that the target node has to reconnect to new outbound peers by `VERSION/VERACK` exchange. We can see that the `VERSION`'s normalized count is 43.57 times greater than the one in the normal case, and the `VERACK`'s normalized count is 30 times greater than the one in the normal case. Thus, the correlation of the normal case and the under-Defamation case is $\rho = 0.88$ that is lower than the threshold, and also, we get that the outbound peer reconnection rate is $c = 5.3$, which is greater than the upper bound of τ_c that is 2.1 reconnections per minute.

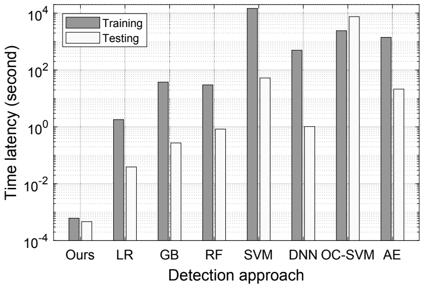

Fig. 11. Comparison of the training and testing latencies of the detection approaches, and our approach is on the far left in "Ours"

Therefore, the proposed message traffic features can distinguish the under-BM-DoS and under-Defamation cases from the normal case, and the anomaly-based detection can leverage the Bitcoin message traffic information to detect if the target node is under the presented attacks. After the detection, the Bitcoin node can perform detection response to resist against the attack, e.g., to disconnect the current anomaly connections and rebuild all the peer connections.

Our detection accuracy performance is 100% because the attacker tested against our scheme does not make the effort to avoid the detection. However, attacker which controls its traffic and reduces the traffic amount for the attack would have a smaller impact on the victim, and our detection scheme has the security effect of mitigating the attack. More specifically, against BM-DoS, the attacker consumes less bandwidth/resource of the victim; against Defamation, the attacker takes longer time to succeed in its blacklisting goal. We leave a more intelligent attacker for future work.

Detection Cost Overhead Further, we compare the time latencies (as they show how timely the defense would be) of both training and testing between our approach and the machine learning (ML)-based approaches described in the literature [17]–[22], including Logistic Regression (LR), Gradient Boosting (GB), Random Forest (RF), Support Vector Machine (SVM), Deep Neural Network (DNN), One-Class SVM (OC-SVM) and AutoEncoder (AE). Figure 11 shows that our approach using statistical analysis is at least four orders of magnitudes efficient than the ML-based approaches depending on the certain ML algorithm which is adopted.

VIII. Discussions of Other Potential Countermeasures

Although we should err on the side of caution when making the decision of changing the Bitcoin Core (otherwise it will backfire), this section discusses several potential countermeasures that involve Bitcoin Core changes for further investigation in the future.

Forgoing Ban Score Our research discovers that the ban score is both ineffective against BM-DoS (that can bypass the ban score) and vulnerable against Defamation (that can use the ban score to attack other nodes). Therefore, we recommend forgoing the ban-score mechanism as a preventive approach to eliminate the opportunities for such networking threats exploiting the ban score. Disabling the ban score does not affect any of the other Bitcoin operations. The node disabling the ban score can still communicate with other nodes and participate in the Bitcoin protocols as usual, since ban score is implemented locally and in a distributed manner, i.e., all nodes keep track of their peers' ban scores locally and use them as they see fit, including not taking any control action.

While there are numerous methods to disable or forgo the ban-score mechanism of Bitcoin Core, we test and validate the following methods which has negligible overheads (we do not observe the performance changes):

- **Ban score threshold to ∞.** This modification method is just to omit the checking when the ban score reaches the threshold at 100. With that, the ban score can increase infinitely and the peer identifier will never get banned. One benefit of this method is that the misbehavior tracking is still working and the score is kept, which would have some other use, e.g., peer-health ranking. This method just needs to comment out lines 1059 to 1062 of the code [23] in Bitcoin Core 0.21.0.
- **Disabling the checking.** This modification method is to fully disable the ban-score mechanism, which omits misbehavior checking and tracking. This method needs to comment out the entire *PeerManager::Misbehaving* function, from lines 1051 to 1064 of the code [23] in Bitcoin Core 0.21.0.

In our disclosure to Bitcoin described in Section I, the Bitcoin Core Team acknowledged our work and validations. However, Bitcoin Core decided to keep the ban-score mechanism because of its benefit to deter some networking misbehaviors. We acknowledge that the ban-score mechanism can defend against unintentional and non-malicious misbehaviors or some security threats unaware of our research, as the classical threats which are older and lacking in sophistication generally exist in the real-world networking. However, we expect the risks of our discovered threats to increase in the future especially after the publication of our research and given the high feasibilities of the threats. We plan to continue to communicate with Bitcoin Core to secure Bitcoin and other cyptocurrencies in the future, including investigating the benefit vs. security risk tradeoff and exploring non-binary mechanism solutions for ban score.

Good-score Mechanism Because the current ban-score mechanism has such critical deficiencies, we consider introducing a good-score mechanism to replace it. The good score is used to increase the credit/reputation to a peer from the target node's perspective, and only the target node keeps track of its every peer's good score through increasing the peer's good score by 1 when the peer transmits a valid `BLOCK` message to the target node. With this, an innocent peer can not be banned by the Defamation attack. Despite an attacker may want to masquerade the high-reputation peer to continuously transmit misbehaving messages to the target node, the attacker is hard to find out the good scores associating to different peers which are only recorded by the target node locally.

Authentication To defend against the Defamation attack the typical approach is to provide cryptographic encryption and authentication to every connection. However, that challenges the design principle of the permissionless cryptocurrency P2P network, since that will bring immense networking overhead. Although P2P communication encryption for Bitcoin has been discussed by BIP151 [24] and BIP324 [25], BIP151 has been withdrawn and BIP324 is still under work. It is estimated that there are over 60,000 nodes in the Bitcoin P2P network [26], [27]. Assuming each node maintains 34 connections according to the result presented by a related work [28], the whole Bitcoin P2P network then will have 1,020,000 connections need to be encrypted. This is a significant overhead considering that Bitcoin relies on broadcasting where all packets get relayed numerous times. However, some cryptocurrency like Ethereum does use specific TCP-based transport protocol to provide the encryption and authentication to the communication among Ethereum nodes. For example, Ethereum adopts the RLPx Transport Protocol using asymmetric authenticated encryption function [29] which is different from the application-layer digital signature function for signing the message payload by using the sender's private key. It is worth mentioning that this countermeasure can resist against the Defamation attack, while it is not able to resist against the TCP reset attack which is out of our threat model.

IX. Literature Review

This section performs a literature review in terms of the following aspects.

A. Ban Score in Other Cryptocurrencies

We find that several other cryptocurrencies also use score mechanism to protect the permissionless P2P networking. The Nervos CKB Blockchain network maintains a P2P scoring system to achieve network security [30]. CKB's scoring system is quite different from the Bitcoin's ban-score mechanism, because it does not only track bad behaviors (via subtracting points) but also counts good behaviors (via adding points), so the nodes need to score peers' good and bad behaviors continuously and can retain good (high-score) peers and evict bad (low-score) peers out. Another case in point is the Dash Core [31], which has a similar ban-score mechanism with Bitcoin, i.e., the mechanisms are in place to punish misbehaving peers who take up bandwidth and computing resources by sending false information. If a peer gets a ban score above the threshold (100 by default), they will be banned for 86,400 seconds by default (24 hours). However, Dash Core includes more ban-score rules for tracking misbehavior. In summary, we can see that ban score is not only used by Bitcoin, but Bitcoin Core team even did not document it, and that is why our research is worth perform.

B. DDoS Threats on Bitcoin Ecosystem

It is prevalent that DDoS threats impact operators and financial services of the Bitcoin ecosystem. In [32], the authors analyzed the economic impact of DDoS attacks on a cryptocurrency exchange using event analysis. Vasek et al. [5] performed an empirical analysis of DoS attacks that impact the Bitcoin currency exchange, mining pools, gambling operators, eWallets, etc. Feder et al. [6] investigated the impact of shocks on trading activity at the leading Mt. Gox exchange using Bitcoin between April 2011 and November 2013. It was revealed that the number of large trades on the exchange fell sharply, particularly, the distribution of the daily trading volume becomes less skewed (fewer big trades). Indeed, some protection means such as Cloudflare has been created to protect the online platform from DDoS attacks.

C. DoS Threats on Bitcoin Blockchain

1) Bitcoin Transaction-based DoS Attacks: A typical case in point is the Bitcoin DoS vulnerability (CVE-2018-17144) [7], which was exploited in Bitcoin Core versions 0.14.0 up to 0.16.2, whereby malicious miners try to validate a block containing a transaction that attempts to spend the same input twice, causing the whole Bitcoin infrastructure to crash. Developers thereafter yielded a patch for anyone running nodes up to version 0.16.3, along with an appeal to update the software. In addition, a number of DoS attacks have been studied, such as crafting bogus transactions to impact miner's hash power [33], transmitting out-of-order transactions to eat up victim's RAM resource [26], sending spam transactions to delay non-spam transactions [34], and increasing the transaction transmission rate to flood the mempool of the full node [35]. All the above-related work focused on violating Bitcoin transaction's order, payload, or rate. Differently, our work aims at crafting misbehaving messages to attack the built-in ban-score mechanism.

2) Bitcoin Protocol-based DoS Attacks: State-based (or protocol-based) DoS attacks include multiple state causality based interactions. A case in point is the messages exchange following the `INV`, `GETDATA` and then `BLOCK`/`TX` sequence. This sort of attack is more sophisticated than flooding based DoS attacks. Built on protocol exploiting, an attacker can launch potential reflection and amplification DoS attacks to Bitcoin node [36]. Gervais et al. [37] presented the 20 minutes time-out (for `BLOCK` message) and 2 minutes time-out (for `TX` message) attacks which enable the adversary to launch propagation delay to the connected victim peer. The related attacks rely on the Bitcoin protocol as well as multiple interactions between the attacker and the victim, which is feasible to be applied, while our work is simple and focuses on one interaction.

D. DoS Threats on Bitcoin P2P Networking

1) Attacks against the Bitcoin Node: Yves-Christian et al. [12] studied several ban-score rules and found a way to reset the ban score to 0 through changing a different message type to send, but those ban-score rules are out of date. In [38], the authors described that an attack can create a large number of bogus addresses using `ADDR` messages to exhaust the victim node's memory. Also, Eclipse attacks [9], [10] use crafted `ADDR` messages to poison the target node's peer-table to further control the target node's peer connections, which is inevitably effective to implicate DoS attack [4]. However, the Eclipse attacks in the context of Bitcoin network become increasingly infeasible with the up-to-date Bitcoin Core (version 0.20.0) since the related bugs were fixed [10].

2) Attacks against the networking infrastructure: To this end, the most well-known and effective attack is called partitioning [35]. In contrast to Eclipse attack that is node-based and focusing on poisoning the node's peer-table, the partitioning attack aims to use the network-layer attacking techniques, like route manipulation by BGP hijacking [15], to separate the target nodes from the rest of the Bitcoin Mainnet. However, the partitioning attack itself is hard to form in practice, since it is difficult to compromise a well configured and protected routing system (like ISP/AS).

E. DoS Countermeasures for Bitcoin

The related work about anomaly detection in the context of Bitcoin often used machine learning (ML)-based approaches, including supervised algorithms [17]–[19] and semi-supervised/unsupervised algorithms [20], [21]. Built on the state-of-the-art, [39] analyzes the feasibility and cost in greater details with a focus on those ML-based detection approaches' training and testing latencies and system impact on mining operation. Most recently, [22] presented an anomaly detection engine on top of AutoEncoder (AE) algorithm, which is able to effectively detect the message-based DoS attack.

In contrast to the previous research based on ML, our detection uses statistical analysis. Our previous work in Lightweight and Identifier-Oblivious eNgine (LION) [40] shows that a well-designed statistical analysis-based detection can be more appropriate and practical for the computing-invaluable miner nodes (e.g., how it impacts the application-layer mining operations) and that such anomaly detection is effective against the previously known Bitcoin threats. This paper however focuses on detecting the novel threat discoveries exploiting the misbehavior tracking and identifies and tests the data/information and parameters for the detection of such threats.

X. Conclusion

Cryptocurrency has the beauty of decentralization, immutability and consensus protocol, however, it lacks authenticity and trust which expose a critical vulnerability to the adversary. Though the Bitcoin's ban-score-based misbehavior tracking framework was proposed to prevent nodes from network threats, we demonstrate that it is not only ineffective but also vulnerable. This ban-score rules can be nullified or bypassed, which means an adversary can still use the ban-score protected messages to attack the victim, by crafting bogus messages. Even worse, the ban-score mechanism has a severe side-effect, which enables an attacker to defame the innocent Bitcoin peers due to the ease of launching connection identifier spoofing in the permissionless P2P network. We reveal and prototype a variety of attack vectors against the current ban-score mechanism, and evaluate their impacts on the target node, which result in various implications, such as the mining rate heavily decreasing. The anomaly detection approach without including Bitcoin Core change is proposed, and also, several other potential countermeasures involving Bitcoin Core changes are discussed. We call for more research in securing the networking for cryptocurrency and permissionless blockchain. A formal analysis of the ban-score mechanism could be as part of a future research.

Acknowledgment

This research work was supported by Colorado State Bill 18-086 and by the National Science Foundation under Grants 1922410 and 1921576. The preliminary version of this work appeared as a poster paper [41]; while this work significantly advances that poster by including more in-depth analyses of the ban-score deficiencies and by adding countermeasures against the BM-DoS attack and the Defamation attack.

References

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[2] J. Bonneau, A. Miller, J. Clark, A. Narayanan, J. A. Kroll, and E. W. Felten, "Sok: Research perspectives and challenges for bitcoin and cryptocurrencies," in *2015 IEEE Symposium on Security and Privacy*, 2015, pp. 104–121.

[3] A. Biryukov and I. Pustogarov, "Bitcoin over tor isn't a good idea," in *2015 IEEE Symposium on Security and Privacy*, 2015, pp. 122–134.

[4] M. Conti, E. Sandeep Kumar, C. Lal, and S. Ruj, "A survey on security and privacy issues of bitcoin," *IEEE Communications Surveys Tutorials*, vol. 20, no. 4, pp. 3416–3452, 2018.

[5] M. Vasek, M. Thornton, and T. Moore, "Empirical analysis of denial-of-service attacks in the bitcoin ecosystem," in *Financial Cryptography and Data Security*, 2014, pp. 57–71.

[6] A. Feder, N. Gandal, J. T. Hamrick, and T. Moore, "The impact of DDoS and other security shocks on Bitcoin currency exchanges: evidence from Mt. Gox," *Journal of Cybersecurity*, vol. 3, no. 2, pp. 137–144, 01 2018.

[7] BitcoinCore, "Cve-2018-17144 full disclosure," 2018. [Online]. Available: https://bitcoincore.org/en/2018/09/20/notice/

[8] G. Anderson, "Denial-of-service prevention," 2011. [Online]. Available: https://github.com/bitcoin/bitcoin/pull/517

[9] 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)*, Washington, D.C., Aug. 2015, pp. 129–144.

[10] 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 *IEEE Symposium on Security and Privacy (S&P)*, 2020.

[11] G. O. Karame, E. Androulaki, M. Roeschlin, A. Gervais, and S. Capkun, "Misbehavior in bitcoin: A study of double-spending and accountability," *ACM Transactions on Information and System Security (TISSEC)*, vol. 18, no. 1, 2015.

[12] 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)*, Shanghai, China, Oct 2018.

[13] BitcoinProject, "The set of 26 bitcoin message types," 2020. [Online]. Available: https://developer.bitcoin.org/reference/p2p_networking.html

[14] W. Fan, S. Chang, X. Zhou, and S. Xu, "Conman: A connection manipulation-based attack against bitcoin networking," in *Proceedings of IEEE Conference on Communications and Network Security (CNS)*, Oct. 2021.

[15] M. Apostolaki, A. Zohar, and L. Vanbever, "Hijacking bitcoin: Routing attacks on cryptocurrencies," in *2017 IEEE Symposium on Security and Privacy (S&P)*, 2017, pp. 375–392.

[16] L. Jongeneel, "Python bitcoin library," September 8, 2020. [Online]. Available: https://pypi.org/project/bitcoinlib/

[17] H. Sun Yin and R. Vatrapu, "A first estimation of the proportion of cybercriminal entities in the bitcoin ecosystem using supervised machine learning," in *2017 IEEE International Conference on Big Data (Big Data)*, Dec 2017, pp. 3690–3699.

[18] M. Harlev, H. Sun Yin, K. Langenheldt, R. Mukkamala, and R. Vatrapu, "Breaking bad: De-anonymising entity types on the bitcoin blockchain using supervised machine learning," in *Proceedings of the 51st Hawaii International Conference on System Sciences (HICSS)*, United States, 2018, pp. 3497–3506.

[19] H. Tang, Y. Jiao, B. Huang, C. Lin, S. Goyal, and B. Wang, "Learning to classify blockchain peers according to their behavior sequences," *IEEE Access*, vol. 6, pp. 71 208–71 215, 2018.

[20] J. Hirshman, Y. Huang, and S. Macke, "Unsupervised approaches to detecting anomalous behavior in the bitcoin transaction network," *3rd ed. Technical report, Stanford University*, 2013.

[21] S. SAYADI, S. B. REJEB, and Z. CHOUKAIR, "Anomaly detection model over blockchain electronic transactions," in *Proceedings of the 15th International Wireless Communications Mobile Computing Conference (IWCMC)*, June 2019, pp. 895–900.

[22] J. Kim, M. Nakashima, W. Fan, S. Wuthier, X. Zhou, I. Kim, and S.-Y. Chang, "Anomaly detection based on traffic monitoring for secure blockchain networking," in *IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, Sydney Australia, May 2021.

[23] "Misbehavior score for punishing the misbehaving peer," 2021. [Online]. Available: https://github.com/bitcoin/bitcoin/blob/0.21/src/net_processing.cpp#L1049-L1065

[24] J. Schnelli, "Peer-to-peer communication encryption," 2016. [Online]. Available: https://github.com/bitcoin/bips/blob/master/bip-0151.mediawiki

[25] ——, "Version 2 peer-to-peer message transport protocol," 2019. [Online]. Available: https://github.com/bitcoin/bitcoin/pull/18242

[26] A. Miller and R. Jansen, "Shadow-bitcoin: Scalable simulation via direct execution of multi-threaded applications," in *8th Workshop on Cyber Security Experimentation and Test*, Washington, D.C., Aug. 2015.

[27] 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 (CCS)*, London, United Kingdom, 2019, pp. 817–831.

[28] C. Decker and R. Wattenhofer, "Information propagation in the bitcoin network," in *IEEE P2P 2013 Proceedings*, 2013, pp. 1–10.

[29] "The rlpx transport protocol," 2021. [Online]. Available: https://github.com/ethereum/devp2p/blob/master/rlpx.md

[30] J. Jiang, "P2p scoring system and network security," 2018. [Online]. Available: https://docs.ckb.dev/docs/rfcs/0007-scoring-system-and-network-security/0007-scoring-system-and-network-security

[31] DashCore, "Misbehaving nodes," 2021. [Online]. Available: https://dashcore.readme.io/docs/core-guide-p2p-network-misbehaving-nodes

[32] A. Abhishta, R. Joosten, S. Dragomiretskiy, and L. Nieuwenhuis, "Impact of successful ddos attacks on a major crypto-currency exchange," in *27th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)*, United States, March 2019.

[33] C. Decker and R. Wattenhofer, "Bitcoin transaction malleability and mtgox," in *19th European Symposium on Research in Computer Security (ESORICS)*, Wroclaw, Poland, Sept. 2014, pp. 313–326.

[34] K. Baqer, D. Y. Huang, D. McCoy, and N. Weaver, "Stressing out: Bitcoin "stress testing"," in *Financial Cryptography and Data Security*, 2016, pp. 3–18.

[35] M. Saad, V. Cook, L. Nguyen, M. T. Thai, and A. Mohaisen, "Partitioning attacks on bitcoin: Colliding space, time, and logic," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, 2019, pp. 1175–1187.

[36] J. Tapsell, R. Naeem Akram, and K. Markantonakis, "An evaluation of the security of the bitcoin peer-to-peer network," in *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, July 2018, pp. 1057–1062.

[37] 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 (CCS)*, Denver, Colorado, USA, 2015, pp. 692–705.

[38] A. Biryukov, D. Khovratovich, and I. Pustogarov, "Deanonymisation of clients in bitcoin p2p network," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, Scottsdale, Arizona, USA, 2014, pp. 15–29.

[39] W. Fan, J. Kim, I. Kim, X. Zhou, and S.-Y. Chang, "Performance analyses for applying machine learning on bitcoin miners," in *20th International Conference on Electronics, Information, and Communication (ICEIC)*, Jeju Shinhwa World, South Korea, Jan. 2021.

[40] W. Fan, H.-J. Hong, J. Kim, S. J. Wuthier, M. Nakashima, X. Zhou, E. Chow, and S.-Y. Chang, "Lightweight and identifier-oblivious engine for cryptocurrency networking anomaly detection," *IEEE Transactions on Dependable and Secure Computing*, pp. 1–1, 2022.

[41] W. Fan, H.-J. Hong, S. Wuthier, X. Zhou, Y. Bai, and S.-Y. Chang, "Security analyses of misbehavior tracking in bitcoin network," in *IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, Sydney, Australia, May 2021.