Tyler Frank
zaf455
CS 6393

An Analysis of Studies on Partitioning Attacks in Bitcoin and Proposed Countermeasures

Abstract.
This paper examines two studies of partitioning attacks on Bitcoin. The first study looks at multiple partitioning attacks and possible countermeasures for each. The second proposes a stealthy partitioning attack at the routing level and possible countermeasures. Contributions by the works are identified, as well as limitations. Last, areas for future research are identified.

I. Introduction.

Due to the popularity of the Bitcoin cryptocurrency, currently valued at over $59,000 USD [2], there exists significant incentive for malicious actors to attack Bitcoin. While there are a wide variety of attacks that may be conducted against Bitcoin, this paper focuses on reviewing current research on partitioning attacks due to the potential damage they may cause to Bitcoin or other cryptocurrencies. Multiple vectors exist for attackers to conduct partitioning attacks: network-level, time-based, and the decentralized nature of Bitcoin software.

Motivations for conducting a partition attack vary, and the following are just a few possible reasons. Mining pools may wish to conduct a partitioning attack on a competitor in order to increase the probability that the attacker will find the next block to be added to the blockchain and gain the associated rewards. Nation states may wish to undermine the value of Bitcoin or impact the economies of countries that have adopted Bitcoin as a currency. A large network level adversary may wish to partition off a large portion of the Bitcoin network's hash power, in effect increasing the attacker's portion and potentially allowing the attacker to conduct the 51% attack.

II. Partitioning Attacks on Bitcoin: Colliding Space, Time, and Logic.

Author Contributions.
While Bitcoin is designed to be a decentralized currency, current research has identified several trends of note that show that there is increasing centralization occurring at the Autonomous System (AS) and organization levels. Saad et al. identified several important features regarding the increase in centralization within the Bitcoin network. First, the number of ASes with 50% of the Bitcoin full nodes changed from 50 in 2017 to 24 in 2018, resulting in half of the full nodes being centralized by a factor of 52% [3]. Additionally, both the top 8 ASes and 8 organizations hosted and intercepted 30% of the Bitcoin network traffic respectively [3]. The top 3 ASes intercepted 65% of the mining pool traffic, while a single organization, *Alibaba* intercepted over 50% of the Bitcoin mining pool traffic by itself [3]. Last, [3] identified that 21 organizations intercepted 50% of the total Bitcoin network traffic.

Saad et al. also note that the current design of the internet relies on the Border Gateway Protocol (BGP), which is vulnerable to prefix hijacking. ASes and organizations are able to hijack their

neighbor's BGP routes by advertising more specific prefixes, allowing them to reroute traffic through their infrastructure. Saad et al. found that all ASes with the exception of AS 16509 are vulnerable when fewer than 40 BGP prefixes are hijacked [3]. AS 16509 required over 140 BGP prefixes to be hijacked to become compromised while AS 24940 is vulnerable with only 15 BGP prefixes hijacked [3]. Furthermore, by hijacking 3 ASes, an attacker can isolate over 60% of the Bitcoin hash power [3].

The ability to isolate portions of the Bitcoin network or hash power has large implications. As the value of Bitcoin is largely derived from the confidence investors have in it, any attack that undermines confidence may have a significant impact on the value of Bitcoin. Saad et al. notes that BGP hijacking only 3 ASes can isolate more than 60% of the total Bitcoin hash power potentially allowing mining pools with lower hash rates to attack competitors with the 51% attack due to the spatial partitioning [3]. In order to counter such attacks, Saad et al. proposes that mining pools spread their servers across multiple ASes, a this would increase the cost of the attack [3].

While BGP hijacking occurs at the routing level, Saad et al. also derived an attack based on temporal features of the Bitcoin network. These temporal features were identified by crawling data through the use of Bitnodes, as Bitnodes collects the response time, latency, uptime, latest block, IP addresses, organization, and location of a node [3]. Saad et al. collected Bitcoin network information at 10-minute intervals to gather information on the consensus distribution after a new block was published, as well as one-minute intervals to observe consensus pruning within the Bitcoin network between block publications [3].

During their data collection, [3] found that only 50% of full nodes are current in their view of the blockchain, 10% are forever behind, and the remaining nodes vary in their view of the blockchain. Furthermore, there are occasions where 7,000 nodes lag behind the main blockchain by 1-4 blocks and an attacker may partition ~90% of the network by isolating those lagging nodes [3]. Due to this observed delay, Saad et al. propose an optimization problem that balances between the maximum number of concurrently lagging nodes and having a minimum time required to catch up to the main blockchain [3].

By setting values for delay rate, m number of nodes the attacker seeks to isolate, and the success rate of the attacker, Saad et al. were able to derive a table that calculates the minimum timing constraint an attacker needs to isolate m nodes. Notably, with a success rate of 0.8, delay rate of 0.8, and $m = 500$, an attacker would need approximately 589 seconds to isolate all m nodes with the set success rate [3]. Furthermore, [3] propose that an attacker with 30% of the hash power may continue to sustain the isolated nodes with successive soft forks by feeding those nodes counterfeit blocks. This is due to isolated nodes assuming block delays are due to network issues, not the attacker's reduced hash power [3]. As the temporal attack is only a soft fork, it is possible for the isolated nodes to recover by rejecting the attacker's blocks and reversing all transactions of legitimate users, although a significant update will be required for each of the isolated node's UTXO sets, as well as a system-wide check on transactions being reversed [3]. Saad et al. also propose a countermeasure to these types of temporal attacks; *BlockAware,* which checks the time between the latest block and the current time, querying neighbors if the difference is greater than the expected 10-minute inter-block arrival time [3].

Saad et al. also notes that the previous two attacks can occur simultaneously. Full nodes who are current in their view of the blockchain being vulnerable to BGP hijacking, and nodes that lag behind being vulnerable to temporal attacks [3]. However, this requires that the attacker has both routing and mining capabilities, such as a cloud service provider [3].

The last type of partitioning attack identified by Saad et al. is through logical partitioning. As Bitcoin has no centralized mechanism to enforce software versions, there is a wide variety of Bitcoin clients being used. The most common client identified was the latest version (at the time) of the Bitcoin Core client with 36% of full nodes running 0.16.0 and an additional 27% using 0.15.1, with the remaining nodes using 286 different software clients [3]. Furthermore, 36 vulnerabilities were identified within the NIST National Vulnerability Database, with multiple CVE's being found in all client versions [3].

An attacker may incentivize nodes to use modified software through two possible vectors. First, an attacker may propose a new software variant that has better performance and features, Saad et al. note that "Falcon provides faster connectivity and minimum delay during transaction propagation" [3]. The second approach would be to modify the Bitcoin Core software in a way that appears normal but would help an attacker, such as modifying the number of peer connections from 8 to 11 [3]. Both routes may allow an attacker to increase their capabilities during a partitioning attack. Countermeasures for logical attacks proposed in this study was limited to maintaining the Bitcoin Core client as the favored client by providing the best results, as a centralized authority that validates software client versions would be against Bitcoin's principle of decentralization [3].

Limitations.
A limitation of the work presented in [3], is that the spatial attack proposed relies solely on BGP hijacking the target AS, which is globally observable. As the proposed ASes targeted belong to significant businesses such as AS16276 (Amazon), which has 5.54% of all Bitcoin network traffic, it is likely that the changes to BGP routes would be noticed quickly. This would greatly reduce the attacker's time window, although if the purpose of the attack is to damage the reputation of Bitcoin, this reduced window may be sufficient. Additionally, as the attacker is broadcasting the more specific prefixes, they are easily identified. This reduces the practicality of this type of attack as only ASes and Nation States have the capability to conduct this attack and are likely to face repercussions.

III. A Stealthier Partitioning Attack Against Bitcoin Peer-to-Peer Network.

Author Contributions.
Due to the exposure of the real identity of attackers that use BGP hijacking, Tran et al. propose EREBUS, an attack that leverages the fundamental topology advantages of being a large AS. In order to construct EREBUS, Tran et al. identified key features in both the Bitcoin protocol and capabilities that an AS-level attacker has to construct an attack method that is significantly harder to detect and allows attackers to deny conducting the attack [5].

Bitcoin manages peer connections by having a specific number of incoming and outgoing connections. These connections are determined from two internal tables of the Bitcoin node. EREBUS was constructed after the countermeasures for the Eclipse attack were adopted by the Bitcoin protocol. First, the size of the tables store within the node have increased in size four times because it increases the cost of the Eclipse attack [5]. Second, remote attackers are no longer able to force a Bitcoin node to make peer connections to their botnet [5].

Currently, Bitcoin nodes allow 8 outgoing connection and 117 incoming connections, with nodes accepting any incoming connections from peers with any IP addresses unless they have been banned for invalid messages [5]. Outgoing peers are selected from IP addresses within two internal tables, the NEW table with 65,536 slots and the TRIED table with 16,384 slots [5] The NEW table contains IP addresses that the node has received but has not connected to, while the TRIED table consists of IP addresses that have had a successful outgoing connection [5]. The 8 outgoing connections of a node are selected at random from both the TRIED and NEW tables until all outgoing connections are full [5]. When a node adds an IP address to the NEW table, it hashes the IP prefix group of the peer that relayed that IP address to determine which bucket, of which there are a total of 1,024 [5]. The combination of bucket index and IP prefix is hashed to identify the correct slot [5]. If the slot identified already has an IP address, the node checks to see if the current IP address is considered *terrible*, i.e., it is over 30 days old, or if it has failed connection attempts [5]. *Terrible* IP addresses will be replaced and stored with a timestamp, if the IP address is already in the table, the timestamp will be updated [5].

The TRIED table only allows IP addresses to be inserted by moving the IP address from the NEW table after a successful outgoing connection is made, which prevent attackers from inserting directly into the TRIED table [5]. Tran et al. note that there are 2 scenarios in which an IP address from the NEW table may be moved to the TRIED table. First, if an outgoing connection is made to an IP address in the NEW table and it is successful and second, every 2 minutes an IP address is randomly selected from the NEW table and moved to the TRIED table if it was successful [5]. Additionally, whenever an outgoing connection is created, there is an equal probability that the node will select an IP address from either the TRIED or NEW tables [5]. If the IP address was selected from the NEW table and the outgoing connection was successful, it is moved to the TRIED table and removed from the NEW table [5].

The goal of the EREBUS attack is to fill the TRIED and NEW tables with IP addresses that traverse through the adversary AS [5]. The IP addresses used by the attacker are called Shadow IPs and do not necessarily need to be used by real Bitcoin nodes or even any host [5]. Shadow IP addresses may be existing Bitcoin nodes, or any valid IP address whose victim-to-IP address routes traverse the adversary AS [5]. As the adversary is a large AS, it may collect shadow IP addresses by evaluating its inter-domain routing state, then enumerate all ASes that potentially have a node whose victim-to-node path crosses its own network [5]. The attacker then enumerates all available IP addresses in the identified ASes and adds them to an internal database, checks to see if packets from the victim node indeed do traverse its network by initiating a TCP connection and verifying that a SYN/ACK was received [5].

Tran et al. were able to identify that in 85% of cases, over 100 shadow ASes were available, 99.5% of the tier-1 ASes can target victim nodes in any AS with over 100 unique IP prefix

groups distributed across over millions of shadow IPs, and 80% of cases have over a million shadow IP addresses available [5]. The EREBUS attacker's goal is to influence the victim node's peer connections over a period of several weeks [5].

Peering connections were established by flooding the victim node's NEW table by selecting a shadow IP address and initiating a TCP connection and version handshake with a spoofed IP address [5]. The adversary sends ADDR messages containing 1,000 shadow IP addresses each, which are then inserted into the victim's NEW table [5]. Tran et al. found that it took approximately 30 days to fill the NEW table with 99% shadow IP addresses [5]. The attacker requires a minimum of 100 prefix groups to fill the NEW table, with tier-1 ASes having a 99.5% probability of having at least 100 [5]. Increasing the number of available prefixes to 500 allows an attacker to occupy the majority of the NEW table with as few as 500 ADDR messages.

Tran et al. found that within the first 25 days of attack execution, shadow IP addresses were inserted at a relatively low rate, however after 25 days the existing legitimate IP addresses begin to age past 30 days and becoming *terrible*, resulting in their evection, and increasing the rate of insertion of the attacker [5]. Victim Bitcoin nodes that were less than 50 days old were more susceptible to the EREBUS attack, while an attacker has an 18% success rate with nodes that are older than 50 days with only minimal reboots of the victim node [5]. In order to maintain the 2 shadow IP addresses per second to the victim node, an attacker only needs to maintain a traffic rate of 520 bits per second [5]. Furthermore, attacking multiple nodes does not require extended preparations or attack duration and results in only a linear increase in traffic [5].

The first countermeasure proposed by Tran et al. was the use of 3rd party proxies, although this solution has limited scalability and could lead to centralization [5]. Changes could be made to the Bitcoin protocol, the NEW and TRIED tables have been increased to combat the Eclipse attack 4 times. However, the increase in table size has made EREBUS have a higher success rate [5]. By doubling the number of outgoing connections, EREBUS' success rate is significantly reduced [5]. Combining a reduction in size of the IP tables as well as doubling the number of outgoing connections led to EREBUS only having a 5% success rate after 2 months [5]. Replacing the prefix groups with the AS number for peer selection makes EREBUS significantly harder or impossible for attackers whose shadow IP addresses are distributed in a large number of prefix groups but only hosted in a few ASes [5]. Last, an eviction policy that places a priority on maintaining connections where the most recent propagated block data is fairly current [5].

Limitations.

One of the limitations in the work presented by Tran et al. is only Bitcoin nodes that accepted incoming connections were considered, which numbered ~10,000, while the remaining nodes were considered out of scope [5]. Additionally, the authors did not attack real Bitcoin nodes due to time constraints and the inability to test different countermeasures [5].

Second, EREBUS does reboot the victim node if the probability of occupying all outcoming connections reaches a specific threshold [5]. In order to have a maximum attack impact, full nodes belonging to major mining pools would need to be targeted. It is highly likely that major mining pools would become suspicious when their full node(s) reboot multiple times. This may trigger a deep packet inspection identifying the EREBUS attack. While the victim would still be

unable to attribute the attack to the attacker due to the nature of the spoofed IP addresses, the nonetheless would be able to identify that the attack is still taking place and may consider potential countermeasures.

IV. Suggestions For Future Research

The two studies examined in this paper took place when Stratum V1 was the dominant mining protocol. However, since then, Stratum V2 has been announced. Stratum V2 is designed to be more efficient and to increase security of mining pools [4]. Notable changes with respect to the studies examined in this paper include the following. First, Stratum V2 has been designed to reduce bandwidth consumption to increase the amount of hashing results that are transmitted [4]. This may potentially decrease the number of peering connection messages, possibly inhibiting the EREBUS attack.

Another change included in Stratum V2 is the ability for mining pools to send jobs to workers early for future blocks, even before prior blocks have been found [4]. This may impact the ability of an attacker during a partitioning attack to effectively soft fork the partitioned victims.

Stratum V2 also adds a feature specifically designed to prevent a man-in-the-middle attack. Authenticated Encrypted with Associated Data (AEAD) provides both confidentiality and integrity for encrypted data being transferred, as well as integrity for non-encrypted data [4]. Both studies were conducted on Stratum V1 which does not support encryption. Further research should be directed on how this impacts the proposed attack methods.

Miners also have the ability to choose their own transaction set under Stratum V2, resulting in increased decentralization [4]. This feature is designed to move some of the Bitcoin hash power from the mining pools to the individual miners themselves [4]. Currently, the success of the partitioning attacks depends on the ability of an attacker to be able to partition off a large portion of the Bitcoin network's hash power and as a result, increase the attacker's portion of the total Bitcoin network hash power. This potentially allows an attacker with less than 51% of the total hash power to conduct a 51% attack. Further research should examine how the increase in decentralization impacts the viability of the proposed attacks.

Further research should also examine whether combining the EREBUS attack proposed by Tran et al. and the temporal attack proposed by Saad et al. results in a more effective partitioning attack. Nation state adversaries that are interested in undermining Bitcoin have the capabilities to conduct both attacks simultaneously, and the increased difficulty of detecting the EREBUS attack combined with exploiting the non-uniformity of consensus pruning may be enticing for such adversaries.

Another area that should be investigated is the optimal size for the NEW and TRIED tables of a Bitcoin node. These internal tables were increased in size to increase the cost of the Eclipse attack [5], however the increase in table size also makes the EREBUS attack more likely to succeed. There is the possibility of an optimal solution for table sizes to maximize the cost of both attacks. Furthermore, an additional area is to identify the optimal number of connections to

combat the EREBUS attack, as doubling the outgoing connections greatly reduced EREBUS' effectiveness [5]. As any change in the size of the internal tables or number of connections would require a change to the Bitcoin protocol, further study is merited in this area.

Lastly, Apostolaki et al. propose a secure Bitcoin relay network (SABRE) that is designed to prevent hijacking attacks such as the ones proposed by Saad et al. and Tran et al. [1]. Tran et al. included discussion about the limitations of such secure relays, specifically SABRE. Concerns identified were requiring blind trust in secure relays and the possibility that malicious ASes can provide their own SABRE relays to make their hijacking attacks easier [5]. Both the security aspect and the addition of a malicious SABRE relay by an adversary AS should be studied further.

V. Conclusion

Bitcoin relies heavily on both the current Internet and a decentralized model, both of which are vulnerable to partitioning attacks. The current Internet routing model uses BGP, which is vulnerable to prefix hijacking as described by Saad et al., although the real identity of an attacker is immediately apparent to the public. Tran et al. proposed EREBUS, in which a network-level adversary takes advantage of its network topology to dominate peer connections, allowing for the same capabilities of BGP hijacking without the identity of the attacker being compromised.

The decentralized nature of Bitcoin also results in conditions that an attacker may exploit. Saad et al. examined the consensus protocol and identified that only half of the Bitcoin full nodes stay current on the state of the blockchain, with the remaining nodes being vulnerable to soft forks [3]. Additionally, as there is no centralized capability to enforce software client versions, an attacker may create their own or modify the Bitcoin core client in ways that amplify partitioning attacks.

Lastly, the introduction of Stratum V2 has the potential to have a significant impact on the success rates of the proposed partitioning attacks. Further research should focus on how the introduction of Stratum V2, the size of the NEW and TRIED tables, as well as secure Bitcoin relays such as SABRE affect the current proposed partitioning attacks.

References

1. Apostolaki, M., Marti, G., Muller, J., & Vanbever, L. (2019). Sabre: Protecting Bitcoin Against Routing Attacks. *Proceedings 2019 Network and Distributed System Security Symposium*. https://doi.org/10.14722/ndss.2019.23252
2. *Bitcoin (BTC) price, charts, and news: Coinbase: Bitcoin Price, BTC Price, Bitcoin Coinbase*. Coinbase. (n.d.). Retrieved November 17, 2021, from https://www.coinbase.com/price/bitcoin.
3. Saad, M., Cook, V., Nguyen, L., Thai, M. T., & Mohaisen, A. (2019). Partitioning Attacks on Bitcoin: Colliding Space, Time, and Logic. *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*. https://doi.org/10.1109/icdcs.2019.00119

4. Stratum V2: The next generation protocol for pooled mining. Stratum V2 | The next generation protocol for pooled mining. (n.d.). Retrieved November 21, 2021, from https://braiins.com/stratum-v2?utm_source=help.
5. Tran, M., Choi, I., Moon, G. J., Vu, A. V., & Kang, M. S. (2020). A Stealthier Partitioning Attack Against Bitcoin Peer-to-Peer Network. *2020 IEEE Symposium on Security and Privacy (SP)*. https://doi.org/10.1109/sp40000.2020.00027