Blockchain Focus

DoS on Blockchains: A P2P Layer's View

Sep 27, 2024
Jaehyun Ha | Research Analyst
jaehyunha@prestolabs.io

Contents

Summary

- Blockchain's P2P layer plays a crucial role in maintaining highly reliable connectivity across distributed networks, by establishing the rules for how the nodes connect and communicate with each other. Its key properties include Node Discovery, Node Connectivity, and Data Propagation.

- P2P layer security is less well-known and often overlooked compared to threats like smart contract hacking or consensus failure. However, if the P2P layer is compromised, serious attacks like double spending and selfish mining can be carried out with fewer resources.

- Attacks like the Eclipse Attack and Erebus Attack were developed through academic research to improve mitigations for the blockchain's P2P layer, particularly against DoS attacks. While significant security research has been done for major chains like Bitcoin and Ethereum, smaller emerging chains often lack proper mitigation measures, leading to instances of real-world damage. Therefore, it is crucial to pay more attention to the security of the P2P layer.

---

1. Introduction

As blockchain technology continues to evolve, so do the security threats facing its ecosystem. While security issues like wallet hacking and smart contract vulnerabilities are often top of mind, a more insidious threat is also gaining attention: Denial of Service (DoS) on blockchain networks. For those closely following recent incidents in the blockchain industry, it's clear that these types of threats can lead to severe and prolonged network outages (or safety issues), disrupting the entire system.

A typical DoS occurs when a system, server, or network becomes overwhelmed by excessive traffic or resource requests, rendering it unable to handle legitimate user interactions and effectively function. At this point, one might ask: "Isn't the blockchain network designed to solve the Single Point of Failure (SPOF) problem, as a distributed system?". Unlike the traditional client-server model, where network outages are inevitable if the centralized server fails, a Peer-to-Peer (P2P) network like blockchain operates on the principle that every participant acts as both a server and a client. If one peer goes down, one can simply sync with the network from another peer. So how can such a decentralized system still be vulnerable to DoS attacks?

While decentralization offers resilience, it doesn't make a network immune to all types of DoS vulnerabilities, particularly those stemming from network congestion and resource exhaustion. As blockchain systems evolve to become more decentralized and incorporate diverse components to address various needs, a paradox emerges: the very complexity that enhances functionality also introduces more points of failure (i.e., attack vectors). Today, the targets of DoS attacks in the blockchain ecosystem are wide-ranging and include crypto wallets, centralized exchanges (CEXs), mempools, mining pools, payment channels, smart contracts, and even consensus participants.

In this report, we will specifically focus on DoS attacks targeting blockchain consensus participants from the perspective of the P2P layer. After covering the fundamentals of the P2P layer in blockchain in Section 2, Section 3 will explain why security in the P2P layer is crucial. Then, in Section 4, we will discuss the various types of DoS attacks that have been developed targeting the P2P layer, and finally, in Section 5, we will address how tangible these threats really are.

2. Understanding P2P Layer in Blockchains

Blockchain systems are desired to maintain highly reliable network connectivity across distributed nodes, even in the face of significant network attacks or failures. Here, the P2P layer plays a crucial role in supporting this by establishing the rules for how the nodes connect and communicate with each other, ensuring that the system remains stable and resilient. The P2P layer performs a variety of tasks, but here we will focus on its three key properties: **Node discovery, Node connectivity, and Data propagation**.

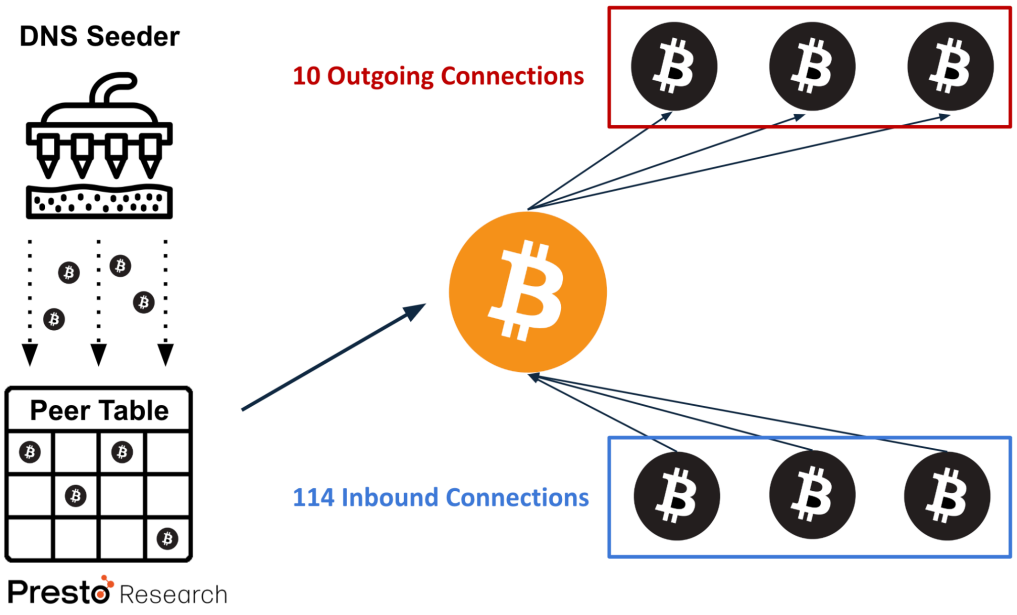
Node Discovery

If Bob has newly joined a permissionless blockchain network, the first thing he needs to do is find a new friend (i.e., peer nodes) to connect and sync with. However, since it's a new environment for Bob, he should be careful not to be friends with malicious nodes, hence he first has to connect to a few well-known or pre-defined nodes to learn about legit peers in the network.

In Bitcoin, node discovery begins with the use of **DNS seeders**, which are predefined servers that provide a list of active nodes to new clients joining the network. When a Bitcoin node starts up, it first queries these DNS seeders, which return a set of IP addresses of known nodes. This initial peer list is stored in each nodes' peer database called "peer tables", and allows the new node (like Bob) to select and establish a maximum of 10 outgoing connections among them. Once connected, Bob can begin participating in the network (i.e., downloading the blockchain, validating blocks & transactions).

Bitcoin's node discovery protocol helps Bob maintain and update peer tables by gossiping peer addresses across the network, and allowing him to establish new outgoing connections if any of his existing ones are terminated. Additionally, Bob's IP address is also gossiped throughout the network, enabling up to 114 inbound connections from other nodes. As a result, Bob can establish up to 125 peer connections in total, thanks to the node discovery protocol. *(*10 outgoing + 114 inbound = 124 connections, why 125 in total? For those who are curious about this, please look into feeler connections)*

Figure 1: Bitcoin's Node Discovery Protocol



Source: Presto Research

Ethereum takes a more structured approach to node discovery using the Kademlia DHT-based protocol. Similar to Bitcoin, when a new Ethereum node joins, it uses bootnodes to start discovering other nodes. From there, it queries these nodes, learns about others in the network, and populates its routing table accordingly. In Ethereum's network, each node has a unique node ID, derived from its public key, which is mapped into a large 256-bit space. Node discovery in Ethereum relies on this ID space, where nodes maintain routing tables filled with other nodes that are "close" (i.e., the XOR distance) in terms of their ID.

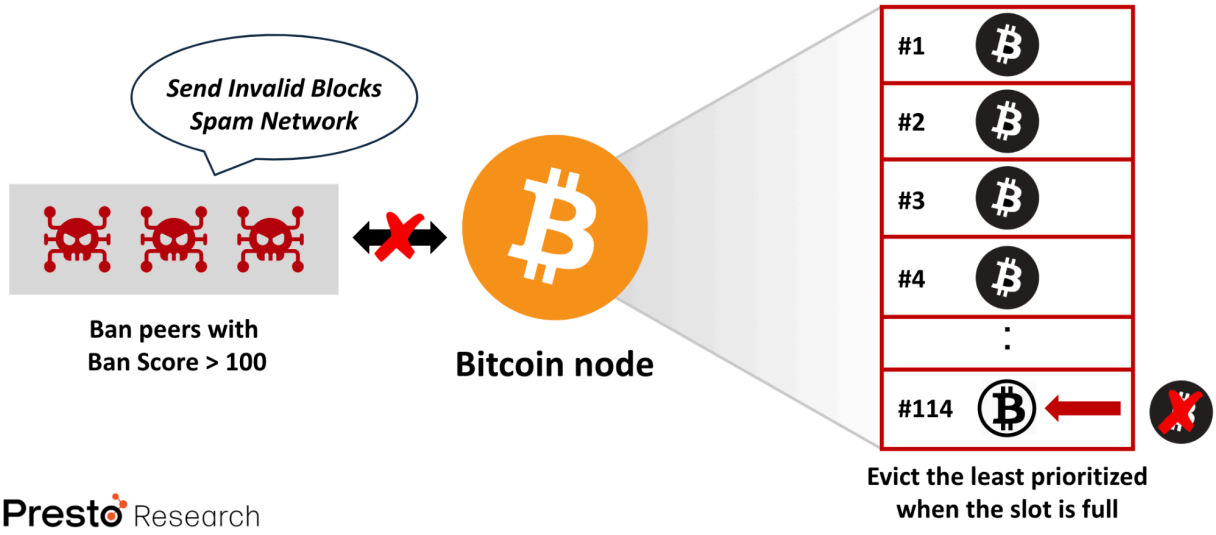
Node Connectivity

After Bob has found his friends, he now needs to decide whether to maintain those friendships. He might reconsider staying friends with those who have misbehaved or decide to part ways with some old friends to make room for new ones.

In Bitcoin, when a node detects harmful behavior—such as sending invalid blocks, spamming the network with redundant data, or violating consensus rules—it assigns a penalty score to the offending peer. The score is assigned based on the severity of the misbehaviour, with 10, 20, or even 100 points given depending on the offense. If the score exceeds a predefined threshold (i.e.,100), the node automatically bans the peer, preventing any further communication with it for a set period (i.e., 24 hours). During this ban, the node refuses connections and messages from the offending peer.

The node connectivity protocol also plays a role when a new inbound connection attempt is made to a node whose connection slots are already full. In this case, Bitcoin runs its peer eviction mechanism—each node prioritizes its existing connections based on several rules (i.e., sent transactions & blocks recently) then evicts the lowest-priority connection to make room for new ones. Ethereum currently did not deploy a peer eviction mechanism in their client implementation.

Figure 2: Bitcoin's Node Connectivity Management



Source: Presto Research

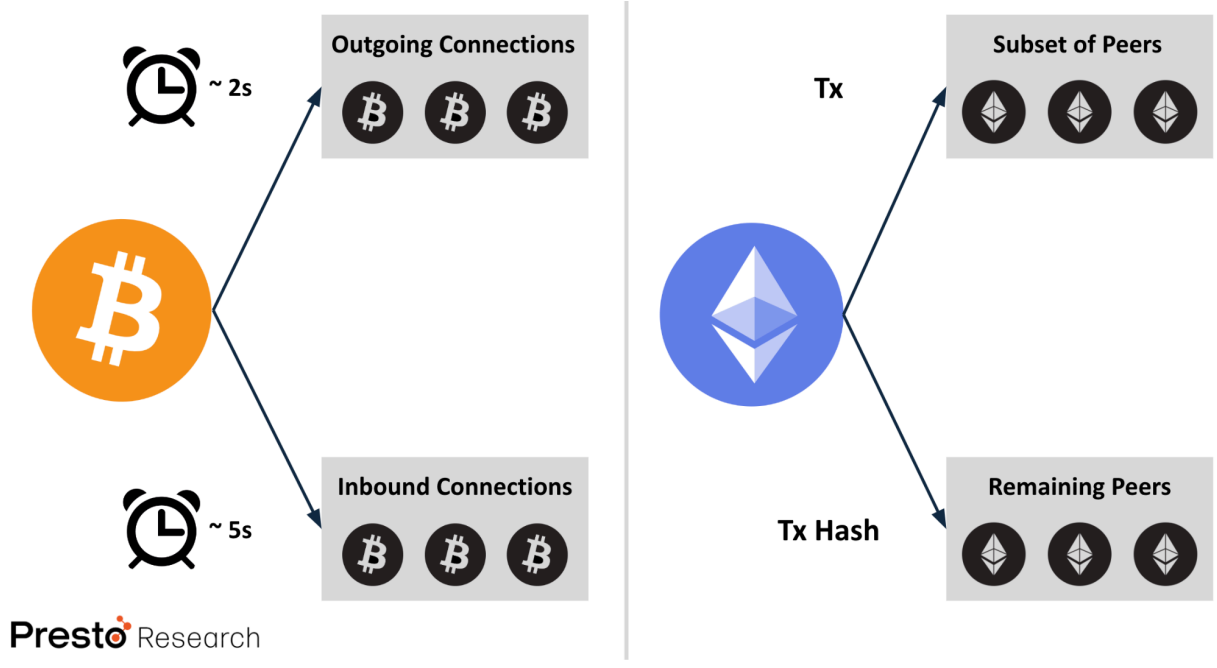
Data Propagation

After reliable network connectivity is achieved through node discovery and node connectivity protocols, now it's time to propagate the data (i.e., blocks and transactions) across the network. However, this process also requires careful protocol design. If, in a naive approach, every node in the blockchain network immediately broadcasted new blocks or transactions to all their connected peers as soon as they received them, the network would be flooded with redundant copies of the same transaction. Peers would receive the same transaction multiple times from different nodes, which can lead to serious network congestion and resource exhaustion.

That's not the end. Such an approach could also threaten transaction privacy. While block explorers allow us to see which addresses are sending assets to others and the amounts, the real-world identities, such as IP addresses, remain hidden (and they should be). However, if transactions are propagated to all peers immediately without any delay, an attacker equipping many nodes could potentially infer who first created and broadcasted the specific transaction. Since nodes know the IP addresses of their connected peers, this could compromise transaction privacy.

For these reasons, each blockchain network has implemented its own data propagation protocols, known as the transaction diffusion process. In Bitcoin's case, each node advertises data (that they have new information: INV messages) to its peers with independent, random delays. For inbound connections, the node generates a random delay with an average of 5 seconds, while for outgoing connections, the delay averages 2 seconds. For Ethereum, the nodes broadcast newly learned transactions without delay to subset ($\sqrt{n}$) of its neighbour peers, and send transaction hashes to the remaining ones.

Figure 3: Data Propagation in Bitcoin and Ethereum



Source: Presto Research

How It Differs From Consensus Layer

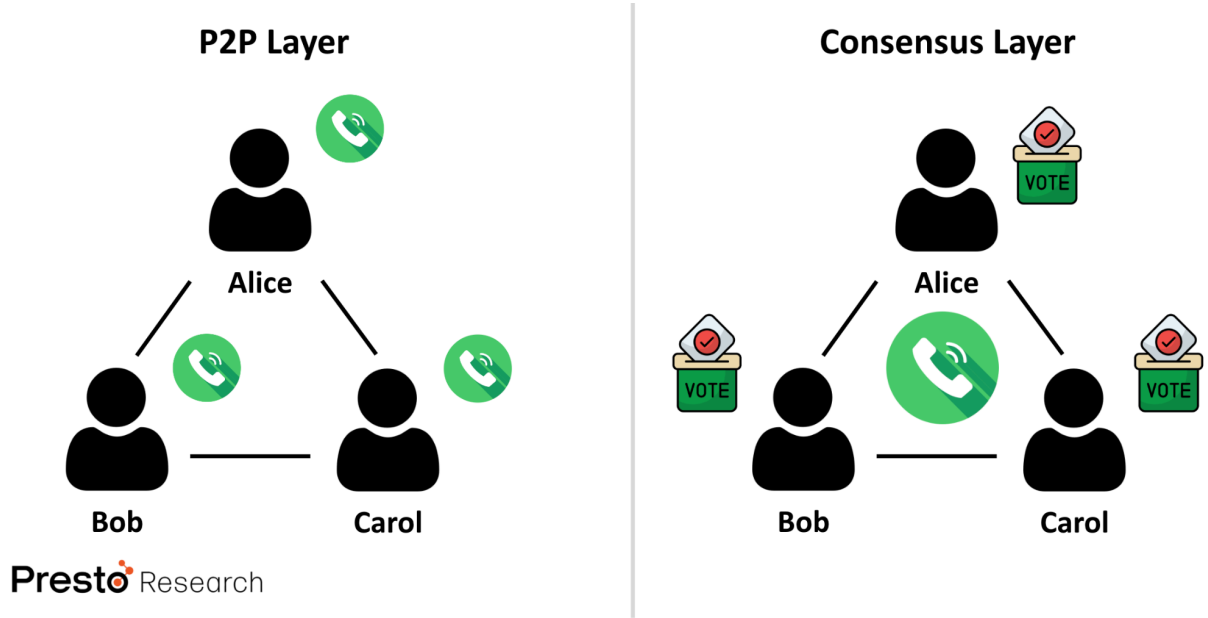
At this point, some might wonder: “So, the P2P layer in blockchain deals with the communication between the nodes. But it seems like the consensus layer also involves a lot of communication between nodes to reach an agreement. So, how are these two different?”

As an easy metaphor to understand the difference between the two, imagine a group of friends trying to decide on a movie to watch together. **The P2P layer is like the method they use to communicate**. Maybe they’re in the same room talking, or they’re using a group chat or video call. This layer ensures that everyone can send and receive messages, so the group can hear each other’s movie suggestions and responses. It handles how the information gets from one person to another, making sure everyone is connected.

The consensus layer, **on the other hand, is like the decision-making process they follow to agree on a movie**. Let’s say they all have to vote on the options, and whichever movie gets the most votes wins. This layer ensures that they all come to a single decision, preventing confusion, like two different groups watching separate movies at the same time. It’s the system or rules they use to make sure everyone ends up agreeing on the same result.

In short, the P2P layer is about connection between the nodes (the communication), while the consensus layer is about agreeing on the decision (the decision-making rules). Both are essential for the group to decide on a movie, just as they are for a blockchain to function properly.

Figure 4: How P2P Layer Differs from Consensus Layer



Source: Presto Research

3. Why is P2P Layer Security in Blockchains Important?

Now the next question is, why do we have to know such details about the P2P layer, and why is securing it important in blockchains? Let's quickly recap what we have discussed in the previous section: **Node discovery** finds legitimate nodes to connect and sync with, **Node connectivity** maintains healthy connections, and the **Data propagation process** broadcasts transactions across the entire network. What if all these do not function well (or are attacked)?

Figure 5: What Happens if P2P Layer do not Function Well?

P2P Layer Function	P2P Layer Malfunction (or Attacked)
Finds legitimate nodes to connect and sync with	Erroneous bootstrapping, connect to malicious nodes
Maintains healthy connections	Cannot terminate malicious connections
Transactions and blocks propagated throughout the network	Weak synchronization in entire network

Source: Presto Research

If node discovery doesn't work properly, new nodes like Bob may face issues during the bootstrapping process and are more likely to connect to malicious or faulty nodes. If node connectivity fails, malicious connections cannot be terminated, leaving Bob flooded from crafted blocks and transactions. Lastly, if data propagation is disrupted, it can lead to poor synchronization across the network, increasing the risk of forks or preventing nodes from syncing with legitimate blocks and transactions.

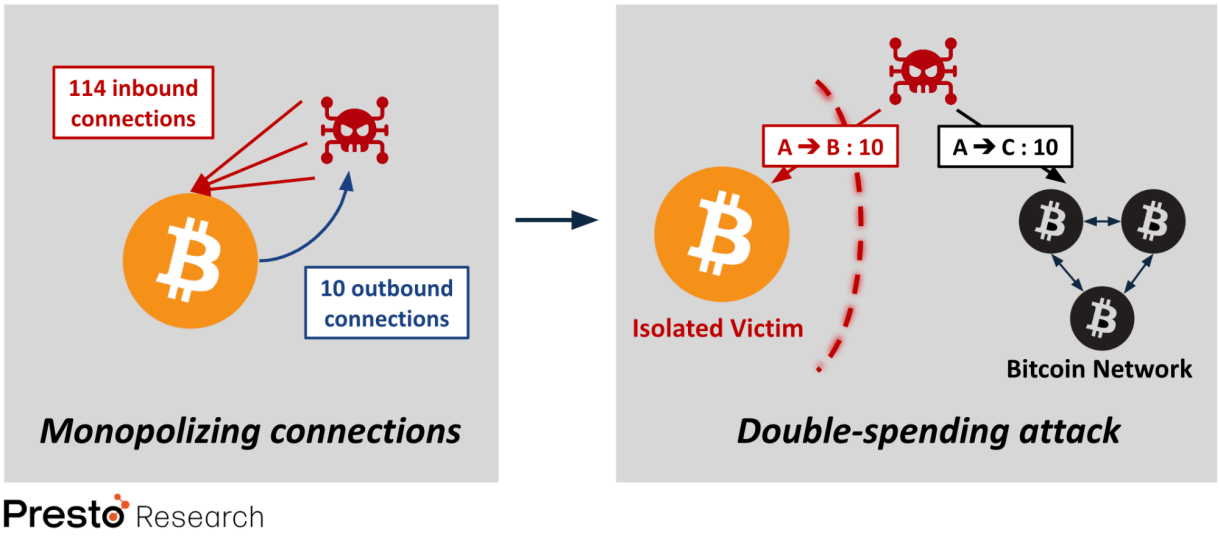
Overall, if the P2P layer does not function properly, **this leads individual (or groups of) network participants to be partitioned (isolated) from the rest of the legitimate network**. Victims' connections will be monopolized by faulty connections, wrong information will keep flowing into them, and there is less chance to re-align with the canonical chain due to weak synchronization (Denial of Service).

The real danger of such a partitioned state is that, **partitioned victims are more vulnerable to extra exploitations, such as double spending attacks**. Double spending attacks were originally known to be something that could only be carried out by attackers with substantial resources, such as in the case of a 51% attack.

For example, in the case of a 51% attack, the attacker can first make a legitimate transaction to a merchant (i.e., the victim), then secretly build an alternate chain in which this transaction does not exist. Once the attacker's alternate chain surpasses the old one and becomes the canonical chain (since it has majority of mining/voting power), they broadcast it to the network, causing the network to accept their new version as the valid one. The previous transaction on the original chain is then invalidated, meaning the attacker regains control of the spent coins, enabling them to "double spend". This attack is indeed a very powerful one, but it is considered an unrealistic theoretical attack because, for current mainstream PoW and PoS chains, it is economically infeasible for an attacker to control the majority of the mining power or voting power.

However, if a victim merchant is partitioned from the rest of the network, a double-spending attack becomes much easier, even without many resources. Since the victim is only connected to the attacker's node, they have no choice but to accept the transactions and blocks provided by the attacker as legitimate. This allows the attacker to use their assets twice—once with the victim and once with the rest of the network—without the victim realizing the assets have already been spent. The victim, believing the crafted transaction is valid, hands over the goods and receives the payment. However, once the partition is lifted and legitimate transactions and blocks from the outside network are restored, the transaction with the victim becomes invalid, while the one sent to the rest of the network remains valid, causing the victim to suffer a financial loss (Figure 6).

Figure 6: Network Partition Leads to Extra Exploitations (e,g., Double-Spending)



Source: Presto Research

4. What Kind of DoS Attacks Exist?

In the previous section, we explained how DoS (i.e., network partitioning) in the P2P layer is dangerous because it can enable attacks like double-spending without needing substantial resources, such as those required for a 51% attack. However, in practice, isolating a victim from the network is not an easy task: if suspicious connection attempts are detected (e.g., 1K connection attempts from similar IPs), victims can also manually block those malicious connections. Thus, in this section, we will explore the efforts made in academia to implement network partitioning in a more stealthy, feasible, and sustainable way.

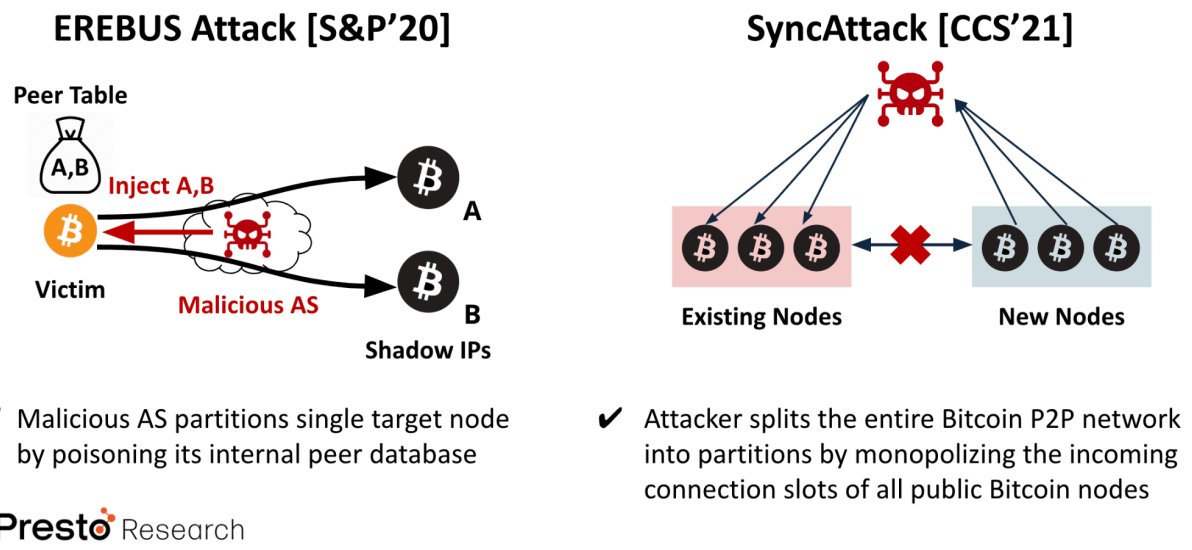
Connection Starvation Attacks

DoS attacks on P2P layers are usually referred to as **connection starvation attacks**. As explained in the previous section (Figure 6), this attack effectively isolates one or group of victim nodes from the rest of the network by occupying all of their available connection slots.

The concept of a connection starvation attack targeting blockchain networks' P2P layer was first introduced by the **Eclipse Attack** (E. Heilman et al., USENIX' 15). The core idea of this attack is straightforward: the attacker deploys a botnet (i.e., network of compromised computers, controlled by an attacker) and sends numerous connection attempts to a victim Bitcoin node with diverse IPs, thereby

occupying all of the victim's connection slots and preventing other legitimate Bitcoin peers from establishing new connections with the victim node. Unless the attacker releases the self-imposed partitioning attack, the victim node becomes unable to communicate with the rest of the P2P network and remains isolated. Not only suffering from exploitations like double spending attacks, the eclipsed victim can also suffer from transaction censorship: their transactions can be censored and dropped by the adversary surrounding them.

Figure 7: Connection Starvation Attacks



Source: Presto Research

In the **Erebus Attack** (M. Tran et al., S&P' 20), a malicious Autonomous System (AS), which can be a tier-1 or tier-2 Internet Service Provider, dominates the peer tables of a victim node with IP addresses under the control of the adversary. As a result, any connections initiated by the victim to these IPs will be routed through the adversary AS, granting it an advantage as a man-in-the-middle attacker. The target becomes isolated when all of its peer connections are routed through the adversary's AS. The novelty of this attack is the **stealthiness**: Since the attack's payload transmission rate is extremely low (520 bit/s), and there is no manipulation of routing in either the data plane or the control plane, the Erebus attack is regarded as a stealthy attack which is very hard to detect. The original study indicates that the victim node can be isolated potentially within 5-6 weeks of attack execution.

SyncAttack (M. Saad et al., CCS' 21) exploits the permissionless nature of Bitcoin to split the entire Bitcoin network into two separate groups. The attacker achieves this by taking control of all inbound connections of reachable nodes currently present in the Bitcoin P2P network while flooding the Bitcoin DNS seeder with the adversary's IPs. As a result, newly-joined nodes are forced to connect only to adversarial nodes. Consequently, the Bitcoin P2P network becomes divided into two distinct groups: the pre-existing node group and the newly entering node group.

5. Conclusion: Are DoS Attacks against P2P Layers a Real Threat?

So far, we have explored the role of the P2P layer in blockchain networks, why its security is crucial, and the relevant academic research that has been conducted in this area. Those who have been following this article from the beginning might wonder: “Now I understand that the P2P layer plays an important role in blockchains, and the DoS attacks at this layer can cause serious damage to victims. But why haven’t I heard about DoS attack incidents on the P2P layer in the news? Is this really practical?”

That’s a valid point. It is true that DoS attacks against P2P layers occur less frequently and are less well-known compared to other types of attacks targeting blockchain systems. Moreover, the attacks mentioned above often assume a fairly powerful attacker (e.g., botnets, ISP as an attacker). It is also because the targets of these connection starvation attacks are typically a small group of nodes rather than the entire network, making these incidents less noticeable compared to other attack cases where the entire network experiences a shutdown.

However, there are real-world examples of DoS attacks targeting the P2P layer of blockchains. In 2020, Monero experienced an eclipse attack from an attacker involving about 130 IP addresses, causing several users’ transactions to be delayed for several minutes. The Gethlighting Attack paper, published at NDSS ‘23, demonstrated that a DoS attack against Ethereum clients is possible even without needing to eclipse all of a target’s peer connections. The Ethereum Foundation acknowledged this vulnerability, which led to a hotfix in Geth 1.11.0. This shows that connection starvation attacks on the P2P layer remain a significant threat, affecting both large and small blockchain networks today.

Although blockchain systems are designed to be decentralized, by 2024, practical challenges have led to the emergence of prioritized entities within these networks. If these entities become the target of DoS attacks in P2P layers that disrupt specific nodes, the impact could extend far beyond a few minutes of dropped transactions or minor financial losses—it could put the entire system at risk. This makes it essential to continue researching mitigation strategies for DoS attacks at the P2P layer, and the public should become more aware of the importance of addressing this issue.

About Presto

Founded in 2014, Presto is a proprietary trading and financial services firm specializing in algorithmic trading across both digital assets and traditional markets. With a focus on delivering exceptional value for clients through a rigorous, research-driven approach to investment and trade execution, Presto processes over 100 million trades daily. The company maintains a global presence with offices in various countries, including Singapore. Presto Research is a research unit within Presto.

Find out more at https://www.prestolabs.io.
Follow Presto for more content: X, LinkedIn
Follow Presto Research for latest research : X, Telegram

Authors

Jaehyun Ha, Research Analyst
X, Telegram, LinkedIn

Required Disclosures

Any expression of opinion (which may be subject to change without notice) is personal to the author and the author makes no guarantee of any sort regarding accuracy or completeness of any information or analysis supplied. The views and opinions expressed herein are those of the author(s) and do not necessarily reflect the views of Presto Labs or its affiliates. This material by itself, is not and should not be construed as an offer or a solicitation to deal in any investment product or to enter into any legal relations. This material is for informational purposes only and is only intended for sophisticated investors, and is not intended to provide accounting, legal, or tax advice, or investment recommendations, or an official statement of Presto Labs or its affiliates. Presto Labs, its affiliates and its employees make no representation and assume no liability to the accuracy or completeness of the information provided. Presto Labs, its affiliates and its employees also do not warrant that such information and publications are accurate, up to date or applicable to the circumstances of any particular case. Certain statements in this document provide predictions and there is no guarantee that such predictions are currently accurate or will ultimately be realized. Prior results that are presented here are not guaranteed and prior results do not guarantee future performance. Recipients should consult their advisors before making any investment decision. Presto Labs or its affiliates may have financial interests in, or relationships with, some of the assets, entities and/or publications discussed or otherwise referenced in the materials. Certain links that may be provided in the materials are provided for convenience and do not imply Presto Labs' endorsement, or approval of any third-party websites or their content. Any use, review, retransmission, distribution, or reproduction of these materials, in whole or in part, is strictly prohibited in any form without the express written approval of Presto Labs. Presto Research and related logos are trademarks of Presto Labs, or its affiliates.