

Initial Evidence of Pervasive Reconnaissance Targeting Ethereum Node Infrastructure

Scott Seidenberger
University of Oklahoma
Norman, OK, USA
seidenberger@ou.edu

Anindya Maiti
University of Oklahoma
Norman, OK, USA
am@ou.edu

Abstract

We hypothesize that P2P overlay network nodes are attractive to attackers due to their visibility and uptime. To test this, we study active reconnaissance and exploitation on Ethereum nodes by deploying a series of honeypots alongside actual Ethereum nodes across globally distributed vantage points. We show that Ethereum nodes face heightened reconnaissance and exploitation attempts targeting specific ports. Furthermore, we find evidence that the threat assessment on our nodes is applicable to the wider P2P network by having performed port scans on other reachable peers. Our findings provide insights into potential mitigation strategies to improve the security of the P2P networking layer.

ACM Reference Format:

Scott Seidenberger and Anindya Maiti. 2025. Initial Evidence of Pervasive Reconnaissance Targeting Ethereum Node Infrastructure. In *The 7th ACM International Symposium on Blockchain and Secure Critical Infrastructure (BSCI '25)*, August 25–29, 2025, Hanoi, Vietnam. ACM, New York, NY, USA, 8 pages. <https://doi.org/10.1145/3709016.3737797>

1 Introduction

Cyber threats are rapidly evolving, with attackers increasingly targeting Internet-connected devices [8, 19]. These devices are attractive for malicious activities, such as deploying malware, acting as command-and-control (C2) intermediaries, or serving as bot-net components. Peer-to-peer (P2P) networks, foundational for decentralized applications, file sharing, and blockchain, make their nodes accessible through open routing information [20]. While essential for network functionality, this transparency inadvertently exposes nodes to significant risks [11]. However, systematic threat assessments focusing on P2P nodes' security remain scarce [34].

This short paper presents an initial study into elevated reconnaissance activities targeting nodes in P2P overlay networks. To investigate these dynamics, we introduce Ethereum as a case study due to its prominent role as one of the largest and most active P2P networks. Ethereum supports over 8,000 publicly accessible nodes globally [18], which regularly broadcast routing and metadata information to support decentralized network functionalities. This open architecture and high visibility is both critical for decentralized connectivity and a potential liability. In Ethereum (and similar PoS blockchains), a few compromised nodes may degrade local connectivity or hamper block propagation, though the network

as a whole is designed to tolerate some proportion of adversarial nodes. However, even partial compromise can be costly (e.g., forced re-syncing, lost attestations, and validator inactivity penalties.)

Threat Model. In this work, we assume adversaries are remote entities scanning IPs to identify publicly reachable Ethereum P2P nodes. Once located, adversaries may perform brute-force credential attempts or exploit known vulnerabilities to compromise the node. The target is the underlying server or cloud instance hosting the Ethereum node rather than the node's cryptographic keys (since beacon nodes do not store private keys for funds). A compromised node could disrupt block propagation, pivot into cloud environments, or launch reflection attacks. We assume attackers can collect P2P node IPs from publicly advertised routing tables and from prior scanning frameworks. We also account for background scanning from benign or commercial entities.

Our contributions in this work are as follows:

- **Measurement:** We deployed a data collection infrastructure on AWS across five geographic regions, including honeypots and beacon nodes, to analyze P2P interactions and the stability of connections across geographical regions.
- **Analysis of Attack Patterns:** Using statistical models accounting for geographic variations, we compared attack patterns on Ethereum nodes and control nodes. Ethereum nodes faced significantly more attacks, including targeted URI requests for sensitive files and cloud metadata.
- **Assessment of Attack Strategies:** SSH login attempts were more frequent on Ethereum nodes, indicating focused attack strategies with reused passwords. Scanning TCP ports revealed many Ethereum nodes had open SSH and RPC ports, increasing their attack surface and highlighting the need for strong security measures.

These findings highlight the need to conduct similar measurement studies on other P2P networks to better understand and mitigate potential security vulnerabilities across decentralized systems. Based on our observations, we recommend network-layer enhancements, such as privacy-preserving routing protocols, and advocate for collaborative threat intelligence sharing. This work serves as a foundation for future studies on securing critical infrastructure in decentralized networks.

2 Background and Related Works

Cloud-based honeypots enable large-scale cyber threat intelligence [34, 41], and often complement intrusion detection systems (IDS) [38]. However, scaling data collection in large honeypot deployments, or honeynets, presents operational challenges [1, 6]. Leveraging prior work, we have deployed a horizontally scalable approach for collecting data from our honeynet alongside Ethereum node



This work is licensed under a Creative Commons Attribution 4.0 International License. *BSCI '25, Hanoi, Vietnam*

© 2025 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-1412-2/2025/08
<https://doi.org/10.1145/3709016.3737797>

metrics and network session data [2]. Complementary to honeypots [39], darknet sensors monitor unused IP address spaces for unsolicited traffic, offering an alternative perspective in internet-scale monitoring [14].

Our methodology extends single-protocol honeypots by utilizing meta-honeypots that handle non-standard services and ports [12, 17, 21, 40], allowing us to collect a broader spectrum of attack vectors. Similar security challenges have been observed in Internet of Things (IoT) devices, which are always on, unattended, and share application-specific characteristics with P2P network nodes [43, 45]. Attackers exploit these vulnerabilities to orchestrate large-scale attacks like the Mirai botnet’s DDoS attacks [3]. We hypothesize that P2P nodes, sharing these characteristics and being more resource-rich, are appealing new targets for attackers. In P2P overlay networks, the exploitation of public routing information is an inherent vulnerability [24]. Studies have analyzed risks from de-anonymized routing data, exposing overlay networks to targeted attacks [7, 30]. Specifically for Ethereum, measurement studies have examined its topology and P2P characteristics to understand the potential attack surface [16, 22, 27, 28, 42]. Prior work also examined node-level threats, including Eclipse attacks [25], partial network partitions [9], and vulnerabilities in blockchain node implementations [5]. Similarly, research on node bootstrapping [44] and misconfigured ports [13] highlights that routine scanning is very prevalent and can reveal open services ripe for exploitation.

Previous work established this attack surface under the proof-of-work (PoW) consensus model [11], where nodes were tightly coupled to the network’s economic layer, as miners had wallets associated with their nodes. Since Ethereum switched to proof-of-stake (PoS) in 2022, nodes are two degrees separated from the economic layer. Applying cloud-based honeypots to the Ethereum network, prior studies have shown evidence that attackers target vulnerable Ethereum nodes [11]. Attackers often locate these targets by extracting public routing information from Ethereum’s Kademlia distributed hash table (DHT). This prior work examines a single vulnerable port, the RPC port associated with the Ethereum client software, to execute malicious commands aimed at stealing Ether from the target node’s wallet [10, 20].

Our work complements and extends these studies by having honeypots listen on all available ports, not just the standard Ethereum RPC ports. This broader approach enables us to explore threats arising from server misconfigurations or vulnerabilities beyond an exposed and insecure RPC port. Considering the shift to PoS consensus, the specific attacks identified in prior studies are now less practical. Therefore, it’s increasingly important to understand the security of the servers running the nodes themselves, rather than focusing solely on the execution client’s RPC port. To address these emerging security concerns, our experimental design tests the hypothesis that P2P overlay network nodes are lucrative targets for attackers seeking persistent hosts.

3 Experimental Setup

To test the hypothesis that P2P network nodes are being targeted, we deployed several mainnet Ethereum nodes and a diverse set of honeypots on globally distributed VPs for a two-month period. The resulting data is part of the *EtherBee* dataset [35].

3.1 Global Vantage Points

We deployed 10 geographically diverse VPs across five AWS regions:

- **NA (us-east-1)**: Virginia, US
- **EU (eu-central-1)**: Frankfurt, DE
- **ME (me-central-1)**: United Arab Emirates, AE
- **AP (ap-east-1)**: Hong Kong, HK
- **SA (sa-east-1)**: São Paulo, BR

The VP servers were assigned to a control group and an experimental group, with a pair of each deployed to five distinct geographic regions. On each, we ran a honeypot platform, T-Pot [40], with a diverse set of honeypot containers with full port coverage at varying levels of interactivity, as well as the Arkime [4] network traffic monitoring service. The experimental group additionally ran Ethereum beacon nodes on mainnet, allowing for a direct comparison of honeypot data between the two groups. The EC2 instances are Ubuntu base images configured as in Table 1; the AMIs (Amazon Machine Images) are made available as artifacts of the dataset collection.

Table 1: Comparison of Control and Experimental group server characteristics.

Characteristic	Control	Experimental (Node)
Disk Size (GB)	128	2500
Instance Type	m7/6i.large	r7/6i.xlarge
Software	T-Pot, Arkime	LH, NM, T-Pot, Arkime

3.2 Ethereum Node Deployment

An Ethereum beacon node combines an Ethereum consensus client and execution client that communicate with other nodes to form the network [15]. Nodes are responsible for enforcing the protocol, serving block data, and acting as entry points for validator clients to propose and attest to new blocks. The consensus client, in our case Lighthouse (LH) [37], communicates with peers over `discv5` and `libp2p` to determine the current canonical head of the chain. The execution client, here Nethermind (NM) [29], uses `discv4` and `devp2p` to share transactions and compute the current global state of the Ethereum Virtual Machine (EVM).

When deployed together, these two clients form a beacon node. We deployed our LH and NM clients with default settings, except for the advertised P2P ports being 64000 (consensus) and 64333 (execution) to avoid conflicts with the honeypots. In our setup, the validator client is simulated by the Lighthouse Attestation Simulator [33], a built-in function that produces simulated attestations each epoch to measure the performance of the node as if it were a validator on the mainnet. This data is key to understanding node performance and health without funding actual validators (at the time of writing this paper, requires approximately USD \$60k worth of ETH per validator). Each client exposes standardized metrics in order to monitor the client’s behavior in detail.

3.3 Honeypot Deployment

The T-Pot platform manages multiple containerized, open-source honeypots at varying interaction levels. We deployed honeypots covering ports 1-63999 with the firewall open to all inbound traffic

on these ports, while ports 64000 and above were reserved for management and the legitimate Ethereum node P2P ports. A full list of the honeypots used, along with their targeted services and ports can be found on the T-Pot Github repository [40] and in Appendix Table 4. Ensuring a broad coverage of services and ports is key to establishing robust baselines for later analysis. For this paper, we define an “attempted intrusion or exploit” or “attack” as an interaction that goes beyond simple network scans, such as repeating brute-force attempts on SSH credentials or querying sensitive URIs. Routine inbound scans or single-connection probes are categorized as “reconnaissance.” [23].

4 Findings

After the 2-month collection period, we recorded **130.9 million reconnaissance and exploitation attempts from 12.5 million unique source IPs** on the honeypots across the 10 VPs. From the 5 beacon nodes, we collected **45.4 million P2P TCP sessions** to use in our analysis. Figure 1 illustrates the global peer coverage of our nodes using the P2P session data.

Peer Tenure. We first empirically established the tenure distribution for peers connected to the beacon nodes throughout the collection period. To quantify tenure, we grouped the P2P sessions of each peer by unique IP address and calculated the total number of distinct days each peer maintained a connection to a beacon node. By counting non-consecutive connection days, we captured the persistence of each peer without penalizing short gaps in connectivity that naturally occur in P2P networks. We generated a distribution of peer tenures of the top 1000 peers by data volume for each region, shown in Figure 2, allowing us to confirm patterns across different geographical contexts. With a consistent average tenure ($\mu = 26.46$, $n = 5000$) across regions, our findings indicate a significant proportion of peers maintained stable, recurrent connections to the beacon nodes. This stability suggests that P2P nodes provide a persistent and reliable environment for adversaries seeking long-term access.

4.1 Ethereum Nodes are Targeted

To determine whether our Ethereum nodes receive more attempted intrusions than our control nodes, we needed to account for both temporal and geographic variations that could influence the number of such attempted intrusions. We employed a generalized linear mixed model (GLMM) with a log-transformed response variable which allowed us to include both fixed effects (experimental vs. control groups) and random effects (geographic regions). The log transformation was applied to stabilize the variance and address the skewness inherent in count data.

The results of the GLMM, see Table 2, indicate that the experimental group received significantly more exploitation attempts than the control group after accounting for geographic variation. The experimental group had a significantly higher log-transformed “attack” count ($\hat{\beta}_1 = 1.111$, $SE = 0.008$, $z = 147.289$, $p < 0.001$). Model diagnostics indicated successful convergence, with the random effects capturing variation attributable to geographic differences. The variance estimate for the group effect ($\sigma^2 = 0.050$, $SE = 0.030$) indicates that the effect size of regional variance is small. This suggests that much of the observed variability can be attributed to the

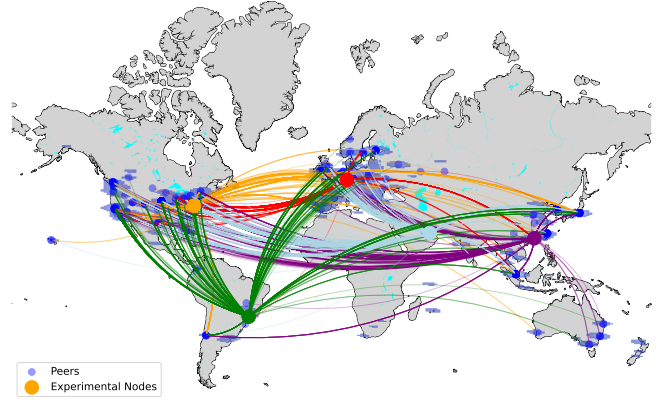


Figure 1: Node locations are derived via IP geolocation cross-referenced with MaxMind and self-reported ENR data; lines in the figure represent top 10% of peers by total data volume.

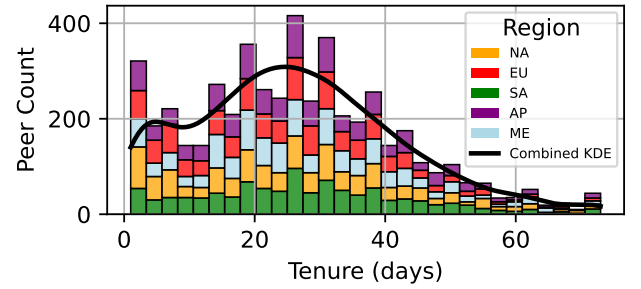


Figure 2: Peer tenure distribution, measured as non-consecutive connection days for peers grouped by unique IPs across regions. The distribution highlights stable, recurrent connections, suggesting P2P nodes provide a platform for long-term adversarial access.

fixed effect of the presence of the P2P node rather than random geographic differences.

These findings support our initial hypothesis that Ethereum nodes are specifically targeted with a higher volume of attacks compared to control nodes, even after accounting for both time and geographic differences. Now, we will explore the nature of those attacks by what ports and services are targeted most commonly.

4.2 Specific Port and Process Targeting

Port Targeting. Figure 3 confirms prior work that Ethereum RPC nodes are targeted, but contributes new findings that the ports are still targeted even under PoS consensus. Furthermore, we show new evidence that the P2P discovery ports are subject to reconnaissance, indicating that adversaries may be seeking vulnerable software to exploit in the node’s networking stack, such as unpatched node software versions or other networking libraries.

Credential Harvesting Requests. URI request data captures attempts by external actors to access specific endpoints or files on the server via HTTP requests, revealing potential attack vectors. By

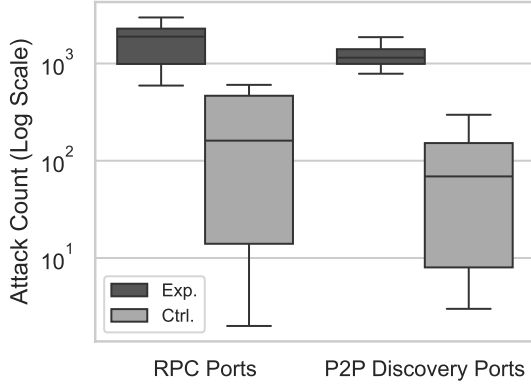


Figure 3: Boxplot of the log-scale distribution of inbound scans, highlighting increased reconnaissance against RPC and discovery ports of experimental group; whiskers for min/max across the 5 vantage points.

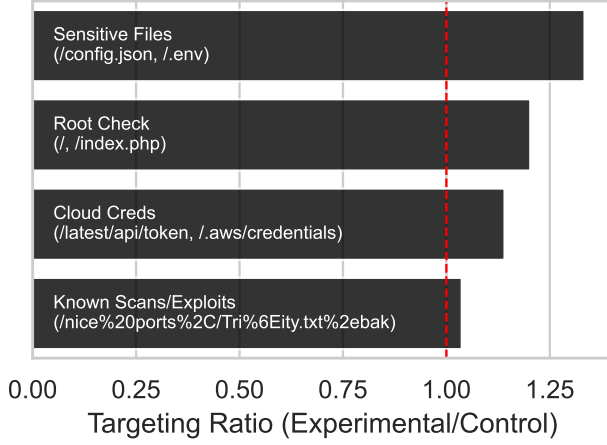


Figure 4: Significant categories of HTTP URI requests with request examples.

comparing the URI requests made to the control and experimental groups, across different regions, we can detect patterns of targeted behavior. Our analysis shows statistically significant differences between these two groups across all five regions, with chi-square results yielding p -values near zero. This indicates a deliberate and non-random targeting of Ethereum nodes, independent of regional deployment.

Our analysis reveals a concerning focus on accessing sensitive files and cloud metadata endpoints. Specifically, requests targeting files such as `.env` files and cloud-related URIs like `/latest/metadata/iam/security-credentials/` and `/.aws/credentials` suggest organized attempts to exploit misconfigurations of the infrastructure. These URIs often store environment variables, credentials, and API keys, potentially allowing attackers to obtain unauthorized access to cloud infrastructure or compromise the nodes. We group similar URI requests into categories and present significant differences in targeting ratio in Figure 4, where a targeting ratio

of 1 would indicate equal attempts between the experimental and control groups.

The recurring patterns in these requests point to a systematic attack strategy aimed at exploiting known vulnerabilities in Ethereum node deployments. The high frequency of requests for `.env` files and cloud-specific endpoints underscores the attackers' focus on harvesting credentials or API keys to gain control over nodes or manipulate their behavior. These findings emphasize the need for robust security practices, such as enforcing least-privilege IAM roles, storing secrets in a managed vault, blocking metadata endpoints at the reverse proxy, and integrating automated secret-scanning and configuration-linting into the CI/CD pipeline. It is important to note that node operators may run their services behind a reverse proxy, which leads to the use of these HTTP servers.

Concentrated Authentication Attempts. We computed Gini coefficients to quantify how SSH login attempts were distributed across username/password pairs. The control group's coefficient (0.25) indicated a relatively even spread of credentials, whereas the experimental group's higher value (0.43) showed that attackers focused on fewer combinations. The Lorenz curve of Figure 5 offers a visual representation of the inequality in distribution by plotting the cumulative share of user/pass combinations against the cumulative share of login attempts. This suggests more sophisticated or targeted strategies against Ethereum nodes. Although average password length was similar in both groups, the experimental set had lower entropy ($p < .001$) and lower reuse frequency, highlighting a distinct pattern of credential-harvesting attempts.

Although general scanning of ports 80/443/22 is a routine expectation for publicly routable hosts on the Internet, our data reveal that Ethereum nodes receive higher-intensity scans focused on `.env`, `/.aws/credentials`, and node-specific endpoints. This aligns with prior findings that always-on, publicly advertised nodes are prime targets.

4.3 Distribution of Reconnaissance Sources

To determine whether inbound reconnaissance originated from the same networks hosting nodes, we analyzed the distribution of sources and node locations at the Autonomous System (AS) level. Recognizing that background activity can vary significantly across different regions and network types, with cloud IPs receiving orders of magnitude more traffic [31], we adjusted for regional effects using control data. By subtracting the attack counts observed by control nodes from those of Ethereum nodes for each AS, we obtained adjusted attack counts that more accurately reflect attacks specifically targeting Ethereum nodes.

Figure 6 shows a dual-density plot overlaying the adjusted attack densities and Ethereum node location densities across IPv4 space, using Gaussian smoothing ($\sigma = 3.5$) to aid visualization. The plot reveals limited overlap between regions of high inbound sources and high concentration of Ethereum nodes, suggesting that reconnaissance activities are not predominantly originating from networks where Ethereum nodes are hosted. Quantitative analysis supports this: the Jaccard Index, measuring the overlap between ASNs in the top 20% for both adjusted event counts and node counts, is low at 0.274. This indicates that only about 27% of ASNs are common between high reconnaissance activity and high node presence.

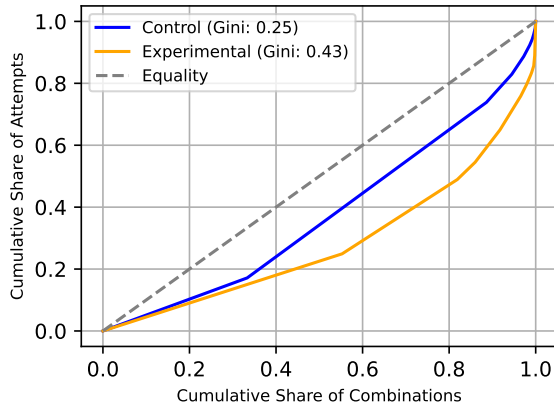


Figure 5: Lorenz curve showing the greater concentration of user-name:password login attempts against nodes.

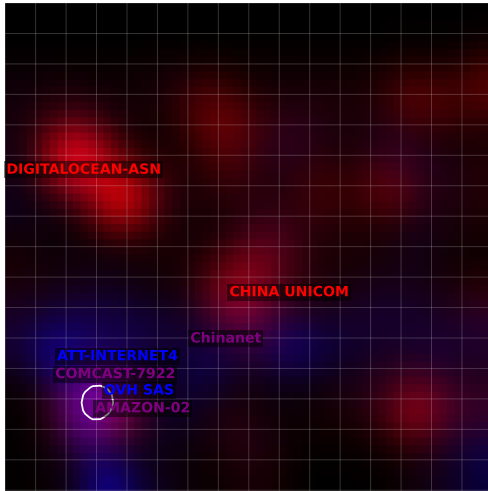


Figure 6: Dual-density plot overlaying adjusted attack densities (red) and Ethereum node densities (blue) in IPv4 space. Areas of overlap appear purple, with an 80% density overlap contour line in white, indicates IP space where attacks originate and nodes co-occur.

Additionally, the substantial Kullback-Leibler Divergence of 6.439 and the large Earth Mover’s Distance of 1,265.287 between the distributions of adjusted attack counts and node counts per ASN indicate significant divergence. These findings support our hypothesis that the reconnaissance and exploitation attempts primarily originate from different networks than those hosting Ethereum nodes, highlighting targeted attack patterns against the networks where Ethereum nodes are concentrated.

4.4 Scanning Data

To understand the exposure of all Ethereum nodes to the activities we observed on our honeypots, we conducted our own reconnaissance and empirical analysis from an adversary’s perspective. Using

a web scraper on a popular node explorer service that regularly crawls the mainnet, we identified 6,185 active beacon nodes recently observed online. We then conducted TCP port scans against these nodes, targeting both the top 100 ports from our honeypot data and default RPC/P2P ports.

Ports are labeled as either closed, filtered, or open. Closed ports are accessible, but have no listening application; filtered ports have indeterminate status, likely due to firewall rules or packet filtering; open ports are actively accepting connections. Even if a host currently has no exploitable vulnerabilities on its open ports, their exposure to the Internet makes it vulnerable to future exploits. The risk extends from the P2P network to the broader Internet, as compromised P2P nodes could serve as unwitting platforms for future attacks.

25.68% (1588/6184) of nodes scanned had at least one open port beyond their P2P ports. 7.45% had a single port open, while 18.22% had multiple open ports. The full distribution of open ports detected is in Figure 7.

Table 3 presents the most prevalent individual ports and port combinations observed. SSH (port 22) was the most common, present on 13.63% of nodes, followed closely by HTTPS (443) and HTTP (80) at approximately 12.7% each. Notably, 2.99% of nodes exposed the Ethereum RPC port (8545), and 2.36% had port 3000 accessible, which is default port for the open-source monitoring tool Grafana.

The most frequent port combinations suggest deliberate administrative configurations. The most common combination was HTTP/HTTPS (80, 443) at 4.66%, followed by a HTTP/HTTPS with SSH access (22, 80, 443) at 1.57%. Of particular concern, 1.12% of nodes exposed both SSH and RPC ports (22, 8545). A significant finding was that 213 nodes (3.44% of the total sample) had default RPC ports (8545 or 8546) exposed to the Internet.

To validate our findings and exclude potential honeypots, we examined port distribution patterns. The absence of large ranges of open ports suggests catch-all honeypots were not present in our dataset. Furthermore, our P2P session data confirmed that we established productive peer connections with all nodes showing open ports, affirming their status as legitimate network participants rather than passive monitoring systems.

The concentration of open ports around common administrative services (SSH, HTTP/S) and monitoring tools (Grafana) indicates that node operators are primarily exposing ports for management purposes. However, the presence of exposed RPC ports and various service combinations suggest some operators may prioritize convenience over security, potentially creating vulnerabilities exploitable by future attacks.

5 Discussion

In this short paper, our measurements suggest that scans targeting Ethereum nodes are often opportunistic, aiming to exploit stable, always-on hosts rather than steal on-node funds. Still, a compromised beacon node can cause localized denial-of-service or serve as a platform to carry out further malicious actions. Similar threats extend to other critical infrastructure networks that feature well-known ports and continuous uptime.

We address a gap in cross-network threat intelligence, studying how P2P nodes can be hijacked for broader attacks. Using

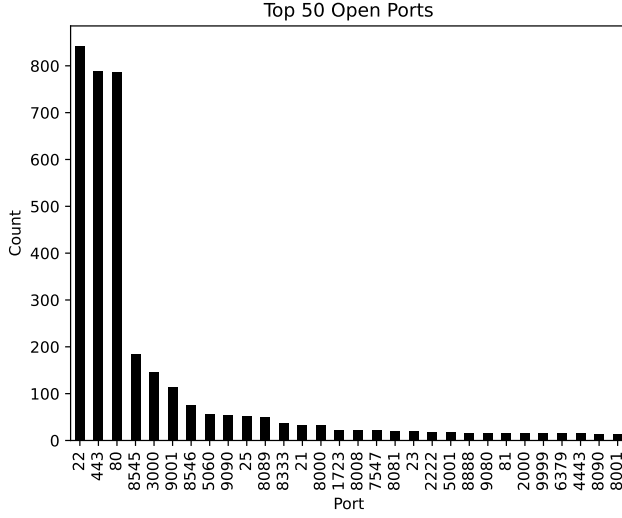


Figure 7: The top 50 open ports by active scanning of online Ethereum beacon nodes excluding P2P ports.

Table 2: Generalized Linear Mixed Model results.

	Coeff.	SE	z	P> z
Fixed Effects				
Intercept	2.804	0.100	28.117	<0.001
Group (Experimental)	1.111	0.008	147.289	<0.001
Random Effects				
Group Variance (Region)	0.050	0.030		
Model Fit Statistics				
Obs	100,000			
Regions	5			

Table 3: Top 5 most common open ports and port combinations.

Port	%	Count	Port Combination	%	Count
22	13.63%	843	(80, 443)	4.66%	288
443	12.75%	788	(22, 80, 443)	1.57%	97
80	12.73%	787	(22, 8545)	1.12%	69
8545	2.99%	185	(22, 3000)	0.87%	54
3000	2.36%	146	(22, 80)	0.65%	40

Ethereum as a high-value, large-scale target provided strong evidence of heightened reconnaissance and exploitation attempts. Our approach can be replicated across other large P2P networks to systematically capture these behaviors.

Implications for Decentralized Overlay Networks. Public P2P overlays, including both native blockchain networks and the critical infrastructure that they secure, rely on transparency to function. However, inadvertently, this transparency simplifies an attacker’s kill chain. Recurrent, standardized threat assessments would help identify and mitigate novel vulnerabilities as these networks grow. The same openness that fosters permissionless participation also creates expanded opportunities for exploitation.

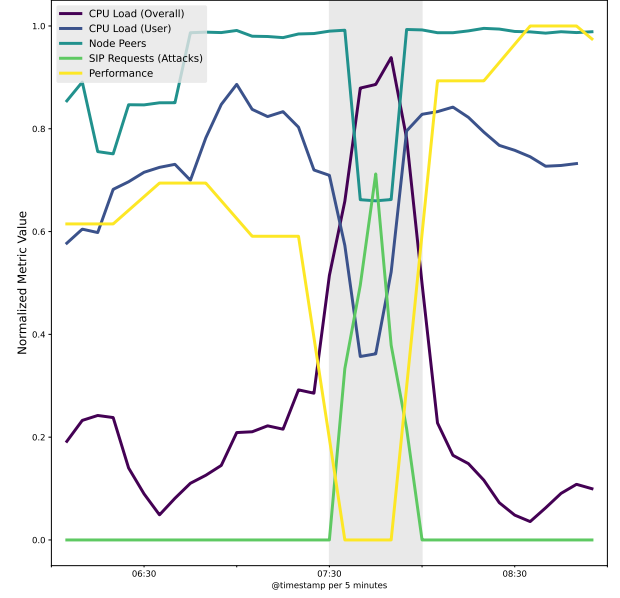


Figure 8: Example of an active attack against the Sentrypeer honeypot on an experimental server, illustrating that during the attack, peer count dropped as the attack rate increased. Overall CPU load increases, decreasing the CPU cycles allocated to user processes, dropping attestation performance. The node recovers post attack. Shown in 5 min ticks with moving average.

Attack-Induced Impacts. Exposed ports can hamper node performance. We observed reflection-based and aggressive scans driving up CPU usage and forcing de-allocation of resources from beacon node tasks, effectively causing denial-of-service. Multiple compromises could form a beachhead for pivoting or partial eclipses, akin to IoT botnets but with more robust resources. As visualized in Figure 8, during an attack recorded by the Sentrypeer [36] honeypot, the affected node lost 20% of its peers and was unable to perform Ethereum beacon node functions, causing a node-level denial-of-service.

In practice, firewalling SSH, restricting RPC endpoints, and placing nodes behind DDoS protection can reduce the attack surface. Future solutions might include on-chain or overlay-based verification of node software versions before peering [26].

6 Conclusion

We presented initial evidence of active reconnaissance and exploitation attempts on Ethereum P2P network nodes by deploying honeypots alongside actual Ethereum nodes across globally distributed VPs. We discovered that these nodes not only faced an increased number of attacks but also specific types targeting particular ports and services. By performing port scans on reachable nodes, we provided evidence that the threat landscape observed on Ethereum nodes is applicable to the broader P2P communities and critical infrastructure that is building dependencies on blockchain networks for security. Our findings offer insights into potential mitigation strategies aimed at enhancing the security of critical blockchain node infrastructure.

References

- [1] Eric Alata, Vincent Nicomette, Mohamed Kaâniche, Marc Dacier, and Matthieu Herrb. 2006. Lessons learned from the deployment of a high-interaction honeypot. In *Proc. EDCC '06*. IEEE, 39–46.
- [2] Hamad Almohannadi, Irfan Awan, Jassim Al Hamar, Andrea Cullen, Jules Pagan Disso, and Lorna Armitage. 2018. Cyber Threat Intelligence from Honeypot Data Using Elasticsearch. In *Proc. IEEE AINA '18*. 900–906.
- [3] Manos Antonakakis, Tim April, Michael Bailey, Matt Bernhard, Elie Bursztein, Jaime Cochran, Zakir Durumeric, J Alex Halderman, Luca Invernizzi, Michalis Kallitsis, et al. 2017. Understanding the mirai botnet. In *Proc. USENIX Security '17*. 1093–1110.
- [4] Arkime Project. 2024. Arkime: Network Analysis & Packet Capture. <https://github.com/arkime/arkime>. Version 5.2.0.
- [5] Jean-Philippe Aumasson, Denis Kolegov, and Evangelia Stathopoulou. 2021. Security Review of Ethereum Beacon Clients. [arXiv:2109.11677](https://arxiv.org/abs/2109.11677) [cs.CR] <https://arxiv.org/abs/2109.11677>
- [6] Paul Barford, Yan Chen, Anup Goyal, Zhichun Li, Vern Paxson, and Vinod Yegneswaran. 2010. Employing honeynets for network situational awareness. *Cyber Situati. Awareness (CSA)* (2010), 71–102.
- [7] Shaileshh Bojja Venkatakrishnan, Giulia Fanti, and Pramod Viswanath. 2017. Dandelion: Redesigning the bitcoin network for anonymity. *POMACS* 1, 1 (2017), 1–34.
- [8] Ismail Butun, Patrik Österberg, and Houbing Song. 2019. Security of the Internet of Things: Vulnerabilities, attacks, and countermeasures. *IEEE Commun. Surv. Tutor.* 22, 1 (2019), 616–644.
- [9] Huashan Chen, Marcus Pendleton, Laurent Njilla, and Shouhuai Xu. 2020. A Survey on Ethereum Systems Security: Vulnerabilities, Attacks, and Defenses. *ACM Comput. Surv.* 53, 3, Article 67 (June 2020), 43 pages. doi:10.1145/3391195
- [10] Zhen Cheng, Xinrui Hou, Runhui Li, Yajin Zhou, Xiapu Luo, Jinku Li, and Kui Ren. 2019. Towards a first step to understand the cryptocurrency stealing attack on ethereum. In *Proc. RAID '19*. 47–60.
- [11] Kota Chin and Kazumasa Omote. 2021. Analysis of attack activities for honeypots installation in ethereum network. In *Proc. IEEE Blockchain '21*. IEEE, 440–447.
- [12] Cowrie. 2023. SSH/Telnet honeypot. <https://github.com/cowrie/cowrie>.
- [13] Jean-Philippe Eisenbarth, Thibault Cholez, and Olivier Perrin. 2022. Ethereum's peer-to-peer network monitoring and sybil attack prevention. *Journal of Network and Systems Management* 30, 4 (2022), 65.
- [14] Claude Fachkha and Mourad Debbabi. 2015. Darknet as a source of cyber intelligence: Survey, taxonomy, and characterization. *IEEE Commun. Surv. Tutor.* 18, 2 (2015), 1197–1227.
- [15] Ethereum Foundation. 2024. Nodes and Clients. <https://ethereum.org/en/developers/docs/nodes-and-clients/>.
- [16] Yue Gao, Jinqiao Shi, Xuebin Wang, Qingfeng Tan, Can Zhao, and Zelin Yin. 2019. Topology measurement and analysis on ethereum p2p network. In *Proc. IEEE ISCC '19*. IEEE, 1–7.
- [17] Glutton. 2023. Generic Low Interaction Honeypot. <https://github.com/mushorg/glutton>.
- [18] Dominic Grandjean, Lioba Heimbach, and Roger Wattenhofer. 2023. Ethereum Proof-of-Stake Consensus Layer: Participation and Decentralization. [arXiv:2306.10777](https://arxiv.org/abs/2306.10777) [cs.DC] <https://arxiv.org/abs/2306.10777>
- [19] Hamed Haddadpajouh, Ali Dehghantanha, Reza M Parizi, Mohammed Aledhari, and Hadis Karimipour. 2021. A survey on internet of things security: Requirements, challenges, and solutions. *Internet of Things* 14 (2021), 100129.
- [20] Kazuki Hara, Teppei Sato, Mitsuyoshi Imamura, and Kazumasa Omote. 2020. Profiling of Malicious Users Targeting Ethereum's RPC Port Using Simple Honeypots. In *Proc. IEEE Blockchain '20*. 1–8.
- [21] Honeytrap. 2021. Advanced honeypot framework. <https://github.com/honeytrap/honeytrap>.
- [22] Xiaoyan Hu, Cheng Zhu, Zhongqi Tong, Wenjie Gao, Guang Cheng, Ruidong Li, Hua Wu, and Jian Gong. 2022. Identifying Ethereum traffic based on an active node library and DEVp2p features. *FGCS* 132 (2022), 162–177.
- [23] Niclas Ilg, Paul Duplys, Dominik Sisejkovic, and Michael Menth. 2023. Survey of contemporary open-source honeypots, frameworks, and tools. *Journal of Network and Computer Applications* (2023), 103737.
- [24] Mitsuyoshi Imamura and Kazumasa Omote. 2019. Network deployments of bitcoin peers and malicious nodes based on darknet sensor. In *Proc. WISA'18*. Springer, 117–128.
- [25] Arkan Hammoodi Hasan Kabla, Mohammed Anbar, Selvakumar Manickam, Taief Alaa Al-Amiedy, Peterson Bernabe Cruspe, Ahmed K. Al-Ani, and Shankar Karuppayah. 2022. Applicability of Intrusion Detection System on Ethereum Attacks: A Comprehensive Review. *IEEE Access* 10 (2022), 71632–71655. doi:10.1109/ACCESS.2022.3188637
- [26] George Kadianakis, Mary Maller, Andrija Novakovic, and Suphanat Chunhapanya. 2023. Proof of Validator: A Simple Anonymous Credential Scheme for Ethereum's DHT. <https://ethresear.ch/t/proof-of-validator-a-simple-anonymous-credential-scheme-for-ethereums-dht/16454>.
- [27] Lucianna Kiffer, Asad Salman, Dave Levin, Alan Mislove, and Cristina Nita-Rotaru. 2021. Under the hood of the ethereum gossip protocol. In *Proc. FC '21*. Springer, 437–456.
- [28] Seoung Kyun Kim, Zane Ma, Siddharth Murali, Joshua Mason, Andrew Miller, and Michael Bailey. 2018. Measuring ethereum network peers. In *Proc. IMC '18*. 91–104.
- [29] Nethermind Project. 2024. Nethermind: The Ethereum Execution Client. <https://github.com/NethermindEth/nethermind>. Version 1.27.1+d2d5d4ae.
- [30] Till Neudecker, Philipp Andelfinger, and Hannes Hartenstein. 2015. A simulation model for analysis of attacks on the bitcoin peer-to-peer network. In *Proc. IFIP/IEEE IM '15*. IEEE, 1327–1332.
- [31] Eric Pauley, Paul Barford, and Patrick McDaniel. 2023. {DScope}: A {Cloud-Native} Internet Telescope. In *Proc. USENIX Security '23*. 5989–6006.
- [32] Eric Pauley and Patrick McDaniel. 2023. Understanding the ethical frameworks of internet measurement studies. In *Proc. IWEC '23*.
- [33] Sigma Prime. 2024. Lighthouse - Attestation Simulator Metrics. <https://lighthouse-book.sigmaprime.io/validator-monitoring.html#attestation-simulator-metrics>.
- [34] SM Zia Ur Rashid, Ashfaqul Haq, Sayed Tanimun Hasan, Md Hasan Furhad, Mohiuddin Ahmed, and Abu Barkat Ullah. 2024. Faking smart industry: exploring cyber-threat landscape deploying cloud-based honeypot. *Wireless Networks* 30, 5 (2024), 4527–4541.
- [35] Scott Seidenberger and Anindya Maiti. 2025. EtherBee: A Global Dataset of Ethereum Node Performance Measurements Coupled with Honeypot Interactions and Full Network Sessions. [arXiv:2505.18290](https://arxiv.org/abs/2505.18290) [cs.NI] <https://arxiv.org/abs/2505.18290>
- [36] SentryPeer. 2024. SentryPeer. <https://sentrypeer.org/>.
- [37] Sigma Prime. 2024. Lighthouse: An Ethereum 2.0 Client. <https://github.com/sigp/lighthouse>. Version v5.2.1-9e12c21.
- [38] Kiran Deep Singh. 2021. Securing of Cloud Infrastructure using Enterprise Honeypot. In *Proc. ICAC3N '21*. 1388–1393.
- [39] Francesca Soro, Thomas Favale, Danilo Giordano, Idilio Drago, Tommaso Rescio, Marco Mellia, Zied Ben Houidi, and Dario Rossi. 2023. Enlightening the darknets: Augmenting darknet visibility with active probes. *IEEE TNSM* 20, 4 (2023), 5012–5025.
- [40] T-Pot Project. 2024. T-Pot: The All In One Honeypot Platform. <https://github.com/telekom-security/tpotce>. Version 24.04.0.
- [41] Jay Thom, Yash Shah, and Shamik Sengupta. 2021. Correlation of Cyber Threat Intelligence Data Across Global Honeypots. In *Proc. IEEE CCWC '21*. 0766–0772.
- [42] Taotao Wang, Chonghe Zhao, Qing Yang, Shengli Zhang, and Soung Chang Liew. 2021. Ethna: Analyzing the underlying peer-to-peer network of ethereum blockchain. *IEEE TNSM* 8, 3 (2021), 2131–2146.
- [43] Yixin Wu, Cheng Huang, Xing Zhang, and Hongyi Zhou. 2020. GroupTracer: Automatic attacker TTP profile extraction and group cluster in Internet of things. *Secur. Commun. Netw.* 2020, 1 (2020), 8842539.
- [44] Karl Wüst and Arthur Gervais. 2016. *Ethereum eclipse attacks*. Technical Report. ETH Zurich.
- [45] Yichi Zhang, Chunhua Yang, Keke Huang, and Yonggang Li. 2022. Intrusion detection of industrial internet-of-things based on reconstructed graph neural networks. *IEEE TNSM* 10, 5 (2022), 2894–2905.

A Acknowledgments

The funding for this study was provided by the Ethereum Foundation under grants FY24-1546 and FY24-1547. We thank them for their continued support and their commitment to solving human coordination problems with fair and open protocols.

B Ethics

To mitigate any potential privacy issues in our collection and sharing of this threat intelligence, we followed best practices for measurement studies [32]. We refrained from disclosing specific IP addresses and implemented anonymization when presenting attack patterns and performed data encryption in both transit and at rest. Additionally, we limited our scanning methods to publicly accessible information and employed non-intrusive, narrowly-scoped port scans to ensure minimal disruption to any given node. Future studies should continue to consider these ethical dimensions, especially as the data collected could inadvertently reveal operational practices of node operators that could put them at greater risk.

C Recommendations for Operators and Protocol Researchers

The detailed insight into network-level distribution of reconnaissance and exploitation attempt sources presented here can inform node operators in deploying proactive measures. Specifically, operators could throttle or block IP spaces associated with high scan rates or inbound activity but low peer density, mitigating potential risks without compromising access to legitimate peers. Our research suggests that by leveraging this data, node operators can implement IP filtering measures and other defensive tactics tailored to known threat vectors.

Another potential security avenue is the use of private peering arrangements or mesh VPNs. For private peering arrangements, node operators could make sure that the core of their peer list is other known actors that have met certain reputation-based criteria¹. An even more secure network security approach would be to establish a large, decentralized, mesh VPN for peer connectivity, where the nodes can granularly control their exposure to the public Internet. This could serve as a necessary mechanism to protect the health of a network that has significant economic value and public dependence.

Our findings suggest that implementing certain network-layer security enhancements does not necessarily have to infringe upon a network’s decentralized or permissionless properties. Selective security measures may help secure critical infrastructure within these networks while preserving their openness and accessibility for new node operators.

D Honeynet Setup Details

Table 4: Honeypot containers with service and port coverage.

Honeypot	Interaction	Target Services	Ports
Adbhoney	Medium	Android Debug Bridge (ADB)	5555/TCP
CiscoASA	Medium	Cisco ASA VPN	5000/UDP, 8443/TCP
Citrixhoneypot	Medium	Citrix Gateway	443/TCP
Conpot	Low	Guardian AST, IEC 104, IPMI, Kamstrup 382	10001/TCP, 2404/TCP, 161/UDP, 623/UDP
Cowrie	High	SSH, Telnet	22-23/TCP
Ddospot	Low	DDoS-related services	19/UDP, 53/UDP, 123/UDP
Dicompot	Low	DICOM Medical Imaging	11112/TCP
Dionaea	High	FTP, SMB, HTTP, MySQL, and more	20-21/TCP, 445/TCP, 1433/TCP, 3306/TCP
Elasticpot	Low	Elasticsearch	9200/TCP
Heralding	Medium	IMAP, SMTP, PostgreSQL	110/TCP, 5432/TCP, 993/TCP
Honeytrap	Medium	Various protocols	All other ports not covered
IPPhoney	Low	Internet Printing Protocol (IPP)	631/TCP
Mailoney	Medium	SMTP, Mail Relay	25/TCP, 587/TCP
Medpot	Low	Medical Device Communication (HL7)	2575/TCP
Redishoneypot	Low	Redis	6379/TCP
Sentrypeer	Medium	SIP	5060/TCP, 5060/UDP
Snare	Low	HTTP	80/TCP
Spiderfoot	Medium	OSINT, Reconnaissance	8080/TCP
Tanner	Medium	API, PHP	Various
Wordpot	Low	WordPress Vulnerabilities	8080/TCP

¹For an ongoing discussion on such proposed solutions, see [26].