International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2022): 7.942

Mitigating Security Risks of Data Exposure in Unauthenticated IPFS Access with Blockchain Based Controls

Srivatsa Chetlur[1], Depavath Harinath[2]

[1]Department of Computer Science, Ramnath Guljarilal Kedia College of Commerce, Osmania University, Hyderabad, Telangana, India

[2]Department of Computer Science, Ramnath Guljarilal Kedia College of Commerce, Osmania University, Hyderabad, Telangana, India

Abstract: ***Blockchain, a novel distributed database technology, facilitates data sharing among computer network nodes. In tandem, the InterPlanetary File System (IPFS) complements this by storing information digitally in an automated manner. The integration of blockchain and IPFS enhances data resilience, accessibility, and sharing efficiency within decentralized systems. IPFS provides a decentralized and efficient storage layer for blockchain, enabling large data and files to be stored off-chain while maintaining references on the blockchain. This reduces the blockchain's load, enhances data availability and fosters scalability in blockchain applications. In this proposed methodology, the objectives of this paper are to provide the best practices by providing a problem statement respectively. The versatile use of IPFS is explored through three distinct scenarios: authorized IPFS node, unauthorized IPFS node within the internal network and unauthorized external user access. Following the node initialization and activation of the daemon, the IPFS node executes the IPFS swarm command to visualize globally connected swarm peer nodes. Utilizing a targeted application as a case study, employing IPFS as the underlying filesystem, a query operation yields the application's IP address. Through syntactically correct swarm connect commands, the IPFS node adeptly establishes connections with the designated application, enabling seamless testing procedures. The research not only delves into the intricacies of IPFS adoption but also outlines practical steps to ensure secure and efficient utilization. The abstract provides a concise overview of the paper's scope, emphasizing the scenarios, key findings, and the offered best practices.***

Keywords: IPFS, CID, DHT, Decentralization Smart Contract

1. Introduction

While cyber world is growing day by day the Security becomes the foremost and integral part of our day-to-day transactions. Generally, HTTP/HTTPS is used for DTA transfer on the Internet which is based on client-server model and centralized data will be unavailable if the server got disrupted due to any reason that makes concerns for users of a single point of failure. However, now a days all the renowned companies are switching over to safer platforms like blockchain technology, a distributed data base of records in which every transaction or digital event that has been taken place or shared among the users or participating agents. Every transaction here is pre-checked which ensures safety and security of transaction. Crypto currency like BITCOIN is a classic and most popular example for this. Not only financial transactions but also where the data is crucial and incorrigible, people are implementing blockchain technology like Telanagana state government has ensured security of their land records and Transactions in DHARANI PORTAL using the same Technology. This distributed ledger system mainly follows IPFS i.e, Inter Planetary File System for storing and sharing the data with utmost security and safety. What is IPFS: As already explained supra, IPFS stands for Inter planetary filesystem which is a complete peer to peer content based file storage system. This is a Web3 Technology which is decentralized, distributed and immutable. IPFS mainly uses concepts like Distributed Hash Table . Mr.Juan Bennet at Protocol Labs created this IPFS in 2015 and currently there are multiple applications being built using IPFS.IPFS is built in such a way that the data is distributed on different nodes which are part of IPFS network and connected to each other for supply of distributed data sharing in which all the nodes have privilege to share and retrieve the data. The network holds copies of data which gives availability of the data in the entire net work. While the normal centralized data storage uses location based addressing and IPFS is built on content-based addressing. This content is managed by distributed hash table commonly termed as DHT. After upload of every new file in the net work an independent hash is generated. In case this generated Hash is edited, a new hash is again created and identified through their content. The raw data is chunked into 256K size of chunks when the file is added to IPFS. Then each chunk is hashed by SHA-256 has function and has value is created for every chunk.

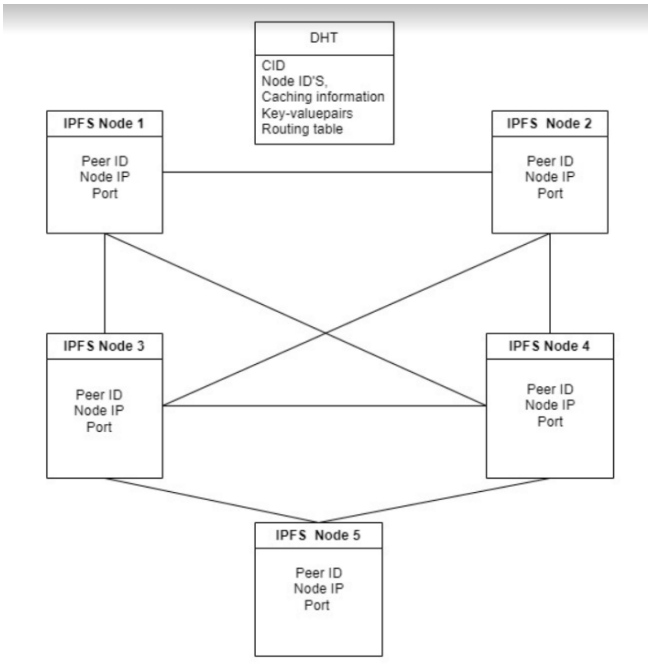
2. Related Work

Interplanetary Filesystem is a distributed, decentralized, immutable and a web3 based technology based file storage protocol. It is used as filesystem which is used to store and share files on a peer-to-peer network basis. [1] Meaning, if someone runs IPFS on their computer and uploads a file to the IPFS network, that file can be viewed and downloaded by anyone else who are also running IPFS.

1) **Decentralization:** IPFS aligns with a unique objective by providing a distributed file system where files are stored across multiple nodes, eliminating the reliance on centralized servers for file hosting and distribution
2) **Content Addressing:** IPFS utilizes content addressing where files are identified by their unique cryptographic hashes based on their content. This enable files to be accessed and retrieved directly from the IPFS network using the appropriate content identifiers.

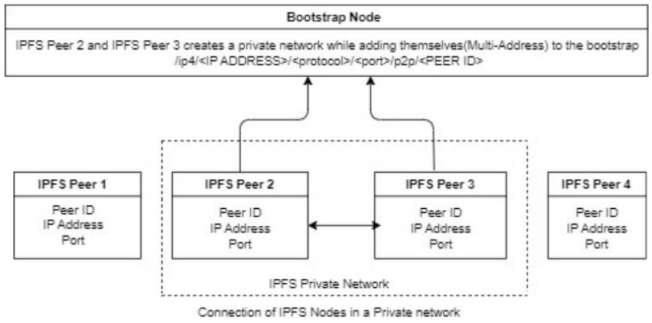
3) **Peer-to-peer Networking:** IPFS operates on peer-to-peer model allowing nodes to communicate directly with other nodes without any intermediaries. Data Integrity and Authenticity: IPFS uses cryptographic hashes to verify the integrity of the files. The content addressing scheme ensures that any modifications to a file will result in a different hash, allowing for secure verification of file integrity.

Interplanetary Filesystem itself is by default a public network where the nodes which participate in the network are connected in a peer-to-peer based technology. The public network in IPFS also enables the nodes to connect share and store files globally. For creating a public IPFS there is an IPFS gateway where, any user can share data through gateway URL where the network will be public and accessible on the internet.



All the nodes in the IPFS network are connected through a public gateway in a peer-to-peer network and can access the DHT directly in order to fetch any file

Private IPFS: In a private IPFS deployment, access to the IPFS network and the content stored within it is restricted to a specific group of authorized users. Private IPFS networks are often used in scenarios where data confidentiality and controlled access are paramount. These networks might be utilized for internal file sharing within an organization, sharing sensitive documents within a closed group, or for applications that require a certain level of privacy and control. Private IPFS networks can be established by configuring firewalls, access controls, and encryption mechanisms to ensure that only authorized individuals or entities can interact with the network.



Connection of IPFS Nodes in a Private network

Quick illustration for the above block diagram in a public network of IPFS, there are 4 IPFS nodes with their Peer ID, Port and IP Address respectively. To form a private network, Peer 2 became the bootstrap node and Peer 3 added their multi-address to Peer 2's config file in the above format, hence making a private network.

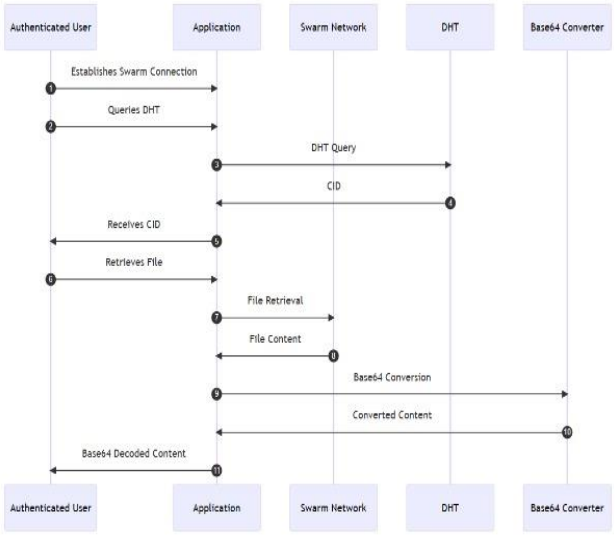
3. Proposed Methodology

Before proceeding with the IPFS methodology, A procedural way/Methodology for accessing an application which uses IPFS as its filesystem.

IPFS is a web3 technology which is decentralized, distributed immutable and a peer-to-peer file system used for sharing and storing of files. A quick scenario mentioned below: In this methodology, there will be three scenarios. One with the authorized IPFS node, second with the unauthorized IPFS

node from the internal network, third with the unauthorized user outside the network respectively. Once the initialization of the IPFS node is done, turn on the daemon. When the node executes the command of IPFS swarm, it will be able to view all the swarm peer nodes connected around the node globally. So, just for the testing purposes we take an application which uses IPFS as its filesystem and query it which results in giving its IP Address. Through a syntactical way of swarm connect, the node will be able to establish a connection with the application successfully. By querying the DHT, the IPFS node will be able to view the contents stored followed by the peer id's who own the files in the application's IPFS swarm network. The CID can be copied and the node will be able to view its contents successfully.

A. 1st scenario



Sequential diagram of an authenticated ipfs node who successfully connects to an application(using ipfs as its filesystem) and accessing data

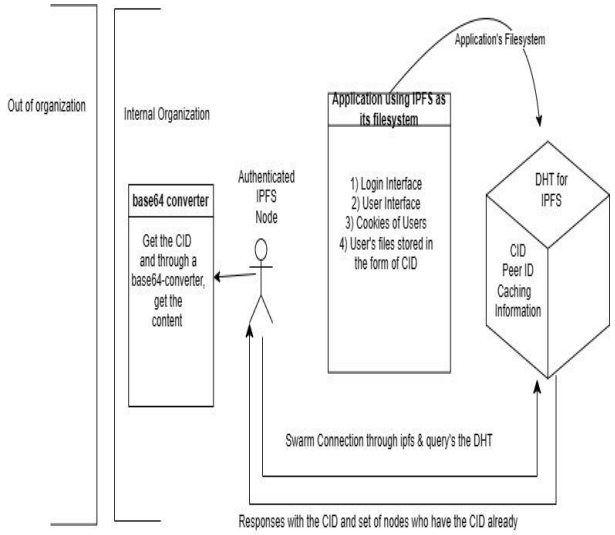
Scenario one explains that how an authenticated user attempts to access the random application which uses IPFS as its file system hosted on the internal server of an organization using the IPFS command line version. The below figure illustrates the sequential flow about how an authenticated user of an organization using IPFS attempts to access the internal files of an application.

Step 1: Turn on the IPFS daemon of the node and take a new terminal.

Step 2: Execute the IPFS id command which shows the multi addresses and identification of the authorized users in the network.

Step 3: When the IPFS node attempts to view the swarm, peers connected around their node by executing the command IPFS swarm peers, the IPFS network replies with the IPFS nodes and their IP addresses with the port they are hosted on followed by their peer ID's.

Step 4: Fetch the peer id of the application hosted on an IPFS network and attempt to connect the IPFS node with the application with the help of swarm connect When the IPFS node attempts to download the CID from the application using IPFS get CID, then the CID gets downloaded to their current directory. By using IPFS cat, the IPFS node will be able to fetch the base64 format of the file. Through a base64 converter the IPFS node will be able to convert the base64 to respective file format (in our case, it was base64 to pdf format).



B. 2nd scenario

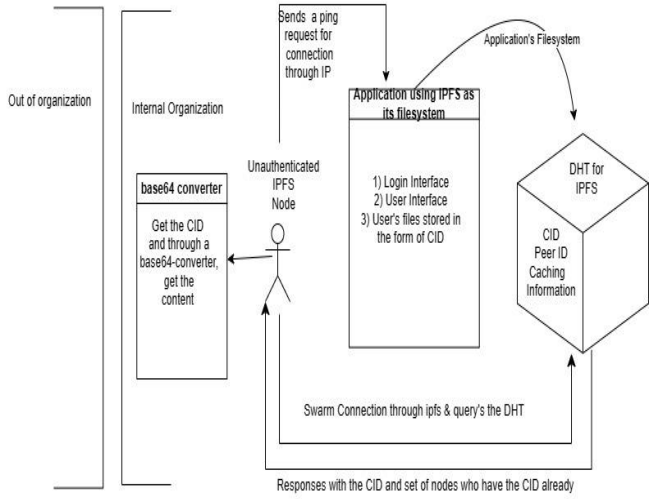
The second scenario explains about an unauthorized IPFS node attempting to access an application which uses IPFS as its file-system hosted in the network.

Step 1: Turn on the daemon and switch to another terminal.

Step 2: While running the command of IPFS swarm peers, the IPFS node will be able to view all the IPFS nodes available in the network and Using the Nmap TCP syn and an aggressive scan, the node could be able to recon more about the application's IPFS.

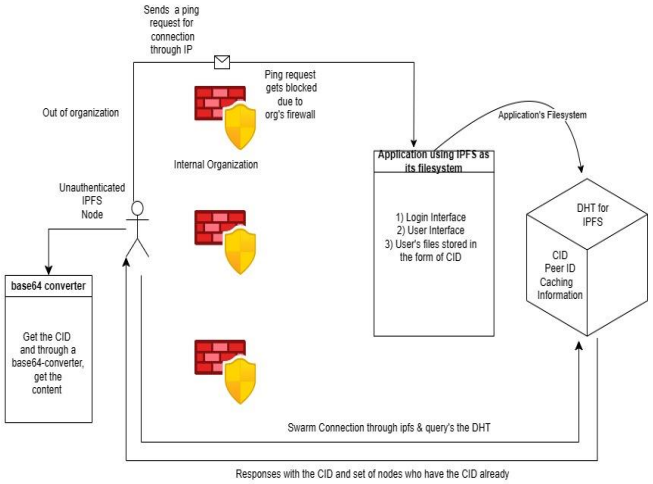
Step 3: The node can then query the DHT with the command IPFS DHT query CID of the file which lists out all the peers who have file in the IPFS network Once the peer id is obtained by doing a trial-and-error method with all the peer id's, the IPFS node will be able to setup a swarm connection with the application's IPFS and when the ipfs node attempts to download the CID from the application using ipfs get CID, then the CID gets downloaded to their current directory.

Step 4: By using IPFS cat, the IPFS node will be able to fetch the base64 format of the file Step 5: With the help of a base64 converter, the IPFS node will be able to convert the format to a readable format (in my case, it is base64 to pdf).



C. 3rd scenario

The third scenario explains about an unauthorized IPFS node attempting to access an application (which uses IPFS as its filesystem) and fetch its files from an outside network.



Step 1: Turning on the IPFS daemon and Execute the IPFS id to view the details of the IPFS node.

Step 2: Execution of the IPFS swarm peers command will let the IPFS node list out all the peers listed in the swarm connection.

Step 3: By performing a social engineering attack, the IPFS node(outside the network) will be able to fetch the CID of a file in the IPFS network(application) and querying the DHT with the command IPFS DHT query CID of the file lists out all the peers who have that file in that swarm network of IPFS and by attempting through trial and error method, the node could attempt to connect to every IPFS id and finally by acquiring the original multiaddress of the application, the IPFS node(outside the network) could attempt to setup a connection with the application through a syntactical way of swarm connection in IPFS. After getting connected with the application with the internal network which uses IPFS as its filesystem, the IPFS node (outside the network) could obtain that CID and use the IPFS command ipfs get CID which downloads that particular file into their present directory.

Step 4: The node might recon for more information about the application hosted on IPFS using an Nmap aggressive scan and after a proper recon for the required data to gather the cid and other information, Using the command ipfs cat CID results in providing the IPFS node (outside the network) with the base64 content of the file.

Step 5: And Using a base64 converter, the node will be able to view all the contents available in that file

D. Mitigations to be followed

To overcome or mitigate this type of security loop holes, some of the best practices can be followed as below:

1) Authorized Node Management: Maintain a list of authorized nodes within the IPFS network. Utilize cryptographic keys and authentication mechanisms for node identification. Regularly review and update the list of authorized nodes as needed.
2) Access Control Lists (ACLs): Implement fine-grained access control by using ACLs to define who can read, write, or modify content. Restrict access to sensitive content to only authorized nodes and users.
3) Secure encryption: Encrypt sensitive content before adding it to IPFS using strong encryption algorithms and make sure to use the well-established libraries and packages
4) Swarm connection validation: Verify the authenticity of the swarm connection requests using cryptographic signatures.

4. Future Scope

As the field of decentralized technologies continues to evolve, the methodologies and lessons outlined in this paper contribute to a broader understanding of IPFS and its practical implementation.

5. Conclusion

In conclusion, the establishment of a robust and well-structured test environment for IPFS implementation proves to be an indispensable component in ensuring the reliability, scalability, and performance of IPFS networks. Through the examination of three distinct scenarios, we have highlighted the significance of adopting best practices when designing and deploying such environments. It is evident that a thoughtfully designed test environment enables researchers, developers, and practitioners to tune their IPFS deployments and develop a more reliable and efficient distributed web ecosystem. In conclusion, the establishment of a comprehensive test environment serves as a cornerstone for unlocking the full potential of IPFS and driving innovation in the decentralized web landscape.

References

[1] https://medium.com/coinsbench/what-is-ipfs-and-quick-demonstrationof-its-setup-838bb44d03d8
[2] IPFS and friends: A qualitative comparison of next generation peer-to-peer data networks, Author: Erik Daniel, Florian Tschorsch, Journal: IEEE Communications Surveys and Tutorials, Volume: 24, Number: 1, Pages: 31-52, Year: 2022, Publisher: IEEE
[3] title: IPFS and friends: A qualitative comparison of next generation peer to-peer data networks author Erik Daniel, Florian schorschjournal IEEE Communications

Surveys Tutorialsvolume 24number 1pages 31-52year 2022publisher IEEE.

[4] title: When blockchain meets distributed file systems: An overview, challenges, and open issues,author=Huang, Huawei and Lin, Jianru and Zheng, Baichuan and Zheng, Zibin and Bian, Jing,journal=IEEE Access,volume=8,pages=50574–50586,year=2020,publisher=IEEE

[5] title:Distributed common data environment using blockchain and Interplanetary File System for secure BIM-based collaborative design, author=Tao, Xingyu and Das, Moumita and Liu, Yuhan and Cheng, Jack CP,journal=Automation in Construction ,volume=130, pages=103851, year=2021, publisher=Elsevier

[6] title: A survey of state-of-the-art on blockchains: Theories, modelings, and tools, author=Huang, Huawei and Kong, Wei and Zhou, Sicong and Zheng, Zibin and Guo, Song, journal=ACM Computing Surveys (CSUR), volume=54, number=2, pages=1–42, year=2021, publisher= ACM New York, NY, USA.

[7] title=IPFS and friends: A qualitative comparison of next generation peer-to-peer data networks,author=Daniel, Erik and Tschorsch, Florian, journal=IEEE Communications Surveys and Tutorials,volume=24, number=1,pages=31–52,year=2022, publisher=IEEE

[8] Depavath Harinath,et.al,"A Review on Security Issues and Attacks in Distributed Systems," Journal of Advances in Information Technology(JAIT), California, USA, Vol. 8, No. 1, pp. 1-9, February, 2017. doi: 10.12720/jait.8.1.1-9

[9] Depavath Harinath, et.al, "Lattice Cryptography- A NTRU Cryptosystem Providing a Quantum Attack Resistant Security System for Cloud Computing", in GIS Science Journal, Volume 11, Issue 4, 2024, ISSN No: 1869-9391 Page No. 159 – 168. DOI:20.18001.GSJ.2024.V11I5.24.41185420

[10] https://medium.com/coinsbench/what-is-ipfs-and-quick- demonstrationof- its-setup-838bb44d03d8

[11] https://medium.com/@csvatsa77/active-directory-a-guide-to-usermanagement- and-network-control-cda0296fc776

[12] Depavath Harinath,et.al"An Interplanetary File System Framework For Invocation and Machine Learning Model Training", Bulletin For Technology And History Journal , Volume 24, Issue 2, 2024, Page No:151-157, DOI:10.37326/bthnlv22.8/1513 and ISSN No : 0391-6715

[13] Depavath Harinath,et.al,"A Review on Security Issues and Attacks in Distributed Systems," Journal of Advances in Information Technology(JAIT), California, USA, Vol. 8, No. 1, pp. 1-9, February, 2017. doi: 10.12720/jait.8.1.1-9

[14] Depavath Harinath, "Enhancing Data Security Using Elliptic Curve Cryptography in Cloud Computing", *International Journal of Science and Research (IJSR)*, https://www.ijsr.net/archive/v5i7/v5i7.php, Volume 5 Issue 7, July 2016, 1884 - 1890, DOI: 10.21275/v5i7.ART2016624

Author Profile

Srivatsa Chetlur, Masters in Computer Application from Osmania University. A researcher with a keen eye for security and programming, delves into the world of blockchain as a security researcher And leverage the coding expertise to identify and patch vulnerabilities in smart contracts and web application which safeguards the integrity of the innovative technology.

Depavath Harinath, Asst. Professor a highly accomplished technology researcher with a Master in Computer Applications (MCA) from the esteemed college of Sreenidhi Institute of Science and Technology (SNIST), India. SNIST boasts UGC autonomy, NAAC 'A+' accreditation, NBA and AICTE approval, and permanently affiliated to Jawaharlal Nehru Technological University (JNTU), Hyderabad and possesses more than twelve years of distinguished teaching experience, having published 24 impactful manuscripts in prestigious international journals with significant citations and extends to The United Kingdom based international patent for an AI Drone for Quantum UAV Farming, demonstrating innovative spirit. Now working as Assistant Professor, Dept. of Computer Science, Ramnath Guljarilal Kedia College of Commerce, Osmania University, Hyderabad, Telangana, India. Research field includes Computer Networks, Network Security, Artificial Intelligence and Machine Learning.