2023 LONDON

4 - 6 October, 2023 / London, United Kingdom

# WEB3 WILL BITE YOU IN THE WEB 2.0 – EXPLORING IPFS THREATS

Morton Swimmer, Cedric Pernet, Roel Reyes, Philippe Z Lin & Vincenzo Ciancaglini

*Trend Micro, Germany / France / Philippines / Canada*

morton_swimmer@trendmicro.com

## ABSTRACT

Web3 applications are composed of a mix storage and trust components, where IPFS is one popular implementation of Web3 storage, although it can operate independently of the trust components. As a fully distributed peer-to-peer system, it is resistant to blocking and takedowns – making it a useful tool for avoiding censorship, but also for phishing and malware threat actors. In this paper we will look at what the IPFS ecosystem is and how it is being abused in phishing attacks. By understanding the technology and the evidence that the problem is growing, defenders will gain an understanding of the complexities of IPFS and be able to avoid being blindsided by this new threat vector.

## 1. INTRODUCTION

In 2015, the InterPlanetary File System (IPFS) was developed as a decentralized peer-to-peer file-sharing system for content delivery and archiving [1]. It can form the basis for websites and applications, making it one of the building blocks of what is referred to as Web3[1]. Participants in the IPFS system around the world are both storing content as well as providing access to that content. In IPFS, a file's content is retrieved by its content address rather than its physical location. All that is needed to access content is the CID, the Content Identifier, which is a data structure mainly comprising a cryptographic hash. Non-participants can access content via gateways. By distributing shared files among all of a networked file system's nodes, as is done by IPFS, participants are guaranteed access to all of the shared files so long as someone still holds a copy. In a centralized network when the server becomes unavailable or is blocked, the hosted content will no longer be available.

IPFS is different from other peer-to-peer systems like BitTorrent in that it allows for shared blocks of data. In IPFS, large files are broken down into smaller chunks that are individually addressed and can be shared across different files if it so happens that the chunks are identical.

One motivation for developing IPFS was to avoid censorship, either by states or by companies restricting access to data. For instance, when Turkey instigated a block on *Wikipedia* [2], a copy of it was made available on IPFS[2]. While this can be considered good, it also means that taking down malicious content is difficult, if not impossible. If phishing content is removed in one node, it may still be available in other nodes. Locating malicious activity in a peer-to-peer network that is also being used for legitimate purposes is a difficult operation. Because of its durable networking, permanent data storage, and absence of regulation, IPFS might be the ideal platform for malicious actors to host and disseminate information in order to further their goals. Currently, it is mainly being misused for hosting phishing content. As the popularity of IPFS increases, criminals will become aware of its other potential benefits.

IPFS is not 'mainstream' yet. The ongoing talk of a 'Cambrian Explosion' of IPFS usage has yet to materialize. IPFS is already being used in legitimate applications, even if uptake is slower than the proponents have predicted. For example, *Audius* [4] is a music streaming service that uses various blockchain technologies for discovery and remuneration. IPFS is used for the actual content distribution and streaming, keeping the hosting costs to a minimum. *WeatherXM* [5] is a crowd-sourced weather application that remunerates contributors using an ERC20[3] token to create a 'weather economy' and uses IPFS for data distribution. This will become a weather 'oracle' that can be used in other blockchain applications, for instance in smart contracts. Despite some lingering issues with the IPFS platform, there are a number of applications in the pipeline and some will undoubtedly emerge soon.

Threats due to IPFS are not completely new. There were reports as early as 2019 on malware using IPFS [7, 8]. While malware on IPFS did not end up being a winning idea, more recently phishing has discovered this platform [9, 10, 11, 12]. As we will see, IPFS phishing took off in 2022 and continues into this year.

## 2. WHAT ARE IPFS AND IPNS?

Participants in IPFS volunteer themselves as nodes in a Distributed Hash Table (DHT)-based overlay network to store data objects. Communications between the nodes is peer-to-peer and the DHT enables a node to find the data it is requesting. An IPFS content creator generates a content identifier (CID) for their document that uniquely identifies it and then advertises its existence to the IPFS network. If someone wants to retrieve that object, the network searches for a node that has saved a copy and retrieves the object from there. We say that the CIDs are immutable because they are based on a cryptographic hash of the object. To allow for mutable content, the InterPlanetary Name Service (IPNS) can be used, as we'll see later. More information about IPFS can be read on the IPFS documentation website[4].

[1] We use the moniker 'Web3' to refer to applications built on the concepts of distributed storage and trust as opposed to being centralized. This is different from 'Web 3.0', a concept coined by Tim Berners-Lee to describe the distribution of personal content, while keeping the centralization of server systems.

[2] An instance of the Turkish Wikipedia is available at the CID bafybeieuutdavvf55sh3jktq2dpi2hkle6dtmebe7uklod3ramihyf3xa4 or more conveniently via the gateway [3].

[3] ERC-20 is a standard for Fungible Tokens on the Ethereum Blockchain [6].

[4] See [13] and [14].

### 2.1 Content-based addressing

The Internet and the World Wide Web is based on location-based addressing. A typical URL points to a server and then the path on that server. For instance, the URL

```
https://www.virusbulletin.com/conference/vb2023/
```

tells a browser to connect to the server at the location 'www.virusbulletin.com' and then ask for the file at '/conference/vb2023/'. If the server is discontinued or the data on the server was moved or removed, then we get the dreaded '404 Not Found' error. Furthermore, there are no guarantees that the content at that location is consistent over time. Content can change by source IP address to provide localized content or to avoid targeting unintended victims, like sandboxes. Content could plausibly be changed in transit, which we combat using transit-layer security (TLS), which establishes a cryptographic link between the server and the client verified with server-side certificates.

Content addressing seeks to fix these issues by using a cryptographic hash scheme to absolutely address the content we ask for. This (normally) obviates the need for a TLS link between a publisher and consumer as the cryptographic hash can be verified easily by the consumer who requested the content using that hash.

For instance, the CID

```
QmYr5ExzJJncpMNhqzhLjkCrRNgm4UmyX28gcYjt5RLYY8
```

points to a document that starts like this:

```
# Version 2022092700, Last Updated Tue Sep 27 07:07:01 2022 UTC
AAA
AARP
ABARTH
ABB
ABBOTT
ABBVIE
...
```

If we change that document to:

```
# Version 2022100600, Last Updated Thu Oct  6 07:07:01 2022 UTC
AAA
AARP
ABARTH
ABB
ABBOTT
ABBVIE
...
```

then we get this CID:

```
QmT7cE6MT7oucfZJpw9sS3qSuQxwuVd1VRkKK4VHfpEL69
```

Even small updates result in completely different IPFS addresses, as you would expect from a cryptographic hash. The hash can (and should) be checked by the receiving client and the user then knows that they are getting the content in the exact version that was expected.

Under the hood, an IPFS object is actually a data structure. For instance, the command:

```
ipfs cat QmQ6Hy9dtDAcSNHeA1GWqhQYqZtYrprg1YsGgnbLj4GkNf
```

will list the file (truncated for brevity):

```
                          The Bulgarian and Soviet Virus Factories
                          ========================================

                                Vesselin Bontchev, Director
                              Laboratory of Computer Virology
                        Bulgarian Academy of Sciences, Sofia, Bulgaria

0) Abstract
===========
...
```

But if we dump the actual object using:

```
ipfs dag get QmQ6Hy9dtDAcSNHeA1GWqhQYqZtYrprg1YsGgnbLj4GkNf
```

we get.

```
{
  "Data": {
    "/": {
      "bytes": "CAIS8LADCiAgICAgICAgICAgIFRoZSBCdWxnYXJpYW4gYW5kIFNvdmlldCBWaXJ1cyBGYWN…
                    …
                …2LCBLaWV2LAoxOTkxLCBJU0JOIDUtODg1MDAtOTMxLVggKGluIFJ1c3NpYW4pCgoaChjwsAM "
    }
  },
  "Links": []
}
```

Large files that don't fit into the 256kb limit are split into chunks and the master CID needs to point to all of them. A large object[5] will look like this:


This is also how IPFS can represent directories where each item is addressed by a content path and CID. While each file in the directory has its own CID, the object can also be retrieved using an address like `/ipfs/<directory CID>/path/to/content`. This is very practical when publishing websites that usually consist of multiple documents.

A website could look like this:

```
$ ipfs ls /ipfs/bafybeibt5fyljzbygv3mxbdbcd3nlyzkuz5uxv2wma2cohqomgxybago5u
QmebEHrJVMVbxkpdLbQP6GFyxDt2z7c8Q86c8T5AMUuQFV 4307     404.html
QmQcStwfddH2jCqwJ3ofMgCq9JZdvDE3EvQK69eWnG3qWA 172346 bundle.min.ace63d … c529057a9.js
QmSnBUG6yBuyfwtroLAGSy7nGRjYWqN9rVa9y3UbHAdjui -        categories/
QmQ1vYdyMrYZSjDA5AfSPSa8MiZViHBDaukw4XwG5stD62 -        fonts/
QmeT2z75wGuERDC5ib8ij9YppwAevaLJmDn8y9rcBvb54b 2875     index.html
QmRtxgDDWPTGD2FahYpjZTAm7cCRzFL6AHdGDVq3V3fhtM 854      index.xml
QmTioEUqpdsUQVJZkjUmbjMipPQCbZzGD69vr9unmaGi9x 26782    main.4e5c639 … b0a84f900798a.css
QmTprUewaMsgMjGeCK5rpU8KxHJScywgjpHRDTdNNUV7PJ 80831    main.css.map
QmQww7SJmFF4vuaUYzxescLXZw9fLycNQQULJH4yEiKtkX -        posts/
QmdoU8Y3uTv45Fdh56GGPLNsWbqY3M3hrE9ebHos1qaUBD 810      sitemap.xml
QmUGBQNz4LKvXAfLRNE8KL2YnXFUPrxCWNwp1zUXySBgnN -        tags/
```

So the directory listing object actually points to a set of files with their own CIDs, so the `index.html` document has the CID:

```
QmeT2z75wGuERDC5ib8ij9YppwAevaLJmDn8y9rcBvb54b
```

but IPFS can also use content paths to retrieve it with the directory CID and the path, like:

```
/ipfs/bafybeibt5fyljzbygv3mxbdbcd3nlyzkuz5uxv2wma2cohqomgxybago5u/index.html
```

[5] We'll use QmRfLZonX6CbDHiFG7G9hYdyDgMCX7AeFq76mMuZ7gTX2Q.

Which then can be retrieved like this:

```
$ ipfs cat /ipfs/bafybeibt5fyljzbygv3mxbdbcd3nlyzkuz5uxv2wma2cohqomgxybago5u/index.html
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta name="generator" content="Hugo 0.80.0" />
            <meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
<meta name="author" content="">
<meta name="description" content="" />
<meta name="keywords" content="" />
<meta name="robots" content="noodp" />
<meta name="theme-color" content="" />
<link rel="canonical" href="http://example.org/" />
…
```

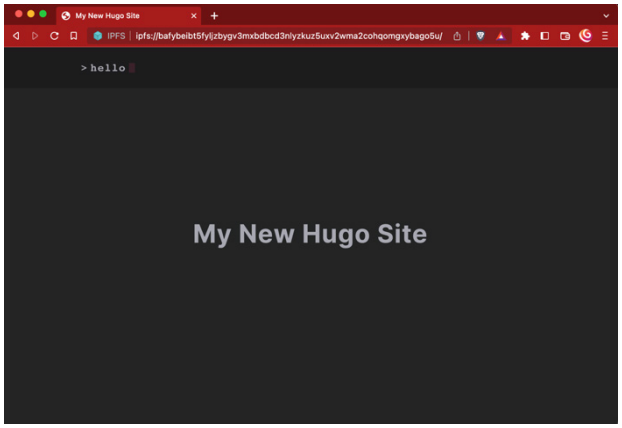
An IPFS-aware browser can address this directedly using the IPFS scheme:

```
ipfs://ipfs/bafybeibt5fyljzbygv3mxbdbcd3nlyzkuz5uxv2wma2cohqomgxybago5u/index.html
```

or just:

```
ipfs://ipfs/bafybeibt5fyljzbygv3mxbdbcd3nlyzkuz5uxv2wma2cohqomgxybago5u/
```

And in this case, it looks like this:




### 2.2 Content retrieval

Finding and addressing the content is a major part of the problem that IPFS solves with the DHT. However, the user still needs to retrieve the content. This is done with a separate protocol called Bitswap. Simply put, once the DHT has found nodes with the blocks, the consuming node can ask all of those nodes for blocks. If the counterparty has the block it will deliver it. Otherwise it will deny having the block. Of course, in practice, nodes are going off- and online all the time, so the consumer may ask for data but never get an answer, leading to timeouts.

### 2.3 Mutable 'naming': IPNS

Websites tend to want to update their content over time. We need a way of mapping some fixed address to the current contents. We can do that with a 'name' service called IPNS.

IPNS, which is the InterPlanetary Name System, uses a public/private cryptography scheme to map an immutable address to different content. First we generate a public and private key that we'll use for a particular piece of content. We publish an IPNS record that contains the IPFS CID and content path and signed with the private key. So, for example:

```
/ipns/k2k4r8oid7ncjwgnpoy979brx3r9ellvvwofht57mc9q4jzlxtydalvf
```

points to

```
/ipfs/QmYr5ExzJJncpMNhqzhLjkCrRNgm4UmyX28gcYjt5RLYY8
```

But if we decide to update that document, we can reassign the `/ipns/...` address to point to the new object:

```
/ipfs/QmT7cE6MT7oucfZJpw9sS3qSuQxwuVd1VRkKK4VHfpEL69
```

By default, the IPNS record is published over the DHT and it is important to republish records on a regular basis as the specification implies that DHT peers will forget IPNS records after 24 hours. As far as we have been able to determine, the most common IPFS node software, *Kubo*[6], republishes every four hours by default.

Note that we can also create IPNS records based on CIDs with paths. For instance:

```
/ipns/k51qzi5uqu5dhr3yemb9kkwr7n8uo2lvscwufrjg9f2x19b5790q1rlg2omc2r
```

can point to

```
/ipfs/QmU4dmCxj7fupy4w2aQv3NRwjG32w4RrR6aPewTigBvhb1/tags/index.html
```

This allows us to create static websites that can be changed over time.

Unfortunately for IPNS, at the time of writing, it is very slow. Whereas DNS lookups are nearly imperceptibly fast, IPFS records need to be retrieved from the DHT and that can take noticeable time if they are regularly being republished. If the publisher stops pushing their records then we may never get the IPNS record we want. This is a known issue and has been recognized as needing a fix.

### 2.4 What is a Content Identifier?

While a CID looks like a hash, it is in fact a data structure that ultimately is a Merkle-DAG hash and a description of the hashing scheme. The Merkle-DAG is needed because under the hood the document is split into chunks. In the easy case, the blocks are sequential and the CID is formed from the hashes of each chunk. In some cases there is a whole tree of hashes and because IPFS will optimize storage by reusing identical chunks, the tree can form a directed acyclical graph (DAG).

There are two versions of CIDs as of this writing. Version 1 usually starts with the letter '`b`' while the most common version, version 0, starts with '`Qm`'. For instance, the CID v0:

```
QmQ6Hy9dtDAcSNHeA1GWqhQYqZtYrprg1YsGgnbLj4GkNf
```

breaks into the following fields:

| version | multibase | codec | Hash function | digest |
|---|---|---|---|---|
| cidv0 | base58btc | dag-pb | sha2-256 | 1A09CA7FD36B590E1BBA9419D7FBDF5FA2F601D48 3C46A663F7E7435B922B07C |

Most of this is implied. The multibase is fixed to base58btc, which is a Bitcoin variant of base58 encoding. The structure encoding is also fixed to the dag-pb codec, which is ProtoBuf-based. Lastly, the digest is fixed to SHA-256.

It was quickly realized that more flexibility was needed and version 1 was introduced with more options. For example:

```
bafybeia2bhfh7u3llehbxouudhl7xx27ul3advedyrvgmp36oq23sivqpq
```

breaks down into:

| version | multibase | codec | Hash function | digest |
|---|---|---|---|---|
| cidv1 | base32 | dag-pb | sha2-256 | 1A09CA7FD36B590E1BBA9419D7FBDF5FA2F601D483C 46A663F7E7435B922B07C |

While there were no real options for these in v0, we have a large selection for both the multibase [15] and codec [16] fields in v1. It is always possible to convert CID v0 into v1, but the inverse is only possible if the SHA-256 hash function was used. The consequences of this should be clear: any given object could have multiple valid CIDs by converting to different multibases and codecs in v1, or just converting from v0 to v1.

### 2.5 Gateways

This is all great if you are using the IPFS protocol, probably via a library or the command line tool. But most people use a web browser and most of these do not support the IPFS protocol. In recognition of this, IPFS gateways were invented that allow access to content via HTTP(S), and in many cases this was built into the software running the nodes. That means that when you run the IPFS daemon, apart from the native API, there is also a tool to resolve IPFS CIDs and IPNS key fingerprints using HTTP URLs. If you want to retrieve the contents of:

[6] *Kubo* is the most common implementation of the IPFS daemon and command line tool. See [14].

```
/ipfs/QmYr5ExzJJncpMNhqzhLjkCrRNgm4UmyX28gcYjt5RLYY8
```

you can also use your browser by going to the URL:

```
http://127.0.0.1:8080/ipfs/QmYr5ExzJJncpMNhqzhLjkCrRNgm4UmyX28gcYjt5RLYY8
```

There a number of public gateways available. For instance:

```
https://ipfs.io/ipfs/QmYr5ExzJJncpMNhqzhLjkCrRNgm4UmyX28gcYjt5RLYY8
```

and

```
https://cloudflare-ipfs.com/ipfs/QmYr5ExzJJncpMNhqzhLjkCrRNgm4UmyX28gcYjt5RLYY8
```

Both will return the same content. Any public and private gateway will be able to retrieve IPFS content and host it this way. A public list of gateways is maintained at *GitHub* [17].

By convention, gateway URLs to access IPFS and IPNS can take two formats:

| Style | Canonical form of access |
|---|---|
| path | `https://{gateway URL}/{ipfs\|ipns}/{CID}/{optional path}` |
| subdomain | `https://{CID}.{ipfs\|ipns}.{gatewayURL}/{optional path}` |

Gateways are a problem for the user who must trust that the gateway is not changing the content after retrieval. The correct way of dealing with this issue is to use the CID hash digest to verify the content retrieved using HTTP[S]. It would be best if the browser would do this for the consumer, but we are not aware of this being done yet. Instead, the browsers seem to trust the gateway server and the TLS connection. This is also an obvious challenge to filtering malicious content as we need to identify the CID in the URL and then block it at all gateways that we know of.

## 2.6 Name services

We covered IPNS, but despite the name, it is not really a human-friendly name service. Also, as we've seen, it can be slow. A few alternatives exist though, the most useable being DNSLink.

### *2.6.1 DNSLink*

The DNSLink [18] method uses a DNS TXT record to map a domain name to an IPFS CID. This requires an old-school domain name and a way of setting the TXT record on the name server.

For example, we'd like to see the content on the IPFS mirror of *Wikipedia*. The domain for this is `en.wikipedia-on-ipfs.org`, and if we ask for this in an IPFS-aware browser we will get the `CID`. The browser will take the hostname and prepend `_dnslink` to it to get `_dnslink.en.wikipedia-on-ipfs.org`[7]. With this, we can retrieve the `TXT` record:

```
% dig +short _dnslink.en.wikipedia-on-ipfs.org TXT
"dnslink=/ipfs/bafybeiaysi4s6lnjev27ln5icwm6tueaw2vdykrtjkwiphwekaywqhcjze"
```

Given this, the browser can retrieve the content from IPFS by parsing the string for the CID and then querying the IPFS network. This also works for IPNS content, in which case the `TXT` field will look like `"dnslink=/ipns/..."`.

## 2.7 Ethereum Name Service

The *Ethereum Name Service* (*ENS*) is a blockchain-based name service that registers domain names on the Ethereum blockchain[8]. *ENS* can be used to make references to IPFS content. While the normal top-level domain, TLD, for *ENS* is `.eth`, *ENS* also supports `.com`, `.org`, `.io`, `.app`, `.xyz`, and `.art`. As with DNSLink, you need to set a DNS TXT record. In this case, the special host name `_ens` is used and the TXT record should be set to `a=[your Ethereum address]`. The Ethereum address is obtained by connecting a wallet to the *ENS* manager app at `epp.ens.domains` and registering the existing domain name that will use the *ENS*[9].

Name resolution can only be performed by supporting browsers which need to check for the `_ens` host name in DNS and then do the actual lookup on the Ethereum blockchain.

Using *ENS* only costs 'GAS' fees when registering the domain name on the blockchain. Otherwise it is a free service.

For those who don't want to or can't use a `.eth`-aware browser, *ENS* uses the `eth.link` domain to connect `.eth` names to the Domain Name System, or DNS, which is what web browsers use to connect to websites. This EthLink service

[7] Note that in the past, _ipfs was used instead of _dnslink.
[8] See [19] for more information.
[9] See [20] for more information.

enables `.eth` domain users to create viewable websites for themselves with their *ENS* names and make them accessible from a standard browser. An alternative to `eth.link` is `eth.limo`. So, for example, `vitalik.eth` can be used by a standard browser as `vitalik.eth.link` or `vitalik.eth.limo`.

### 2.8 Unstoppable Domains

*Unstoppable Domains* also stores domains on a blockchain. They offer the TLDs: `.crypto`, `.nft`, `.888`, `.x`, `.blockchain`, `.bitcoin`, `.wallet`, `.coin`, `.dao` and `.zil` on the Ethereum blockchain. To resolve one of these domains you need to query the blockchain, and JavaScript code is available for this.

Other blockchain-based name services include the venerable *Namecoin* (`.bit`), based on Bitcoin, *Bonfida*, a.k.a. *SNS*, based on Solana Blockchain, *Forever* (`.forever`), based on Ethereum, and *Point* (`.point`), which has its own blockchain, POINT.

## 3. PINNING AND PUBLISHING TOOLS

Ironically, one problem with IPFS data is that it is potentially ephemeral. IPFS will store data that has previously been retrieved in its cache, but the garbage collection process will remove some (or all) of those objects after some time. The criteria for removal depend on the configuration of the individual node. While this is great for content that was retrieved to be read and then disposed of, it's not great for content that one wants to publish and make sure is available for others to read indefinitely.

Pinning is the mechanism whereby you to tell IPFS to keep a given object on your node permanently. The `ipfs pin` command is used in the *Kubo* client, for example. However, this is not redundant. If your node crashes or is cut off from the Internet, your content may not be available. In some cases, cached copies will be available, but that is not something you can count on. For this reason, there are third-party remote pinning services. These are often brokers who either provide pinning themselves on their own infrastructure, or work with other systems to provide storage. This needs to be paid for, so often these systems use cryptocurrency to remunerate their third-party nodes. However, how does the customer of such a service know that their content is actually being pinned as many times as they have paid for? In trust-less systems, this is done by some form of proof of storage, often zero knowledge proof based.

One application of IPFS is to publish websites. For this reason, there are more fully featured IPFS website publishing tools, for instance *Pinata* and *Estuary*[10]. These help a web publisher go from code to website easily and then maintain it over time, as well as handling pinning.

## 4. BROWSER SUPPORT

The success of IPFS will depend heavily on browser support, and that is currently lacking. Ideally, a browser would have built-in protocol support for IPFS. This doesn't seem to be the case for any browser yet. Instead, the best support connects to the local IPFS daemon when the IPFS or IPNS protocol is specified in the address bar. The next best support is to recognize the protocol, but convert it to a gateway request.

The *Brave* browser has the best current support. It recognizes the running daemon and if it is not running, will use a gateway that can be configured. An `ipfs://` or `ipns://` address is therefore usable just like a regular HTTP URL.

The *Opera Crypto* browser (not the standard version) will also recognize the IPFS addresses [30] and translate them to a gateway lookup.

For the rest of the browsers we are aware of, support is available only through the IPFS Companion [31]. It is officially available for *Firefox*, *Chrome*, *Edge* and *Opera*. Other browsers based on these are probably also supported.

However, the mobile versions of these are not supported. A special case is the *Mises* mobile browser[11], the developers of which have indicated that they will support it in version 2. The *Agregore* browser[12] also has IPFS support, but it is not a mainstream browser. Mobile is a bad fit for IPFS as the constant IPFS peering traffic would be a severe drain on bandwidth and battery. It would make more sense to integrate gateways into near-line telco infrastructure.

## 5. THREATS

The low-hanging fruit for IPFS is its use in phish and malware distribution, where its ephemeral nature is often a benefit. We will cover real-world examples in the findings section. There the potential is greater than that, though.

### 5.1 Phishing

Phishing using IPFS is currently done by putting IPFS gateway URLs in the email. We assume attackers have the capacity

[10] Examples as of writing are *Estuary* [21], *Filebase* [22], *Infura* [23], *NFT.Storage* [24], *Pinata* [25], *Web3.Storage* [26], *Eternum* [27], *Fleek* [28] and *FileCoin* [29].
[11] *Mises* is a fork of the *Chromium* browser: [32].
[12] See the project at [33].

to change the gateway server in the phishing URL very quickly if needed. IPNS may also pose advantages, once practical, although location-based content will no longer be possible this way.

What we don't see yet is the use of the `ipfs://` or `ipns://` protocols yet. This is because of the lack of browser support for native IPFS. If some mainstream browser decides to implement protocols handlers for these, this could change things radically.

Threat actors can get around the problem of the slowness of IPFS retrieval by the gateways by priming them. That is, after publishing their content on the IPFS network, call various gateways with the appropriate CID. The gateways will load and cache that content.

### 5.2 Malware

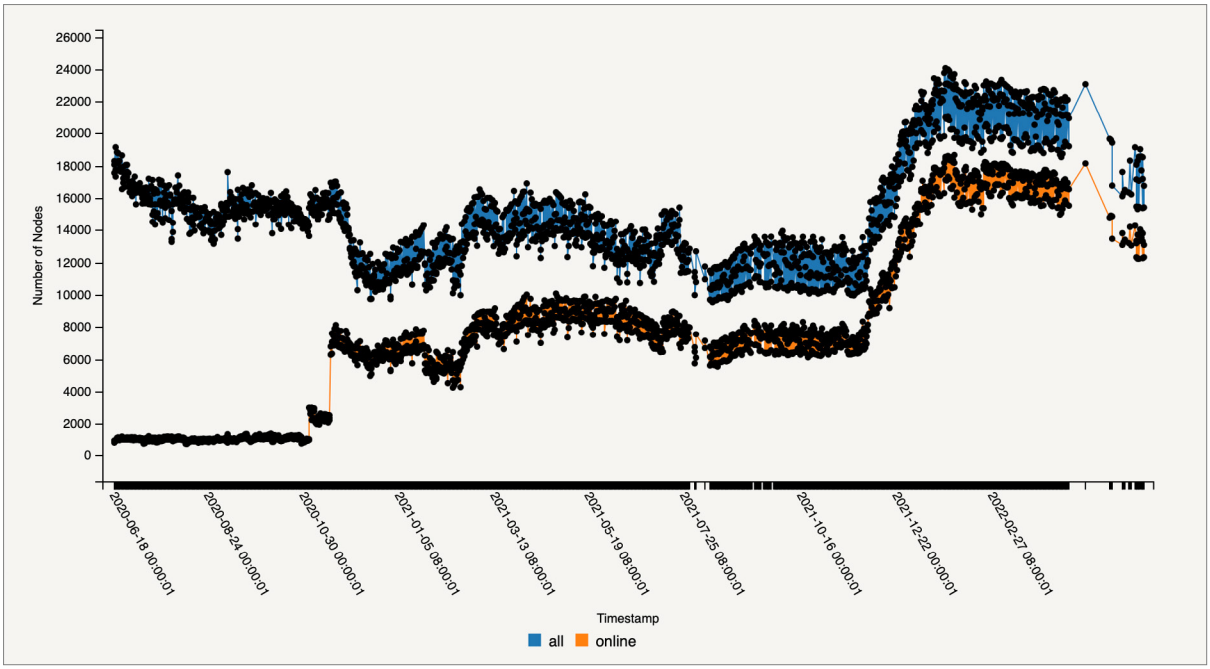
Hosting malware on IPFS may bring threat actors various advantages in terms of takedown-resistance and sandbox-avoidance. Takedown-resistance stems from the caching of copies of content throughout the whole IPFS network. Finding the one server to take down is impossible. In reality, IPFS content doesn't get replicated as much as it could in theory, so if the threat actor doesn't arrange for pinning to other servers it may only reside permanently on one machine. However, locating that machine is nearly impossible, given the way IPFS works. Malware poses a threat to the practice of sandboxing if we do not monitor for IPFS activity. There are also other aspects of IPFS malware that are challenging for sandboxes, but we have not observed any of these behaviours yet.

## 6. FINDINGS

We wanted to see how bad the problem of IPFS was and how it is developing. Besides finding some threat reports we also looked at our own data.

### 6.1 Size of IPFS

IPFS has been around since 2015, but has it been catching on? The Weizenbaum Institute published statistics on the size of the IPFS network from 2020 until mid-2022 [34].



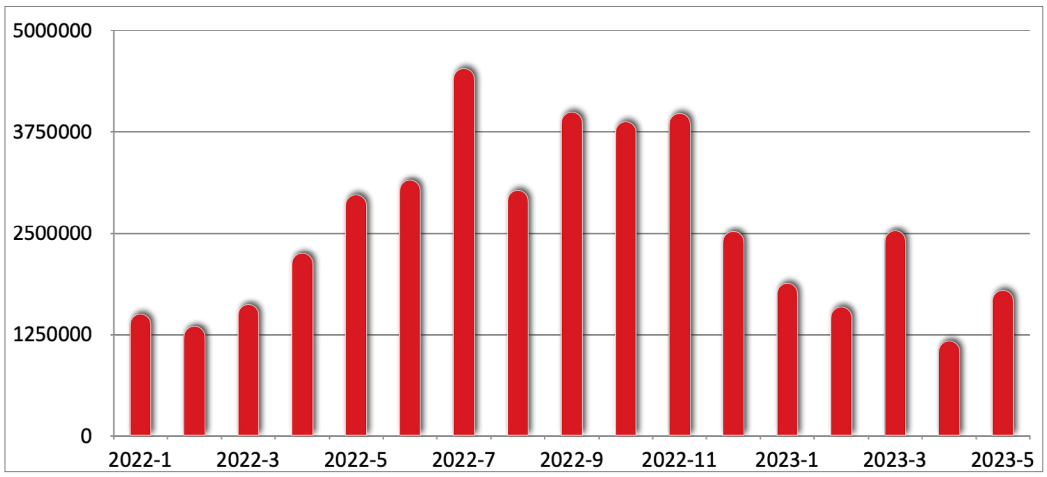

We can see that it was relatively steady until December 2021, at which time it grew fairly rapidly. And while it declined somewhat until the summer, there are about 16k nodes online, which makes for a fairly stable network. Unfortunately, the Weizenbaum Institute stopped regular scanning of IPFS nodes in 2022, so we don't have any more recent independent data.

### 6.2 Corporate use of IPFS

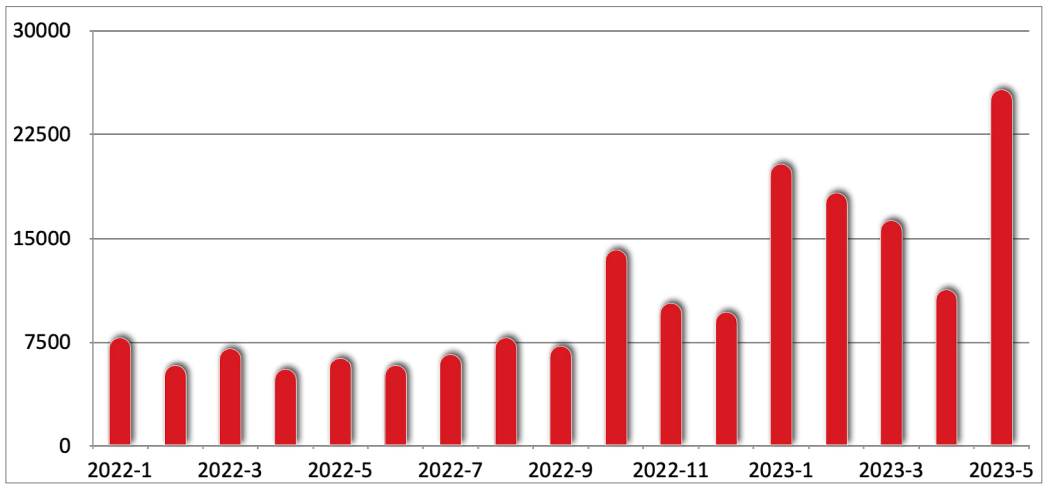
How IPFS is used in a corporate environment is very under-explored. In a presentation at the IPFS Camp in 2020, *Netflix* claimed to be exploring its use in container distribution [35]. In a corporate environment IPFS offers so-called swarm keys to separate the peer-to-peer network from the main network. But this makes it more difficult to measure and we will have to wait for better measurements to be found to understand corporate use better.

### 6.3 Growth in Trend's telemetry

While this is not the same methodology as the Weizenbaum Institute used, we were able to track IPFS gateway usage in our own data as a proxy for general IPFS usage.



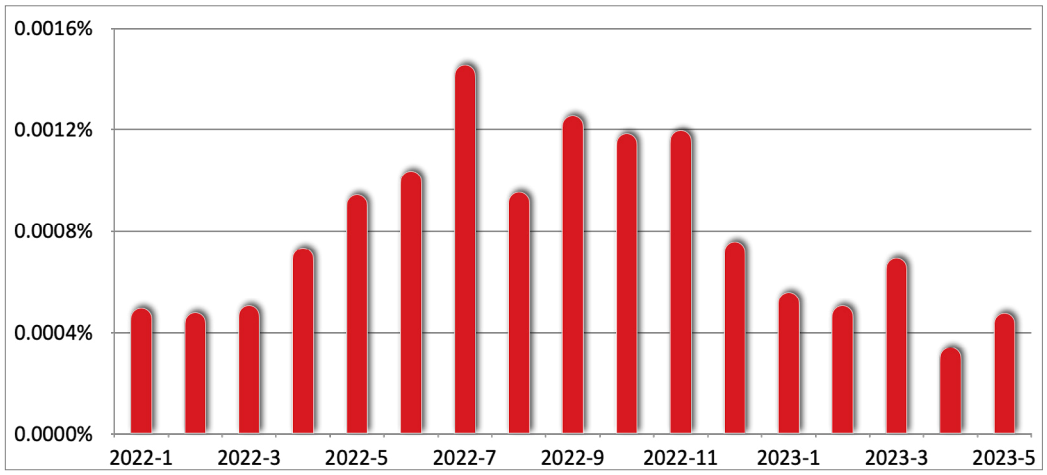

*IPFS gateway usage per month.*




*IPNS gateway usage per month.*

We can see that IPFS was doing well mid-2022 but has since slipped. We were able to trace that boost to the demise of *Siasky*, which ceased operations in November 2022 [36]. IPNS use is really tiny but has been growing in fits and starts.

By percentage, the IPFS traffic is still tiny compared with the total volume of events:



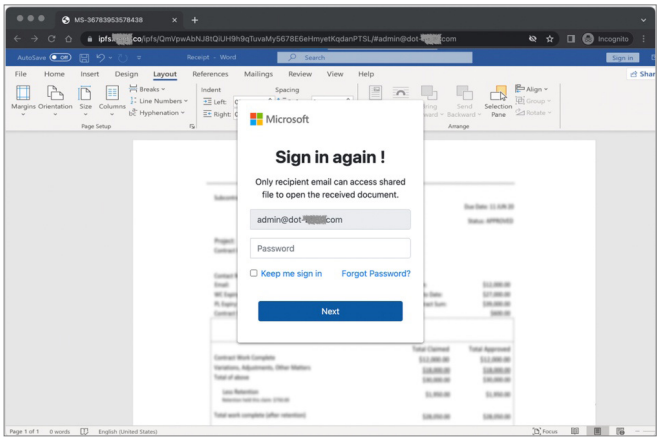

And IPNS traffic is negligible. All things considered, is this a non-issue? Unfortunately, we already see significant abuse.

### *6.3.1 IPFS spear phishing*

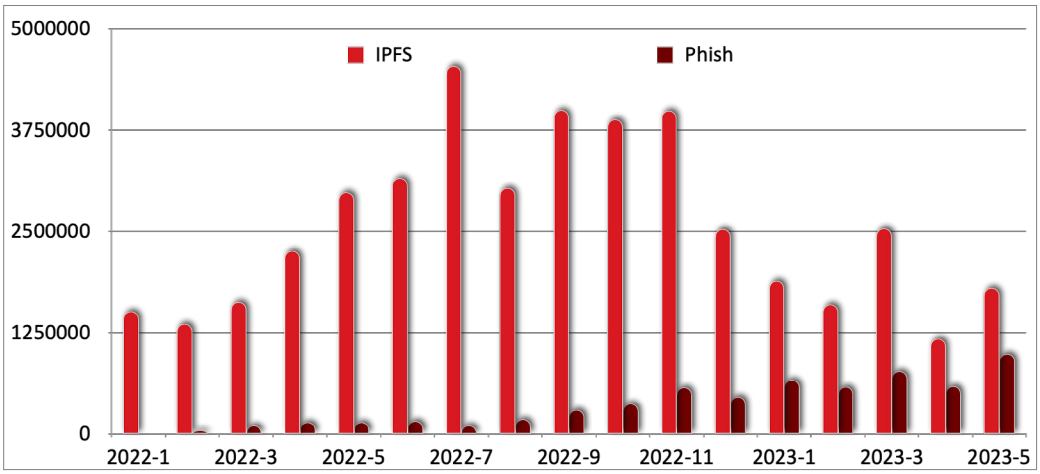
There are a variety of phish, but one type of spear phishing is relatively easy to find. The gateway URLs look like this:

```
https://<CID>[.]ipfs[.]w3s.link/aws.html?email=<email-address>
https://<gateway-server>/ipfs/<CID>?email=<email-address>
https://<gateway-server>/ipfs/<CID>#<email-address>
```

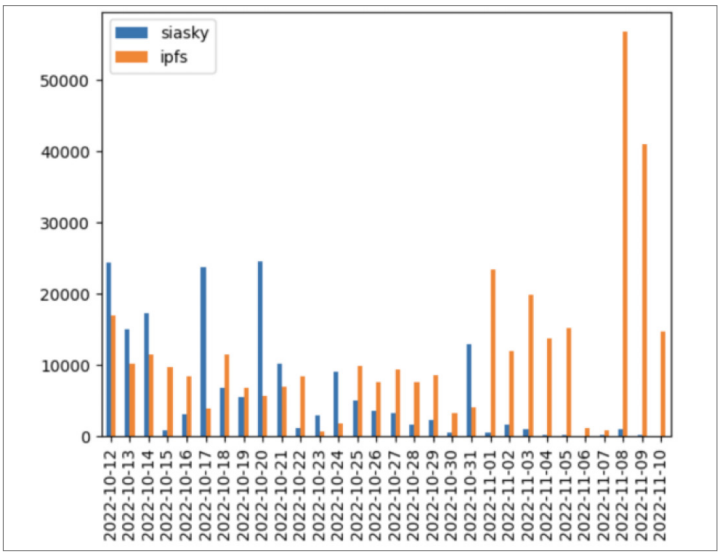
or variants of those. For example:



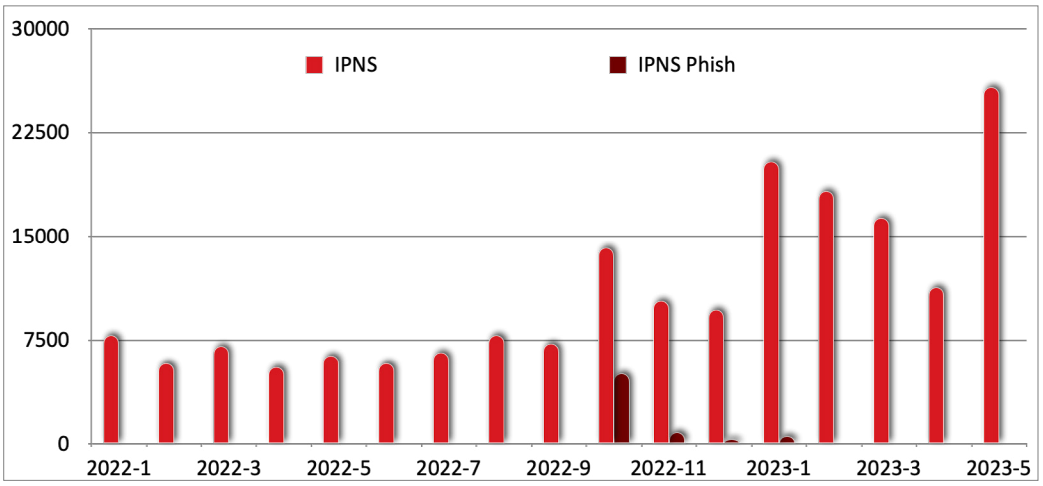

As a proportion of the IPFS traffic, this type of phish has been growing steadily:




As before, it should be noted that IPFS phish also got a boost from the demise of *Siasky* in late 2022. We can see this effect in a separate, more detailed, analysis we did in 2022:



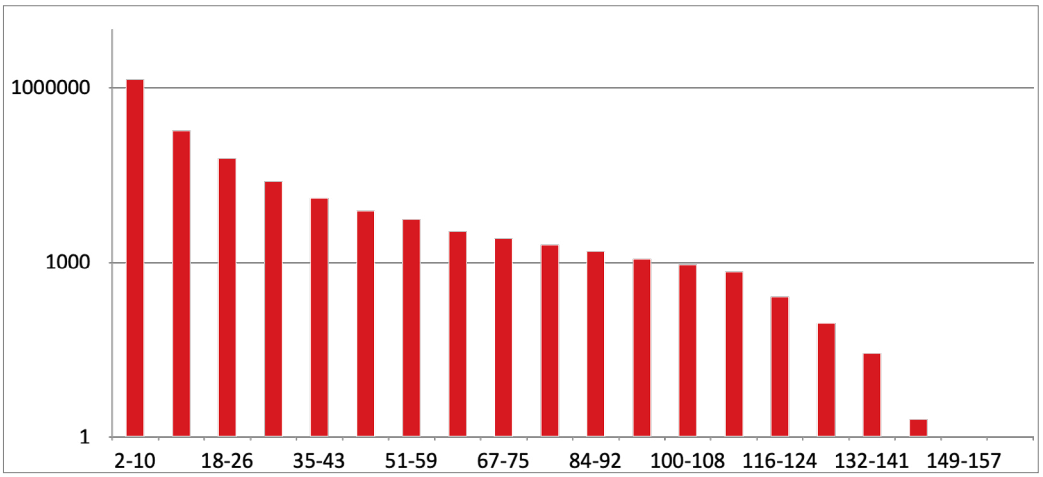
IPNS phish are rare and only really registered when *Siasky* ceased operation:




At the moment, they don't play any meaningful role, and probably won't until the performance issues are resolved.

### 6.4 Blocking IPFS gateway access

We also wanted to know how viable it was to just block a given gateway URL. For that, we looked at the distribution of domains that host a given CID.




A significant number of IPFS objects are being accessed via a large number of gateways. The maximum we saw was a single IPFS CID accessed by 165 different gateways. The bulk was below 100.

The consequence of this is that blocking a complete URL to block phish is only of limited use. Better would be to block that CID on all known gateway servers. Even better: generate generic patterns for blocking the CID on any gateway, known or unknown.

### 6.5 Malware detection

We wanted to see if malware, hosted by IPFS, would be blocked by traditional anti-virus solutions. With the well-known EICAR file saved to IPFS we downloaded it with the IPFS client to a machine running *Trend*'s *Titanium* product. As expected, as soon as the file touched the file system, it was detected.


We expect this result will be same for all leading anti-virus products. However, gateway scanning will likely fail as the IPFS protocol works very differently from HTTP.

It doesn't look like malware authors have started embracing IPFS yet. In a previous paper, Cedric Pernet found a mere 180 malware samples [37], and when repeating some of the previous research, over the period of January 2022 to June 2023, no

malware samples could be retrieved via IPFS. This may mean that any samples that had been available were removed by the creators or by anti-virus over time.

IPFS could provide malware creators with some advantages, given that IPFS is designed to fly under the radar. In particular, such malware might be missed for a variety of reasons in sandboxes, although it is prudent not to disclose of how before we have observed it. We will see if that changes over time.

## 7. CONCLUSIONS

The threat from IPFS is currently mainly from phish. The main advantages it offers threat actors is that it is hard to block, harder to take down, and not on most defender's radar yet. On the other hand, IPFS phish is still a small proportion of the overall phish and malware is nearly negligible. However, it is growing.

IPFS is still developing and being improved upon. There are a number of start-ups around the technology, as well as similar competing technologies, that are actively being financed by venture capital. The motivation to build this new Web3 architecture on distributed technology is to create a data and compute platform that their start-ups can build on without restrictions. Many VC-backed companies saw their products killed by *Twitter*, *Facebook* and others when they removed or restricted access to the data in these social media platforms. Censorship avoidance, on the other hand, seems to be a convenient justification that is easier to sell.

For this reason, it is likely that development on IPFS will continue, the kinks in the system will be ironed out and the network will continue to grow. There are more developments in the IPFS ecosystem that are concerning and need to be watched carefully before they become new threat vectors.

It's unlikely that IPFS will go mainstream soon, but that will not stop threat actors from being the primary beneficiaries in the short term.

## REFERENCES


[1] Benet, J. IPFS - Content addressed, versioned, P2P file system. 2014.

[2] Phippen, J. W. Why Turkey Blocked Access to Wikipedia. The Atlantic. 29 April 2017. https://www.theatlantic.com/news/archive/2017/04/turkey-blocks-wikipedia/524859/.

[3] Turkish Wikipedia. https://tr.wikipedia-on-ipfs.org/wiki/.

[4] Audius. https://audius.co/.

[3] Vryonis, P. IPFS Camp 2022 and CoD Summit². WeatherXM. 9 November 2022. https://blog.weatherxm.com/ipfs-camp-2022-and-cod-summit²-a2a208866d8.

[6] ERC-20: Token Standard. https://eips.ethereum.org/EIPS/eip-20.

[4] Arghire, I. New Malware Lays P2P Network on Top of IPFS. Security Week. June 2019. https://www.securityweek.com/new-malware-lays-p2p-network-top-ipfs%E2%80%99.

[5] Anomali Threat Research. The InterPlanetary Storm: New Malware in Wild Using InterPlanetary File System's (IPFS) p2p network. June 2019. https://www.anomali.com/blog/the-interplanetary-storm-new-malware-in-wild-using-interplanetary-file-systems-ipfs-p2p-network.

[6] Agregado, K.; Udquin, K. IPFS: The New Hotbed of Phishing. Trustwave. July 2022. https://www.trustwave.com/en-us/resources/blogs/spiderlabs-blog/ipfs-the-new-hotbed-of-phishing/.

[7] Brumaghin, E. Threat Spotlight: Cyber Criminal Adoption of IPFS for Phishing, Malware Campaigns. Talos Intelligence. November 2022. https://blog.talosintelligence.com/ipfs-abuse/.

[8] Cofense. Phishing Campaigns Abusing Web3 Platforms Increases 482% in 2022. November 2022. https://cofense.com/blog/phishing-campaigns-abusing-web3-platforms-increases-482-in-2022.

[9] Pernet, C. IPFS phishing on the rise, makes campaign takedown more complicated. Tech Republic. August 2022. https://www.techrepublic.com/article/ipfs-phishing-on-the-rise-makes-campaign-takedown-more-complicated/.

[13] IPFS docs. https://docs.ipfs.tech.

[14] IPFS docs. ipns://docs.ipfs.tech/.

[14] Kubo. https://github.com/ipfs/kubo.

[15] Multibase list. https://github.com/multiformats/multibase.

[16] Multicodec list. https://github.com/multiformats/multicodec.

[17] List of stable IPFS gateways. https://github.com/ipfs/public-gateway-checker/blob/master/src/gateways.json.

[18] DNSLink. https://dnslink.dev.

[19] ENS. https://ens.domains/.

[20] brantly.eth. Step-by-Step Guide to Importing a DNS Domain Name to ENS. 26 August 2021. https://medium.com/the-ethereum-name-service/step-by-step-guide-to-importing-a-dns-domain-name-to-ens-d2d15feb03e8.

[21] Estuary. https://estuary.tech/.

[22] Filebase. https://filebase.com/.

[23] Infura. https://infura.io/.

[24] NFT.Storage. https://nft.storage/.

[25] Pinata. https://www.pinata.cloud/.

[26] Web3.Storage. https://web3.storage/.

[27] Eternum. https://www.eternum.io/.

[28] Fleek. https://fleek.co/hosting/.

[29] FileCoin. https://filecoin.io.

[10] Opera. Opera adds Unstoppable Domains support to desktop and iOS browsers, providing you with seamless access to the decentralized web. 28 April 2021. https://blogs.opera.com/desktop/2021/04/opera-web3-support-unstoppable-domains-nft/.

[31] IPFS Companion. https://docs.ipfs.tech/install/ipfs-companion.

[32] Mises browser. https://github.com/mises-id/mises-browser-chromium.

[33] Agregore browser. https://agregore.mauve.moe

[11] Balduf, L.; Henningsen, S.; Florian, M.; Rust, S.; Scheuermann, B. Monitoring Data Requests in Decentralized Data Storage Systems: A Case Study of IPFS. 2022. Cornell University. https://arxiv.org/abs/2104.09202.

[35] IPFS Camp - Edgar Lee of Netflix. https://youtu.be/wNfk05D887M.

[36] Vorick, D. Full Shutdown of Skynet Labs Websites and Services. Skynet. 12 October 2022. https://skynetlabs.com/news/skynet-labs-full-shutdown.

[12] Pernet, C.; Horejsi, J.; Lu, L. IPFS: A New Data Frontier or a New Cybercriminal Hideout? Trend Micro. March 2023. https://www.trendmicro.com/vinfo/us/security/news/cybercrime-and-digital-threats/ipfs-a-new-data-frontier-or-a-new-cybercriminal-hideout.