



SANS Network Security: Las Vegas Sept 4-9.



Handler on Duty: Guy Bruneau

Threat Level: **Green**

Homepage

Diaries

Podcasts

Jobs

Data

Tools

Contact Us

About Us

Slack Channel

Mastodon

Bluesky

X

previous

next

IPFS phishing and the need for correctly set HTTP security headers

Published: 2023-03-15. Last Updated: 2023-03-15 11:22:07 UTC

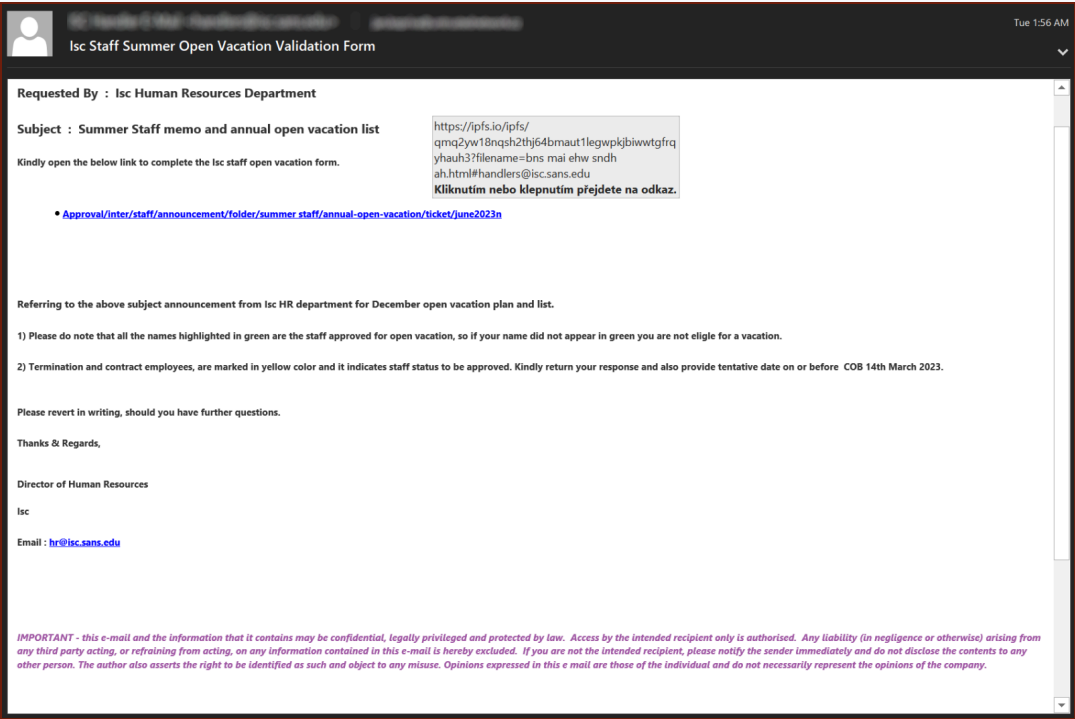
by [Jan Kopriva](#) (Version: 1)



0 comment(s)

In the last couple of weeks, I've noticed a small spike in the number of phishing messages that carried links to fake HTML login pages hosted on the InterPlanetary File System (IPFS) – an interesting web-based decentralized/peer-to-peer data storage system. Unfortunately, pretty much any type of internet-connected data storage solution is used to host malicious content by threat actors these days, and the IPFS is no exception. In fact, it seems to have been used to host phishing pages since at least the beginning of 2022^[1].

The recent wave of phishing messages is therefore not new in its use of the distributed file system, nor in the social engineering techniques it uses. What makes it somewhat interesting, besides the fact that it depends on IPFS, is that it also shows quite nicely the need for organizations to ensure that security-related headers are set by their web servers. This is because although, as you may see from the examples shown below, all the e-mails linking to pages hosted on IPFS were different...





Homepage

Diaries

Podcasts

Jobs

Data

Tools

Contact Us

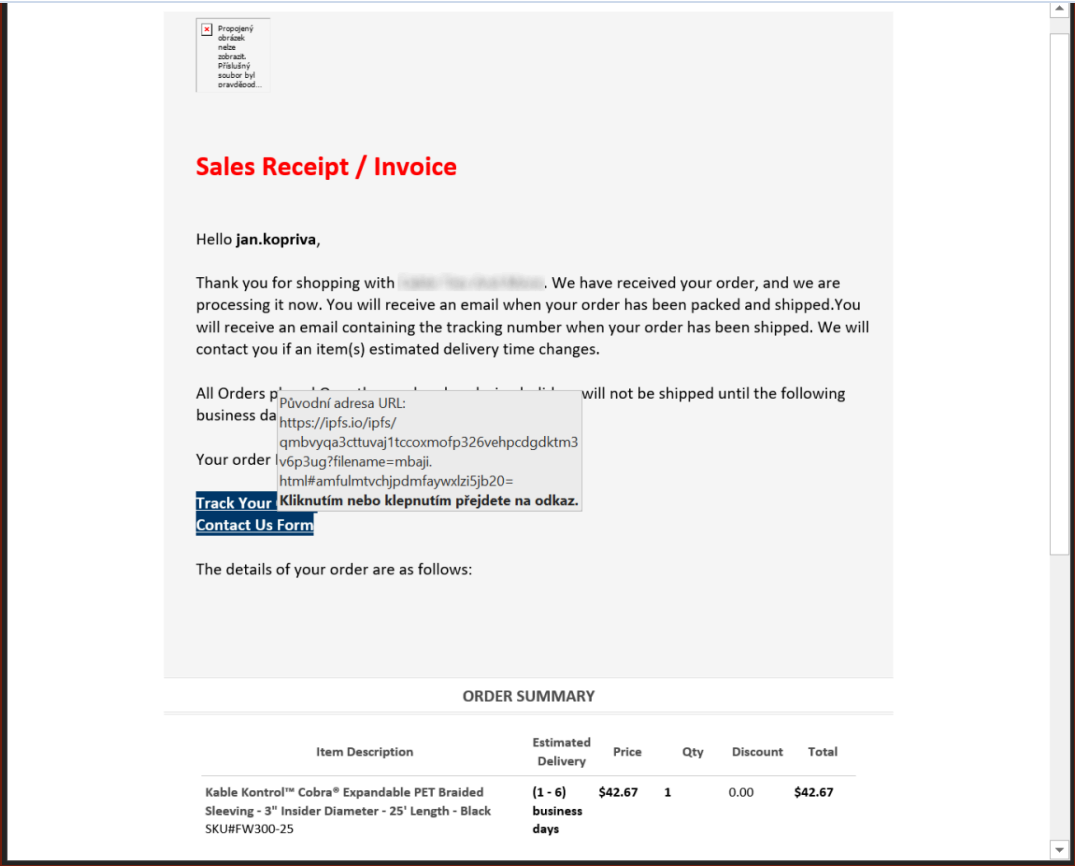
About Us

Slack Channel

Mastodon

Bluesky

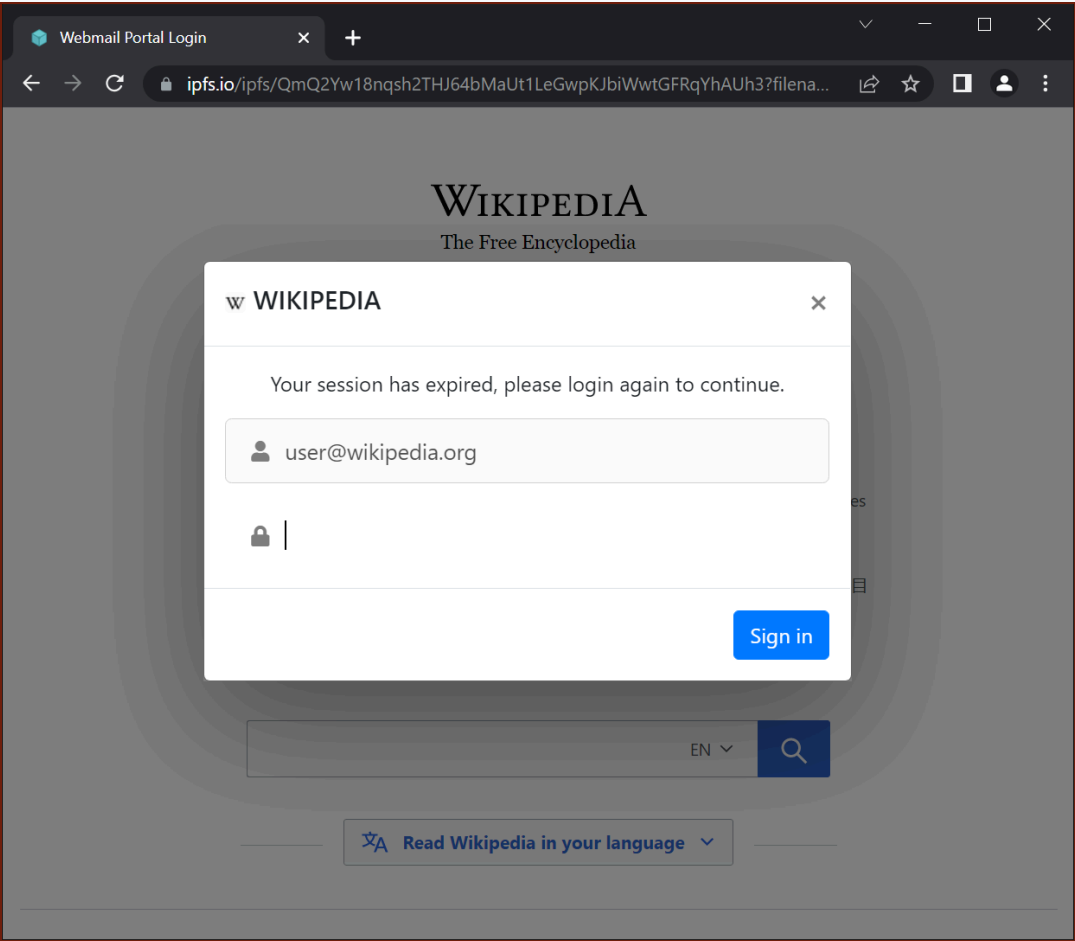
X



...the HTML pages they linked to were very similar and all used the same clickjacking-related “login overlay over a legitimate website” trick.

This technique has been with us for a while now. It is based on the use of a HTML page, on which a fake login form is placed over an iframe, in which a legitimate website is loaded from a domain corresponding to the e-mail address of the recipient of the original phishing message. Since the e-mail address is (usually) passed to the phishing page in a parameter and the corresponding legitimate website is loaded dynamically, the result may look relatively believable, even though the fake login page was not tailor-made for a specific “target” individual or organization.

You may see an example of a reasonably good-looking result of this technique in the following image.



However, if the Content Security Policy and/or the X-Frame-Options HTTP headers are set, which is one of the standard defenses against clickjacking[2], the resulting login page is much less believable, as you may see in the following image...



Sign In

Sign Up

Homepage

Diaries

Podcasts

Jobs

Data

Tools

Contact Us

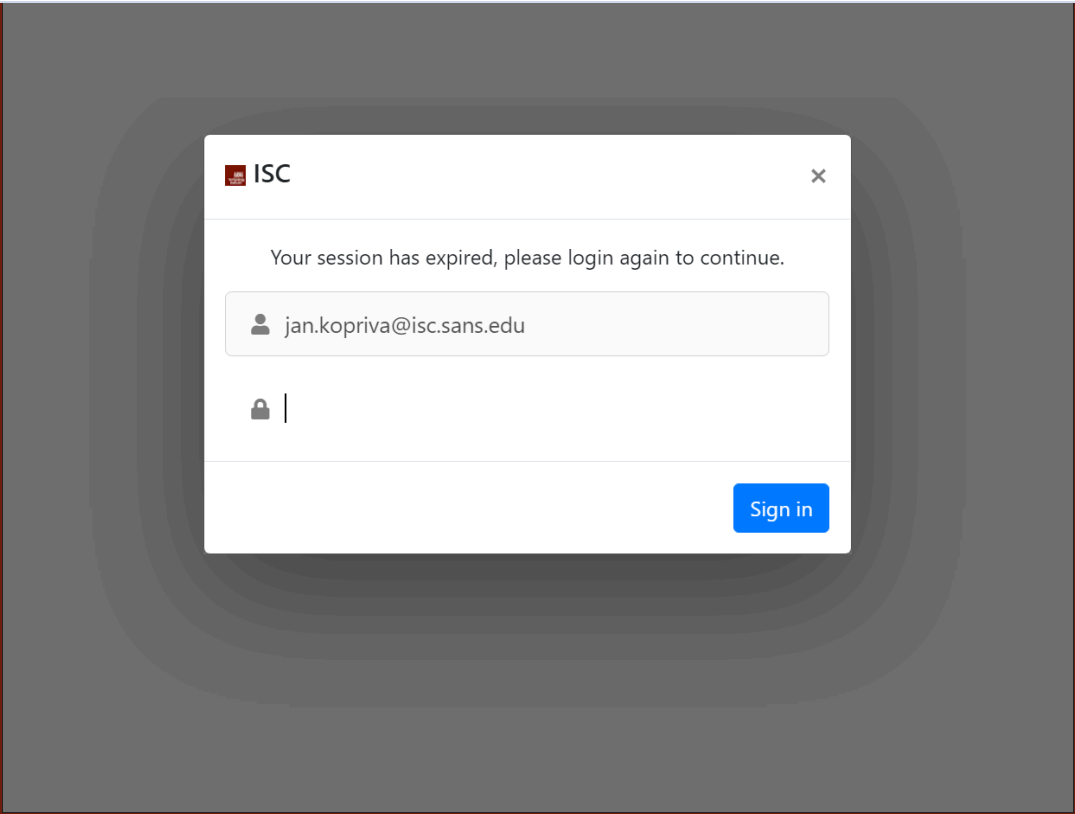
About Us

Slack Channel

Mastodon

Bluesky

X



Since clickjacking defenses have historically been used primarily to protect websites with sensitive functionalities and/or login forms, relevant security headers may not always be set for the “main website” of an organization.

However, as this phishing shows (and as do many others we’ve seen before), the lack of these headers on almost any website can potentially be a problem. Therefore, maybe the time has come to make CSP and other HTTP security headers the norm and not the exception. Although their use can sometimes be a little problematic, the corresponding issues can always be solved, and the simple use of few HTTP headers can make phishing attempts, such as the ones mentioned above, much less effective.

- [1] https://www.trendmicro.com/en_us/research/22/l/web3-ipfs-only-used-for-phishing---so-far.html
[2] <https://owasp.org/www-community/attacks/Clickjacking>

Jan Kopriva
[@jk0pr](#)
[Nettles Consulting](#)

Keywords: [HTML](#) [HTTP](#) [Phishing](#)

[0 comment\(s\)](#)

previous

next

Comments

[Login here to join the discussion.](#)


[Top of page](#)

[Diary Archives](#)




Sign In


[Sign Up](#)

 [Homepage](#)


 [Diaries](#)


 [Podcasts](#)

 [Jobs](#)

 [Data](#)

 [Tools](#)


 [Contact Us](#)

 [About Us](#)

 [Slack Channel](#)

 [Mastodon](#)

 [Bluesky](#)

 [x](#)