

# An Overview of Web3 TxPhish

BlockSec

<https://blocksec.com>



# Content

---

- Features of Web3 TxPhish
- Evolution of Web3 TxPhish
- Methods to Protect Users

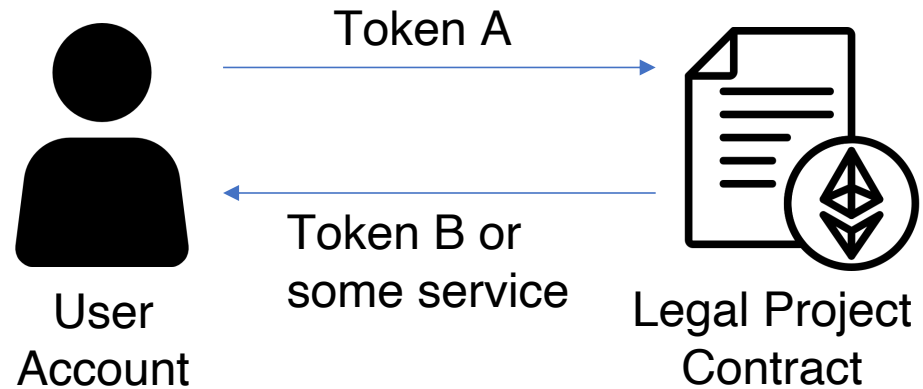
# Section 1

---

## Features of Web3 TxPhish

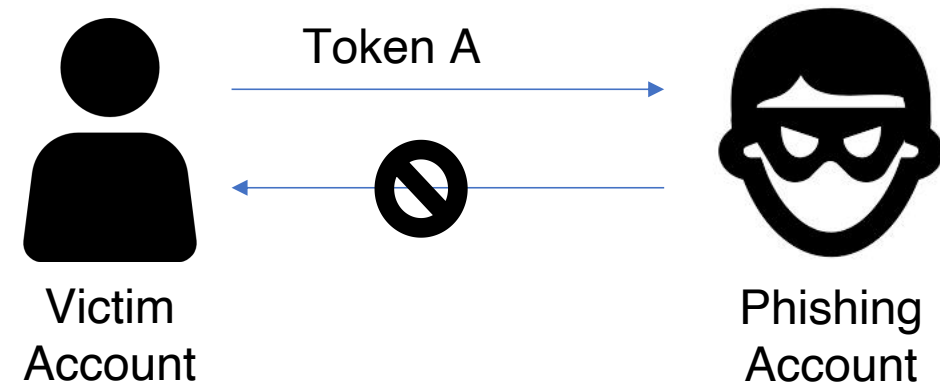
# What is Web3 TxPhish

Visiting official website



Sign Legal Transaction

Visiting Phishing website



Sign Phishing Transaction

# Type of Phishing Transactions

---

- Transfer Phishing
  - Directly transfer ETH
  - Invoke TransferFrom in the token contract
- Approval Phishing
  - Authorize a phishing account to control tokens

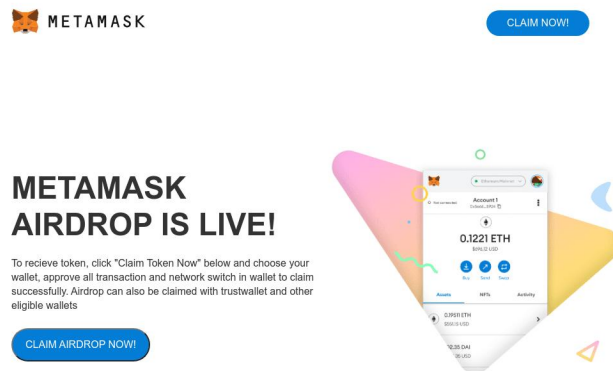
# Type of Phishing Transactions

---

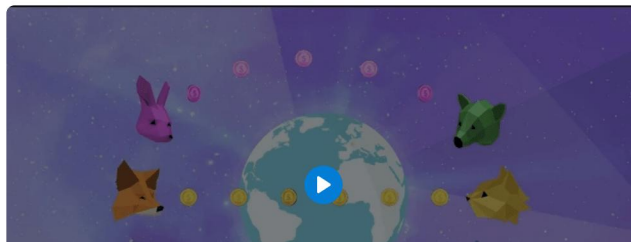
- Zero-dollar purchase phishing
  - Sign an order that sells tokens at a low price

# Example

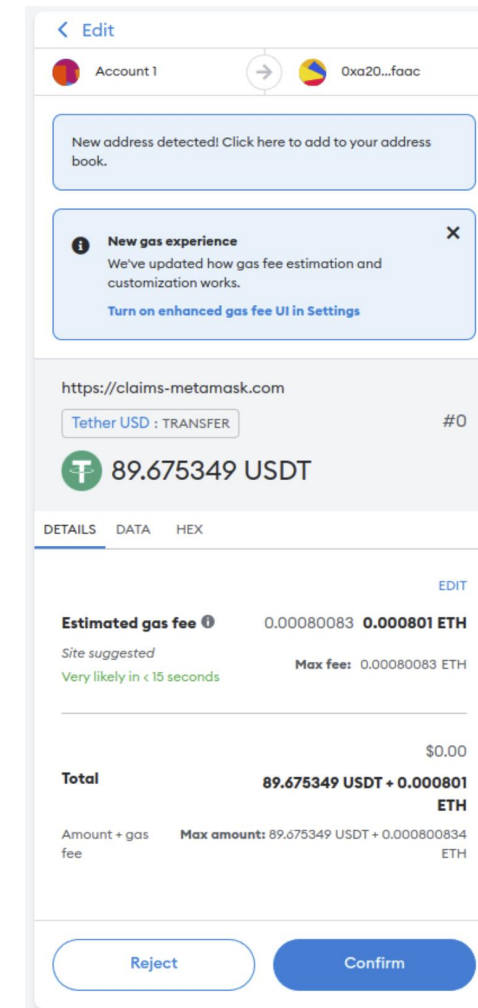
- [claims-metamask.com](https://claims-metamask.com) (fake MetaMask)



What is MetaMask?



Hi! Need help?



# Features of Phishing Websites

---

- Suspicious url
  - `claims-metamask.com`
- Suspicious interaction process
  - automatically ask you to connect wallets and sign transactions
- Suspicious transactions
  - try to withdraw all of your tokens



# Some Tips to Identify Phishing Websites

---

- click on other links in the page
- click the discord and twitter buttons

# Huge Loss of Web3 TxPhish

---

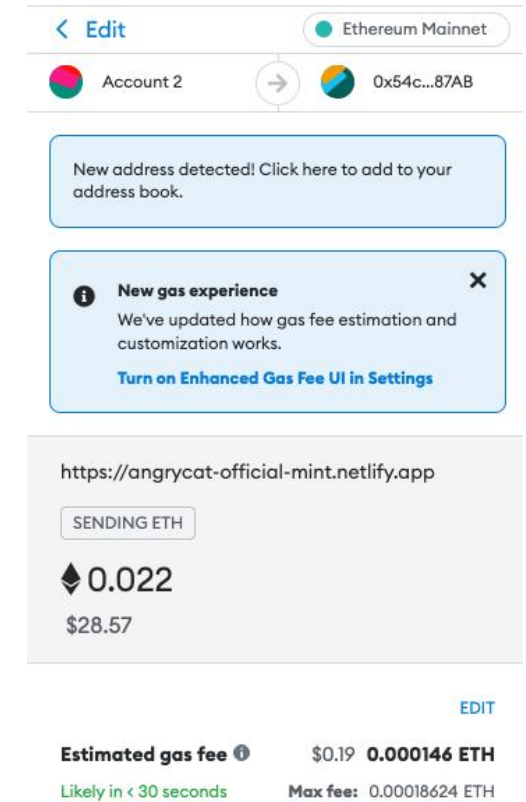
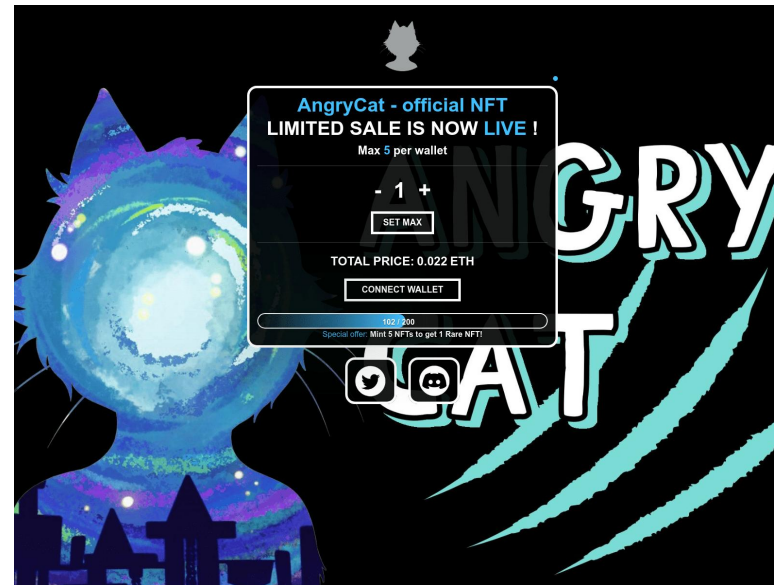
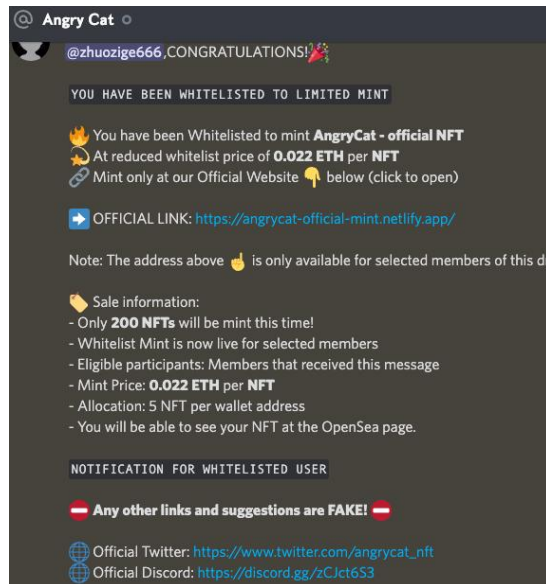
- From January 2023 to December 2023
- 12 large-scale phishing incidents
- Total Loss exceeding \$85 million

# Section 2

---

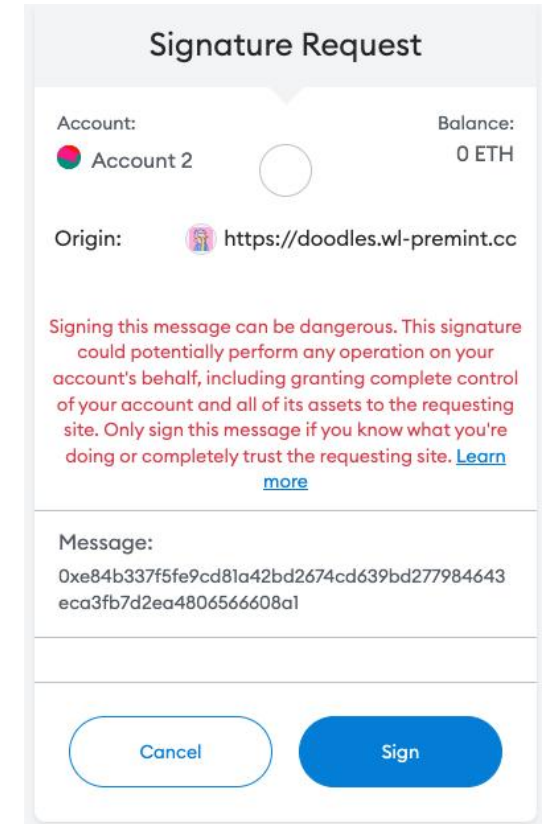
## Evolution of Web3 TxPhish

# First Type of TxPhish Campaign



# Evolution of Phishing Transactions

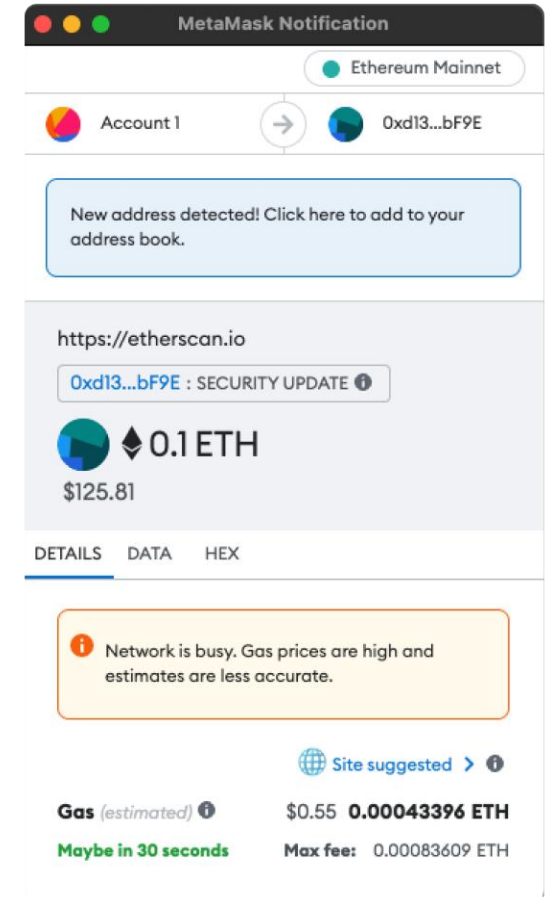
- Leverage eth\_sign
  - [directly ask users to sign the transaction hash](#)
  - now has been disabled by most of wallets



# Evolution of Phishing Transactions

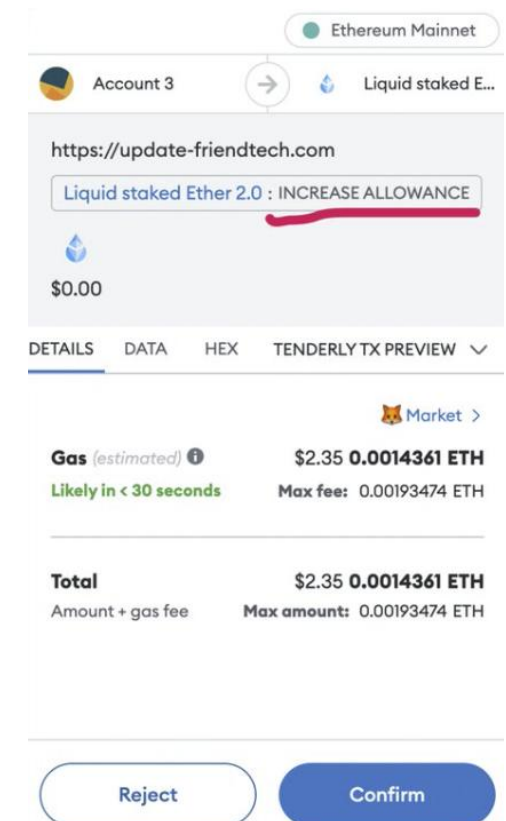
- Deploy Phishing contracts
  - Function names contain keywords
    - like security update, claim, airdrop

```
SecurityUpdates.sol ×
mainnet > SecurityUpdates.sol
1  pragma solidity ^0.8.7;
2
3  contract SecurityUpdates {
4      address private owner;
5      constructor() {
6          owner = msg.sender;
7      }
8      function withdraw() public payable {
9          require(msg.sender == owner, "Bro? Are you idiot?");
10         payable(msg.sender).transfer(address(this).balance);
11     }
12     function SecurityUpdate() public payable {
13         if (msg.value > 0) payable(owner).transfer(address(this).balance);
14     }
15 }
```




# Evolution of Phishing Transactions

- Leverage IncreaseAllowance mechanism
  - increases the current allowance by the given amount




# Evolution of Phishing Transactions

- Leverage permit mechanism
  - [allow others to spend tokens with a off-chain signature](#)

 Ethereum Mainnet  
Account 5

Balance  
0.135063 ETH


 <https://revokeme.cash>


**Signature request**

Only sign this message if you fully understand  
the content and trust the requesting site.


[Verify third-party details](#)

**Permit**

Owner:  Account 3

Spender:  0x000...0000

Value: 115847239543529489859238425  
834851258693125600000000000  
00000000

Nonce: 0 

Reject

Sign



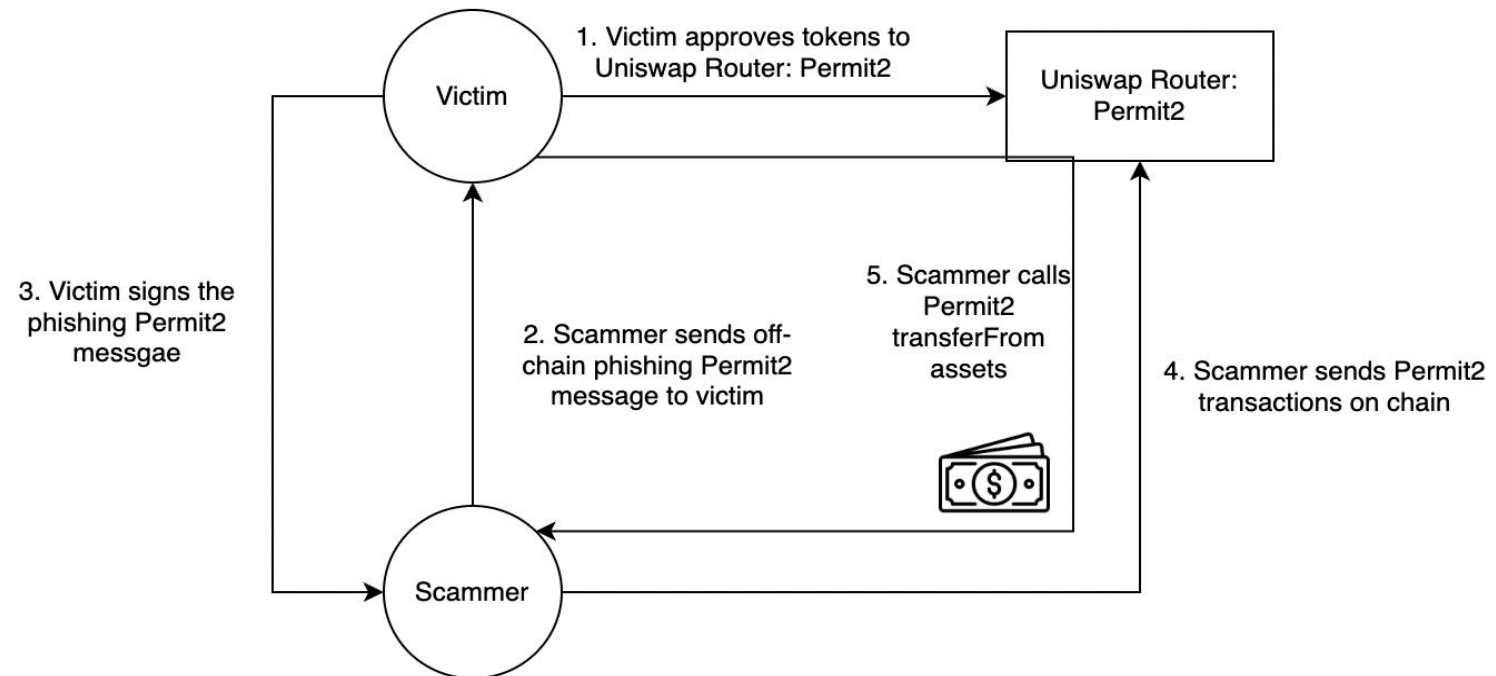
# Evolution of Phishing Transactions

---

- Leverage permit2 mechanism
  - Not all of tokens support permit
  - User can first approve tokens to Uniswap Router: Permit2
  - Then others can control the user's tokens via the permit signature


# Evolution of Phishing Transactions

- Explanation of permit2 phishing




# Evolution of Phishing Transactions




- Leverage seaport order protocol
  - lures a user to sign a order with a low price

 Ethereum Mainnet  
Account 3

Balance  
0.011282 ETH

 <https://opensea.io>

**Signature request**  
Only sign this message if you fully understand the content and trust the requesting site.  
[Verify contract details](#)

**Message**  
Offerer:  Account 3  
**Offer:**  
**0:**  
Item Type: 2  
Token:  0x035...d364  
IdentifierOrCriteria: 756 

REJECT

SIGN

# Evolution of Phishing Transactions

---

- Leverage seaport proxy upgrade protocol
  - OpenSea will register a proxy contract for new users
  - Scammer lures the victim to sign a malicious proxy upgrade transaction to change the proxy implementation
  - Then scammer get control of the proxy contracts and transfers all victim's NFTs

# Evolution of Phishing Transactions

- Leverage seaport proxy upgrade protocol
  - Example

Transactions	Internal Transactions	Token Transfers (ERC-20)	Contract	Events	Analytics	Comments
Latest 2 from a total of 2 transactions						
Transaction Hash	Method	Block	Age	From	To	Value Txn Fee
0x4f17c32570ed34b...	Multicall	17269390	1 hr 6 mins ago	PinkDrainer: Wallet 1	OwnableD...eProxy	0 ETH 0.00546176
0x938363b2496d08...	Upgrade To	17269389	1 hr 6 mins ago	*👉👉👉.eth	OwnableD...eProxy	0 ETH 0.00217195

**MetaSleuth** @MetaSleuth

Damn, these phishing people are really genius. 🧐 We find Pink-drainer used a new phishing scam to drain users' NFT assets.

11:55 AM · May 16, 2023 · 13.3K Views

9 20 43 8

Post your reply

**MetaSleuth** @MetaSleuth · May 16

When users create an account in Opensea, Opensea will create a proxy contract and let users approve their NFTs to the proxy contract.

2 9 1,267

**MetaSleuth** @MetaSleuth · May 16

Based on this feature, the pink-drainer drained the victim to sign an upgradeTo() function, to change the Opensea proxy implementation to the pink-drainer's contract. For example, [etherscan.io/tx/0x938363b2496d08...](https://etherscan.io/tx/0x938363b2496d08...)

11 3 10 2,274

# Evolution of TxPhish Campaign

---

- Adopt wallet drainer for large-scale deployment
  - phishing toolkits that automatically prompt users to connect their wallets, scan their tokens, and generate phishing transactions
  - aims to drain users' tokens entirely

# Evolution of TxPhish Campaign

- Example of wallet drainer from github

LICENSE	Add files via upload	2 months ago
README.md	Add files via upload	2 months ago
index.html	Add files via upload	2 months ago
settings.js	Add files via upload	2 months ago

README

MIT license

NOTE: This is just a demo script, to demonstrate how a real drainer works, it's easily detected, so try it out, and get a real drainer at :

- ✉ Contact: <https://t.me/Cryptohacker1402>
- 👤 Channel: <https://t.me/cryptohackers1402>

How to use the BNB Drainer:

1. Click on star and fork in the upper right corner
2. Open up the settings.js file and replace the marked words in the first row with your bsc adress
3. Go to <https://www.netlify.com/> and import your site via github
4. Open your drainer site: <https://officialchainlink.netlify.app/> and send the link to other people
5. That's it. The money goes to the bsc wallet adress you have linked in the settings.js file

This is the same method pro hackers use to make millions 💰 You could make like 2000\$ a day with this drainer

# Evolution of TxPhish Campaign

---

- The first well-known wallet drainer
  - [Monkey Drainer](#)
  - from 2022-10 to 2023-03
  - drained over \$24 million



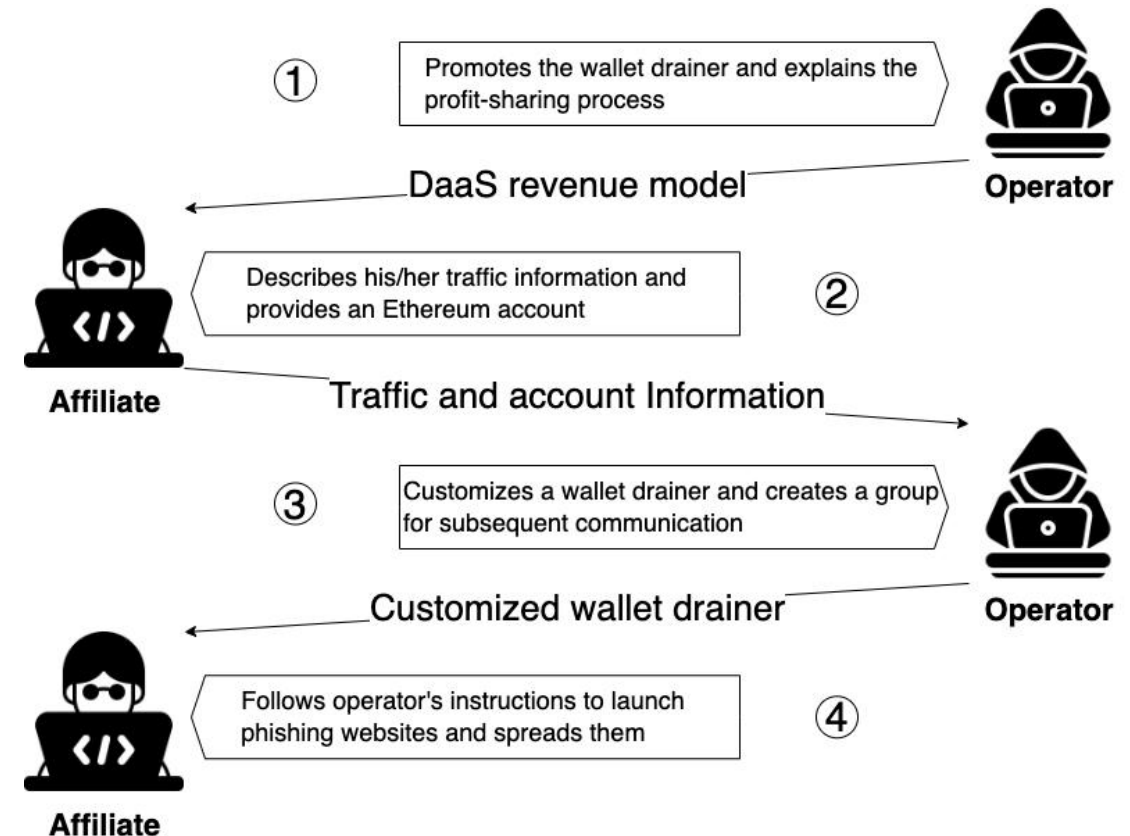
# Evolution of TxPhish Campaign

---

- Drainer-as-a-Service (DaaS)
  - one-time payment
  - a subscription fee and a percent of profits
  - [a percent of profits](#)

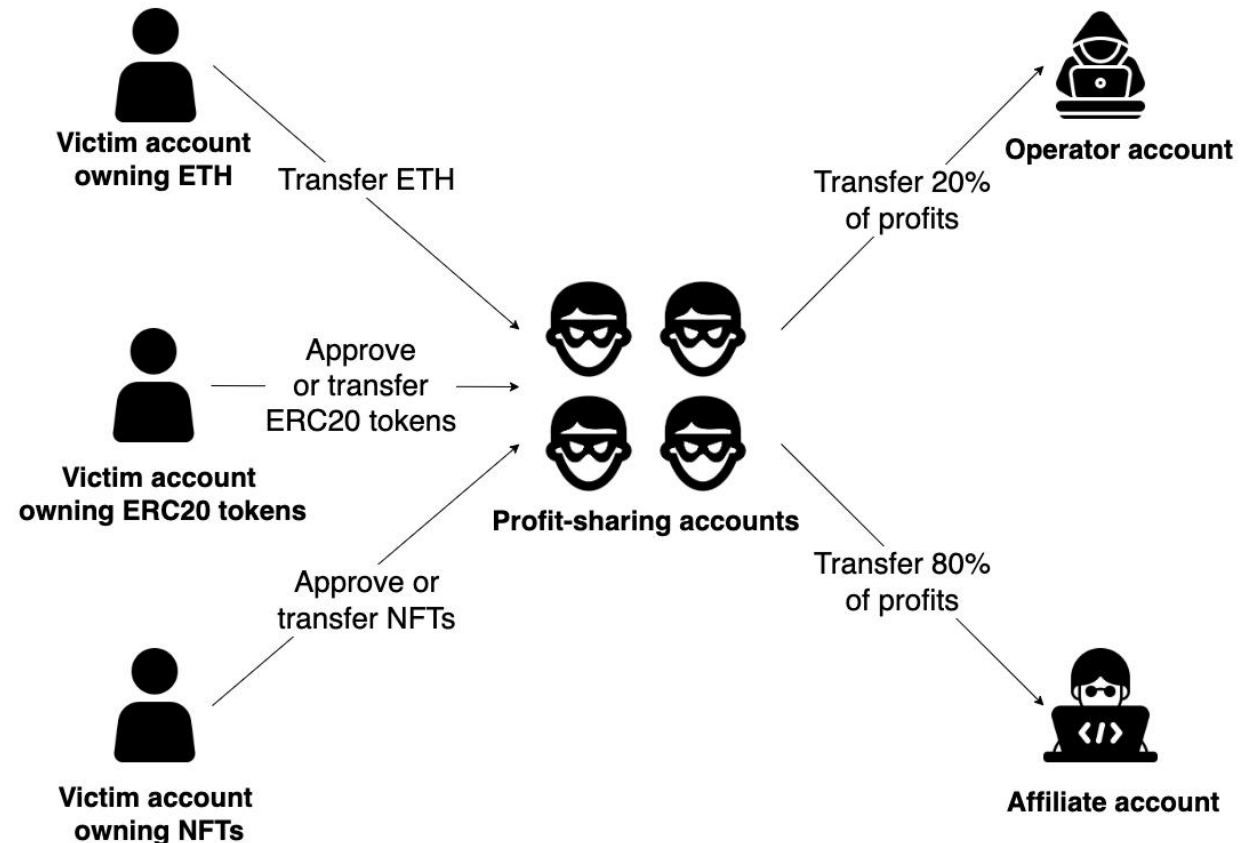
# Evolution of TxPhish Campaign

- Anatomy of DaaS



# Evolution of TxPhish Campaign

- Process of profit-sharing



# Evolution of TxPhish Campaign

---

- Example of profit-sharing transactions

---

[0xc16aDd8bA17ab81B27e930Da8a67848120565d8c](#) 

 [0x5E0102e6448b602FCd955FCFc7cEeA9a36E7e5f0](#) (Fake\_Phishing66332)  

└ Transfer 5.418472326788925 ETH From [Fake\\_Phishing66332](#) To [0x000000...F0675296](#)

└ Transfer 21.6738893071557 ETH From [Fake\\_Phishing66332](#) To [0x71F191...164677Ef](#)

---

# Evolution of TxPhish Campaign

---

- Example of profit-sharing transactions

---

◇ [inferno-drainer-4.eth](#) (Fake\_Phishing182232) 

📄 [0x0000A4998724E52F0886edFf693aCA33f9900000](#) (Fake\_Phishing186430)  

---

▶ From [0x0633A6...C318d722](#) To [0x27f9b0...5E71880f](#) For 1,595,313.21159427115369634 (\$448,064.45)  [Merit Circle...](#) (MC...)

▶ From [0x0633A6...C318d722](#) To [Fake\\_Phishing180395](#) For 398,828.302898567788424084 (\$112,016.11)  [Merit Circle...](#) (MC...)

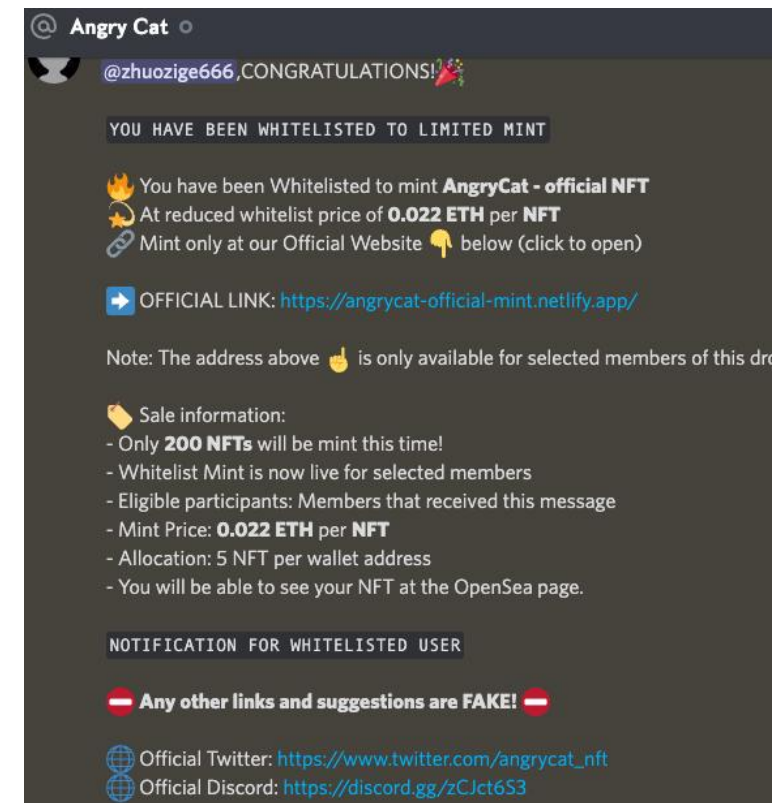
# Evolution of TxPhish Campaign

---

- Top DaaS Families (from 2023-03 to 2023-12)
  - [Inferno Drainer](#) \$26.5M
  - [Pink Drainer](#) \$6.7M
  - [Angel Drainer](#) \$4.5M

# Evolution of Promotion Channels

- Randomly send messages in Discord and twitter



# Evolution of Promotion Channels

- Create a fake twitter account





# Evolution of Promotion Channels

---

- Compromise discord servers, twitter accounts, or official websites
  - [Orbiter Finance](#) (steal discord token via malicious Javascript code)
  - [OpenAI CTO](#) (SIM swap scam)
  - [Vitalik Buterin](#) (SIM swap scam)
  - [Balancer](#) (DNS hijack)

# Evolution of Promotion Channels

- Example
  - Vitalik Buterin's twitter got hacked



# Summary of Web3 TxPhish

---

- Phishing Transactions become more complex
- Drainer-as-a-Service grows rapidly
- Scammers hack popular projects to promote phishing websites

# Section 3

---

## Methods to Protect Users

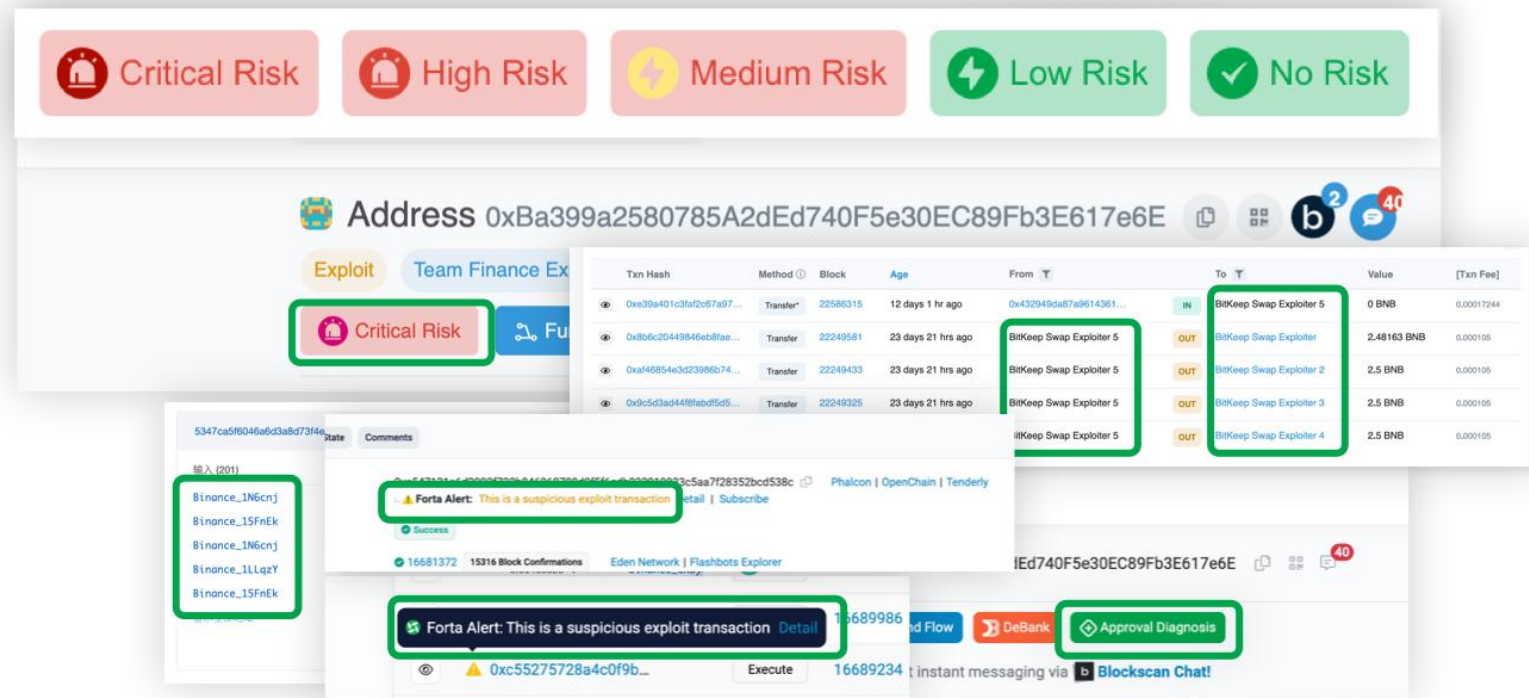
# Methods to Protect Users

---

- From user's perspective
  - check the website
  - check the account
  - simulate the transaction
  - **Never sign a transaction you don't understand**

# Methods to Protect Users

- [Install MetaSuite Plugin](#)

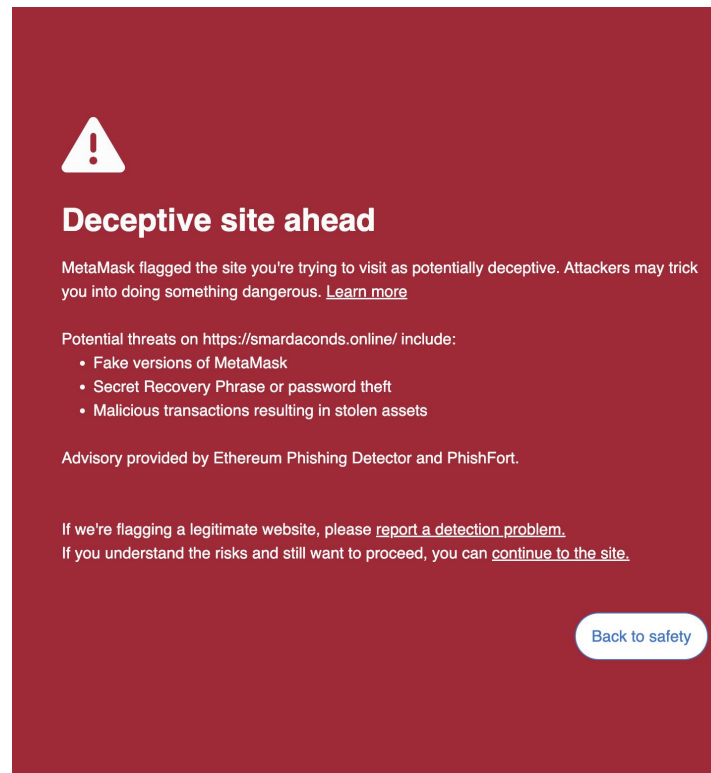


The screenshot displays the MetaSuite interface with several key components:

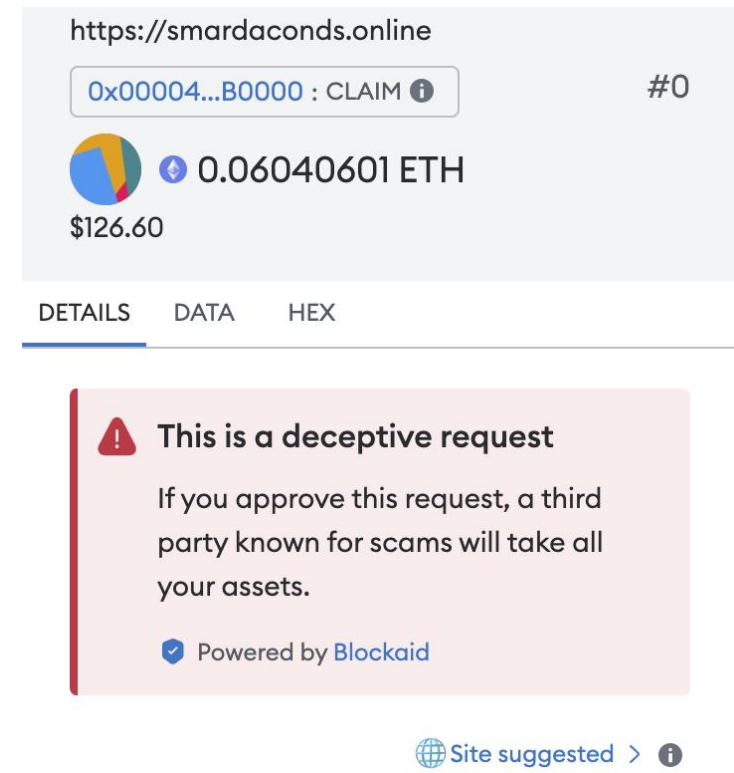
- Risk Level Indicators:** A row of five buttons at the top indicates risk levels: Critical Risk (red), High Risk (red), Medium Risk (yellow), Low Risk (green), and No Risk (green).
- Address:** The main address shown is 0xBa399a2580785A2dEd740F5e30EC89Fb3E617e6E.
- Transaction History Table:** A table listing transactions with columns for Txn Hash, Method, Block, Age, From, To, Value, and [Txn Fee]. Several transactions are highlighted with green boxes, showing they were sent to "BitKeep Swap Exploiter 5".
- Alerts:** A "Forta Alert" is displayed, stating "This is a suspicious exploit transaction".
- Navigation and Tools:** Various buttons and links are visible, including "Exploit", "Team Finance Ex", "Critical Risk", "Fu", "Approval Diagnosis", and "Blockschat Chat".

# Methods to Protect Users

- From Web3 wallet's perspective
  - maintain a blacklist of phishing websites and accounts



A red warning box with a white exclamation mark icon. The text reads: "Deceptive site ahead". Below this, it says "MetaMask flagged the site you're trying to visit as potentially deceptive. Attackers may trick you into doing something dangerous. [Learn more](#)". It then lists "Potential threats on https://smardaconds.online/ include:" followed by three bullet points: "Fake versions of MetaMask", "Secret Recovery Phrase or password theft", and "Malicious transactions resulting in stolen assets". At the bottom, it says "Advisory provided by Ethereum Phishing Detector and PhishFort." and "If we're flagging a legitimate website, please [report a detection problem](#). If you understand the risks and still want to proceed, you can [continue to the site](#)." A "Back to safety" button is at the bottom right.



A light blue box showing transaction details for "https://smardaconds.online". It displays a transaction ID "0x00004...B0000 : CLAIM" with a link icon, a balance of "0.06040601 ETH" and "\$126.60", and a "#0" label. Below are tabs for "DETAILS", "DATA", and "HEX". Below this is a pink box with a red exclamation mark icon and the text "This is a deceptive request". It says "If you approve this request, a third party known for scams will take all your assets." and "Powered by Blockaid". At the bottom right is a link "Site suggested > i".

# Methods to Protect Users

---

- Our work to detect phishing websites (CCS'23)
  - Step I: Crawl each new registered HTTPS certificate
  - Step II: Use some keywords to look for potential phishing website
  - Step III: Access the website and trigger the transaction signing behavior – just like a user clicks the link on the websites
  - Step IV: Use the transaction simulation to detect the phishing transaction



# Methods to Protect Users

---

- Our work has been merged into Forta Scam Detector



# Methods to Protect Users

---

- From Web3 wallet's perspective
  - add transaction simulation as a basic feature

# Methods to Protect Users

---

- From centralized exchange's perspective
  - block fund flows originating from phishing accounts



**MetaSleuth**   
@MetaSleuth

Remember the \$10,000,000 scammer?

Recently, our phishing detection system ([github.com/blocksecteam/w...](https://github.com/blocksecteam/w...)) has identified 270 phishing websites associated with 0x1661F1 (Fake\_Phishing66321). 🕵️ Our system shows the scammer is very skilled at using fake pages to lure victims to approve their tokens.

Upon analyzing the fund flow of Fake\_Phishing66321, we have discovered that it transferred 870 ETH to 0xDFFBF7, and 806.5 ETH has been deposited into [@SimpleSwap\\_io](https://simpleswap.io). Seems the bad guy loves SimpleSwap.

# Methods to Protect Users

- Examples by [MetaSleuth](#)



<https://blocksec.com>

Twitter:

[@BlockSecTeam](https://twitter.com/BlockSecTeam)

[@phalcon\\_xyz](https://twitter.com/phalcon_xyz)

[@MetaDockTeam](https://twitter.com/MetaDockTeam)

[@MetaSleuth](https://twitter.com/MetaSleuth)

[contact@blocksec.com](mailto:contact@blocksec.com)

WeChat Public Account

