Twitter Thread by ■ Mikko Ohtamaa

■ Mikko Ohtamaa
@moo9000

1/ ERC-20 token standard approve() has caused an unnecessary cost of $53.8M for #Ethereum and #DeFi users

This is bad. Continue reading why and how to avoid this in the future.

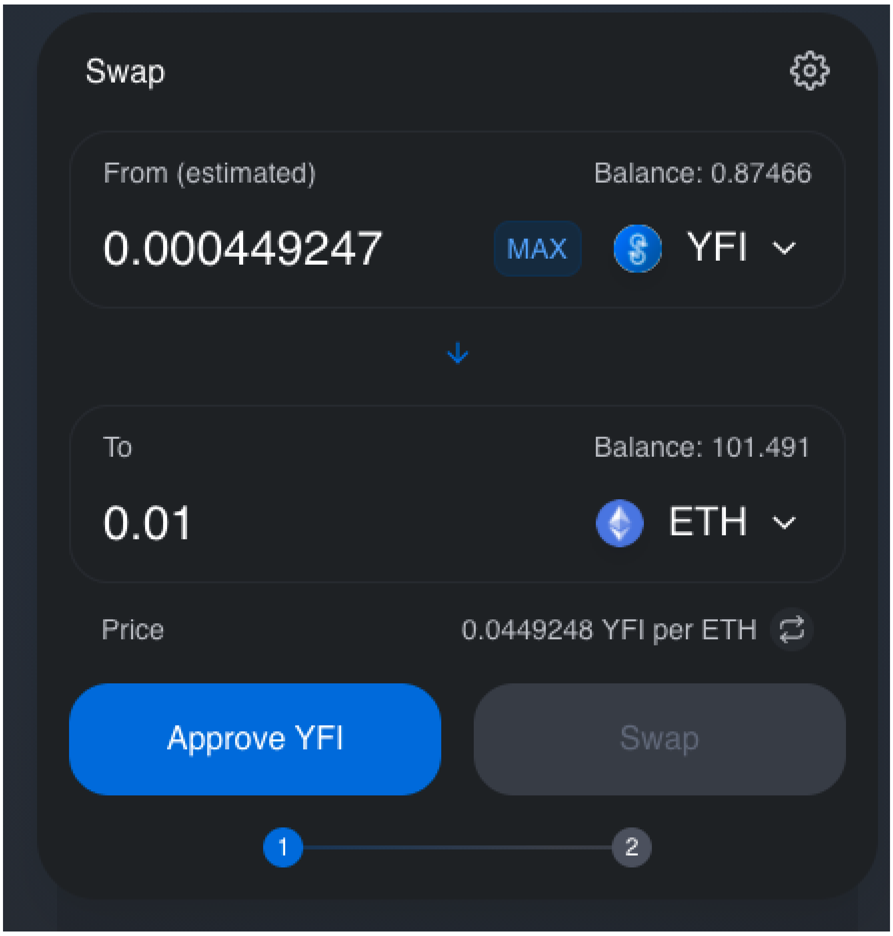
■■■

2/ Before you go all rage on the flaws of my analysis, please read the whole Twitter thread for disclaimers and caveats.

ARE YOU COMING TO BED?
I CAN'T. THIS IS IMPORTANT.
WHAT?
SOMEONE IS WRONG ON THE INTERNET.

3/ approve() is an unnecessary step of ERC-20 tokens when they interact with smart contracts.

You know this because when you do a Uniswap trade you need press two transaction buttons instead of one.



Swap

From (estimated) Balance: 0.87466

0.000449247 YFI

To Balance: 101.491

0.01 ETH

Price 0.0449248 YFI per ETH

4/ Why there is approve() - you can read the history in this Twitter thread

https://t.co/ZHXzPBbqJJ

> 1/ I just spend my Saturday morning on a call with a crypto fund explaining to them how #Ethereum ERC-20 token approve() function works
>
> I am too old for this shit. pic.twitter.com/7EYfOaRP5L
>
> — \U0001f42e Mikko Ohtamaa (@moo9000) August 29, 2020

5/ I queried all approve() transactions on Google BigQuery public dataset and calculated their ETH cost and then converted this to the USD with the current ETH price.

6/ These queries were made possible to awesome @EthereumETL team. They have created Google BigQuery dataset from real-time blockchain data. You can query over terabytes of Ethereum data FOR FREE.

7/ Here are instructions on how to execute your own queries:

https://t.co/YnK1p8u75n

(But links are outdated because Google Cloud has new UI)

8/ Total transaction on #Ethereum blockchain: 989,461,092

Woo! One billion transactions will be done in a few days!

9/ Total ERC-20 transfers(): 304,382,558

Includes only Externally Owned Accounts, EOAs.

10/ Total ERC-20 approves(): 14,921,106

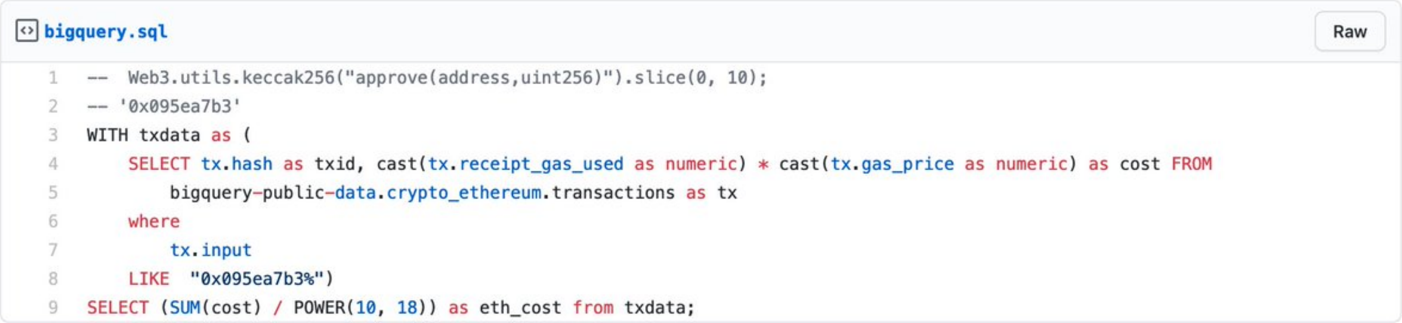
Includes only Externally Owned Accounts, EOAs.

11/ The gas cost ERC-20 approves():

41327.870139658684 ETH

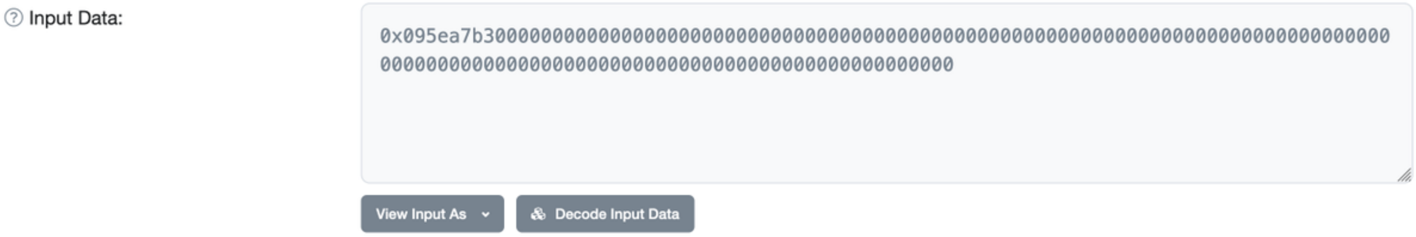
12/ Here is my query

https://t.co/FqNWBAxjwt



```sql
--  Web3.utils.keccak256("approve(address,uint256)").slice(0, 10);
-- '0x095ea7b3'
WITH txdata as (
    SELECT tx.hash as txid, cast(tx.receipt_gas_used as numeric) * cast(tx.gas_price as numeric) as cost FROM
        bigquery-public-data.crypto_ethereum.transactions as tx
    where
        tx.input
    LIKE  "0x095ea7b3%")
SELECT (SUM(cost) / POWER(10, 18)) as eth_cost from txdata;
```

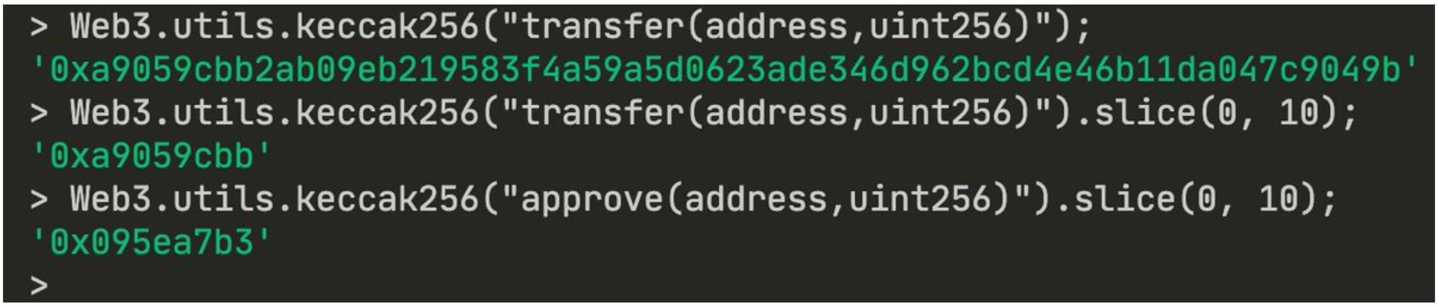
13/ You can query Ethereum transactions that call a particular smart contract function by the 4-byte signature of the function that is 1st parameter of tx data field.



Input Data:

```
0x095ea7b3000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
```

14/ Binary function signatures are 4-bytes keccak256() hash of the @solidity_lang signature string. Here is an example:



```
> Web3.utils.keccak256("transfer(address,uint256)");
'0xa9059cbb2ab09eb219583f4a59a5d0623ade346d962bcd4e46b11da047c9049b'
> Web3.utils.keccak256("transfer(address,uint256)").slice(0, 10);
'0xa9059cbb'
> Web3.utils.keccak256("approve(address,uint256)").slice(0, 10);
'0x095ea7b3'
>
```

15/ If you are a @solidity_lang or Vyper developer, consider ditching ERC-20 and include alternative token standard in your next token.

Alternatives for ERC-20 include:

ERC-777
ERC-667
ERC-827
ERC-223
(did I miss any?)

16/ Most of the new token standards, like ERC-777 are backwards compatible and work with ERC-20 enabled centralised exchanges.

Centralised exchanges do not need to do anything to support these new, better, token standards that make smart contract and #DeFi interactions safer.

17/ Or let's put it this way...

Every time someone creates a new ERC-20 token, hundreds of thousands of dollars die.

Let's actively demanding non-ERC 20 tokens from developers.

And if that does not work I suggest we start punching ERC-20 developers to face over the internet

18/ Newer token standards may "increase the attack surface", but in practice, this has not been a problem for high-quality #DeFi projects since 2018 or so.

18/ Note that raw approve() cost calculation comparison to newer token standard is not 1:1.

Newer token standards need some similar mechanism to pass user data as the part of the transaction, but this cost is lower compared to additional approve() tx.

Prove me wrong.

19/ FIN

CC @dmihal @FrancescoRenziA @abcoathup

Now I am going to climb the mountain, get fresh air and visit monkies