Breaking Web3
Exploitation Techniques
For Cryptocurrency Hacking

author: six
Agenda

1. Introduction to web3 hacking
2. Eth Transactions
3. Tools for interacting with Eth EVM
4. Remix IDE, ERC20, Truffle, Ganace
5. Substrate/Rust, Extrinsic, PolkadotJS
6. ChaosPallet
7. From web2 to web3 -> Node keys
8. When you don't see it aka social engineering in web3
9. Spilling the Tea
six ~ > whoami
> Web3 hacker
> Founder of CCTF
> Co-founder of QRUCIAL
> Polkadot Head Ambassador of Eastern Europe
> Just does what he loves.
What the web3 aka...

Reverse engineering buzzwords & the hype

proof of deposit  ink!  swap  remix ide  decentralized

token  crypto  dApp  coin  crypto wallet

bridge  web3  ink!  Javascript

metaverse  solidity  tx  substrate

blockchain  smart contract  signed  wrapped coin

javascript  🤯
Basic stuff: Eth tx
Tools to pwn all the Eth

+ [github.com/crytic/awesome-ethereum-security](https://github.com/crytic/awesome-ethereum-security)
Starter technique

anyone can kill your contract #6995

Open
devops199 opened this issue 22 hours ago · 12 comments

devops199 commented 22 hours ago · edited

I accidentally killed it.

https://etherscan.io/address/0x863df6bfa4469f3ead0be8f9f2aae51c91a907b4

Multisig issue: https://github.com/openethereum/parity-ethereum/issues/6995
TXDATA Replay Attack

Function: mintWithReceipt(address recipient, uint256 amount, uint256 uuid, uint8 v, bytes32 r, bytes32 s) ***

MethodID: 0x05b084df

[0]: 0000000000000000000000000000000070e6c8b2596f828f9ce4790df0e6519265b045f14
[1]: 00000000000000000000000000000000000000000000008dafa88e340e3ed80000
[2]: 000000000000000000000000000000000000000000000000000000000000000017c6fbc6f
[3]: 0000000000000000000000000000000000000000000000000000000000000000001b
[4]: 2a17615cf385bc09ffefbf966e0085f65256833764a88907a5ac76e2a395976
[5]: 5347c8812f0bfaa5df77e832f14d618ce535e8908136bb87ec699d8c1b6d64

wBanano hack TXs:

https://polygonscan.com/tx/0xbcfc3f1192d63a0d240995619b8896c406d1ba6fa7c2fc81503057d61c98bba41
https://bscscan.com/tx/0x60c3ae26d1a1d2b525a425aadadbde30bf7efdc09a125086cc7aab9b347daf684
Demo - Solidity

- Interact a smart contract (explorer)
- ERC20 in Remix
- Call from another smart contract
- Exploit reentrancy
Demo

- Ethkeygen

- Generate ECDSA keys

- Reference: wBan Hack
Hacking Substrate - Why?

Features of Substrate:
- Libp2p Networking
- Byzantine Fault Tolerance (BFT) Consensus
- The WebAssembly developer tool
- Immediate project execution on Polkadot
- A cross-platform database storage abstraction
- Seamless communication with all cloud nodes
- Parachains -> sovereign blockchains.
- Relay chain -> Governance mechanism, Parachain auctions, NPoS
  - Shared state/sec between the Relay Chain and Parachains.
- Parathread -> pay-as-you-go model.
- Governance -> in practice, through PolkadotJS !!!! $$$ ! :)))
Attack from inside: chaosscope

RUNTIME
(APIs, Execution Environment)

EXECUTIVE MODE
(Orchestration)

RUNTIME MODULES
(from FRAME or custom)

SYSTEM MODULE
(Core Types, Utilis, Events)

SUPPORT LIBRARY

https://github.com/paritytech/chaoscope
Okey, but how do we hack it?

- Chaos Pallet
- Build and run Substrate -> substrate.io
- How-NOT-to-build-a-pallet (ideas)
- Lets speak about the ink! pallets too!

How not to build a pallet -> https://github.com/apopiak/how-not-to-build-a-pallet
Attacking: ink! smart contracts

- Run locally by 'substrate-contracts-node'

- Connect to the node with PolkadotJS
  https://polkadot.js.org/apps/

- Execute smart contracts using ContractsUI
  https://github.com/paritytech/contracts-ui
  https://contracts-ui.substrate.io/

ink! VS Solidity
https://ink.substrate.io/ink-vs-solidity
Note on what web3 is NOT meant to be:

- SELL BITCOIN
- WAIT TILL DUMP
- BUY AGAIN WHEN CHEAP
Life of a standard MetaMask user

+ Social Engineering
Not always the brightest attackers -> Awareness <3
Spilling the tea: advanced techniques

- Proxy contract hacking (same tools)
- Reverse engineering (octopus helps)
- Consensus and network attacks
- Flash Loan
- Logic bugs 😢
- MEV
- Vulns like Uncle Maker

--> https://eprint.iacr.org/2022/1020

To Be Continued
Contact

Matrix: @hexff:matrix.org
Twitter: @DaveTheSix
CCTF room: #CCTF:matrix.org
Audit request: hello@qrucial.io