Mitigating Smart Contract Attacks In Real-World

Chaofan Shou
Fuzzland / UC Berkeley
shou@fuzz.land

Smart Contract Attacks

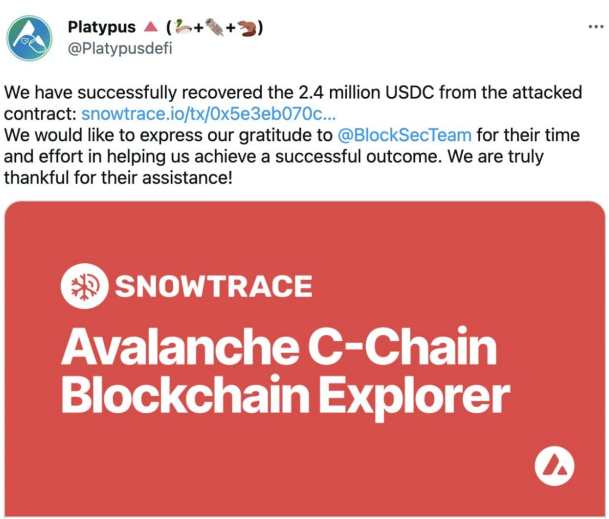
Has Already Led to $100M+ Loss in 2024

UwU Lend Hack – $21M Loss
Sonne Finance Hack - $20M Loss
PrismaFi Hack - $11M Loss
LiFi Hack - $10M Loss
WooFi Hack - $8M Loss
VeloCore Hack - $6.9M Loss
MIM Spell Hack - $6.5M Loss
Gamma Hack - $6.3M Loss
SSS Token Hack - $4.8M Loss

I heard whitehats are <u>frontrunning attacks</u> to stop hacker

Attack Frontrun

Clone attackers' exploits before the transaction is included in the block. Send the cloned transaction with higher gas fee.



Sounds easy. I will build my frontrun bot.
But, how effective is it?

Attack Frontrun

In 2024

$0/100M Rescued

in

0/82 Incidents

NGMI???
What’s the root cause?

#1

99.1% Hacks are Sent to {Private RPC} or On {Chains Without Mempool}

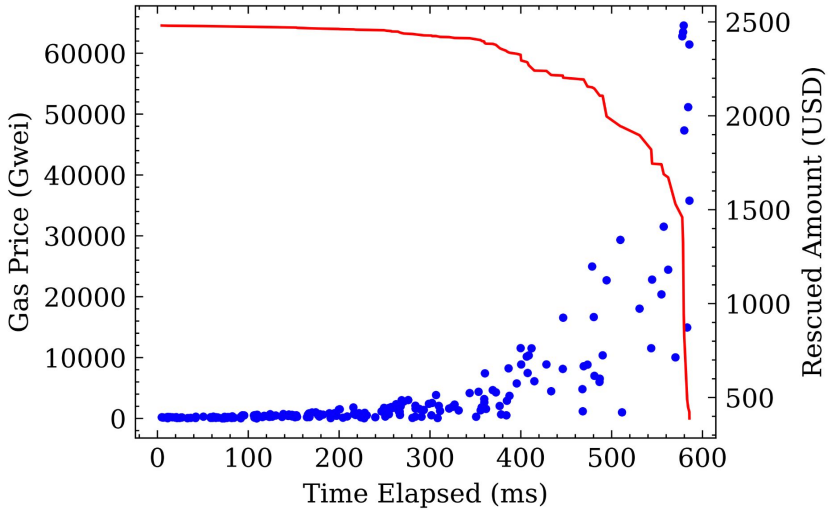
No Frontrun Bot Can See These Hacks

#2

Competitions Make the Bots

{Burn 95%+ Rescued Funds}

Our Experiment Shows Frontrun Bots Compete with Each Others By Raising Gas. Some Bots Even Make the Attempt No Longer Profitable.



Attackers Commonly Conduct {Prework} Before Attacks

Attack

0xa17fdb80472...	Make Flash L...	19687887	97 days ago	Hedgey Finance Expl...	OUT	0xC793113F...4241bF2b3	0 ETH	0.0073006
0x03c152d73b...	0x14c0bbe5	19687872	97 days ago	Hedgey Finance Expl...	OUT	0xC793113F...4241bF2b3	0 ETH	0.00746719
0xf6dfdd6b152...	0x60806040	19687858	97 days ago	Hedgey Finance Expl...	OUT	Contract Creation	0 ETH	0.02828623

Prework: Attack Contract Deployment + Configuration

Attackers May Conduct {Additional Work} After Attacks to Pick Up Remaining Funds

Second Attack

0x606a49471e...	0x85b4a4fc	20318983	9 days ago	LI.FI Exploiter 2	OUT	0xB9556993...B97344924	0 ETH	0.02150702
0x78d876f4fc3...	Transfer	20318983	9 days ago	LI.FI Exploiter 2	OUT	0xea72074d...dFA5d1685	0.1 ETH	0.00026588
0x9004e33d66...	0x60808060	20318983	9 days ago	LI.FI Exploiter 2	OUT	Contract Creation	0 ETH	0.00429941
0x001cbced9c...	0x60808060	20318983	9 days ago	LI.FI Exploiter 2	OUT	Contract Creation	0 ETH	0.0023842
0x967b2460ba...	Transfer	20318980	9 days ago	0x588F83D1...4A2a50114	IN	LI.FI Exploiter 2	0 ETH	0.000294
0x86fe6933f03...	0x85b4a4fc	20318976	9 days ago	LI.FI Exploiter 2	OUT	0x3c81476D...37d4F3176	0 ETH	0.02006093

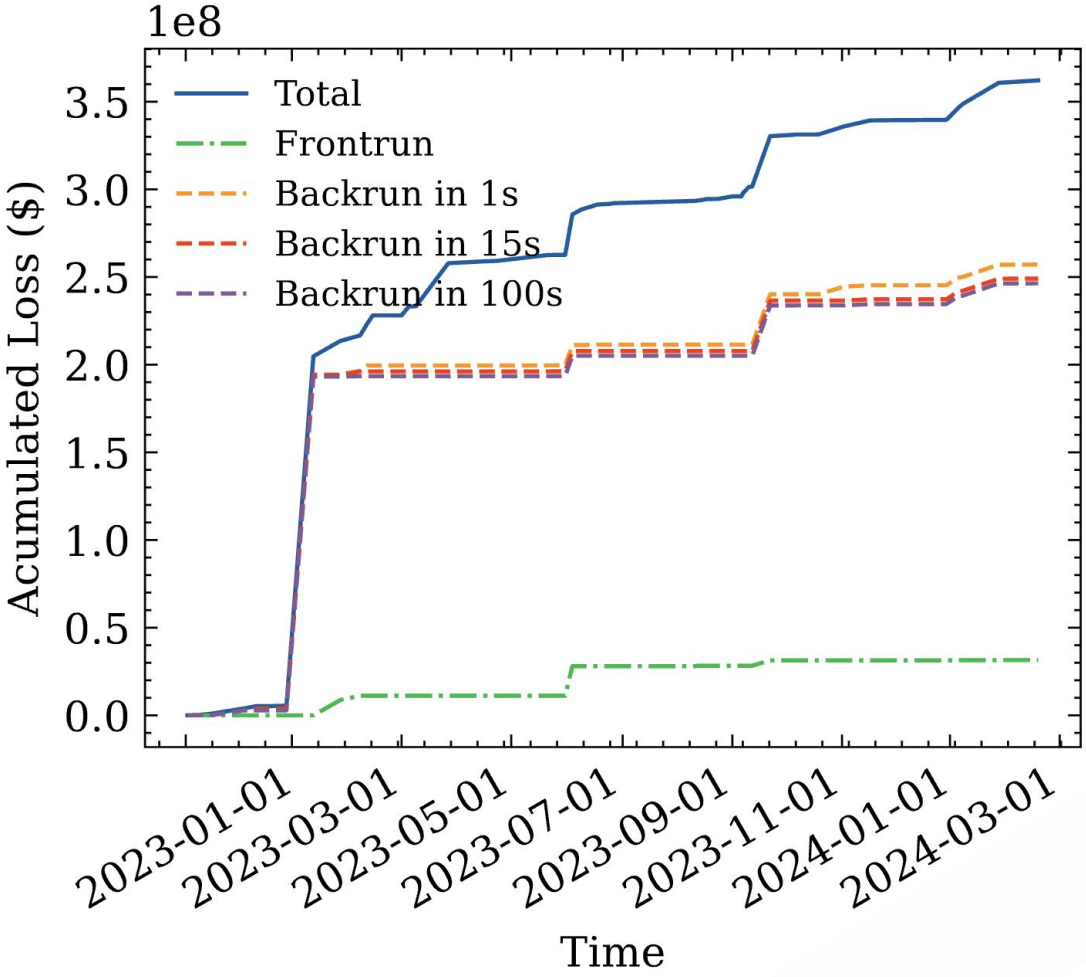
Attack

What If....

- **We can Hijack:** figure out attack from prework

What If....

- **We can Hijack:** figure out attack from prework
- **We can Backrun:** hack remaining funds at risk after seeing the initial attack



Suppose we have an oracle....

57% fund can be rescued by backrun.

26% fund can be rescued by hijacking.

Attack Hijacking

Convert Prework to Attack

```
function 0xcb0d9b88(uint256 v0, bytes v1) public {
    ...
    require(msg.sender == owner);
    require(tx.origin == msg.sender);
    require(0x60b0a6.... == keccak256(tx.origin));
    ...
    addr.flashloan(this, s19, v0, v0, 0);
    ...
    ret, res = stringToAddress(v1);
    require(owner == res);
    ...
}
```

Find arguments by program repair techniques (e.g., bruteforcing)

Entry function for Onyx Protocol exploit

Attack Hijacking

Convert Prework to Attack

```
function 0xcb0d9b88(uint256 v0, bytes v1) public {
    ...
    require(msg.sender == owner);
    require(tx.origin == msg.sender);
    require(0x60b0a6.... == keccak256(tx.origin));
    ...
    addr.flashloan(this, s19, v0, v0, 0);
    ...
    ret, res = stringToAddress(v1);
    require(owner == res);
    ...
}
```

Entry function for Onyx Protocol exploit

Bypass authentications by using heuristics and forced execution.

We can flip any `require` in the exploit!

Attack Backrun

Adapt Attack to New Targets

```
1 → CALL   Paraswap: Augustus V6.uniswapV3SwapCallback Calldata (amount0Delta=-33,000,000,000, amount1Delta=6,463,332,789,527,457,985, data=(lo
2 → CALL   Uniswap V3: OPSEC.swap Calldata (recipient=[Receiver]MEV Bot: 0x698...DD
1 → STATICCALL   Wrapped Ether (WETH).balanceOf Calldata ([Receiver]MEV Bot: 0x698...D
1 → CALL   Wrapped Ether (WETH).withdraw Calldata (wad=6,463,332,789,527,457,985)
1 → CALL   value: 6.450406123948403069 MEV Builder: 0x22...d5f.fallback(raw data)
```

```
0x000000000000000000000000 6980a47bee930a4584b09ee79ebe46484fbdbd
d000000000000000000000000 0cc396f558aae5200bb0abb23225accafca31e
270000000000000000000000 77e06c9eccf2e797fd462a92b6d7642ef85b0a
440000000000000000000000c02aaa39b223fe8d 0e5c4f27ead9083c756c
c200000000000000000000000000000000000000 000000000000000000000b
b880000000000000000000006a7eff1e2c55ad6eb91bebb5ded49257f3fed
980000000000000000000000c02aaa39b223fe8d0a0e5c4f27ead9083c756c
c200000000000000000000000000000000000000 000000000000000000027
10
```

Paraswap Initial Attack Trace

Find new victims by finding related addresses (e.g., same bytecode)

Attack Backrun

Adapt Attack to New Targets

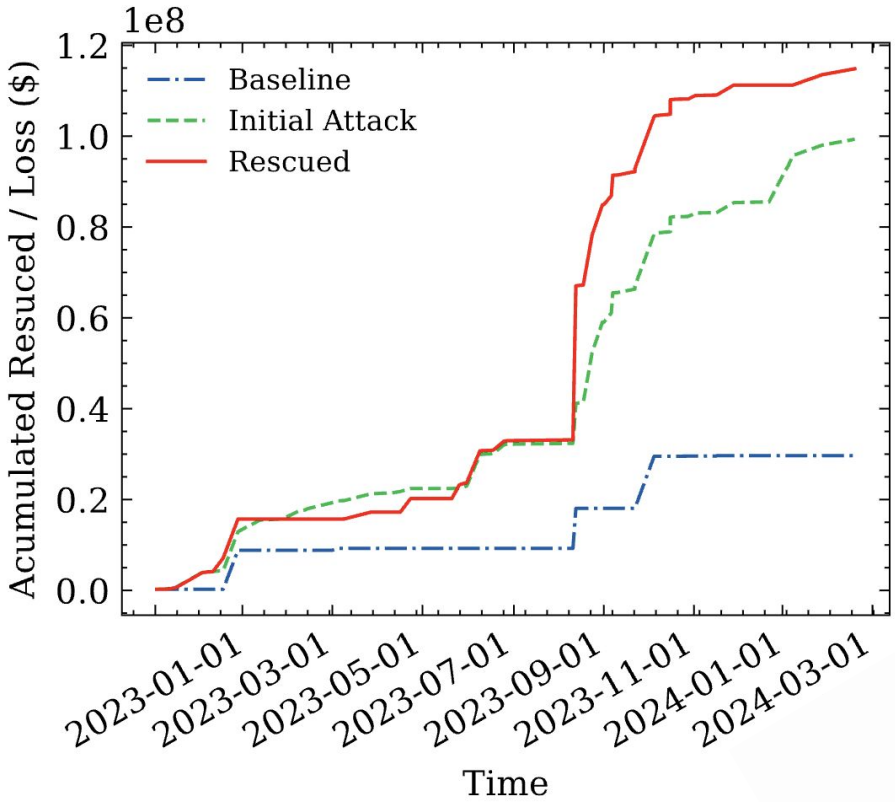
```
1 → CALL   Paraswap: Augustus V6.uniswapV3SwapCallback Calldata (amount0Delta=-33,000,000,000, amount1Delta=6,463,332,789,527,457,985, data=(lo
2 → CALL   Uniswap V3: OPSEC.swap Calldata (recipient=[Receiver]MEV Bot: 0x698...DD
1 → STATICCALL   Wrapped Ether (WETH).balanceOf Calldata ([Receiver]MEV Bot: 0x698...D
1 → CALL   Wrapped Ether (WETH).withdraw Calldata (wad=6,463,332,789,527,457,985)
1 → CALL   value: 6.450406123948403069 MEV Builder: 0x22...d5f.fallback(raw data)
```

```
0x0000000000000000000000006980a47bee930a4584b09ee79ebe46484fbdbd
d0000000000000000000000000cc396f558aae5200bb0abb23225accafca31e
270000000000000000000000077e06c9eccf2e797fd462a92b6d7642ef85b0a
440000000000000000000000c02aaa39b223fe8d0a0e5c4f27ead9083c756c
c200000000000000000000000000000000000000000000000000000000000b
b880000000000000000000006a7eff1e2c355ad6eb91bebb5ded49257f3fed
980000000000000000000000c02aaa39b223fe8d0a0e5c4f27ead9083c756c
c200000000000000000000000000000000000000000000000000000000027
10
```

Paraswap Initial Attack Trace

Find valid arguments for new victims by program repair techniques (e.g., bruteforcing)

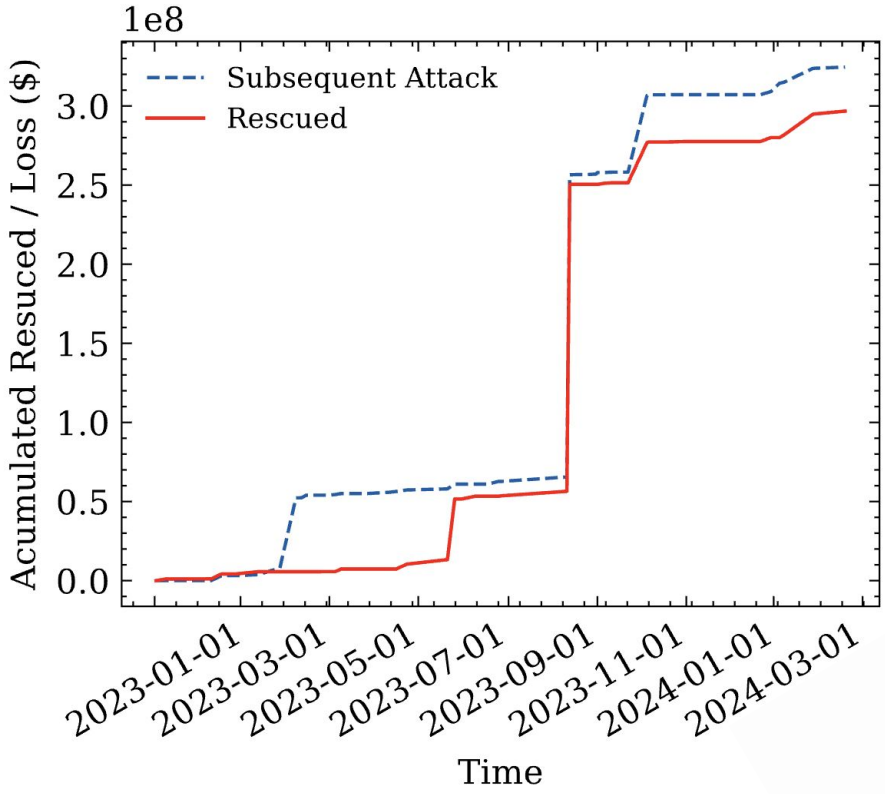
Hijack Backtesting



We can even rescue 20% more funds than those stolen in initial attacks!

We can find more optimal values (e.g., flashloan amount) for attacks than the attackers.

Backrun Backtesting



We can rescue 94% funds stolen in subsequent attacks!

In Real-World....

{$15M}

Rescued

- Backrun for Sonne Finance ($6.8M)
- Hijack for Sonne Finance ($3.4M)
- Hijack for Nexera ($2.8M)
- Backrun for [redacted] AA Wallet ($1M)
- Backrun for [redacted] Swap ($0.5M)
- Hijack for Teller V2 ($0.4M)
- Backrun for Dough Finance ($0.3M)
- ...

Why didn’t you prevent LiFi and UwU Lend hack last month?

This work has been deployed in Fuzzland real-time smart contract analysis solutions.

fuzzland

This work is made possible with following companies and organizations, but does not reflect their views.

DEFIHACKLABS Let's make web3 more secure! manifold DEDAUB VERILOG ironblocks Chainbound blocknative GOPLUS

This work is also supported by UC Berkeley SkyLab Sponsors but does not reflect their views.

accenture AMD anyscale Google IBM intel MOHAMED BIN ZAYED UNIVERSITY OF ARTIFICIAL INTELLIGENCE SAMSUNG SDS SAP VMWARE UNIVERSITY RESEARCH FUND vmware

Pepe images in the slides are from Frens

You can also backrun attacks today!

https://github.com/fuzzland/ityfuzz

Chaofan Shou
Fuzzland / UC Berkeley
shou@fuzz.land
For Business Inquiry: jeff@fuzz.land