Confusum Contractum:
Confused Deputy Vulnerabilities in Ethereum Smart Contracts

Fabio Gritti, Nicola Ruaro, Robert McLaughlin, Priyanka Bose, Dipanjan Das
Ilya Grishchenko, Christopher Kruegel, and Giovanni Vigna

University of California, Santa Barbara
In the past 2 years ~2 Billion USD have been stolen from blockchain applications!
In the past 2 years ~2 Billion USD have been stolen from blockchain applications!

Poly Network Suffers Record-Breaking $600.3 Million Hack

Axie Infinity’s Ronin Network Suffers $625M Exploit

Hackers steal around $200 million from crypto lender Euler Finance

Nomad crypto bridge loses $200 million in ‘chaotic’ hack
In the past 2 years ~2 Billion USD have been stolen from DeFi!

Poly Network Suffers Record-Breaking $600.3 Million Hack

Interoperability protocol Poly Network has suffered an exploit today. The attacker has made off with at least $625M Exploit

Axie Infinity’s Ronin Network Suffers $625M Exploit

Hackers steal around $200 million from crypto lender Euler Finance

Nomad crypto bridge loses $200 million in ‘chaotic’ hack
Decentralized Finance (DeFi)

- Financial products running on the blockchain:
  - Investments
  - Lending
  - Credit
  - Insurance
  - …
Decentralized Finance (DeFi)
Decentralized Finance (DeFi)

- DeFi apps are implemented with Smart Contracts
Smart Contracts 101

- **Smart Contract** is a program that runs on top of a VM (EVM) and implements the business logic of an application.
Smart Contracts 101

0x1A2a1c938CE3eC39b6D47113c7955bAa9DD454F2
Smart Contracts 101

0x1A2a1c938CE3eC39b6D47113c7955bAa9DD454F2

Public Functions

F1
F2
F3
F4
Smart Contracts 101

From:
0x76cf6361F21dBA36eAa88649FEeBD9F18d94647

Calldata:
0xdeadbeef68656c6f20776f726c64

Public Functions

0xA2a1c938CE3eC39b6D47113c7955bAa9DD454F2
Smart Contracts 101

0x76cf6361F21dB7A36eAa88649FEEeBD9F18d94647

TX

From:
0x76cf6361F21dB7A36eAa88649FEEeBD9F18d94647

Calldata:
0xdeadbeef68656c6f20776f726c64

Function ID

0x1A2a1c938CE3eC39b6D47113c7955bAa9DD454F2

Public Functions

F1
F2
F3
F4
Smart Contracts 101

From: 0x76cf6361F21dB7A36eAa88649FEEeBD9F18d94647
Calldata: 0xdeadbeef68656c6c6f20776f726c64

Function Arguments

0x1A2a1c938CE3eC39b6D47113c7955bAa9DD454F2

Public Functions

F1
F2
F3
F4
Smart Contracts 101

From: 0x76cf6361F21dB7A36eAa88649FEEeBD9F18d94647
Calldata: F1("hello world")

0x1A2a1c938CE3eC39b6D47113c7955bAa9DD454F2
Smart Contracts 101

0x76cf6361F21dB7A36eAa88649FEEeBD9F18d94647

0x1A2a1c938CE3eC39b6D47113c7955bAa9DD454F2

TX

PUSH
GAS
... CALL

F1

F2

F3

F4

Public Functions
Smart Contracts 101

0x76cf6361F21dB7A36eAa88649FEEBD9F18d94647

0x1A2a1c938CE3eC39b6D47113c7955bAa9DD454F2

Public Functions

F1 F2 F3 F4

TX

PUSH GAS CALL iTX
Smart Contracts 101

PUSH GAS ... CALL

F1

iTX1

iTX2

iTX3

iTX4

iTX5

iTX6

iTX7
Smart Contracts 101
Confused Deputy Vulnerabilities in Ethereum Smart Contracts
Confused Deputy

- Bug class introduced by Norman Hardy in 1988
Confused Deputy
Confused Deputy

No relationship of trust between P1 and P2
Confused Deputy

P3: trusted middleman
Confused Deputy

P1 can trick P3 to perform the action A on P2
Confused Deputy

P1 → P3 → P2

Attacker  Confused Deputy  Target
Confused Deputy

Confused Contract

P1 → P3 → P2
Attacker Confused Deputy Target

Attacker
Confused Contract
Contract Target
Confused Contract

What does privilege escalation look like?
Confused Contract

Attacker A → TX → C1
Confused Contract

Origin: Attacker
msg.sender: Attacker

Attacker -> TX -> C1
Confused Contract

Attacker

Origin: Attacker
msg.sender: Attacker

TX

C1

PUSH GAS
CALL
Confused Contract

Attacker

Origin: Attacker
msg.sender: Attacker

TX

C1

F1

iTX

C2

FX

Origin: Attacker
msg.sender: C1
Confused Contract

Origin: Attacker
msg.sender: Attacker

Attacker

TX

F1

C1

iTX

FX

Origin: Attacker
msg.sender: C1

C2
**Confused Contract**

`msg.sender` is used by contracts for different kind of operations
Confused Contract

```
Origin: Attacker
msg.sender: C1

if (msg.sender == C1)
    { do_something }
```
Confused Contract

Attacker

TX

C1

iTX

C2

Origin: Attacker

msg.sender: C1

... transfer(msg.sender, dst)

...
Confused Contract

If C1 allows everyone to execute a CALL, an attacker can "borrow" its identity for privileges escalation in another contract!
If C1 allows everyone to execute a CALL, an attacker can “borrow” its identity for **privileges escalation** in another contract!
Identify Confused Contract Vulnerabilities

Confused Contract

Contract Target

PUSH GAS

CALL

F1

FX
Identify Confused Contract Vulnerabilities

TX → Confused Contract → iTX → Contract Target
Identify Confused Contract Vulnerabilities

Confused Contract

Contract Target

PUSH GAS
... CALL
Identify Confused Contract
Identify Confused Contract

Smart Contract Binary

10101001 101001
01010110 110000
10101001
01010110
10101001
01010110
10100101
11000010
11000010
Identify Confused Contract

Smart Contract Binary
Identify Confused Contract

Smart Contract Binary

PUSH
GAS
...
CALL

F1

F2

F3

F4
Identify Confused Contract

CALL Inspection

Smart Contract Binary

PUSH GAS

…

CALL

10101001 101001
0101010 110000
10101001
01010110

F1

F2

F3

F4

10101001 101001
0101010 110000
10101001
01010110

PUSH GAS

…

CALL
CALL Inspection

- Directed Symbolic execution to understand if a CALL instruction is reachable by an attacker
- If reachable, can attacker control the destination of this CALL?
- If yes, we found a confused contract
Identify Confused Contract Vulnerabilities

Confused Contract

PUSH GAS

CALL

Contract Target

TX

F1

iTX

FX
How to find a Contract Target?
Confused Contract

Millions of smart contracts!
Confused Contract

Millions of smart contracts!
Confused Contract

Millions of smart contracts!
Confused Contract

$A = ?$

Millions of smart contracts!
Let’s simplify
Identify Contract Target

- An instance of a Contract Target: ERC20 token contracts

ERC20 Contract (Token Contract)
Identify Contract Target

- An instance of a Contract Target: ERC20 token contracts

- ERC20 Contracts hold a “state” on behalf of other contracts: their tokens balance

ERC20 Contract (Token Contract)
Identify Contract Target

- An instance of a Contract Target: ERC20 token contracts
- ERC20 Contracts hold a “state” on behalf of other contracts: their **tokens balance**
- This state is automatically identifiable! (check details on the paper)
Exploit

Origin: Attacker
msg.sender: Attacker

Attacker → TX → Confused Contract
Exploit

**Attacker**

**Origin:** Attacker

**msg.sender:** Attacker

**Confused Contract**

**PUSH GAS**

... **CALL**
Exploit

Origin: Attacker
msg.sender: Attacker

TX
F1

Origin: Attacker
msg.sender: ConfusedCont.

PUSH
GAS
... CALL

ERC20
Contract

Confused
Contract

Transfer
Exploit

Attacker

PUSH GAS
CALL

Origin: Attacker
msg.sender: Attacker

Confused Contract

Transfer

Origin: Attacker
msg.sender: ConfusedCont.

ERC20 Contract

... transfer(msg.sender, Attacker) ...
Evaluation

- **2,000,000+** smart contracts
  - Deployed between December 2020 → December 2022

- **529** potential Confused Contracts
Evaluation

- **2,000,000+** smart contracts
  - Deployed between December 2020 → December 2022
- **529** potential Confused Contracts
  - 84 warnings Confused Contract + Contract Target
Evaluation

- **2,000,000+ smart contracts**
  - Deployed between December 2020 → December 2022

- **529 potential Confused Contracts**
  - 84 warnings Confused Contract + Contract Target

- We automatically generated exploits for a total value of more than $1,000,000!
Identify Targets

ERC20 Contract Target

Contract Target

Contract Target

Contract Target

Contract Target

Contract Target

Contract Target

Contract Target

Contract Target

Contract Target
Conclusion

- **Confused Contract** is a class of vulnerability inspired by the confused deputy bug class, but applied in the context of Blockchain

- Attackers can “borrow” the identity of another contract to perform actions on their behalf

- We estimated more than a million dollar of possible financial damage
Thanks!

✉️ degrigis@ucsb.edu
🐦 @degrigis