Confusum Contractum:

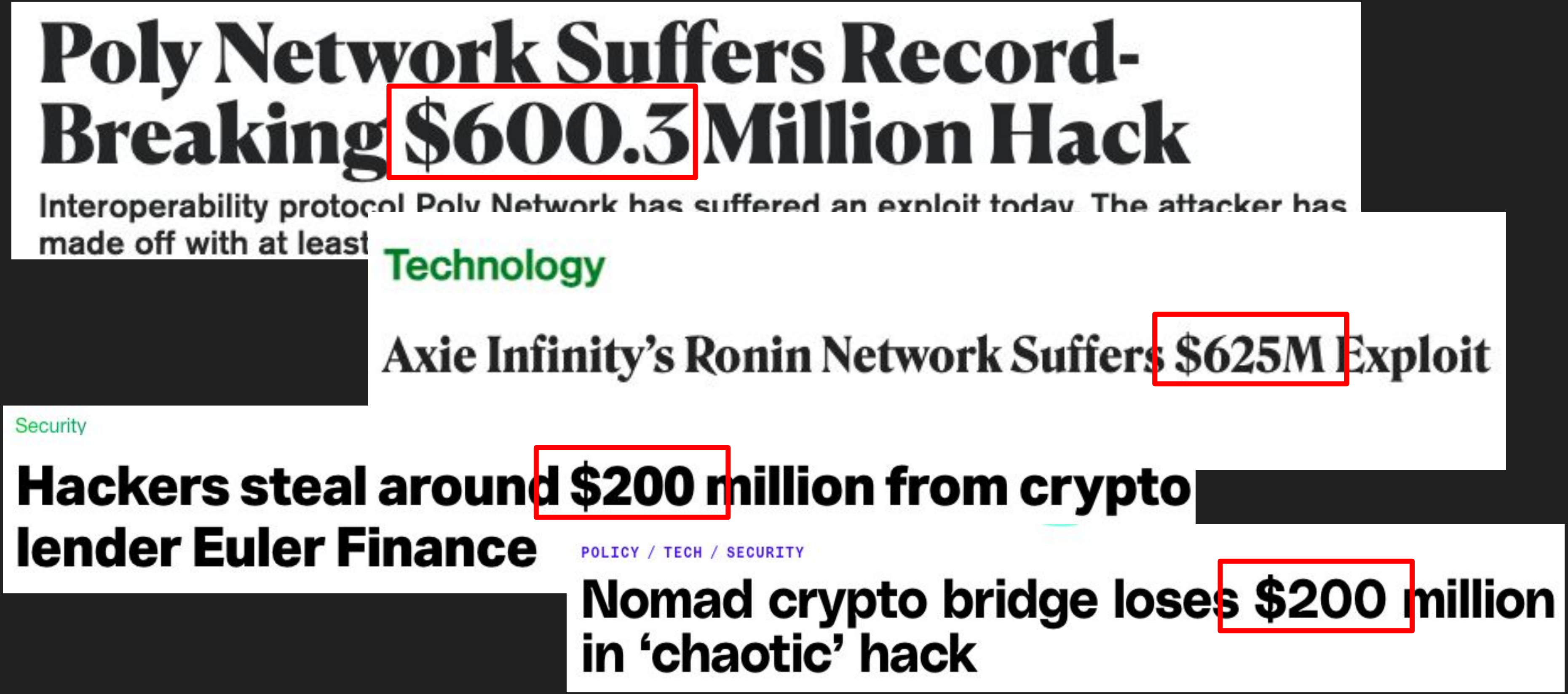
Confused Deputy Vulnerabilities in Ethereum Smart Contracts

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

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

Technology

Axie Infinity’s Ronin Network Suffers $625M Exploit

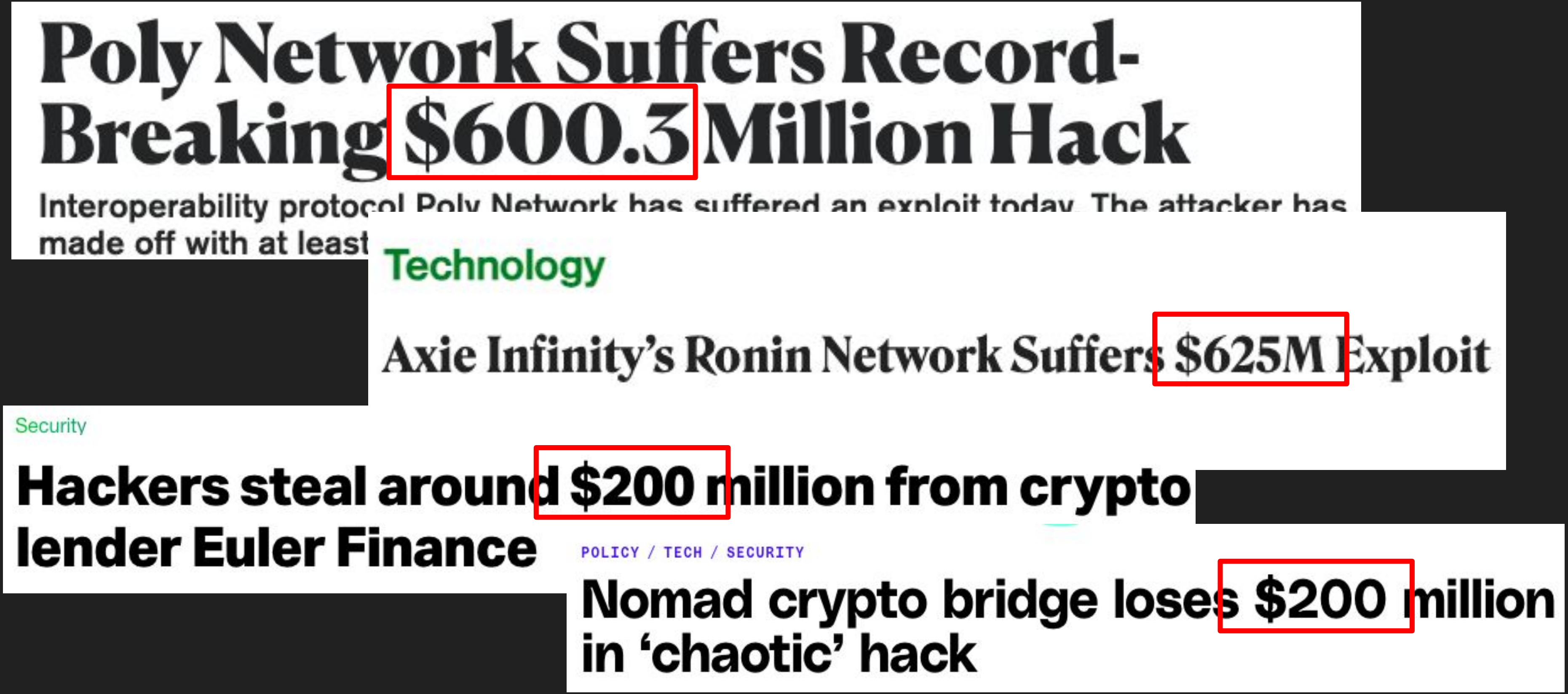
Security

Hackers steal around $200 million from crypto lender Euler Finance

POLICY / TECH / SECURITY

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

Technology

Axie Infinity's Ronin Network Suffers $625M Exploit

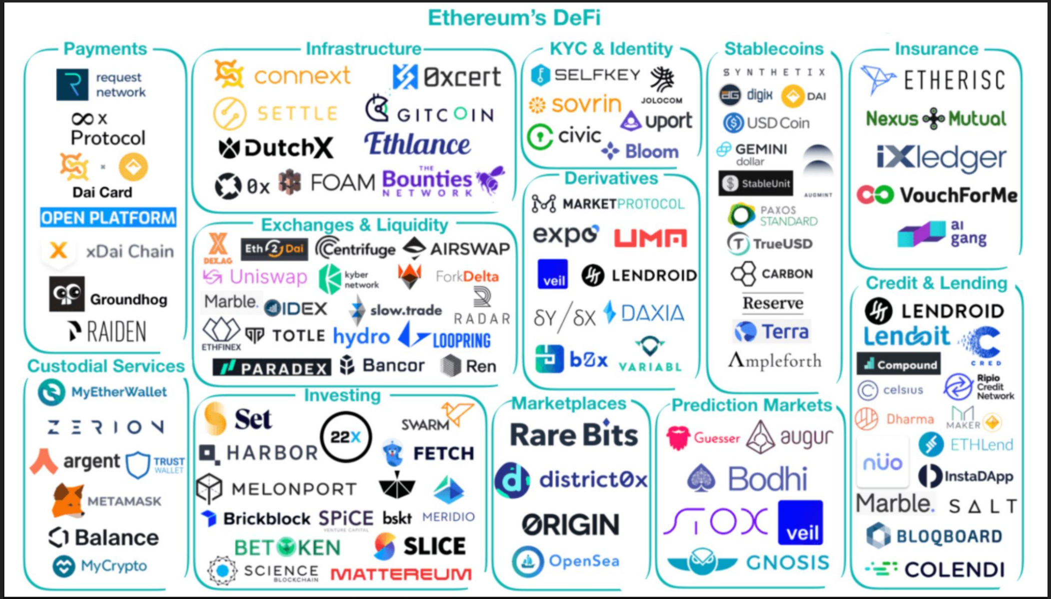
Security

Hackers steal around $200 million from crypto lender Euler Finance

POLICY / TECH / SECURITY

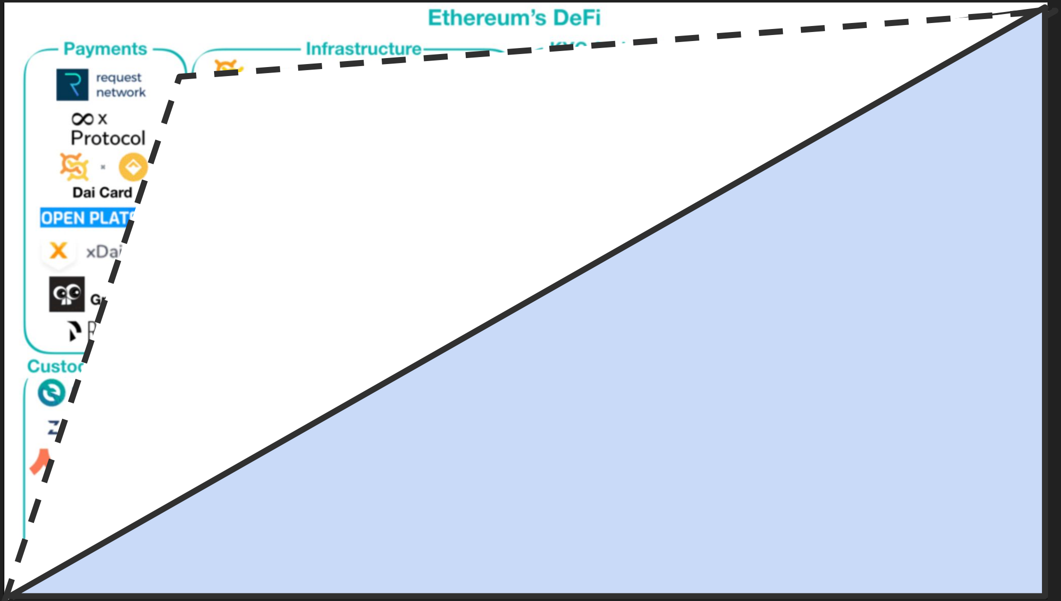
Nomad crypto bridge loses $200 million in 'chaotic' hack

Decentralized Finance (DeFi)

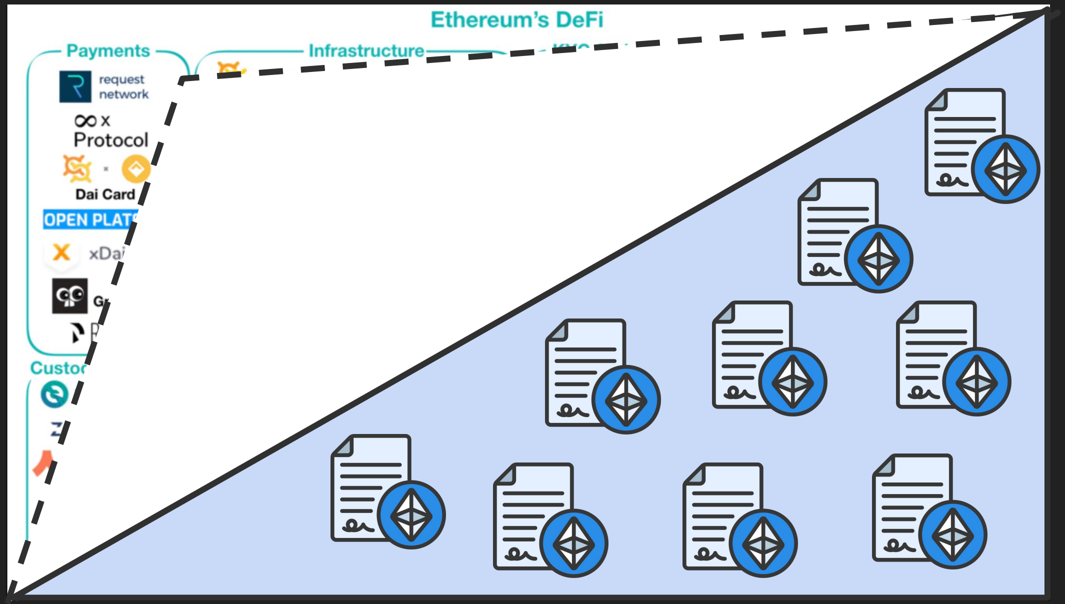


- Financial products running on the blockchain:
  - Investments
  - Lending
  - Credit
  - Insurance
  - …

Decentralized Finance (DeFi)

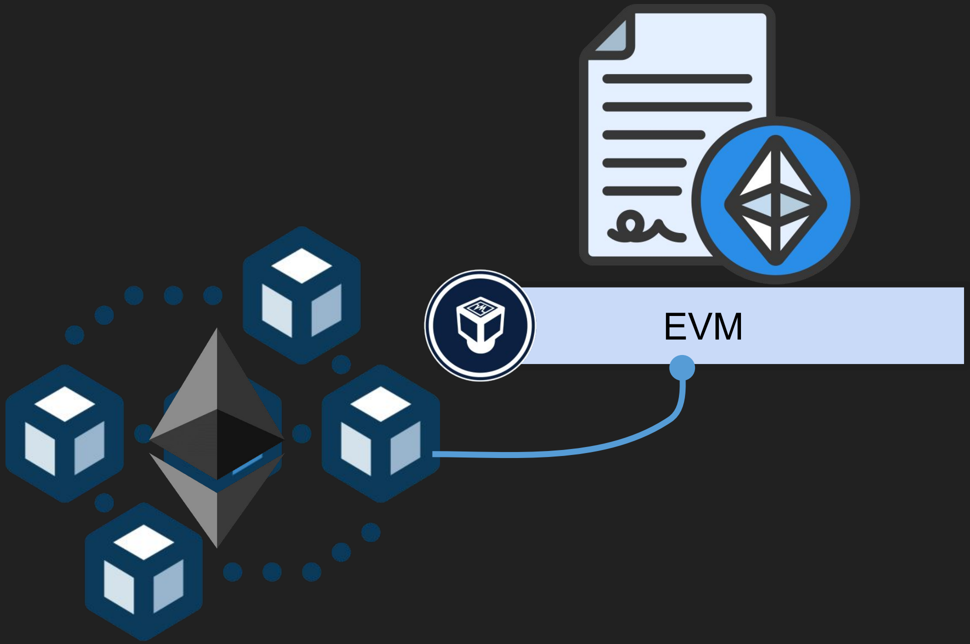


Decentralized Finance (DeFi)



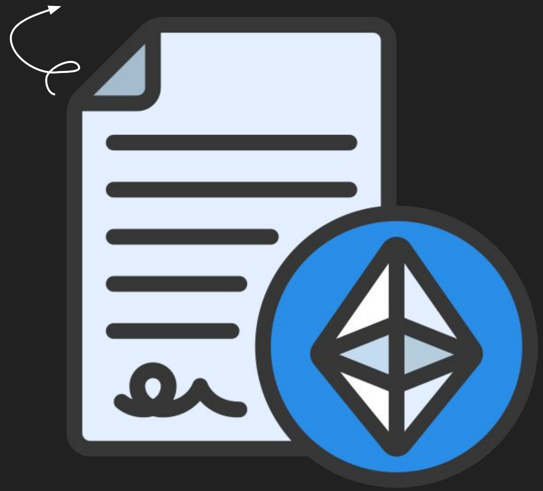
- DeFi apps are implemented with Smart Contracts

Smart Contracts 101



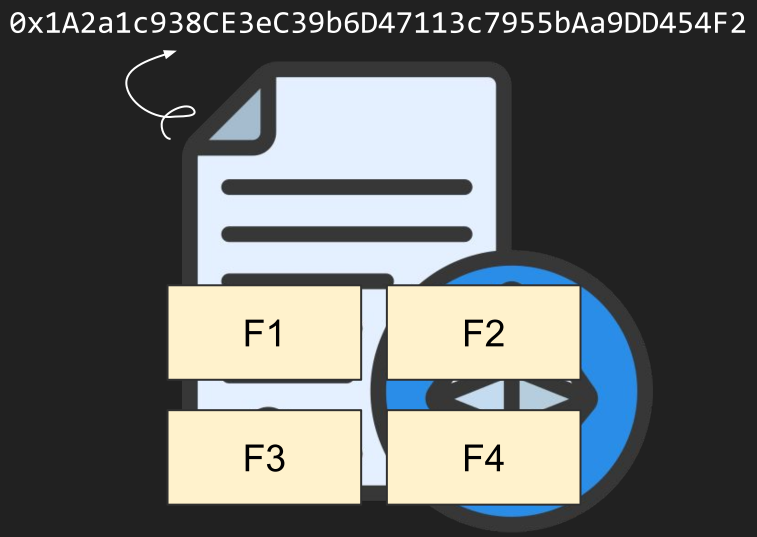
- Smart Contract is a program that runs on top of a VM (EVM) and implements the business logic of an application

Smart Contracts 101



0x1A2a1c938CE3eC39b6D47113c7955bAa9DD454F2

Smart Contracts 101



0x1A2a1c938CE3eC39b6D47113c7955bAa9DD454F2

F1

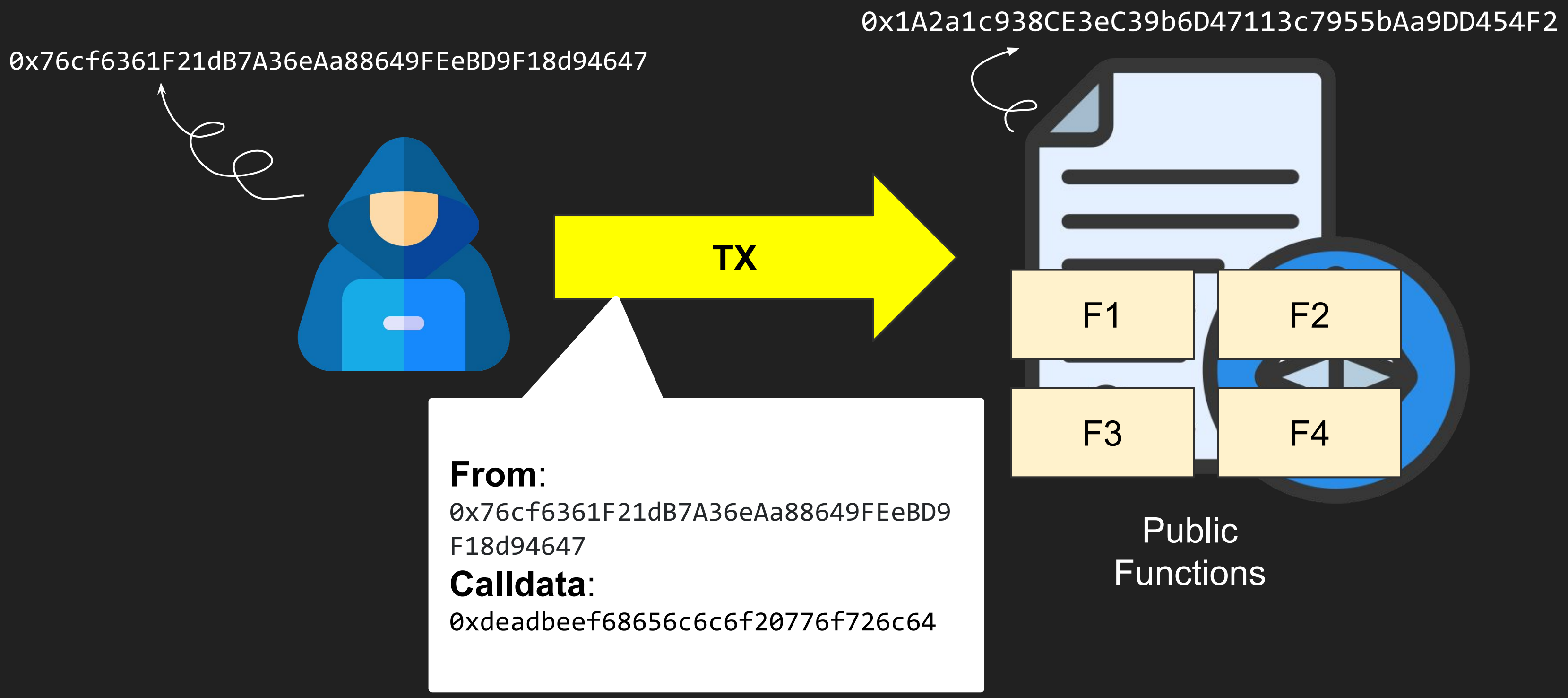
F2

F3

F4

Public Functions

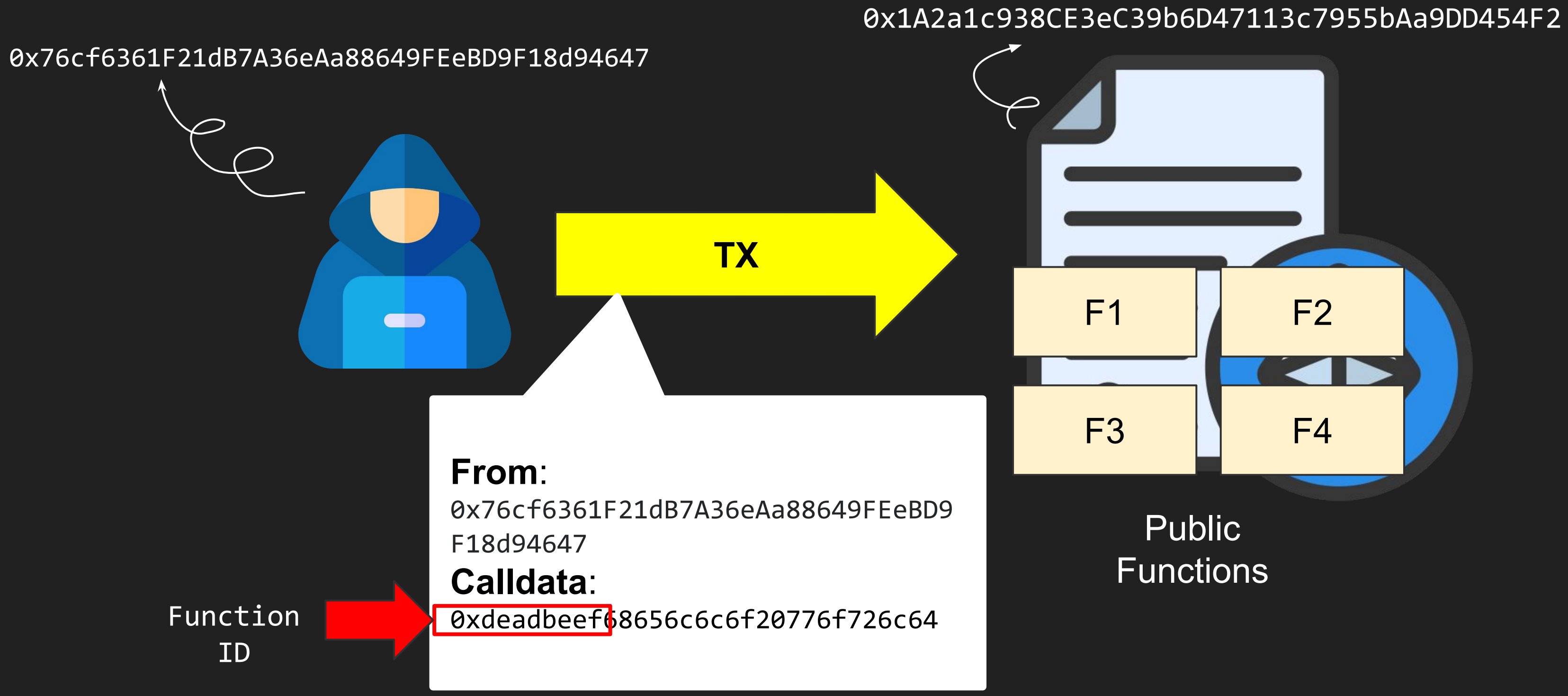
Smart Contracts 101



0x76cf6361F21dB7A36eAa88649FEeBD9F18d94647
0x1A2a1c938CE3eC39b6D47113c7955bAa9DD454F2
TX
F1
F2
F3
F4
Public Functions

From:
`0x76cf6361F21dB7A36eAa88649FEeBD9F18d94647`
Calldata:
`0xdeadbeef68656c6c6f20776f726c64`

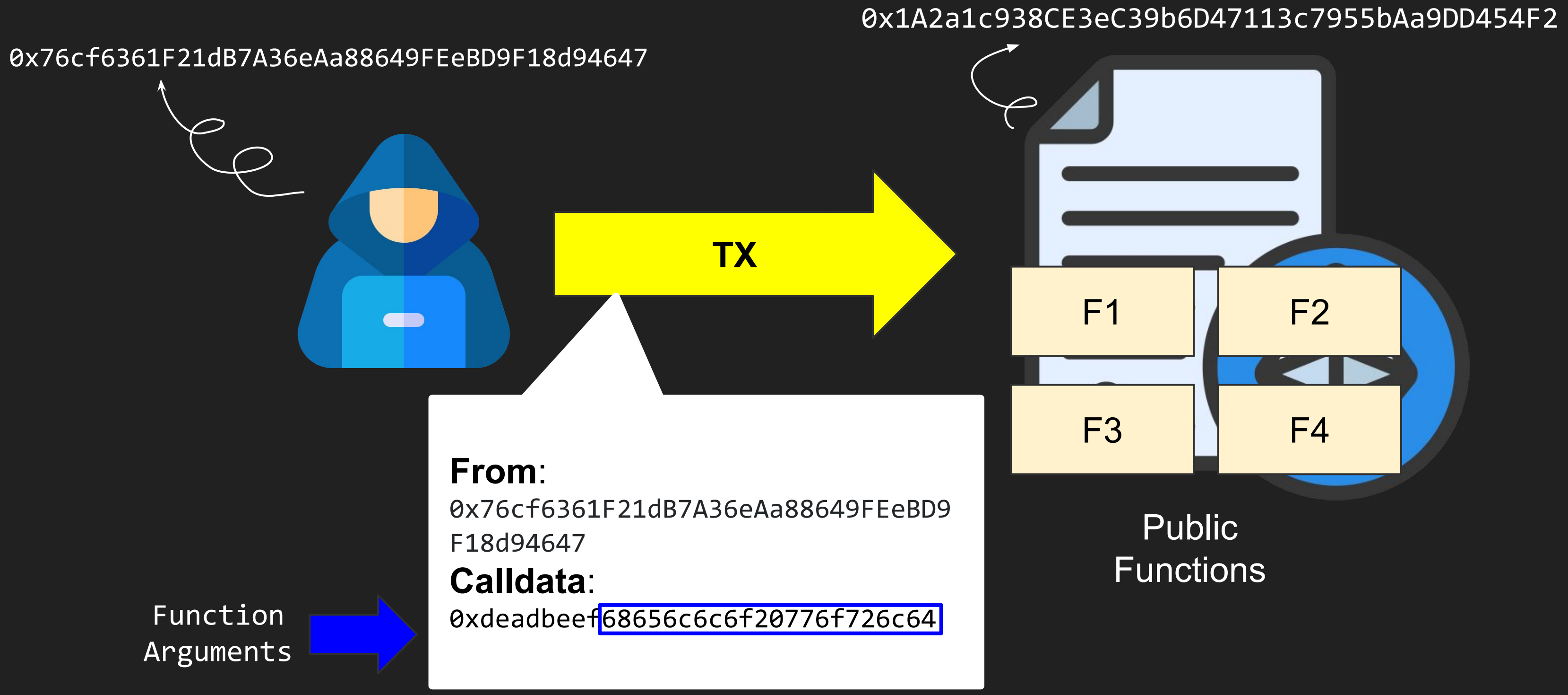
Smart Contracts 101


0x76cf6361F21dB7A36eAa88649FEeBD9F18d94647
0x1A2a1c938CE3eC39b6D47113c7955bAa9DD454F2
TX
F1
F2
F3
F4
Public Functions
Function ID

From:
0x76cf6361F21dB7A36eAa88649FEeBD9F18d94647

Calldata:
0xdeadbeef68656c6c6f20776f726c64

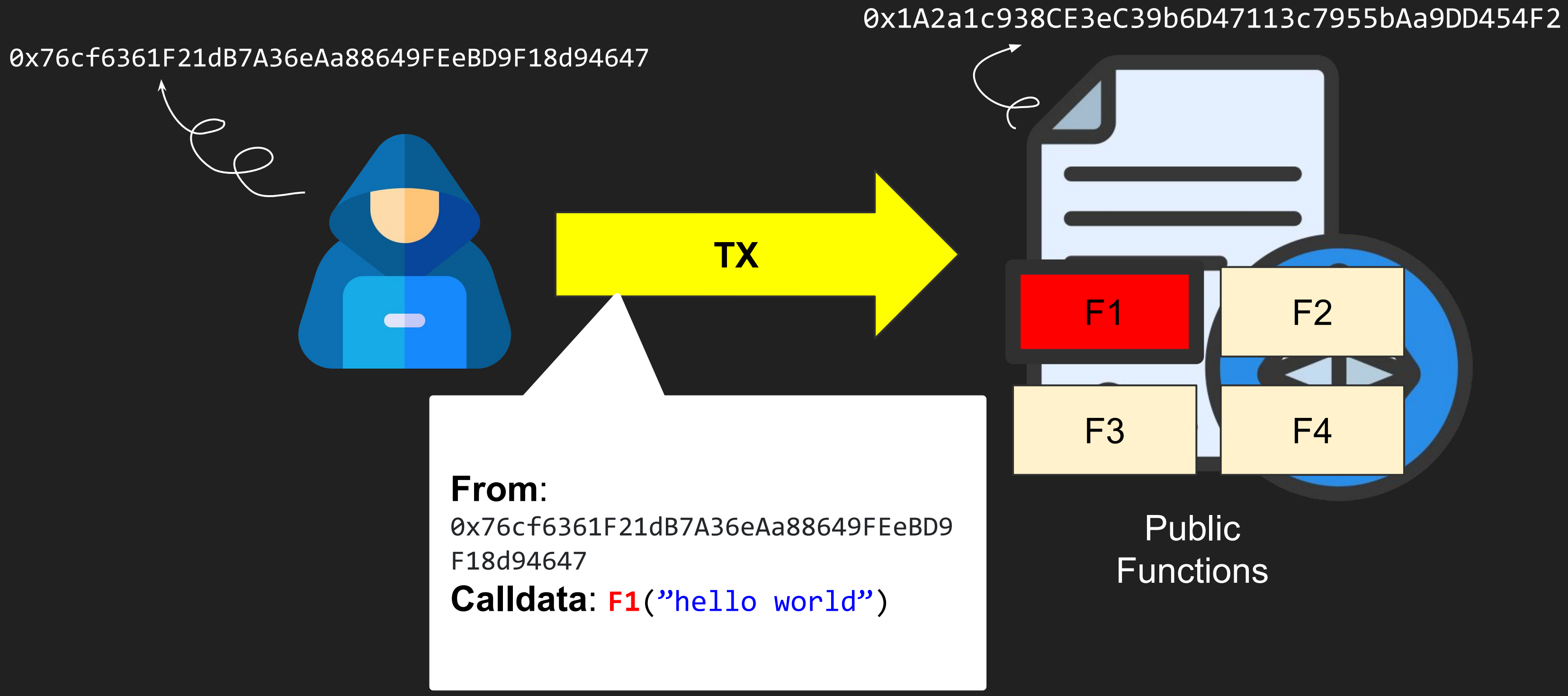
Smart Contracts 101


0x76cf6361F21dB7A36eAa88649FEeBD9F18d94647
0x1A2a1c938CE3eC39b6D47113c7955bAa9DD454F2
TX
F1
F2
F3
F4
Public Functions
Function Arguments

From:
0x76cf6361F21dB7A36eAa88649FEeBD9F18d94647

Calldata:
0xdeadbeef68656c6c6f20776f726c64

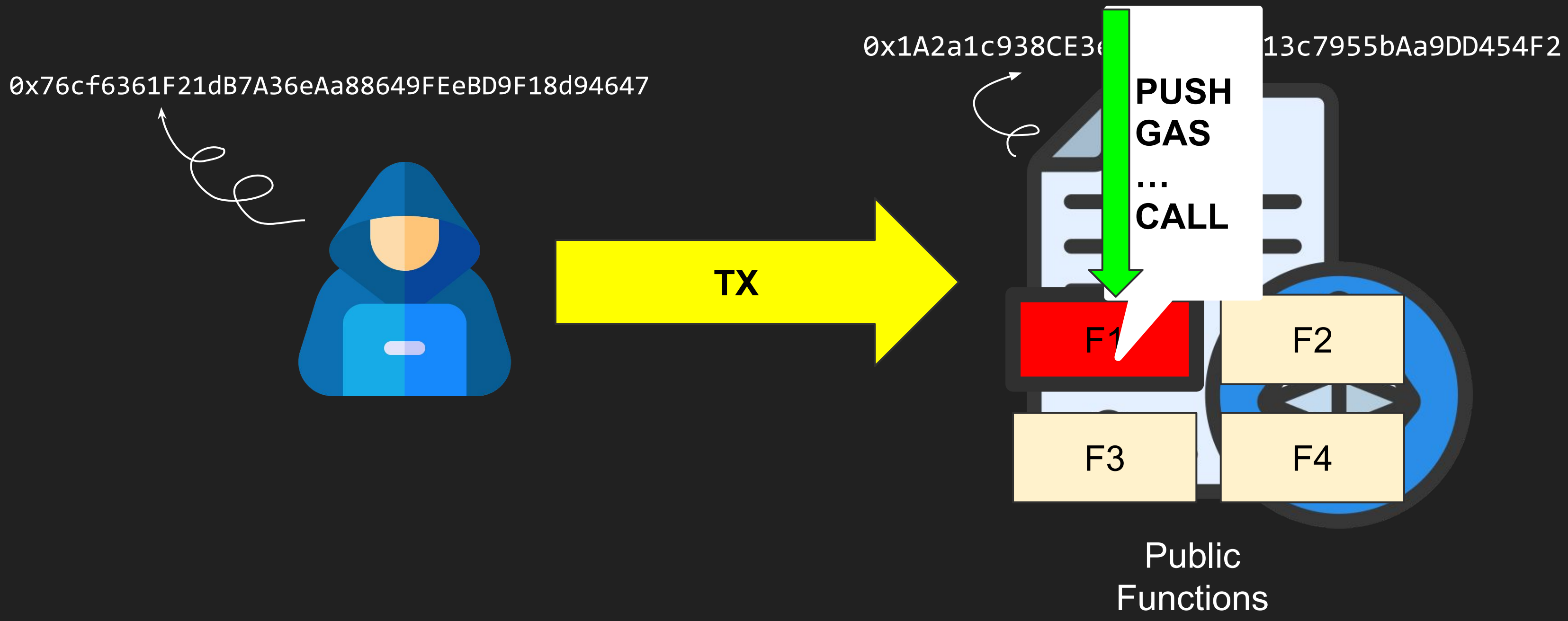
Smart Contracts 101


0x76cf6361F21dB7A36eAa88649FEeBD9F18d94647
0x1A2a1c938CE3eC39b6D47113c7955bAa9DD454F2
TX
F1
F2
F3
F4
Public Functions

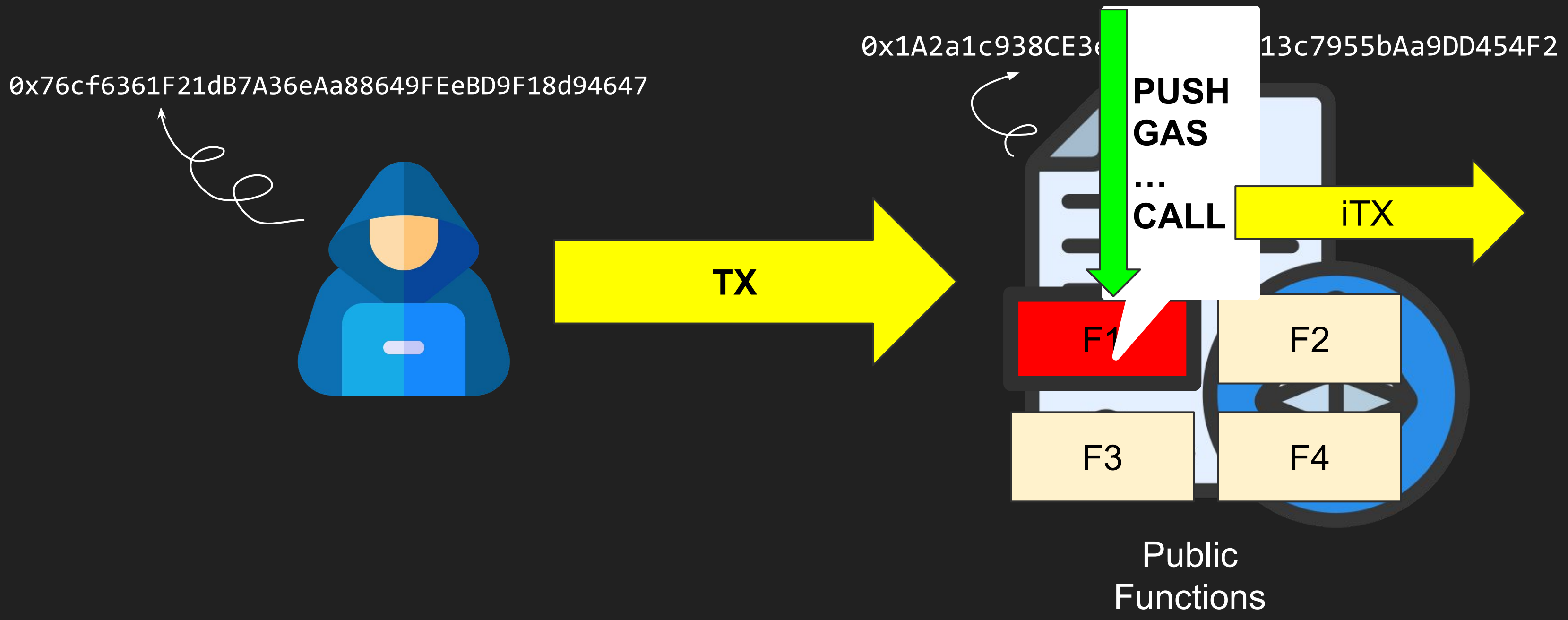
From:
`0x76cf6361F21dB7A36eAa88649FEeBD9F18d94647`

Calldata: `F1("hello world")`

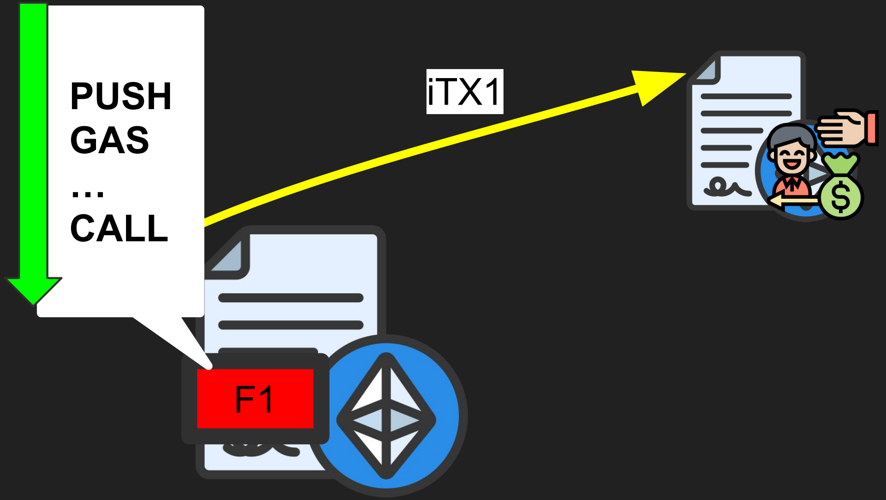
Smart Contracts 101


0x76cf6361F21dB7A36eAa88649FEeBD9F18d94647
0x1A2a1c938CE3e 13c7955bAa9DD454F2
TX
PUSH
GAS
...
CALL
F1
F2
F3
F4
Public
Functions

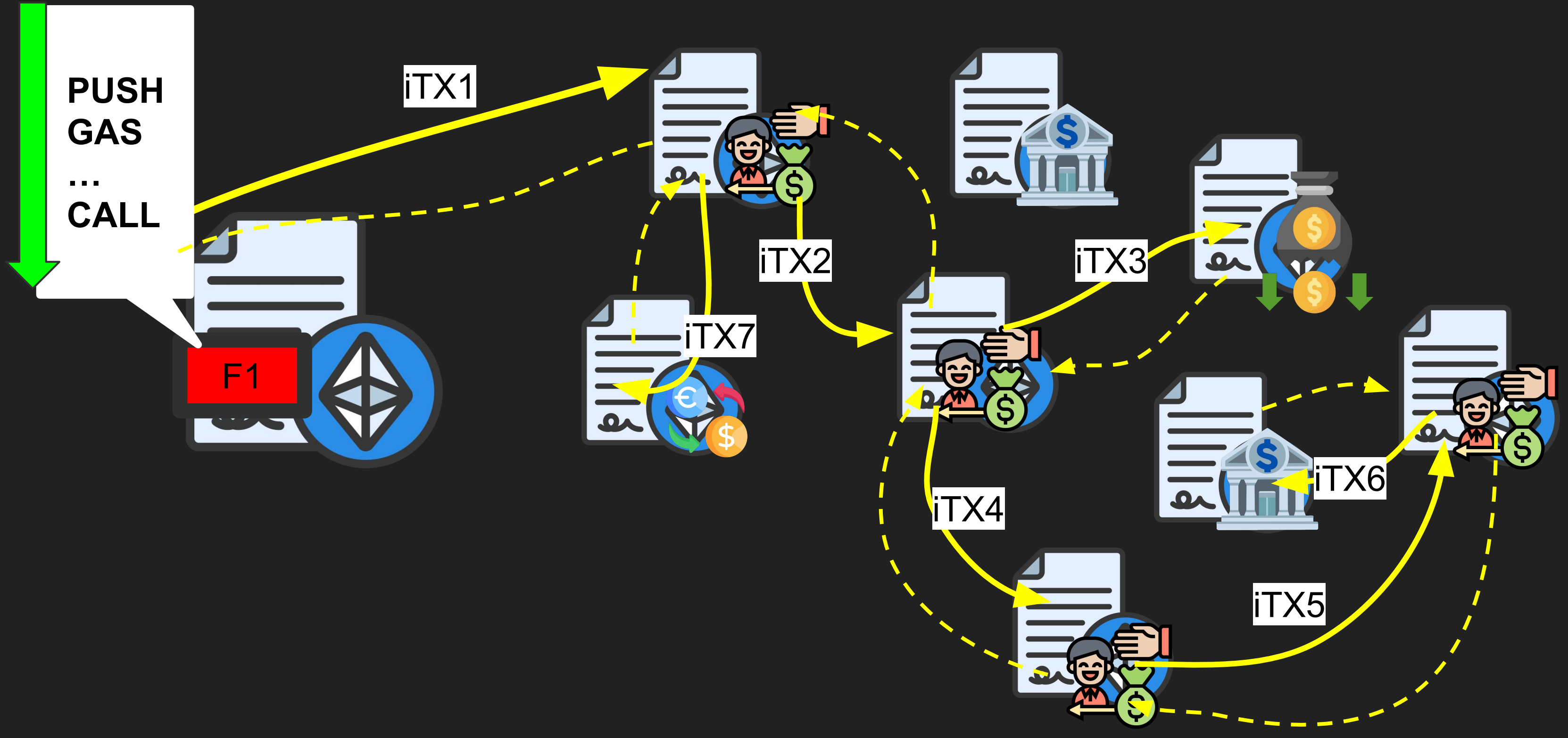
Smart Contracts 101



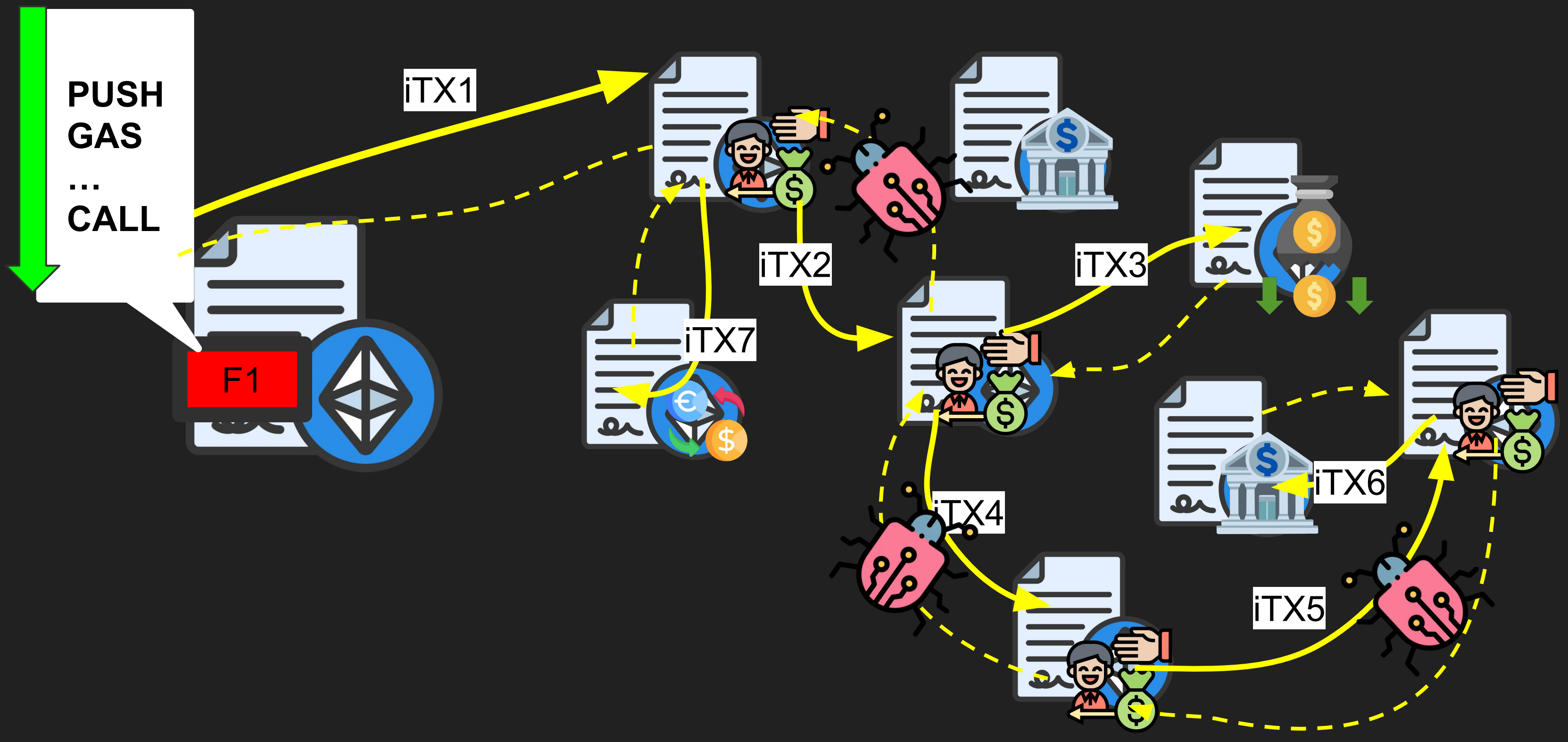
Smart Contracts 101



Smart Contracts 101



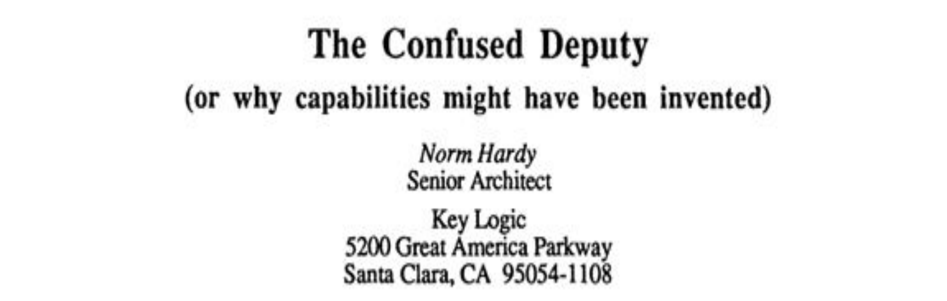
Smart Contracts 101



Confused Deputy Vulnerabilities in Ethereum Smart Contracts

Confused Deputy

- Bug class introduced by Norman Hardy in 1988



The Confused Deputy

(or why capabilities might have been invented)

Norm Hardy
Senior Architect

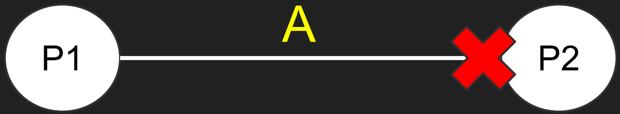
Key Logic
5200 Great America Parkway
Santa Clara, CA 95054-1108

Confused Deputy

P1

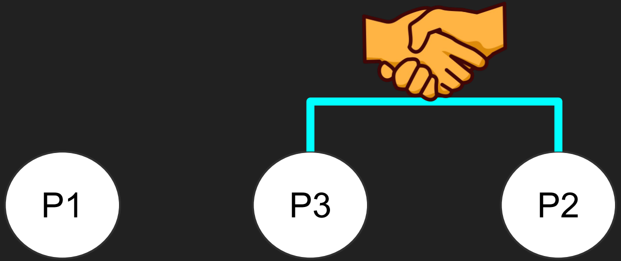
P2

Confused Deputy



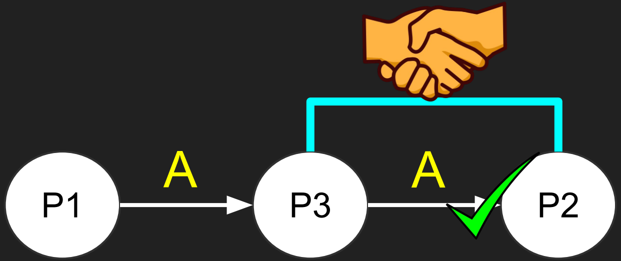
No relationship of trust between P1 and P2

Confused Deputy



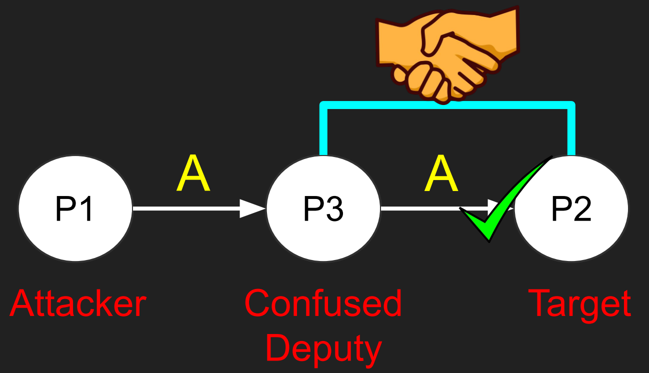
P3: trusted middleman

Confused Deputy

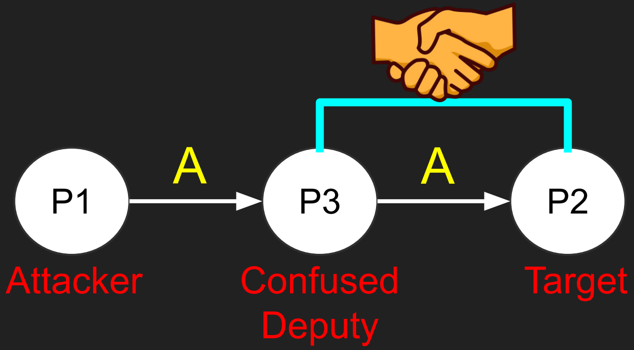


P1 can trick P3 to perform the action **A** on P2

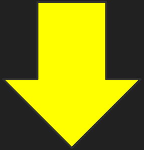
Confused Deputy



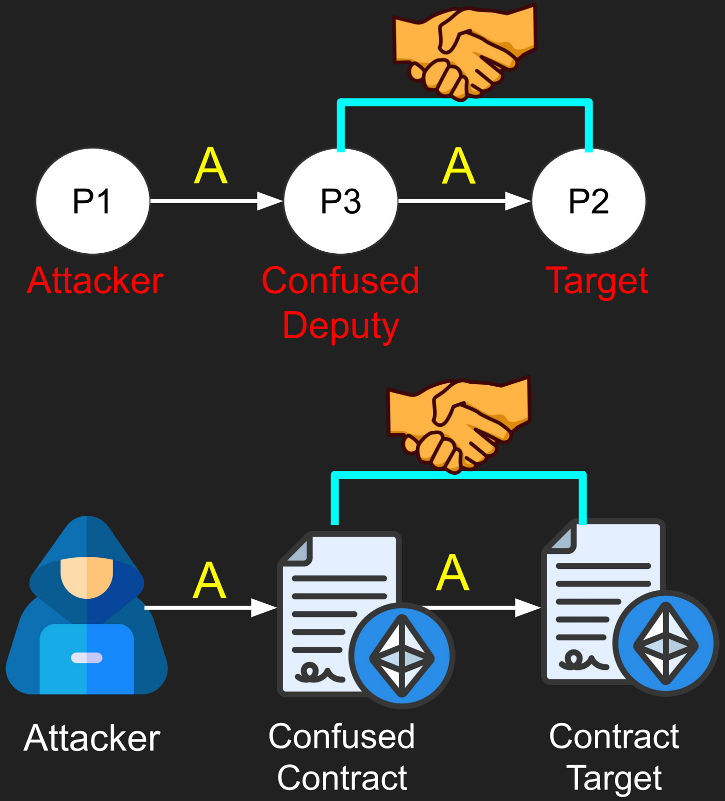
Confused Deputy



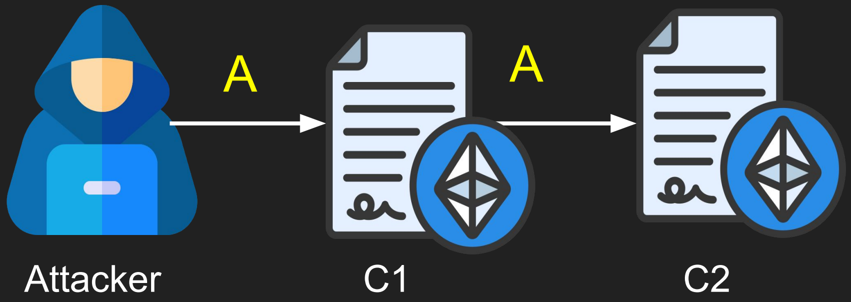
Confused Deputy



Confused Contract

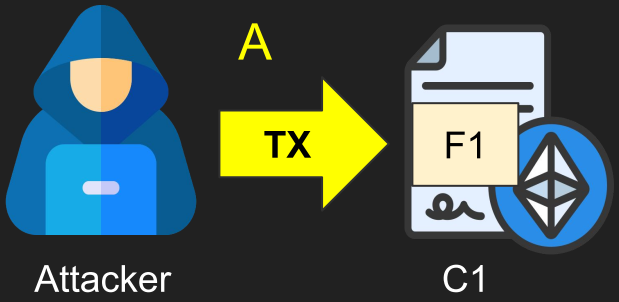


Confused Contract

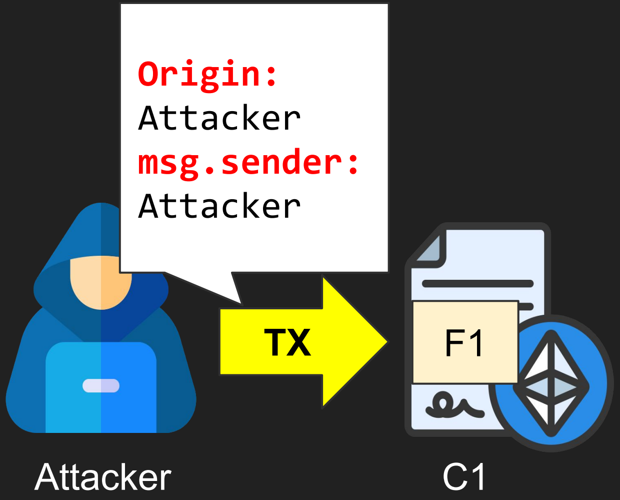


What does privilege escalation look like?

Confused Contract

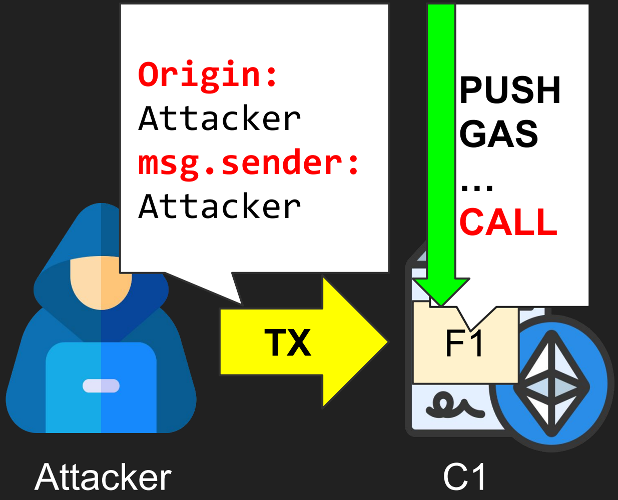


Confused Contract

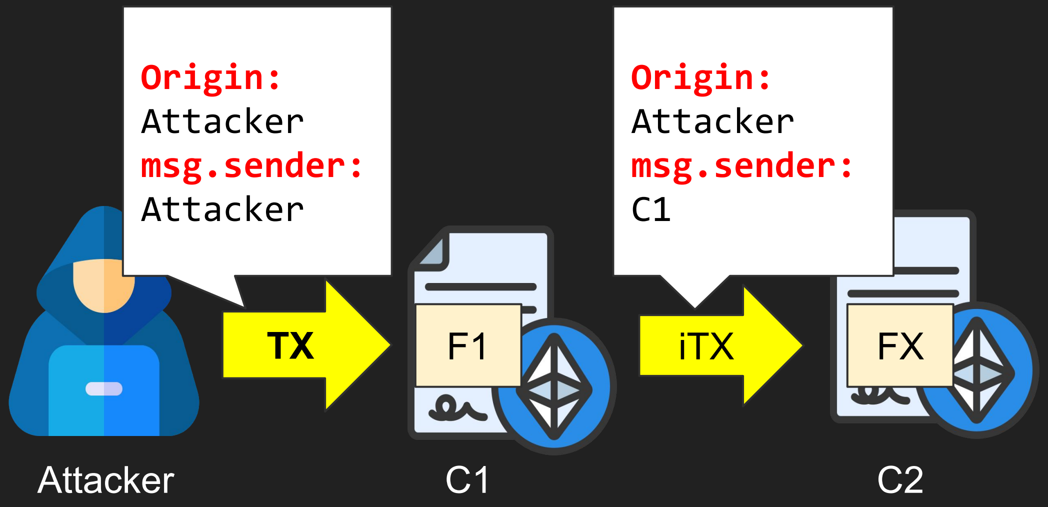


Origin: Attacker
msg.sender: Attacker

Confused Contract



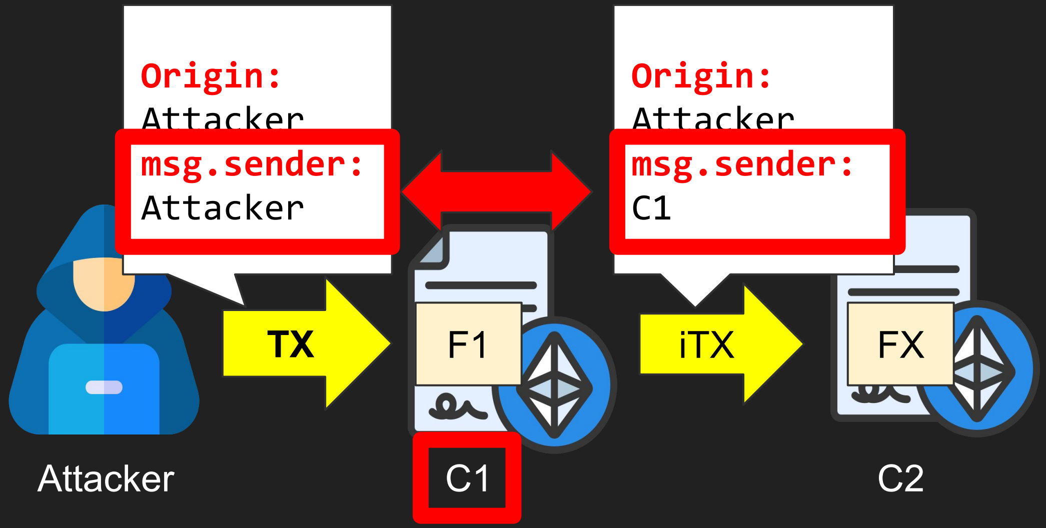
Confused Contract



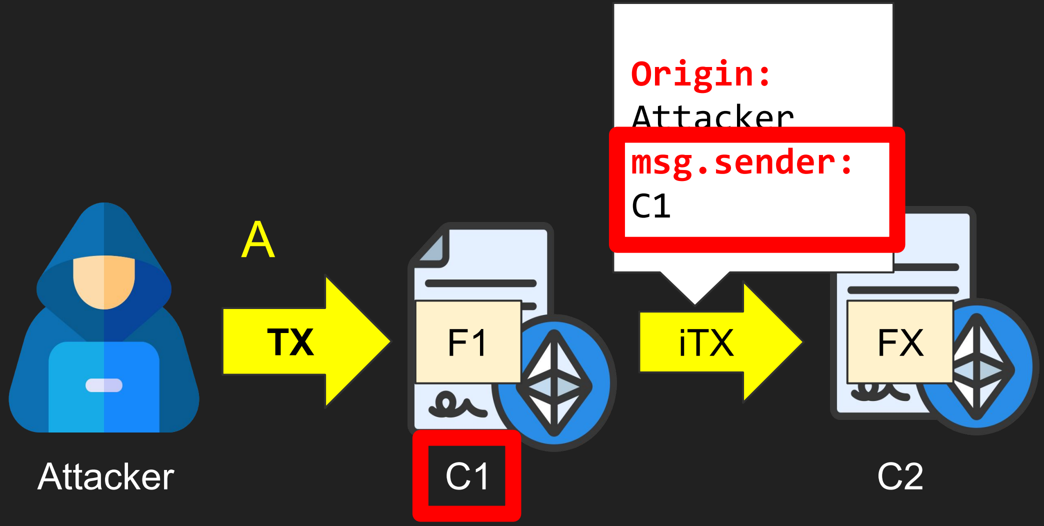
Origin: Attacker
msg.sender: Attacker

Origin: Attacker
msg.sender: C1

Confused Contract

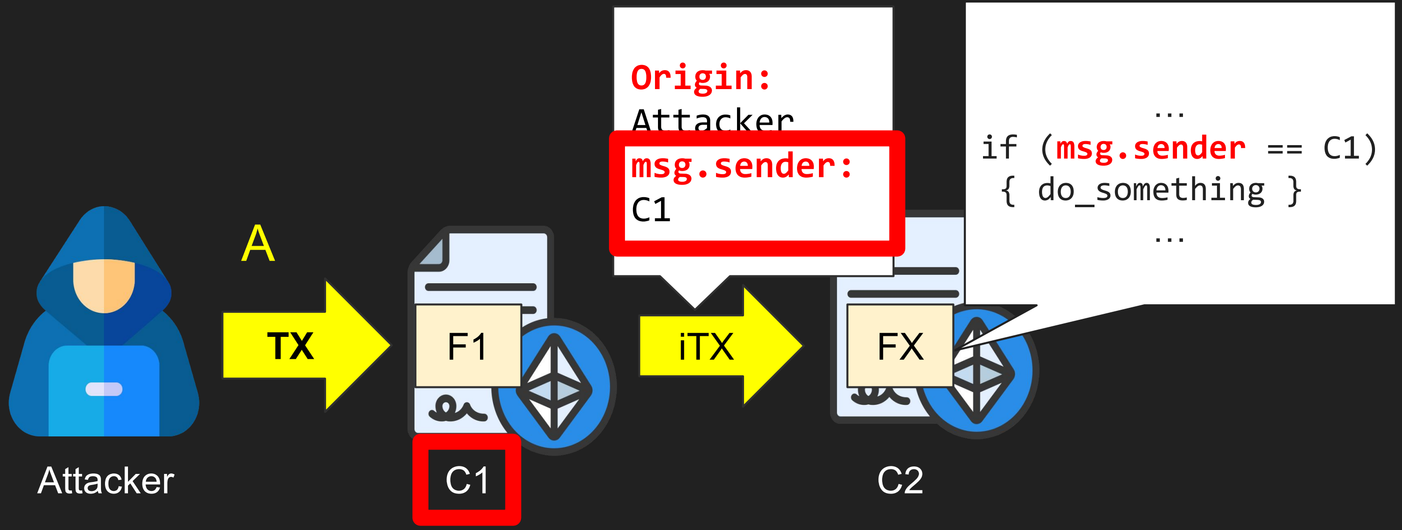


Confused Contract



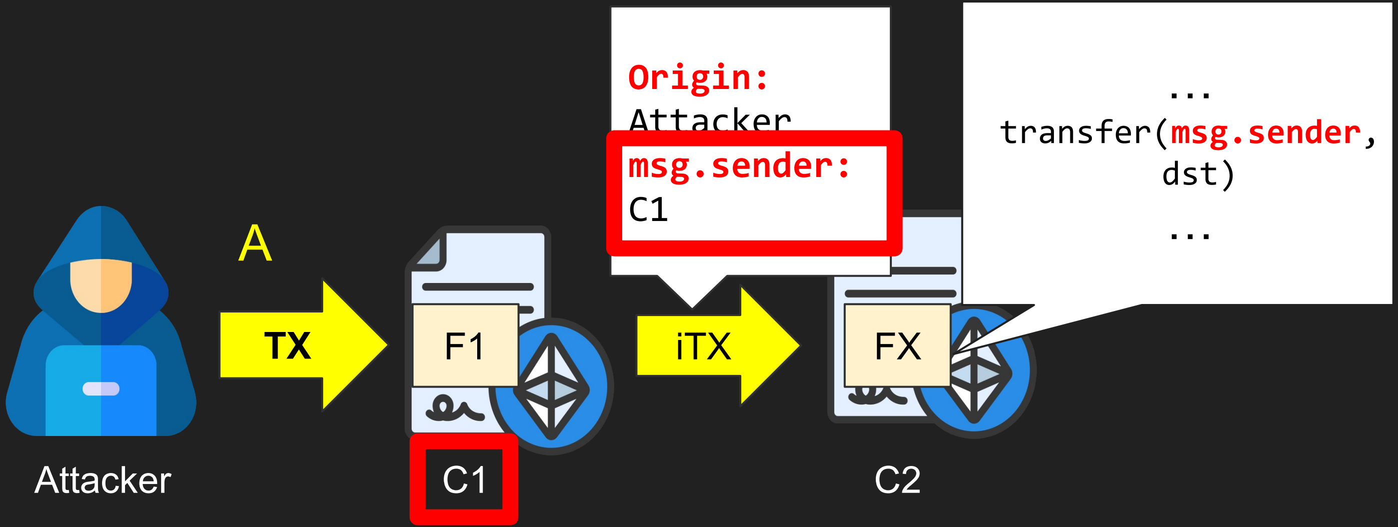
msg.sender is used by contracts for different kind of operations

Confused Contract



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

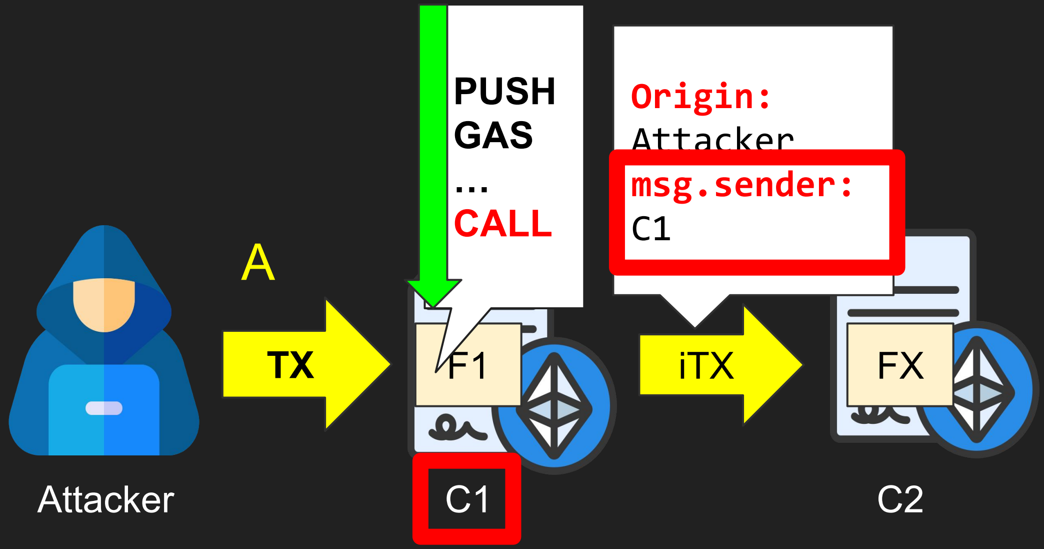
Confused Contract



```
Origin:
Attacker
msg.sender:
C1
```

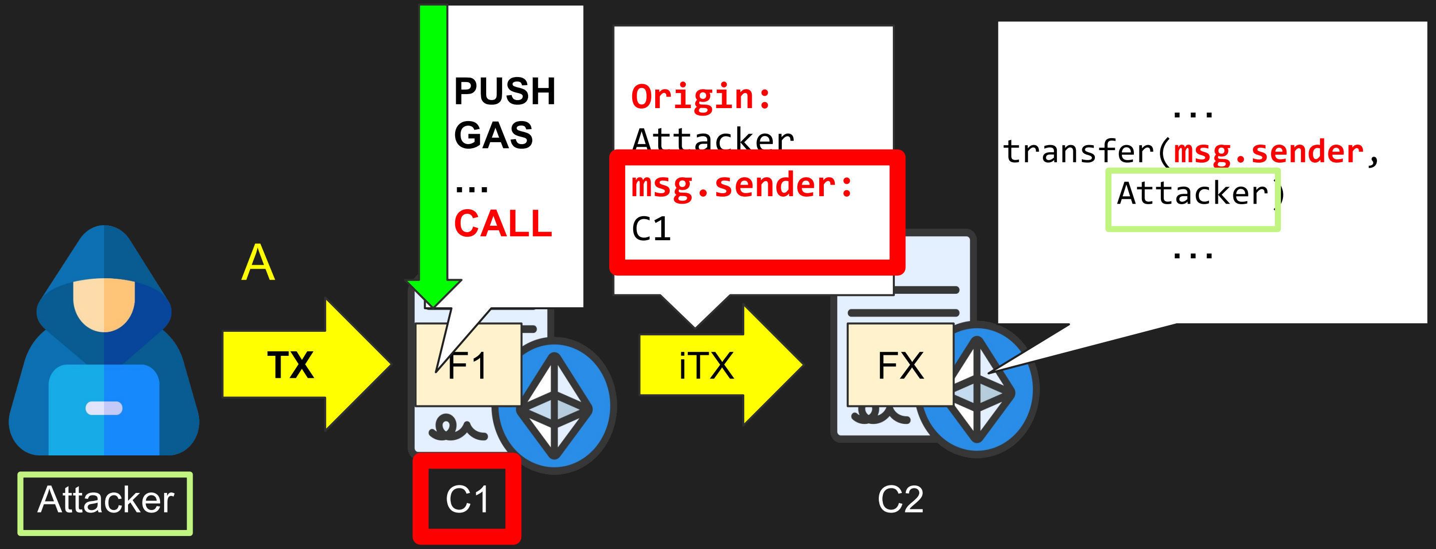
```
...
transfer(msg.sender,
dst)
...
```

Confused Contract



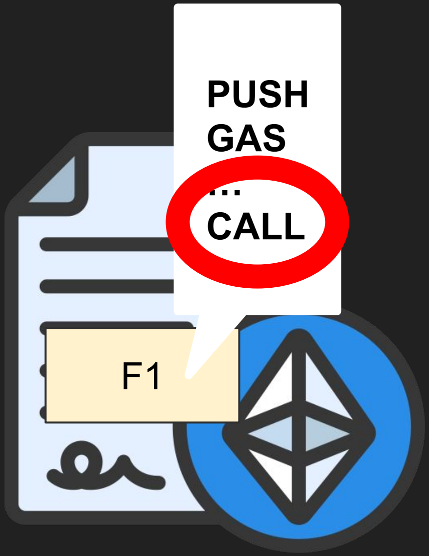
If C1 allows everyone to execute a CALL, an attacker can “borrow” its identity for privileges escalation in another contract!

Confused Contract



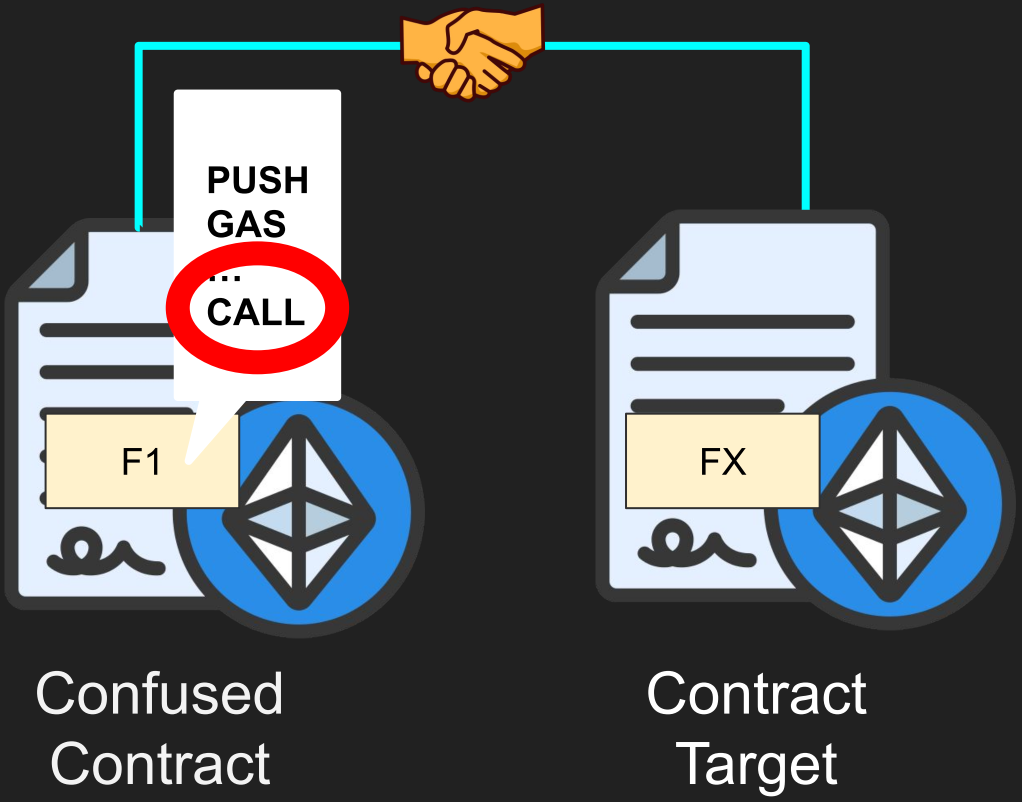
If C1 allows everyone to execute a CALL, an attacker can "borrow" its identity for privileges escalation in another contract!

Identify Confused Contract Vulnerabilities

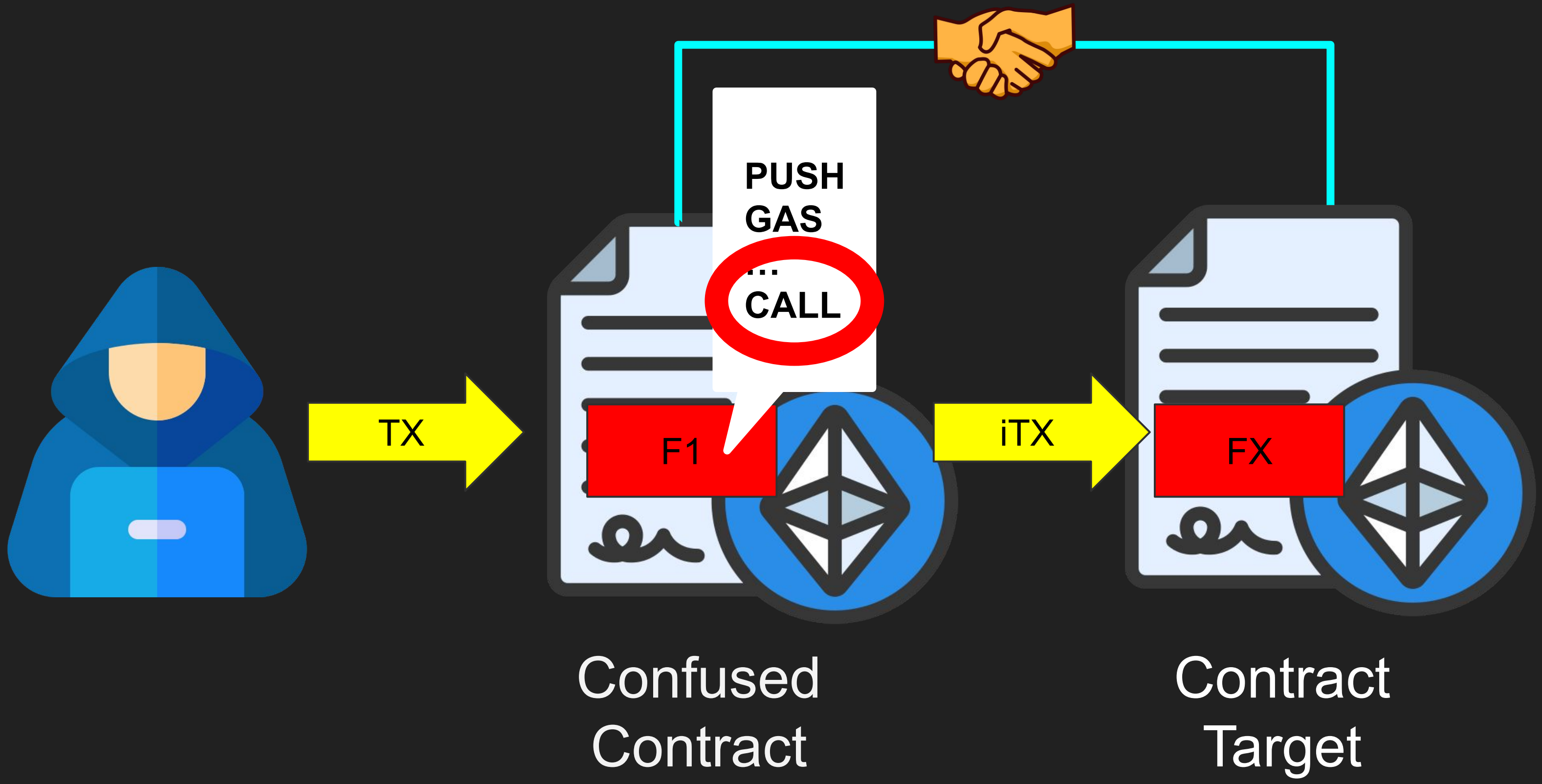


Confused Contract

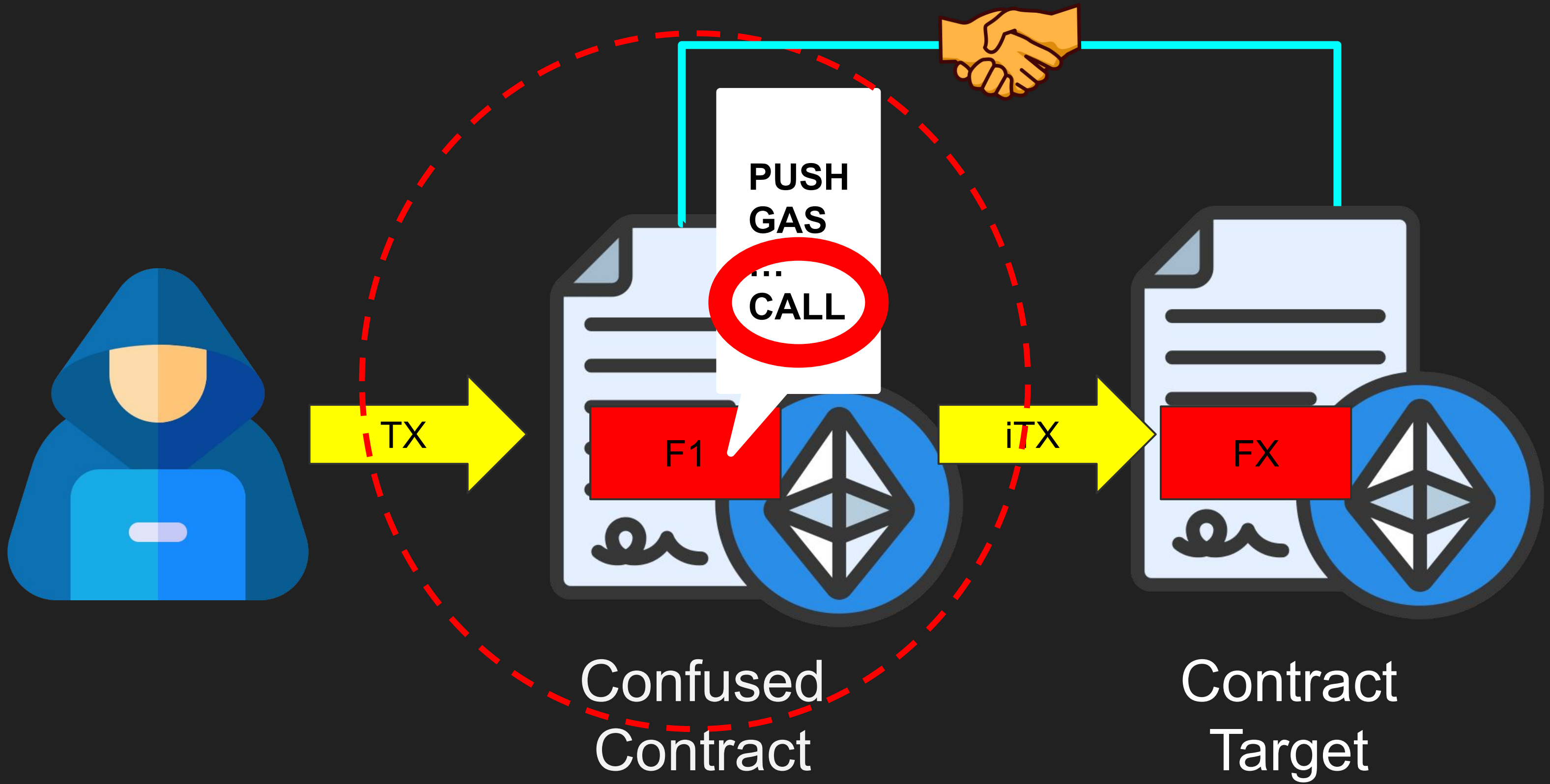
Identify Confused Contract Vulnerabilities



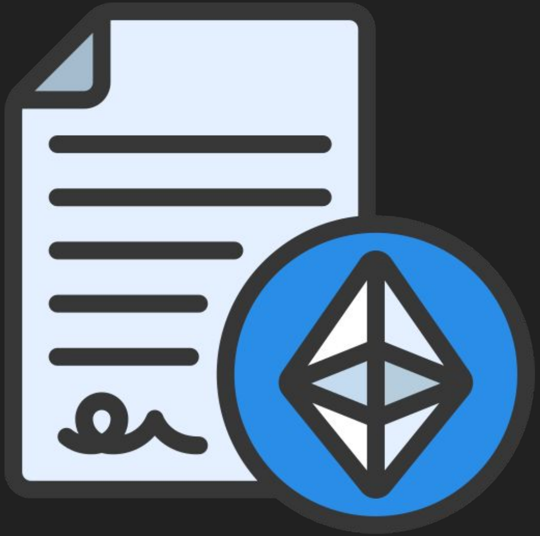
Identify Confused Contract Vulnerabilities



Identify Confused Contract Vulnerabilities

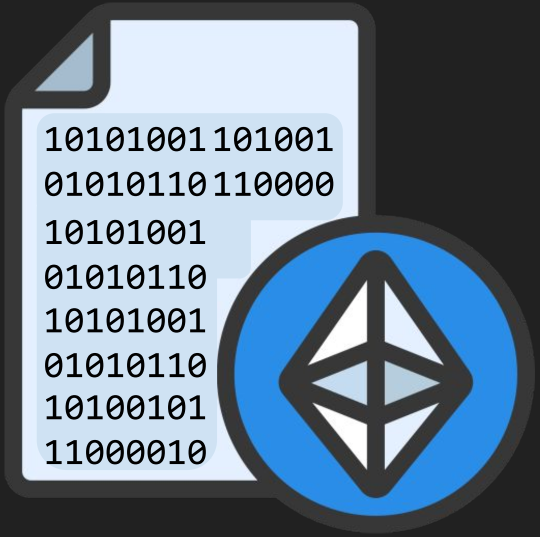


Identify Confused Contract



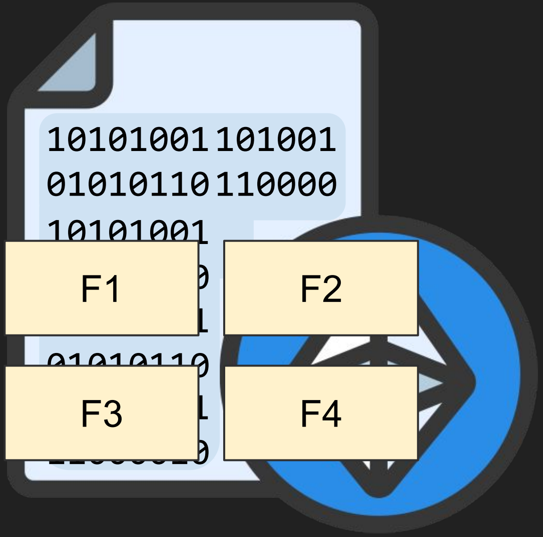
Smart Contract

Identify Confused Contract



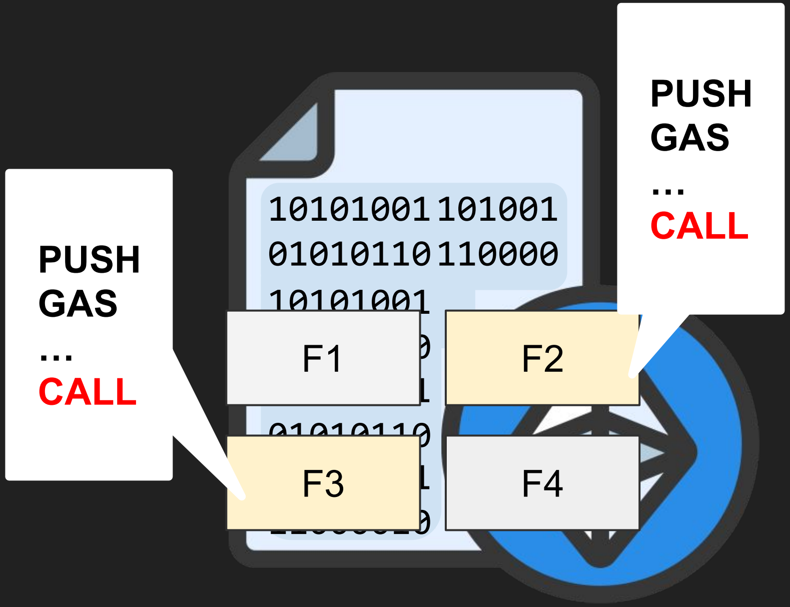
Smart Contract Binary

Identify Confused Contract



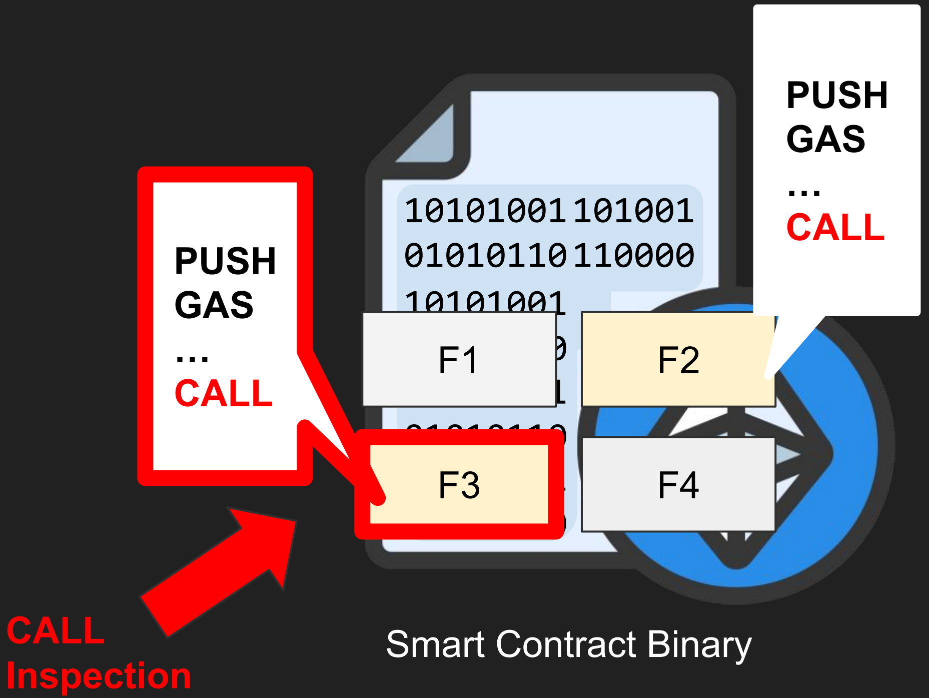
Smart Contract Binary

Identify Confused Contract



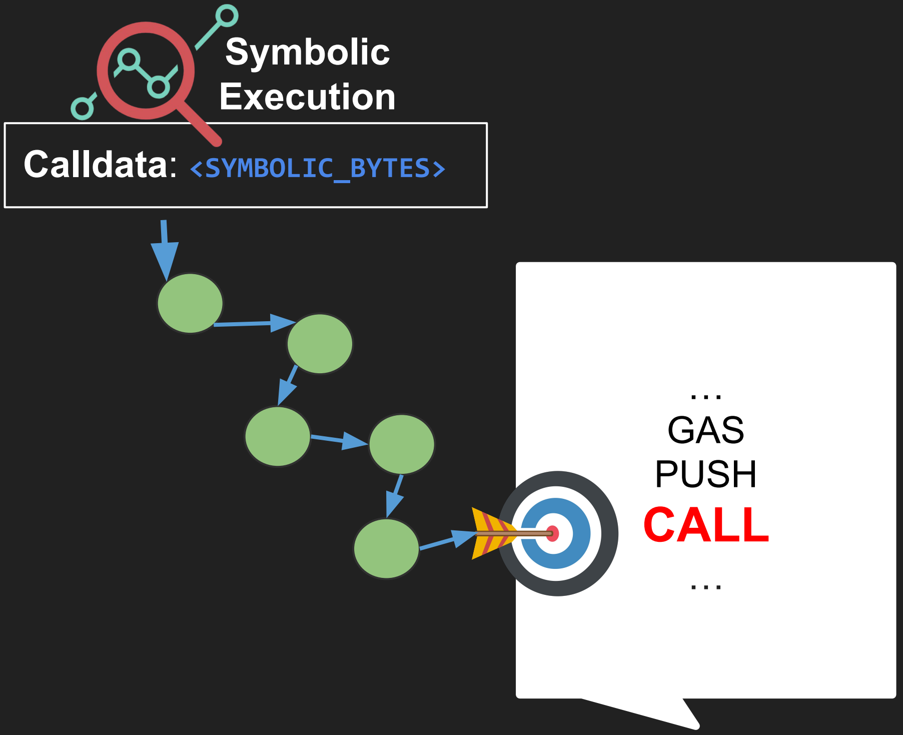
Smart Contract Binary

Identify Confused Contract



Smart Contract Binary

CALL Inspection



Symbolic Execution

Calldata: <SYMBOLIC_BYTES>

...

GAS

PUSH

CALL

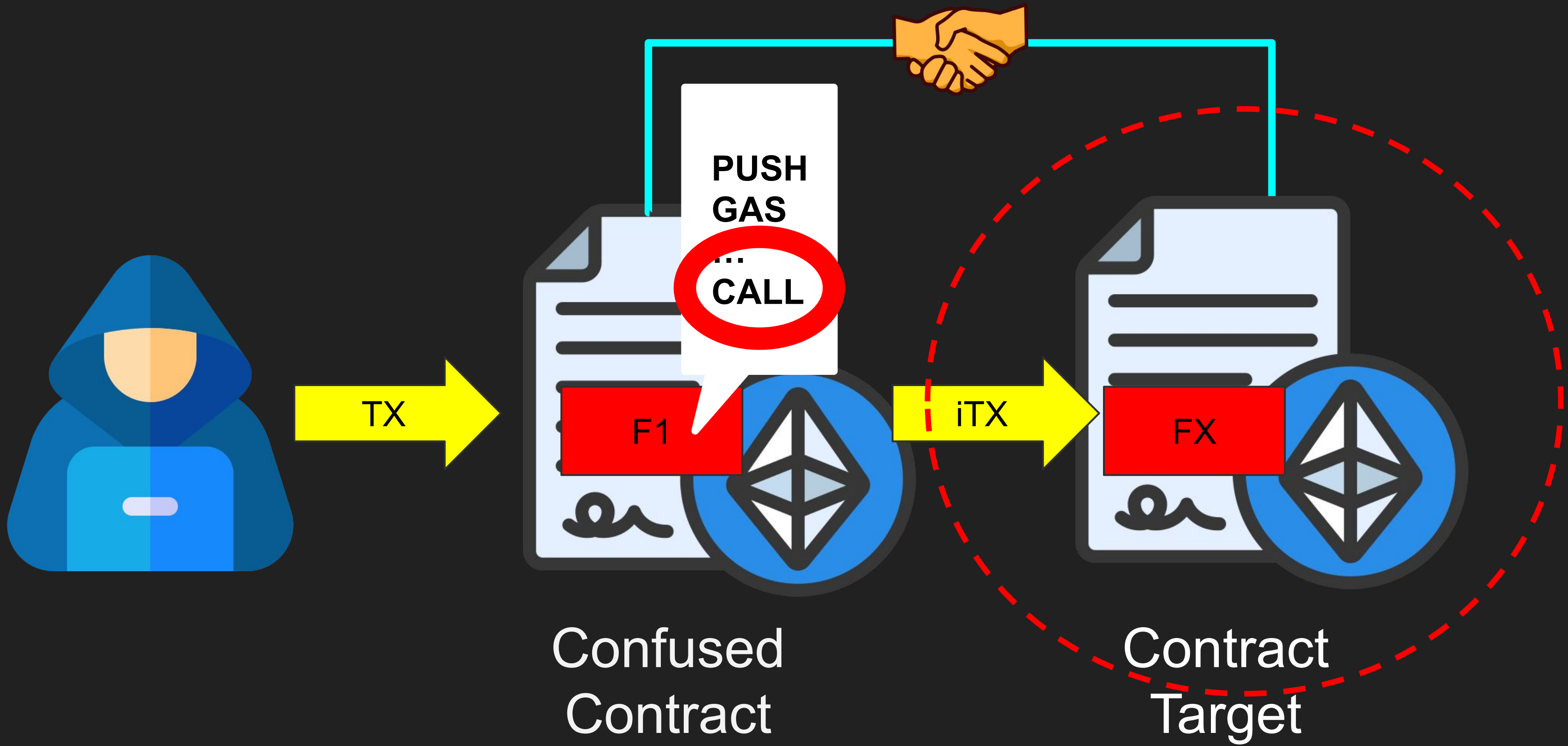
...

- Directed Symbolic execution to understand if a CALL instruction is reachable by an attacker

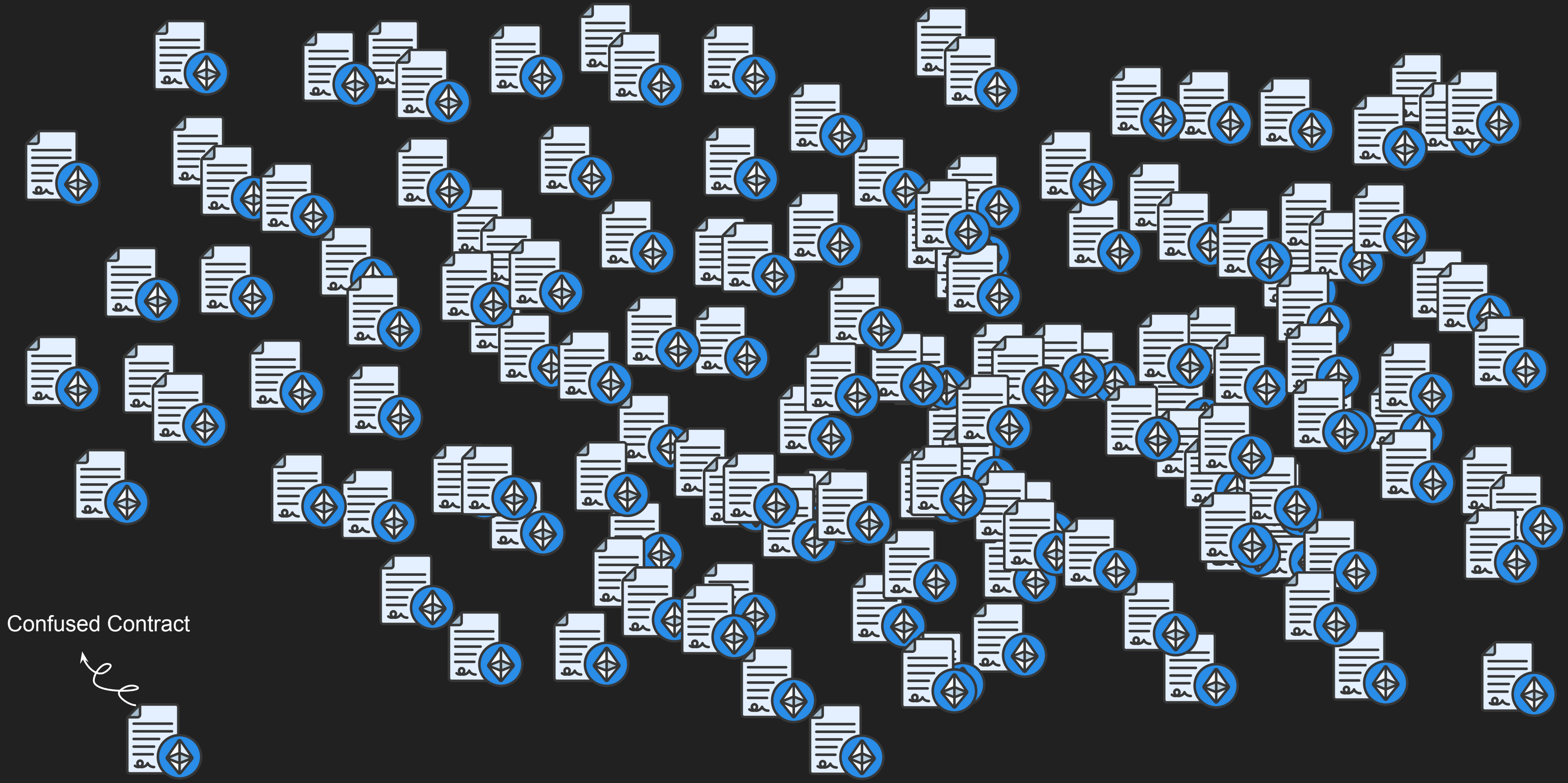
- If reachable, can attacker control the destination of this CALL?

- If yes, we found a **confused contract**

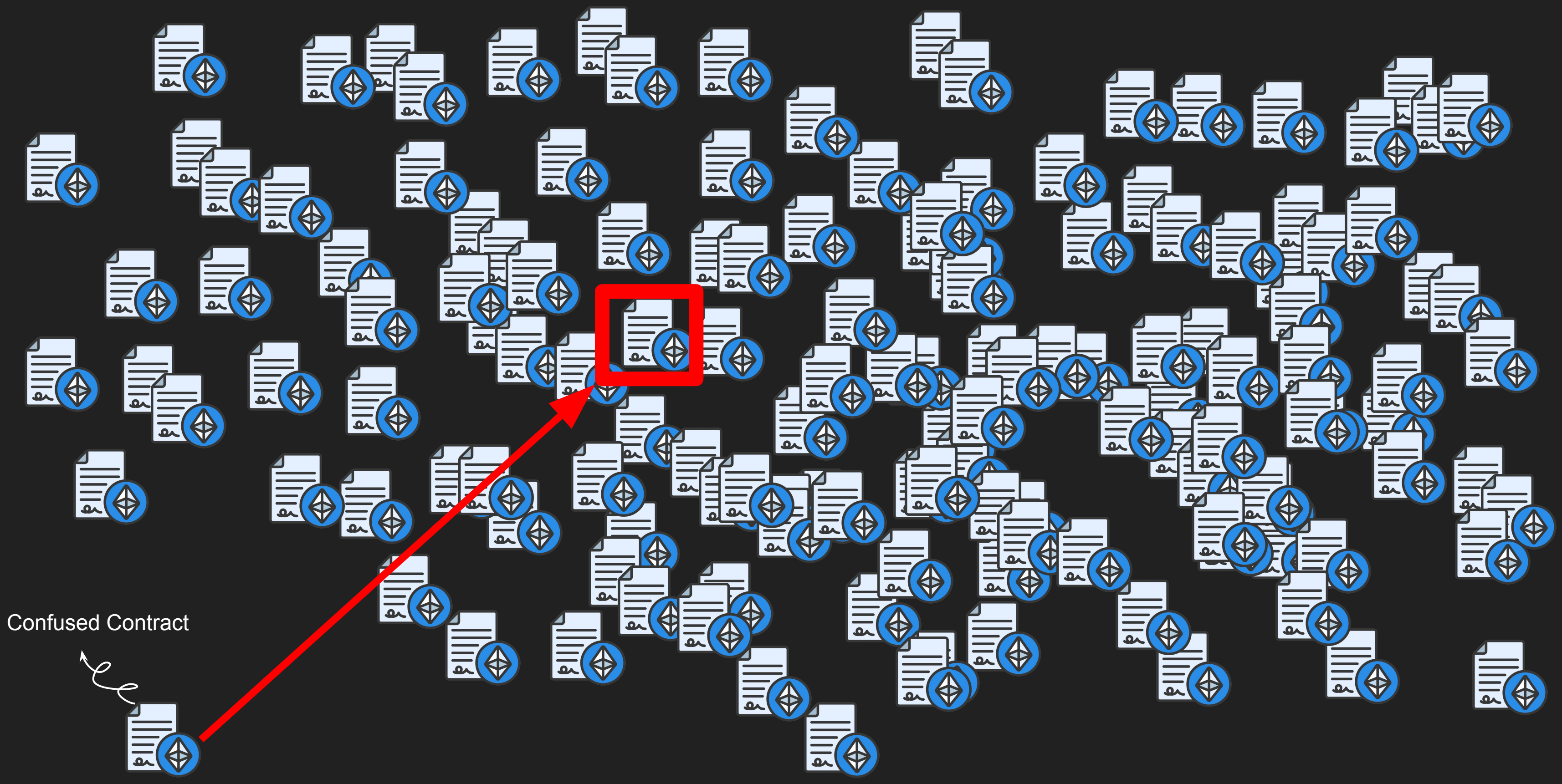
Identify Confused Contract Vulnerabilities



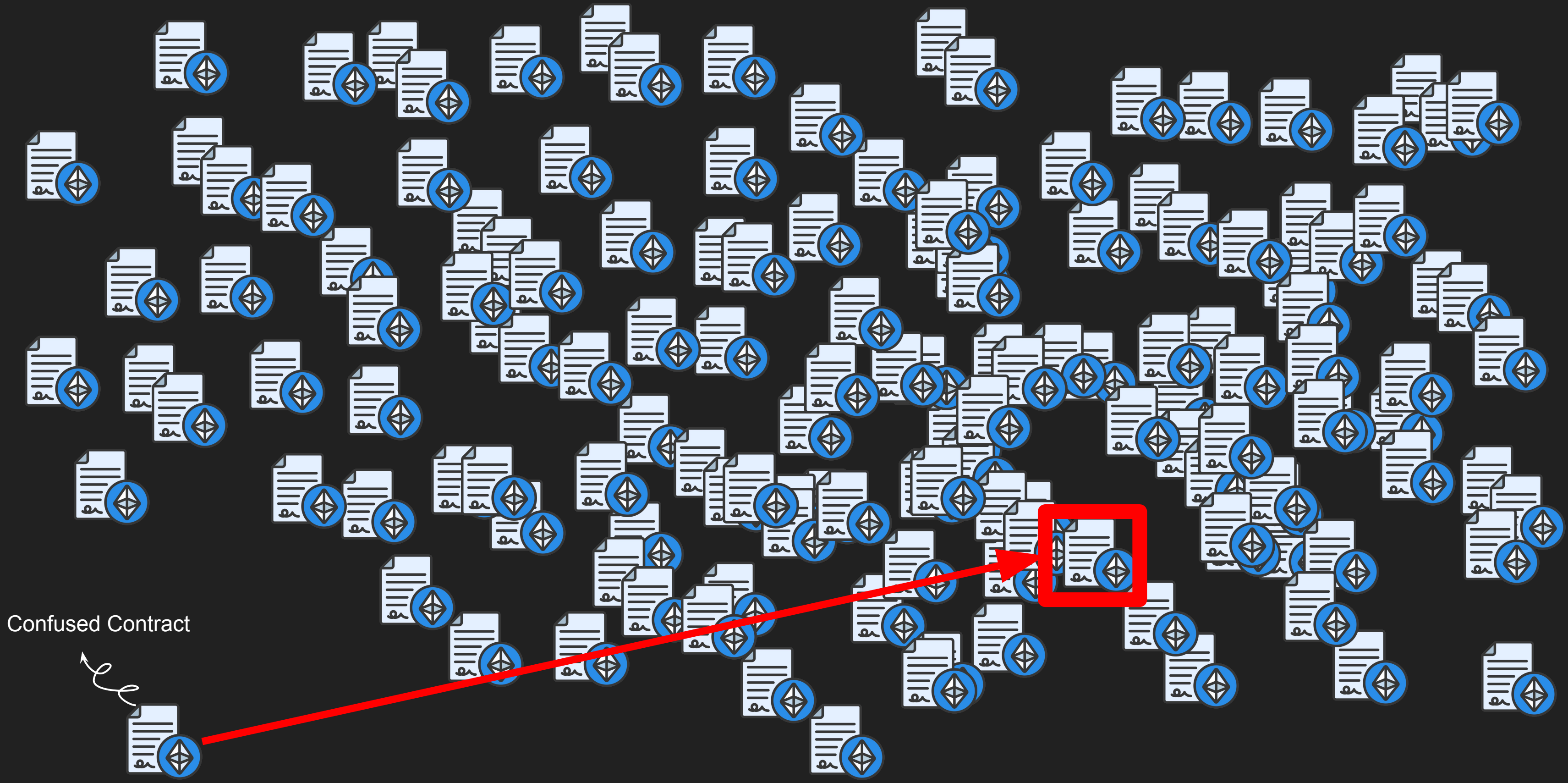
How to find a Contract Target?



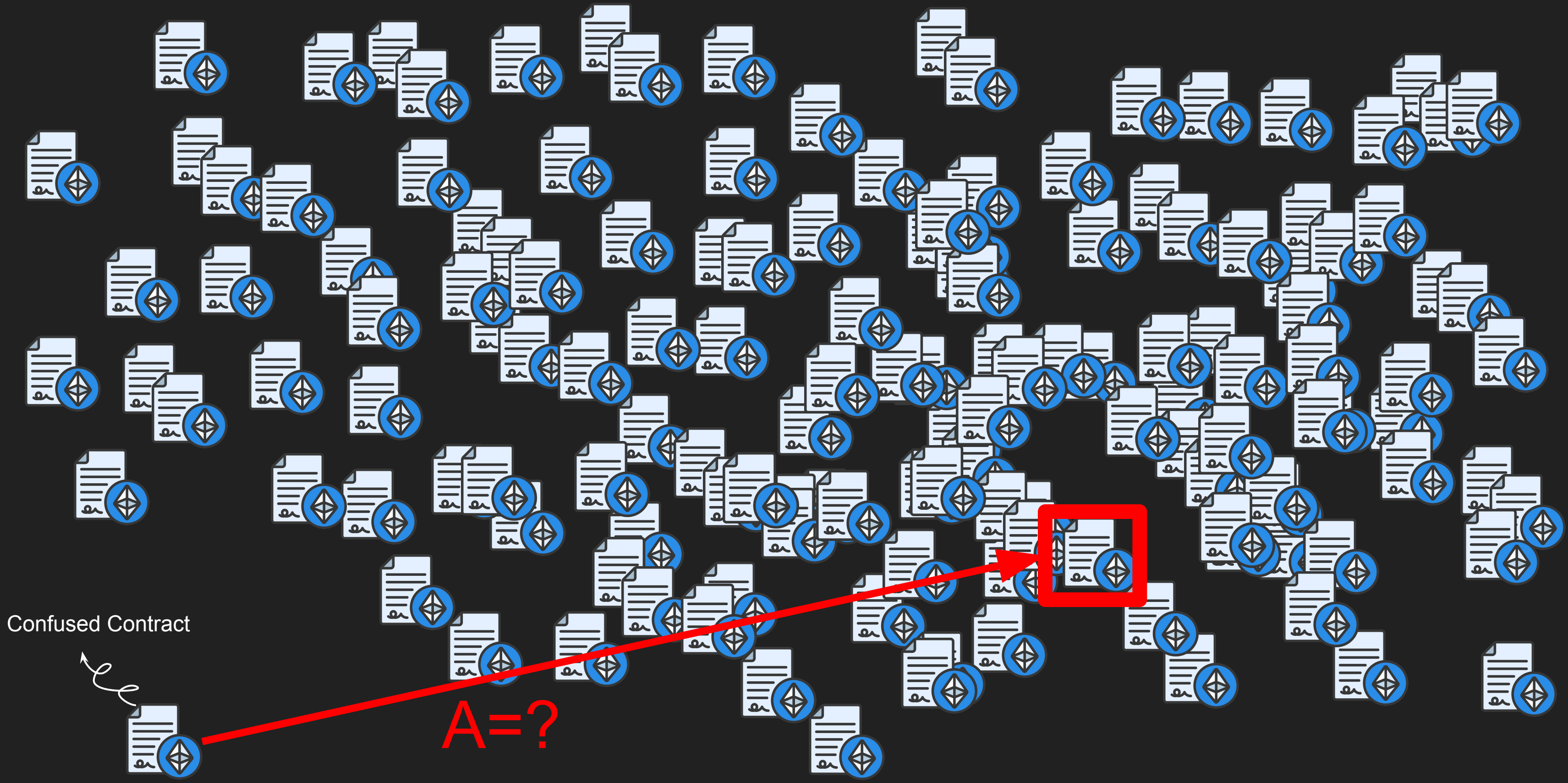
Millions of smart contracts!



Millions of smart contracts!



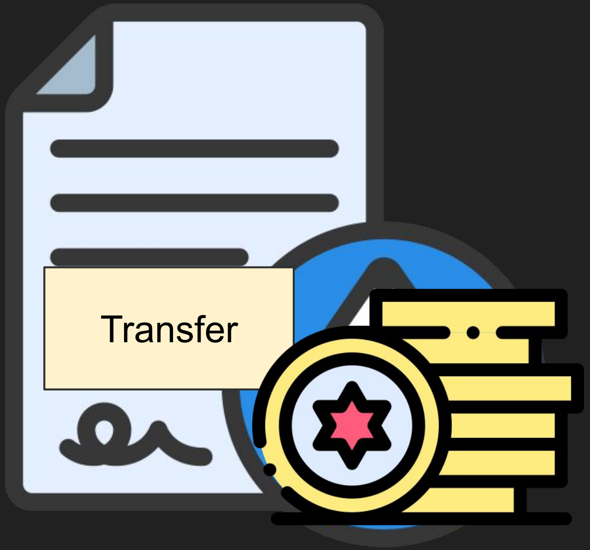
Millions of smart contracts!



Millions of smart contracts!

Let’s simplify

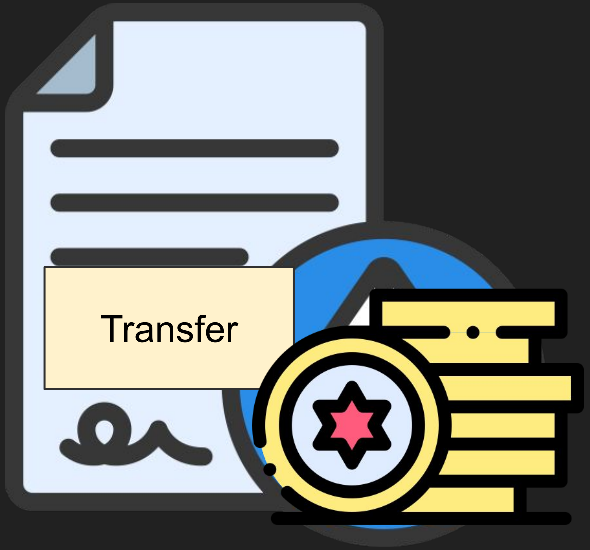
Identify Contract Target



ERC20 Contract
(Token Contract)

- An instance of a Contract
  Target: ERC20 token contracts

Identify Contract Target

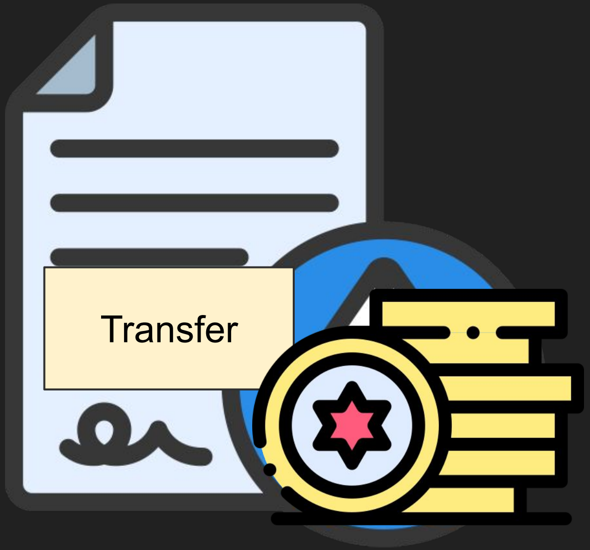


ERC20 Contract
(Token Contract)

- An instance of a Contract Target: ERC20 token contracts

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

Identify Contract Target



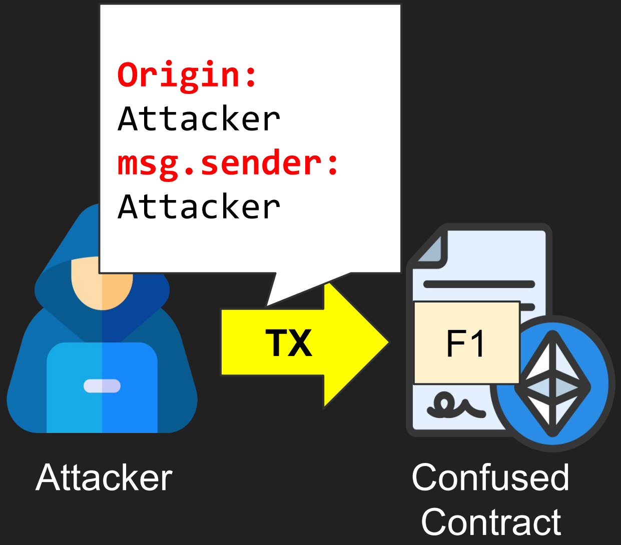
ERC20 Contract
(Token Contract)

- An instance of a Contract Target: ERC20 token contracts

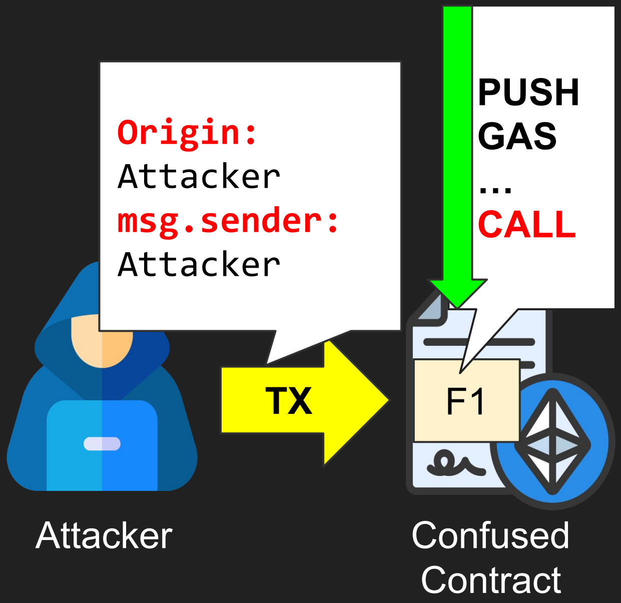
- ERC20 Contracts hold a “state” on behalf of other contracts: their **tokens balance**

- This state is automatically identifiable! (check details on the paper)

Exploit


Origin:
Attacker
msg.sender:
Attacker
TX
F1
Attacker
Confused Contract

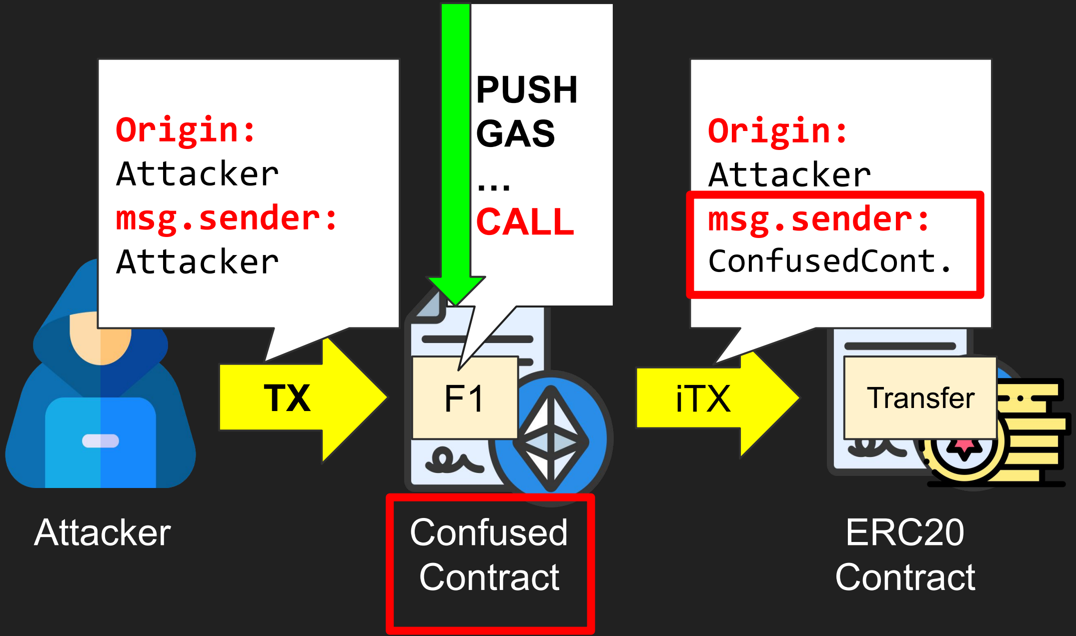
Exploit



Origin:
Attacker
msg.sender:
Attacker

PUSH
GAS
...
CALL

Exploit

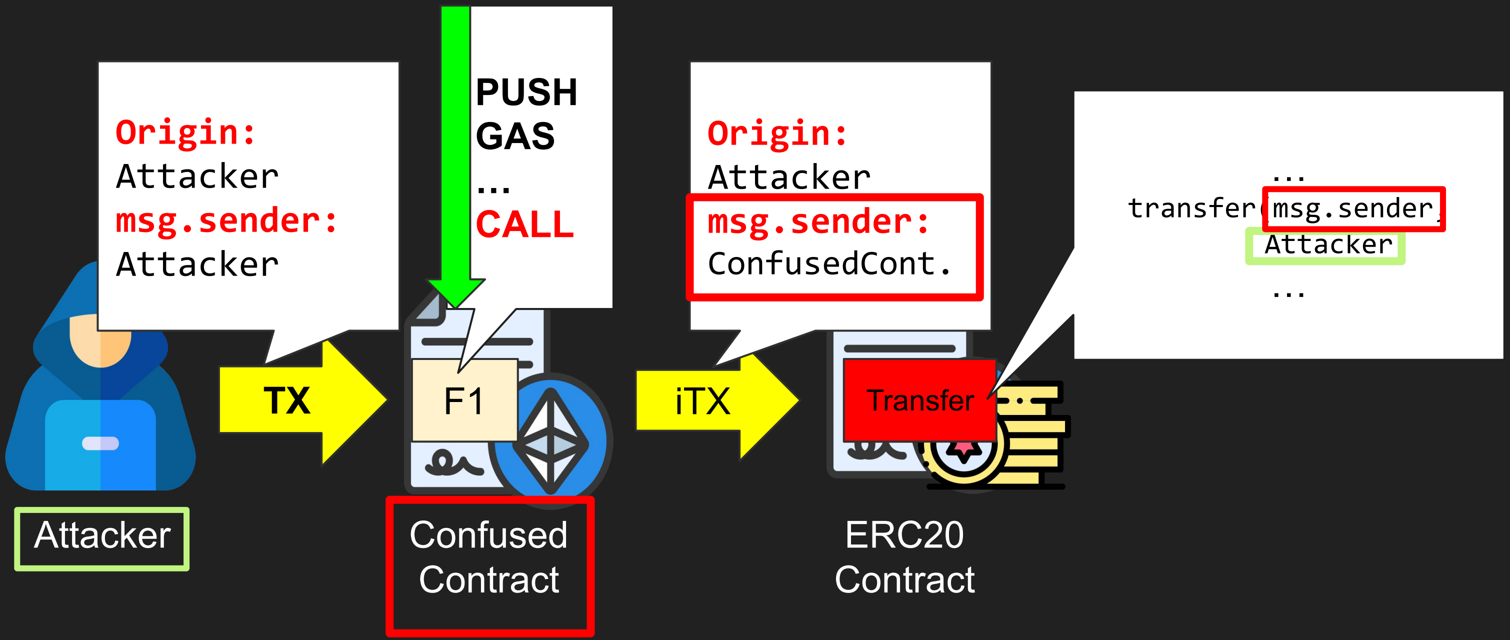


Origin:
Attacker
msg.sender:
Attacker

PUSH
GAS
...
CALL

Origin:
Attacker
msg.sender:
ConfusedCont.

Exploit



```
Origin:
Attacker
msg.sender:
Attacker
```

```
PUSH
GAS
...
CALL
```

```
Origin:
Attacker
msg.sender:
ConfusedCont.
```

```
        ...
transfer(msg.sender,
        Attacker
        ...
```

Evaluation

- 2,000,000+ smart contracts
    - Deployed between December 2020 → December 2022
- 529 potential Confused Contracts

Evaluation

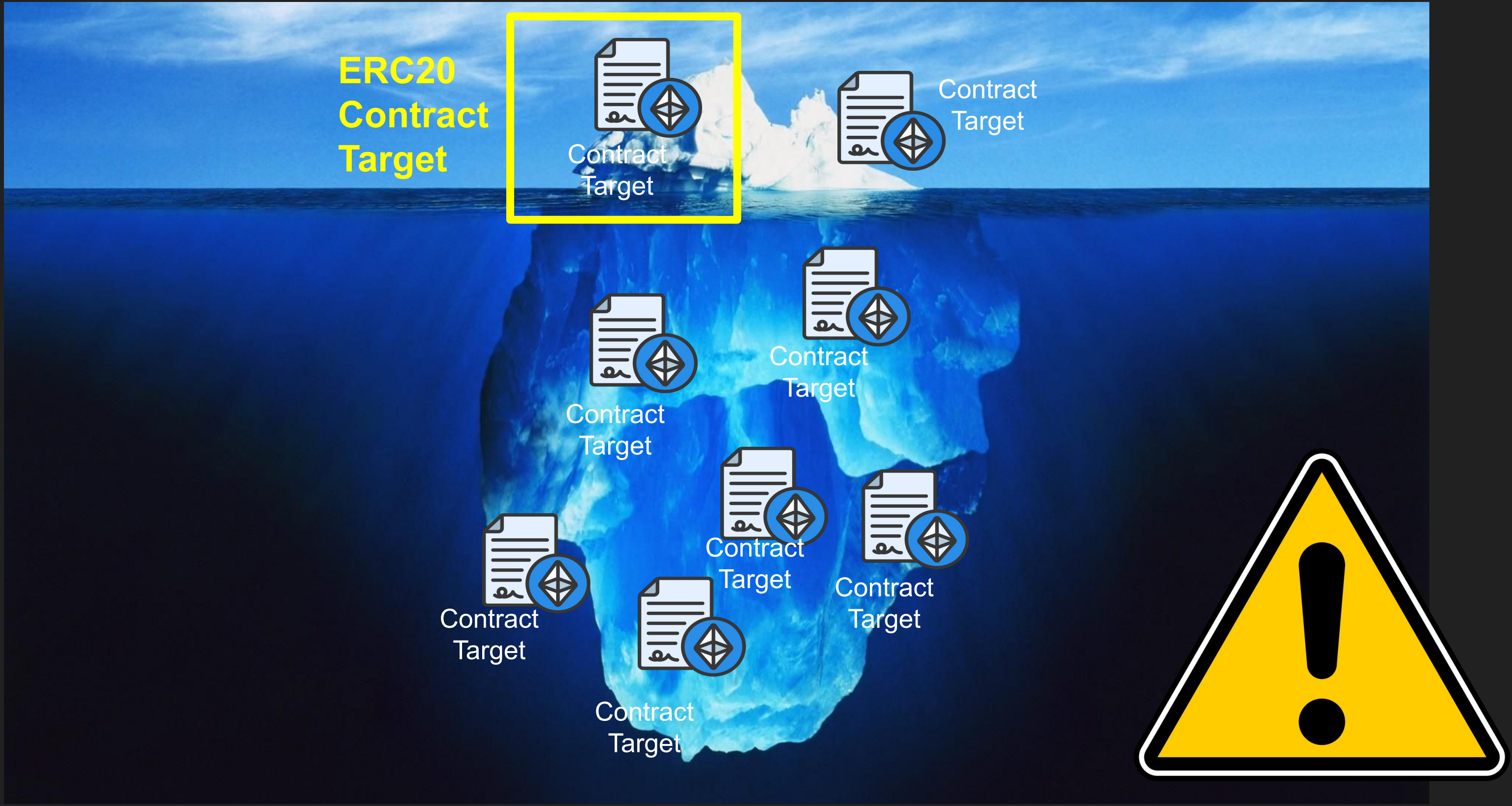
- 2,000,000+ smart contracts
  - Deployed between December 2020 → December 2022
- 529 potential Confused Contracts
  - **84** warnings Confused Contract + Contract Target

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



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