

About us



Automation Lead at [Lido](#)

Building solid on-chain and off-chain monitoring tools

Responsible for the quality of major Lido products

Prior to Lido worked for 6 years as a [Quality and Automation Engineer](#)

Quality assurance

Security testing



@d_gusakov



@Gusakov_dv



PhD in Computer Science from Victoria University of Wellington, NZ

Researcher-in-Residence at [the Forta Foundation](#)

Analyzing attacks

Working with community to strengthen attack detection

Prior to Forta worked at [Microsoft](#) for 14 years on the Defender product line

Security Research

Data Science



@cseifert



@christian_forta

Web3 is getting hacked!

Attack Stages

Inverse Finance

(\$1.2M)

Funding

Tornado Cash Funding

Preparation

Suspicious Contract
Creation

Exploitation

Flashloan
Flashbot Tx

**Money
Laundering**

Tornado Cash Money
Laundering

Tornado Cash Funding



Jun-16-2022 08:45:36 AM +UTC - <https://etherscan.io/address/0x7b792e49f640676b3706d666075e903b3a4deec6#internaltx>

Address **0x7b792E49f640676B3706d666075E903B3A4deEc6**

Exploit

Buy ▾

Exchange ▾

Earn ▾

Gaming ▾

Transactions Internal Txns Erc20 Token Txns Analytics Comments

⌵ Latest 4 internal transactions ☐

Parent Txn Hash	Block	Age	From	To	Value
0x84ee1ce4dd2aa5113a...	14972410	106 days 12 hrs ago	Tornado.Cash: 1 ETH	Inverse Finance Exploiter	0.975623 Ether

Suspicious Contract Creation

Jun-16-2022 08:47:50 AM +UTC - <https://etherscan.io/tx/0xfb5a4d1aef98458f673f301c2e713613662ad621e8f57065a4da58a6401c0b4d>



Transactions

For [0x7b792e49f640676b3706d666075e903b3a4deec6](#) Inverse Finance Exploiter

Sponsored: - 1inch - The most efficient DEX aggregator. Recover up to 95% of gas spendings. [Swap now!](#)

A total of 38 transactions found

First < Page 1 of 1 > Last :

Txn Hash	Method ^①	Block	Age	From	To	Value	Txn Fee
0xfb5a4d1aef98458f673f...	0x00806040	14972418	106 days 12 hrs ago	Inverse Finance Exploiter	Contract Creation	0 Ether	0.11407894

Transactions Internal Txns ERC20 Token Txns Contract Events Analytics Comments

Are you the contract creator? [Verify and Publish](#) your contract source code today!

Note: We also found another [2 contracts](#) with exact matching byte codes

[Decompile ByteCode](#)

[Switch to Opcodes View](#)

[Similar Contracts](#)

```
0x6080604052600436101095760003560e1c8063be9a655511610095578063daa589f411610064578063daa589f41461030e578063eee57d8c14610337578063ef30053614610360578063fe0d94c114610377578063ff
414b64146103a057610109565b8063be9a655514610275578063c5bea6c1461028c578063c7e42b1b146102b5578063ca96014b146102d157610109565b8063920f5c84116100dc578063920f5c84146101a25780639283fd
e1146101df578063a656bd0c146101f6578063ab829eaf14610221578063b31712241461024a57610109565b8063264e88931461010e578063468f02d214610137578063817e63cc1461016257806388a6eed91461018b575b
600080fd5b34801561011a57600080fd5b50610135600480360381019061013091906120d2565b6103dd565b005b34801561014357600080fd5b5061014c61085d565b6040516101599190612821565b60405180910390f35b
34801561016e57600080fd5b5061018960048036038101906101849190612083565b610904565b005b34801561019757600080fd5b506101a0610979565b005b3480156101ae57600080fd5b506101c9600480360381019061
01c4919061210e565b610b2d565b6040516101d691906127c6565b60405180910390f35b3480156101eb57600080fd5b506101f4610cc6565b005b34801561020257600080fd5b5061020b610fd6565b604051610218919061
2821565b60405180910390f35b34801561022d57600080fd5b5061024860048036038101906102439190612260565b610fdcd565b005b34801561025657600080fd5b5061025f611252565b60405161026c0190612821565b60
405180910390f35b34801561028157600080fd5b5061028a611258565b005b34801561029857600080fd5b506102b360048036038101906102a9190612260565b61126f565b005b6102cf60048036038101906102ca919061
1fa8565b611320565b005b3480156102dd57600080fd5b506102e7860048036038101906102f39190611fd1565b6115de565b6040516103059190612821565b60405180910390f35b34801561031a57600080fd5b5061033560
048036038101906103309190612034565b61169b565b005b34801561034357600080fd5b506103560048036038101906103599190611fd1565b611797565b005b34801561036c57600080fd5b506103756118565b6005b34
801561038357600080fd5b5061039e60048036038101906103999190612260565b611899565b005b3480156103ac57600080fd5b506103c760048036038101906103c29190611fa8565b611cf7565b6040516103d491906128
21565b60405180910390f35b60006000905490610100a900473ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
.....
```

Flashloan/ Flashbot Usage



Jun-16-2022 08:47:58 AM +UTC - <https://etherscan.io/address/0x7b792e49f640676b3706d666075e903b3a4deec6#internaltx>

Transaction Details



Flashbots ⓘ

Private Transaction ⓘ

Featured: Bridging tokens between Ethereum, Layer 2 and other chains? Browse through the Blockscan [bridges list](#).

Overview

Internal Txns

Logs (54)

State

Comments

Transaction Hash: 0x958236266991bc3fe3b77feaacea120f172c0708ad01c7a715b255f218f9313c

Status: Success

Block: 14972419 676319 Block Confirmations











Timestamp: 106 days 12 hrs ago (Jun-16-2022 08:47:58 AM +UTC) | Confirmed within 1 min

Transaction Action: Flash Loan 27,000 WBTC From Aave Protocol V2

Exploit Impact

Jun-16-2022 08:47:50 AM +UTC - <https://phalcon.blocksec.com/tx/eth/0x958236266991bc3fe3b77feaceae120f172c0708ad01c7a715b255f218f9313c>

Balance Changes

Address	Token	Balance	Value
0xf508c58ce37ce40a40997c715075172691f92e2d [Receiver]	 anYvCrv3Crypto	+245,337.73387519	-
	 USDT	+99,976.294967	-
	 WBTC	+53.24446541	-
anYvCrv3Crypto	 yvCurve-3Crypto	+4,906.75467750397441431	-
	 anYvCrv3Crypto	-245,337.73387519	-
Curve.fi: DAI/USDC/USDT Pool	 USDT	-10,099,976.294967	-
0x464c71f6c2f760dda6093dcb91c24c39e5d6e18c	 aWBTC	+0.00004148	-
aWBTC	 WBTC	+24.3	-
0xd51a-Vyper_contract	 WBTC	-77.54446541	-
	 USDT	+10,000,000	-

Money Laundering



Jun-16-2022 08:47:58 AM +UTC - <https://etherscan.io/tx/0x37e015682d3d989a90f7e47ee0c12a3bc58a96a671b6eeb9691e03e79ac179d4>

Transaction Details < | >

Flashbots ⓘ

Private Transaction ⓘ

Feature Tip: Add private address tag to any address under [My Name Tag](#) !

Overview

Internal Txns

Logs (13)

State

Comments

Transaction Hash:

0x37e015682d3d989a90f7e47ee0c12a3bc58a96a671b6eeb9691e03e79ac179d4

Status:

✓ Success

Block:

✓ 14972437 676459 Block Confirmations

Timestamp:

🕒 106 days 12 hrs ago (Jun-16-2022 08:52:48 AM +UTC) | ⌚ Confirmed within 1 min

💡 Transaction Action:

- ▶ Swap 42.59557232 WBTC For 785.666048274022634622 Ether On Uniswap V3
- ▶ Swap 7.98666981 WBTC For 148.22480221160635588 Ether On Uniswap V3
- ▶ Swap 2.66222327 WBTC For 57,811.229075 USDC On Uniswap V3
- ▶ Swap 57,811.229075 USDC For 49.413422222684903223 Ether On Uniswap V3

Money Laundering

Jun-16-2022 08:56:47 AM +UTC - <https://etherscan.io/tx/0xf9953c26d229c42938f681ce348322c92a5178965a6631a0f09fcadbac16a9d7>

Jun-16-2022 08:56:47 AM +UTC - <https://etherscan.io/tx/0xec27c61ae0c5a3f3f8a48bbb7b1f38781205ee1b8a978ee83e0b512c1bb6e22b>
















Transactions

For 0x7b792e49f640676b3706d666075e903b3a4deec6 Inverse Finance Exploiter

Sponsored:  - 1inch - The most efficient DEX aggregator. Recover up to 95% of gas spendings. [Swap now!](#)

A total of 38 transactions found

First < Page 1 of 1 > Last ⋮

Txn Hash	Method ^①	Block	Age	From	To	Value	Txn Fee
 0x0426bec92942e576ab...	Deposit	14972747	106 days 11 hrs ago	Inverse Finance Exploiter	  Tornado.Cash: Router	100 Ether	0.02585584
 0x7070db6c9893be0a04...	Deposit	14972743	106 days 11 hrs ago	Inverse Finance Exploiter	  Tornado.Cash: Router	100 Ether	0.02673976
 0xe8944881bab325997b...	Transfer Tokens	14972736	106 days 11 hrs ago	Inverse Finance Exploiter	  Wormhole: Portal Token ...	0 Ether	0.00365016
 0xe042ae41ce47f33458...	Deposit	14972736	106 days 11 hrs ago	Inverse Finance Exploiter	  Tornado.Cash: Router	100 Ether	0.0258806
 0xec27c61ae0c5a3f3f8a...	Approve	14972676	106 days 11 hrs ago	Inverse Finance Exploiter	  Wormhole: WBNB Token	0 Ether	0.00240049

Comprehensive **security** strategy

Pre-deployment

- Template contracts
- Audit

Post-deployment

- Bug bounties
- **Real-time monitoring and alerting**
- Incident/emergency response



Decentralized **security camera and
alarm system** for Web 3

Detection Bots

“Security cameras”

A script (piece of logic) that any developer can write and publish to the Forta Network.

Detection bots tell the network what to watch.

Scan Node

“Alarm system”

Runs the detection bots against each block of transactions.

The nodes power the detection bots, and keep them running 24/7.

Forta Explorer ([link](#))

★ Forta explorer

HomeBot SearchNetworkAirdropForta App

FORT token Airdrop is Live | Claim yours [here](#)

The Forta Network Explorer

Search by Bot ID / Address / Alert Name

Advanced Search

LatestAccess alerts data

All networksEthereumPolygonBSCAvalancheOptimismFantomArbitrum

Daily Alerts

Severity	Alert	Details
Medium	Borrow amount close to total supply	0x354D6F37482C3e533b0d3AaF6557083D72C3
High	Flashloan detected	0x9e2634834e27189f7c257081f47010e655d4c361
High	High Gas Price	Gas price in transaction
Medium	Drastic liquidity change for time period	Market ICVX had a 1635.04% change in total borrow
Medium	Lido Beacon rewards decreased	Rewards decreased from 510.923 ETH to 508.966 b
High	Low balance of Lido account on The Graph	Balance is 517.37 GRT. It is too low!
Medium	DEFAULT_ADMIN Role Renounced	Account 0x9B2416B99D3c2545Aa0c5989FEb201e
Critical	Poly Cross Chain Event With Wrong Parameters	The parameters of the CrossChainEvent and the poly
High	stMATIC rewards decreased	stMATIC rewards has decreased by -917% from 611
Medium	Large exit from pool tTUSD	Account 0x5Ccd441F9051e845b6C7BF8472cd8cf
High	UUPSUpgradable contract self-destroyed	UUPSUpgradable contract on 0xfcc5a1c4bd06e44
High	UMA Deployer Watch - Unexpected Transaction	UMA Deployer transaction with non-whitelist address

★ Forta explorer

HomeBot SearchNetworkAirdropForta App

FORT token Airdrop is Live | Claim yours [here](#)

Network Activity

Information about the health and activity of the Forta Network, including blockchain coverage and active node operators

Name	Node operators	Detection Bots
1 Ethereum	1782	566
2 Polygon	1454	178
3 BSC	105	70
4 Avalanche	96	56
5 Optimism	89	35
6 Fantom	47	27
7 Arbitrum	87	23

Attack Stages

Funding

- TC funding
- Exchange funding
- New account
- Bridge funding

Preparation

- Sleep minting
- Attack contract creation
- Ice Phishing Token Approvals
- Token impersonation

Exploitation

- Flashloan price manipulation
- Flashbot usage
- Ice Phishing Token Transfers
- Rug pulls
- Exploit (reentrancy, failed access control, etc.)

Money Laundering

- TC deposits
- Exchange deposits
- Exchange into native tokens
- Bridge deposits
- Wash trading

Attack Stages



Inverse Finance (\$1.2M)



Attack Exploration (link)



Discover new attacks

You are in **semi-automatic** attack detection mode.

Please, select the network, the time period, and the set of bots to be analyzed.

⚡ Mainnet ▾ 14/09/2022 - now ✕ 🗺 Forta General Kit ▾ **Analyze**

Group **1/10618**



Stage passed **4/4**

Last stage **Laundering**

Alerts



⚡ Mainnet ▾ 14/09/2022 - now ✕ 🗺 Forta General Kit ▾ ✕ ⭐ Presets ▾ + Add filter

📄 0xe74b28c2eae8679e3ccc3a94d5d0de83ccb84705 14 ⋮

☰ Latest 20 from a total of 218 alerts

High Gas Use Detection	Gas Used by Transaction		28 Sep 05:22:34
Tornado Cash funded account interacted with contract	0xdf120a3e38ba6bf32752e47d2d827c1ddd848bab interacted with contract 0x14d1b27d79e97e96622618f9d4fa9b1e1e9ef082	Funding	28 Sep 05:22:37
A text message has been sent	You are a hero. This is top news. I applaud you. A lot of people visit your wallet on etherscan. I will never blame you. I need your help, so would you have the courage to send me the funds? I will start a company and share...	Laundering	23 Sep 19:31:34
A text message has been sent	We want to cooperate with you and resolve this matter immediately. Accept the terms of the bounty and return the funds within 24 hours before September 22nd UST by 23:59 while we can still consider this a white-hat...	Laundering	21 Sep 17:19:09
Address has native token transfer	value: 1 to: 0xe74b28c2eae8679e3ccc3a94d5d0de83ccb84705		21 Sep 17:19:02
Alert combiner identified an EOA with past alerts mapping to attack behavior	{'attacker_address': '0xe74b28c2eae8679e3ccc3a94d5d0de83ccb84705'} likely involved in an attack		21 Sep 11:26:11
Alert combiner identified an EOA with past alerts mapping to attack behavior	{'attacker_address': '0xe74b28c2eae8679e3ccc3a94d5d0de83ccb84705'} likely involved in an attack		21 Sep 08:51:45
A text message has been sent	bro, return the funds you don't need , protect this industry .	Laundering	21 Sep 01:31:15

Flash Loan

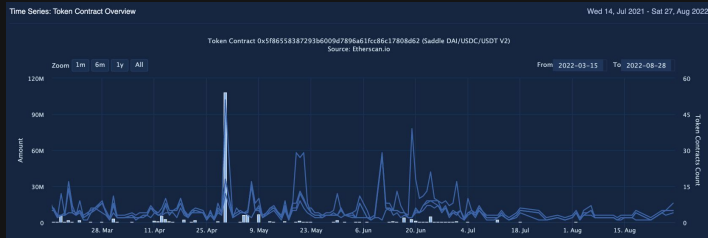
BotID: 0x55636f5577694c83b84b0687eb77863850c50bd9f6072686c8463a0cbc5566e0

Attack Stage: **Exploitation**

What is It?

Flash loans are temporary loans (has to be paid back in one transaction) that allows borrowers to obtain large amount of tokens. This is used, for example, for arbitrage.

Flash loans are also utilized by attackers to manipulate prices temporarily. This could be used to exploit a vulnerable protocol (e.g. by taking out an undercollateralized loan)



How to detect?

Identify all transactions obtaining a flash loan

Assess whether profit exceeds a particular threshold.

★ Forta explorer Home Bot Search Network Airdrop | Forta App

FORT token Airdrop is Live | Claim yours here

Alert 0xbaf96ca34e7e626402c983dffa15d82e5e1af3d4e24a5c676574e3bca3dca on ethereum network

Alert Flashloan detected High Severity

Alert id 0xbaf96ca34e7e626402c983dffa15d82e5e1af3d4e24a5c676574e3bca3dca (FLASHLOAN-ATTACK-WITH-HIGH-PROFIT)

Transaction 0xbaf96ca34e7e626402c983dffa15d82e5e1af3d4e24a5c676574e3bca3dca

Block 15435913

Bot ID 0x55636f5577694c83b84b0687eb77863850c50bd9f6072686c8463a0cbc5566e0

Timestamp Mon, 29 Aug 2022 19:24:20 GMT

Description 0xaf29832f6a244137a0b810be1a8c2b461f1c9 launched flash loan attack and made profit > \$100000

Rug Pulls

BotID: 0x580d14bed37f523d14edcfa83ae87e168ac333a98f70c4f9991357e1b7ee855f

Attack Stage: **Exploitation**

What is It?

Rug pulls are tokens that are hyped by creators. As they are traded on DEXes, creators may dump existing tokens or dump newly created tokens. The price crashes and remaining token holders are left holding the bag of worthless tokens.

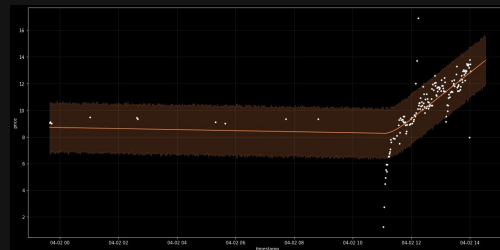


How to detect?

Obtain price information from DEXes (utilizing common ABI)

Trigger on significant price drops.

Price fluctuations are common, however. In order to reduce noise, one needs to apply a time series anomaly detection approach that takes into account historical information (seasonality, volatility).



Exploit Simulation

BotID: 0xe8527df509859e531e58ba4154e9157eb6d9b2da202516a66ab120deabd3f9f6

Attack Stage: **Preparation**

What is It?

For certain attacks (e.g. economic attacks, reentrancy attacks), attackers need to deploy a smart contract. Several indicators can help to determine whether a smart contract is malicious (e.g. was it created through Tornado Cash funded EOA; is it verified on Etherscan?)

These contracts contain all the code that is needed to execute the exploit.

How to detect?

Upon smart contract deployment, locally fork the chain using Ganache. Invoke all exposed functions (essentially fuzzing the smart contract).

```
PUSH1 0x80
PUSH1 0x40
MSTORE
PUSH1 0x04
CALLDATASIZE
LT
PUSH2 0x002d
JUMPI
PUSH1 0x00
CALLDATALOAD
PUSH1 0xe0
SHR
DUP1
PUSH4 0xa15db5c5
EQ
PUSH2 0x0039
JUMPI
DUP1
PUSH4 0xa18271f7
```

Assess whether large amounts of tokens are transferred into the attacker's wallet or contract.

Identifies the attack before it is executed on-chain.

Custom Bot Development

Define the Requirements

Step 1

Based on threat model define the requirements:

- What is the logic?
- What alerts will the bot emit?
- What data do you need?
- What chains should the bot run on? What differences exist between the chains?

Implement and Test

Step 2

Implement using Python/
JavaScript/ TypeScript SDK

Test using unit tests

Test retroactively on existing attack
transactions/ blocks

Execute locally against live
transaction feed

Deploy and Subscribe

Step 3

Deploy in a permissionless way to the Forta Network using the CLI or Forta App. It will be deployed onto several nodes to create redundancy and increase alert reliability.

Log and alert data for the bot can be viewed and monitored through bot stats page.

Alert subscriptions can be configured to receive alerts on Telegram, Slack, Discord and accessed through the GraphQL API.



Get started at:

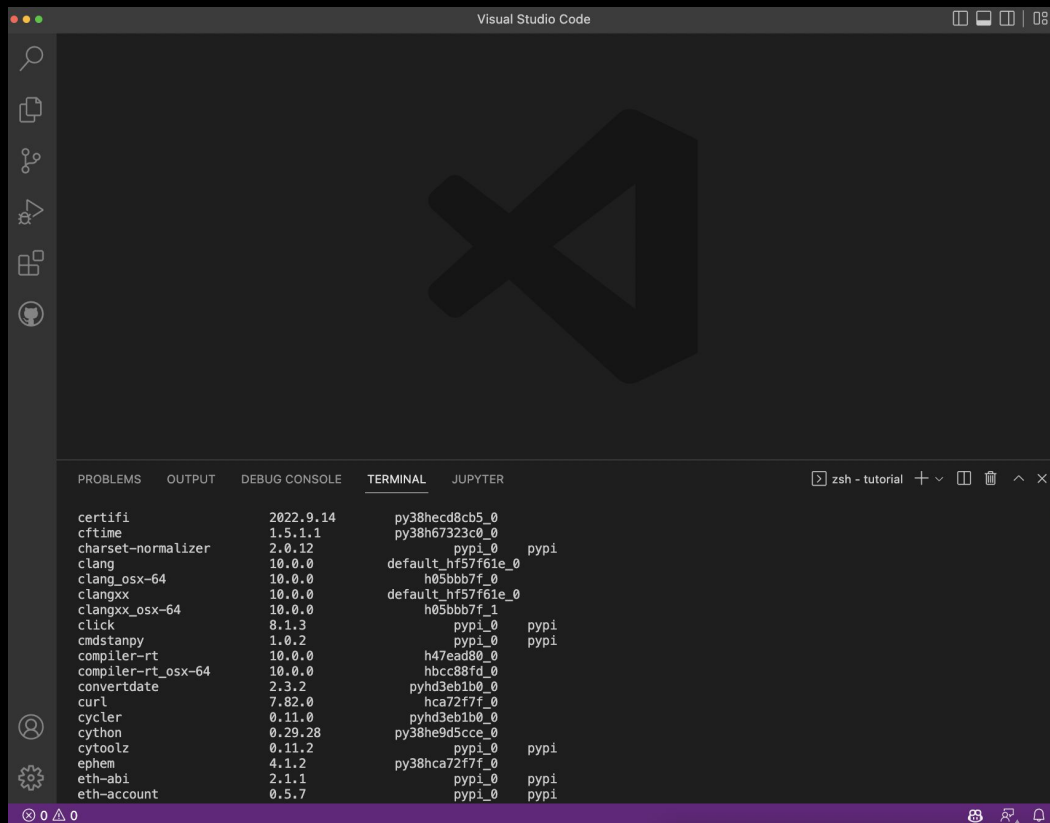
<https://docs.forta.network/en/latest/quickstart/>

Custom Bot Development

Setup Dev Environment

Step 1

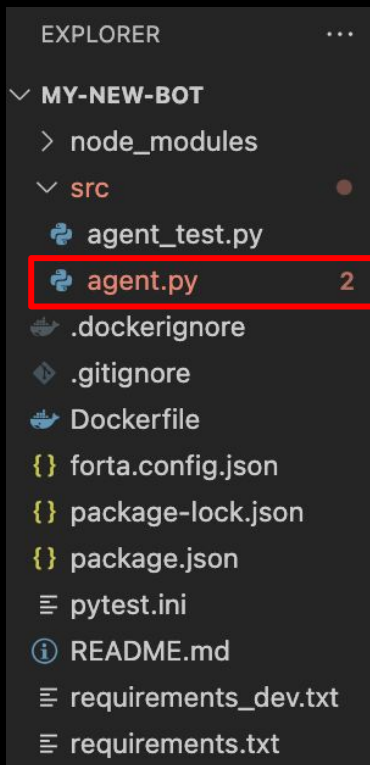
- Node.js v12+ (which includes the Node package manager i.e. npm)
- Conda & Python v3.6+ (only if you want to use Python SDK)
- Docker v20+



Custom Bot Development

Initialize Bot Step 2

```
$ mkdir my-new-bot  
$ cd my-new-bot  
$ npx forta-agent@latest init  
--python
```

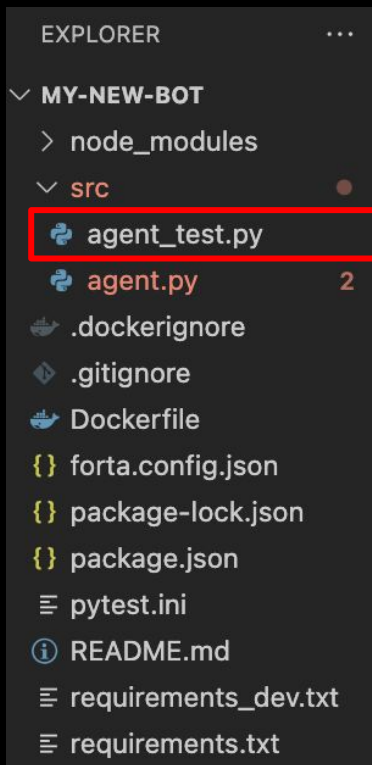


Main Bot Logic

Custom Bot Development

Initialize Bot Step 2

```
$ mkdir my-new-bot  
$ cd my-new-bot  
$ npx forta-agent@latest init  
--python
```

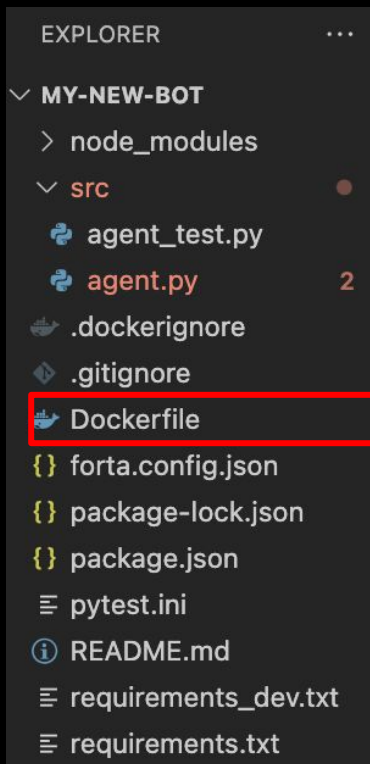


Unit tests

Custom Bot Development

Initialize Bot Step 2

```
$ mkdir my-new-bot  
$ cd my-new-bot  
$ npx forta-agent@latest init  
--python
```

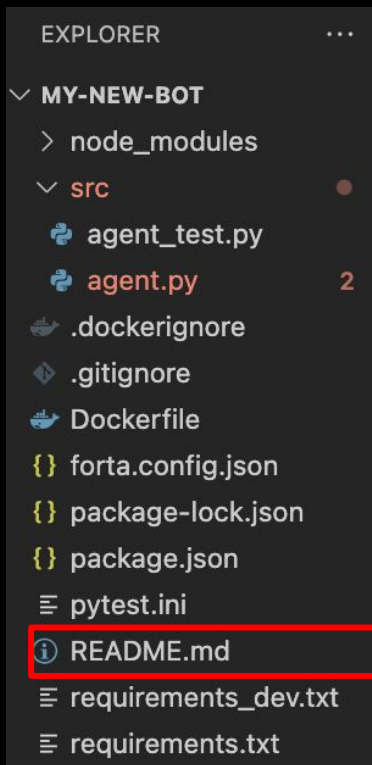


Docker build file

Custom Bot Development

Initialize Bot Step 2

```
$ mkdir my-new-bot  
$ cd my-new-bot  
$ npx forta-agent@latest init  
--python
```



Documentation

Custom Bot Development

Create Documentation Step 3

Capture

- Title
- Description
- Supported Chains
- Alerts
- Test Data

```
1  # Large Tether Transfer Agent
2
3  ## Description
4
5  This agent detects transactions with large Tether transfers
6
7  ## Supported Chains
8
9  - Ethereum
10 - List any other chains this agent can support e.g. BSC
11
12 ## Alerts
13
14 Describe each of the type of alerts fired by this agent
15
16 - FORTA-1
17   - Fired when a transaction contains a Tether transfer over 10,000 USDT
18   - Severity is always set to "low" (mention any conditions where it could be
19     something else)
20   - Type is always set to "info" (mention any conditions where it could be something
21     else)
22   - Mention any other type of metadata fields included with this alert
23
24 ## Test Data
25
26 The agent behaviour can be verified with the following transactions:
27
28 - 0x3a0f757030beec55c22cbc545dd8a844cbbb2e6019461769e1bc3f3a95d10826 (15,000 USDT)
```

Custom Bot Development

Implement Bot Logic

Step 4

- Process Tx
- Filter for USDT events
- Normalize value
- Assess against threshold
- Emit alert

```
def handle_transaction(transaction_event) -> list:
    findings = []

    # limiting this agent to emit only 5 findings so that the alert feed is not spammed
    global findings_count
    if findings_count >= 5:
        return findings
```

Main Function

Custom Bot Development

Implement Bot Logic

Step 4

- Process Tx
- Filter for USDT events
- Normalize value
- Assess against threshold
- Emit alert

```
def handle_transaction(transaction_event) -> list:
```

```
    findings = []
```

Returns a list of findings

```
    # limiting this agent to emit only 5 findings so that the alert feed is not spammed
```

```
    global findings_count
```

```
    if findings_count >= 5:
```

```
        return findings
```

```
        findings.append(Finding({
            'name': 'High Tether Transfer',
            'description': f'High amount of USDT transferred: {normalized_value}',
            'alert_id': 'FORTA-1',
            'severity': FindingSeverity.Low,
            'type': FindingType.Info,
            'metadata': {
                'to': to,
                'from': from_,
            }
        })))
        findings_count += 1
```

```
    return findings
```

Custom Bot Development

Implement Bot Logic

Step 4

- Process Tx
- Filter for USDT events
- Normalize value
- Assess against threshold
- Emit alert

```
def handle_transaction(transaction_event) -> list:
    findings = []

    # limiting this agent to emit only 5 findings so that the alert feed is not spammed
    global findings_count
    if findings_count >= 5:
        return findings
```

Findings contain all pertinent information

```
findings.append(Finding({
    'name': 'High Tether Transfer',
    'description': f'High amount of USDT transferred: {normalized_value}',
    'alert_id': 'FORTA-1',
    'severity': FindingSeverity.Low,
    'type': FindingType.Info,
    'metadata': {
        'to': to,
        'from': from_,
    }
}))
findings_count += 1
```

return findings

Custom Bot Development

Implement Bot Logic

Step 4

- Process Tx
- Filter for USDT events
- Normalize value
- Assess against threshold
- Emit alert

```
ERC20_TRANSFER_EVENT = '{"name":"Transfer","type":"event","anonymous":false,"inputs":['  
TETHER_ADDRESS = '0xdAC17F958D2ee523a2206206994597C13D831ec7'  
TETHER_DECIMALS = 6
```

```
# filter the transaction logs for any Tether transfers
```

```
tether_transfer_events = transaction_event.filter_log(  
    ERC20_TRANSFER_EVENT, TETHER_ADDRESS)
```

Event Filter

```
for transfer_event in tether_transfer_events:
```

```
    # extract transfer event arguments
```

```
    to = transfer_event['args']['to']
```

```
    from_ = transfer_event['args']['from']
```

```
    value = transfer_event['args']['value']
```

```
    # shift decimals of transfer value
```

```
    normalized_value = value / 10 ** TETHER_DECIMALS
```

```
# if more than 10,000 Tether were transferred, report it
```

```
if normalized_value > 10000:
```

```
    findings.append(Finding({
```

Custom Bot Development

Implement Bot Logic

Step 4

- Process Tx
- Filter for USDT events
- Normalize value
- Assess against threshold
- Emit alert

```
ERC20_TRANSFER_EVENT = '{"name":"Transfer","type":"event","anonymous":false,"inputs":['  
TETHER_ADDRESS = '0xdAC17F958D2ee523a2206206994597C13D831ec7'  
TETHER_DECIMALS = 6
```

```
# filter the transaction logs for any Tether transfers  
tether_transfer_events = transaction_event.filter_log(  
    ERC20_TRANSFER_EVENT, TETHER_ADDRESS)  
  
for transfer_event in tether_transfer_events:  
    # extract transfer event arguments  
    to = transfer_event['args']['to']  
    from_ = transfer_event['args']['from']  
    value = transfer_event['args']['value']  
    # shift decimals of transfer value  
    normalized_value = value / 10 ** TETHER_DECIMALS  
  
    # if more than 10,000 Tether were transferred, report it  
    if normalized_value > 10000:  
        findings.append(Finding({
```

Assess for
condition

Custom Bot Development

Implement Bot Logic

Step 4

- Process Tx
- Filter for USDT events
- Normalize value
- Assess against threshold
- Emit alert

```
ERC20_TRANSFER_EVENT = '{"name":"Transfer","type":"event","anonymous":false,"inputs":['  
TETHER_ADDRESS = '0xdAC17F958D2ee523a2206206994597C13D831ec7'  
TETHER_DECIMALS = 6
```

```
# filter the transaction logs for any Tether transfers  
tether_transfer_events = transaction_event.filter_log(  
    ERC20_TRANSFER_EVENT, TETHER_ADDRESS)  
  
for transfer_event in tether_transfer_events:  
    # extract transfer event arguments  
    to = transfer_event['args']['to']  
    from_ = transfer_event['args']['from']  
    value = transfer_event['args']['value']  
    # shift decimals of transfer value  
    normalized_value = value / 10 ** TETHER_DECIMALS  
  
    # if more than 10,000 Tether were transferred, report it  
    if normalized_value > 10000:  
        findings.append(Finding({
```

Emit finding

Custom Bot Development

Test, test, test

Step 5

- Unit test
- Backtest
- Live test

```
• (base) christianseifert@christians-mbp my-new-bot % conda activate forta
• (forta) christianseifert@x86_64-apple-darwin13 my-new-bot % npm run test
```

```
> forta-agent-starter@0.0.1 test
> python3 -m pytest
```

```
===== test session starts =====
platform darwin -- Python 3.8.13, pytest-6.2.5, py-1.11.0, pluggy-1.0.0
rootdir: /Users/christianseifert/forta/tutorial/my-new-bot, configfile: pytest.ini
plugins: env-0.6.2, web3-5.23.0
collected 2 items

src/agent_test.py .. [100%]

===== 2 passed in 0.11s =====
```

```
○ (forta) christianseifert@x86_64-apple-darwin13 my-new-bot %
```

Custom Bot Development

Test, test, test Step 5

- Unit test
- **Backtest**
- Live test

Test Data

The agent behaviour can be verified with the following transactions:

```
- 0x3a0f757030beec55c22cbc545dd8a844cbbb2e6019461769e1bc3f3a95d10826 (15,000 USDT)
○ (forta) christianseifert@x86_64-apple-darwin13 my-new-bot % npm run tx 0x3a0f757030beec55c22cbc545dd8a844cbbb2e6019461769e1bc3f3a95d10826

> forta-agent-starter@0.0.1 tx
> forta-agent run --tx 0x3a0f757030beec55c22cbc545dd8a844cbbb2e6019461769e1bc3f3a95d10826

1 findings for transaction 0x3a0f757030beec55c22cbc545dd8a844cbbb2e6019461769e1bc3f3a95d10826 {
  "name": "High Tether Transfer",
  "description": "High amount of USDT transferred: 15000.0",
  "alertId": "FORTA-1",
  "protocol": "ethereum",
  "severity": "Low",
  "type": "Info",
  "metadata": {
    "to": "0x191a95DaC026F3A002C66e6C61C484FAb9D65D17",
    "from": "0x02f4F75Ce4498CAfFEA57f5ab0F7D7831D6B1fC6"
  },
  "addresses": []
}
```

Custom Bot Development

Test, test, test Step 5

- Unit test
- Backtest
- Live test

```
Ⓢ (forta) christianseifert@x86_64-apple-darwin13 my-new-bot % npm run start
```

```
> forta-agent-starter@0.0.1 start  
> npm run start:dev
```

```
> forta-agent-starter@0.0.1 start:dev  
> nodemon --watch src --watch forta.config.json -e py --exec "forta-agent run"
```

```
[nodemon] 2.0.20  
[nodemon] to restart at any time, enter `rs`  
[nodemon] watching path(s): src/**/* forta.config.json  
[nodemon] watching extensions: py  
[nodemon] starting `forta-agent run`  
listening for blockchain data...  
fetching block 15677316...  
0 findings for transaction 0xeee697d5b82e2351dc301a8e6fba58b5298ef8e38e0f7144c33961137a8f2530  
0 findings for transaction 0xae33ad2449e4583e56fdbed4e2e0555f962afb8fcc891348b0c1b91c921d4820  
0 findings for transaction 0x7a98b034f1bef87627c555e46afe29e3e2400000e05b87b29ae260c0d7523c2  
0 findings for transaction 0x4f9cd992f3c3992c7b5267fa8120862a3824a8958cfa6b15ee94c090a9ebedad  
0 findings for transaction 0xdaf21ba63d6c37a6b779bc9ffade7fc6dfe53bf0bdcd3693063500ce172e20b7  
0 findings for transaction 0xeb96067823e1846b137690f6aeabb73adb273d5375468336c193c03a1bd1021a  
0 findings for transaction 0x4ae4b2375b52dfd18bdf525cc5d52a96bcde283d807504cd3a707deb7ca2e0de  
0 findings for transaction 0x7bd1baf996a60c1718e8eb78b49cce0abb8cfd0ceb5f7fc8fef93cdf78cce38f  
0 findings for transaction 0x79116d3c5e0aeb289389fd11008bd1b779ce7927e4be42018f33f28c1aff58e7  
0 findings for transaction 0xac06643b920a3af8821395066df17f33b0dfe78d9752358343f8853e6ff5c3ad  
0 findings for transaction 0xd2a7c1ec929706d9a6858ccc9ca6df7fba1b0e30d67aa2e9d249a5b8bd561738  
0 findings for transaction 0x6c78fb38e111763d35b1868845a22c8f89c8b88b9fc70ec872ace165955bc4fe  
0 findings for transaction 0xa4f45e542b9afa6b52155a6e87d217b41f50d33e9e8f2c5b7fa51cae5419922  
0 findings for transaction 0x0fa0f938b74606a7940bff943922d8dbc0b5eea5e4a70849e04d2a96a645d72c  
0 findings for transaction 0xbef8ab66a94ed1c790b9e8ad21fb66aec1d9c3661401a66c68795e7dadb5094c  
0 findings for transaction 0xcf8c43a2b599fdb14a3453f27ab8d9cb5c9a44b31f5cf9d7ad4068b7349a0b80  
0 findings for transaction 0xf52fab238f460f6cceeef806669c0ba73aa28b3fa33424d89a7b1fb8bf4212e3c  
0 findings for transaction 0x140be7900923842db6e8d7471f578cbdeb9e70e3b50f9f2c0a671f453ed2f39
```

Custom Bot Development

Deploy Step 6

- Deploy using CLI
 - npm run publish
- Need some MATIC

```
pushing agent image to repository...
Using default tag: latest
The push refers to repository [disco.forta.network/forta-agent-starter-intermediate]
a3bd9b377696: Preparing
a3d39cbd2c5e: Preparing
74c6c5766a92: Preparing
debf465db75d: Preparing
5961f4982fbd: Preparing
21f246e23bd3: Preparing
7f30cde3f699: Preparing
fe810f5902cc: Preparing
dfd8c046c602: Preparing
4fc242d58285: Preparing
fe810f5902cc: Waiting
4fc242d58285: Waiting
dfd8c046c602: Waiting
7f30cde3f699: Waiting
21f246e23bd3: Waiting
debf465db75d: Pushed
74c6c5766a92: Pushed
a3d39cbd2c5e: Pushed
7f30cde3f699: Mounted from bafybeidkrwu6dii3aki2gb2m27b57huberunwrpnrxgxnuqbe7jgfvwtwi
fe810f5902cc: Mounted from bafybeidkrwu6dii3aki2gb2m27b57huberunwrpnrxgxnuqbe7jgfvwtwi
dfd8c046c602: Mounted from bafybeidkrwu6dii3aki2gb2m27b57huberunwrpnrxgxnuqbe7jgfvwtwi
4fc242d58285: Mounted from bafybeidkrwu6dii3aki2gb2m27b57huberunwrpnrxgxnuqbe7jgfvwtwi
a3bd9b377696: Pushed
5961f4982fbd: Pushed
21f246e23bd3: Pushed
latest: digest: sha256:095bfb7149587f8fc40bccbed553bc1998715ec23f78f8aba56a59efc34b48c2 size: 2418
bafybeidifuk5aytsp3s55z4e3sgfe5jg7qlh55l5jn436oiby5kd7cnf44: Pulling from 095bfb7149587f8fc40bccbed553bc1998715ec23f78f8aba56a59efc34b48c2
Digest: sha256:095bfb7149587f8fc40bccbed553bc1998715ec23f78f8aba56a59efc34b48c2
latest: Pulling from 095bfb7149587f8fc40bccbed553bc1998715ec23f78f8aba56a59efc34b48c2
Digest: sha256:095bfb7149587f8fc40bccbed553bc1998715ec23f78f8aba56a59efc34b48c2
Status: Downloaded newer image for disco.forta.network/095bfb7149587f8fc40bccbed553bc1998715ec23f78f8aba56a59efc34b48c2
disco.forta.network/095bfb7149587f8fc40bccbed553bc1998715ec23f78f8aba56a59efc34b48c2
```

Custom Bot Development

Deploy Step 6

- Deploy using CLI
 - npm run publish
- Need some MATIC

forta-agent-starter v0.0.1

ENABLED

+ Add Stake

developed by 0x2835a787d8d724181F97ec97d9882dD7b3F2be

BOT INFORMATION

Bot ID	0x135f...e3ab
Image	bafybe...8941 ↗
Created	a day ago
Last Updated	a day ago
Networks Scanned	Ethereum

STAKED • TOTAL

0 FORT

Active Stake	0 FORT
Inactive Stake	0 FORT
Slashed	0 FORT
Node Operators	5
Frozen Stake	No

[Documentation](#)

[Log Data](#)

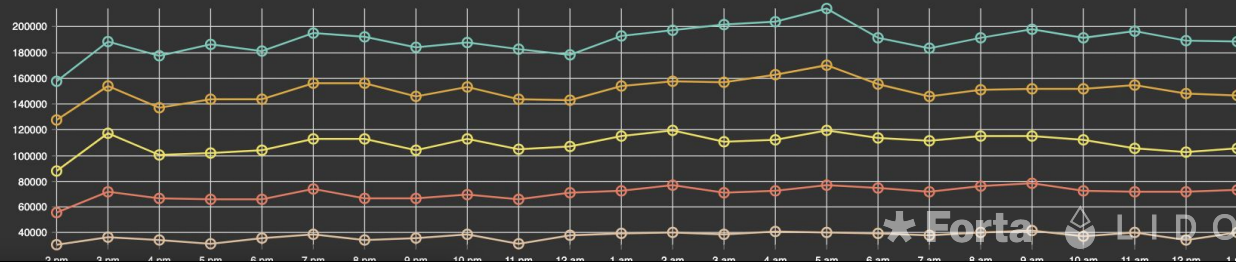
[Subscribe to this bot](#)

Detection Bot Health

Last 24h

Transactions: Received (count) | ▾

Number of transactions the bot was sent by the node operator.



Custom Bot Development

Subscribe Step 7

- Various mechanisms:
 - Telegram
 - Email
 - Slack
 - Discord
 - Webhook

forta-agent-starter v0.0.1 ENABLED

developed by 0x2835a787d8d74724181F97ec97d9882dD7b3F2be

[Add Stake](#)

BOT INFORMATION

Bot ID	0x135f...e3ab
Image	bafybe...8941 🔗
Created	a day ago
Last Updated	a day ago
Networks Scanned	Ethereum

STAKED • TOTAL

0 FORT

Active Stake	0 FORT
Inactive Stake	0 FORT
Slashed	0 FORT
Node Operators	5
Frozen Stake	No

[Documentation](#)

[Log Data](#)

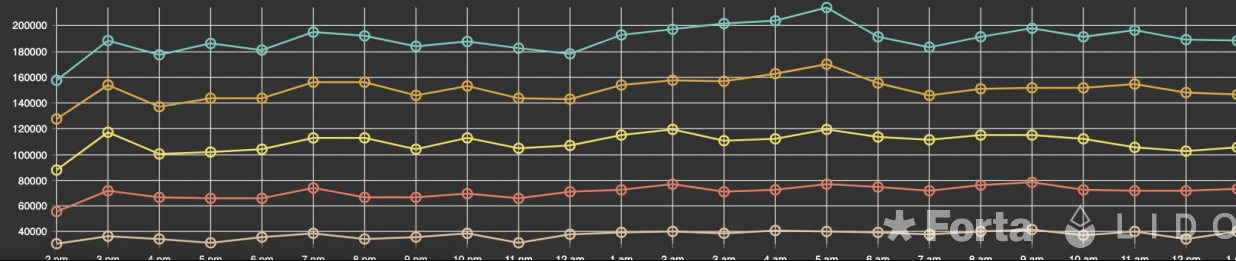
[Subscribe to this bot](#)

Detection Bot Health

Last 24h

Transactions: Received (count) | ▾

Number of transactions the bot was sent by the node operator.



Custom Bot Development

Subscribe Step 7

- Various mechanisms:
 - Telegram
 - Email
 - Slack
 - Discord
 - Webhook

Add Subscription

This form is for being notified every time there's an alert related to some address or detection bot.

Select Subscription type

Single Bot
Subscribe to alerts of a single bot.

Bot to watch

Enter Bot ID...

Notify me via

Email

Slack

Discord

Telegram

Webhook

Enter Email Address...

Subscribe

Bot Development Exercises

Exercise 1

Detect large USDC transfers

- Simple operational bot alerting on large (10K+) USDC transfers
- Filter for Events (ERC20_TRANSFER/USDC Token Contract)
- Normalize value
- Threshold on the value

Exercise 2

Detect low balances

- Bot to identify low balances. Allows to monitor your own address
- Assess balance with each block/tx
- Alert when it falls below a threshold
- Cache so you don't receive alert barrage

Exercise 3

Flashloan resulting in losses in Yearn Dai Vault

- Assess whether flashloan protocol and Vault were touched in tx
- Assess whether flashloan was obtained
- Assess vault balance before and this block to derive difference
- Threshold on diff and alert



Get started at:

<https://github.com/forta-network/forta-bot-workshop>

Bot Development Contest



Context 9

Identify attacked protocol

- Win up to 3,000 USD
- Alerts today identify attacks and expose slew of addresses involved in the transaction. Post analysis needs to be performed to identify what protocol was attacked
- Bot's goal is to identify the protocol attacked (e.g. by analyzing token transfers)

<https://docs.forta.network/en/latest/contest9-forta/>

Part 2.

Let's look at the **protocol code**

Why we need **protocol alerting**?

	Generic alerting	Dedicated alerting
Allow detection of generic attacks?	Yep	Kinda
Gives confidence in YOUR protocol safety	50 / 50	> 90%
Main attack vectors detected	Both generic and specific attacks	Generic changes and uncertainties in the protocol, but not attacks itself
Set-up	You can use existing bots	You need to build a bot yourself

We need them ALL

Typical protocol alerts

Operations

Repetitive events

- ◆ Oracle reports
- ◆ Rewards distribution
- ◆ Funds deposits
- ◆ DAO Voting
- ◆ ...

Predictions of possible issues in operations

- ◆ Sloppy oracles
- ◆ Low balance of executors
- ◆ Unexpected funds movements
- ◆ Unexpected vote content
- ◆ ...

Typical protocol alerts

Security

Inconsistency in protocol invariants

- ◆ Bridge balance difference (bridge hack)
- ◆ Issuing of the tokens with no actual backup (protocol hack)
- ◆ Minting NFTs with no actual backup (protocol hack)
- ◆ Unexpected fund transfers (protocol ownership loss)
- ◆ ...

Typical protocol alerts

Security

Events that should not happen silently

- ◆ Huge withdrawals
- ◆ Huge balance changes
- ◆ ...

ACL changes

- ◆ Role granted/revoked
- ◆ Ownership transferred
- ◆ ...

Events that should never happen

- ◆ Roles or ownership transfers to the EOA or null address
- ◆ Self-destruct of the protocol contract
- ◆ Changes in immutable slots values
- ◆ ...

Let's go deeper!
Practical examples

When we start?

You should start thinking about alerts and analyzing code before the deployment

ADR

Review contracts
architecture for
early issues
detection

Develop

Add stuff necessary
for proper alerting.
Events, view
methods, etc.

Review

Check that we have
all we need for
monitoring and
alerting

Deploy

Develop and deploy
detection bots.
Set-up alerting
channels

Defining critical events



ACL changes

Each ACL change should be alerted

Granting or revoking critical permissions should be supplied with the critical alerts

Ownership transfers

To EOA

To unknown contract

To Null address

State changes

Ordinary state changes

Huge changes in ordinary state

Critical state changes

Most common ACL contracts

`@openzeppelin/contracts/access/AccessControl.sol`

`@openzeppelin/contracts/access/Ownable.sol`

Defining protocol invariants

Amount minted = Amount deposited

Source bridge balance \geq Target bridge balance

Collateral value $>$ Loan value

...

Defining **repetitive events** and ways to predict issues with it

Repetitive Event	Possible issues	How to predict
Oracle report	Quorum not reached	Monitor quorum participation and difference in the reports
Rewards distribution	Rewards are not distributed in time	Off chain executor monitoring
Stake deposits	Huge amount of funds in buffer	Off chain executor monitoring
Validator keys upload	New keys are not uploaded	Monitor current available keys number

Code review

based on known hacks and vulnerabilities

A good way to protect your contracts from being hacked is to investigate known hacks and make sure none of them is applicable for your code.



All alerts should be acted

Otherwise, they shouldn't exist at all

Stay up-to date with the alerts

Set-up on-call system for critical alerts if possible

PagerDuty



Opsgenie

Stay up-to date with the alerts

Use separate chats for the info feed and critical alerts



 **Lido Onchain Alerts**

10:58

Lido Mainnet Alerts: [HIGH] Significant Balancer Pool size ...

Critical and High



 **Lido Onchain Updates**

11:08

Lido Mainnet Alerts: [INFO] EasyTrack: New motion create...

All feed

Actions on alerts

RunBooks

RunBook is a comprehensive description of the alert itself and actions to be taken on it.

→ **Description**

→ **Severity**

→ **Confirmation**

→ **Resolution**

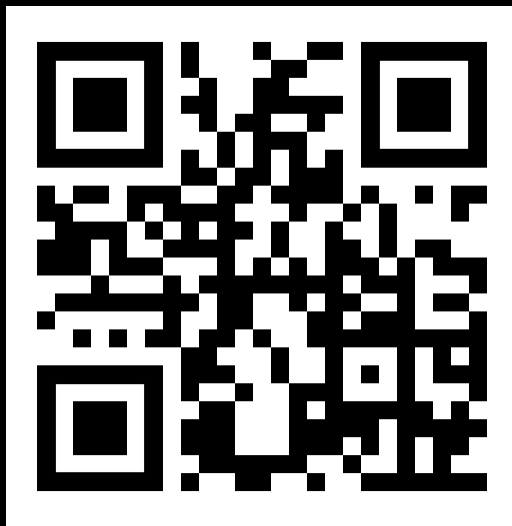
→ **Escalation**

→ **Notes and links**



On-call person or person on-duty should know what actions should be done when the alert fires

Check out Lido **RunBook**



cutt.ly/4BtVNBq

Actions on alerts

Emergency brakes



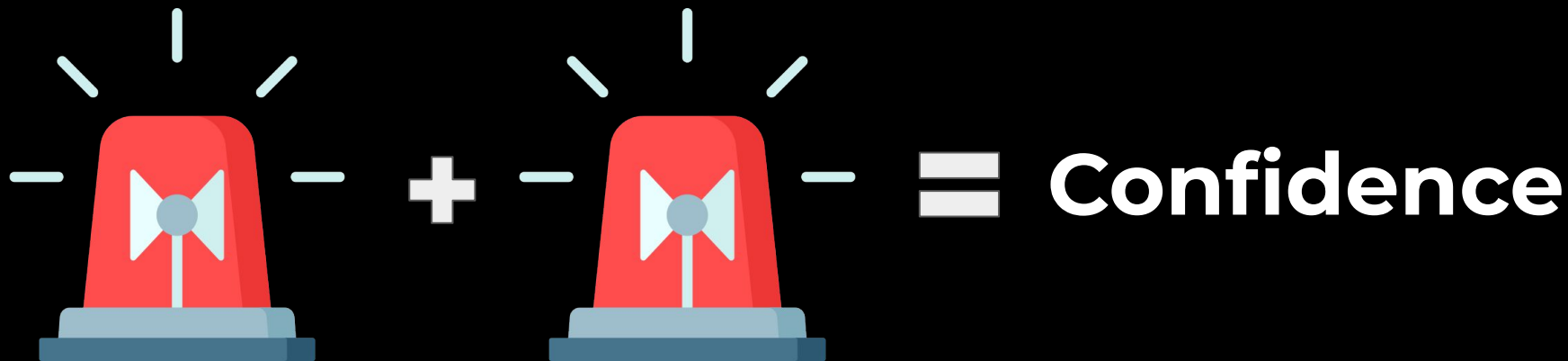
```
@openzeppelin/contracts/security/Pausable.sol
```

```
function deposit() external payable whenNotPaused
```



Your contracts should have “emergency brakes”
Detecting hacks without ability to stop it is useless

One more **thing**



Alerting for the most critical stuff should be duplicated

It is time to create *your own alerts!*

Protocol Alerts Exercises

Exercise 1

Operational monitoring and alerts

- ◆ Define main operational aspects of the protocol
- ◆ Define repetitive events
- ◆ Describe events ABI, alert texts and severity
- ◆ Think about protocol specific operations that you need to be alerted about

Exercise 2

Security monitoring and alerts

- ◆ Define critical events and state changes in the protocol
- ◆ Define ACL model
- ◆ Determine protocol invariants
- ◆ Implement alerts for all points above



Get started at:

<https://github.com/forta-network/forta-bot-workshop>

Alerting checklist



cutt.ly/TBo7IWY

Existing Lido-Forta bots



github.com/lidofinance/alerting-forta



Join Workshop Telegram Group:



<https://t.me/+r-DE0dNqvSFjNmNh>