# 

Forta Network Analytics

Advanced Software Engineering Project

SE 6387

Professor Wei Yang

Team Members

| Mohit Anand | mohit.anand@utdallas.edu |
| --- | --- |
| Rupin Jairaj | rupin.jairaj@utdallas.edu |
| Wenlong Yang | wenlong.yang@utdallas.edu |

# 

# 

[PancakeSwap bots](#_k47u0ji8is2g)

[Pair Created](#_is4zs55rfqfu)

[Auto CAKE Admin Operations](#_tbwuz7psxoe)

[Delegate Votes Change detection bot](#_99ctw8jv3cev)

[Detects large LP deposits/withdrawals to/from a Pancakeswap pool](#_r5elt6ni16cr)

[Large LP Token Deposit/Withdraw](#_r1ddmals9wrz)

[Large Swap Bot](#_6z9tnwt9odkl)

[Lottery-Admin-Operations](#_z2r8xucuf3w6)

[Prediction-Admin-Operations](#_z4chddq5v4yx)

[Set Masterchef Settings bot](#_uwxuh1svr2p1)

[ChainLink](#_qnbjkpuxop7a)

[ChainLink Agent](#_ng948av9oy6i)

[ETH](#_ente3pxbaosn)

[Changes in Block Difficulty](#_7q94izpv6eve)

[GMX bots](#_f2t18kh206qw)

[Detect-many-closing](#_3b60jt4fakj7)

[Detect-many-opening](#_202ofbx71tvo)

[Detect Profitable Trades](#_9ewbg8bpzjcs)

[Large Open and Increase Positions](#_3h4t0heqo0ns)

[Large Size Decrease](#_8fvdmz38fq8w)

[Unusual Profit and Loss](#_vw3azl5wexji)

[Polygon](#_rugexilm3a38)

[Deposit Agent](#_kr1ogva1kvq0)

[AAVE](#_2t1ng8rr58tm)

[Aave Governance Monitor](#_f14x122uyl8b)

[Aave Loan Monitor](#_5r34o1xwj86f)

[Aave Fully Drained](#_wskslnonndu3)

[Aave Treasure Fees Monitor](#_lzmstt8ed2hc)

[Aave Borrow Collateral Ratio Monitor](#_zfl23sv85tli)

[BNBx BOTS](#_fmcx6j70845w)

[Operations agent](#_l07rkk6lrwz8)

[Sucious Amount Minted](#_6jp0miump49q)

[Forta BOTS](#_irmu21ra75e7)

[Credit limit usage](#_87n92wtk6lwj)

[Big Agent](#_pmfy849zw018)

[Forta bot](#_rnhs4ygcp6hg)

[Governance Monitor](#_crduzbetj3go)

[Polygon Stader Childpool](#_hqp7tj8klauo)

[Starter](#_wydpkvwrf7ei)

[Rand Mono Repo](#_82i38me5v7e8)

[Workshop Demo](#_x5pmbn54yjyv)

[Liquidation Liquidity Depth](#_jirjdjbq74f3)

[Large Transfer out](#_72zfo2qcpen0)

[Dai Bridge](#_cf3ob2i02dh4)

[Dai Bridge bot](#_bk3cb2fnq52v)

[Dai Bridge Bot Eff](#_nukax18gr56)

[Balance Checker](#_1ba4u9c50he0)

[Biconomy](#_ph249j3cnwk6)

[Non Executor Send Fund](#_ruqksmbpkslg)

[Standard Monitoring](#_rq595ikvddi7)

[Depositor Failed](#_20pgpw7kr1qv)

[Depositor Success](#_84yrilvfmqze)

[Executor Balance](#_7hs33g6l8z88)

[Failed Internal TXn](#_fkh042qi21qv)

[Time Series Analyzer](#_8pdimcluh7b3)

[Highstreet Standard Monitoring](#_olgiylt1sy53)

[Unizen Monitoring](#_u4gfwod5lemw)

[Standard Mointoring](#_b1vvutgmdu8v)

[Libdo](#_sgunkquijq5e)

[Bridge Arbitrum Bot](#_35jjuuc5i2cu)

[Bridge Balance Bot](#_lnbzh3hxyz61)

[Bridge Etherum Bot](#_wx7ntenu89p7)

[Miscellaneous](#_848mxr1ww7vo)

[Balance Checker](#_rdjtgfuitmz6)

[Drippie](#_c8nct0i83xrb)

[Ribus](#_grs0llsm31yt)

[Ice Phishing Detection](#_wmxqrcjcpirl)

# 

# PancakeSwap bots

### Pair Created

Project Name: Pair-Created

Link: <https://github.com/NethermindEth/Forta-Agents/tree/main/PancakeSwap-Bots/Pair-Created>

Purpose: Detects new pair creation on Pancakeswap DEX

Rules:

1. Filter transaction logs by function ABI: function createPair(address tokenA, address tokenB) external returns (address pair)
2. Extract function arguments: tokenA and tokenB
3. Return (tokenA, tokenB)

Legend:

* tokenA: The address of the first token of the created pair.
* tokenB: The address of the second token of the created pair.

### Auto CAKE Admin Operations

Project Name: Auto-CAKE-Admin-Operations

Link: <https://github.com/NethermindEth/Forta-Agents/tree/main/PancakeSwap-Bots/Auto-CAKE-Admin-Operations>

Purpose: Bot to detect admin operations on CakeVault contract from PancakeSwap

Rules:

1. Filter transaction logs for function calls or events
   1. Function call ABI:
      1. function setAdmin(address \_admin)
      2. function setTreasury(address \_treasury)
      3. function setPerformanceFee(uint256 \_performanceFee)
      4. function setCallFee(uint256 \_callFee)
      5. function setWithdrawFee(uint256 \_withdrawFee)
   2. Events ABI:
      1. event Pause()
      2. event Unpause()
2. Extracts the function arguments
3. Returns (functionName/EventName, [function arguments])

### Delegate Votes Change detection bot

Project Name: Large-Delegators-Vote-Change

Link: [https://github.com/NethermindEth/Forta-Agents/tree/main/PancakeSwap-Bots/Large-Delegators-Vote-Change](https://github.com/NethermindEth/Forta-Agents/tree/main/PancakeSwap-Bots/Auto-CAKE-Admin-Operations)

Purpose: Detects large delegators vote balance change in PancakeSwap Cake token

Rules:

1. Filter transaction logs by events ABI: event DelegateVotesChanged(address indexed delegate, uint previousBalance, uint newBalance)
2. Extract delegateAddress, previousBalance and newBalance
3. Return finding if:
   1. The newBalance is greater than a configured threshold
   2. The previousBalance was less than a configured minimum

### Detects large LP deposits/withdrawals to/from a Pancakeswap pool

Project Name: Large-LP-Deposit-Withdrawal

Link: [https://github.com/NethermindEth/Forta-Agents/tree/main/PancakeSwap-Bots/Large-LP-Deposit-Withdrawal](https://github.com/NethermindEth/Forta-Agents/tree/main/PancakeSwap-Bots/Auto-CAKE-Admin-Operations)

Purpose: Detects large LP deposits/withdrawals to/from a Pancakeswap pool

Rules:

1. Filter transaction logs by events ABI: event Mint(address indexed sender, uint amount0, uint amount1), event Burn(address indexed sender, uint amount0, uint amount1, address indexed to)
2. Use Ether.js to create new contract for the following ABI:
   1. function token0() public view returns (address)
   2. function token1() public view returns (address)
   3. function totalSupply() public view returns (uint256)
   4. function balanceOf(address) public view returns (uint256)
3. Return finding if totalSupply greater than configured threshold

### Large LP Token Deposit/Withdraw

Project Name: Large-LP-Deposit-Withdrawal

Link: <https://github.com/NethermindEth/Forta-Agents/tree/main/PancakeSwap-Bots/Large-LP-Token-Deposit-Withdraw>

Purpose: Detects Large LP Token Deposits or Withdrawals on Pancakeswap

Rules:

1. Filter transaction logs by the following events ABI:
   1. event Deposit(address indexed user, uint256 indexed pid, uint256 amount)
   2. event Withdraw(address indexed user, uint256 indexed pid, uint256 amount)
   3. event EmergencyWithdraw(address indexed user, uint256 indexed pid, uint256 amount)
2. Extract function arguments: pid, amount
3. Uses getEthersProvider() from “forta-agent” library to interact with the blockchain and execute the following contracts:
   1. function balanceOf(address account) external view returns (uint256)
   2. function name() external view returns (string memory)
4. Contracts from step 3 give us the LP balance of a pool associated with the address.
5. Return finding if the amount returned is greater than a configured threshold.

### Large Swap Bot

Project Name: Large-Swap

Link: <https://github.com/NethermindEth/Forta-Agents/tree/main/PancakeSwap-Bots/Large-Swap/>

Purpose: This bot detects large swaps (i.e. the amount of tokens swapped in any Pancake pair contract is a significant fraction of the pair's pool reserves) in the PancakeSwap protocol.

Rules:

1. Filter transaction logs by the following events ABI:
   1. event Swap(address indexed sender, uint amount0In, uint amount1In, uint amount0Out,uint amount1Out,address indexed to)
2. Extract the address parameter value.
3. Use the helper function from “forta-agent” (getEthersProvider) to execute the following function ABI’s on the blockchain to validate if the token pairs are valid:
   1. function token0() external view returns (address)
   2. function token1() external view returns (address)
4. Returns findings if the following conditions hold:
   1. Withdraw of Token0 is greater than 0 and (percentage of Token0 leaving is greater than a configured threshold or percentage of Token1 entering the pool is greater than a configure threshold)
   2. Withdraw of Token1 is greater than 0 and (percentage of Token1 leaving is greater than a configured threshold or percentage of Token0 entering the pool is greater than a configure threshold)

### Lottery-Admin-Operations

Project Name: Lottery Admin Operations

Link: <https://github.com/NethermindEth/Forta-Agents/tree/main/PancakeSwap-Bots/Lottery-Admin-Operations>

Purpose: This bot detects the following PancakeSwap Lottery Admin Operations:

1. setMinAndMaxTicketPriceInCake function call
2. setMaxNumberTicketsPerBuy function call
3. NewRandomGenerator event
4. NewOperatorAndTreasuryAndInjectorAddresses event

Note: We have different transaction handlers for each

Rules:

1. Every transaction handler is called for every event and logs are filtered based on the ABIs in rule 2.
2. Filter transaction logs for the following events and function calls:
   1. event NewRandomGenerator(address indexed randomGenerator)
   2. event NewOperatorAndTreasuryAndInjectorAddresses(address operator,address treasury,address injector)
   3. function setMinAndMaxTicketPriceInCake(uint256 \_minPriceTicketInCake,uint256 \_maxPriceTicketInCake)
   4. function setMaxNumberTicketsPerBuy(uint256 \_maxNumberTicketsPerBuy)
3. Extract the parameters and return the finding as a pair (function/event name from the list in the Purpose, the function parameters)

### Prediction-Admin-Operations

Project Name: PancakeSwap\_-\_Prediction\_Admin\_Operations

Link: <https://github.com/NethermindEth/Forta-Agents/tree/main/PancakeSwap-Bots/Prediction-Admin-Operations>

Purpose: Detects Admin and Operator Operations in PancakePredictionV2

Rules:

1. Filter transaction logs for the following event ABIs:
   1. event Pause(uint256 indexed epoch)
   2. event Unpause(uint256 indexed epoch)
   3. event NewOperatorAddress(address operator)
   4. event NewAdminAddress(address admin)
   5. event NewOracle(address oracle)
   6. event NewTreasuryFee(uint256 indexed epoch, uint256 treasuryFee)
2. Return a custom finding for each event that occurs in the transaction logs.

### 

### Set Masterchef Settings bot

Project Name: PancakeSwap\_-\_Set\_Masterchef\_Settings

Link: <https://github.com/NethermindEth/Forta-Agents/tree/main/PancakeSwap-Bots/Set-Masterchef-Settings>

Purpose: Detects function calls that modify settings in the PancakeSwap Masterchef contract

Rules:

1. Filter transaction logs for the following function ABIs:
   1. function setMigrator(address \_migrator)
   2. function dev(address \_devaddr)
   3. function add(uint256 \_allocPoint, address \_lpToken, bool \_withUpdate)
   4. function set(uint256 \_pid, uint256 \_allocPoint, bool \_withUpdate)
   5. function updateMultiplier(uint256 multiplierNumber)
2. Return a custom finding for each function call that occurs in the transaction logs.

# ChainLink

### ChainLink Agent

Project Name: ChainLink\_Agent

Link: N.A.

Purpose: Detects price updates

Rules:

1. Filters transaction logs for the following event ABIs:
   1. event AnswerUpdated(int256 indexed current, uint256 indexed roundId, uint256 updatedAt)
2. Executes the getAnswer() smart contract on the ChainLink protocol to get the previous round’s answer (price).
3. Return a finding if the price difference between the current round’s answer and the previous round’s answer is greater than a configured threshold.

# ETH

### Changes in Block Difficulty

Project Name: ETH\_Changes\_in\_Block\_Difficulty

Link: <https://github.com/NethermindEth/Forta-Agents/tree/main/ETH-Merge-Bots/Changes-in-Block-Difficulty>

Purpose: Detect unusual changes in block difficulty

Rules:

1. Maintains an array in memory to store the block mining difficulty for the last 100 (or some configured limit) blocks.
2. Returns a finding if the difficulty of the current block crosses some configured threshold.
3. Calculates a moving average of bot difficulties captured so far. If the percentage change of the current bot difficulty exceeds the moving average by a configured threshold we return a finding.

# GMX bots

### Detect-many-closing

Project Name: Detect-many-closing

Link: <https://github.com/NethermindEth/Forta-Agents/tree/main/GMX-Bots/Detect-many-closing>

Purpose: Detects Many Position Closing From An Account Within A Time-Frame.

Description. Returns a finding every time an account closes many positions in a certain amount of time.

Rules:

1. Filter transaction logs by the following events ABI:
   1. event DecreasePosition(bytes32 key,address account,address collateralToken,address indexToken,uint256 collateralDelta,uint256 sizeDelta,bool isLong,uint256 price,uint256 fee)
2. When filtering the logs store the number of times an address occurs in the event DecreasePosition in a map
3. Return finding if an address has more than a configured threshold number of closing positions in the in-memory map

### 

### Detect-many-opening

Project Name: Detect-many-closing

Link: <https://github.com/NethermindEth/Forta-Agents/tree/main/GMX-Bots/Detect-many-opening>

Purpose: Detects Many Position Opening From An Account Within A Time-Frame.

Description. Returns a finding every time an account opens many positions in a certain amount of time.

Rules:

1. Filter transaction logs by the following events ABI:
   1. event IncreasePosition(bytes32 key,address account,address collateralToken,address indexToken,uint256 collateralDelta,uint256 sizeDelta,bool isLong,uint256 price,uint256 fee)
2. When filtering the logs store the number of times an address occurs in the event IncreasePosition in a map
3. Return finding if an address has more than a configured threshold number of opening positions in the in-memory map

### Detect Profitable Trades

Project Name: Detect-profitable-trades

Link: <https://github.com/NethermindEth/Forta-Agents/tree/main/GMX-Bots/Detect-profitable-trades>

Purpose: This bot detects accounts using GMX that have an unusual amount of profitable swaps.

Rules:

1. Filter transaction logs by the following swap event ABI:
   1. event Swap(address account, address tokenIn, address tokenOut, uint256 amountIn, uint256 amountOut, uint256 amountOutAfterFees, uint256 feeBasisPoints)
2. Keep an in-memory dictionary of the format:
   1. Key: accountID
   2. Value: [number of profitable trades, totalTrades, totalProfit]
3. For every transaction event, after updating the in-memory dictionary, check if there are any accounts that have more than the configured number of profitable trades. If so, return a finding.

### Large Open and Increase Positions

Project Name: Large-Open-and-Increase-Positions

Link: <https://github.com/NethermindEth/Forta-Agents/tree/main/GMX-Bots/Large-Open-and-Increase-Positions>

Purpose: Detects the opening of a large position and increase of an existing large position.

Rules:

1. Filter transaction logs for the following events:
   1. event IncreasePosition(bytes32 key, address account, address collateralToken, address indexToken,uint256 collateralDelta,uint256 sizeDelta,bool isLong,uint256 price,uint256 fee)
   2. event UpdatePosition(bytes32 key, uint256 size, uint256 collateral, uint256 averagePrice, uint256 entryFundingRate, uint256 reserveAmount, int256 realisedPnl)
2. Extract the arguments specifically the key, sizeDelta and account.
3. Return a finding if the key for both events match and sizeDelta is greater than a configured threshold.

### Large Size Decrease

Project Name: Large-Size-Decrease

Link: <https://github.com/NethermindEth/Forta-Agents/tree/main/GMX-Bots/Large-size-decrease>

Purpose: Detects the opening of a large position and increase of an existing large position.

Rules:

1. Filter transaction logs for the following events:
   1. event DecreasePosition(bytes32 key, address account, address collateralToken, address indexToken,uint256 collateralDelta,uint256 sizeDelta,bool isLong,uint256 price,uint256 fee)
   2. event ClosePosition(bytes32 key, uint256 size, uint256 collateral,uint256 averagePrice,uint256 entryFundingRate,uint256 reserveAmount,int256 realisedPnl)
2. Return a finding the sizeDelta is larger than the configured threshold (achieved using a few helper functions from ether.js to help with BigNumber calculations with precision)
   1. NOTE: we also use a helper function to check if it was a close position event and add that as a boolean isClosed to the finding.

### Unusual Profit and Loss

Project Name: Large-Size-Decrease

Link: <https://github.com/NethermindEth/Forta-Agents/tree/main/GMX-Bots/Unusual-PnL>

Purpose: Detects unusual PnL when closing a position.

Rules:

1. Filter transaction logs for the following event ABI:
   1. event ClosePosition(bytes32 key, uint256 size, uint256 collateral,uint256 averagePrice,uint256 entryFundingRate,uint256 reserveAmount,int256 realisedPnl)
2. The following args are used for checking if a finding has to be returned:
   1. realisedPnl
   2. size // positionSize
3. Return a finding if realisedPnl is greater than a configured threshold and pnlToSize [(realizedPnl / size)\*100] is greater than a configured amount.

# Polygon

### Deposit Agent

Project Name: Deposit\_Agent

Link: N.A.

Purpose: Detects if blocks contain token transfer events from Main chain to Child chain but hasn’t been validated yet.

Rules:

1. Filter transaction logs for the following event ABI:
   1. event LockedERC20(address indexed depositor, address indexed depositReceiver, address indexed rootToken, uint256 amount)
   2. event StateSynced(uint256 indexed id, address indexed contractAddress, bytes data)
2. Event (a) is called every time an owner of tokens on the Main chain has approved the transfer of a fixed amount of tokens to the child token
3. Event(b) is called internally for the purpose of Proof Of Stake (Pos) work by validators on the Polygon network.
4. Return a finding if there are lock events in a block but no sync events corresponding to validate or the sync events payload and lock events payload don’t match.

# AAVE

### Aave Governance Monitor

Project Name: AE\_Aave\_Governance\_Monitor

Link: N.A.

Purpose: Detects transaction for AAVE governance events.

Rules:

1. Filter transaction logs for the following event ABI names:
   1. ProposalCreated
   2. ProposalExecuted
   3. ProposalQueued
   4. ProposalCanceled
   5. VoteEmitted
   6. ExecutorAuthorized
   7. ExecutorUnauthorized
   8. GovernanceStrategyChanged
   9. OwnershipTransferred
   10. VotingDelayChanged
2. Return a custom finding specifically for each governance event.

### Aave Loan Monitor

Project Name: AE\_Aave\_Loan\_Events\_Monitor

Link: N.A.

Purpose: Bot that monitors for Deposit, Withdraw, Borrow, Repay, and LiquidationCall events.

Rules:

1. Filter transaction logs for the following event ABI names:
   1. Function ABI: Borrow, Deposit, FlashLoan, LiquidationCall, FLASHLOAN\_PREMIUM\_TOTAL, LENDINGPOOL\_REVISION, MAX\_NUMBER\_RESERVES, MAX\_STABLE\_RATE\_BORROW\_SIZE\_PERCENT, finalizeTransfer, getAddressesProvider, getConfiguration, getReserveData, getReserveNormalizedIncome, getReserveNormalizedVariableDebt, getReservesList, getUserAccountData, getUserConfiguration, initReserve, initialize, liquidationCall, view, rebalanceStableBorrowRate, repay, setConfiguration, setPause, setReserveInterestRateStrategyAddress, setUserUseReserveAsCollateral, swapBorrowRateMode, withdraw
   2. Event ABI: Paused, RebalanceStableBorrowRate, Repay, ReserveDataUpdated, ReserveUsedAsCollateralDisabled, ReserveUsedAsCollateralEnabled, Swap, Unpaused, Withdraw
2. Returns a finding if the transaction logs contain any of these events/function ABIs.

### Aave Fully Drained

Project Name: AE\_Aave\_Treasury\_Fees\_Monitor

Link: N.A.

Purpose: Detects if an asset is fully drained from a contract.

Rules:

1. Filter transaction logs for the event with ABI: event Transfer(address indexed from, address indexed to, uint256 value)
2. For each address use the provider interface to interact with the blockchain and get the account’s balance.
3. Return a finding if the ‘from’ address involved in the transaction has a balance of 0 for this specific token after this transaction.

### Aave Treasure Fees Monitor

Project Name: AE\_Aave\_Treasury\_Fees\_Monitor

Link: N.A.

Purpose: Bot that monitors FlashLoan events and alerts when the amount of fees collected is greater than either the configured or statistical threshold.

Rules:

1. Filter transaction logs based on the events with the FlashLoan ABI signatures.
2. Scale the premium value and query the price oracle (using the price oracle contract) for the current price of the token in ETH
3. Return a finding if the scaled premium is greater than then mean + 3 standard deviations or less than the mean + 3 standard deviations but greater than a configured threshold.

### Aave Borrow Collateral Ratio Monitor

Project Name: AE\_Aave\_Borrow\_Collateral\_Ratio\_Monitor

Link: N.A.

Purpose: Bot that monitors the ratio between borrowed and collateral amounts for each aToken

Rules:

1. Execute getReserverData(token\_Address) and get the following parameters:
   1. availableLiquidity, totalStableDebt, totalVariableDebt
2. totalDebt = totalStableDebt + totalVariableDebt
3. totalLiquidity = availableLiquidity + totalDebt
4. Return a finding if totalLiquidity is greater than 0 and totalDebt/totalLiquidity is greater than the configured threshold.

# BNBx BOTS

### Operations agent

Project Name : bnbx-bot-operations-agent

Link - NA

Purpose - Alerts on/off Chain Bot Delays and Failures

Rules :

* Checks status for 5 transaction , start delegation transaction, completed delegation transaction ,
* start undelegation transaction , undelegation update transaction , and complete undelegation transaction
* For start delegation transaction if the difference in hours (diffhours : current time - last start delegation time) is less than 36 throw the status as Start Delegation Failed else return the findings which has the following values name , description , alertid , protocol , type , severity and metada
* For completed delegation transaction if the difference in hours greater than 12 throw the status as Failed else return the findings
* For start undelegation transaction if the difference in hours greater than 169 throw the status as Failed else return the findings
* For undelegation update transaction if the difference in hours greater than 12 throw the status as Failed else return the findings
* For completed undelegation transaction if the difference in hours greater than 193 throw the status as Failed else return the findings

### Sucious Amount Minted

Project Name : bnbx-suspicious-amount-agent

Link - NA

Purpose - Alerts on suspicious amount of BNBx minted, unstaked, Rewards

Rules :

1. Fetches the value from the transaction
   1. If values is greater than threshold then 500 then raise a flag and message as Large amount of BnBx minted
2. Fetches the amount details from the transaction
   1. If the amount is less than minimum threshold which is 1 - Low amount of BNBx Reward Received
   2. If the amount is greater than minimum threshold which is 20 - High amount of BNBx Reward Received
3. Fetch the amount in bnbx from the transaction
   1. If the amount in bnbx is greater than minimum bnbx unstaked threshold which is 500 - Large amount of BNBx unstaked

# Forta BOTS

### Credit limit usage

Project name - iron-bank-credit-limit-usage-agent

Link - <https://github.com/LimeChain/forta-agent-scripts/>

Purpose - Detects if credit limit usage is more than target value

Rules :

Filter transaction by event Borrow(address borrower, uint256 borrowAmount, uint256 accountBorrows, uint256 totalBorrows)

Computes credit usage which is accountBorrowsNum.mul(100).div(creditLimitsNum).toFixed(2)

If credit usage is more than threshold which is 90% - Credit limit usage is more than threshold

### Big Agent

Project name - forta-bot-wizard-bigagent

Link - NA

Purpose - Forta Agent template for monitoring emitted events from a bunch of agents

1. Account Balance
2. Address Watch
3. Admin Event
4. Contract Variable Monitor
5. Multi Signing
6. Governance
7. Monitor Function Calls
8. New Contract interaction
9. Tornado Cash monitor
10. Transaction failure count

Rules :

* Compares the log of the transaction values such as accountName,accountAddress,accountBalance,thresholdEth,numAlerts,protocolName,developerAbbreviation,protocolAbbreviation,alertType,alertSeverity
* Applies basic rules such as comparing threshold to current amount and raises a flag
* If balance < threshold add an alert to the findings
* if less than the specified number of hours has elapsed, just increment the counter for

the number of alerts that would have been generated

* Address watch
  + check type, this will fail if 'type' is not valid
  + check severity, this will fail if 'severity' is not valid
  + get list of addresses to watch
  + check if an address in the watchlist was the initiator of the transaction
* Admin Events
  + get the Array of events for a given contract(contractEventConfig, currentContract, adminEvents, contracts)
  + find the abi for the contract the proxy is pointing to and get the event signatures
  + load the ABI from the specified file
  + iterate over each item in parsedLogs and evaluate expressions (if any) given in the

### Forta bot

Project Name - euler-forta

Link - NA

Purpose - Euler-Forta Bot Suite

Rules : Forta Agent template for monitoring emitted events from a bunch of agents

1. Balance Monitor
2. Exchange Rate Monitor
3. Ice-phishing monitor
4. New Contract information
5. Retrancy Monitor
6. Tornado Cash Monitor

* Compares the log of the transaction values such as accountName,accountAddress,accountBalance,thresholdEth,numAlerts,protocolName,developerAbbreviation,protocolAbbreviation,alertType,alertSeverity
* Applies basic rules such as comparing threshold to current amount and raises a flag
* If balance < threshold add an alert to the findings
* if less than the specified number of hours has elapsed, just increment the counter for

the number of alerts that would have been generated

### Governance Monitor

Project Name - forta-bot-wizard-bigagent

Link - NA

Purpose - Forta Agent template for monitoring emitted events from a bunch of agents

1. Account Balance
2. Address Watch
3. Admin Event
4. Contract Variable Monitor
5. Multi Signing
6. Governance
7. Monitor Function Calls
8. New Contract interaction
9. Tornado Cash monitor
10. Transaction failure count

Rules :

* Compares the log of the transaction values such as accountName,accountAddress,accountBalance,thresholdEth,numAlerts,protocolName,developerAbbreviation,protocolAbbreviation,alertType,alertSeverity
* Applies basic rules such as comparing threshold to current amount and raises a flag
* If balance < threshold add an alert to the findings
* if less than the specified number of hours has elapsed, just increment the counter for

the number of alerts that would have been generated

* Address watch
  + check type, this will fail if 'type' is not valid
  + check severity, this will fail if 'severity' is not valid
  + get list of addresses to watch
  + check if an address in the watchlist was the initiator of the transaction
* Admin Events
  + get the Array of events for a given contract(contractEventConfig, currentContract, adminEvents, contracts)
  + find the abi for the contract the proxy is pointing to and get the event signatures
  + load the ABI from the specified file
  + iterate over each item in parsedLogs and evaluate expressions (if any) given in the log

### Polygon Stader Childpool

Project Name : forta-agent-polygon-stader-childpool

Link - NA

Purpose - Forta Agent for maticX childpool

Rules :

1. Fetches the findings from the transaction log : unit, ethersProvider, ChildPool, instantPoolMatic
2. If instantPoolMatic is less than MIN\_INSTANT\_POOL\_BALANCE = 100000 - Low Instant Pool Matic
3. Compares the current address to that of findings address and if not same - Instant Pool Owner Change
4. If Amount Fetched from findings greater than MATICX\_SWAP\_THRESHOLD = 50000 - Large amount of Matic swapped

### Starter

Project Name : Forta Agent Starter

Link - NA

Purpose - Forta Agent Typescript starter project

Rules :

1. limiting this agent to emit only 5 findings so that the alert feed is not spammed : findingsCount >= 5
2. extract transfer event arguments
3. if more than 10,000 Tether were transferred, report it

### Rand Mono Repo

Project Name : forta-rand-monorepo

Link - https://github.com/Rand-Network/forta-agents

Purpose : Forta Bots to monitor Rand ecosystem

Rules :

* Compares the log of the transaction values such as accountName,accountAddress,accountBalance,thresholdEth,numAlerts,protocolName,developerAbbreviation,protocolAbbreviation,alertType,alertSeverity
* Applies basic rules such as comparing threshold to current amount and raises a flag
* If balance < threshold add an alert to the findings
* if less than the specified number of hours has elapsed, just increment the counter for

the number of alerts that would have been generated

* Address watch
  + check type, this will fail if 'type' is not valid
  + check severity, this will fail if 'severity' is not valid
  + get list of addresses to watch
  + check if an address in the watchlist was the initiator of the transaction
* Admin Events
  + get the Array of events for a given contract(contractEventConfig, currentContract, adminEvents, contracts)
  + find the abi for the contract the proxy is pointing to and get the event signatures
  + load the ABI from the specified file
  + iterate over each item in parsedLogs and evaluate expressions (if any) given in the log

### Workshop Demo

Project Name : forta-workshop-demo

Link - NA

Purpose - Forta Agent Python starter project

Rules :

1. filter the transaction logs for any Tether transfers
2. extract transfer event arguments
3. if more than 10,000 Tether were transferred, report it

### Liquidation Liquidity Depth

Project Name - ibagreement-v2-liquidation-liquidity-depth

Link - https://github.com/LimeChain/forta-agent-scripts/

Purpose - Detects if the collatoralUSD doesn't be lower when liquidation has been finished

Rules -

Extract the transaction log and assign values

* const agreementContract = new ethers.Contract(agreementAddr, agreementABI, provider)
  + let closeFactor = await agreementContract.closeFactor(
  + closeFactor = ethers.utils.formatUnits(closeFactor, 18
  + closeFactor = new BigNumber(closeFactor)
  + const clbl = new BigNumber(collateralBalance.toString())
  + const liquidationAmount = clbl.multipliedBy(closeFactor)
  + Check if findings has the predefined markers
* Returns the finding and return and the transaction filters

### Large Transfer out

Project Name - large-transfer-out

Link - NA

Purpose - Bot identifies large native asset transfers that didn't exist X days ago

Rules -

* initializes the state variables that are tracked across tx and blocks it is called from test to reset state between tests
* Check if the older value is less than 50 ETH , if true return the append result to findings High amount of native token transferred
* Return Findings

# Dai Bridge

### Dai Bridge bot

Project Name - dai-bridge-bot

Link - NA

Purpose -Forta bot that detects the condition if the total supply of DAI on an L2 exceeds the amount of DAI locked in the mainnet escrow

Rules :

1. Fetches the values from function call NetworkManagerCurr
2. Extract function argument currBlockTimeStamp, findings, DAI\_L1, i, currData, L1\_escrowBal, \_a, l2\_metadata;
3. Export the handleblock

### Dai Bridge Bot Eff

Project Name - dai-bridge-bot-eff

Link - https://github.com/MakC-Ukr/dai\_bridge-forta-agent

Purpose - Forta bot that detects the condition if the total supply of DAI on an L2 exceeds the amount of DAI locked in the mainnet escrow"

Rules:

1. Fetches the values from function call NetworkManagerCurr
2. Extract function argument currBlockTimeStamp, findings, DAI\_L1, i, currData, L1\_escrowBal, \_a, l2\_metadata;
3. Export the handleblock

### Balance Checker

Project Name - dai-bridged-balance-checker

Link - https://github.com/MakC-Ukr/dai\_bridge-forta-agent/tree/master/L1-bot

Purpose - Checks if the L2 total supply of DAI is less than the amount of DAI deposited on L1 escrow for MakerDAO

Rules :

1. Fitler the transaction by : function provideHandleBlock(DaiL1Address, erc20Abi, l1EscrowAddressOp, l1EscrowAddressArb, apiUrl, queryOp, queryArb, headers, daiL2Address)
2. Extract information and compares it to Chain id
3. Returns the findings

# Biconomy

### Non Executor Send Fund

Project Name - hal-bico-nonexecutor-sendfund

Link - NA

Purpose - Standard Monitoring nonexecutor-sendfund

Rules :

* filterFunction accepts either a string or an Array of strings and here it will only pass in one string at a time to keep the synchronization with the expressions that we need to evaluate
* if there is an expression to check, verify the condition before creating an alert
* Create findings functionName,name,address,functionType,functionSeverity,parsedFunction.args,config.protocolName,config.protocolAbbreviation , config.developerAbbreviation,expression
* Export the findings

### Standard Monitoring

Project Name - hal-bico-standard-monitoring

Link - NA

Purpose - Standard Monitoring

Rules:

* if(txEvent.transaction.from.toLowerCase() === "0xeefd474e80b6caea43f212d964409c473684e3fe" && txEvent.transaction.from.toLowerCase() === "0x1439eda7f9a911b9120e9a0dafb60eae317f7685" && txEvent.transaction.from.toLowerCase() === "0x4fb5df81b644e3bd5ad0ba07dce2b67559c764e0" && txEvent.transaction.from.toLowerCase() === "0x600be30999eb256f2bef451b69950f7dc84ac6b1"){
* return findings;
* } biconomy-executor-addresses
* filterFunction accepts either a string or an Array of strings and here it will only pass in one string at a time to keep the synchronization with the expressions that we need to evaluate
* loop over the Array of results the transaction may contain more than one function call to the same function
* if there is an expression to check, verify the condition before creating an alert
* Create findings functionName,name,address,functionType,functionSeverity,parsedFunction.args,config.protocolName,config.protocolAbbreviation , config.developerAbbreviation,expression
* Export Findings

### Depositor Failed

Project name - hal-biconomy-depositerc20-failed

Link - NA

Purpose - Standard Monitor

Rules :

* if(txEvent.transaction.from.toLowerCase() === "0xeefd474e80b6caea43f212d964409c473684e3fe" && txEvent.transaction.from.toLowerCase() === "0x1439eda7f9a911b9120e9a0dafb60eae317f7685" && txEvent.transaction.from.toLowerCase() === "0x4fb5df81b644e3bd5ad0ba07dce2b67559c764e0" && txEvent.transaction.from.toLowerCase() === "0x600be30999eb256f2bef451b69950f7dc84ac6b1"){
* return findings;
* } biconomy-executor-addresses
* filterFunction accepts either a string or an Array of strings and here it will only pass in one string at a time to keep the synchronization with the expressions that we need to evaluate
* loop over the Array of results the transaction may contain more than one function call to the same function
* if there is an expression to check, verify the condition before creating an alert
* Create findings functionName,name,address,functionType,functionSeverity,parsedFunction.args,config.protocolName,config.protocolAbbreviation , config.developerAbbreviation,expression
* Export Findings

### Depositor Success

Project name - hal-biconomy-depositerc20-success

Link - NA

Purpose - Standard Monitor

Rules :

* if(txEvent.transaction.from.toLowerCase() === "0xeefd474e80b6caea43f212d964409c473684e3fe" && txEvent.transaction.from.toLowerCase() === "0x1439eda7f9a911b9120e9a0dafb60eae317f7685" && txEvent.transaction.from.toLowerCase() === "0x4fb5df81b644e3bd5ad0ba07dce2b67559c764e0" && txEvent.transaction.from.toLowerCase() === "0x600be30999eb256f2bef451b69950f7dc84ac6b1"){
* return findings;
* } biconomy-executor-addresses
* filterFunction accepts either a string or an Array of strings and here it will only pass in one string at a time to keep the synchronization with the expressions that we need to evaluate
* loop over the Array of results the transaction may contain more than one function call to the same function
* if there is an expression to check, verify the condition before creating an alert
* Create findings functionName,name,address,functionType,functionSeverity,parsedFunction.args,config.protocolName,config.protocolAbbreviation , config.developerAbbreviation,expression
* Export Findings

### Executor Balance

Project Name - hal-biconomy-executor-balance

Link - NA

Purpose - Executor Balance Monitoring

Rules -

* console.log(`${name} balance on ${Network[network]} is ${(result/decimal).toFixed(3)} ${symbol}`)

### Failed Internal TXn

Project Name - hal-detect-failed-internal-txn

Link - NA

Purpose - This bot detect the successful transaction with internal failure

Rules :

* Filter through the transaction and extract findings
* if(!txEvent.status) return findings

### Time Series Analyzer

Project Name - time-series-analyzer-template

Link - NA

Purpose - Forta detection bot template that allows to turn a noisy alert time series into a more precice alert.

Rules -

* initializes the state variables that are tracked across tx and blocks it is called from test to reset state between tests
* maintains a time stamps; holds up to TIMESTAMP\_QUEUE\_SIZE in memory return: None
* get time for block to derive date range for query
* get all alerts for date range ( BOT\_ID, ALERT\_NAME, CONTRACT\_ADDRESS, start\_date, end\_date)
* build time series model without last bucket
* for any values we do have that are 0, replace with median
* don't alert if current value is 0 because there are reliability issues leading to bot not running and resulting in 0 alerts. Once the reliability increases, this condition can be removed.
* Export Findings

### Highstreet Standard Monitoring

Project Name - hal-highstreet-standard-monitoring

Link - NA

Purpose - Standard Mointoring

Rules -

* if(txEvent.transaction.from.toLowerCase() === "0xeefd474e80b6caea43f212d964409c473684e3fe" && txEvent.transaction.from.toLowerCase() === "0x1439eda7f9a911b9120e9a0dafb60eae317f7685" && txEvent.transaction.from.toLowerCase() === "0x4fb5df81b644e3bd5ad0ba07dce2b67559c764e0" && txEvent.transaction.from.toLowerCase() === "0x600be30999eb256f2bef451b69950f7dc84ac6b1"){
* return findings;
* } biconomy-executor-addresses
* filterFunction accepts either a string or an Array of strings and here it will only pass in one string at a time to keep the synchronization with the expressions that we need to evaluate
* loop over the Array of results the transaction may contain more than one function call to the same function
* if there is an expression to check, verify the condition before creating an alert
* Create findings functionName,name,address,functionType,functionSeverity,parsedFunction.args,config.protocolName,config.protocolAbbreviation , config.developerAbbreviation,expression
* Export Findings

### Unizen Monitoring

Project Name - hal-unizen-standard-monitoring

Link - NA

Purpose - Standard Mointoring

Rules -

* if(txEvent.transaction.from.toLowerCase() === "0xeefd474e80b6caea43f212d964409c473684e3fe" && txEvent.transaction.from.toLowerCase() === "0x1439eda7f9a911b9120e9a0dafb60eae317f7685" && txEvent.transaction.from.toLowerCase() === "0x4fb5df81b644e3bd5ad0ba07dce2b67559c764e0" && txEvent.transaction.from.toLowerCase() === "0x600be30999eb256f2bef451b69950f7dc84ac6b1"){
* return findings;
* } biconomy-executor-addresses
* filterFunction accepts either a string or an Array of strings and here it will only pass in one string at a time to keep the synchronization with the expressions that we need to evaluate
* loop over the Array of results the transaction may contain more than one function call to the same function
* if there is an expression to check, verify the condition before creating an alert
* Create findings functionName,name,address,functionType,functionSeverity,parsedFunction.args,config.protocolName,config.protocolAbbreviation , config.developerAbbreviation,expression
* Export Findings

### Standard Mointoring

Project Name - hal-woonk-standard-monitoring

Link - NA

Purpose - Standard Monitoring

Rules -

* filterFunction accepts either a string or an Array of strings and here it will only pass in one string at a time to keep the synchronization with the expressions that we need to evaluate
* loop over the Array of results the transaction may contain more than one function call to the same function
* if there is an expression to check, verify the condition before creating an alert
* Create findings functionName,name,address,functionType,functionSeverity,parsedFunction.args,config.protocolName,config.protocolAbbreviation , config.developerAbbreviation,expression
* Export Findings

# Libdo

### Bridge Arbitrum Bot

Project Name - lido-l2-bridge-arbitrum-bot

Link - NA

Purpose - Lido Detection Bot for Arbitrum part of L2 bridge

Rules -

* Compare block or tx handling should take no more than 60 sec. If not all processing is done it will be done later in background
* report findings from init.
* Run agent handler
* if all handlers have finished execution drop timeout and resolve promise
* Export Findings

### Bridge Balance Bot

Project Name - lido-l2-bridge-balance-bot

Link - NA

Purpose - Lido Detection Bot for L2 bridges balance

Rules -

* Compare block or tx handling should take no more than 60 sec. If not all processing is done it will be done later in background
* report findings from init.
* Run agent handler
* if all handlers have finished execution drop timeout and resolve promise
* Export Findings

### Bridge Etherum Bot

Project Name - lido-l2-bridge-ethereum-bot

Link - NA

Purpose - Lido Detection Bot for Ethereum part of L2 bridges

Rules -

* Compare block or tx handling should take no more than 60 sec. If not all processing is done it will be done later in background
* report findings from init.
* Run agent handler
* if all handlers have finished execution drop timeout and resolve promise
* Export Findings

# Miscellaneous

### Balance Checker

Project Name: BalanceChecker

Link: N.A.

Purpose: Detects if an account’s balance falls below 0 or a configured upper limit.

Rules:

1. Uses a Provider instance from ether.js to interact with the blockchain and fetch the balance for the account using - provider.getBalance(exports.ACCOUNT, blockEvent.blockNumber).
2. Return a finding if the account balance falls below 0 or is greater than a configured upper limit.

### Drippie

Project Name: Drippie\_Detection

Link: N.A.

Purpose: A template agent that holds multiple sub bots that monitor Ether for a variety of events.

### Ribus

Project Name: RibusBot

Link: N.A.

Purpose: A template agent that holds multiple sub bots that monitor Ether for a variety of events.

### 

### 

### 

### Ice Phishing Detection

Project Name: Ice\_Phishing\_Detection\_2.0

Link: N.A.

Purpose: This bot detects if an account gains a large number of approvals and if it transfers the approved funds.

Rules:

1. Filters transaction logs for these events:
   1. event Approval(address indexed owner, address indexed spender, uint256 value)
   2. event Approval(address indexed owner, address indexed spender, uint256 indexed tokenId)
   3. event ApprovalForAll(address indexed owner, address indexed spender, bool approved)
   4. event Transfer(address indexed from, address indexed to, uint256 value)
   5. event Transfer(address indexed from, address indexed to, uint256 indexed tokenId)
   6. event TransferSingle(address indexed operator, address indexed from, address indexed to, uint256 tokenId, uint256 value)
   7. event TransferBatch(address indexed operator, address indexed from, address indexed to, uint256[] tokenIds, uint256[] values)
2. Keeps track of addresses involved in Approval events in an LRU cache.
3. Filters out approval revokes.
4. Constantly clears out the LRU cache of old records beyond a configured threshold.
5. If we notice a high number of approvals for a specific spender we record the spender’s address.
6. Return a finding with the snapshot of the LRU cache after a configured amount of time.