BOT RULES

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

Depositer 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

Depositer 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