FORTA INFINITY
WEAVIATE TESTING RESULTS
PROMPT VS KEYWORDS

PROMPT:

1. **DIRECT QUESTIONS:**

    Actual Result is #1 (prompts): 1, 2, 4, 5, 6, 7, 8, 10, 11, 12, 13
    Average Certainty for actual result is #1: 0.9402995000000001
    Actual Result is top #5 (prompt- rank): prompt 3- 4th , 9- 3rd , and 13- 2nd.

Prompt 3#: Give a bot that detects and alerts on different types of evasion tactics used by smart contracts on the Ethereum blockchain

3. Give a bot that detects and alerts on different types of evasion tactics used by smart contracts on the Ethereum blockchain

Expected: forta-bot-evasion
Actual Result (Certainty):
A. BlockSec Phishing Alert - 0.9424762725830078
B. Attack Detector Feed (beta alt) - 0.9421588182449341
C. Passphrase detection bot - 0.9412917494773865
D. forta-bot-evasion - 0.9383700489997864

Result: Expected result is 4th on list. ✅

A. Description: The Web3 bot, as depicted in the provided code, serves the primary function of monitoring and alerting against various forms of phishing and scams occurring on the blockchain, with a particular focus on Ethereum. It's equipped to identify different types of fraudulent activities, including ice-phishing, fraudulent NFT orders, address poisoning, native ice-phishing, and phishing website detection. The bot's source code comprises JavaScript modules detailing its interaction with the blockchain, handling of smart contract transactions, categorization of alerts, and encryption of sensitive data. It utilizes APIs and tools like Moralis and AlchemySDK to gather transaction details and employs predefined patterns and signatures to analyze transaction data and generate alerts. Noteworthy components include modules for defining constants and ABIs, processing phishing transactions, managing alerts, handling specific marketplace transactions, recognizing phishing attacks through contract interactions, and communicating with external APIs like Etherscan. The bot's operation involves real-time monitoring, encrypted alert generation, and periodic updates of token data. Overall, it's a

comprehensive security tool designed to safeguard users' digital assets by proactively detecting and mitigating blockchain-based phishing and scamming activities.

B. The provided source code represents a web3 bot responsible for handling alerts and blocks in a blockchain environment. The bot's functionality includes detecting attacks, emitting findings related to suspicious activities such as fraudulent transactions, and persisting bot state. It leverages external data sources such as manual alert lists and false positive lists to enhance its detection capabilities. Additionally, the bot emits findings for potential false positives and mitigates them accordingly. The code demonstrates the bot's ability to interact with the blockchain by querying blocks and contracts to extract relevant information for analysis. Moreover, it showcases mechanisms for handling exceptions and ensuring the bot's state persistence for uninterrupted operation. Overall, the bot's purpose revolves around monitoring and responding to alerts and blocks in a blockchain network to detect and mitigate potential threats and suspicious activities.
C. The bot is designed to monitor Ethereum smart contracts, alerting on specific transaction patterns to detect critical and informational events, potentially indicating vulnerabilities or noteworthy activities. It surveils changes like signer or ownership alterations, whitelist modifications, reward claims, and gaming-related actions. Configuration changes are facilitated through an underlying management system, while transaction logs and event emissions on the Ethereum blockchain are analyzed based on the settings in 'bot-config.json'. The bot's implementation involves utility functions, validation, and configuration logic to ensure accuracy and facilitate evaluation, parsing, and state management. The source code comprises various components like utility tests, configuration validation, Ethereum interaction, and core bot logic, all structured to allow extensibility and adaptation for diverse smart contract monitoring scenarios
D. This web3 bot specializes in detecting and alerting on various evasion tactics employed by smart contracts on the Ethereum blockchain, with a focus on metamorphic contracts and red pill contracts. It employs transaction traces and static analysis of bytecode to identify these tactics. Metamorphic contracts dynamically alter their bytecode while maintaining the same address, facilitated by the CREATE2 opcode, and the bot scrutinizes factory contracts deploying these \"mutant\" contracts, issuing alerts categorized as \"Info\" and \"Suspicious.\" Red pill contracts are designed to discern simulation environments and activate malicious functions solely on the mainnet, with the bot examining bytecode patterns indicative of such behavior. Transaction assessments are probabilistic, with scores assigned to indicators and an overall probability calculated. Alerts are triggered only if probabilities surpass a predefined confidence level set in `options.py`. Key components of the source code include `options.py`, `findings.py`, `utils.py`, and `agent.py`, with the latter containing the primary logic for transaction analysis. While specifics of the `ioseeth` library are not provided, it is referenced in `agent.py` for defining metrics and indicators. Overall, the bot operates within Ethereum to identify suspicious smart contract behaviors, offering alerts for further scrutiny or auditing.

PROMPT #9: I want a security bot that consists of multiple parts that involve detecting attacks on Ethereum-like smart contract protocols in real-time and handling false positives within the Web3 ecosystem.

9. I want a security bot that consists of multiple parts that involve detecting attacks on Ethereum-like smart contract protocols in real-time and handling false positives within the Web3 ecosystem.

   Expected: Attack Detector Feed (beta)
   Actual Result (Certainty):
   A. BlockSec Phishing Alert - 0.9374871253967285
   B. Attack Detector Feed (beta alt) - 0.9336826503276825
   C. Attack Detector Feed (beta) - 0.9319761395454407

Result: Expected bot was #3 but certainties are very close. ✅

A. The Web3 bot, as depicted in the provided code, serves the primary function of monitoring and alerting against various forms of phishing and scams occurring on the blockchain, with a particular focus on Ethereum. It's equipped to identify different types of fraudulent activities, including ice-phishing, fraudulent NFT orders, address poisoning, native ice-phishing, and phishing website detection. The bot's source code comprises JavaScript modules detailing its interaction with the blockchain, handling of smart contract transactions, categorization of alerts, and encryption of sensitive data. It utilizes APIs and tools like Moralis and AlchemySDK to gather transaction details and employs predefined patterns and signatures to analyze transaction data and generate alerts. Noteworthy components include modules for defining constants and ABIs, processing phishing transactions, managing alerts, handling specific marketplace transactions, recognizing phishing attacks through contract interactions, and communicating with external APIs like Etherscan. The bot's operation involves real-time monitoring, encrypted alert generation, and periodic updates of token data. Overall, it's a comprehensive security tool designed to safeguard users' digital assets by proactively detecting and mitigating blockchain-based phishing and scamming activities.
B. The provided source code represents a web3 bot responsible for handling alerts and blocks in a blockchain environment. The bot's functionality includes detecting attacks, emitting findings related to suspicious activities such as fraudulent transactions, and persisting bot state. It leverages external data sources such as manual alert lists and false positive lists to enhance its detection capabilities. Additionally, the bot emits findings for potential false positives and mitigates them accordingly. The code demonstrates the bot's ability to interact with the blockchain by querying blocks and contracts to extract relevant information for analysis. Moreover, it showcases mechanisms for handling exceptions and ensuring the bot's state persistence for uninterrupted operation. Overall, the bot's purpose revolves around monitoring and responding to alerts and blocks in a blockchain network to detect and mitigate potential threats and suspicious activities.
C. The provided code comprises several components designed to detect attacks on Ethereum-like smart contract protocols in real-time and manage false positives within the

Web3 ecosystem. The Attack Detector Feed combines alerts from various base bots, employing anomaly detection and heuristic approaches to identify attacks across four stages: Funding, Preparation, Exploitation, and Money Laundering. DynamoUtils facilitates interaction with DynamoDB for storing and managing detected attacks and entities. Findings module creates structured alerts for better understanding detected activities. BlockChainIndexerService interacts with blockchains to retrieve contract information. L2Cache provides a caching layer for persistent data storage. Utils Test contains test cases for utility functions, while DynamoUtilsTest ensures proper DynamoDB interaction. Constants module holds various thresholds and keys. BlockchainIndexerServiceMock aids in testing without actual external calls. AgentTest validates the agent's functionality, while Agent integrates all components to detect and manage security incidents. Web3Mock and BlockchainIndexerServiceTest are for testing purposes. Tabular predicts false positives, and FPMitigator utilizes models for mitigation. DownloadData fetches and formats transaction-related data, and Autoencoder defines a model for encoding graph data, useful for anomaly detection. Together, these components form a bot within the Forta detection network for real-time security incident detection on the blockchain.

PROMPT #13: Recommend me a bot that detects MEV bots that uses different strategies to extract value from a protocol by adding, removing or reordering transactions in a block

13. Recommend me a bot that detects MEV bots that uses different strategies to extract value from a protocol by adding, removing or reordering transactions in a block

    Expected:Forta MEV bot detector
    Actual Result (Certainty):
    A. High Value Transaction - 0.9166077077388763
    B. Forta MEV bot detector - 0.9144038558006287
    C. Attack Detector Feed (beta alt) - 0.9142635464668274

Result: Expected result is #2 ✅

A. The Web3 Bot serves to detect transactions on various blockchain networks exceeding a configured high-value threshold, signaling potential suspicious activity. It operates across multiple chains like Ethereum, Optimism, BNB Smart Chain, and others. The bot alerts with an \"Info\" severity level, computes anomaly scores, and generates labels for suspicious transactions and addresses. Its key functions include initialization, transaction and block handling, configurable thresholds, and data persistence management. The bot's purpose is to monitor and alert anomalies in transaction values, contributing to security and transparency in the Web3 ecosystem.
B. The web3 bot described in the provided source code appears to be focused on inspecting and analyzing transactions on the Ethereum blockchain. It utilizes classifiers to identify and categorize different types of smart contracts and transactions, particularly focusing on decentralized finance (DeFi) protocols such as 0x and Balancer. The bot's purpose seems to be understanding and interpreting the activities happening within these protocols, including swaps, transfers, and other interactions, in order to provide

insights or perform specific actions based on the observed behavior. The source code includes specifications for different protocols and functionalities, along with classifiers to parse and interpret various types of transactions and events.

2. PARAPHRASED QUESTIONS

Actual Result is #1: **1, 4, 5, 7, 10, 13**
Actual Result is not #1: prompt 2, 3, 6, 8, 9, 11, 12, 14, 15
Actual Result is top #5 (prompt-rank): prompt 3-2nd, prompt 15 - 4th
Actual Result is not top #5 (prompt-rank): prompt 2- N/A, prompt 8-8th, prompt 9-N/A, prompt 11 - 7th , prompt 12- N/A, prompt 6- N/A

Prompt #2: What type of Web3 bot would be recommended for monitoring and alerting on decreases in the virtual prices of AA (senior) and BB (junior) tranches within Idle Finance's Perpetual Yield Tranches in the DeFi ecosystem on the Ethereum blockchain

2. Recommend a bot that identifies fraudulent tokens engaged in rake scams, which impose an extra swap fee on the Uniswap decentralized exchange and divert it to an externally owned account.

Expected:rake-scam-token-detection-bot
Actual Result (Certainty):
A. large-profit-bot - 0.9435552060604095
B. Large Balance Decrease - Polygon Ether Bridge - 0.9226204454898834
C. alperp-unusual-swap-profit-tx - 0.9217552244663239

Result: Expected result not in top 10 ❌

A. The Large Profit Bot, designed for Web3, is engineered to track blockchain transactions across multiple chains, identifying those yielding substantial profits for the initiator. It defines a significant profit as exceeding $10,000 or 5% of a token's total supply. Leveraging Moralis and block explorer APIs for data acquisition, the bot advises users to integrate their API keys for optimal performance. Supporting Ethereum, Optimism, BNB Smart Chain, Polygon, Fantom, Arbitrum, and Avalanche, it issues \"LARGE-PROFIT\" alerts with medium severity upon detecting lucrative transactions. These alerts furnish metadata such as the transaction initiator, the called contract, anomaly scores, and profit amounts. Employing labels, it supplements details regarding the entity receiving the profit, confidence levels, and the transaction's large profit status. The bot's functionality is vetted via predefined transactions on Ethereum and Polygon networks. Its JavaScript source code encompasses files for testing, configuration, data fetching, utilities, and the core logic for monitoring transactions, detecting large profits, and generating alerts.

B. The Large Balance Decrease Bot is tailored to oversee and pinpoint noteworthy reductionss in token balances within designated protocols across diverse blockchains like Ethereum, Optimism, Binance Smart Chain, Polygon, Fantom, Arbitrum, and Avalanche. It issues alerts upon detecting either a complete drainage of a protocol's token balance or a substantial decrease therein. One alert, labeled BALANCE-DECREASE-ASSETS-ALL-REMOVED, denotes a critical severity level, signaling a complete depletion of a protocol's token balance, while the BALANCE-DECREASE-ASSETS-PORTION-REMOVED alert, marked as medium severity, signifies a notable yet incomplete reduction. These alerts furnish metadata encompassing initial and final transaction hashes, the affected asset, and an anomaly score quantifying the event's abnormality. Implementation-wise, the bot employs ARIMA for predictive analysis, evaluates current token balances via smart contracts' balanceOf calls, and monitors Transfer events to track token movements, accommodating both ERC-20 tokens and native blockchain tokens. Utilizing persistence and anomaly detection mechanisms, it calculates anomaly scores by comparing observed events against the total number of relevant transactions, offering insights into the event's rarity. Use cases include safeguarding DeFi protocol security by flagging potential exploits or rug pulls and alerting stakeholders about significant balance fluctuations affecting protocol operation or token economics.
C. The High Percentage of Profitable Trade Detection Bot is tailored to monitor the ALPERP protocol on the Binance Smart Chain (BSC), aiming to pinpoint accounts exhibiting an abnormally high frequency of profitable swap transactions. Upon detecting such activity, the bot generates an alert flagged as \"suspicious\" with a medium severity level. To trigger this alert, an account's trading behavior must surpass a default threshold of over 90% profitable trades, following a grace period of 5 trades, both adjustable in the `src/agent.config.ts` file. The bot's source code comprises several components: 'agent.spec.js' for testing, 'agent.config.js' for configurations, 'utils.js' (currently empty), and 'agent.js' housing the core logic. Within 'agent.js,' functionalities include setting up Chainlink price feed interfaces, managing trade history state, and scrutinizing swap events to detect suspiciously high profitable trade ratios, potentially indicative of manipulative tactics like front-running. Integrated into the Forta framework, this bot contributes to safeguarding the integrity of ALPERP protocol trading activities by alerting participants to potentially illicit behaviors

Expected result description: This web3 bot appears to be designed to monitor transactions on decentralized exchanges (DEXs) like Uniswap and others. It seems to analyze transactions for specific swap functions and events related to token transfers. The bot is structured to handle transactions, filter relevant events, and provide findings based on the executed swap functions and associated events. Additionally, it incorporates network management functionality to adapt to different blockchain networks. Overall, the bot seems focused on detecting and analyzing swap transactions and associated events within decentralized exchange environments.

Prompt #3: Is there a bot that detects scam activity preferably based on notification EOAs.

3. Is there a bot that detects scam activity preferably based on notification EOAs.

Expected:Scam Notifier Bot

Actual Result (Certainty):
1. Scam Detector Feed (beta2) - 0.9296514987945557
2. scam-notifier-bot - 0.9295835494995117
3. Evidence of Phishing Agent - 0.9236517548561096\

Result: Expected result is #2 ✅

---

1. The purpose of the specific web3 bot is to gather and analyze data related to blockchain transactions, particularly focusing on Ethereum and ERC20/ERC721 tokens. It collects information such as transaction amounts, gas prices, token transfers, and contract interactions. The bot's source code includes functionality to query a data source using GraphQL, retrieve transaction data, aggregate it, and construct a graph representation for further analysis. Additionally, there is an autoencoder neural network model implemented for encoding transaction and address data, likely for feature extraction or anomaly detection purposes.
2. The Web3 Bot, also known as the SCAM NOTIFIER BOT, serves the purpose of monitoring on-chain transactions to identify potential scam activities. By scrutinizing transaction data for suspicious messages and sender behavior, such as known notifier addresses, it flags addresses possibly linked to scams. Depending on the content and sender status, it generates alerts like SCAM-NOTIFIER-EOA, SCAM-NOTIFIER-CONTRACT, VICTIM-NOTIFIER-EOA, or NEW-SCAM-NOTIFIER. Additionally, it can elevate regular addresses to notifier status upon flagging at least two scam addresses. Utilizing a Neo4j database, it stores scam-related transactions and maintains records of notifiers and potential scam addresses. Primarily focused on Ethereum blockchain transactions, each alert includes a description of suspicious activity, severity level, exploit type, and relevant metadata such as addresses and messages, aiming to keep users informed about potential scams in the Web3 ecosystem

Prompt #6: Recommend me a bot that acts like a pretend version of a smart contract and its functions. Preferably the one testing in a development setting to mimic how a contract works without needing real blockchain setups or network links.

6. Recommend me a bot that acts like a pretend version of a smart contract and its functions. Preferably the one testing in a development setting to mimic how a contract works without needing real blockchain setups or network links.

   Expected:malicious-smart-contract-ml-v3-beta

Actual Result (Certainty):
1. suspicious-contract-creation - 0.9118551909923553
2. token-impersonation-bot - 0.9111163318157196
3. BlockSec Phishing Alert - 0.90456822514534\

Result: Expected result is not there but that's because there is no description available for this bot.

Prompt #8: Is there any bot sifts through transaction logs to find relevant events using their signatures and the contract address.

8. Is there any bot sifts through transaction logs to find relevant events using their signatures and the contract address.

   Expected:Forta Staking Events

Actual Result (Certainty): 1. pstake-stkbnb-operational-event-bot - 0.92950838804245
2. Passphrase detection bot - 0.9269896149635315
3. Address Poisoning Detector - 0.924113333225250
.....
8. Forta Staking Events - 0.9196819067001343
Result: Expected result is #8 ❌

1. The web3 bot described within the provided source code appears to be a monitoring agent designed to analyze Ethereum smart contract transactions. It is configured to detect specific events emitted by smart contracts and generate alerts based on predefined criteria. The bot is initialized with configuration parameters such as contract addresses, ABIs, and event expressions, allowing it to filter transaction logs and identify relevant events. Upon initialization, the bot sets up event listeners for each contract and event of interest. When handling transactions, it iterates through the logs of each contract's events, evaluating any associated expressions to determine if alerts should be generated. The alerts contain information about the emitted event, including contract details, event name, severity, and any associated conditions. The bot's purpose seems to be focused on monitoring smart contract activity and alerting users to specific events based on predefined criteria.

2. The bot is designed to monitor Ethereum smart contracts, alerting on specific transaction patterns to detect critical and informational events, potentially indicating vulnerabilities or noteworthy activities. It surveils changes like signer or ownership alterations, whitelist modifications, reward claims, and gaming-related actions. Configuration changes are facilitated through an underlying management system, while transaction logs and event emissions on the Ethereum blockchain are analyzed based on the settings in 'bot-config.json'. The bot's implementation involves utility functions, validation, and configuration logic to ensure accuracy and facilitate evaluation, parsing, and state management. The source code comprises various components like utility tests, configuration validation, Ethereum interaction, and core bot logic, all structured to allow extensibility and adaptation for diverse smart contract monitoring scenarios.
3. The web3 bot described is designed to detect suspicious transactions related to address poisoning on the Ethereum blockchain. It analyzes transaction logs to identify potential phishing attempts, including zero-value, low-value, and fake-token address poisoning. The bot employs heuristics and rules to flag transactions involving known phishing contracts or exhibiting suspicious behavior. It also tracks anomaly scores and maintains lists of detected phishing contracts for efficiency. The source code comprises functions to parse transaction logs, check for known phishing contracts, and assess transaction characteristics to determine potential threats. Additionally, it includes mock classes for testing purposes, enabling simulated interactions with the Ethereum blockchain. Overall, the bot aims to enhance security by identifying and alerting on potentially malicious activities within the Ethereum ecosystem.

Expected Result Description:

Prompt #9: Which bot examines transactions and block events to find notable decreases in token balances, which could signal possible exploitation or malicious behavior.

9. Which bot examines transactions and block events to find notable decreases in token balances, which could signal possible exploitation or malicious behavior.

   Expected:Large Balance Decrease - zkSync Era Diamond Bridge
   Actual Result (Certainty):
   A. Large Balance Decrease - Near Rainbow Bridge - 0.9578222632408142
   B. Large Balance Decrease - Optimism Gateway V1 Bridge - 0.9544870853424072
   C. Large Balance Decrease - Hop Protocol DAI Bridge - 0.9542330503463745

Result: Expected result is not in top 10 but that's because most of 'Large Balance Decrease' bots fit the prompt as it is common to them so this result might not reflect its capabilites.

A. The Web3 bot outlined in the provided code is designed to monitor and raise alerts for significant decreases in token balances within a specified protocol across various blockchain networks, including Ethereum, Optimism, Binance Smart Chain, Polygon, Fantom, Arbitrum, and Avalanche. It generates two main types of alerts: one triggered when the entire token balance is drained, denoting critical severity and labeled as an

exploit, and another when a substantial portion of the balance is removed, classified with medium severity. These alerts contain metadata such as transaction hashes, affected assets, percentage decreases, and anomaly scores to indicate the rarity of the event. The bot employs transaction filtering, balance checks, and statistical models, possibly including ARIMA for time series analysis, to identify suspicious reductions. It also assigns labels to transactions and addresses involved, categorizing them as \"Suspicious,\" \"Victim,\" or \"Attacker,\" aiding in understanding transaction nature and roles. The code includes setups for handling transactions and blocks, initializing the bot, monitoring events, computing anomaly scores, and storing historical data for trend analysis. Overall, this bot serves as a security measure to detect and notify unauthorized withdrawals or transfers from a protocol, potentially indicating malicious activities like exploits or hacks, facilitating timely response to security incidents.

B. The Large Balance Decrease Bot is engineered to oversee crypto wallets or contracts across various blockchain networks, detecting notable reductions in token balances, which may suggest potential exploits. It operates on Ethereum, Optimism, Binance Smart Chain, Polygon, Fantom, Arbitrum, and Avalanche. Alerts are issued for complete asset drainage or significant token removal, categorized by severity and exploit type. By monitoring transfer events and calculating balance changes, anomalies are flagged, leveraging an ARIMA model for time series forecasting and anomaly scoring. The bot ensures state persistence through database operations and can be configured via the `bot-config.json` file, with testing capabilities using specified block numbers and contract addresses. Its source code comprises unit tests, database interaction helpers, and the main bot logic for transaction handling and anomaly detection. In essence, it acts as a financial security measure for blockchain ecosystems, furnishing timely notifications of critical balance reductions indicative of potential exploitation, facilitating prompt corrective measures by stakeholders.
C. The Web3 bot is designed to monitor and identify significant decreases in token balances within specified protocols across various blockchain platforms including Ethereum, Optimism, Binance Smart Chain, Polygon, Fantom, Arbitrum, and Avalanche. It alerts users to two types of events: critical decreases where a protocol's token balance is entirely drained, potentially indicating an exploit, and medium severity decreases where a substantial portion of the balance is removed. The bot labels transactions and addresses as \"Suspicious,\" \"Victim,\" and \"Attacker\" as appropriate, and calculates anomaly scores based on event frequency. The primary components of the source code include the main bot logic in `agent.js`, a data persistence manager in `persistence.helper.js`, and unit tests in `agent.spec.js`. The bot also handles data aggregation, manages storage options, and provides a walkthrough for testing its behavior. It utilizes environment variables for configuration and supports both local and remote data storage.

Expected Description:
The \"Large Balance Decrease Bot\" is designed to monitor smart contracts across multiple blockchain networks to detect significant decreases in protocol balances, potentially signaling exploits. It operates on Ethereum, Optimism, Binance Smart Chain, Polygon, Fantom, Arbitrum,

and Avalanche. It issues alerts of two types: \"BALANCE-DECREASE-ASSETS-ALL-REMOVED\" for complete token balance drainage and \"BALANCE-DECREASE-ASSETS-PORTION-REMOVED\" for substantial reductions. Metadata includes transaction hashes, affected assets, and anomaly scores. The code analyzes transactions using ARIMA modeling to predict normal fluctuations and identifies deviations. It persists data in an external database and undergoes rigorous unit testing. Overall, the bot aims to provide crucial security alerts for smart contracts experiencing abnormal balance decreases, potentially indicative of exploits or attacks.

Prompt #11: Which bot identifies transactions that involve substantial Tether transfers, particularly on the blockchain like Ethereum

11. Which bot identifies transactions that involve substantial Tether transfers, particularly on the blockchain like Ethereum

Expected: forta-agent-starter

Actual Result (Certainty): 1. harry-bot-threatintel - 0.9474129676818848
2. testbot - 0.9469777345657349
3. bot-heartbeat-beta - 0.945927768945694
4. VDODO-Account-Balance-Bot - 0.9444511830806732
....
7. forta-agent-starter - 0.9430810213088989\

Result: Expected result is #7 but all the recommended bots fit the prompt. (checked description) ✅

1. The bot's primary objective is to monitor the Ethereum blockchain for signs of malicious activities, particularly those involving the unauthorized draining of funds from users, such as scam transactions. It analyzes smart contract activities, paying particular attention to Tether (USDT) transfers associated with potential malicious behavior. The code structure delineates modules for detecting various indicators of scams, including the creation of new smart contracts, suspicious Tether transfers using specific functions, and potentially deceptive function signatures. The bot employs severity levels to manage alert frequency and utilizes real-time Tether price data to assess potential financial impacts. Additionally, it maintains a list of addresses to ignore, likely to filter out known benign entities.
2. The purpose of the specific web3 bot described in the provided source code is to monitor and analyze Ethereum blockchain transactions and blocks for certain events, specifically related to Tether (USDT) transfers and suspicious transactions. The bot is designed to generate findings or alerts based on specific criteria. 1. For Tether Transfers: - It checks for Tether (USDT) token transfers. - If there are no Tether transfers in a transaction, it returns empty findings. - If a Tether transfer with an amount exceeding 10,000 USDT is detected, it generates a finding with details about the transaction, such as the sender, recipient, and the transferred amount. 2. For Suspicious Transactions: - It fetches results from a remote HTTP server based on the block number and network. - If the

results contain any data (indicating a suspicious transaction), it generates a finding with the name \"Suspicious Tx,\" providing the description from the results. The bot's primary function is to identify and report Tether transfers exceeding a certain threshold and to report suspicious transactions based on external analysis results fetched from the HTTP server. It uses the FORTA framework for handling findings, including severity and type classifications.

3. The \"Large Tether Transfer Agent\" operates on the Ethereum blockchain to identify Tether (USDT) transactions surpassing 10,000 USDT. Upon detection, it triggers an alert tagged with the ID FORTA-1, indicating low severity and informational type. These alerts contain sender and receiver addresses for the flagged transactions. The agent serves to inform about significant USDT movements, potentially shedding light on large capital shifts or impactful events in the cryptocurrency domain. The source code provides insights into initializing the agent, managing blockchain transactions, and executing the detection logic within the `handleTransaction` function, which focuses on filtering and evaluating Tether transfers to meet the predefined criteria.

Expected Results Description: The specific web3 bot described is a \"Forta watcher\" designed to monitor high tether transfer activity on the Ethereum blockchain. It operates by analyzing transaction events and identifying instances where specific function signatures associated with high tether transfers are invoked. The bot initializes by reading configuration data from external files and setting up contracts and associated function signatures to monitor. When a relevant transaction event is detected, the bot generates findings containing details such as contract name, function name, alert ID, severity, and metadata like transaction hash. The source code consists of JavaScript files responsible for initializing the bot, handling transaction events, and creating findings based on predefined criteria. It utilizes the Forta Agent framework for efficient interaction with the Ethereum blockchain and abstraction of blockchain-specific functionalities. Overall, the bot aims to enhance security by promptly detecting and alerting users to high tether transfer activities on Ethereum.

Prompt #12: Is there a bot Overall, the bot aims to enhance security by identifying and alerting on potentially malicious activities within the Ethereum ecosystem.

12. Is there a bot Overall, the bot aims to enhance security by identifying and alerting on potentially malicious activities within the Ethereum ecosystem.

    Expected:Tornado Cash Funded Account Interaction

Actual Result (Certainty):
1. BlockSec Phishing Alert - 0.9480542242527008
2. harry-bot-threatintel - 0.9443542957305908
3. phishing-scam-detection-ml-bot - 0.9405215382575989\

Result: Expected result is not in top 10 ❌ but the prompt is very vague and other bots do satisfy prompts needs according to descriptions.

1. The Web3 bot, as depicted in the provided code, serves the primary function of monitoring and alerting against various forms of phishing and scams occurring on the blockchain, with a particular focus on Ethereum. It's equipped to identify different types of fraudulent activities, including ice-phishing, fraudulent NFT orders, address poisoning,

native ice-phishing, and phishing website detection. The bot's source code comprises JavaScript modules detailing its interaction with the blockchain, handling of smart contract transactions, categorization of alerts, and encryption of sensitive data. It utilizes APIs and tools like Moralis and AlchemySDK to gather transaction details and employs predefined patterns and signatures to analyze transaction data and generate alerts. Noteworthy components include modules for defining constants and ABIs, processing phishing transactions, managing alerts, handling specific marketplace transactions, recognizing phishing attacks through contract interactions, and communicating with external APIs like Etherscan. The bot's operation involves real-time monitoring, encrypted alert generation, and periodic updates of token data. Overall, it's a comprehensive security tool designed to safeguard users' digital assets by proactively detecting and mitigating blockchain-based phishing and scamming activities.

2. The bot's primary objective is to monitor the Ethereum blockchain for signs of malicious activities, particularly those involving the unauthorized draining of funds from users, such as scam transactions. It analyzes smart contract activities, paying particular attention to Tether (USDT) transfers associated with potential malicious behavior. The code structure delineates modules for detecting various indicators of scams, including the creation of new smart contracts, suspicious Tether transfers using specific functions, and potentially deceptive function signatures. The bot employs severity levels to manage alert frequency and utilizes real-time Tether price data to assess potential financial impacts. Additionally, it maintains a list of addresses to ignore, likely to filter out known benign entities.
3. The web3 bot described is tailored to detect potential phishing scammers within the Ethereum blockchain using machine learning (ML) techniques. It employs an ensemble approach with the EasyEnsembleClassifier and lightgbm classifiers to handle imbalanced datasets. By analyzing transaction patterns and behaviors, including features like transaction counts, block numbers, and transaction values, the bot assigns a score between 0 and 1 to each address, with higher scores indicating a greater likelihood of being involved in phishing activities. Alerts are triggered when the prediction score surpasses a predefined threshold, leading to the flagging of addresses as potential phishing scammers. Performance metrics derived from evaluation against a test dataset include a precision of 0.69, recall of 0.44, and F1-score of 0.54. The bot's functionality is encapsulated within the `agent.py` file, orchestrating the analysis and employing concurrency for efficient processing. Supported chains currently include only the Ethereum blockchain, with findings labeled and formatted for further investigation upon detection of phishing activity.

Expected Result Description: This web3 bot is designed to detect interactions between accounts funded by Tornado Cash and any non-Tornado Cash contracts on supported chains like Ethereum, BNB Smart Chain, Optimism, Polygon, and Arbitrum. It identifies such interactions as suspicious and assigns them a low severity level. The bot calculates an anomaly score based on the frequency of these interactions compared to total contract interactions processed, with variations across different chains. The purpose is to alert users about potential misuse of funds anonymized through Tornado Cash. The bot's source code includes functions to initialize the

bot, handle transactions, and manage cache for efficient processing. It utilizes external APIs for rate calculation and maintains a cache for contract code retrieval to optimize performance.

Prompt #14: Which bot monitors blockchain smart contracts, primarily on Ethereum and triggers alerts when specific conditions are met during transaction activities, focusing on smart contract interactions?

14. Which bot monitors blockchain smart contracts, primarily on Ethereum and triggers alerts when specific conditions are met during transaction activities, focusing on smart contract interactions?

    Expected:hal-highstreet-standard-monitoring
    Actual Result (Certainty):
    A. pstake-stkbnb-operational-event-bot - 0.9455776810646057\
    B. Passphrase detection bot - 0.9444133043289185\
    C. BlockSec Phishing Alert - 0.9405323565006256
    ....\
    D. hal-highstreet-standard-monitoring - 0.9351710081100464\

Result: Expected result is #10 but the prompt is vague and other bots do satisfy prompts needs according to descriptions and reasons provided.

A. The web3 bot described within the provided source code appears to be a monitoring agent designed to analyze Ethereum smart contract transactions. It is configured to detect specific events emitted by smart contracts and generate alerts based on predefined criteria. The bot is initialized with configuration parameters such as contract addresses, ABIs, and event expressions, allowing it to filter transaction logs and identify relevant events. Upon initialization, the bot sets up event listeners for each contract and event of interest. When handling transactions, it iterates through the logs of each contract's events, evaluating any associated expressions to determine if alerts should be generated. The alerts contain information about the emitted event, including contract details, event name, severity, and any associated conditions. The bot's purpose seems to be focused on monitoring smart contract activity and alerting users to specific events based on predefined criteria.
B. The bot is designed to monitor Ethereum smart contracts, alerting on specific transaction patterns to detect critical and informational events, potentially indicating vulnerabilities or noteworthy activities. It surveils changes like signer or ownership alterations, whitelist modifications, reward claims, and gaming-related actions. Configuration changes are facilitated through an underlying management system, while transaction logs and event emissions on the Ethereum blockchain are analyzed based on the settings in 'bot-config.json'. The bot's implementation involves utility functions, validation, and configuration logic to ensure accuracy and facilitate evaluation, parsing, and state management. The source code comprises various components like utility tests, configuration validation, Ethereum interaction, and core bot logic, all structured to allow extensibility and adaptation for diverse smart contract monitoring scenarios.
C. The Web3 bot, as depicted in the provided code, serves the primary function of monitoring and alerting against various forms of phishing and scams occurring on the blockchain, with a particular focus on Ethereum. It's equipped to identify different types of fraudulent activities, including ice-phishing, fraudulent NFT orders, address poisoning,

native ice-phishing, and phishing website detection. The bot's source code comprises JavaScript modules detailing its interaction with the blockchain, handling of smart contract transactions, categorization of alerts, and encryption of sensitive data. It utilizes APIs and tools like Moralis and AlchemySDK to gather transaction details and employs predefined patterns and signatures to analyze transaction data and generate alerts. Noteworthy components include modules for defining constants and ABIs, processing phishing transactions, managing alerts, handling specific marketplace transactions, recognizing phishing attacks through contract interactions, and communicating with external APIs like Etherscan. The bot's operation involves real-time monitoring, encrypted alert generation, and periodic updates of token data. Overall, it's a comprehensive security tool designed to safeguard users' digital assets by proactively detecting and mitigating blockchain-based phishing and scamming activities.

Expected Result Description:
The provided web3 bot is a sophisticated JavaScript program crafted to engage with blockchain smart contracts, detecting specific conditions or patterns during transaction activities. It operates by scrutinizing blockchain transactions and triggering alerts when predefined criteria are fulfilled, primarily focusing on smart contract interactions within the Ethereum blockchain. Structured as a module with two core functions, namely `initialize` and `handleTransaction`, it conducts initialization tasks and processes transaction objects to identify rule violations. The source code comprises crucial elements like ABI utility functions, comparison mechanisms for values, address and function call monitoring, standardized finding generation, expression evaluation, and integration with the Forta agent framework. Additionally, the bot features obfuscated code for security and may rely on specific configurations and databases for proper functioning. In essence, it serves as a customizable Ethereum smart contract monitor, issuing alerts based on configured rules and conditions.

Prompt #15:
Very similar to Prompt #2

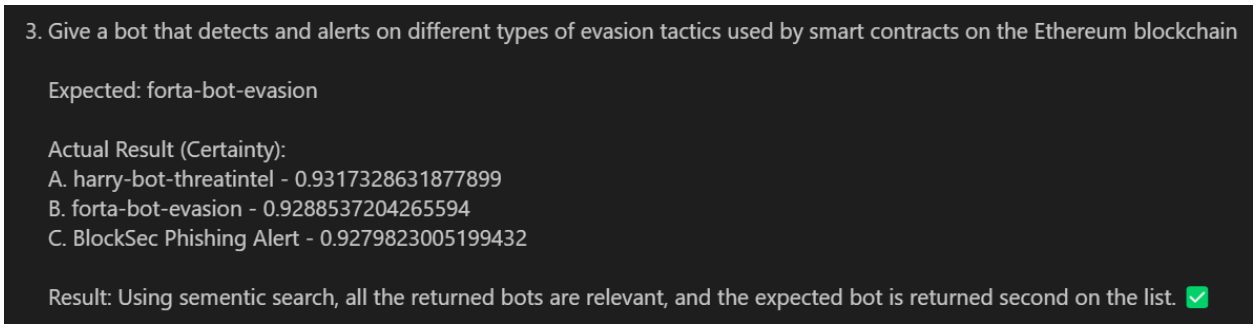
KEYWORDS:

- DIRECT QUESTIONS:

  Actual Result is #1: **1, 2, 5, 6, 7, 10, 11, 12**
  Actual Result is top #5: **3- 2nd, 4- 2nd, 9- 2nd, 13- 3rd**
  Actual Result is not top #5: **8**

Prompt #3: Give a bot that detects and alerts on different types of evasion tactics used by smart contracts on the Ethereum blockchain.

Actual Result is #2



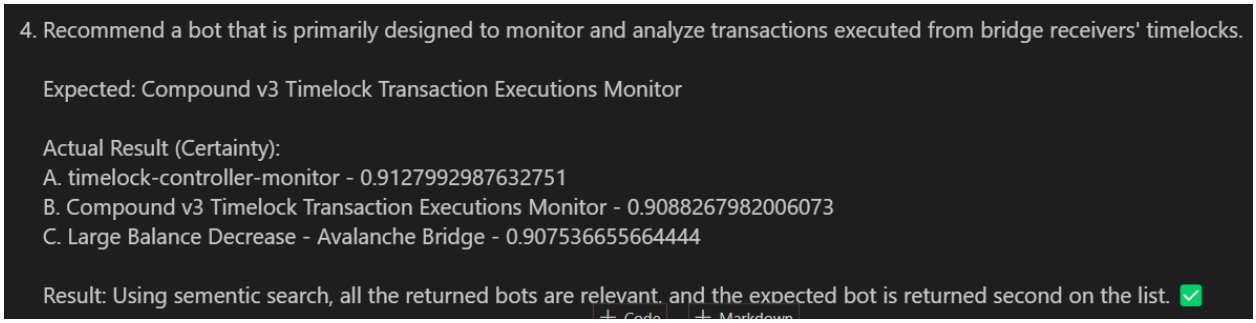
A. Description: The bot's primary objective is to monitor the Ethereum blockchain for signs of malicious activities, particularly those involving the unauthorized draining of funds from users, such as scam transactions. It analyzes smart contract activities, paying particular attention to Tether (USDT) transfers associated with potential malicious behavior. The code structure delineates modules for detecting various indicators of scams, including the creation of new smart contracts, suspicious Tether transfers using specific functions, and potentially deceptive function signatures. The bot employs severity levels to manage alert frequency and utilizes real-time Tether price data to assess potential financial impacts. Additionally, it maintains a list of addresses to ignore, likely to filter out known benign entities.

B. Description: This web3 bot specializes in detecting and alerting on various evasion tactics employed by smart contracts on the Ethereum blockchain, with a focus on metamorphic contracts and red pill contracts. It employs transaction traces and static analysis of bytecode to identify these tactics. Metamorphic contracts dynamically alter their bytecode while maintaining the same address, facilitated by the CREATE2 opcode, and the bot scrutinizes factory contracts deploying these \"mutant\" contracts, issuing alerts categorized as \"Info\" and \"Suspicious.\" Red pill contracts are designed to

discern simulation environments and activate malicious functions solely on the mainnet, with the bot examining bytecode patterns indicative of such behavior. Transaction assessments are probabilistic, with scores assigned to indicators and an overall probability calculated. Alerts are triggered only if probabilities surpass a predefined confidence level set in `options.py`. Key components of the source code include `options.py`, `findings.py`, `utils.py`, and `agent.py`, with the latter containing the primary logic for transaction analysis. While specifics of the `ioseeth` library are not provided, it is referenced in `agent.py` for defining metrics and indicators. Overall, the bot operates within Ethereum to identify suspicious smart contract behaviors, offering alerts for further scrutiny or auditing.

Prompt #4: Recommend a bot that is primarily designed to monitor and analyze transactions executed from bridge receivers' timelocks.

Actual Result is #2

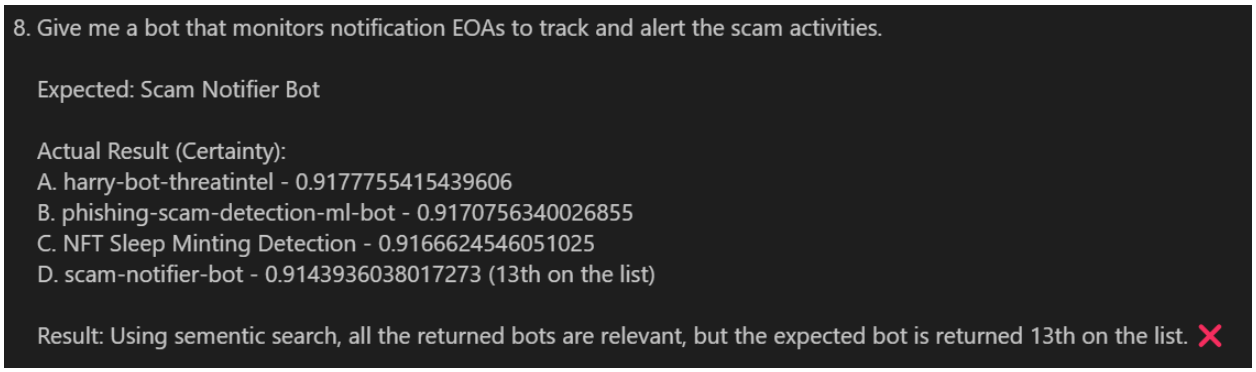


A. The specific web3 bot is designed to handle transactions related to role changes in a Timelock Controller contract on the Ethereum blockchain. It parses transaction logs to detect role grants, revokes, and renouncements, generating findings accordingly. Additionally, it distinguishes between different roles and assesses the severity of role changes based on predefined criteria. This bot contributes to monitoring and enforcing access control policies within smart contracts, ensuring the integrity and security of decentralized applications.

B. The Timelock Transaction Executions Monitor Bot operates as a web3 bot on the Polygon blockchain, primarily focusing on monitoring and analyzing transactions executed from bridge receivers' timelocks. It listens for `ExecuteTransaction` events, matches them with previous proposal logs, and emits findings accordingly. Findings include notifications for successfully executed proposals, alerts for incomplete executions, and suspicious flags for unlinked transactions. Debugging logs aid in cases where proposal parameters are inaccessible due to historical data fetching. Configuration is customizable through `agent.config.ts`, supporting Polygon and providing sample test data. The source code comprises various modules like `constants.js`, `agent.spec.js`, `finding.js`, `utils.js`, and `agent.js`, which collectively integrate the bot's functionality. Ultimately, it acts as an automated auditor, ensuring governance transaction transparency and integrity for Polygon stakeholders.

Prompt #8: Give me a bot that monitors notification EOAs to track and alert the scam activities.

Actual Result is #13



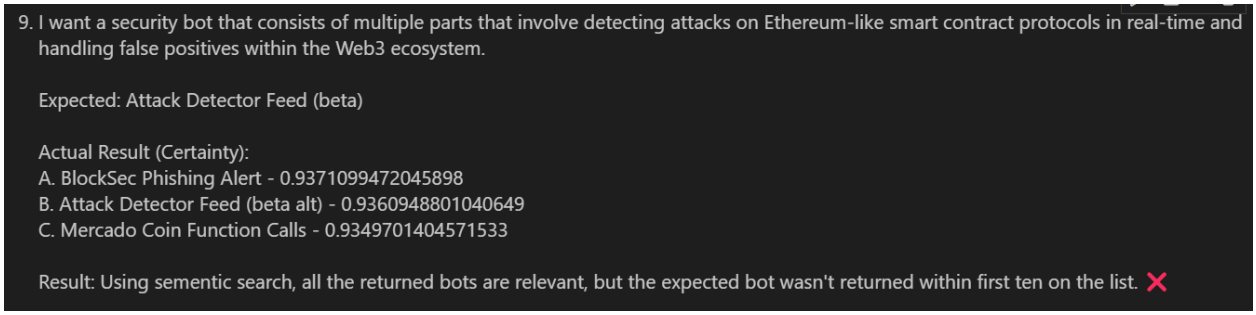
- **A.** The bot's primary objective is to monitor the Ethereum blockchain for signs of malicious activities, particularly those involving the unauthorized draining of funds from users, such as scam transactions. It analyzes smart contract activities, paying particular attention to Tether (USDT) transfers associated with potential malicious behavior. The code structure delineates modules for detecting various indicators of scams, including the creation of new smart contracts, suspicious Tether transfers using specific functions, and potentially deceptive function signatures. The bot employs severity levels to manage alert frequency and utilizes real-time Tether price data to assess potential financial impacts. Additionally, it maintains a list of addresses to ignore, likely to filter out known benign entities.
- **B.** The web3 bot described is tailored to detect potential phishing scammers within the Ethereum blockchain using machine learning (ML) techniques. It employs an ensemble approach with the EasyEnsembleClassifier and lightgbm classifiers to handle imbalanced datasets. By analyzing transaction patterns and behaviors, including features like transaction counts, block numbers, and transaction values, the bot assigns a score between 0 and 1 to each address, with higher scores indicating a greater likelihood of being involved in phishing activities. Alerts are triggered when the prediction score surpasses a predefined threshold, leading to the flagging of addresses as potential phishing scammers. Performance metrics derived from evaluation against a test dataset include a precision of 0.69, recall of 0.44, and F1-score of 0.54. The bot's functionality is encapsulated within the `agent.py` file, orchestrating the analysis and employing concurrency for efficient processing. Supported chains currently include only the Ethereum blockchain, with findings labeled and formatted for further investigation upon detection of phishing activity.
- **C.** This bot detects various suspicious activities related to NFT transactions on the Ethereum network. It encompasses functionalities to identify both NFT transfer mismatches and approval mismatches, allowing it to comprehensively monitor and alert on potential security issues associated with NFT transfers and approvals. The bot's

source code includes functions to filter relevant events, analyze transaction details, and generate findings accordingly, alerting users to potential security risks in NFT transactions.

Expected Description: The Web3 Bot, also known as the SCAM NOTIFIER BOT, serves the purpose of monitoring on-chain transactions to identify potential scam activities. By scrutinizing transaction data for suspicious messages and sender behavior, such as known notifier addresses, it flags addresses possibly linked to scams. Depending on the content and sender status, it generates alerts like SCAM-NOTIFIER-EOA, SCAM-NOTIFIER-CONTRACT, VICTIM-NOTIFIER-EOA, or NEW-SCAM-NOTIFIER. Additionally, it can elevate regular addresses to notifier status upon flagging at least two scam addresses. Utilizing a Neo4j database, it stores scam-related transactions and maintains records of notifiers and potential scam addresses. Primarily focused on Ethereum blockchain transactions, each alert includes a description of suspicious activity, severity level, exploit type, and relevant metadata such as addresses and messages, aiming to keep users informed about potential scams in the Web3 ecosystem.

Prompt #9: I want a security bot that consists of multiple parts that involve detecting attacks on Ethereum-like smart contract protocols in real-time and handling false positives within the Web3 ecosystem.

Actual Result wasn't returned within top ten



A. The Web3 bot, as depicted in the provided code, serves the primary function of monitoring and alerting against various forms of phishing and scams occurring on the blockchain, with a particular focus on Ethereum. It's equipped to identify different types of fraudulent activities, including ice-phishing, fraudulent NFT orders, address poisoning, native ice-phishing, and phishing website detection. The bot's source code comprises JavaScript modules detailing its interaction with the blockchain, handling of smart contract transactions, categorization of alerts, and encryption of sensitive data. It utilizes APIs and tools like Moralis and AlchemySDK to gather transaction details and employs predefined patterns and signatures to analyze transaction data and generate alerts. Noteworthy components include modules for defining constants and ABIs, processing phishing transactions, managing alerts, handling specific marketplace transactions, recognizing phishing attacks through contract interactions, and communicating with

external APIs like Etherscan. The bot's operation involves real-time monitoring, encrypted alert generation, and periodic updates of token data. Overall, it's a comprehensive security tool designed to safeguard users' digital assets by proactively detecting and mitigating blockchain-based phishing and scamming activities.

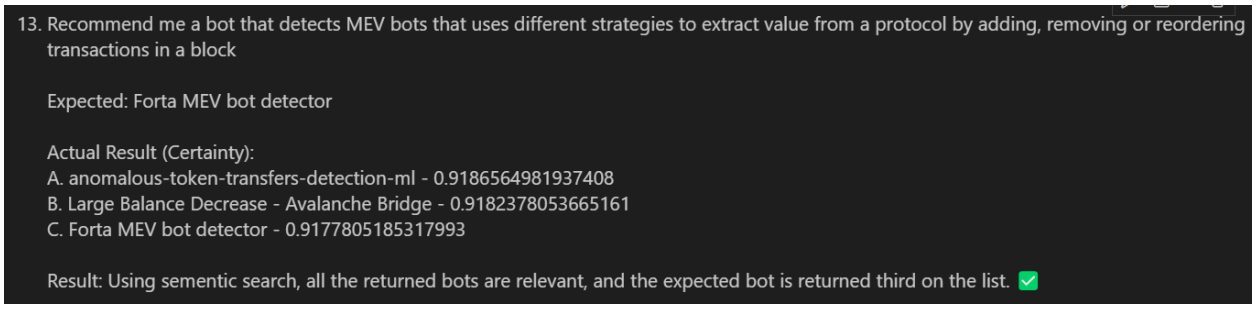
B. The provided source code represents a web3 bot responsible for handling alerts and blocks in a blockchain environment. The bot's functionality includes detecting attacks, emitting findings related to suspicious activities such as fraudulent transactions, and persisting bot state. It leverages external data sources such as manual alert lists and false positive lists to enhance its detection capabilities. Additionally, the bot emits findings for potential false positives and mitigates them accordingly. The code demonstrates the bot's ability to interact with the blockchain by querying blocks and contracts to extract relevant information for analysis. Moreover, it showcases mechanisms for handling exceptions and ensuring the bot's state persistence for uninterrupted operation. Overall, the bot's purpose revolves around monitoring and responding to alerts and blocks in a blockchain network to detect and mitigate potential threats and suspicious activities.

C. The provided source code outlines a web3 bot designed to monitor Ethereum smart contracts and transactions for various conditions. Built with JavaScript and utilizing the Forta SDK, the bot features modules for detecting contract events, tracking transactions, monitoring variable values, validating configurations, and providing utility functions for Ethereum-specific tasks. With its ability to alert on specific events, function calls, and threshold breaches, the bot serves as a comprehensive monitoring agent for Ethereum blockchain activity, aimed at enhancing security and operational maintenance.

Expected Description: The provided code comprises several components designed to detect attacks on Ethereum-like smart contract protocols in real-time and manage false positives within the Web3 ecosystem. The Attack Detector Feed combines alerts from various base bots, employing anomaly detection and heuristic approaches to identify attacks across four stages: Funding, Preparation, Exploitation, and Money Laundering. DynamoUtils facilitates interaction with DynamoDB for storing and managing detected attacks and entities. Findings module creates structured alerts for better understanding detected activities. BlockChainIndexerService interacts with blockchains to retrieve contract information. L2Cache provides a caching layer for persistent data storage. Utils Test contains test cases for utility functions, while DynamoUtilsTest ensures proper DynamoDB interaction. Constants module holds various thresholds and keys. BlockchainIndexerServiceMock aids in testing without actual external calls. AgentTest validates the agent's functionality, while Agent integrates all components to detect and manage security incidents. Web3Mock and BlockchainIndexerServiceTest are for testing purposes. Tabular predicts false positives, and FPMitigator utilizes models for mitigation. DownloadData fetches and formats transaction-related data, and Autoencoder defines a model for encoding graph data, useful for anomaly detection. Together, these components form a bot within the Forta detection network for real-time security incident detection on the blockchain.

Prompt #13: Recommend me a bot that detects MEV bots that uses different strategies to extract value from a protocol by adding, removing or reordering transactions in a block

Actual Result is #3



A. The detection bot described in the provided source code aims to identify anomalous transactions involving token transfers on the Ethereum blockchain. It leverages machine learning models to assess transaction features and determine whether a transaction is normal or suspicious. By analyzing various attributes such as token types, transfer counts, and account activity periods, the bot assesses the likelihood of an anomaly and emits findings accordingly. The bot's purpose is to help monitor token transactions and identify potential security threats or irregularities in Ethereum network activity.

B. The web3 bot described in the provided source code is designed to monitor and detect suspicious activities related to token transfers within a blockchain network. It tracks various metrics such as balance changes and transaction volumes to identify potential anomalies, such as significant decreases in asset holdings or unexpected transfer patterns. The bot utilizes historical data and statistical analysis techniques to assess the likelihood of suspicious behavior and generates alerts accordingly. Additionally, it leverages persistence functionality to store and retrieve data across multiple blockchain blocks, ensuring continuous monitoring and analysis over time. Overall, the bot aims to enhance security and integrity within the blockchain ecosystem by proactively identifying and mitigating potential threats related to token transfers.

C. The web3 bot described in the provided source code appears to be focused on inspecting and analyzing transactions on the Ethereum blockchain. It utilizes classifiers to identify and categorize different types of smart contracts and transactions, particularly focusing on decentralized finance (DeFi) protocols such as 0x and Balancer. The bot's purpose seems to be understanding and interpreting the activities happening within these protocols, including swaps, transfers, and other interactions, in order to provide insights or perform specific actions based on the observed behavior. The source code includes specifications for different protocols and functionalities, along with classifiers to parse and interpret various types of transactions and events.

2. PARAPHRASED QUESTIONS

Actual Result is #1: **1, 2, 4, 10, 13**
Actual Result is top #5: **5- 3rd, 8- 3rd, 11- 5th**
Actual Result is not top #5: **3- 9th, 6- N/A, 7- N/A, 9- N/A, 12- N/A, 14- N/A, 15- N/A**

Conclusion

Prompt Search

Result on Direct Questions
Actual Result is #1: **1, 2, 4, 5, 6, 7, 8, 10, 11, 12, 13**
Actual Result is top #5: **prompt 3- 4th , 9- 3rd , and 13- 2nd**
Total Number of Actual Result within top 5: **13**

Result on Paraphrased Questions
Actual Result is #1: **1, 4, 5, 7, 10, 13**
Actual Result is top #5: **prompt 3-2nd, prompt 15 - 4th**
Actual Result is not top #5: **prompt 2- N/A, prompt 8-8th, prompt 9-N/A, prompt 11 - 7th , prompt 12- N/A, prompt 6- N/A**
Total Number of Actual Result within top 5: **8**

Keywords Search

Result on Direct Questions
Actual Result is #1: **1, 2, 5, 6, 7, 10, 11, 12**
Actual Result is top #5: **3- 2nd, 4- 2nd, 9- 2nd, 13- 3rd**
Actual Result is not top #5: **8**

Total Number of Actual Result within top 5: **12**

Result on Paraphrased Questions
Actual Result is #1: **1, 2, 4, 10, 13**
Actual Result is top #5: **5- 3rd, 8- 3rd, 11- 5th**
Actual Result is not top #5: **3- 9th, 6- N/A, 7- N/A, 9- N/A, 12- N/A, 14- N/A, 15- N/A**

Total Number of Actual Result within top 5: **8**

In conclusion, prompt search performed better in terms of the number of actual results within the top 5 as well as the average ranking of the expected bot in the returned bot list.