Blockchain-Based Intrusion Detection System (IDS)

About:

This project is a secure, decentralized framework designed to detect, collect, and share malicious IP addresses across organizations using Machine Learning (ML) and Blockchain technology. It addresses the challenges of traditional IDS systems—like isolated detection, centralized failure points, and tamperable logs—by introducing a collaborative approach, where security data is immortalized on a public blockchain and updated in near real time.

How It Works:

1. Data Collection & Feature Engineering
    - Utilized the CICIDS2017 dataset, a benchmark intrusion detection dataset.
    - Extracted features like packet sizes, flow duration, and connection states.
2. Anomaly Detection using Machine Learning
    - Built a supervised ML classifier (e.g., Random Forest or Decision Tree) to detect malicious traffic.
    - Labeled IPs based on their behavior (normal vs. anomalous).
    - Output: List of malicious source IPs.
3. IP Blacklist Generation
    - Collected unique, attack-originating IPs.
    - Formatted into a blacklist that can be exported to CSV or JSON.
4. Smart Contract Deployment (Ethereum Sepolia Testnet)
    - Designed a Solidity smart contract with functionalities to:
        - Add malicious IPs.
        - Prevent duplicate submissions.
        - Allow global verification of reported IPs.
    - Used the Brownie framework to:
        - Compile and deploy the contract.
        - Interact via Python scripts (deploy.py).
        - Sign transactions using MetaMask or a test wallet.
    - The smart contract ensures immutability—once a malicious IP is recorded, it cannot be deleted or altered.
5. Real-Time Threat Sharing
    - Security appliances (e.g., firewalls, IDS sensors) can query the smart contract to fetch updated blacklists.
    - Promotes collaborative defense across networks.

Tech Stack:

Layer	Tools/Frameworks
ML	Python, Pandas, Scikit-learn, NumPy
Blockchain	Solidity, Ethereum (Sepolia), Brownie, Web3.py
Smart Contract Wallet	MetaMask
Dataset	CICIDS2017
Testing Tools	Infura

Smart Contract Features:

- **Function**: addMaliciousIP(ipAddress)
  Adds a new malicious IP after validation.
- **Function**: isBlacklisted(ipAddress)
  Allows any user/system to verify whether an IP is blacklisted.
- **Function**: getAllIPs()
  Fetches the entire blacklist for external security systems to use.

Benefits:

Benefit	Description
Tamper-Proof	Blockchain makes the blacklist immutable.
Distributed Trust	Removes reliance on a central authority.
Auditability	IP submissions are permanently verifiable.
Collaboration	Encourages cross-organizational defense.
Automation Ready	Smart contracts are easily integrated into modern IDS pipelines.

Real-World Applications:

- **ISPs** can subscribe to the blacklist to protect users in real-time.
- **Enterprises** can collaboratively build and maintain their own IP reputation networks.
- **Governments** can use it as a backbone for threat intelligence infrastructure.

Future Enhancements:

- Add **reputation scoring** to weigh IP severity.
- Use **zero-knowledge proofs** to protect sensitive data in private environments.
- Extend to detect **domain names**, URLs, or malware hashes.
- Integrate with **SIEM systems** and real-time threat feeds (e.g., MISP, AlienVault).