1. **Smart Contract Security:** What are the key security considerations when deploying smart contracts on Ethereum?

Ans:

Here are key security considerations involved in Deploying smart contracts on Ethereum:

1. Reentrancy Prevention:

Implement reentrancy protection to prevent malicious contracts from repeatedly calling back into vulnerable functions. Use the reentrancy Guard pattern

1. Gas Limit Considerations:

Be mindful of gas limits when deploying or interacting with contracts.

1. Avoiding Unnecessary Fallback Function:

Avoid using the fallback function (function () external payable) unless necessary. If used, ensure it has minimal gas cost and complexity to prevent potential reentrancy vulnerabilities.

1. Handling External Calls:

Carefully handle external calls, especially when interacting with untrusted contracts. Use checks-effects-interactions pattern to minimize reentrancy vulnerabilities.

1. Access Control:

Implement proper access control mechanisms to restrict unauthorized access to sensitive functions or data. Use modifiers like onlyOwner or onlyAdmin to restrict access.

1. Consider Immutable Feature:

Design contracts to be as immutable as possible once deployed. Minimize the use of functions that can alter critical parameters or configurations after deployment.

1. Caution When Generating Randomness:

Generating secure random numbers on the blockchain is challenging. Use trusted methods like Oracle for generating randomness in the smart contract.

1. Testing the Contract:

Carefully test the smart contract code under various conditions, including edge cases and adversarial scenarios. Use tools like automated testing frameworks and static analyzers.

1. Testing in Test Network:

Carefully test the smart contract code by deploying in test nets like Sepolia, Goerli etc under various conditions.