School: ..... Campus: .....

Academic Year: ..... Subject Name: ..... Subject Code: .....

Semester: ..... Program: ..... Branch: ..... Specialization: .....

Date: .....

Applied and Action Learning

(Learning by Doing and Discovery)

Name of the Experiement : Security First – Understanding Blockchain Attacks

Objective/Aim:

To understand the various types of security attacks in blockchain systems, analyze their working mechanisms, and explore possible defense strategies to secure blockchain networks.

Apparatus/Software Used:

- Remix IDE (for Ethereum smart contract testing)
- Ganache (local blockchain simulation)
- MetaMask (wallet interaction)

Theory/Concept:

Blockchain is designed to be secure and tamper-resistant, but vulnerabilities can still arise due to coding errors, consensus weaknesses, or network exploitation.
Understanding these attacks is crucial for designing secure blockchain systems.

Common Blockchain Attacks:

Attack Type	Description	Impact
51% Attack	When an entity controls more than 50% of mining power, allowing them to rewrite transactions.	Double spending, network control
Sybil Attack	Attacker creates multiple fake identities (nodes) to gain majority influence.	Disrupts consensus, causes spam
Double Spending Attack	Spending the same coins more than once by exploiting transaction timing.	Loss of funds, trust issues
Replay Attack	Resending a transaction on a different blockchain or network.	Transaction duplication
Smart Contract Vulnerability Attack	Exploiting logical or coding bugs in smart contracts (e.g., reentrancy).	Unauthorized withdrawals, data theft
DDoS Attack	Overloading nodes with traffic, causing service disruption.	Network downtime

Procedure:

- Create a Vulnerable Smart Contract
  Write a Solidity contract with an unsafe withdraw() function (no state update before transfer).
- Deploy the Contract on Remix IDE
  Connect MetaMask and deploy to a local Ganache blockchain.
- Write an Attacker Contract
  Create another contract that repeatedly calls the vulnerable contract's withdraw function before the balance updates.
- Execute the Attack
- Call the attack contract's function.
- Observe how funds are repeatedly withdrawn from the vulnerable contract.
- Analyze the Results
- Check the victim's contract balance (should become zero).
- Review the transaction logs to understand the sequence.
- Implement Fix
- Update the contract to use the Checks-Effects-Interactions pattern or ReentrancyGuard modifier.
- Redeploy and verify the fix.

Observation Table:

Step	Action	Expected Result	Observed Result
1	Deposit 1 Ether	Funds added to bank	✅ Successful
2	Deploy Attacker contract	Connected to bank	✅ Successful
3	Call attack()	Balance repeatedly withdrawn	⚠️ Bank drained
4	Fix code and redeploy	Funds protected	✅ Secure

ASSESSMENT

Rubrics	Full Mark	Marks Obtained	Remarks
Concept	10		
Planning and Execution/ Practical Simulation/ Programming	10		
Interpretation Result and	10		
Record of Applied and Action Learning	10		
Viva	10		
Total	**50**		

Signature of the Student:

Name :

Regn. No.

Signature of the Faculty: