HGame_Week2_WP

Vesperina#000cdc

Misc

1.Invest on Matrix

一些二维码共性：

1.都有三个定位码，位于二维码左上、右上、左下三个角。

2.定位码外围一圈必定是白色，大小约1像素

3.坐标（18，18）黑点及外围一圈，是一个小校验码

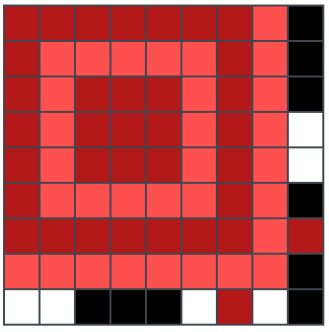
4.第6行和第6列除了定位码部分，其余都是黑白相间，用于连接定位码，防止扫描歪斜。

此外，二维码有纠错机制，如图中蓝色区域，详细了解发现它包含 15 位数据，记录了纠错等级（L/M/Q/H）和掩码模式（0-7）。

所以买一个2pts的hint可以知道所有蓝色区域的像素颜色（导致好几个区域买完后的有效信息只有一两行ww（还能省

Format Info Pattern

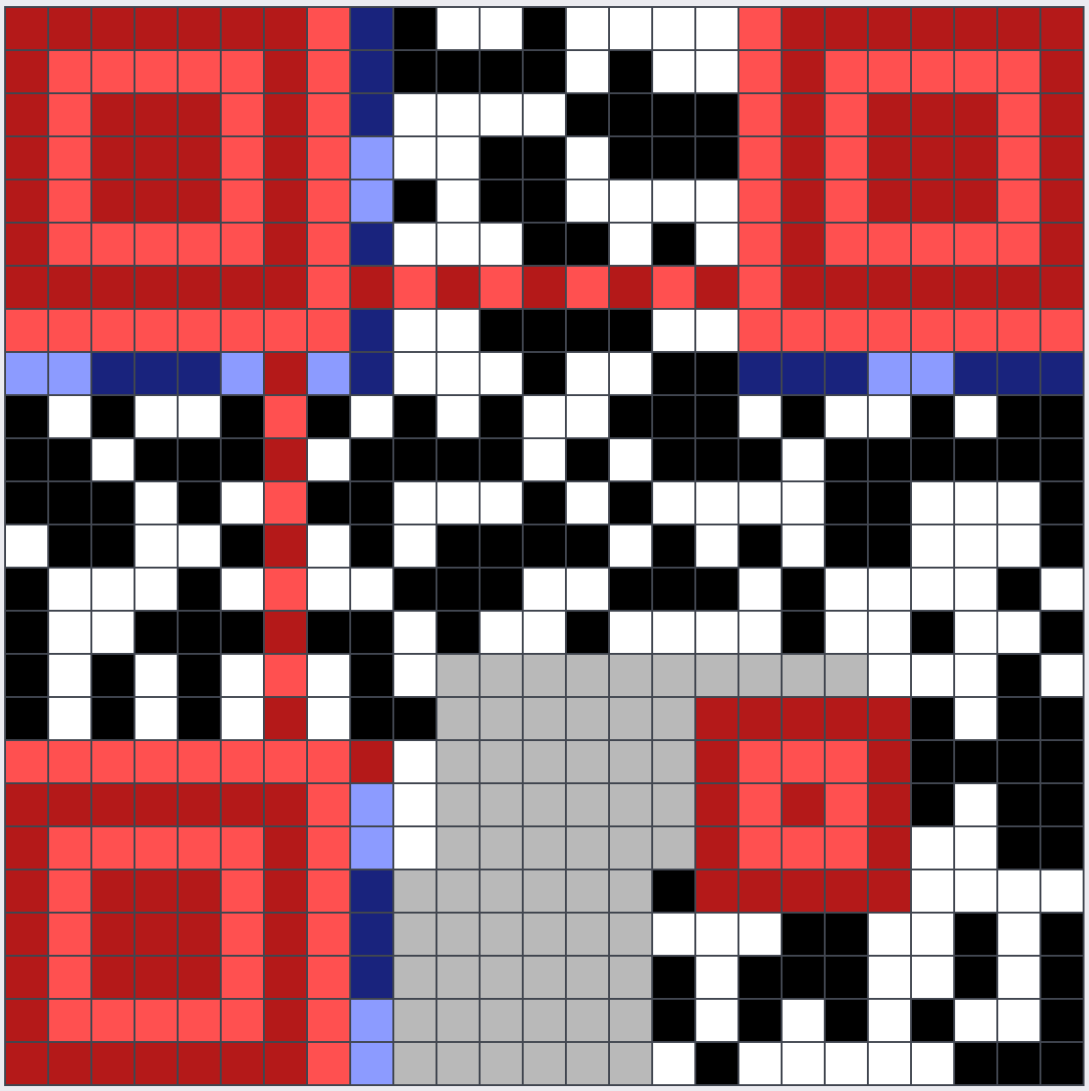
Top Left



Error Correction Level: L M Q H

Mask Pattern : 0 1 2 3 4 5 6 7

Save Cancel



最后识别出来是这个

Number of missing bytes (erasures) : **13 bytes (29.55%)**

Data blocks :

**["01000000","10010101","01110011","00000101","00100101","01000100","10000101","11110011","0001010
000011","11110000","11101100","00??0001","11101100","00010???","???
01100","00110101","00100011","01001010","01011100","00111001","00110011","001?????","????????","????
10110","10011110","00100010","11001000","01001010","00101010","01001111","011?1?0?","1?1?
1???","???????1","01011001","01110011","01101100","11101011","11100101","11100110","01010110"]**

----------Block 1----------

Reed-Solomon Block :

[64,149,115,5,37,68,133,243,21,0,240,236,0,236,0,0,53,35,74,92,57,51,0,0,0,0,0,0,158,34,200,74,42,79,0,0,0,8

Syndrome :

[248,78,202,152,1,186,3,61,188,223,218,220,70,159,184,69,172,21,28,10,114,203,189,108,208,252,215,114]

Number of Errors : **13**

Coefficient of the error location polynomial : **[99,200,88,0,165,101,223,101,24,120,210,110,4]**

Error Position : **[7,8,9,16,17,18,19,20,21,28,29,31,34]**

Error Magnitude : **1110101 11101101 1111001 110110 10111100 11101 11001101 1011011 10110111 11101100 10001 10001 1000011**

Final data bits :

0100000010010101011100110000010100100101010001001000010111110011000101010100001111110000111

[0100] [00001001] [01010111001100000101001001010100010010000101111100110001010101000011111 1000011]

Mode Indicator : **8-bit Mode (0100)**

Character Count Indicator : **9**

Decoded data : **W0RTH_1T?**

Final Decoded string : **W0RTH_1T?**

hgame{W0RTH_1T?}

2.Vidar Token

F12打开发现connect wallet按钮并没有甚么用，

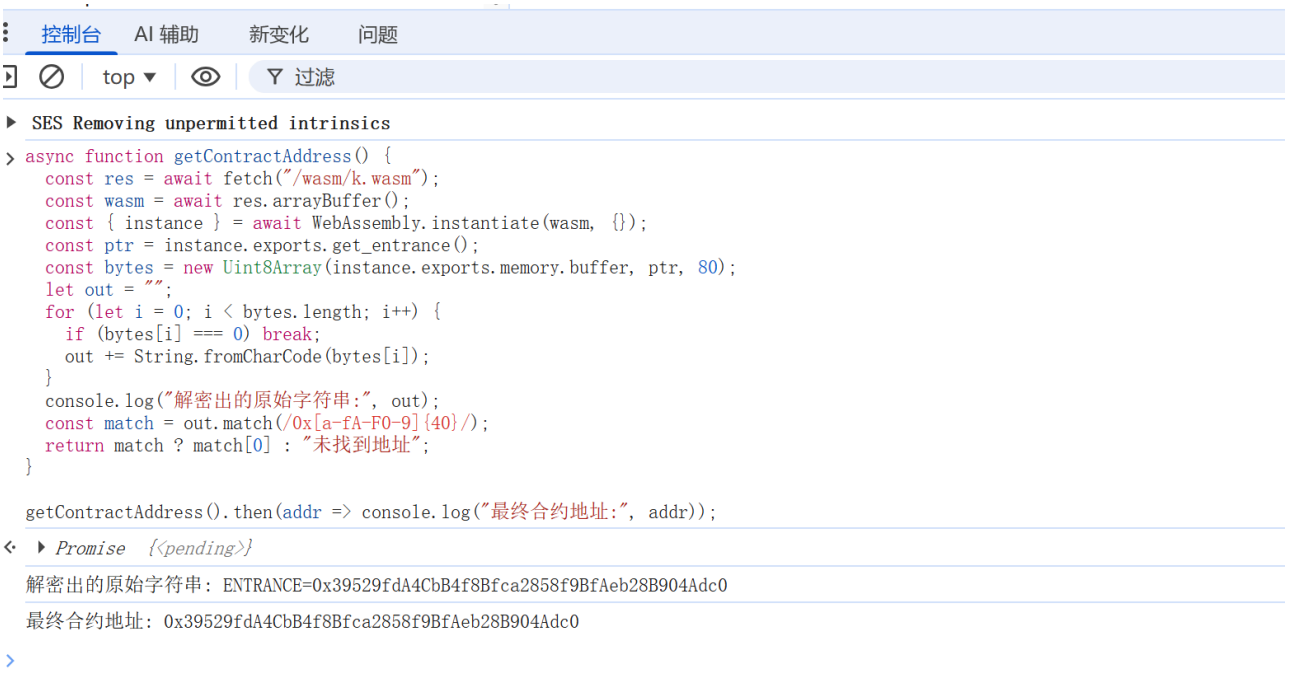
```
async function checkEligibility() {
  vaultStatusEl.textContent = "尝试读取元数据... (。•̀ᴗ-)✧";
  try {
    if (!entranceAddress) {
      const res = await fetch("/wasm/k.wasm", { method: "GET" });
      if (res.ok) {
        const wasm = await res.arrayBuffer();
        const { instance } = await WebAssembly.instantiate(wasm, {});
        const ptr = instance.exports.get_entrance();
        const text = readCString(instance.exports.memory, ptr, 80);
        const match = text.match(/ENTRANCE=(0x[a-fA-F0-9]{40})/);
        entranceAddress = match ? match[1] : "";
      }
    }
```

通过加载k.wasm，调用instance.exports.get_entrance()这个函数获取一下合约地址

```
async function getContractAddress() {

  const res = await fetch("/wasm/k.wasm");

  const wasm = await res.arrayBuffer();

  const { instance } = await WebAssembly.instantiate(wasm, {});

  const ptr = instance.exports.get_entrance();

  const bytes = new Uint8Array(instance.exports.memory.buffer, ptr,
80);

  let out = "";

  for (let i = 0; i < bytes.length; i++) {
    if (bytes[i] === 0) break;

    out += String.fromCharCode(bytes[i]);

  }
  console.log("解密出的原始字符串:", out);

  const match = out.match(/0x[a-fA-F0-9]{40}/);
```

```
  return match ? match[0] : "未找到地址";

}

getContractAddress().then(addr => console.log("最终合约地址:",
addr));
```

得到合约地址：0x39529fdA4CbB4f8Bfca2858f9BfAeb28B904Adc0



找一下vidar_coin地址

```
(async () => {

    const provider = new
ethers.JsonRpcProvider(window.location.origin + "/rpc");

    const vault = new
ethers.Contract("0x39529fdA4CbB4f8Bfca2858f9BfAeb28B904Adc0",
["function tokenURI(uint256) view returns (string)"], provider);

    const uri = await vault.tokenURI(0);
```

```
    const data = JSON.parse(atob(uri.split(",")[1]));

    console.log("NFT 元数据:", data);

    console.log("VidarCoin 地址:", data.vidar_coin);

})();
```

VidarCoin 地址: 0xc5273abfb36550090095b1edec019216ad21be6c

查询一下代币信息

```
(async () => {
    const provider = new
ethers.JsonRpcProvider(window.location.origin + "/rpc");

    const coinAddress =
"0xc5273abfb36550090095b1edec019216ad21be6c";

    const abi = [
        "function name() view returns (string)",
        "function symbol() view returns (string)"
    ];

    const coin = new ethers.Contract(coinAddress, abi, provider);

    const name = await coin.name();
    const symbol = await coin.symbol();

    console.log("代币名称 (Name):", name);
    console.log("代币符号 (Symbol):", symbol);

})();
```

代币符号 (Symbol):
0x6960606a647c742a413152683052756230 7e2c6c4f48562a32756258763372705e353335373e366632657c
这个symbol挺长，正常代币符号应该是3-6位大写符号，所以这个长串大概是密文。

```
(async () => {
    const provider = new ethers.JsonRpcProvider(window.location.origin + "/rpc");

    const coinAddress = "0xc5273abfb36550090095b1edec019216ad21be6c";

    const abi = [
        "function name() view returns (string)",
        "function symbol() view returns (string)"
    ];

    const coin = new ethers.Contract(coinAddress, abi, provider);

    const name = await coin.name();
    const symbol = await coin.symbol();

    console.log("代币名称 (Name):", name);
    console.log("代币符号 (Symbol):", symbol);

})();
```

```
▸ Promise {<pending>}
代币名称 (Name): VidarCoin
代币符号 (Symbol): 0x6960606a647c742a413152683052756230 7e2c6c4f48562a32756258763372705e353335373e366632657c
```

本来好像是应该是逆向wasm文件的，但是实在搞不出来，只能试试根据flag格式猜了（目移

```
(async () => {
    const hex =
"6960606a647c742a413152683052756230 7e2c6c4f48562a32756258763372705e
353335373e366632657c";
    const bytes = hex.match(/.{1,2}/g).map(byte => parseInt(byte,
16));
    const key = [0x01, 0x07];

    const flag = bytes.map((b, i) => {
        let charCode = b ^ key[i % key.length];
        return String.fromCharCode(charCode);
    }).join('');

    console.log("Flag:", flag);
})();
```

Flag: hgame{u-@6So1Ute1y-kNOW-3rc_w4sw_222697a3b}

```
> (async () => {
    const hex = "6960606a647c742a4131526830527562307e2c6c4f48562a3275625876337270 5e353335373e366632657c";
    const bytes = hex.match(/.{1,2}/g).map(byte => parseInt(byte, 16));
    const key = [0x01, 0x07];

    const flag = bytes.map((b, i) => {
        let charCode = b ^ key[i % key.length];
        return String.fromCharCode(charCode);
    }).join('');

    console.log("Flag:", flag);
})();
Flag: hgame{u-@6SolUte1y-kNOW-3rc_w4sw_222697a3b}
```

但输入不对（汗流浃背jpg），由于很相近所以只能改词（目移

hgame{u_Abso1ute1y_kNOW_3rc_w4sm_222697a3b}