

---

# Online Jukebox

DOS Attack using Selenium  
WebDriver + Python Web  
Server + Cron Job

---



---

Presented by  
Sai Harshinee Roopakula  
19577

# Table of Contents

- Introduction
  - Selenium
- Problem Statement
- Environment Setup
- Design - Code
  - Online Jukebox
  - DOS attack using Selenium JUnit/WebDriver
- Implementation
- Conclusion
- Bibliography

# Introduction - Selenium

Selenium is an open-source set of tools for testing web applications to operate across different browsers and operating systems.

## **Selenium Integrated Development Environment (IDE)**

A Firefox plugin that lets testers to record their actions as they follow the workflow that they need to test.

## **Selenium Remote Control (RC)**

The flagship testing framework that allowed more than simple browser actions and linear execution.

It makes use of the full power of programming languages such as Java, C#, PHP, Python, Ruby and PERL to create more complex tests.

## **Selenium WebDriver**

The successor to Selenium RC which sends commands directly to the browser and retrieves results.

## **Selenium Grid**

A tool used to run parallel tests across different machines and different browsers simultaneously which results in minimized execution time.

## **Advantages of Selenium**

- Selenium is an open-source tool.
- Can be extended for various technologies that expose DOM.
- Has capabilities to execute scripts across different browsers.
- Can execute scripts on various operating systems.
- Supports mobile devices.
- Executes tests within the browser, so focus is NOT required while script execution is in progress.
- Can execute tests in parallel with the use of Selenium Grids.

# Problem Statement

## Project Description:

Online Jukebox + DOS attack using Selenium JUnit/WebDriver + Python Web Server + Cron Job

## Process:

- Continue the previous Online Jukebox + DOS attack using Selenium IDE + Python Web Server task
- Replace the "DOS attack using Selenium IDE" with "DOS attack using Selenium WebDriver".
- The DOS attack should be done on a command line using Cron Job

# Environment Setup

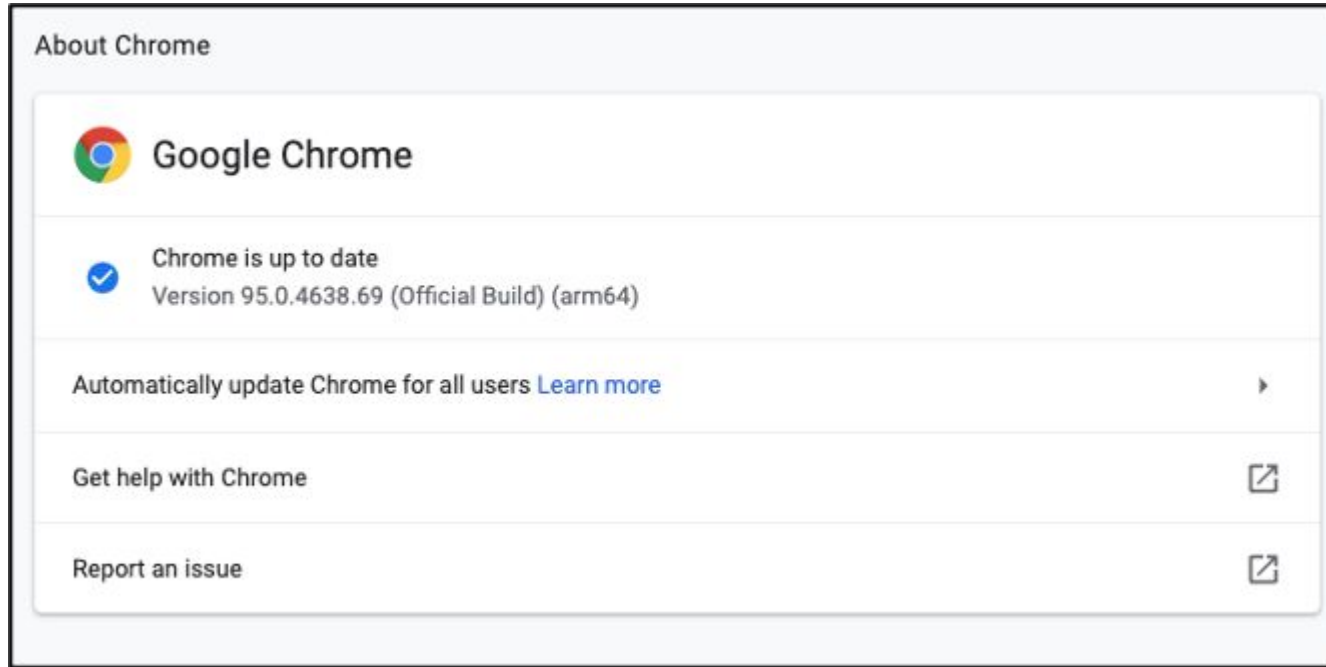
## Step 1: Install selenium

Command: pip3 install selenium

```
Hemanths-MacBook-Pro:~ hemanthharshinee$ pip3 install selenium
DEPRECATION: Configuring installation scheme with distutils config files is deprecated and will be removed in a future version. To help transition to the PEP 517 based system, please see this issue: https://github.com/pypa/pip/issues/7558
Collecting selenium
  Downloading selenium-4.0.0-py3-none-any.whl (954 kB)
    |████████████████████████████████████████| 954 kB 3.4 MB/s
Collecting urllib3[secure]>=1.26
  Downloading urllib3-1.26.7-py2.py3-none-any.whl (138 kB)
    |████████████████████████████████████████| 138 kB 8.9 MB/s
Collecting trio-websocket>=0.9
  Downloading trio_websocket-0.9.2-py3-none-any.whl (16 kB)
Collecting trio>=0.17
  Downloading trio-0.19.0-py3-none-any.whl (356 kB)
    |████████████████████████████████████████| 356 kB 4.9 MB/s
Collecting idna
  Downloading idna-3.3-py3-none-any.whl (61 kB)
    |████████████████████████████████████████| 61 kB 2.2 MB/s
Collecting sortedcontainers
  Downloading sortedcontainers-2.4.0-py2.py3-none-any.whl (29 kB)
Collecting attrs>=19.2.0
  Downloading attrs-21.2.0-py2.py3-none-any.whl (53 kB)
    |████████████████████████████████████████| 53 kB 5.7 MB/s
Collecting async-generator>=1.9
  Using cached async_generator-1.10-py3-none-any.whl (18 kB)
Collecting sniffio
  Using cached sniffio-1.2.0-py3-none-any.whl (10 kB)
```

# Environment Setup

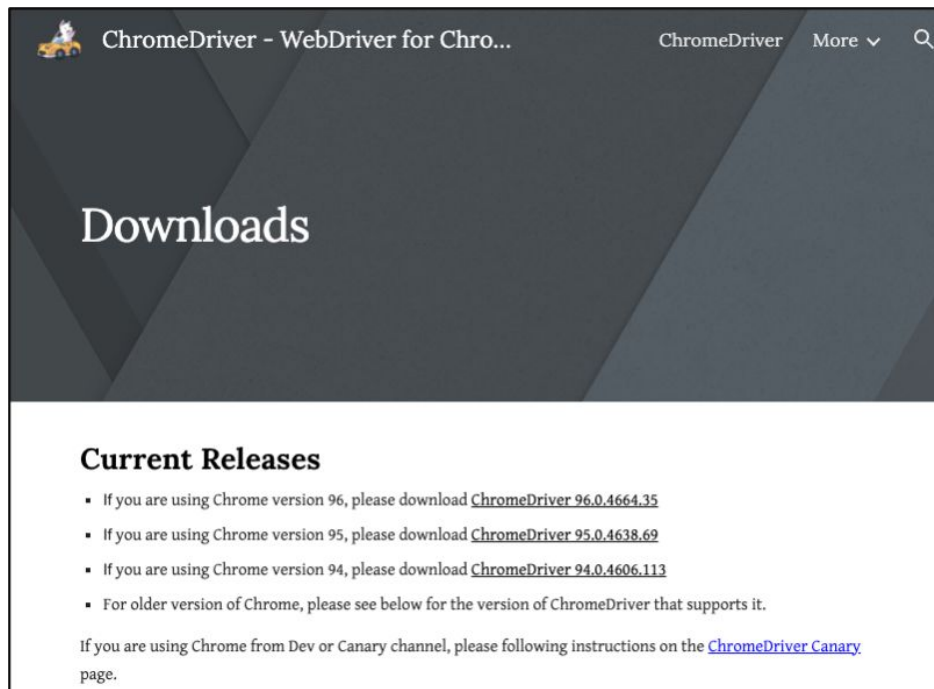
**Step 2:** Check the Google Chrome version - Click on 3 dots of the right side of the webpage and click on Help



# Environment Setup

## Step 3: Download the chrome driver







Go the website <https://chromedriver.chromium.org/downloads> and download the chrome driver corresponding to the Chrome version above.



# Environment Setup

Download the zip file according to the OS you use. Chrome driver will be installed. Place it in the project folder you are working.

## Index of /95.0.4638.69/

	<u>Name</u>	Last modified	Size	ETag
	<a href="#">Parent Directory</a>		-	
	<a href="#">chromedriver linux64.zip</a>	2021-11-08 09:31:24	9.52MB	a1fdedf1bdec0e140458bef35006cdc4
	<a href="#">chromedriver mac64.zip</a>	2021-11-08 09:31:26	7.81MB	f97559e10cbe2bd4ef29a019958af514
	<a href="#">chromedriver mac64 ml.zip</a>	2021-11-08 09:31:28	7.39MB	5d7f30a75623bc63d2d3705928a82102
	<a href="#">chromedriver win32.zip</a>	2021-11-08 09:31:30	5.74MB	82a40345336b35b9c3f1779f5089cfd3
	<a href="#">notes.txt</a>	2021-11-08 09:31:35	0.00MB	0fe40272d622c011b7523dcca7a7397



# Design - Code - Online Jukebox

```
Welcome Guide SeleniumWebdriveDOS.py jukebox_without_reCAPTCHA.html
<html>
  <head>
    <title>JukeBox Form without reCAPTCHA</title>
    <h1>
      Online JukeBox
    </h1>
  </head>
  <body>
    <form name="select" action="/cgi-bin/without_reCAPTCHA.py" method="get">
      <h4>
        Select a Song
      </h4>
      <input type="radio" name="song" value="Song_1">Song 1</a>
      <br>
      <input type="radio" name="song" value="Song_2">Song 2</a>
      <br>
      <input type="radio" name="song" value="Song_3">Song 3</a>
      <br>
      <input type="radio" name="song" value="Song_4">Song 4</a>

      <h4>
        Credit Card Number
      <br><input type="text" name="cc_number">
      <h4>
      <br>
      <input type="submit" value="Submit">
    </form>
  </body>
</html>
```

# Design - Online Jukebox

← → ↻ ⓘ File | /Users/hemanthharshinee/html/jukebox\_without\_reCAPTCHA.html

📁 Apps 📺 YouTube 📁 Learning 📁 AI 📁 Projects 📁 MSCS 📁 Web Dev 📁 Typescript

## Online JukeBox

**Select a Song**

☐ Song 1  
☐ Song 2  
☐ Song 3  
☐ Song 4

**Credit Card Number**

# Design - Code - without\_reCAPTCHA.py

```
Welcome Guide SeleniumWebdriveDOS.py jukebox_without_reCAPTCHA.py
#!/Users/hemanttharshinee/opt/anaconda3/bin/python3

import cgi, cgitb
import pymysql.cursors

cgitb.enable()
form = cgi.FieldStorage()

song_placeholder = form.getvalue('song')
cc_number_placeholder = form.getvalue('cc_number')

print('Content-type:text/html\r\n\r\n')
print('<html>')
print('<head>')
print('<title>JukeBox Transaction</title>')
print('</head>')
print('<body>')
print('Song you select: %s' % (song_placeholder))
print('<br>')
print('Credit card Number is: %s' % (cc_number_placeholder))
print('</body>')
print('</html>')

connection = pymysql.connect(host='localhost',
                             user='root',
                             password='Redhawk@205',
                             database='JUKEBOX',
                             cursorclass=pymysql.cursors.DictCursor)

with connection:
    with connection.cursor() as cursor:
```

# Design - Code - SeleniumWebdriveDOS.py

```
import unittest
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.common.by import By

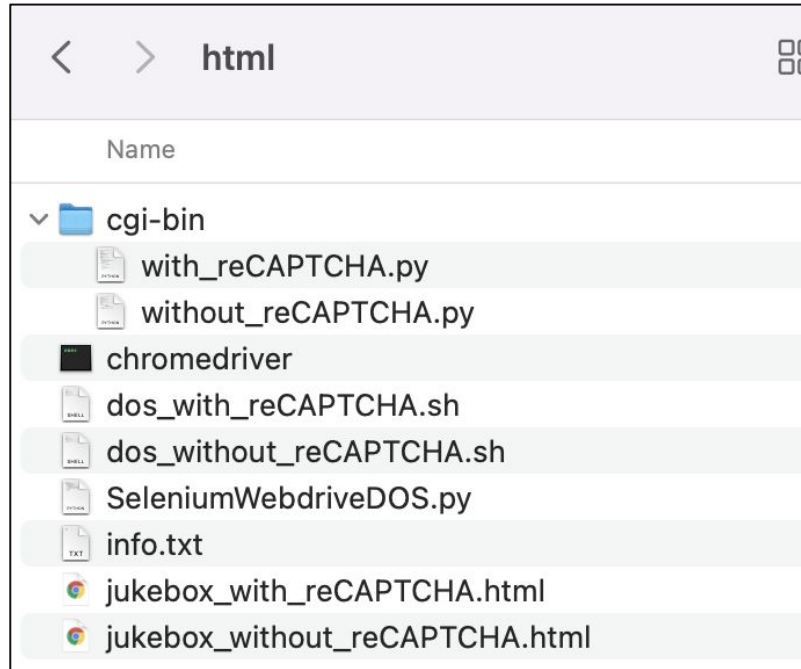
class ChromeSearch(unittest.TestCase):
    def setUp(self):
        s = Service('/Users/hemanthharshinee/Downloads/chromedriver')
        self.driver = webdriver.Chrome(service=s)

    def test_search_in_python_org(self):
        for i in range(10): #Simulate a DOS attack to run 10 times
            driver = self.driver
            driver.get("http://localhost:8000/jukebox_without_reCAPTCHA.html") #the website to be attacked
            driver.find_element(By.XPATH, '/html/body/form/input[3]').click()
            driver.find_element(By.XPATH, '/html/body/form/h4[2]/input').send_keys("ATTACK")
            driver.find_element(By.XPATH, "/html/body/form/h4[3]/input").submit()
            driver.back()
            i+=1
        print("Simulated DOS attack has been completed...")

    def tearDown(self):
        self.driver.close()

if __name__ == "__main__":
    unittest.main()
```

# Design - Code - All the required files.



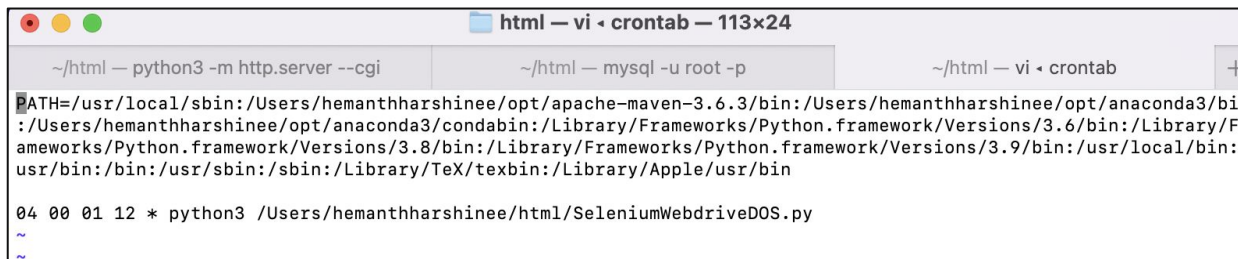
# Implementation

**Make sure the Python Web server is running**

Command: `python3 -m http.server --cgi`

~/html — python3 -m http.server --cgi	~/html — mysql -u root -p
<pre>Last login: Tue Nov 30 23:55:53 on ttys000 You have new mail. (base) hemanthharshinee@Hemanths-MacBook-Pro ~ % cd html (base) hemanthharshinee@Hemanths-MacBook-Pro html % python3 -m http.server --cgi Serving HTTP on :: port 8000 (http://[::]:8000/) ...</pre>	

**Schedule a Cron job to perform DOS Attack**



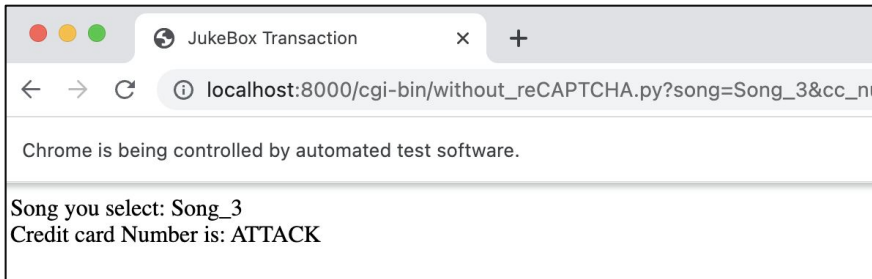
The screenshot shows a terminal window titled "html — vi • crontab — 113x24". The terminal displays the contents of the crontab file, which includes the PATH environment variable and a cron job entry. The PATH variable is set to include the system bin directory, the user's local bin directory, and the Anaconda3 bin directory. The cron job entry is scheduled to run every 4 minutes at 00:00:04, executing the command `python3 /Users/hemanthharshinee/html/SeleniumWebdriveDOS.py`.

```
PATH=/usr/local/sbin:/Users/hemanthharshinee/opt/apache-maven-3.6.3/bin:/Users/hemanthharshinee/opt/anaconda3/bin:/Users/hemanthharshinee/opt/anaconda3/condabin:/Library/Frameworks/Python.framework/Versions/3.6/bin:/Library/Frameworks/Python.framework/Versions/3.8/bin:/Library/Frameworks/Python.framework/Versions/3.9/bin:/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin:/Library/TeX/texbin:/Library/Apple/usr/bin

04 00 01 12 * python3 /Users/hemanthharshinee/html/SeleniumWebdriveDOS.py
```

# Implementation

## During Attack



## Result on Command Line

```
(base) hemanthharshinee@Hemanths-MacBook-Pro ~ % cd html
(base) hemanthharshinee@Hemanths-MacBook-Pro html % python3 -m http.server --cgi
Serving HTTP on :: port 8000 (http://[::]:8000/) ...
::1 - - [30/Nov/2021 23:58:39] "GET / HTTP/1.1" 200 -
::1 - - [30/Nov/2021 23:58:40] code 403, message CGI script is not a plain file ('/cgi-bin/')
::1 - - [30/Nov/2021 23:58:40] "GET /cgi-bin/ HTTP/1.1" 403 -
::1 - - [30/Nov/2021 23:58:56] "GET /cgi-bin/without_reCAPTCHA.py?song=Song_3&cc_number=1212 HTTP/1.1" 200 -
::1 - - [30/Nov/2021 23:59:06] "GET /cgi-bin/without_reCAPTCHA.py?song=Song_1&cc_number=10000000 HTTP/1.1" 200 -
::1 - - [01/Dec/2021 00:04:07] "GET /jukebox_without_reCAPTCHA.html HTTP/1.1" 200 -
::1 - - [01/Dec/2021 00:04:08] code 404, message File not found
::1 - - [01/Dec/2021 00:04:08] "GET /favicon.ico HTTP/1.1" 404 -
::1 - - [01/Dec/2021 00:04:09] "GET /cgi-bin/without_reCAPTCHA.py?song=Song_3&cc_number=ATTACK HTTP/1.1" 200 -
::1 - - [01/Dec/2021 00:04:09] "GET /jukebox_without_reCAPTCHA.html HTTP/1.1" 304 -
::1 - - [01/Dec/2021 00:04:09] "GET /cgi-bin/without_reCAPTCHA.py?song=Song_3&cc_number=ATTACK HTTP/1.1" 200 -
::1 - - [01/Dec/2021 00:04:10] "GET /jukebox_without_reCAPTCHA.html HTTP/1.1" 304 -
::1 - - [01/Dec/2021 00:04:10] "GET /cgi-bin/without_reCAPTCHA.py?song=Song_3&cc_number=ATTACK HTTP/1.1" 200 -
::1 - - [01/Dec/2021 00:04:10] "GET /jukebox_without_reCAPTCHA.html HTTP/1.1" 304 -
::1 - - [01/Dec/2021 00:04:10] "GET /cgi-bin/without_reCAPTCHA.py?song=Song_3&cc_number=ATTACK HTTP/1.1" 200 -
::1 - - [01/Dec/2021 00:04:11] "GET /jukebox_without_reCAPTCHA.html HTTP/1.1" 304 -
::1 - - [01/Dec/2021 00:04:11] "GET /cgi-bin/without_reCAPTCHA.py?song=Song_3&cc_number=ATTACK HTTP/1.1" 200 -
::1 - - [01/Dec/2021 00:04:11] "GET /jukebox_without_reCAPTCHA.html HTTP/1.1" 304 -
::1 - - [01/Dec/2021 00:04:12] "GET /cgi-bin/without_reCAPTCHA.py?song=Song_3&cc_number=ATTACK HTTP/1.1" 200 -
::1 - - [01/Dec/2021 00:04:12] "GET /jukebox_without_reCAPTCHA.html HTTP/1.1" 304 -
::1 - - [01/Dec/2021 00:04:12] "GET /cgi-bin/without_reCAPTCHA.py?song=Song_3&cc_number=ATTACK HTTP/1.1" 200 -
::1 - - [01/Dec/2021 00:04:12] "GET /jukebox_without_reCAPTCHA.html HTTP/1.1" 304 -
::1 - - [01/Dec/2021 00:04:13] "GET /cgi-bin/without_reCAPTCHA.py?song=Song_3&cc_number=ATTACK HTTP/1.1" 200 -
::1 - - [01/Dec/2021 00:04:13] "GET /jukebox_without_reCAPTCHA.html HTTP/1.1" 304 -
::1 - - [01/Dec/2021 00:04:13] "GET /cgi-bin/without_reCAPTCHA.py?song=Song_3&cc_number=ATTACK HTTP/1.1" 200 -
::1 - - [01/Dec/2021 00:04:14] "GET /jukebox_without_reCAPTCHA.html HTTP/1.1" 304 -
::1 - - [01/Dec/2021 00:04:14] "GET /cgi-bin/without_reCAPTCHA.py?song=Song_3&cc_number=ATTACK HTTP/1.1" 200 -
::1 - - [01/Dec/2021 00:14:07] "GET /jukebox_without_reCAPTCHA.html HTTP/1.1" 200 -
```

# Implementation

## Result in Crontab

```
Received: by Hemanths-MacBook-Pro.local (Postfix, from userid 501)
id E76F1594CAA8; Wed, 1 Dec 2021 00:04:14 -0800 (PST)
From: hemanthharshinee@Hemanths-MacBook-Pro.local (Cron Daemon)
To: hemanthharshinee@Hemanths-MacBook-Pro.local
Subject: Cron <hemanthharshinee@Hemanths-MacBook-Pro> python3 /Users/hemanthharshinee/html/
SeleniumWebdriveDOS.py
X-Cron-Env: <PATH=/usr/local/sbin:/Users/hemanthharshinee/opt/apache-maven-3.6.3/bin:/Users/
hemanthharshinee/opt/anaconda3/bin:/Users/hemanthharshinee/opt/anaconda3/condabin:/Library/
Frameworks/Python.framework/Versions/3.6/bin:/Library/Frameworks/Python.framework/Versions/3.8/bin:/
Library/Frameworks/Python.framework/Versions/3.9/bin:/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin:/
Library/TeX/texbin:/Library/Apple/usr/bin>
X-Cron-Env: <SHELL=/bin/sh>
X-Cron-Env: <LOGNAME=hemanthharshinee>
X-Cron-Env: <USER=hemanthharshinee>
Message-Id: <20211201080414.E76F1594CAA8@Hemanths-MacBook-Pro.local>
Date: Wed, 1 Dec 2021 00:04:14 -0800 (PST)

./Users/hemanthharshinee/opt/anaconda3/lib/python3.8/unittest/suite.py:84: ResourceWarning: unclosed
<socket.socket fd=7, family=AddressFamily.AF_INET6, type=SocketKind.SOCK_STREAM, proto=6,
laddr=('::1', 49441, 0, 0), raddr=('::1', 49437, 0, 0)>
return self.run(*args, **kwargs)
ResourceWarning: Enable tracemalloc to get the object allocation traceback

-----
Ran 1 test in 13.440s

OK
Simulated DOS attack has been completed...
```



# Implementation

## Database before and after attack

```
MySQL [jukebox]> select * from transaction;
```

song	cc_number
Song_1	121212
Song_1	121212
Song_1	121212
Song_1	121212
Song_1	121212
Song_1	121212
Song_1	121212
Song_1	121212
Song_1	121212
Song_1	121212
Song_1	1111
Song_1	1111
Song_1	1111
Song_1	1111
Song_1	1111
Song_1	1111
Song_1	1111
Song_1	1111
Song_1	1111
Song_1	1111
Song_1	1111
Song_1	1111
Song_1	1111
Song_1	1111
Song_1	121212
Song_1	121212
Song_1	121212
Song_1	121212
Song_1	121212
Song_3	1212
Song_1	10000000

27 rows in set (0.000 sec)

```
MySQL [jukebox]> select * from transaction;
```

song	cc_number
Song_1	121212
Song_1	121212
Song_1	121212
Song_1	121212
Song_1	121212
Song_1	121212
Song_1	121212
Song_1	121212
Song_1	1111
Song_1	1111
Song_1	1111
Song_1	1111
Song_1	1111
Song_1	1111
Song_1	1111
Song_1	1111
Song_1	1111
Song_1	1111
Song_1	1111
Song_1	121212
Song_1	121212
Song_1	121212
Song_1	121212
Song_3	1212
Song_1	10000000
Song_3	ATTACK
Song_3	ATTACK
Song_3	ATTACK
Song_3	ATTACK
Song_3	ATTACK
Song_3	ATTACK
Song_3	ATTACK
Song_3	ATTACK
Song_3	ATTACK
Song_3	ATTACK
Song_3	ATTACK

37 rows in set (0.001 sec)

# Conclusion

DOS Attack was successfully done using Selenium Web driver using Cronjob.

# Bibliography

[https://npu85.npu.edu/~henry/npu/classes/qa/selenium\\_tutorialspoint/slide/exercise\\_tutorialspoint.html](https://npu85.npu.edu/~henry/npu/classes/qa/selenium_tutorialspoint/slide/exercise_tutorialspoint.html)

Link to view the presentation

[https://docs.google.com/presentation/d/1\\_LvN1M7nzJBqrw2A1nvCU7vuxfIP6GI6JDoB3KDSZP0/edit?usp=sharing](https://docs.google.com/presentation/d/1_LvN1M7nzJBqrw2A1nvCU7vuxfIP6GI6JDoB3KDSZP0/edit?usp=sharing)