|  |  |
| --- | --- |
| **Practicum Case** |  |
| COMP6548  Programming for Penetration Testing |
| **Cyber Security** | **<Case Code>** |
| ***Valid on*** *[Odd/Even/Compact] Semester Year 9999/9999* | **Revision 00** |

## Learning Outcomes

* Socket Network Program for Penetration Testing
* Web Vulnerability with Programming
* Additional tools for Penetration Testing

## Topic

* Web Recon and Enumeration

## Subtopics

* Slow Loris DoS Attack
* HTTP Request and Response using Python.
* Brute-Forcing Directory and File Location.
* Generate and Read Directory & Filename Wordlist with Python.

## Soal

*Case*

**DoSer and DirBuster**

Miracle Foundation, one of the charity organization wants you to test their website, there are two things that you need to do, they are:

* Slow Loris Denial of Service attack.
* Directory Listing.

There are some prerequisites that you need to complete, they are:

* You already have XAMPP installed in your PC.
* You need to have python programming language with **socket** and **requests** **library** installed.

You need to make 2 programs using python to complete these tasks, here are some steps that you need to follow:

1. There is a folder called 08-web in the zip, you need to copy it to your **Document Root** folder (the default folder is in “C:/xampp/htdocs”).
2. Open XAMPP Control Panel.
3. Press “Start” button on a module called “Apache”.
4. Try opening the website on your browser.

Graphical user interface, text, application

Description automatically generated

**Figure 1. The Website**

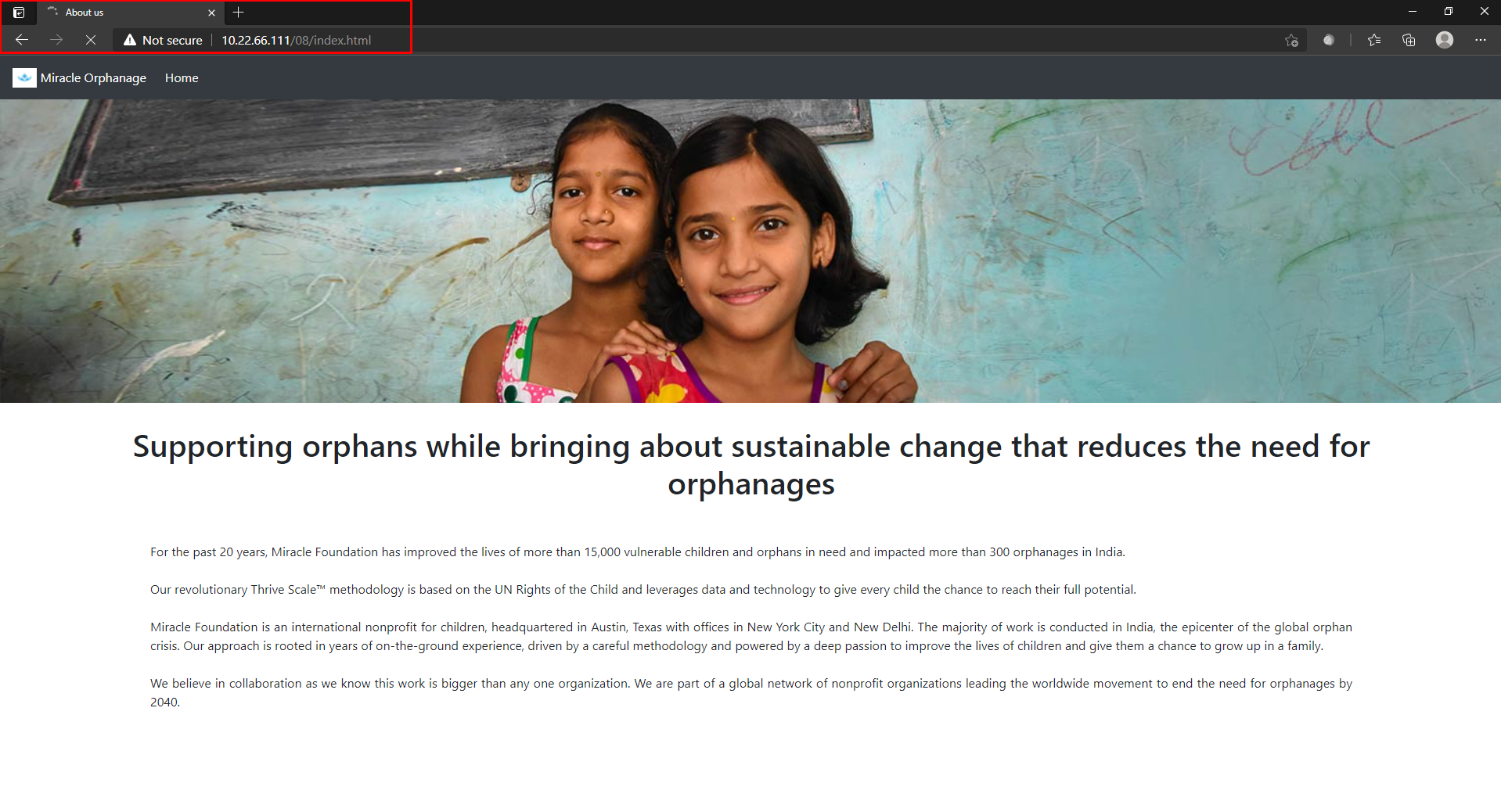
1. Try the **Slow Loris** **attack** with **python socket library**. You need to make a program that can make the website incapable of refreshing the page and the Network Usage is pretty high.



**Figure 2. XAMPP Process in Task Manager before Slow Loris Attack**

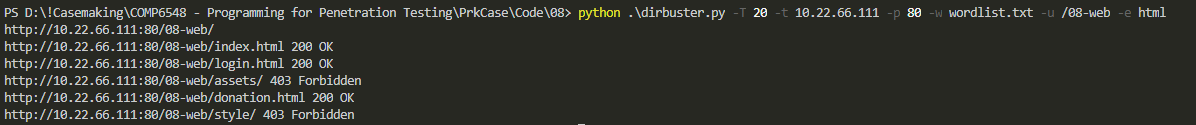


**Figure 3. XAMPP Process in Task Manager after Slow Loris Attack**



**Figure 4. The Website Incapable of Refreshing the Page**

1. Make a python program to brute force the directory and file inside the web app based on the **HTTP Status Code**.



**Figure 5. Dirbuster Result**