|  |  |
| --- | --- |
| **Assignment Case** | Diagram  Description automatically generated |
| COMP6548001  Programming for Penetration Testing |
| **Cyber Security** | **O223-COMP6548-CH02-02** |
| ***Valid on*** *Odd Semester Year 2021/2022* | **Revision 00** |

1. Seluruh mahasiswa tidak diperkenankan untuk:

*All students are not allowed to:*

* + - Berdiskusi dan/atau bekerja sama dengan mahasiswa lainnya

*Discuss and/or work together with other student participants*

* + - Melihat sebagian atau seluruh jawaban mahasiswa lain

*Seeing a part or the whole answer from another student*

* + - Membuka dan menyalin dari **BUKU** atau **CATATAN**, **VIDEO** dari pengajar (recording kelas, VBL, Youtube, dsb) dan **REFERENSI** lainnya

*Open and copy from any resources such as notes, videos (class recording, VBL, Youtube, etc) and other references*

* + - Membuka dan menyalin jawaban dari internet (google, stackoverflow, dsb)

*Open and copy answer from the internet (google, stackoverflow, etc)*

* + - Mengerjakan soal yang tidak sesuai dengan tema yang ada di soal,

*Working with another theme which is not in accordance with the existing theme in the matter of the case,*

* + - Melakukan tindakan kecurangan lainnya,

*Committing other dishonest actions,*

* + - Secara sengaja maupun tidak sengaja melakukan segala tindakan kelalaian yang menyebabkan hasil karyanya berhasil dicontek oleh orang lain / kelompok lain.

*Accidentally or intentionally conduct any failure action that cause the results of the project was copied by someone else / other groups.*

1. Jika mahasiswa terbukti melakukan tindakan seperti yang dijelaskan butir 1 di atas, maka **nilai mahasiswa** yang melakukan kecurangan (menyontek maupun dicontek) akan di – **NOL** – kan.

*If the student is proved to the actions described in point 1 above, the score of the student which committed dishonest acts (cheating or being cheated) will be “Zero”*

1. Perhatikan jadwal pengumpulan jawaban, segala jenis pengumpulan jawaban di luar jadwal tidak dilayani.

*Pay attention to the submission schedule, all kinds of submission outside the schedule will not be accepted*

1. Bila Anda tidak membaca peraturan ini, maka Anda dianggap telah membaca dan menyetujuinya

*If you have missed to read these regulations, so you are considered to have read and agreed on it*

1. Persentase penilaiaan untuk matakuliah ini adalah sebagai berikut:

*Marking percentage for this subject is described as follows:*

|  |
| --- |
| **Tugas Mandiri**  *Assignment* |
| 100% |

1. Software yang digunakan pada matakuliah ini adalah sebagai berikut:

*Software will be used in this subject are described as follows:*

|  |
| --- |
| **Software**  *Software* |
| VM Kali Linux 2019.1  VMware Workstation 16  Python 3  XAMPP |

## Ekstensi file yang harus disertakan dalam pengumpulan tugas mandiri untuk matakuliah ini adalah sebagai berikut:

*File extensions should be included in assignment collection for this subject are described as follows:*

|  |
| --- |
| **Tugas Mandiri**  *Assignment* |
| PY |

## Soal

*Case*

**speedyNinja**

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 **necessary** **libraries** installed (**Beautiful Soup 4** and **Python Requests**).

You need to make a program using python to complete this task, here are some steps that you need to follow:

1. There is a folder called **speedyNinja** 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” and “MySQL”.
4. Try opening the website on your browser.
5. Open PHPMyAdmin and create a database called “speedyNinja”
6. Backup the **create+insert.sql** inside **speedyNinja folder** into the database that you’ve created (by dragging and dropping or use “Import” menu).

For visitors, **speedyNinja** is nothing but a normal website to book a town delivery service. However, recently, there's an information leak that **speedyNinja** is actually providing an illegal service to hire a thief and some state officials had use this service to illegally obtain confidential documents. The local government want to find out all the parties that are involved in this transaction, so you are tasked to create a program using **Python Programming Language** (**must be using python 3**) in **Linux Environment** using **requests** and **BeautifulSoup** module to retrieve the information from **speedyNinja's** database, taking advantage of the website SQL Injection Vulnerability. There will be several features for the program, such as **dump current database name**, **tables, columns,** and **entry data from each table**. Below is the detailed description and specification of the program:

1. **Argument List**

|  |  |  |
| --- | --- | --- |
| Short Argument | Long Argument | Explanation |
| -u | --url | - Set website URL (e.g. localhost:80/speedyNinja) that will be used to extract information |
| -h | --help | - If the argument is **used**, the program will **display a help message** to show how to use the argument |
| Here’s the help message to show how to use the argument: | | |

1. **Program Flow**

* **Process number 1,** validate the URL argument given if it is appropriate based on the **response’s status code** when **requesting to the URL** with conditions below:
* Validate requested URL, show a message when the **URL is not found** **or valid** and exit the program

|  |
| --- |
| Example Command: |
| Result: |

* **Continue** to the next process when the requested URL is found
* **Process number 2**, bypass the authentication system using **SQL Injection Attack** to gain access to the website with conditions below:
* Show a **failed message** when it is **failed to get authentication** from the website and exit the program.

|  |
| --- |
|  |

* Show a **success message** when the program has **successfully getting the authentication** from the website. Show the **CSRF token value** that is obtained from the website and continue to the next process.

|  |
| --- |
|  |

* Process number 3, before you retrieve data from the database, **generate total column** for the **SQL Injection Attack**. Note that the result won’t be displayed, but **is needed** for the next process.
* Process number 4, **start retrieving data** **from the database**. Show the result of data obtained from the database with conditions below:
  + Show the **name of the database**.

|  |
| --- |
|  |

* + Show **all tables** in the database. This includes **name of the tables, name of all the columns in the tables, and the data type of each columns**.

|  |
| --- |
|  |

* + Note that this must also be done with **SQL Injection Attack**.
* **Process number 5**, **start** **retrieving data from the table** with conditions below:
  + Ask user to input the name of the table to retrieve the data from.

|  |
| --- |
|  |

* + Validate that table exist in database. Show a **failed message** if the table **doesn’t exist** in database.

|  |
| --- |
|  |

* + Show all the **entries data** in the related table if the table **exist** in database.

|  |
| --- |
|  |

* + User can type **“exit”** to quit the program.

|  |
| --- |
|  |

* Remember that you need to **focus on the functionality of the program** rather than the view of the program.
* Good luck.