**Tercer Parcial**

1. Demuestre y documente un ataque de Man In The Middle (MITM) para su Cliente - Servidor Multiaplicación.
   1. Esta demostración debe colocar un tercer elemento entre el Cliente y el Servidor, aunque este elemento (el MITM) puede estar alojado en el cliente o en el servidor según le convenga para su demostración.
   2. Debe al menos involucrar dos equipos de cómputo (uno donde reside el Cliente, otro donde reside el Servidor). En cualquiera de ellos reside el MITM.
   3. Debe documentarse la herramienta utilizada o desarrollada (según el caso) para implementar el MITM.
   4. Debe demostrarse el funcionamiento al profesor.
      1. Funcionamiento normal.
      2. Funcionamiento normal con MITM
      3. Funcionamiento alterando credenciales o información que fluye entre el Cliente y el Servidor en el MITM.
2. Dependiendo del esquema de MITM seleccionado, proponga 2 controles para evitarlo.
   1. Implemente al menos un control para que el esquema de MITM no sea tan fácil de implementar en su Cliente- Servidor Multiaplicación.
   2. Documente la prueba del control mostrando su efectividad.
3. Revise qué ocurrió en su equipo Servidor con alguna herramienta forense.
   1. Determinar mediante las evidencias el momento en que su aplicación funcionó de forma normal, comparándole con evidencias de un comportamiento en el que hay un MITM.

1ª) El MITM se coloco en el Cliente, en el siguiente diagrama se muestra la posición donde se alojó el MITM.

1b) No se involucro dos equipos debido a que todo el proyecto desde el comienzo se realizó usando solamente una maquina.

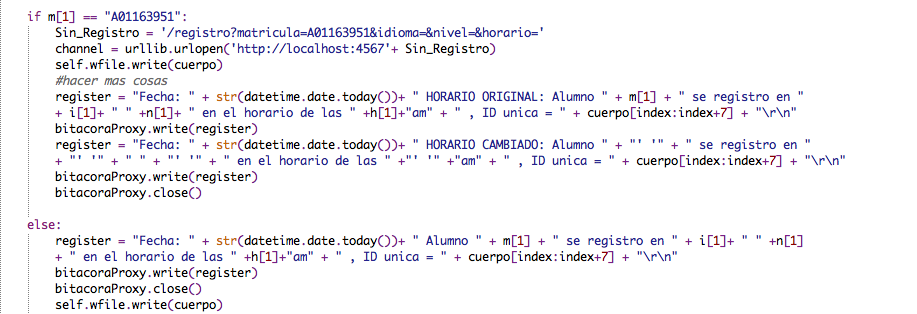
1c) La herramienta que se usó para el MITM fue desarrollada, básicamente se realizó un HTTP Proxy Server usando el lenguaje de programación Python. A continuación se muestra el código y una pequeña explicación.



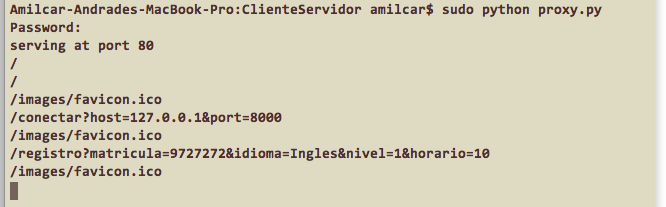
**Imagen 1: Código HTTP Proxy Server (MITM)**

Como se aprecia en el código anterior, primero se importan las librerías que se necesitan para echar andar el MITM, Por otro lado se declara una variable llamada *PORT*, esta variable se usara más adelante para que el Proxy use el puerto 80 para que escuche por los *Get Request* ya que este puerto es el default para el protocolo HTTP, después se crea una clase llamada *Proxy*, básicamente aquí comienza la creación del servidor, A continuación, se usa llama ala función *do\_GET(),* esta función escucha por todos los *Get Request* que genere el cliente (browser), se captura el *Query String* que se manda en el *Get Request* y este se guarda y se envía al servidor “bueno”.

Básicamente nuestro Proxy server escucha por *Get Request* que estén relacionados a la inscripción de nuevos estudiantes en cursos, básicamente monitorea a los alumnos que se están inscribiendo en los cursos y genera una bitácora. Por otro lado , nuestro servidor también puede alterar información, a continuación se muestra la otra parte de código donde se afectan las credenciales.

**Imagen 2: Código para alterar credenciales**

Aquí el MITM compara si la matricula que venia en el *Query String* es la misma que la matricula a la que le queremos hacer daño en este la matricula será la *A01163951,* el MITM toma la información original y la modificada y la guarda en la bitácora. Por otro lado, la información original la modifica por valores vacíos, ósea que el alumno con la matricula *A01163951* no se inscribirá a ningún curso. Por último a continuación se muestran algunas pruebas del HTTP Proxy Server.



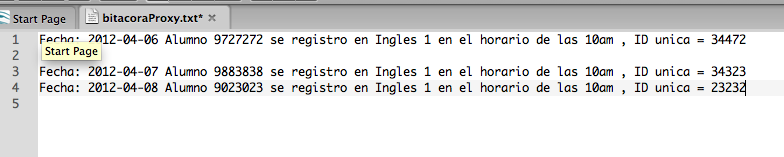
**Imagen 3: Prueba en la terminal del MITM corriendo.**



**Imagen 4: Servidor “Bueno”, funcionando. NOTA: tomar en cuenta la dirección en este caso está apuntando al LOCALHOST:4567/ … .Aquí es donde se encuentra operando el servidor bueno (Sinatra)**



**Imagen 5: Servidor Proxy “MITM”, funcionando. NOTA: tomar en cuenta la dirección en este caso está apuntando al LOCALHOST:80/ … .Aquí es donde se encuentra operando el MITM.**



**Imagen 6: Bitácora del MITM**

2. El primer control que podríamos hacer seria que los métodos cambiaran de GET a POST, con esto la información no seria tan fácil de obtener ya que ya la petición ya no vendría en el Query String, ahora vendría en el cuerpo del HTTP Request, en un campo especificó que no es tan fácil de obtener, suponiendo que los métodos que usáramos fueran POST, el MITM tendría que generar los HEADERS y el BODY del HTTP Request. Cabe tomar en cuenta que este control solamente haría más difícil que el MITM funcionara, pero de todas formas nuestro Cliente – Servidor seria vulnerable.