REDES DE COMUNICACIONES II

Práctica 2 - Seguridad y criptografía

Volver a: Prácticas

1 Introducción

1.1 Desarrollo de la práctica

Python

Puesto que la práctica debe desarrollarse en Python, es conveniente tener ya cierta soltura con el lenguaje antes de empezar a codificar. Para ello, se darán sesiones en las clases de prácticas, y se subirá material al aula virtual.

Depuración

Python, a pesar de ser un lenguaje interpretado, puede (Y DEBE) ser depurado igual que C. De nuevo insistimos en que una práctica con ésta NO debe desarrollarse a base de prints y chapuzas similares. No solo porque no es correcto, sino porque es mucho más lento. Existen diversas alternativas para IDEs para Python, elige el que te resulte más cómodo y pierde el miedo a depurar.

Orden de implementación

Es importante ordenar adecuadamente el desarrollo de las distintas características que debe cumplir la práctica para no retrasarte innecesariamente:

- Como regla general, ataca el problema por partes. Por ejemplo, sería buena idea codificar las funciones que cifran y firmar ficheros aparte, hasta estar seguro de que funcionan correctamente. Luego, integra el código en el resto del programa, lo que facilita y agiliza la depuración. El mismo razonamiento aplica a otros componentes de la práctica: hazte una lista, idéntificalos y diséñalos y codíficalos por partes.
- Una buena forma de empezar a "jugar" con el API y comprender su funcionamiento es con el comando `curl`. Por ejemplo, la siguiente sentencia utiliza endpoint para obtener la clave pública de un usuario:

  ```
  # curl --verbose -H "Authorization: Bearer AaFBe2d7894C1D63" -H "Content-Type: application/json" --data '{"userID":"mi_nia"}' -X POST https://vega.ii.uam.es:8080/api/users/getPublicKey
  ```

  Podrás hacer llamadas similares, cambiando los parámetros necesarios, para entender el formato concreto de cada método del API, antes de codificar esas llamadas en el cliente. Provoca también errores para ver cómo responde el API.

Volver a: Prácticas