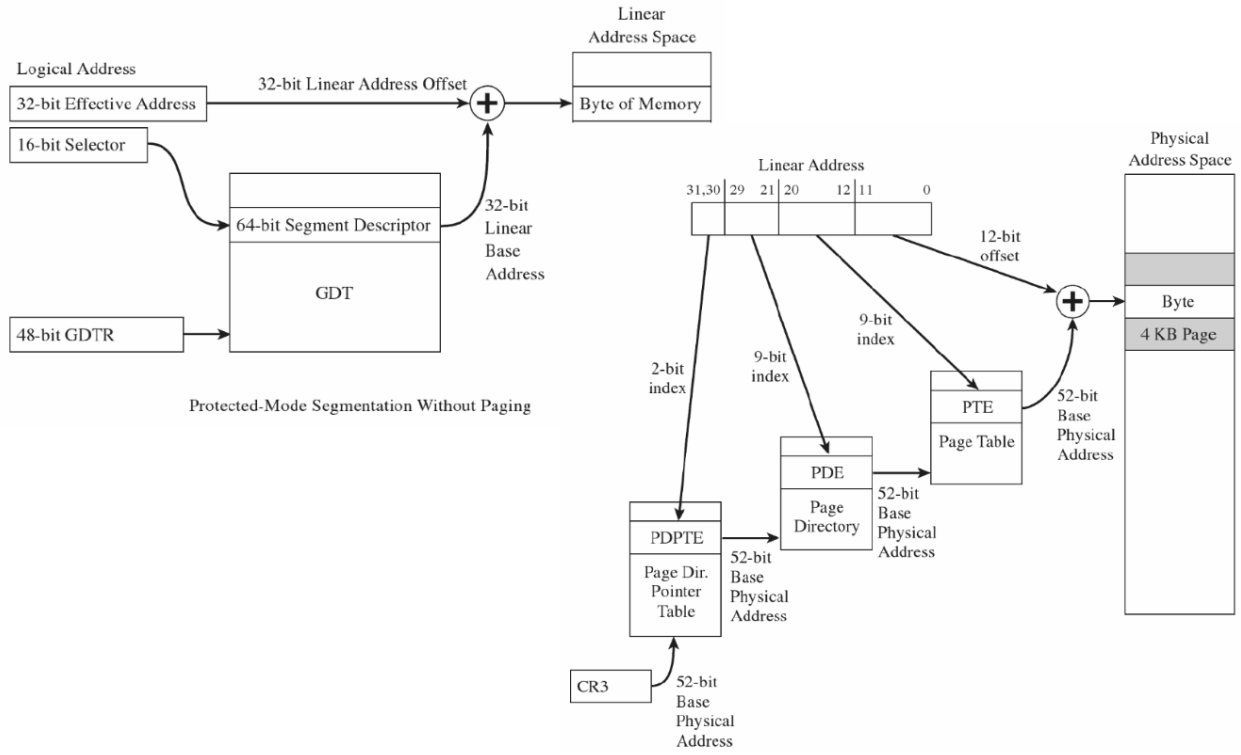
1. **Dibuja un diagrama que muestre el proceso de traducción de dirección lógica a dirección física para una arquitectura Intel x86 (PAE incluido, mostrando el nombre de las diferentes direcciones que toman parte en dicho proceso). (2.5 p.)**



2. **Qué diferencia hay entre dirección virtual y dirección física? (0.5 p.)**

Las direcciones virtuales (o direcciones lineales) son aquella que son utilizadas por un binario una vez se ejecuta (proceso). Estas direcciones son una abstracción introducida por el procesador y el sistema operativo y no representan la dirección real donde los datos están almacenados en los chips de memoria.

Las direcciones físicas son aquellas que realmente identifican las posiciones dentro de un chip de memoria en las que se almacenan datos.

A través de una dirección virtual se debe llegar a la dirección física para obtener los valores que un proceso puede necesitar.

3. **Qué diferencia hay entre un binario y un proceso? (0.5 p.)**

Un binario es un archivo ejecutable cuando está en el disco (file system). Ejemplos de ellos son los PE o los ELF.

Un proceso es un binario que se ha cargado en memoria y está listo para ejecutarse.

4. **Cómo es posible que cada proceso de usuario tenga el mismo espacio de direcciones? (0.5 p.)**

Es posible gracias al sistema de direccionamiento de memoria abstracto que utiliza tanto el procesador como el sistema operativo.

Un proceso X y un proceso Y pueden acceder a una misma dirección virtual Z y obtener diferentes valores ya que debido a la arquitectura de memoria introducida tanto el procesador como por el sistema operativo, un proceso de usuario no trabaja directamente con direcciones físicas y son éstos (el procesador y el sistema operativo) los que se encargan de transformar esas direcciones lógicas a direcciones físicas.

5. **En qué dos "conceptos" está dividido un PE y qué representan? (0.5 p.)**

Cabeceras y secciones. Las cabeceras contienen metadatos y las secciones contienen los datos del binario en si (por ejemplo, las instrucciones).

6. **Qué significa que un binario esté dinámicamente o estáticamente enlazado? Qué inconvenientes tiene que un binario esté estáticamente enlazado a la hora de hacer un análisis estático? (0.5 p.)**

Que un binario esté dinámicamente enlazado significa que dependerá de otras librerías (e.g. .dll o .so) que implementarán funcionalidades que el propio binario necesita (imports).

Si un binario está estáticamente enlazado, esas funcionalidades implementadas en otras librerías de las que depende se incrustaran en el propio binario.

Un binario estáticamente enlazado dificulta el análisis por qué no es tan fácil identificar qué funciones utilizará una vez se ejecute (listando los imports).

7. **Explica lo que es un "import"? (0.5 p.)**

Un import es una función de la que depende un binario y que está implementada por una librería independiente a dicho binario.

8. **Explica lo más detallado posible qué significa "resolver un import"? (1 p.)**

Resolver un import significa identificar qué dirección tiene en memoria un import (una función de la que depende un binario) una vez éste import esté cargado en memoria.

Para saber qué dirección tendrá un import, lo primero que debe ocurrir es que la librería que implementa dicha función/import se cargue en memoria. Una vez la librería se haya cargado en el espacio de memoria del proceso, es cuando se puede saber en qué dirección se ha cargado la función en específico.

9. **Por defecto, en un PE, cuando se resuelven los imports? (0.25 p.)**

En un PE los imports se resuelven cuando el binario se ejecuta (load time) (a través de la IAT y la INT).

10. **Por defecto, en un ELF, cuando se resuelven los imports? (0.25 p.)**

Los imports en un ELF se resuelven una vez se van a utilizar (a través de la PLT y la GOT). Este proceso se conoce como lazy loading.

11. **De toda la estructura de datos de un PE, enumera y explica dos campos. (0.5 p.)**

AddressOfEntryPoint: En general, es la dirección (RVA) de la primera instrucción que se ejecutará una vez el binario se ejecute.

ImageBase: Dirección virtual inicial (del primer byte del ejecutable) en la que el ejecutable querría cargarse en memoria (“preferred” virtual address).

12. **Enumera y explica dos secciones ya sea de un PE, un ELF, o de ambos. (0.5 p.)**

.text: Esta sección contiene las instrucciones de un binario.
.rodata: Esta sección contendrá los valores de las variables que son sólo de lectura.

13. **Explica qué son las secciones y segmentos de un ELF y en qué se diferencian? (0.5 p.)**

Las secciones, igual que en un PE, contienen los datos del ejecutable. Una sección no tiene porqué cargarse en memoria (y esa es una de las cosas que definen los segmentos).

Los segmentos pueden englobar una o varias secciones y es lo que define qué se cargará en memoria una vez el binario se ejecute. Los segmentos definen los permisos (lectura, escritura, ejecución) con los que una o varias secciones se cargarán en memoria.

14. **Como utilizan los creadores de malware las siguientes features de los PE:**
    a. **Resources (0.25 p.)**

Los Resources son utilizados para incrustar otros payloads que el programa malicioso pueda contener. Ejemplos de ello serían stage 2 implants (otros programas maliciosos que se vayan a soltar (drop) en el sistema) o exploits.

b. TLS Callbacks (0.25 p.)

Los TLS Callbacks se utilizan como un mecanismo de evasión para ejecutar código antes de que el flujo de ejecución salte a la dirección especificada por el campo AddressOfEntryPoint de un PE.

15. Qué harías para identificar qué tipo de fichero es un fichero que te han proporcionado sin extensión? (0.5)

Lo copiaría a un sistema Linux y utilizaría la herramienta “file”. En Windows también se podría utilizar TrID.

16. Enumera cuatro herramientas que conozcas para hacer análisis estático de binarios y defínelas usando tus propias palabras. (0.5)

PEView: Una herramienta para analizar ficheros de tipo PE (sólo en modo lectura). Para PE32+ es mejor utilizar CFF Explorer que mejora el análisis de los PE32+, permite extraer fácilmente los resources, editar el PE, ir a offsets fácilmente, etc.

readelf/objdump: Herramientas para analizar ficheros ELF.

IDA Pro: Herramienta comercial para analizar estática y dinámicamente diferentes tipos de binarios (para muchas plataformas). Una de sus ventajas es la información que da relacionada con la API de Windows al analizar ficheros de formato PE.

radare2: Framework de código libre formado por diferentes herramientas (rabin2, rahash2, r2) para analizar (dinámica y estáticamente) y modificar binarios para diferentes plataformas. Una de sus ventajas a parte de ser de código libre es lo versátil que es cuando es necesario automatizar tareas o hacer análisis en masa.