**PE file (Portable Executable file)** là định dạng tệp tin thực thi được sử dụng trên các hệ điều hành Windows. Đây là định dạng chuẩn cho các chương trình, thư viện động (.dll), và các tệp thực thi (.exe) chạy trên nền Windows.

**1. Cấu trúc cơ bản của PE file**

PE file được thiết kế để phù hợp với kiến trúc của hệ điều hành Windows, bao gồm nhiều phần để chứa mã, dữ liệu, và thông tin cần thiết để thực thi chương trình. Các phần chính của PE file bao gồm:

**a. DOS Header**

* Đây là phần đầu tiên của PE file, tương thích với hệ điều hành MS-DOS.
* Chứa một thông điệp nhỏ "This program cannot be run in DOS mode" để báo lỗi nếu tệp được chạy trên môi trường không hỗ trợ Windows.

**b. PE Header (Signature)**

* Đánh dấu tệp là PE file bằng chuỗi ký tự PE\0\0.
* Chứa thông tin cơ bản như loại tệp (32-bit hoặc 64-bit), thời gian biên dịch, kích thước của các thành phần, và điểm vào chính của chương trình (Entry Point).

**c. Optional Header**

* Không thực sự "tùy chọn", mà là nơi chứa thông tin chi tiết về cách tải và thực thi chương trình.
* Các thông tin quan trọng:
  + **Entry Point Address**: Điểm bắt đầu thực thi chương trình.
  + **Base Address**: Địa chỉ mặc định để tải file vào bộ nhớ.
  + **Section Alignment**: Quy tắc sắp xếp các section trong bộ nhớ.

**d. Section Table**

* Một bảng mô tả các phần (section) của PE file, như mã thực thi, dữ liệu tĩnh, tài nguyên, và dữ liệu khởi tạo.
* Một số section thường gặp:
  + .text: Chứa mã thực thi của chương trình.
  + .data: Chứa dữ liệu khởi tạo.
  + .rdata: Chứa dữ liệu chỉ đọc (như hằng số).
  + .rsrc: Chứa tài nguyên (như hình ảnh, chuỗi, biểu tượng).
  + .reloc: Chứa thông tin để tái định vị nếu tệp không thể tải vào địa chỉ cơ sở mặc định.

**e. Import/Export Table**

* **Import Table**: Liệt kê các thư viện và hàm mà tệp cần sử dụng (ví dụ, các API từ Windows DLLs).
* **Export Table**: Liệt kê các hàm hoặc tài nguyên mà tệp cung cấp cho các chương trình khác.

**f. Debug Information**

* Cung cấp thông tin để hỗ trợ gỡ lỗi (nếu được bật khi biên dịch).

**2. Quy trình tải PE file**

1. **Nạp vào bộ nhớ (Loading):** Windows Loader đọc PE Header và Section Table để biết cách tải tệp vào bộ nhớ.
2. **Tái định vị (Relocation):** Nếu địa chỉ cơ sở không khả dụng, hệ điều hành sẽ di chuyển các phần cần thiết.
3. **Liên kết động (Dynamic Linking):** Hệ điều hành giải quyết các tham chiếu đến các thư viện bên ngoài (như kernel32.dll, user32.dll).
4. **Thực thi (Execution):** CPU bắt đầu thực thi từ Entry Point đã xác định.

**3. Ưu điểm của định dạng PE**

* **Khả năng di động:** PE file có thể được sử dụng trên các nền tảng Windows khác nhau (32-bit, 64-bit).
* **Tích hợp chặt chẽ với Windows Loader:** Cho phép tối ưu hóa việc quản lý bộ nhớ và tải thư viện động.
* **Hỗ trợ liên kết động:** Giảm kích thước tệp nhờ tái sử dụng các thư viện dùng chung.

**4. Công cụ phân tích PE file**

* **PEiD**: Dùng để phát hiện trình biên dịch hoặc công cụ đóng gói.
* **Dependency Walker**: Xem các thư viện mà PE file cần.
* **CFF Explorer**: Phân tích cấu trúc PE file chi tiết.
* **IDA Pro**: Phân tích mã thực thi và tệp nhị phân.

**5. Ứng dụng thực tế**

* Phân tích phần mềm độc hại: Các chuyên gia an ninh mạng thường phân tích PE file để tìm hiểu cách thức hoạt động của malware.
* Kiểm tra và gỡ lỗi: Các lập trình viên sử dụng công cụ để tối ưu và sửa lỗi tệp thực thi.
* Reverse engineering: Khám phá cấu trúc và logic bên trong PE file nhằm phân tích hoặc sửa đổi tệp.