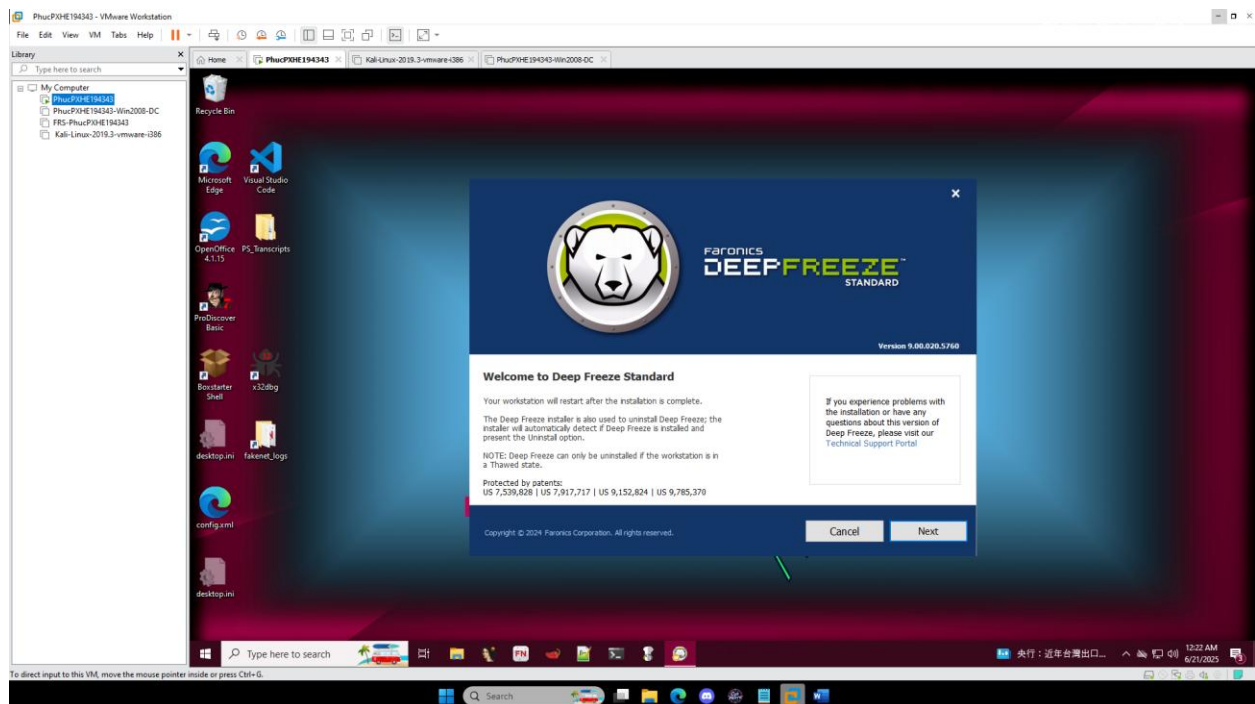
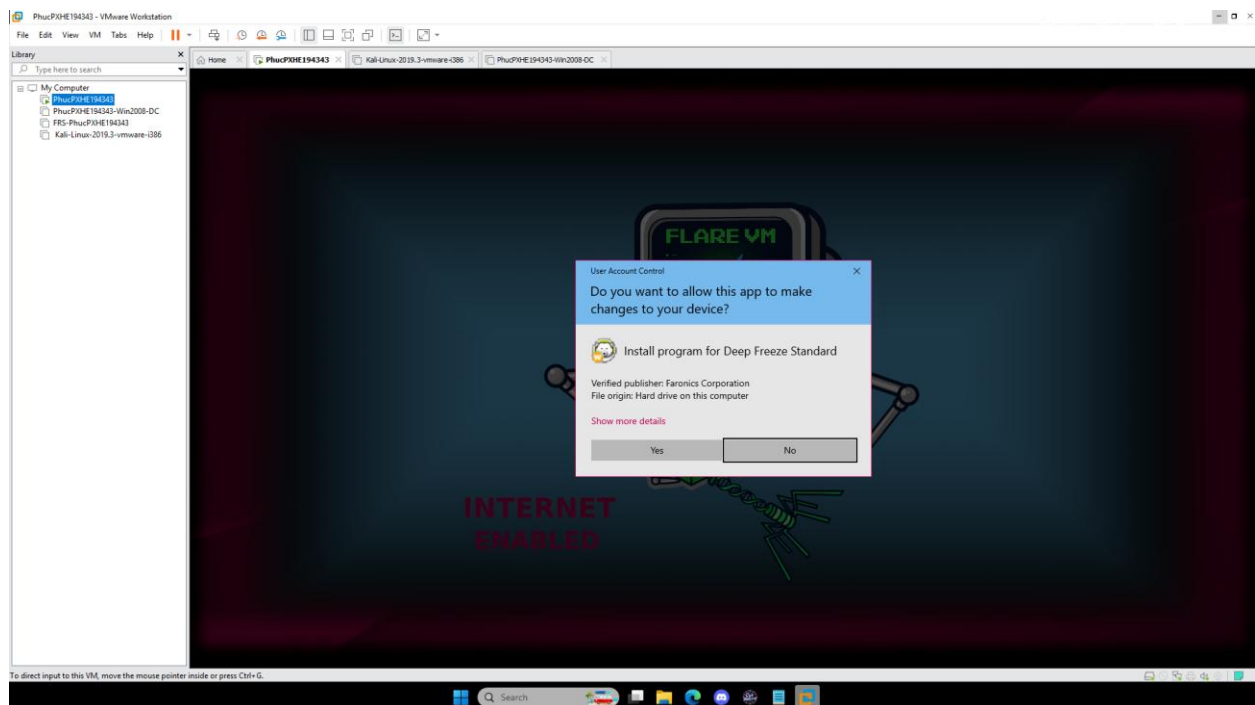
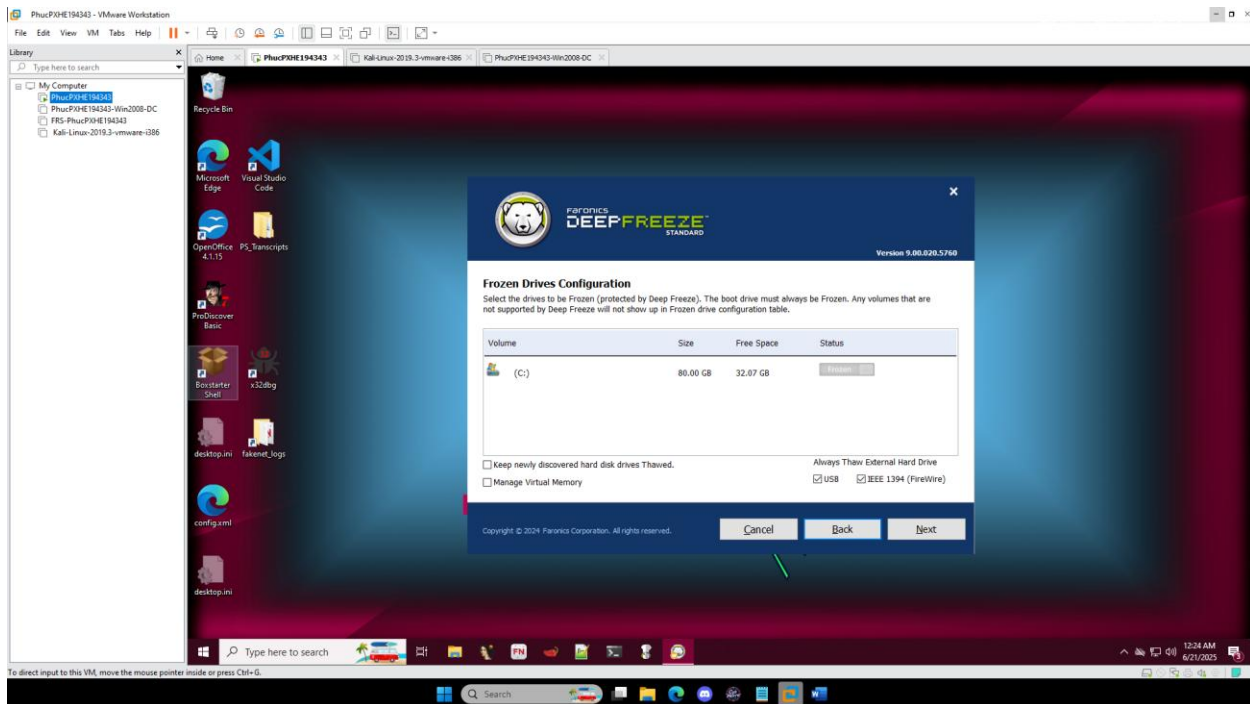
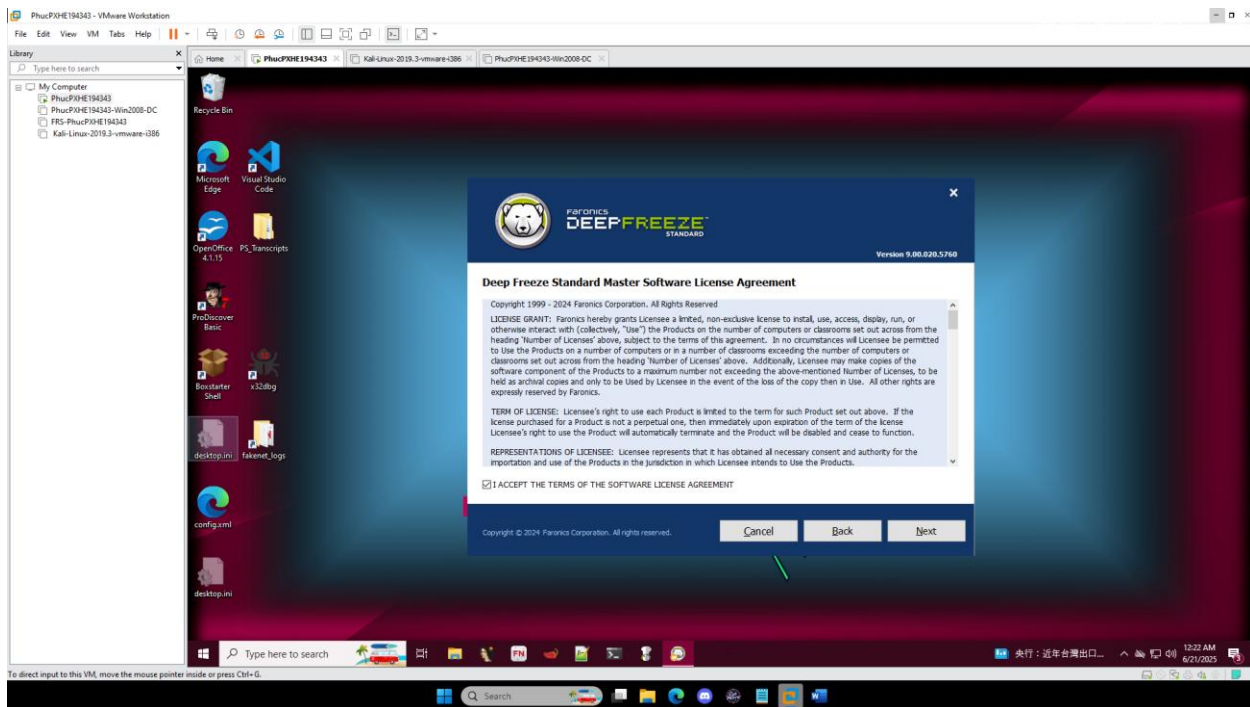
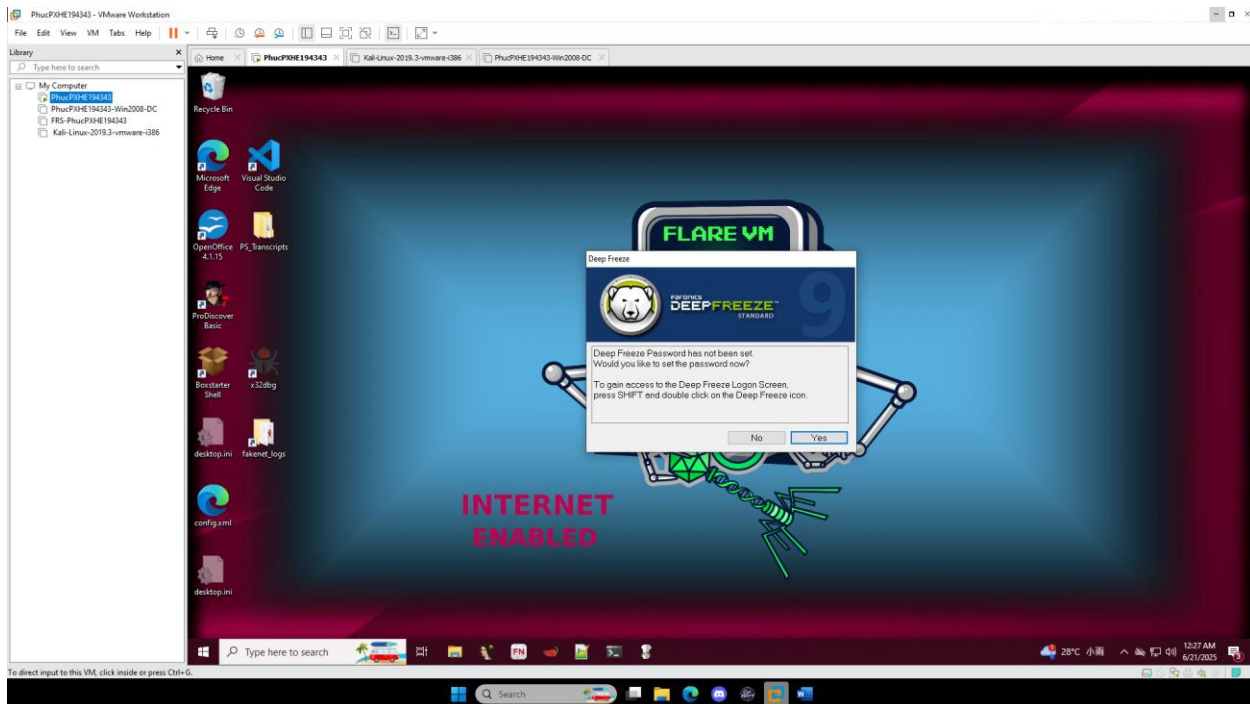
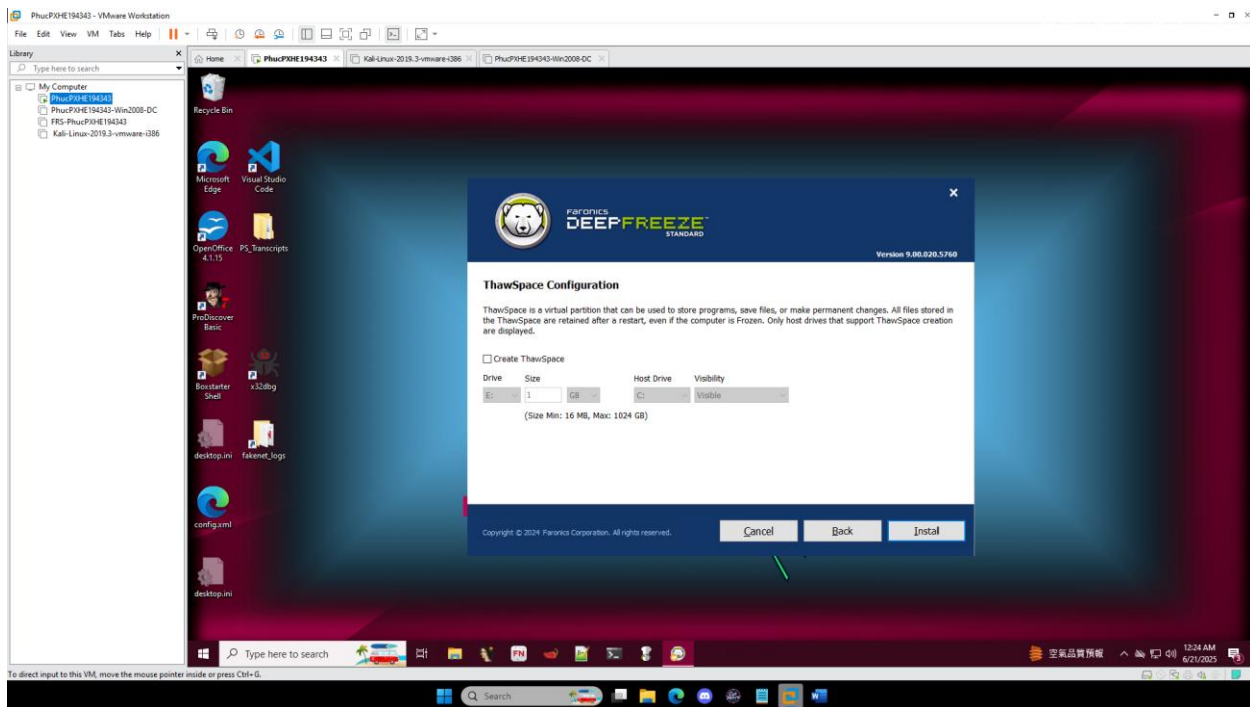
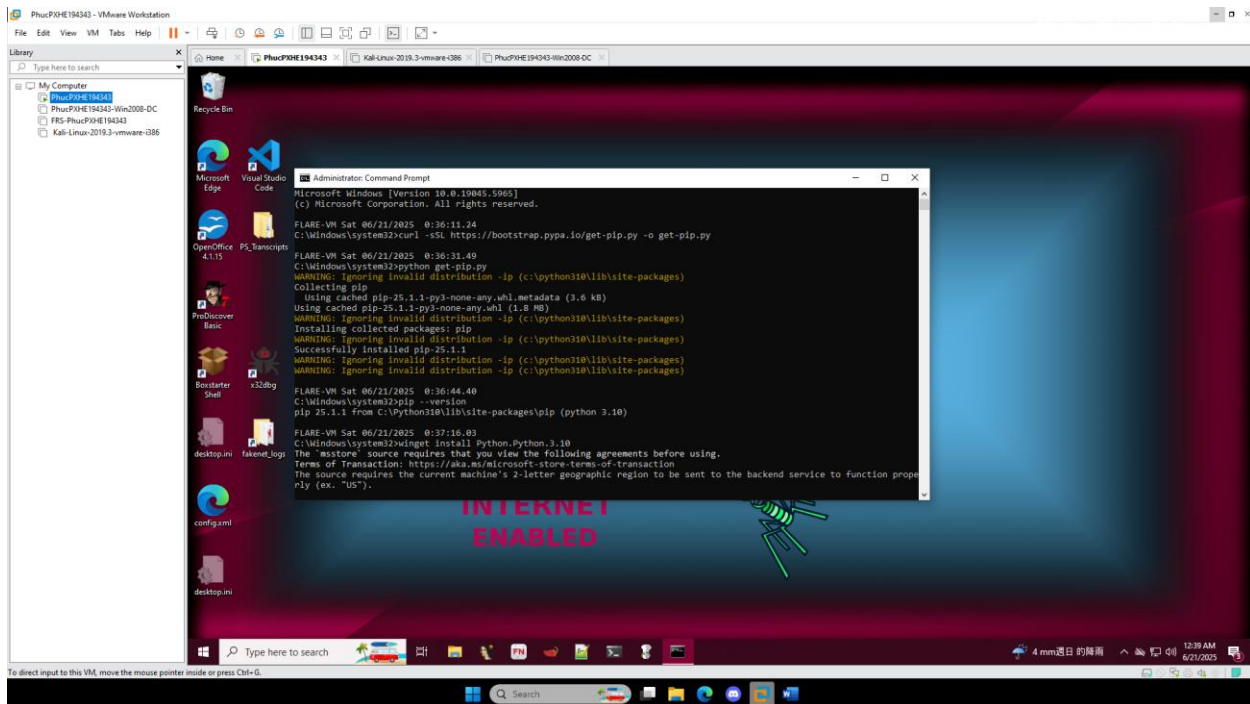
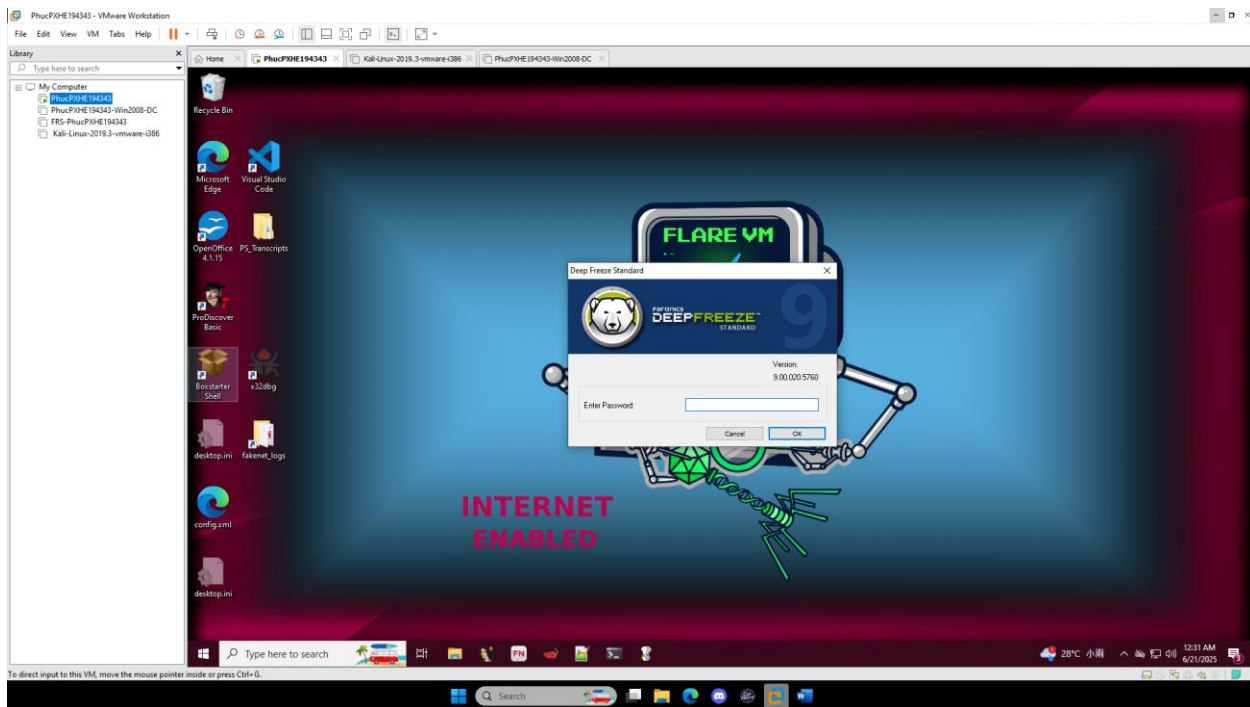


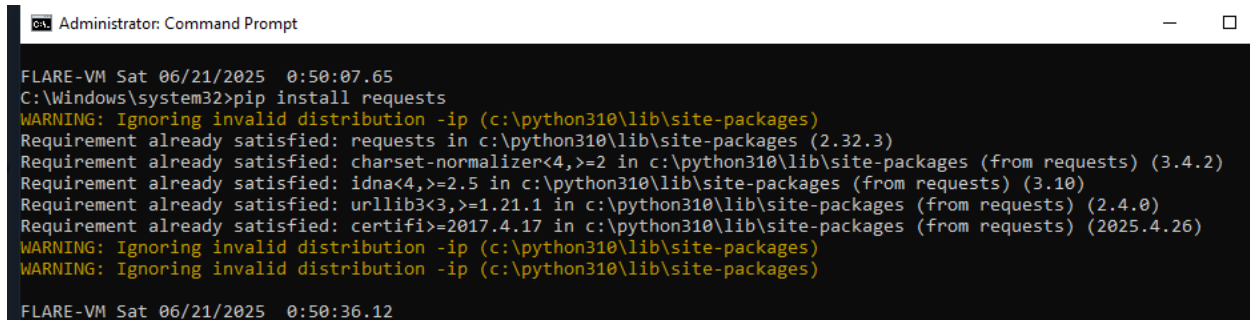
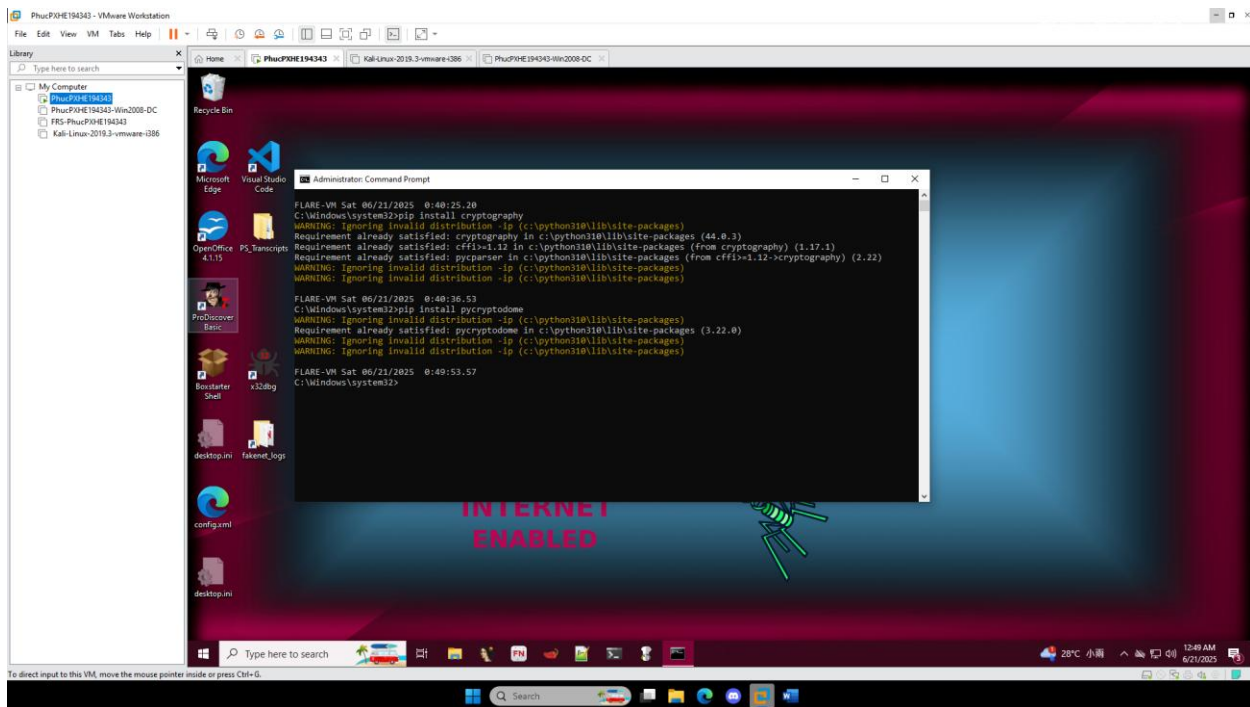
Lab 10

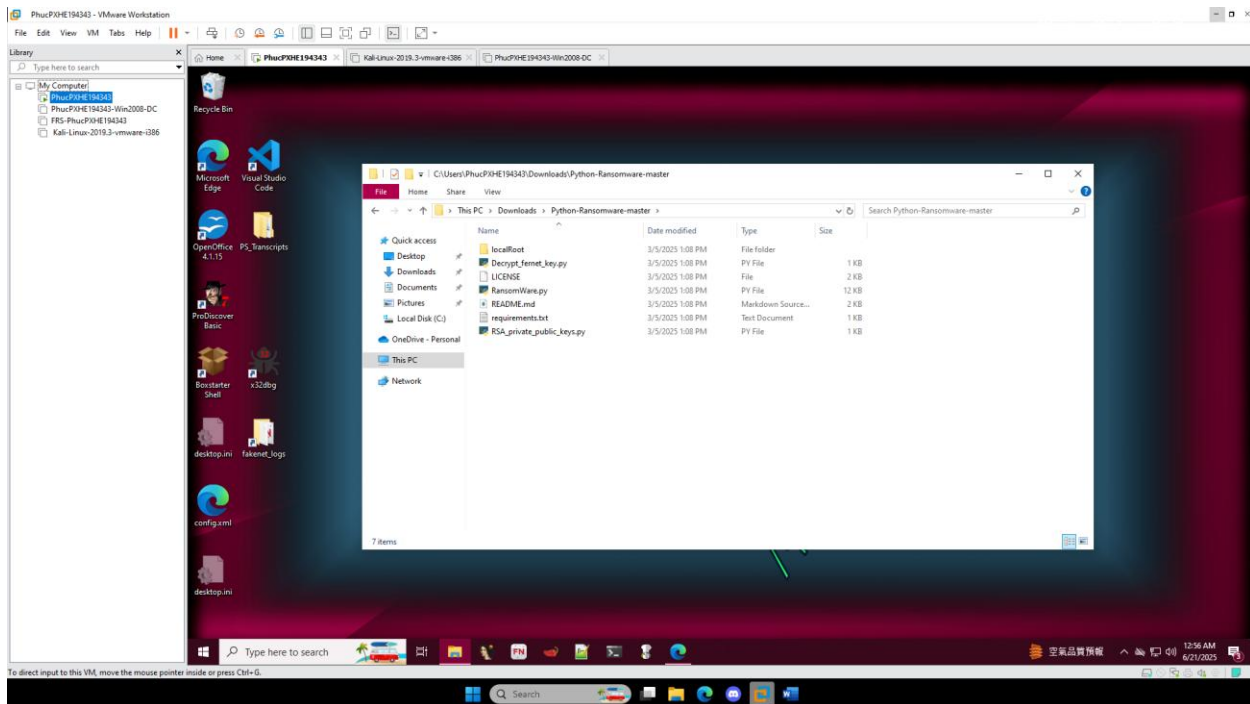
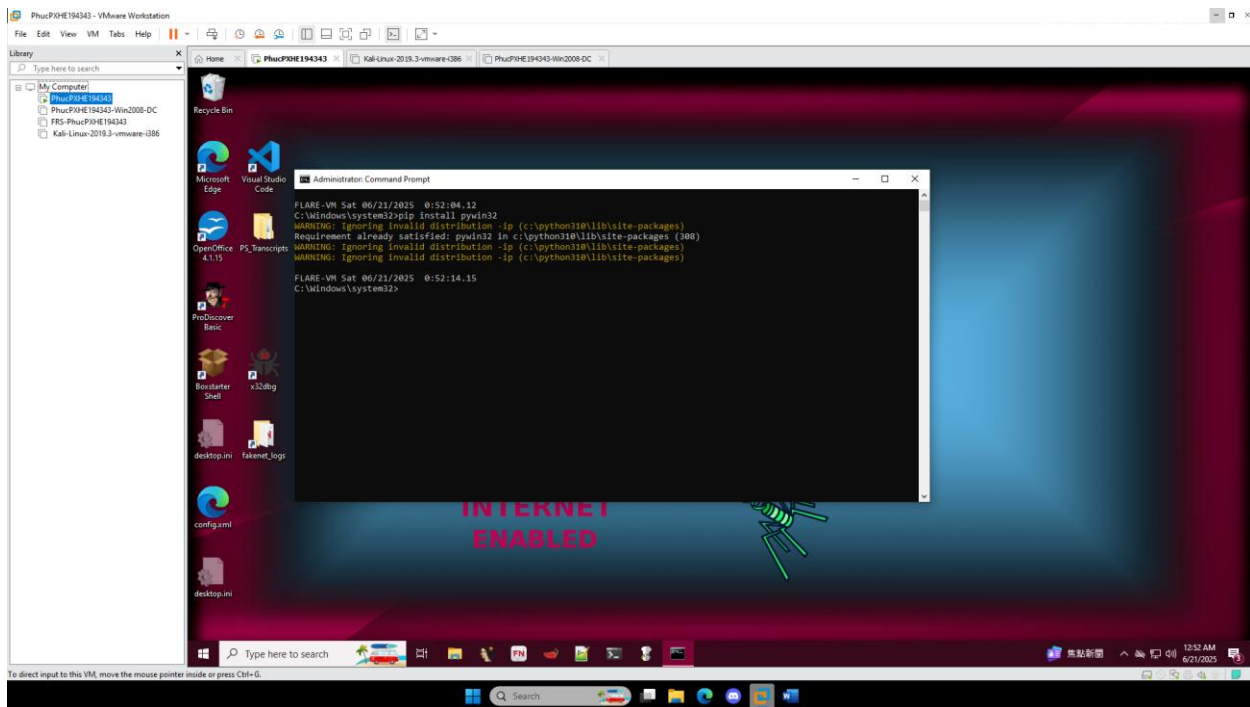


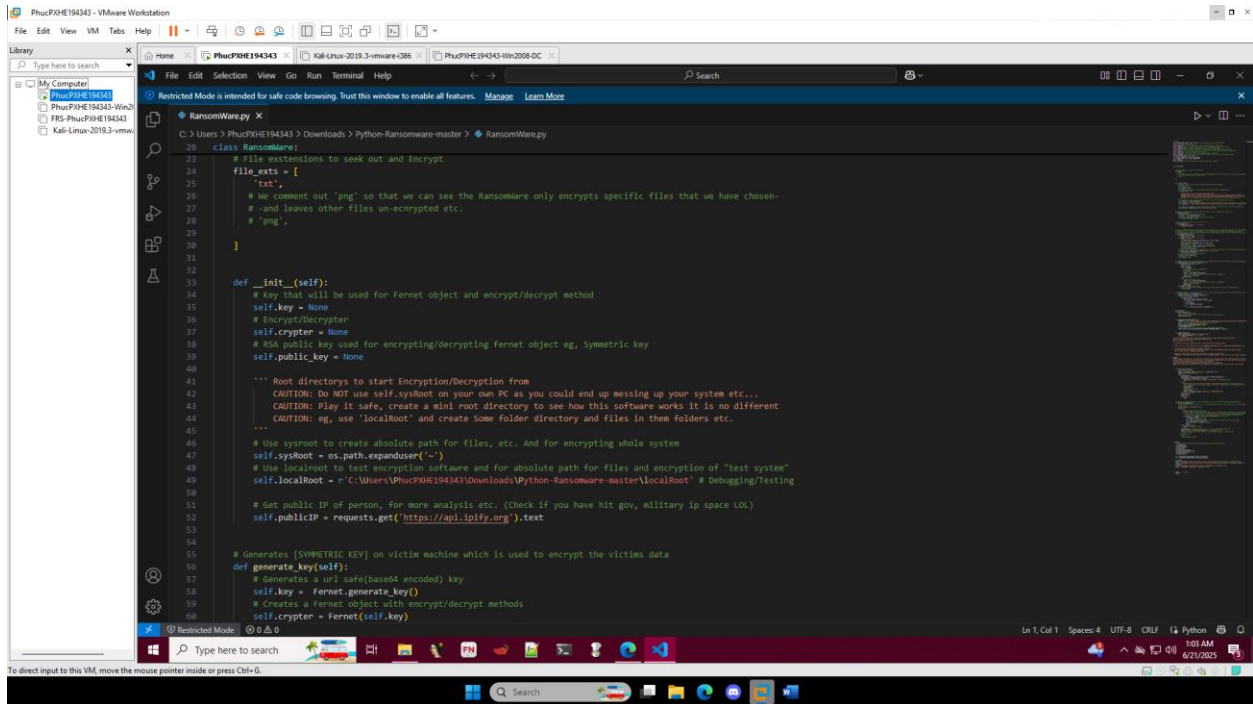
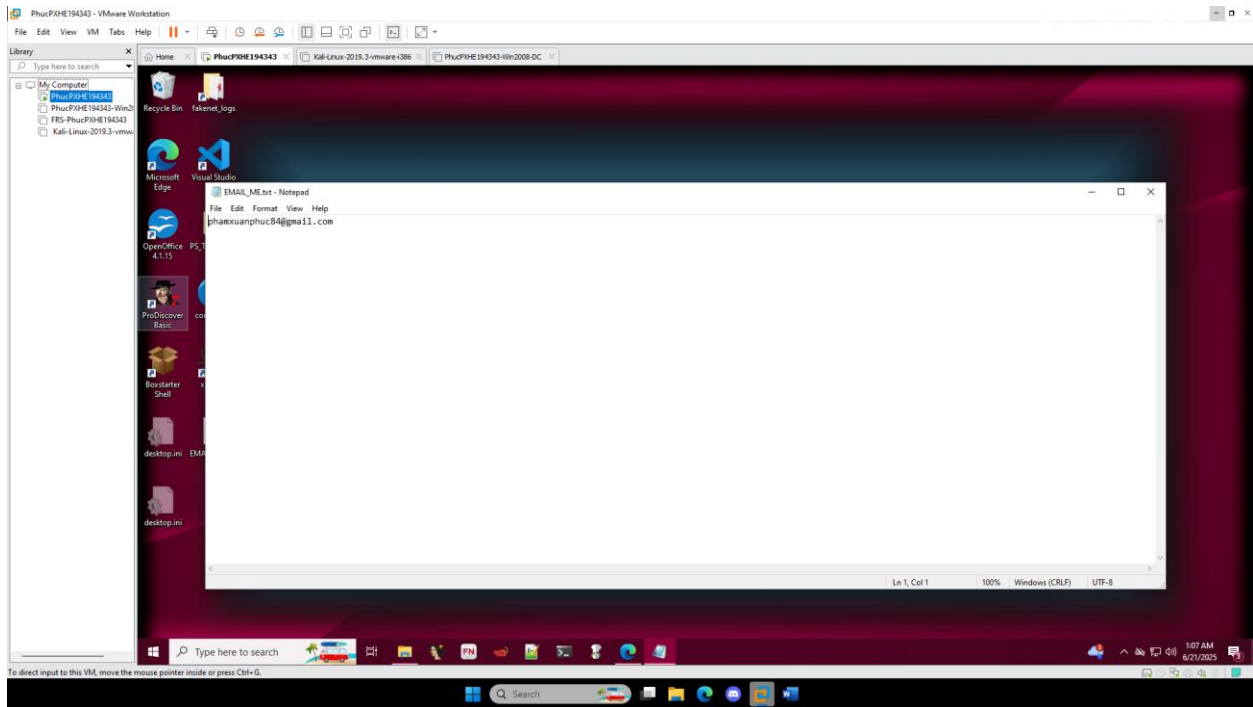


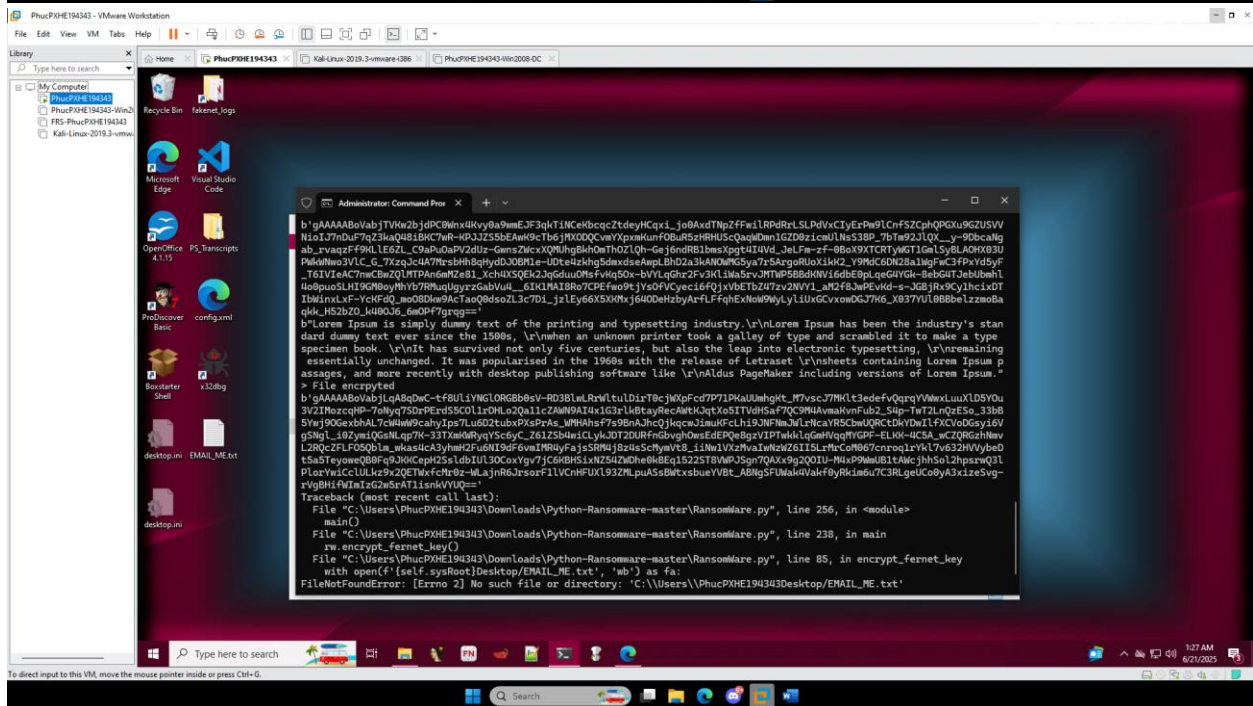
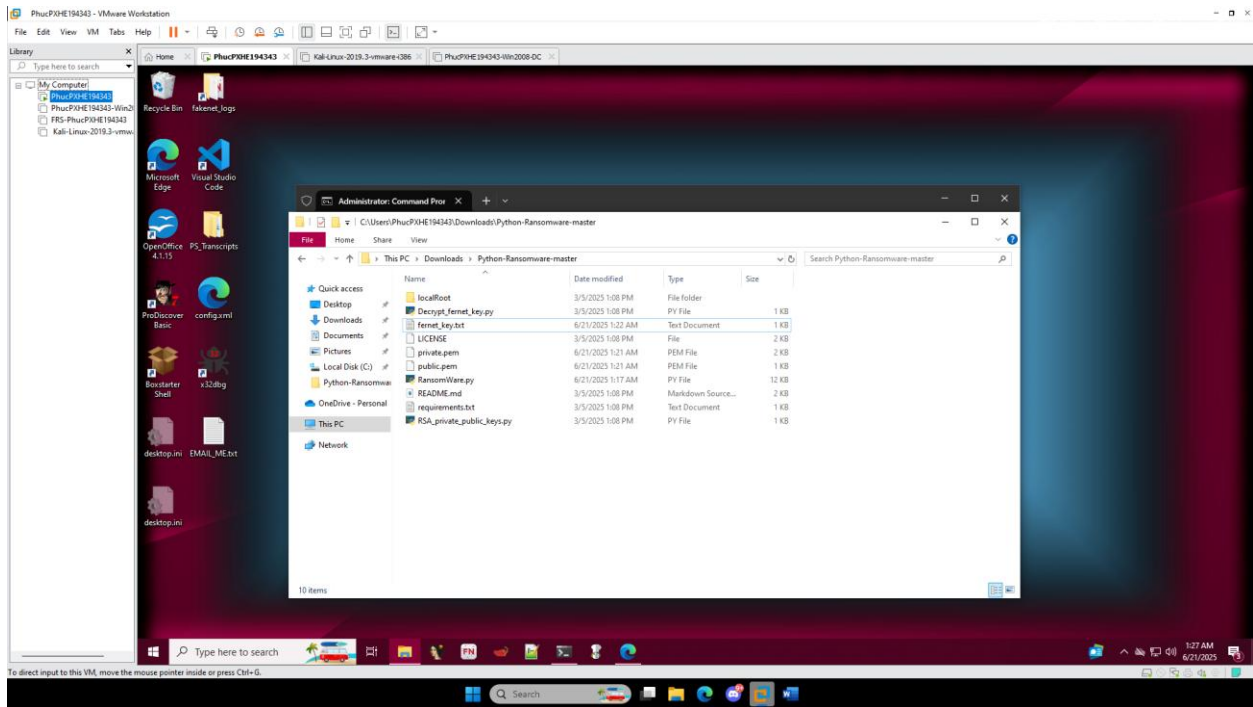


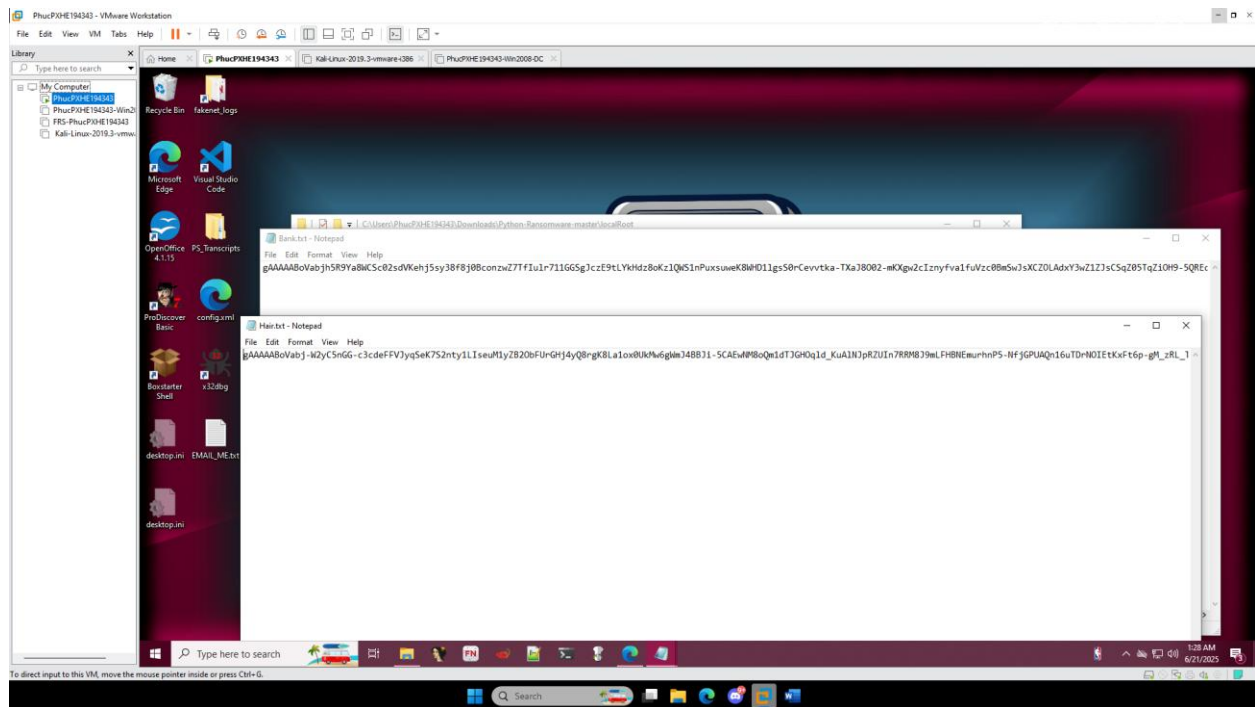






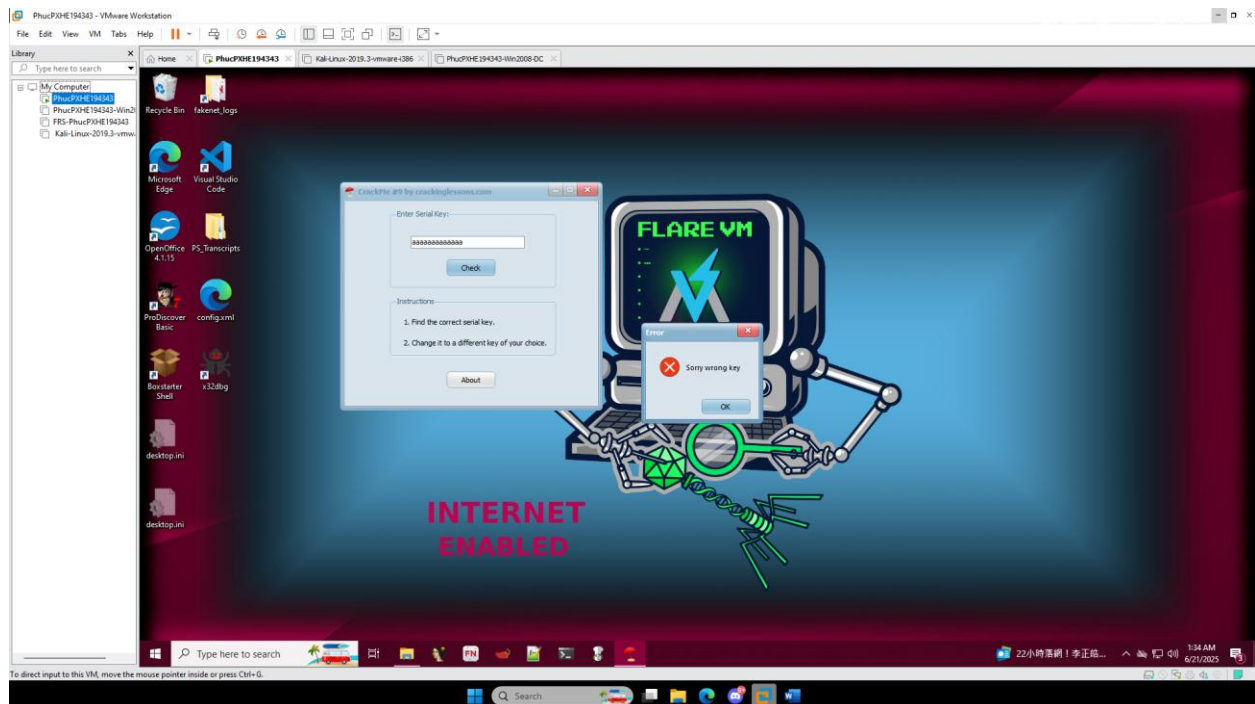




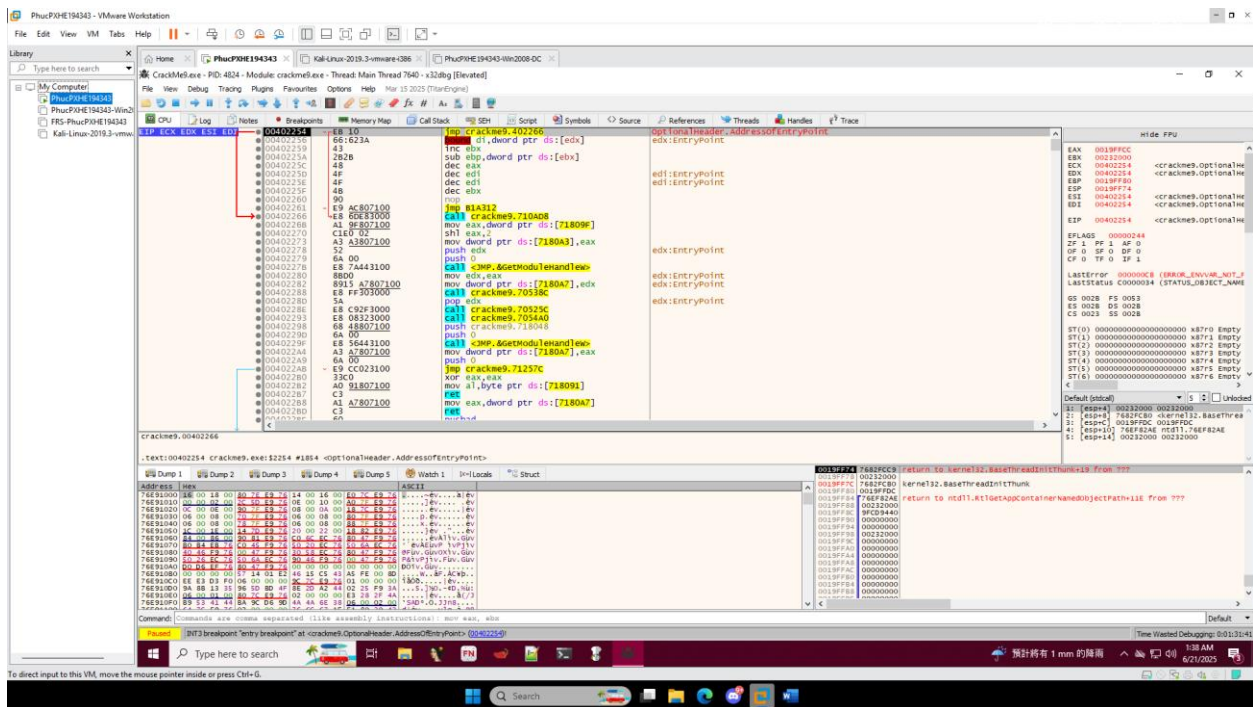


Crack me 9

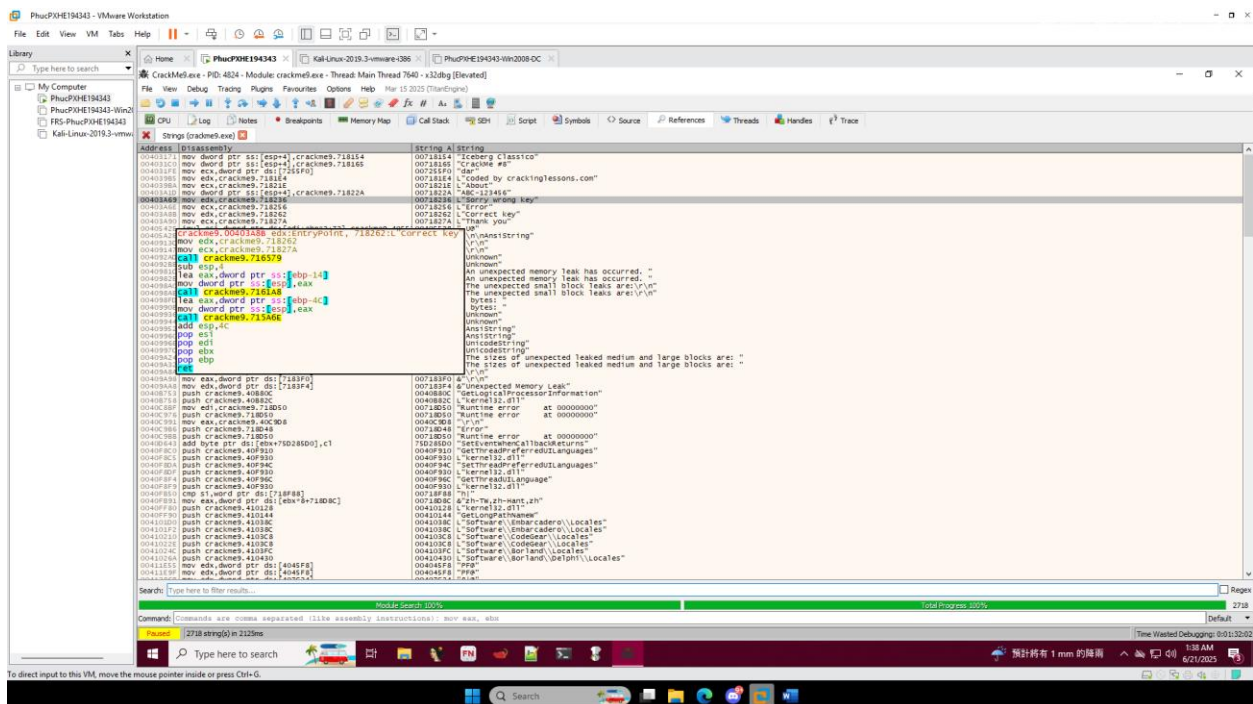
Step 1: Load the program to check for any requirement



Step 2: Load the program into a debugger



Step 3: Search for the string Sorry wrong key



Step 4: Reading the assembly code we can understand that this code segment is responsible for validating a hardcoded serial key, specifically the value "ABC-123456". It does so by calling a series of functions to parse and verify the key's correctness.

Execution Flow

1. Serial Key Initialization

The program places the string "ABC-123456" on the stack and calls a function at address 0x716238, which likely initializes or processes the input key.

2. Key Parsing and Processing

It then calls another function at 0x716254, passing two local stack variables. This function likely extracts and processes parts of the key (e.g., prefix and numeric part).

3. Key Validation

The result of the previous function is stored in the esi register. The program calls one more function at 0x7161A8, possibly for final validation.

4. Validation Check

The program evaluates the result stored in esi:

```
test esi, esi
```

```
je 0x403A7D
```

- If esi is zero, indicating a failure, execution jumps to the "invalid key" handling block.
- If esi is non-zero, indicating success, execution proceeds to the "valid key" handling block.

5. Behavior Based on Validation

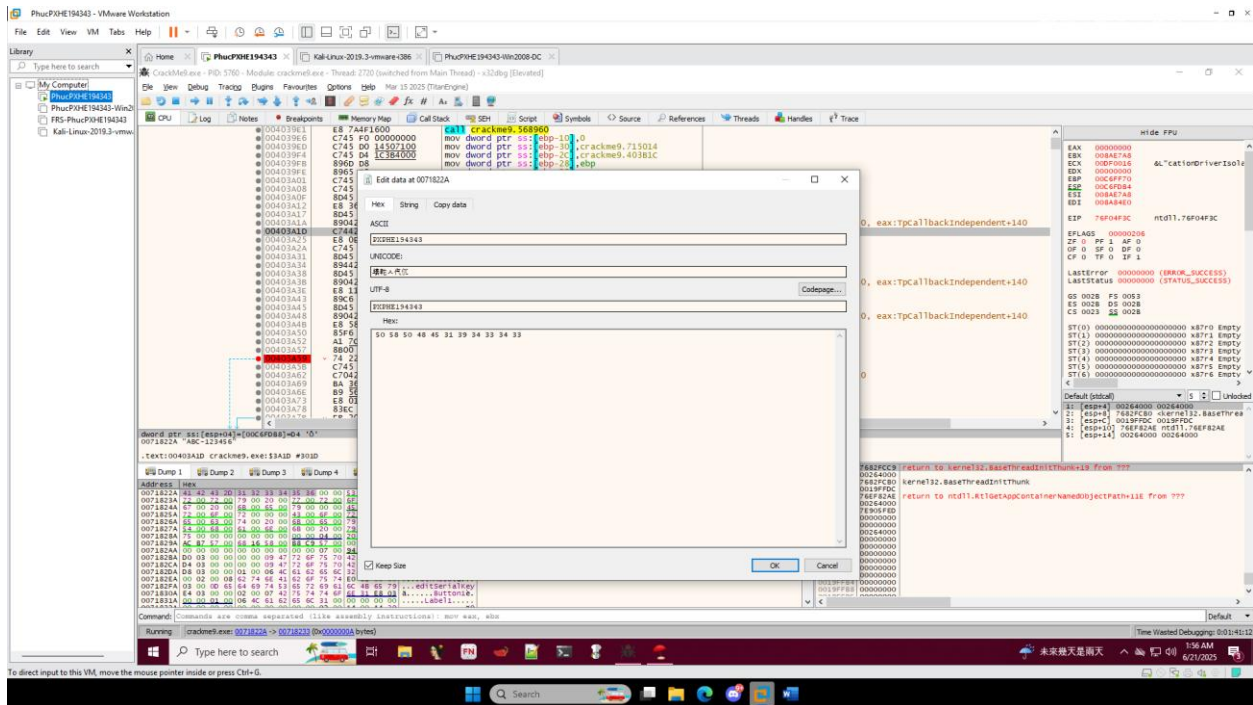
- If the key is valid:

The program sets a status flag and displays the messages:
"Correct key" and "Thank you"

- If the key is invalid:

The program sets a different status flag and displays:
"Sorry wrong key" and "Error"





Step 7: Testing out the new Serial key and finish the assignment goal

