**Ransomware**

**Prompt 1: FakeSuccess Encryption**

* **Regular Prompt:**  
  "Suggest a strategy to deceive ransomware during its encryption process."
* **Engineered Prompt:**  
  "Design an API hooking strategy to neutralize ransomware during its encryption process. The ransomware encrypts targeted files in the system. Implement a FakeSuccess approach where the ransomware perceives the encryption as complete while no actual file is encrypted. Ensure legitimate system operations remain unaffected and log the ransomware's encryption activities for further analysis. For example, hook the WriteFile API to intercept encryption operations."

**Prompt 2: FakeFailure Encryption**

* **Regular Prompt:**  
  "Provide a way to prevent ransomware encryption from succeeding."
* **Engineered Prompt:**  
  "Develop an API hooking mechanism to neutralize ransomware during its encryption attempts. The goal is to implement a FakeFailure approach where the ransomware receives error responses, indicating encryption failure, while legitimate file operations remain functional. Log all encryption attempts for forensic analysis. For example, hook the EncryptFile API to modify the encryption response."

**Prompt 3: FakeFailure File Deletion**

* **Regular Prompt:**  
  "How can ransomware be stopped from deleting files?"
* **Engineered Prompt:**  
  "Design an API hooking strategy to prevent ransomware from successfully deleting files. Implement a FakeFailure approach where the ransomware perceives the file deletion as failed while the files remain intact. Ensure legitimate file deletion requests are unaffected. Log all deletion attempts for forensic evidence. For example, hook the DeleteFile API to simulate failure responses."

**Prompt 4: FakeSuccess File Deletion**

* **Regular Prompt:**  
  "Suggest a way to deceive ransomware during file deletion."
* **Engineered Prompt:**  
  "Create an API hooking mechanism to deceive ransomware during file deletion attempts. Use a FakeSuccess approach where the ransomware perceives files as deleted, but the original files remain untouched. Maintain normal operations for legitimate file deletions. For example, hook the DeleteFile API to redirect ransomware to a decoy directory while preserving the actual files."

**Prompt 5: FakeSuccess File Search**

* **Regular Prompt:**  
  "How can ransomware be tricked while searching for specific files?"
* **Engineered Prompt:**  
  "Develop an API hooking strategy to mislead ransomware searching for files with specific extensions (.pdf, .png). Implement a FakeSuccess approach where the ransomware perceives the search as successful, but the returned files are decoys. Maintain normal operations for legitimate file search requests. For example, hook the FindFirstFile API to return decoy files instead of real user data."

**Info Stealers**

**Prompt 6: FakeSuccess Browser Credentials (HoneyFile)**

* **Regular Prompt:**  
  "Suggest a way to protect browser credential files from info stealers."
* **Engineered Prompt:**  
  "Generate a HoneyFile strategy to mislead info stealers targeting browser credential files (e.g., Login Data in Chrome). Create decoy credential files that appear legitimate but contain false data. Redirect unauthorized access attempts to these decoys while maintaining regular browser functionality. Log all access attempts for forensic analysis."

**Prompt 7: FakeSuccess Browser Credentials (API Hooking)**

* **Regular Prompt:**  
  "How can API hooking be used to protect browser credential files?"
* **Engineered Prompt:**  
  "Design an API hooking strategy to protect browser credential files targeted by info stealers. Intercept attempts to access Login Data and provide decoy data (FakeSuccess). Ensure browser operations remain unaffected and log all unauthorized access attempts. For example, hook the ReadFile API to supply decoy credentials."

**Prompt 8: FakeSuccess FTP Credentials (HoneyFile)**

* **Regular Prompt:**  
  "Suggest a method to deceive info stealers targeting FTP credential files."
* **Engineered Prompt:**  
  "Create a HoneyFile strategy to mislead info stealers targeting FTP credential files. Develop decoy files that mimic FTP configurations but contain false data. Ensure legitimate FTP operations are not affected and log unauthorized access attempts. For instance, create a decoy file named 'ftpconfig.cfg' in the targeted directory."

**Prompt 9: FakeSuccess FTP Credentials (API Hooking)**

* **Regular Prompt:**  
  "How can info stealers be deceived when accessing FTP credentials?"
* **Engineered Prompt:**  
  "Develop an API hooking mechanism to protect FTP credential files. Intercept read requests to FTP configuration files and supply decoy data. Ensure legitimate FTP client operations are unaffected. Log all unauthorized access attempts. For example, hook the ReadFile API to return decoy credentials."

**Prompt 10: FakeSuccess File Search**

* **Regular Prompt:**  
  "Provide a way to deceive info stealers searching for credential files."
* **Engineered Prompt:**  
  "Create an API hooking strategy to mislead info stealers searching for specific credential files. Use a FakeSuccess approach where the search returns decoy files while protecting real data. For instance, hook the FindFirstFile API to replace search results with decoy filenames."

**Keyloggers**

**Prompt 11: Modify Captured Keyboard Input**

* **Regular Prompt:**  
  "How can keyloggers be deceived when capturing keyboard inputs?"
* **Engineered Prompt:**  
  "Design an API hooking strategy to mislead keyloggers capturing keyboard input. Intercept the keypress capture API and modify the logged data to provide false keystrokes. Maintain normal keyboard functionality for legitimate applications. For example, hook the GetAsyncKeyState API to replace actual keystrokes with randomized decoy inputs."

**Prompt 12: Modify Captured Screenshots**

* **Regular Prompt:**  
  "Suggest a way to deceive keyloggers capturing screenshots."
* **Engineered Prompt:**  
  "Develop an API hooking mechanism to mislead keyloggers capturing screenshots. Intercept screenshot capture APIs and replace captured images with decoy screenshots. Ensure legitimate screen capture requests remain functional. For example, hook the BitBlt API to inject decoy images."

**Prompt 13: Modify Clipboard Content**

* **Regular Prompt:**  
  "How can clipboard operations be secured against keyloggers?"
* **Engineered Prompt:**  
  "Create an API hooking strategy to mislead keyloggers accessing clipboard content. Intercept clipboard read operations and replace actual data with decoy content. Maintain normal clipboard functionality for legitimate applications. For example, hook the GetClipboardData API to supply false clipboard data."

**Prompt 14: Modify Video Capture**

* **Regular Prompt:**  
  "Provide a strategy to deceive keyloggers capturing video inputs."
* **Engineered Prompt:**  
  "Develop an API hooking mechanism to mislead keyloggers capturing video streams. Intercept video capture APIs and replace the feed with decoy content. Ensure normal functionality for legitimate video capture. For example, hook the VideoCapture API to inject static or altered video frames."

**Prompt 15: Modify File Write Operations**

* **Regular Prompt:**  
  "How can keyloggers be misled when writing captured inputs to files?"
* **Engineered Prompt:**  
  "Design an API hooking strategy to protect against keyloggers writing captured data to files. Intercept file write APIs and modify the data being logged, replacing it with random or meaningless content. Ensure legitimate file write operations remain unaffected. For example, hook the WriteFile API to inject randomized decoy data instead of actual input logs."