THICK CLIENT PENTESTING CHECKLIST

OWASP Based Checklist 🌟🌟

80+ Test Cases 🚀🚀

INFORMATION GATHERING

1. Information Gathering

□ Find out the application architecture (two-tier or three-tier)

□ Find out the technologies used (languages and frameworks)

□ Identify network communication

□ Observe the application process

□ Observe each functionality and behavior of the application

□ Identify all the entry points

□ Analyze the security mechanism (authorization and authentication)

2. Tools Used

□ CFF Explorer

□ Sysinternals Suite

□ Wireshark

□ PEid

□ Detect It Easy (DIE)

□ Strings

GUI TESTING

1. Test For GUI Object Permission

□ Display hidden form object

□ Try to activate disabled functionalities

□ Try to uncover the masked password

2. Test GUI Content

□ Look for sensitive information

3. Test For GUI Logic

□ Try for access control and injection-based vulnerabilities

□ Bypass controls by utilizing intended GUI functionality

□ Check improper error handling

□ Check weak input sanitization

□ Try privilege escalation (unlocking admin features to normal users)

□ Try payment manipulation

4. Tools Used

□ UISpy

□ Winspy++

□ Window Detective

□ Snoop WPF

FILE TESTING

1. Test For Files Permission

□ Check permission for each and every file and folder

2. Test For File Continuity

□ Check strong naming

□ Authenticate code signing

3. Test For File Content Debugging

□ Look for sensitive information on the file system (symbols, sensitive data, passwords, configurations)

□ Look for sensitive information on the config file

□ Look for Hardcoded encryption data

□ Look for Clear text storage of sensitive data

□ Look for side-channel data leakage

□ Look for unreliable log

4. Test For File And Content Manipulation

□ Try framework backdooring

□ Try DLL preloading

□ Perform Race condition check

□ Test for Files and content replacement

□ Test for Client-side protection bypass using reverse engineering

5. Test For Function Exported

□ Try to find the exported functions

□ Try to use the exported functions without authentication

6. Test For Public Methods

□ Make a wrapper to gain access to public methods without authentication

7. Test For Decompile And Application Rebuild

□ Try to recover the original source code, passwords, keys

□ Try to decompile the application

□ Try to rebuild the application

□ Try to patch the application

8. Test For Decryption And DE obfuscation

□ Try to recover original source code

□ Try to retrieve passwords and keys

□ Test for lack of obfuscation

9. Test For Disassemble and Reassemble

□ Try to build a patched assembly

10. Tools Used

□ Strings

□ dnSpy

□ Procmon

□ Process Explorer

□ Process Hacker

REGISTRY TESTING

1. Test For Registry Permissions

- □ Check read access to the registry keys
- □ Check to write access to the registry keys

2. Test For Registry Contents

- □ Inspect the registry contents
- □ Check for sensitive info stored on the registry
- □ Compare the registry before and after executing the application

3. Test For Registry Manipulation

- □ Try for registry manipulation
- □ Try to bypass authentication by registry manipulation
- □ Try to bypass authorization by registry manipulation

4. Tools Used

- □ Regshot
- □ Procmon
- □ Accessenum

NETWORK TESTING

1. Test For Network

□ Check for sensitive data in transit

□ Try to bypass firewall rules

□ Try to manipulate network traffic

2. Tools Used

□ Wireshark

□ TCPview

ASSEMBLY TESTING

1. Test For Assembly

□ Verify Address Space Layout Randomization (ASLR)

□ Verify SafeSEH

□ Verify Data Execution Prevention (DEP)

□ Verify strong naming

□ Verify ControlFlowGuard

□ Verify HighentropyVA

2. Tools Used

□ PESecurity

MEMORY TESTING

1. Test For Memory Content

- □ Check for sensitive data stored in memory

2. Test For Memory Manipulation

- □ Try for memory manipulation
- □ Try to bypass authentication by memory manipulation
- □ Try to bypass authorization by memory manipulation

3. Test For Run Time Manipulation

- □ Try to analyze the dump file
- □ Check for process replacement
- □ Check for modifying assembly in the memory
- □ Try to debug the application
- □ Try to identify dangerous functions
- □ Use breakpoints to test each and every functionality

4. Tools Used

- □ Process Hacker
- □ HxD
- □ Strings

TRAFFIC TESTING

1. Test For Traffic

□ Analyze the flow of network traffic

□ Try to find sensitive data in transit

2. Tools Used

□ Echo Mirage

□ MITM Relay

□ Burp Suite

COMMON VULNERABILITIES TESTING

1. Test For Common Vulnerabilities

□ Try to decompile the application

□ Try reverse engineering

□ Try to test with OWASP WEB Top 10

□ Try to test with OWASP API Top 10

□ Test for DLL Hijacking

□ Test for signature checks (Use Sigcheck)

□ Test for binary analysis (Use Binscope)

□ Test for business logic errors

□ Test for TCP/UDP attacks

□ Test with automated scanning tools (Use Visual Code Grepper - VCG)