Ghanshyam Singh(255)

🛠 Tool Name:

Dependency Walker & dex2jar

📜 History:

- **Dependency Walker** was originally developed by Microsoft as a utility to view dependencies of Windows modules (.exe, .dll, etc.).
- **dex2jar** is an open-source project created to convert Android .dex (Dalvik Executable) files to Java .jar files, enabling easier analysis of Android apps using Java decompilers.

📄 Description:

A combined approach using Dependency Walker and dex2jar helps forensic analysts examine Windows and Android binaries, allowing reverse engineering, malware analysis, and static code inspection.

📌 What Is This Tool About?

- **Dependency Walker** allows investigators to map DLL dependencies, detect missing or suspicious modules, and analyze API calls used by Windows executables.
- **dex2jar** converts Android application binaries (.apk/.dex) into Java-readable format, which can then be inspected for suspicious behavior, code flow, or malware.

☆ Key Characteristics / Features:

Dependency Walker:

1. Lists all dependent modules (.dll, .ocx, etc.)
2. Detects missing or invalid modules
3. Exports detailed call tree of functions
4. Displays imported/exported functions with parameters
5. Highlights dependency conflicts
6. Supports 32-bit and 64-bit modules
7. Shows delay-load dependencies

dex2jar:

8. Converts .dex to .jar for Java analysis
9. Supports command-line interface

10. Compatible with Android Studio and JD-GUI
11. Useful in reverse engineering APKs
12. Helps analyze obfuscated code
13. Lightweight and open-source
14. Frequently updated with community support
15. Integrates with other tools like jadx, JADX-GUI

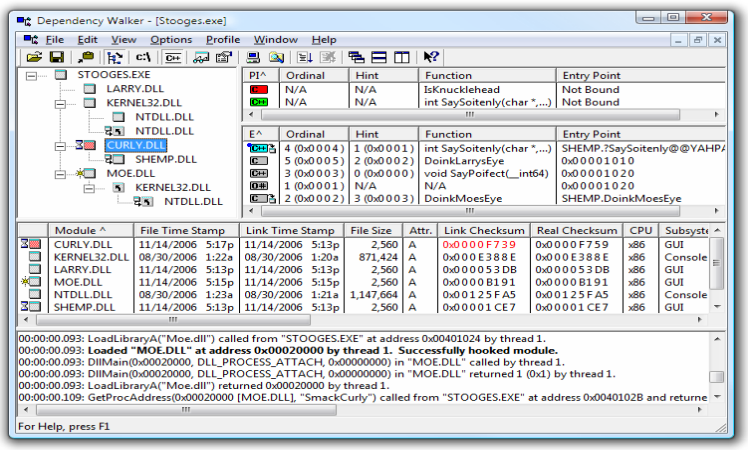
🔗 Types / Modules Available:

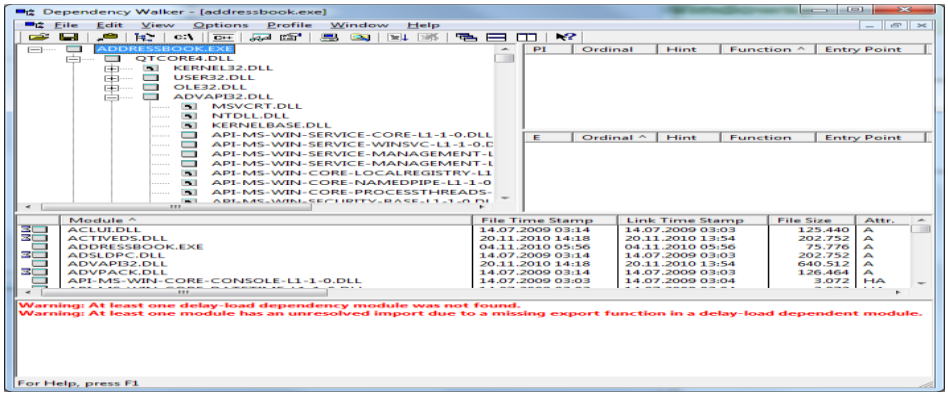
- **Dependency Walker GUI**
- **Dependency Tree Viewer**
- **dex2jar CLI tool**
- **APK -> dex -> jar workflow**
- **Integration with JD-GUI or JADX**

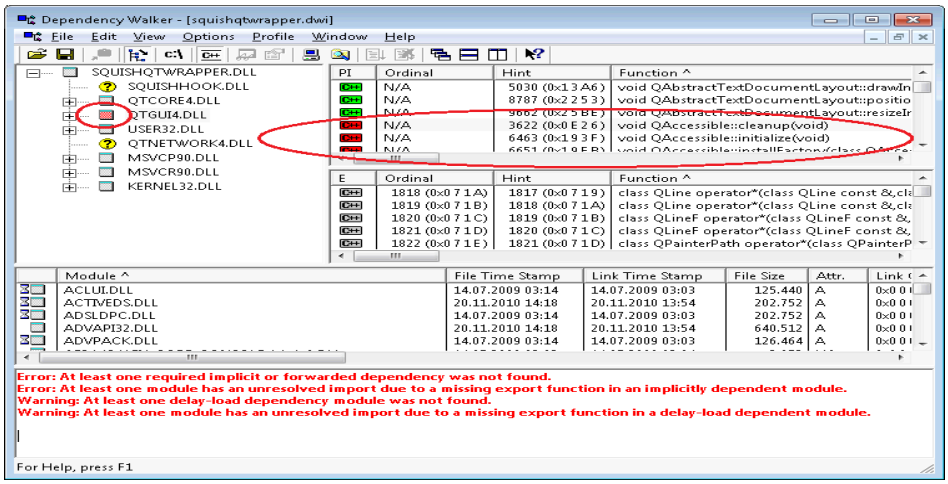
🎯 How Will This Tool Help?

- Investigates suspicious behavior in Windows executables
- Identifies malicious DLL injections or backdoors
- Reconstructs source code from Android APKs
- Analyzes malware hidden in mobile apps
- Understands app behavior without execution
- Supports both desktop and mobile forensics
- Helps trace API misuse or unauthorized access

🖼 Proof of Concept (PoC) Images:







□ 15-Liner Summary:

1. Analyzes Windows binaries (.exe/.dll)
2. Maps dependencies and identifies missing ones
3. Used in malware investigation
4. Useful for static analysis
5. dex2jar converts Android apps for reverse engineering
6. Reveals Java source from APKs
7. Compatible with multiple tools
8. CLI and GUI interfaces
9. Fast and efficient
10. No installation required (portable)
11. Helps detect injected libraries
12. Extracts hidden functionality in apps
13. Useful in IP theft or malware inspection
14. Supports batch processing

15. Frequently used by security researchers

⏱ Time to Use / Best Case Scenarios:

- During binary static analysis (before execution)
- After APK acquisition in mobile forensics
- During malware sandbox evasion checks
- To investigate suspicious software or mobile apps
- Pre-investigation phase of source-level review

🕵 When to Use During Investigation:

- Reverse engineering unknown binaries
- Extracting functionality from suspicious APKs
- Detecting DLL hijacking in Windows
- Understanding data leaks from Android apps
- During intellectual property theft cases
- During black-box testing or software audits

🥷💻 Best Person to Use This Tool & Required Skills:

Best User: Malware Analyst / Mobile App Security Expert / Digital Forensics Investigator

Required Skills:

- Understanding of OS internals (Windows & Android)
- Experience with reverse engineering
- Familiarity with Java and DEX bytecode
- Knowledge of dependency and dynamic link libraries
- Comfort with CLI tools and basic programming

□ Flaws / Suggestions to Improve:

- **Dependency Walker**: No longer maintained officially; limited support for modern Windows APIs
- **dex2jar**: May not handle heavily obfuscated code well
- Needs better integration with modern decompilers
- No built-in GUI for dex2jar

- Limited detection of native libraries inside APKs

✅ **Good About the Tool:**

- Lightweight and easy to use
- Effective for static analysis
- Reveals deep internals of both Windows and Android binaries
- Bridges gap between raw binaries and readable code
- Can be used offline, perfect for air-gapped environments
- Excellent for training and demonstration purposes