

# **Proof-of-Concept Report:-**

## **Tool Name:**

Pe\_unmapper & PE Bear

## **Description:-**

### **PE Bear:**

PE Bear is a lightweight and flexible tool designed for analyzing Windows PE (Portable Executable) files. It provides a clean interface to explore headers, sections, and import/export tables.

### **PE Unmapper:**

pe\_unmapper is a complementary tool or plugin that unmapped memory-loaded PE files (such as packed or injected executables) and reconstructs them into valid, analyzable binaries.

---

## **What Is This Tool About?**

These tools are essential for reverse engineering and malware analysis. PE Bear offers a detailed static view of the PE structure, while pe\_unmapper reconstructs executables from memory dumps, correcting headers, IATs, and entry points, making post-dump binaries useful for further investigation.

---

## **Key Characteristics / Features:**

- Portable, no installation required
- PE header parsing (DOS, NT, Optional, Section)
- Rebuilds memory-resident PE structures
- Entry Point (OEP) recovery
- IAT and section header reconstruction

- Supports x86 and x64 binaries
  - Plugin support (PE Bear)
  - Fast binary analysis with hex editor
  - CLI and GUI available
  - Minimal resource usage
  - Visual section mapping and entropy graphs
  - Integration with debuggers like x64dbg
  - Unpacking support for packed malware
  - Can process malformed or corrupted PE dumps
  - Signature and timestamp inspection
- 

### **Types / Modules Available:**

- PE Header Viewer
  - Section Viewer
  - Hex Editor
  - Section Rebuilder (pe\_unmapper)
  - Import Address Table Fixer
  - Entry Point Finder
  - CLI Dump Tool
- 

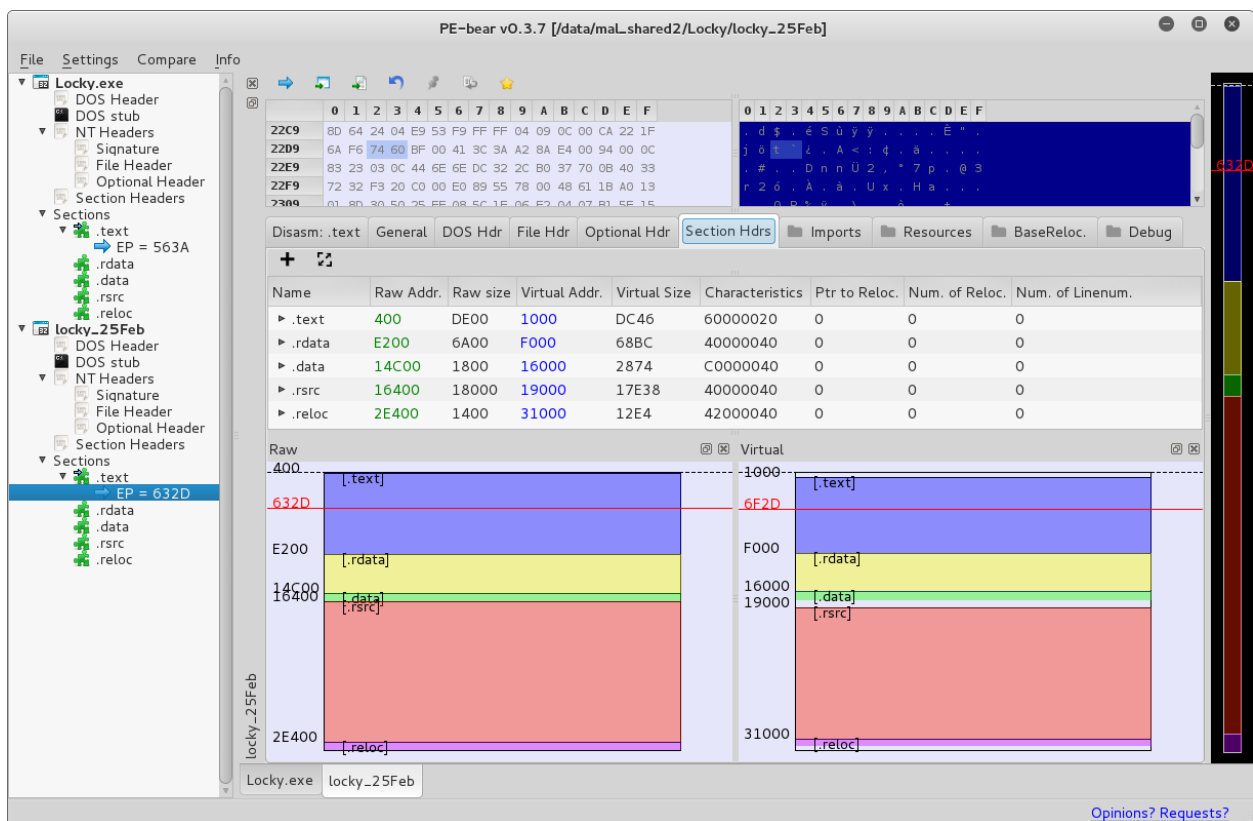
### **How Will This Tool Help?**

- Analyzes malicious binaries and detects packing
- Reconstructs and dumps memory-injected payloads

- Supports reverse engineering in malware analysis
- Prepares executables for loading into IDA Pro or Ghidra
- Useful in forensic memory dump analysis
- Allows section and IAT analysis even after in-memory execution

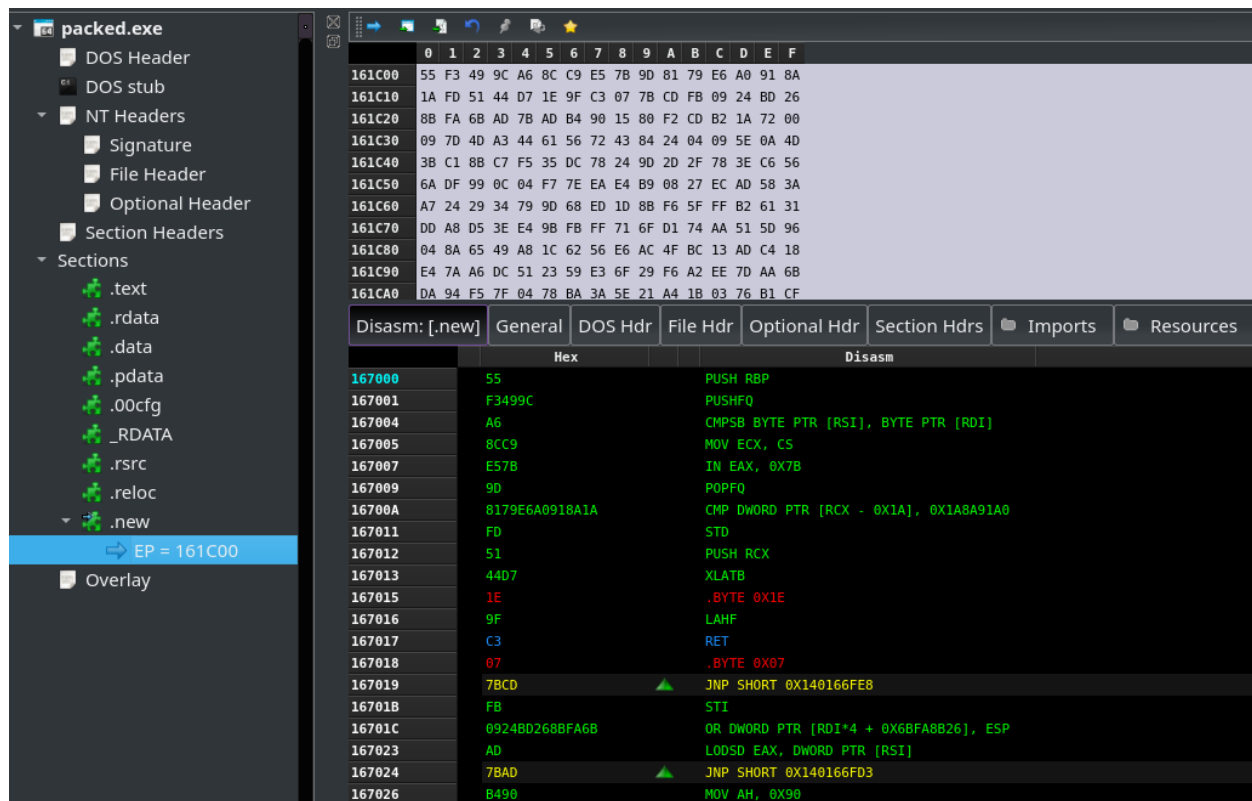
## Proof of Concept (POC) Images:

### PE-Bear:



pe\_unmapper CLI finding and fixing OEP:

<https://github.com/hasherezade/pe-bear/issues/11#issuecomment-1428172722>



## 15-Liner Summary:

- Static PE analysis with PE Bear
- Portable and lightweight
- Memory dump rebuilding with pe\_unmapper
- Entry Point (OEP) recovery
- Import Table reconstruction
- Supports x86/x64 binaries
- CLI automation supported
- Detects packing/obfuscation
- Great for malware RE workflows

- Integrated hex editing
  - Signature and timestamp parsing
  - Entropy-based analysis
  - Plugin extensibility
  - Cross-platform via Wine (for PE Bear)
  - Used by REs, IR teams, and forensic units
- 

#### **Time to Use / Best Case Scenarios:**

- After capturing a memory dump of a suspicious process
  - When static PE fails to load in IDA/Ghidra
  - While analyzing packed or obfuscated malware
  - After process hollowing detection
  - During red-team operations to unpack payloads
- 

#### **When to Use During Investigation:**

- Post-infection memory analysis
  - Reverse engineering of custom loaders
  - Packed malware reconstruction
  - Forensic analysis of malicious binaries
  - Reconstruction before AV/sandbox testing
-

## **Best Person to Use This Tool & Required Skills:**

### **Best Users:**

- Malware Analysts
- Reverse Engineers
- Incident Responders
- Threat Researchers

### **Required Skills:**

- Understanding of PE file structure
  - Familiarity with debuggers (e.g., x64dbg, WinDbg)
  - Experience with memory dump acquisition
  - Optional: scripting with Batch/Python for automation
- 

### **Flaws / Suggestions to Improve:**

- Limited support for .NET or managed PE files
  - No GUI integration for pe\_unmapper (CLI only)
  - Lack of in-tool logging panel (for pe\_unmapper)
  - Suggestion: Add automatic import resolution using debug symbols
  - Suggestion: Integration with Ghidra via API plugin
- 

### **Good About the Tool:**

- Accurate and fast PE parsing
- Excellent for binary repair and unpacking workflows
- Portable with no dependencies

- Helpful for both manual and automated RE workflows
- Continuously updated and well-documented