Unpacking tips and tricks

Samuel Chevet

w4kfu@lse.epita.fr
http://www.lse.epita.fr

12 February 2013

Why this talk ?

- Previously in w4kfu's talk : Anti-Debug
- Malicious software
- Video Games
- Share Reverse Engineering stuff
- Fun

Packer

- Compress executable
- Prepend decompression stub
- Decompression stub is standalone
- Indistinguishable to the casual user
- Single executable
- Unpack and transfer control to it
- Original entry point
- Exist for DOS, Microsoft Windows and others OS
- Command line as GUI based

Packer Benefit

- Less storage space
- Marketing a product via internet
- Less time for data transfer
- Resistant to casual reverser
- Target must be unpacked or rebuilt

Packer Disavantages

- Everything come at a price
- Antivirus problem
- More time to decompress
- Unpacked at some stage
- Dumped to disk ?

Protector

- Derive of the simple packer
- Packer aim to reduce size
- Add code to protect against reverse engineering
- Size will considerably increase
- Malicious software

Detection

- Signature based
- Opcode-sequence-based
- Tag
- Additional heuristics
  - OEP outside first section
  - More than one executable section
  - ImportTable position uncommon
  - LoadLibrary and GetProcAddress in ImportTable
  - TLS
  - Unknow instruction
- Anti Re-Protect
- Compiler startup code

Detection

Anti

- Replace instruction
- Polymorphism
- Metamorphism

Toolz

- PEiD
- Protection ID
- RDG packer Detector
- ...

Import Table

- List of functions not part of the application
- Called imports
- Operating systems DLL's, or homemade
- Different OS Version
- Application don't know where

Import Table

- OptionalHeader->DataDirectory[]
- IMAGE_DIRECTORY_ENTRY_IMPORT

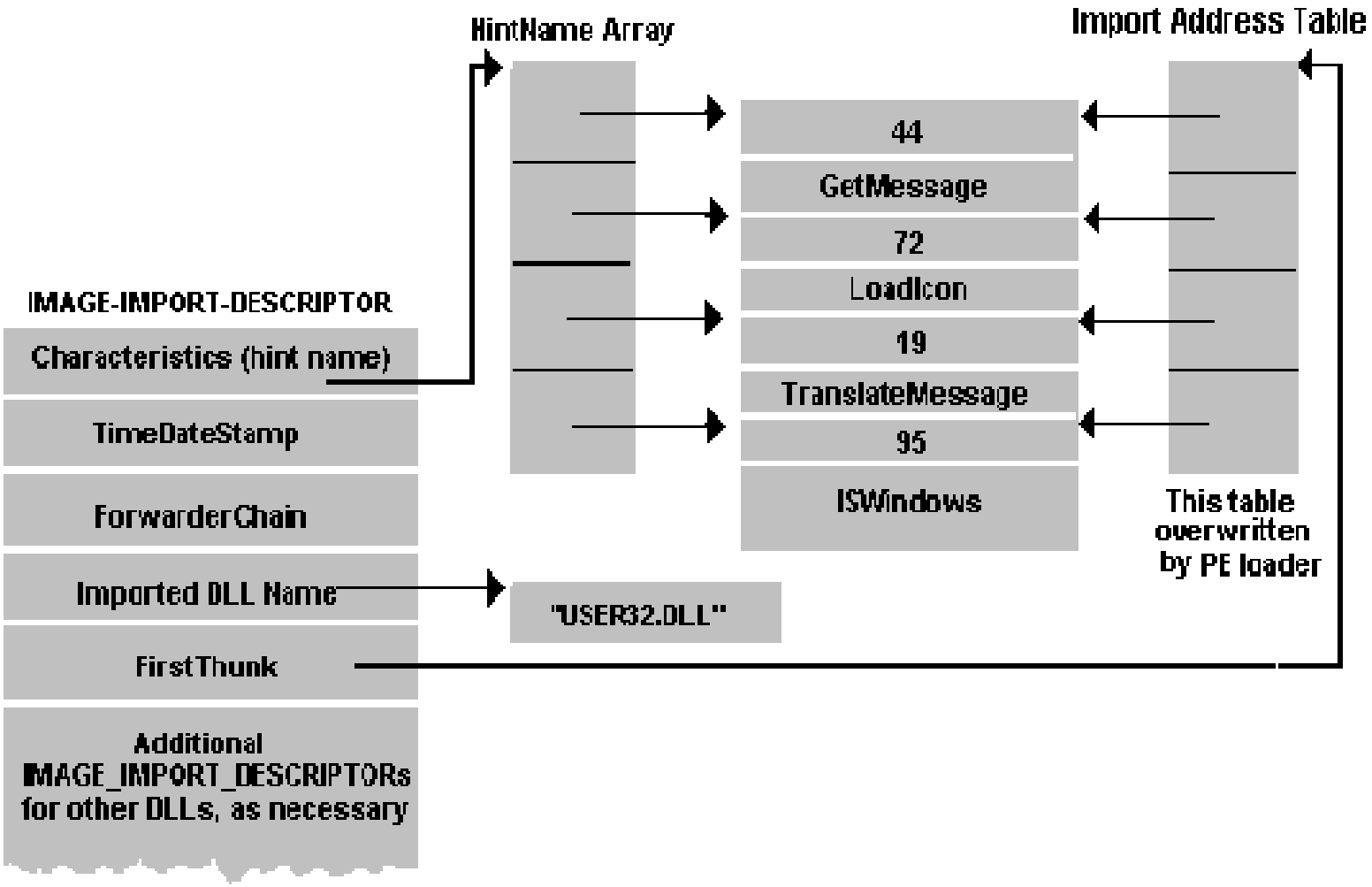
IMAGE_IMPORT_DESCRIPTOR

- OriginalFirstThunk
- TimeDateStamp
- ForwarderChain
- Name
- FirstThunk

Import Address Table

- Loader loads DLL
- Construct IAT
- All ptr in FirstThunk contain API's address
- call [addr], jmp [addr]

Peering inside the PE



Original OEP

- Trace the code
- ESP trick
- VirtualProtect()
- Use Exceptions

Fix PE

- Offset OEP
- Offset IAT
- Sections characteristics
- And more when there is some protection

Import rebuilding

- Packers/Protector destroy Import Table
- Correct RVA and Size of Import Table
- IMAGE_IMPORT_DESCRIPTOR nulled one
- OriginalFirstThunk, FirstThunk and Name must be well informed

Anti-Dumping

- Mutex
- CPUID
- Delete loader
- Header modification
- Page level protection

TLS Callbacks

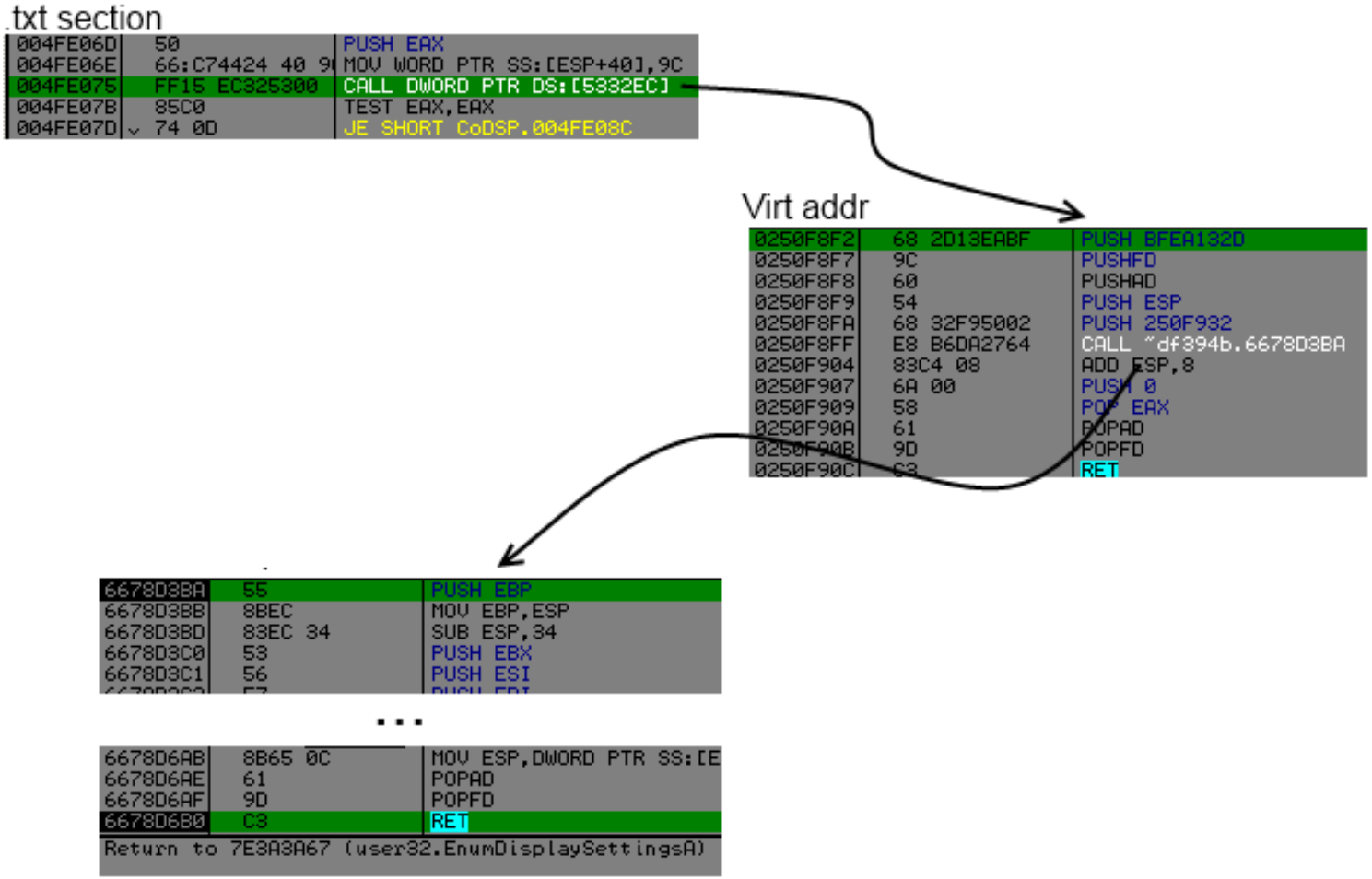
- Thread Local Storage
- Execute code before EP
- Debugger detection
- Decryption routines
- Hook

Stolen Bytes

- Portions of code
- Removed from original
- Usually near entry point
- Executed from allocated memory
- Restore them before dump

API Redirection

- IAT partially or completely destroyed
- Call to APIs are redirected
- Routines located into allocated memory or in protector stub



API Redirection

- Stolen instructions
- Control transfered back in the middle
- Routines located into allocated memory or in protector stub
- Load whole DLL image
- Redirect API
- Difficult to set breakpoints

API Redirection

- Inject !!!
- Scan for call dword ptr / jmp dword ptr
- Is outside PE ?
- Is not an API ?
- Hook routine
- Call it
- Use against himself !

Nanomites

- JCC instruction
- Some Opcodes
- Replace by int3
- 2 Process ! Father and son
- Inject the father
- Father : WaitForDebugEvent()
- Son : Scan 0xCC (int3)
- Reverse, or comportemental study
- Thruth table
- Maybe opcode will be restored to avoid performance down

Triggers

- Detect if protection has been deleted
- Developpers can use SDK
- Invincible enemy
- Camera bug
- Redirect call will return on the next instruction
- Return value modification

Conclusion

- INJECT !

Real Conclusion

- Really fun !
- Code your own toolz
- Don’t use unpacker ! Write your own
- Internet connection permanent
- Kill market of multimedia library and occasion

Questions ?

Thank you for your attention

- @w4kfu
- blog.w4kfu.com
- w4kfu@lse.epita.fr