

Unpacking tips and tricks

Samuel Chevet

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

12 February 2013

Why this talk ?

Unpacking tips
and tricks

Samuel Chevet

Presentation

Why this talk ?

Packer

Protector

Detection

Basics knowledges

Import Table

Import Address Table

Process

Protector
Techniques

Conclusion

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

- 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

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

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

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

- 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

Anti

- Replace instruction
- Polymorphism
- Metamorphism

Toolz

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

Presentation

Why this talk ?

Packer

Protector

Detection

Basics knowledges

Import Table

Import Address Table

Process

Protector

Techniques

Conclusion

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

- OptionalHeader->DataDirectory[]
- IMAGE_DIRECTORY_ENTRY_IMPORT

IMAGE_IMPORT_DESCRIPTOR

- OriginalFirstThunk
- TimeDateStamp
- ForwarderChain
- Name
- FirstThunk

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

Peering inside the PE

Presentation

Why this talk ?

Packer

Protector

Detection

Basics knowledges

Import Table

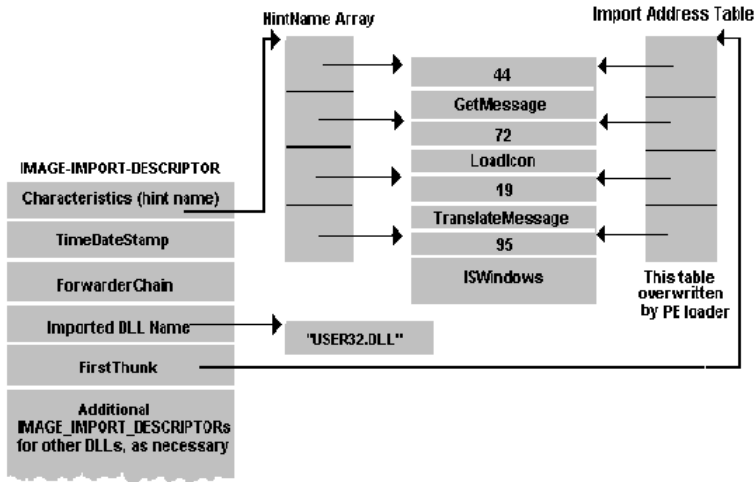
Import Address Table

Process

Protector

Techniques

Conclusion



Unpacking tips
and tricks

Samuel Chevet

Presentation

Process

Original OEP

Fix PE

Import rebuilding

Protector
Techniques

Conclusion

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

Unpacking tips
and tricks

Samuel Chevet

Presentation

Process

Original OEP

Fix PE

Import rebuilding

Protector
Techniques

Conclusion

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

- 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

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

Unpacking tips
and tricks

Samuel Chevet

Presentation

Process

Protector
Techniques

Anti-Dumping

TLS Callbacks

Stolen Bytes

API Redirection

Nanomites

Triggers

Conclusion

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

- 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

txt section

004FE06D	58	PUSH EAX
004FE06E	66:C74424 40 9	MOV WORD PTR SS:[ESP+40],9C
004FE075	FF15 EC325300	CALL DWORD PTR DS:[5332EC]
004FE07B	85C0	TEST EAX,EAX
004FE07D	74 00	JE SHORT C603P,004FE08C

Virt addr

0250F8F2	68 2013E8BF	PUSH DWORD 000
0250F8F7	9C	PUSHFD
0250F8F8	60	PUSHAD
0250F8F9	54	PUSH ESP
0250F8FA	68 32F95002	PUSH 250F932
0250F8FF	E8 B60A2764	CALL "df394b,6678D3BA
0250F904	83C4 08	ADD ESP,8
0250F907	6A 00	PUSH 0
0250F909	58	POP EAX
0250F90A	61	POPAD
0250F90E	9D	POPFD
0250F90C	EB	RET

6678D3BA	55	PUSH EBP
6678D3BB	8BEC	MOV EBP,ESP
6678D3BD	83EC 34	SUB ESP,34
6678D3C0	53	PUSH EBX
6678D3C1	56	PUSH ESI
6678D3C2	57	PUSH EDI

...

6678D6A8	8B65 0C	MOV ESP,DWORD PTR SS:[E
6678D6AE	61	POPAD
6678D6AF	9D	POPFD
6678D6B0	C3	RET

Return to 7E3A3A67 (user32.EnumDisplaySettingsA)

Unpacking tips
and tricks

Samuel Chevet

Presentation

Process

Protector
Techniques

Anti-Dumping

TLS Callbacks

Stolen Bytes

API Redirection

Nanomites

Triggers

Conclusion

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

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

- 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

Unpacking tips
and tricks

Samuel Chevet

Presentation

Process

Protector
Techniques

Anti-Dumping

TLS Callbacks

Stolen Bytes

API Redirection

Nanomites

Triggers

Conclusion

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

Unpacking tips
and tricks

Samuel Chevet

Presentation

Process

Protector
Techniques

Conclusion

- INJECT !

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

Thank you for your attention

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