Qualitative and Quantitative Evaluation of Software Packers

María Bazús, **Ricardo J. Rodríguez**, José Merseguer

rjrodriguez@unizar.es ※ @RicardoJRdez ※ www.ricardojrodriguez.es

Universidad Zaragoza

Department of Computer Science and Systems Engineering
University of Zaragoza, Spain

December 12, 2015

NoConName 2015
Barcelona (Spain)

$whoami

- Ph.D. on Comp. Sci. (Univ. of Zaragoza, Spain) (2013)
- Assistant Professor at University of Zaragoza
  - Performance analysis on critical, complex systems
  - Secure Software Engineering
  - Advance malware analysis
  - RFID/NFC Security
- Not prosecuted ⌣̈
- Speaker at NcN, HackLU, RootedCON, STIC CCN-CERT, HIP, MalCON, HITB...

Universidad Zaragoza
1542

Agenda

Universidad Zaragoza
1542

Agenda

Universidad Zaragoza 1542

Introduction (I): Reverse Engineering

WTF?

- To analyse a binary program with machine-code vision
- Types of analysis:
  - Static analysis ($\Downarrow$ not executed, $\Uparrow$ all paths explored)
  - Dynamic analysis ($\Uparrow$ truly executed, $\Downarrow$ but just one path explored!)

Universidad Zaragoza
1542

Introduction (I): Reverse Engineering

WTF?

- To analyse a binary program with machine-code vision
- Types of analysis:
  - Static analysis ($\Downarrow$ not executed, $\Uparrow$ all paths explored)
  - Dynamic analysis ($\Uparrow$ truly executed, $\Downarrow$ but just one path explored!)

RE uses: legitimate and illegitimate

- ✓ Find software bugs
- ✓ Get interoperability with legacy systems
- ✓ Detect malicious software
- X Detect vulnerabilities to create/spread malware
- X Obtain (or avoid) software license duplication

Universidad Zaragoza 1542

Introduction (II): RE Tools

Static analysis

- Disassemblers
- Decompilers

Dynamic analysis

- Debuggers
  - Trace execution
  - Breakpoints
  - View internal data
- Dumpers

Universidad Zaragoza 1542

Introduction (III): Anti-RE Techniques

Definition, pros and cons of software packers

- Avoidance techniques for static and dynamic analysis into binaries
  - Make RE tasks harder
  - On the contrary, they have a strong impact on binary performance: execution time, memory consumption

Universidad Zaragoza
1542

Introduction (III): Anti-RE Techniques

Definition, pros and cons of software packers

- Avoidance techniques for static and dynamic analysis into binaries
  - Make RE tasks harder
  - On the contrary, they have a strong impact on binary performance: execution time, memory consumption

They are used for...

- Binary protection before distribution (to keep the intellectual (?) property)
- Avoid a malware to be positively detected as malicious by an anti-virus

Universidad Zaragoza
1542

Introduction (IV): Software Packers

Software packer: What is it?

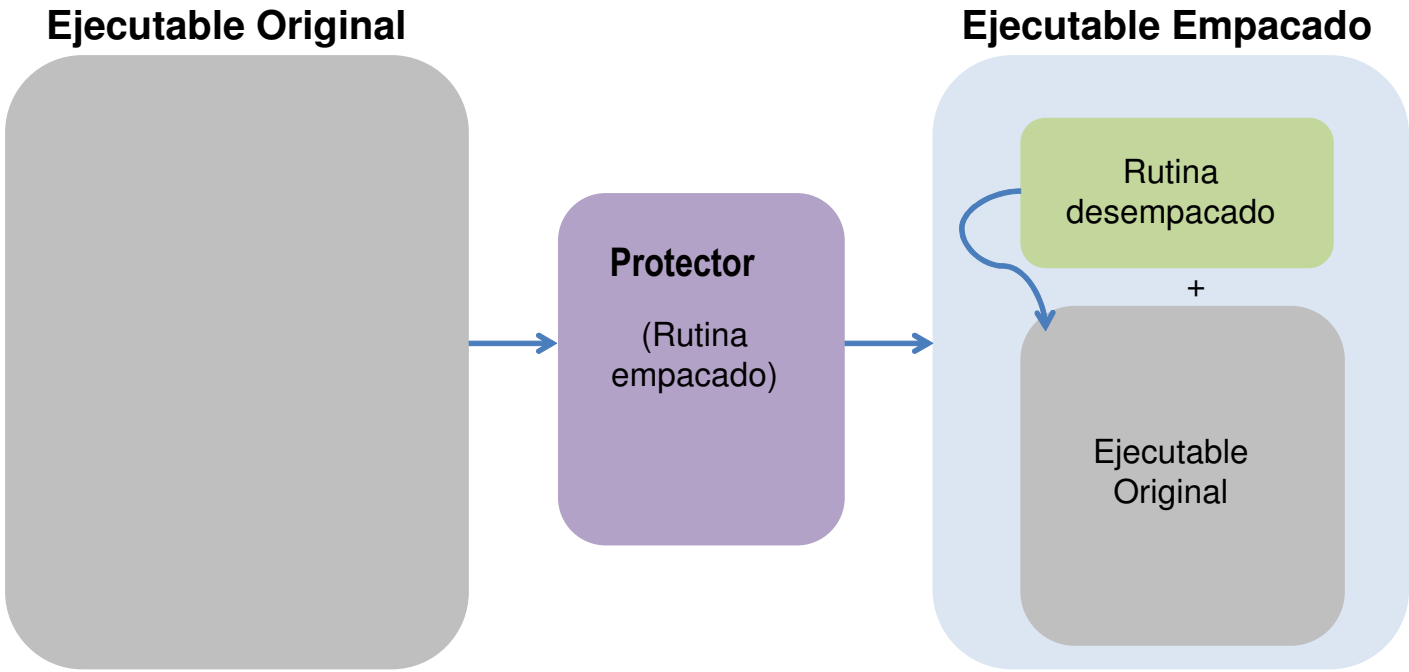
- Tools for binary protection (legitimately)
- Once upon a time. . . : just compressors
- They evolve to protectors, including anti-RE techniques
- Normally used in Windows environments

Universidad Zaragoza 1542

Introduction (V): Software Packers

How do they "protect" a binary?

- Packed executable = Original executable + unpacking routine

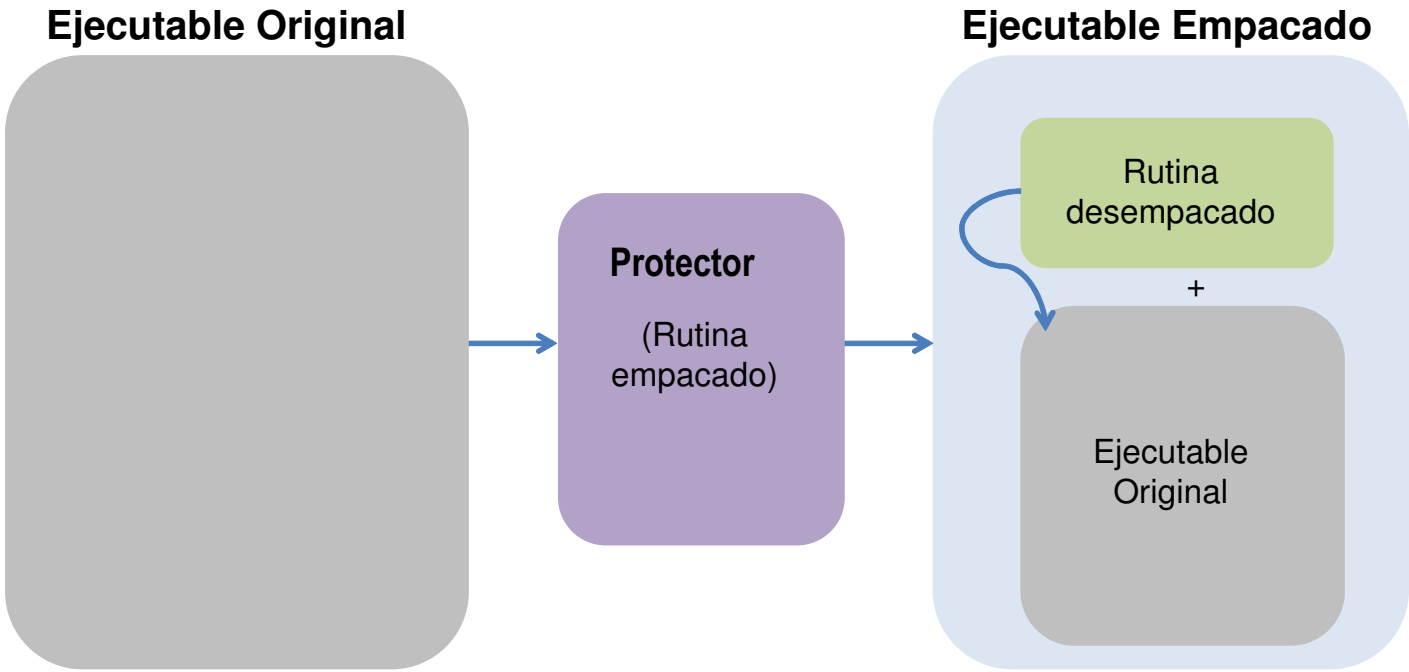


Universidad Zaragoza
1542

Introduction (V): Software Packers

How do they "protect" a binary?

- Packed executable = Original executable + unpacking routine



How are they reversed?

- Find unpacking routine end ⇒ dump binary from memory to disk!

Introduction (VI): Main Goals

Analysing a bunch of software packers for...

- Create a taxonomy of protection techniques
- Create a benchmark for testing
- Evaluate the selected packers:
  - Qualitatively: protection strength
  - Quantitatively: reliability and performance

Universidad Zaragoza
1542

Agenda

Universidad Zaragoza 1542

Related Work

- $\nexists$ comparative analysis of current software packers
- Automation of malware unpacking ([RHD+06], [KPY07], [MCJ07],[GFC08], [JCL+10])
- Using DBI to analyse malware ([RAG16])
- Closest work: performance of software packers in Linux embedded systems ([KLC+10])

Contributions

- Taxonomy of software packers
- Current software packers evaluation
  - Qualitatively: protection strengths
  - Quantitatively: reliability and performance (exec. time, memory consumption, binary size)

Universidad Zaragoza 1542

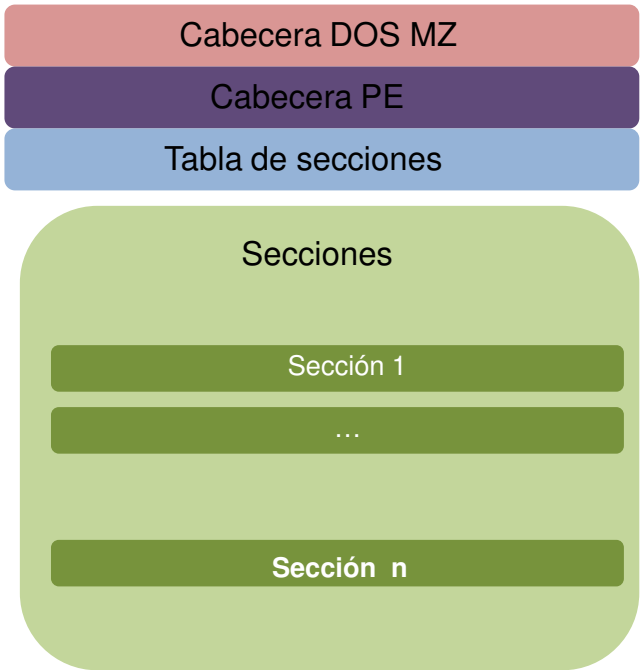
Agenda

Universidad Zaragoza 1542

Previous Concepts (I): PE Header

What is it?

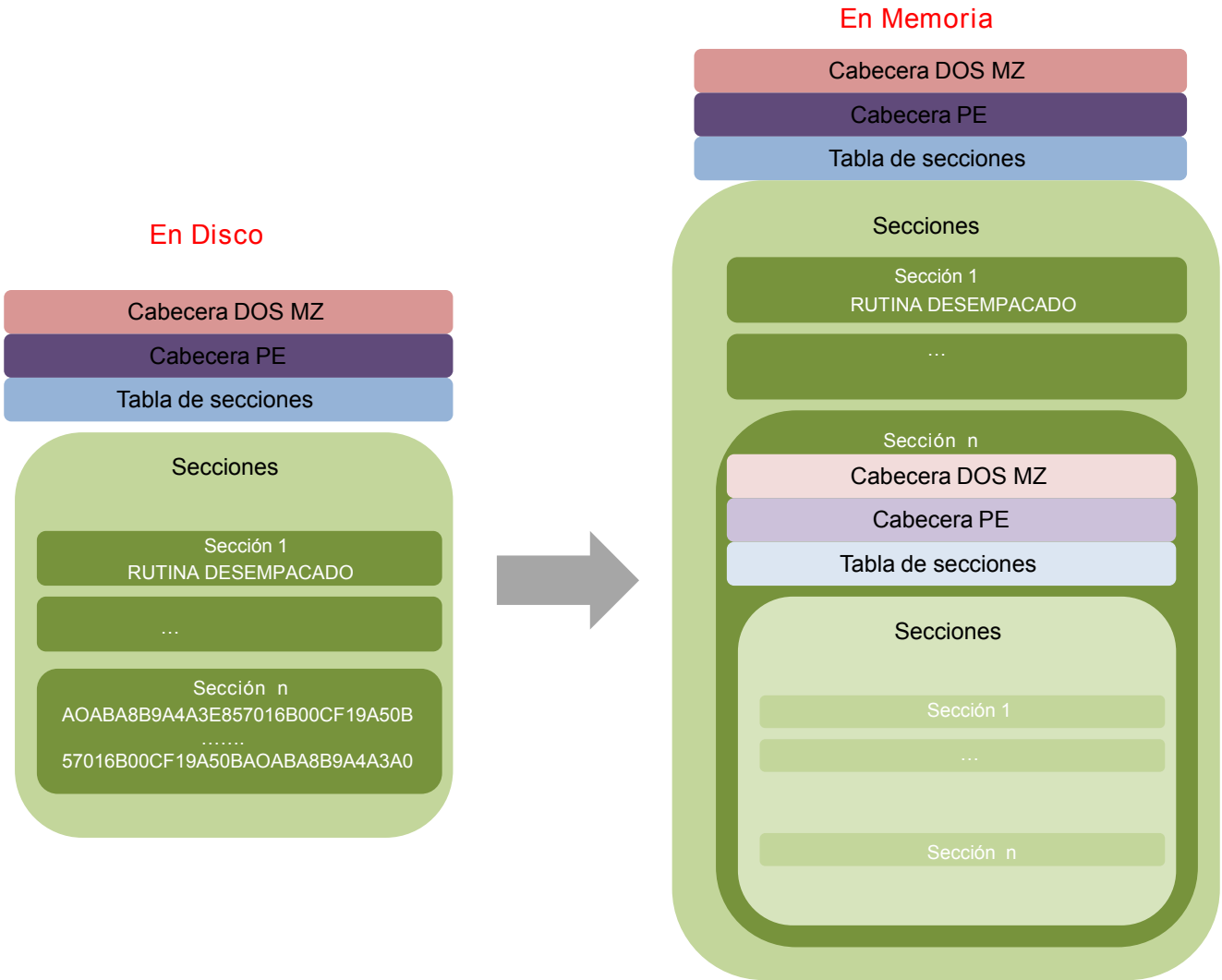
- Windows standard format for binaries (.exe, .dll, .sys, . . .)
- Header (characteristics) + Sections (data & code)



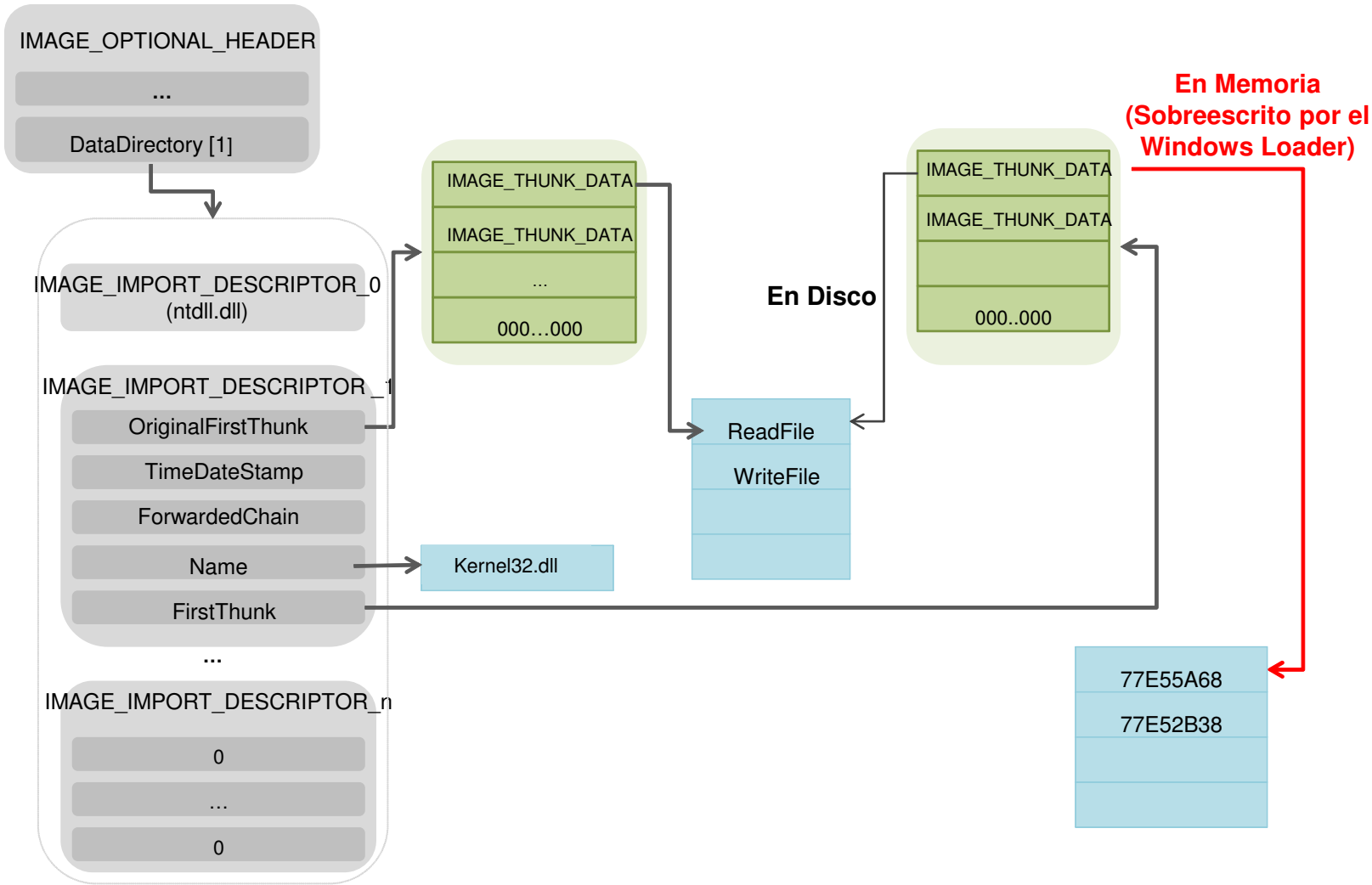
- DOS Header:
  - *e_lfnew* offset to PE Header
- PE Header:
  - *ImageBase*
  - *AddressOfEntryPoint*
  - *DataDirectory[1]*

Universidad Zaragoza
1542

Previous Concepts (II): Loading Process



Previous Concepts (III): Import Address Table



Universidad Zaragoza
1542

Agenda

Universidad Zaragoza 1542

Software Protection (I): Static Analysis

Copy protection

- Main aim: Avoid to illegitimately copy a software
- Techniques: License management, register keys, hardware dongles

Universidad Zaragoza
1542

Software Protection (I): Static Analysis

Copy protection

- Main aim: Avoid to illegitimately copy a software
- Techniques: License management, register keys, hardware dongles

Compression and Encryption

- Main aim: Avoid static analysis of code
- Techniques: to compress or to encrypt the original code (+ resource compression)

Universidad Zaragoza 1542

Software Protection (I): Static Analysis

Copy protection

- Main aim: Avoid to illegitimately copy a software
- Techniques: License management, register keys, hardware dongles

Compression and Encryption

- Main aim: Avoid static analysis of code
- Techniques: to compress or to encrypt the original code (+ resource compression)

Anti-disassembling

- Main aim: Make disassembling process quite harder
- Techniques: Add junk code, dummy code, or code permutations

Universidad Zaragoza 1542

Software Protection (II): Dynamic Analysis (1)

Anti-Debugging

- Main aim: Avoid to debug a binary
- Techniques:
  - Windows API-based: Using Windows internals functions
  - Process and threads: Using internal info about the process or thread in execution
  - Hardware registers: Using CPU registers
  - Timing: Measuring latency between ins. or other time issues
  - Exception-based: Based on the exception handling behaviour

Universidad Zaragoza 1542

Software Protection (II): Dynamic Analysis (2)

Anti-Dumping

- Main aim: Avoid a dump process to properly execute
- Techniques:
  - Import Table obfuscation
  - Nanomites: Substitute JX/JMP by INT3 assembler ins.
  - Stolen Bytes
  - Guard Pages
  - Virtualization: Rewriting of binary code to be executed by an embedded VM. Note that original binary code will never be allocated in memory!

Universidad Zaragoza 1542

Software Protection (II): Dynamic Analysis (3)

Anti-VM & Anti-Patching

Anti-VM

- Main aim: Avoid execution in VMs
- Techniques: a lot... (ask to A. Ortega, I. Rodríguez or... read our awesome two-columns paper at IEEE LATAM$\ddot{\smile}$)

Universidad Zaragoza
1542

Software Protection (II): Dynamic Analysis (3)

Anti-VM & Anti-Patching

Anti-VM

- Main aim: Avoid execution in VMs
- Techniques: a lot... (ask to A. Ortega, I. Rodríguez or... read our awesome two-columns paper at IEEE LATAM$\ddot{\smile}$)

Anti-Patching

- Main aim: Avoid alteration of binary code
- Techniques: Compute and check CRC integrity (or other hash function)

Universidad Zaragoza 1542

Agenda

Universidad Zaragoza 1542

Software Packers Under Study

Packer	Version	Last release	License	Type
ACProtect	2.1.0	09/2008	149$	Compressor&Protector
Armadillo	9.62	06/2013	aprox. 299$	Compressor&Protector
ASPack	2.32	08/2012	34€- 208€	Compressor&Protector
ASProtect	1.69	09/2013	117€- 347€	Compressor&Protector
Enigma	3.7	06/2013	149$- 299$	Protector&Virtualizer
EXE Stealth	4.19	06/2011	296,31€	Compressor&Protector
ExeCryptor	2.3.9	03/2006	135$- 749$	Compressor&Protector
ExPressor	1.8	01/2010	14$- 360$	Compressor&Protector
FSG	2.0	05/2004	Free	Compressor
MEW	11	11/2004	Free	Compressor
Obsidium	1.5	10/2013	139€- 289€	Protector&Virtualizer
PECompact	3.02.2	02/2010	89,95$499$	Compressor&Protector
PELock	1.06	01/2012	89$- 289$	Compressor&Protector
PESpin	1.33	05/2011	Free	Compressor&Protector
Petite	2.3	02/2005	170€	Compressor&Protector
Smart Packer	1.9	06/2013	59€- 399€	Compressor&Bundler
TeLock	0.98	08/2001	Free	Compressor&Protector
Themida	2.2	10/2013	199€- 399€	Protector&Virtualizer
UPX	3.91	02/2013	Free	Compressor
VMProtect	2.13	05/2013	99$- 399$	Protector&Virtualizer
YodasProtector	1.03.3	08/2006	Gratuita	Compressor&Protector

Universidad Zaragoza 1542

Agenda

Universidad Zaragoza 1542

Qualitative Evaluation (I): Anti-Static Analysis

Software Packer	Compression	Encryption	Anti-Disassembling
ACProtect	✓	–	✓
Armadillo	✓	✓	✓
ASPack	✓	✓	–
ASProtect	✓	✓	✓
Enigma	✓	✓	✓
EXE Stealth	✓	–	–
ExeCryptor	✓	–	✓
ExPressor	✓	–	–
FSG	✓	–	–
MEW	✓	✓	–
Obsidium	✓	✓	✓
PECompact	✓	–	–
PELock	✓	–	–
PESpin	✓	✓	✓
Petite	✓	–	–
SmartPacker	✓	✓	–
TeLock	✓	–	–
Themida	✓	✓	✓
UPX	✓	–	–
VMProtect	✓	–	✓
Yodas Protector	✓	–	✓

Universidad Zaragoza

Qualitative Evaluation (II): Anti-Dynamic Analysis

Protector	Anti-Debugging	Anti-Dumping	Anti-VM	Anti-Patching
ACProtect	✓	✓	–	–
Armadillo	✓	✓	–	✓
ASPack	✓	–	–	–
ASProtect	✓	✓	–	✓
Enigma	✓	✓(High)	✓	✓
EXE Stealth	✓	✓	✓	✓
ExeCryptor	✓(High)	✓(High)	✓	✓
ExPressor	✓	✓	✓	✓
FSG	–	–	–	–
MEW	–	–	–	–
Obsidium	✓(High)	✓(High)	✓	✓
PECompact	✓	✓	–	✓
PELock	✓	✓	–	–
PESpin	✓	✓	–	–
Petite	✓	✓	–	–
SmartPacker	–	–	–	–
TeLock	✓	✓	–	–
Themida	✓(Alto)	✓(High)	✓	✓
UPX	–	–	–	–
VMProtect	✓	✓(High)	✓	✓
Yodas Protector	✓	✓	–	–

Universidad Zaragoza 1542

Quantitative Evaluation (I): Benchmark Creation

Name	Version	Language	Type	Computatic type	Origin
GNU go	3.8	C	AI - Games	Integer	SPEC CINT 2006
hmmer	3.0	C	Genetics	Integer	SPEC CINT 2006
h264ref	18.5	C	Video compression	Integer	SPEC CINT 2006
mcf	1.3	C	Combinatorial	Integer	SPEC CINT 2006
namd	2.8	C++	Biology, molecular simulation	Float	SPEC CFP 2006
povray	3.7	C++	Renderization	Float	SPEC CFP 2006
calculix	2.6	Fotran90 & C	Structural mechanics	Float	SPEC CFP 2006
palabos	1.4.	C++	Fluid dynamics	Float	Own
aescrypt	3.0.9	C	Cryptography	I/O	Own
bzip2	1.0.5	C	Compression	I/O	SPEC CINT 2006
ffmpeg	0.10	C	Video/audio format conversion	I/O	Phoronix Test Suite
md5	1.2	Delphi 7	Cryptography, Hash	I/O	Own

Universidad Zaragoza 1542

Quantitative Evaluation (II): Benchmark Algorithm

```
1: app = [bzip2, gnugoo, hmmer, h264ref, mcf, namd, povray...]
2: packer = [ACProtect, ASPack, Armadillo, Enigma, EXEStealth, ...]
3: for nRep = 1 to 45 do
4:     for app do
5:         execute app {Original binary}
6:         get process measurements
           GetProcessTimes(); GetProcessMemoryInfo()
7:     end for
8:     for packer do
9:         for app do
10:            execute app.packer {Packed binary}
11:            get process measurements
               GetProcessTimes(); GetProcessMemoryInfo()
12:        end for
13:    end for
14: end for
```

Universidad Zaragoza 1542

Quantitative Evaluation (III): Reliability (1)

	Integer comp.	Float comp.	High I/O	Total
ACProtect	25%	50%	75%	50%
Armadillo	50%	100%	75%	75%
ASPack	100%	100%	75%	91%
ASProtect	100%	100%	75%	91%
Enigma	75%	75%	100%	83%
EXE Stealth	100%	75%	100%	91%
ExeCryptor	25%	50%	0%	25%
ExPressor	100%	100%	100%	100%
FSG	0%	50%	100%	50%
MEW	100%	50%	100%	50%
Obsidium	100%	100%	100%	100%
PECompact	25%	100%	100%	75%
PELock	25%	100%	75%	66%
PESpin	0%	50%	75%	41%
Petite	50%	25%	50%	41%
Smart Packer	50%	75%	100%	75%
TeLock	25%	50%	50%	41%
Themida	25%	100%	100%	75%
UPX	100%	100%	100%	100%
VMProtect	100%	100%	100%	100%
YodasProtector	25%	25%	50%	33%
	61%	**76%**	**83%**	

Universidad Zaragoza

Quantitative Evaluation (III): Reliability (2)

Some remarks

- Obsidium, EXEStealth
  - Protected binaries given by the companies
- FSG, MEW, PESPin, Petite, TELock, YodasProtector
  - "*Runtime error: R6002 floating point support not loaded*" ⇒ Reported problem with binaries compiled with MSVC++5 o greater
- ACProtect, Armadillo, PELock, PESpin
  - Correct binary without all protections available
- VMProtect, Themida, EXECryptor, Armadillo y PELock
  - Demo version tool used :(

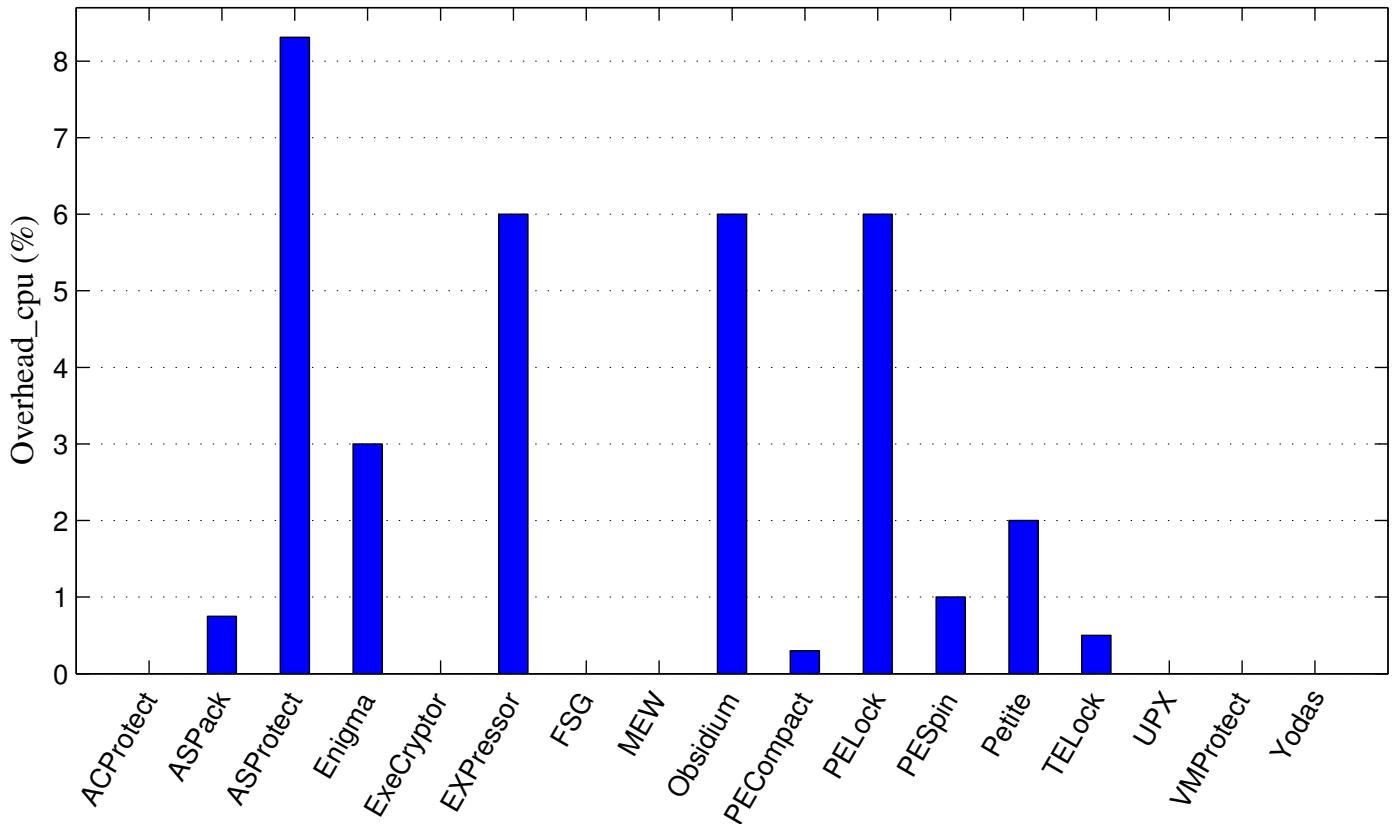
Universidad Zaragoza
1542

Quantitative Evaluation (IV): Performance

CPU Overhead

0% **UPX,FSG, MEW** (just compressors). **EXECryptor, VMProtect** (demo version). **ACProtect, YodasProtector**

6-8% ASProtect, EXPressor, Obsidium, PELock



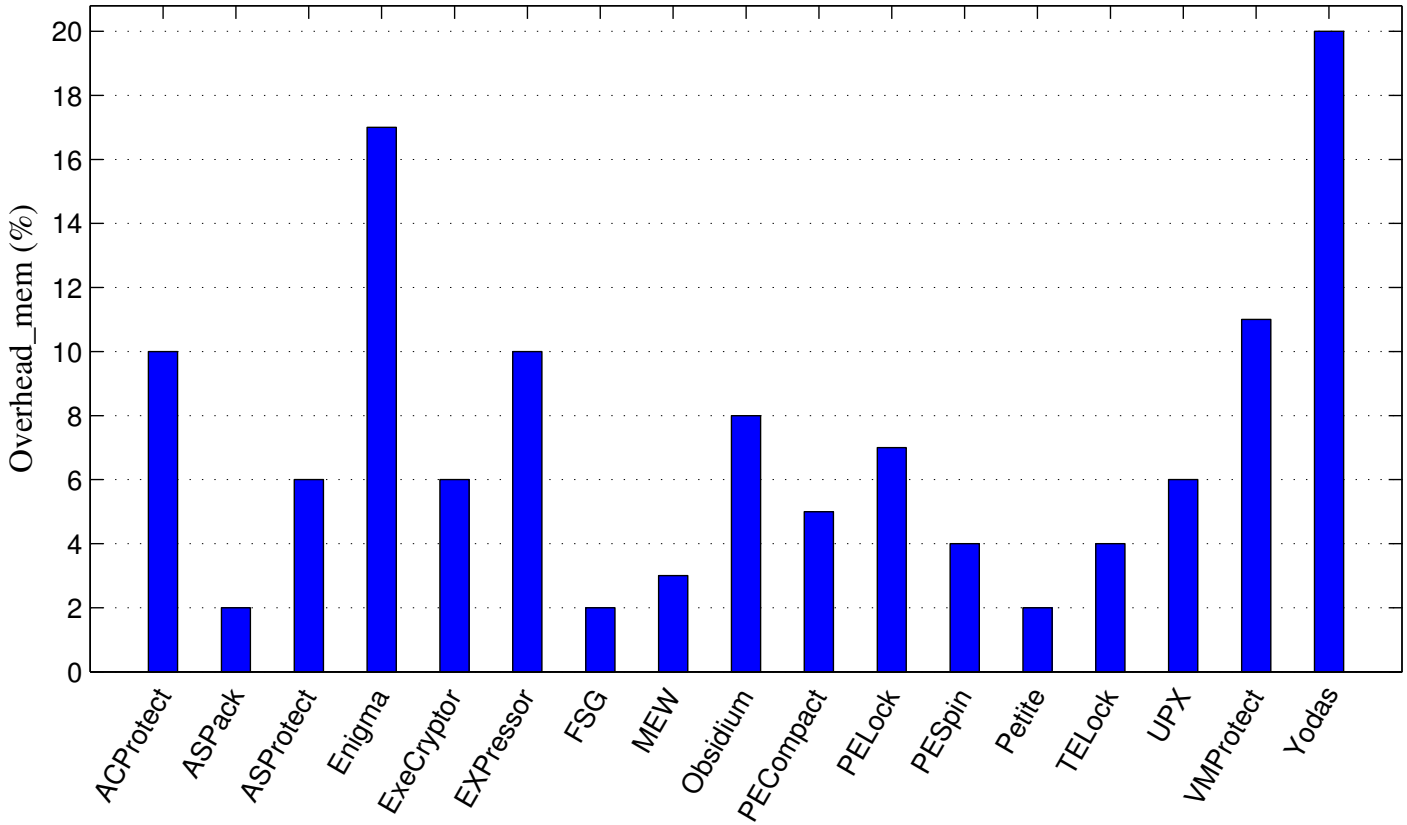
Quantitative Evaluation (V): Performance

Memory Overhead

< 6% **ASPack, FSG, MEW, PECompact, PESpin, Petite, TELock**

16% Enigma (virtualization)

20% Yodas (no CPU Overhead!)



Quantitative Evaluation (VI): Binary Size

Binary Size over Software Types – without checking binary execution

Packer	Integer	Samples	Float	Samples	High I/O	Samples
ACProtect	−4.45%	4	−49.67%	4	145.60%	4
Armadillo	545.89%	4	−1.48%	4	2512.54%	4
ASPack	−75.28%	3	−73.30%	4	−42.35%	4
ASProtect	−59.88%	3	−67.04%	4	132.00%	4
Enigma	160.17%	4	−42.38%	4	909.41%	4
EXE Stealth	166.96%	4	−52.25%	4	955.58%	4
ExeCryptor	−20.96%	4	−59.06%	4	19.99%	4
ExPressor	−62.31%	4	−73.37%	4	−2.01%	4
FSG	−70.43%	2	−67.36%	3	−48.22%	4
MEW	−70.45%	2	−74.31%	3	−54.02%	4
Obsidium	−11.01%	4	−68.32%	4	223.23%	4
PECompact	−73.12%	3	−74.10%	4	−48.62%	4
PELock	−66.78%	3	−62.11%	4	161.82%	3
PESpin	−54.37%	4	−57.88%	4	2.94%	4
Petite	−69.68%	3	−80.50%	3	−62.47%	4
Smart Packer	2653.04%	4	19.16%	3	16162.21%	4
TeLock	−51.08%	2	−66.58%	3	−24.05%	3
Themida	238.13%	4	−49.29%	2	1277.15%	3
UPX	−67.39%	3	−65.58%	4	−47.71%	4
VMProtect	247.70%	4	23.87%	4	1099.84%	4
YodasProtector	−12.50%	1	−53.28%	2	2.18%	3

Universidad Zaragoza

Agenda

Universidad Zaragoza 1542

Conclusions and Future Work (I)

Work Summary

- 252 binaries analysed (21 packers over 12 benchmark binaries)
- Evaluation of protection strength, reliability, performance
- Full report (freely!) available at

  `http://webdiis.unizar.es/~ricardo/files/PFCs-TFGs/Analisis.Rendimiento.Protectores/Memoria_PFC_AnalisisProtectoresSw.pdf`

Universidad Zaragoza 1542

Conclusions and Future Work (I)

Work Summary

- 252 binaries analysed (21 packers over 12 benchmark binaries)
- Evaluation of protection strength, reliability, performance
- Full report (freely!) available at

  `http://webdiis.unizar.es/~ricardo/files/PFCs-TFGs/Analisis.Rendimiento.Protectores/Memoria_PFC_AnalisisProtectoresSw.pdf`

Conclusions

- Packers make RE tasks harder but DO NOT TOTALLY AVOID them
- Disadvantages:
  - The longer the packer is out there, the less effective → unprotection techniques quickly spread
  - Instability of certain software packers with given binaries
  - High performance overhead (in terms of memory and execution time)
  - False positives by some anti-virus products

1542

Conclusions and Future Work (II)

Future work

- Extend to more software packers
- Extend the benchmark binaries
  - Other measures?
- Extend to other OS
- Evaluate full versions of certain software packers (Armadillo, EXECryptor, Themida, PELock, VMProtect)

Universidad Zaragoza
1542

Conclusions and Future Work (II)

Future work

- Extend to more software packers
- Extend the benchmark binaries
  - Other measures?
- Extend to other OS
- Evaluate full versions of certain software packers (Armadillo, EXECryptor, Themida, PELock, VMProtect)

Acknowledgements

- María Bazús, a really impressive student
- ♥ NcN staff
- All you for hearing me!

Universidad Zaragoza 1542

Conclusions and Future Work (III)

L-sponsored slide

Telegram chat: `https://goo.gl/4qDsEh`

(`https://telegram.me/joinchat/AHZySjwVA4tcRpHOCrxdGQ`)

References

- **[RHD+06]** Paul Royal, Mitch Halpin, David Dagon, Robert Edmonds & Wenke Lee. **PolyUnpack: Automating the Hidden-Code Extraction of Unpack-Executing Malware**. *Georgia Institute of Technology*, 2006.
- **[KPY07]** Min Gyung Kang, Pongsin Poosankam & Heng Yin. **Renovo: A Hidden Code Extractor for Packed Executables**. Carnegie Mellon University, 2007.
- **[MCJ07]** Lorenzo Martignoni, Mihai Christodorescu & Somesh Jha. **OmniUnpack: Fast, Generic, and Safe Unpacking of Malware**. In Proceedings of the 23rd Annual Computer Security Applications Conference (ACSAC'07). IEEE Computer Society, 2007.
- **[GFC08]** Fanglu Guo, Peter Ferrie & Tzi-Cker Chiueh. **A Study of the Packer Problem and Its Solutions**. Procs. 11th Int. Symp. on RAID, 2008.
- **[JCL+10]** Guhyeon Jeong, Euijin Choo, Joosuk Lee, Munkhbayar Bat-Erdene & Heejo Lee. **Generic Unpacking using Entropy Analysis**. IEEE: 5th International Conference on Malicious and Unwanted Software, 2010.
- **[RAG16]** Ricardo J. Rodríguez, Javier Alonso & Iñaki Rodríguez Gastón. **Towards the Detection of Isolation-Aware Malware**. Submitted IEEE LATAM, under review process.
- **[KLC+10]** Min-Jae Kim, Jin-Young Lee, Hye-Young Chang, SeongJe Cho, Minkyu Park, Yongsu Park & Philip A. Wilsey. **Design and Performance Evaluation of Binary Code Packing for Protecting Embedded Software against Reverse Engineering**. IEEE Int. Symp. on Object/Component/Service-Oriented Real-Time Distributed Computing, 2010.

Zaragoza 1542

Qualitative and Quantitative Evaluation of Software Packers

María Bazús, **Ricardo J. Rodríguez**, José Merseguer

rjrodriguez@unizar.es ※ @RicardoJRdez ※ www.ricardojrodriguez.es

Universidad Zaragoza

Department of Computer Science and Systems Engineering
University of Zaragoza, Spain

December 12, 2015

NoConName 2015
Barcelona (Spain)