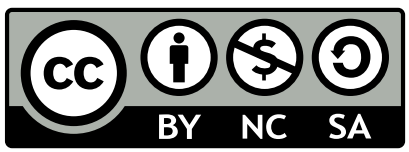
On Challenges in Verifying Trusted Executable Files in Memory Forensics

Daniel Uroz, **Ricardo J. Rodríguez**

Dept. of Computer Science and Systems Engineering
University of Zaragoza, Spain

June 3, 2020

DFRWS virtual EU 2020



Outline

Universidad Zaragoza

Outline

Universidad Zaragoza

Introduction

Incident response

- Common incident: **presence of malicious software** (malware)
- **Different types of analysis to get hints**:
  - **Computer forensics: disks + memory**
    - **Disk forensics**: analysis of device drives
    - **Memory forensics**: analysis of data within the system memory under study
  - Network forensics

Universidad Zaragoza

Introduction

Disk vs. memory

- Sometimes, **access to physical device drives are difficult to achieve**
- Think about **current limits of storage capacity versus memory capacity**
  - Terabytes versus gigabytes
  - **Facilitates the initial triage**
- **Some data only resides into memory**
- Memory stores the current state of the system, which is dumped into a data file for analysis (*memory dump*)

Universidad Zaragoza

Introduction

What does the memory dump contain?

- **Full of data** to analyze
- **Every element susceptible to analyze is termed as a *memory* artifact**
  - Retrieved through appropriate internal OS structures or using a pattern-like search
- Snapshot of the running processes, logged users, open files, or open network connections – **everything that was running at acquisition time**
- It may contain also **recent system resources freed**
  - Normally, memory is not zeroed out when freed

Universidad Zaragoza
1542

Introduction

What does the memory dump contain?

- **Full of data** to analyze
- **Every element susceptible to analyze is termed as a *memory* artifact**
  - Retrieved through appropriate internal OS structures or using a pattern-like search
- Snapshot of the running processes, logged users, open files, or open network connections – **everything that was running at acquisition time**
- It may contain also **recent system resources freed**
  - Normally, memory is not zeroed out when freed

Some problems here...

- Memory inconsistency if acquired in live systems
- Incorporation of temporal dimension (Pagani et al., 2019)

Universidad Zaragoza

Introduction

Code signing

- **Helps to establish trust in computer software**
  - Authenticate the software publisher
  - Guarantee code integrity through the validation of the digital signature shipped within the software
- **Commonly used in modern operating systems**, such as Windows
  - If the program binary is not signed and was downloaded from the Internet, Windows UAC asks for permission
  - Similarly, if the program binary is properly signed, the user is not asked for permission

Universidad Zaragoza
1542

Introduction

Malware + code signing

- **Deceive users to execute malicious programs**
- Code signing used in malware in-the-wild
  - **Not very common**
  - **Compromised certificates or issued directly to malware developers**
- Certification authorities (CAs) make **revocation process of abused certificates**

Universidad Zaragoza

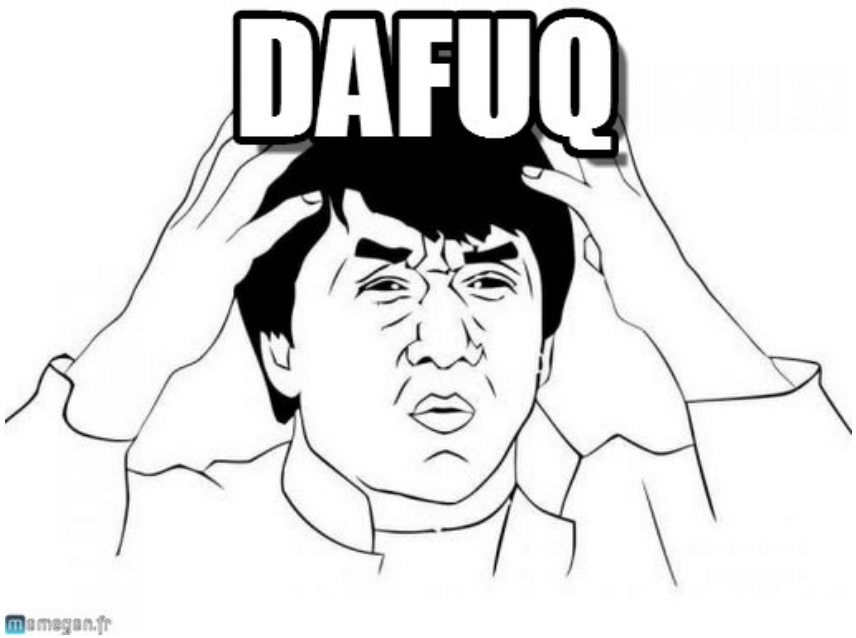
Introduction

Can we use code signing as a preliminary step to prioritize the list of suspicious processes in a memory dump that need further analysis?

Universidad Zaragoza

Introduction

Not very fruitful, though

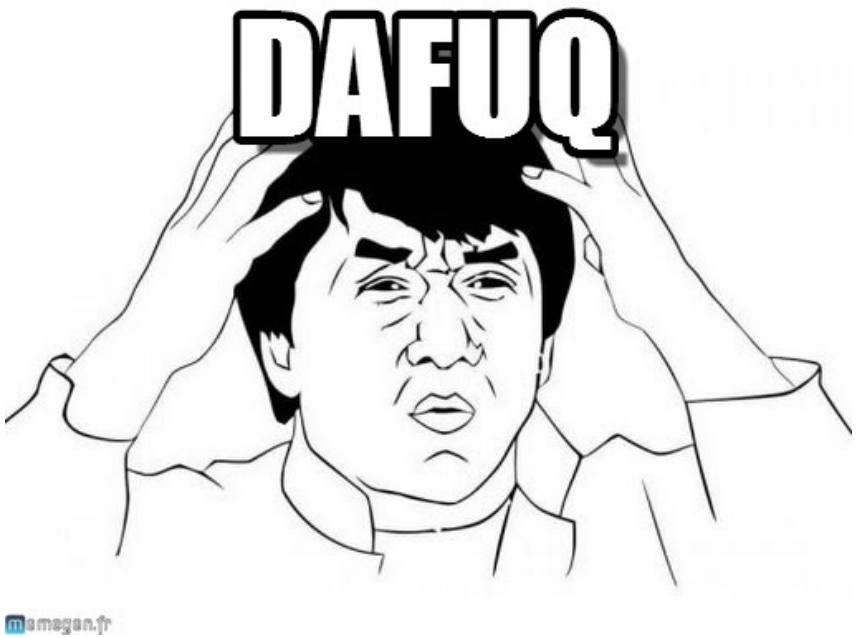


Main problems

- **Inaccuracy of process vs. program application**
- **Memory swapping**
- **Relocation**

Introduction

Not very fruitful, though



Main problems

- **Inaccuracy of process vs. program application**
- **Memory swapping**
- **Relocation**

To what extent can these issues negatively affect the signature computation? Any other issues of relevance? Any solution?

Universidad Zaragoza

Outline

Universidad Zaragoza

Previous Concepts

Microsoft Authenticode

- **Code-signing standard used by Windows to digitally sign files that adopt the Windows PE format**
- **Follows the PKCS#7 structure**: signature (hash value of the PE file), a timestamp (optional) and the certificate chain
- **Supports MD5 (backward compatibility), SHA-1, and SHA-256 hashes**
  - **A PE can be dual-signed**
- The certificate chain is built to a trusted root certificate by using **X.509 chain-building rules**

Universidad Zaragoza
1542

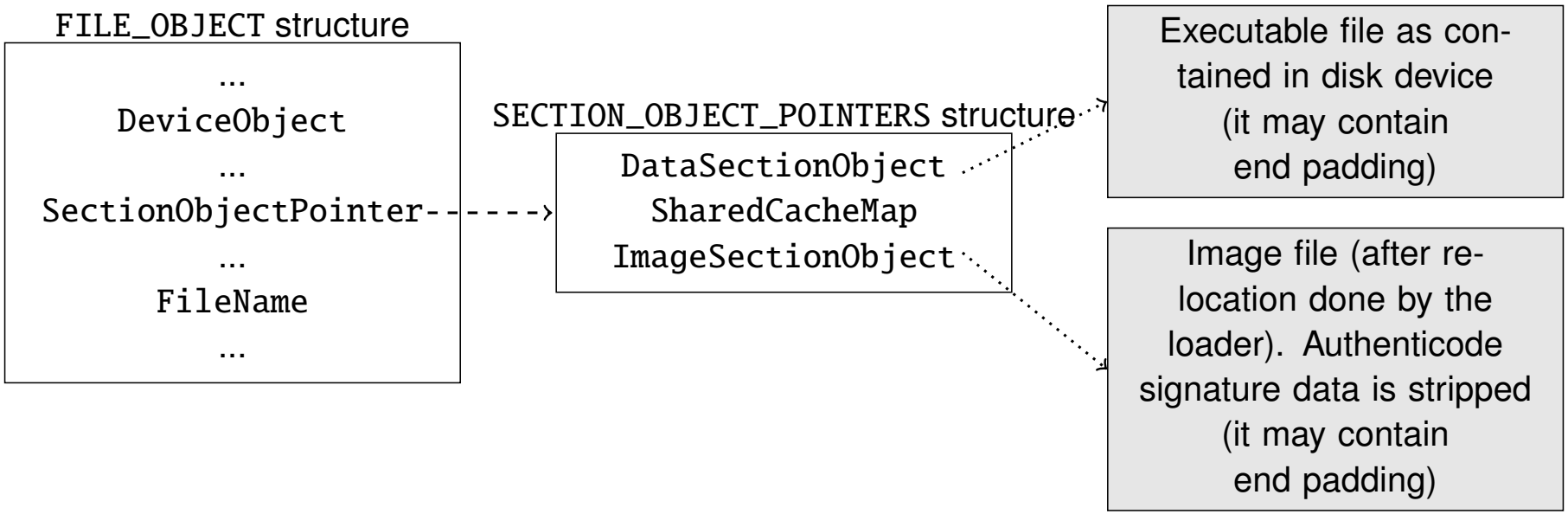
Previous Concepts
Microsoft Authenticode

- **Shipped in two ways: embedded signature or catalog-based**
  - Both follows the **Abstract Syntax Notation One (ASN.1) format**
  - **Embedded signature is a** `WIN_CERTIFICATE` **structure** in the `Security` directory entry within the `Data directories` array of the PE optional header
  - **Catalog-based: catalog (**`.cat`**) files**
    - Collect digital signatures of an arbitrary number of files
    - **Signed, to prevent unauthorized modifications**
    - Located in the `system32/catroot` directory
    - `catdb` database, which follows the Extensible Storage Engine format
- **Verification of a signature is done by the WINTRUST and CRYPT32** DLLs (for more details regarding the execution flow, see the paper)

Universidad Zaragoza

Previous Concepts

File Objects



- **Logical interface between kernel and user-mode processes and the corresponding file data stored in the physical disk**
- **Kernel-level structure** used to track a single open instance of a file
- Stores a **pointer to a** `SECTION_OBJECT_POINTERS` **structure**
  - Stores file-mapping and cache-related information for a file stream
  - **Three opaque pointers**: `DataSectionObject`, `SharedCacheMap`, and `ImageSectionObject`

Universidad Zaragoza 1542

Previous Concepts

File Objects

```
String os = System.getProperty("os.name");
if (os.startsWith("Windows 9") || os.equals("Windows Me")) {
    throw new RuntimeException(
```

- `DataSectionObject`
  - **Track state information for a data file stream**
  - May contain end padding bytes by memory alignment issues
- `ImageSectionObject`
  - **Track state information for an executable file stream**
  - May contain end padding bytes by memory alignment issues
  - **Contains the image file after the relocation process was done to prepare the image for execution**

Interesting findings regarding `ImageSectionObject` objects

- **The section containing the embedded signature is missed** (*stripped by the PE loader prior loading?*)
- **Relocation section still remains**

Universidad Zaragoza

Outline

Universidad Zaragoza

sigcheck

- **Verify digital signatures of executable files in memory dumps**
  - Works for `.exe`, `.dll`, and `.sys` files
- **Plugin for Volatility 2.6**. Dependencies:
  - `tasks`: to retrieve the list of processes in execution
  - `modules`: to retrieve the list of drivers
  - `devicetree`: to retrieve the driver objects for a given module
  - `filescan`: to retrieve the list of file objects
  - `dumpfiles`: to obtain the list of memory addresses associated to a `FileObject`
- **Released under GNU/GPLv3 license**:
  - `https://github.com/reverseame/sigcheck`

Universidad Zaragoza

sigcheck

- **Pseudo-algorithm with full details in the paper**
- **Possible outcomes**:
  - File object cannot be retrievable
  - Errors while parsing PE content (partial content)
  - PE rebuilt failed or checksum mismatch
  - Authenticode mismatch (it may be caused by an incorrect image base address)
  - Catalog-signed
  - Not signed (it may be caused a catalog-signed file with an incorrect image base address)
  - Verification of certificate chain (relies on OpenSSL): `CERT_EXPIRED`, `CERT_UNTRUSTED`, `CERT_FORMAT_ERROR`, `CERT_VERIFICATION_SUCCESS`, `CERT_REVOKED`
- **Side product:** `sigvalidator`
  - Python script to verify PE files (as in disk)

Universidad Zaragoza

Outline

Universidad Zaragoza

Experiments and Discussion

Effectiveness

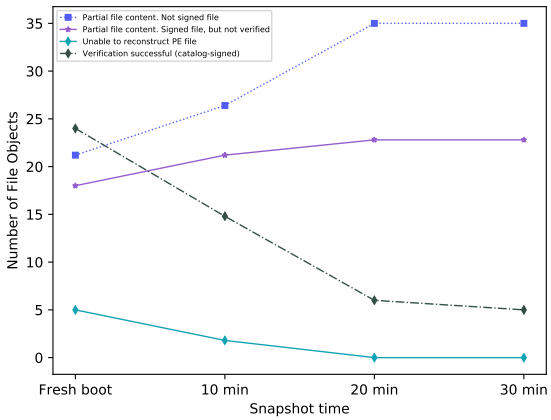
Description of experiments

- **32-bit and 64-bit Windows 7 SP1** OS Enterprise edition Build 7601
  - Windows 8.1 and Windows 10 were **discarded because** `dumpfiles` **was not working appropriately**
- VMware Workstation 15 Pro 15.0.2 build-10952284
- **Common workstation software installed**:
  - Mozilla Firefox 69.0.3, Google Chrome 77.0.3865.120, Libre Office 6.3.1, Notepad++ 7.8, and Adobe Reader DC 2019.012.20036
- **Every VM was executed 10 times in 4 different time moments**:
  - At a fresh boot and after 10, 20, and 30 min of user activity
  - Data documents opened and viewing YouTube content with Mozilla Firefox

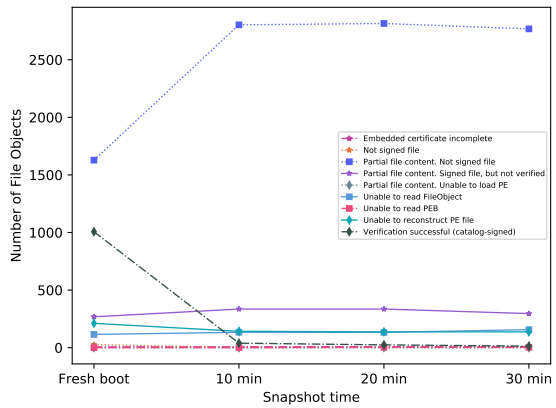
Universidad Zaragoza
1542

Experiments and Discussion

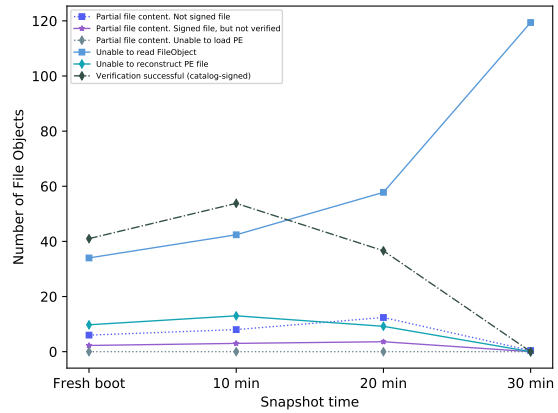
Effectiveness – plots



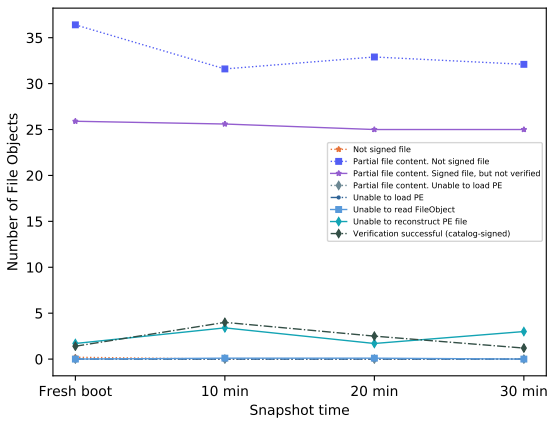
(a) Win7 x86 EXE files



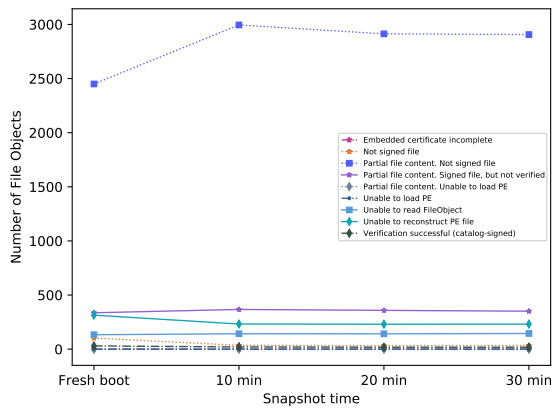
(b) Win7 x86 DLL files



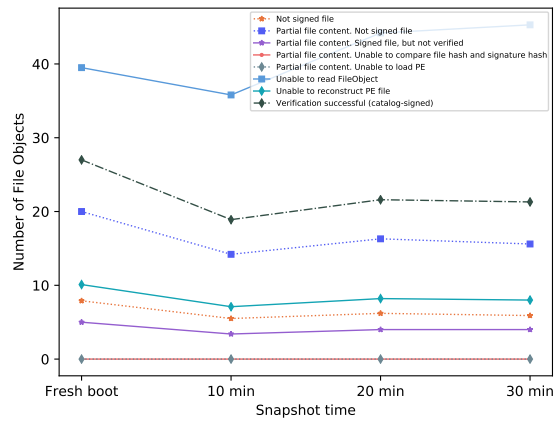
(c) Win7 x86 SYS files



(a) Win7 x64 EXE files



(b) Win7 x64 DLL files



(c) Win7 x64 SYS files

Experiments and Discussion

Effectiveness

Discussion of results

- **More chances of retrieving file objects with complete data at fresh boot**
- **32-bit scenarios provide better results**
- **Almost half of x86 driver files are successfully verified as catalog-signed files, while executable and DLL files reach more than 30%**
- **Many file objects are partially retrieved**
- **None of the file objects retrieved in any scenario contained the Authenticode signature as full content**

Universidad Zaragoza

Experiments and Discussion

Analysis of signed malware – description of experiments

Label	AVClass result
malware01	`polycrypt`
malware02	`perfectdefender`
malware03	`trustedzone`
malware04	`megacortex`
malware05	`sobit`
malware06	`pernefed`
malware07	`perfectdefender`
malware08	`geral`
malware09	`perfectdefender`

(SHA 256 of the samples in the paper)

- **Certificate issues**: expired and revoked by issuer; only expired or only revoked by the issuer; and self-signed certificate
- **Three tools**: Windows UAC, SysInternal's sigcheck tool, and `sigvalidator` (side-product of our plugin)

Experiments and Discussion

Analysis of signed malware

Discussion of results

- **Both Windows UAC and SysInternal's sigcheck are more focused on the publisher trustfulness**
  - Unlike our plugin, **a certificate expired is not reported**
- **Messages shown by the Windows UAC are less intuitive for users**
- **Self-signed certificate is detected by the three tools, although text messages differ slightly**
- **Most interesting result**: *malware04*, which has a revoked certificate
  - SysInternal's sigcheck warns about it
  - Windows UAC tells the user that the file comes from a verified publisher, probably caused by the parameter settings when calling to `WinVerifyTrust`
  - `sigcheck` **returns a successful verification, since it does not perform any certificate revoking checking** (currently unsupported by OpenSSL)

Universidad Zaragoza

Outline

Universidad Zaragoza

Limitations

Data incompleteness

- **Memory swapping, page smearing and demand paging**
- **A single flipped byte affects to the Authenticode signature calculation**
- **Solutions**:
  - *Guarantee that the Windows PE loader locks memory pages that contain file objects*
  - *Use a combined analysis of a memory dump and their corresponding swap files*

Universidad Zaragoza

Limitations

Data incompleteness

- **Memory swapping, page smearing and demand paging**
- **A single flipped byte affects to the Authenticode signature calculation**
- **Solutions**:
  - *Guarantee that the Windows PE loader locks memory pages that contain file objects*
  - *Use a combined analysis of a memory dump and their corresponding swap files*

Data changes caused by PE relocation

- **Changes on the image base address due to relocation**
- **In 64-bit, we cannot bruteforce** (in theory, 2^{52} possibilities as upper bound, considering 4KiB memory page alignment)
  - In fact this value is lower, giving the current limitations on 64-bit processors for virtual memory
- **Collisions considering the PE checksum is high** ($1/2^{24}$)
- **Solution**: *database storing the version information, Authenticode hash signature, and image address values for every system file* (unfeasible in practice)

Universidad Zaragoza 1542

Limitations

Catalog-signed files

- **Many system files in Windows are catalog-signed PE files**
- **Catalog can be updated with every Windows update**
- **Solution**: *centralized CERT database of catalog data*

Universidad Zaragoza

Limitations

Catalog-signed files

- **Many system files in Windows are catalog-signed PE files**
- **Catalog can be updated with every Windows update**
- **Solution**: *centralized CERT database of catalog data*

Executable file and process inconsistencies

- **Mapped file can be legit, but its corresponding process can be malicious**
- **Code injection or process hollowing are undetected** by our tool
  - Also undetected by other forensic programs (e.g., Process Explorer, Process Hacker)
- **One of the biggest challenges in memory forensics: guarantee that the binary code of an image file was unmodified**
- **Solution**: *use of plugins to detect malicious code or similarity digest comparison*

Universidad Zaragoza 1542

Outline

Universidad Zaragoza

Related Work

- **Analysis and the presence of Authenticode-signed files in malware landscape**
  - 18% of collected samples are signed binaries, with very few (only 11 samples) using revoked certificates
  - Other work highlighted three out of four samples being correctly validated as Authenticode-signed files
- **Weaknesses of code signing in Windows**
  - Inadequate client-side protections, publisher-side key mismanagement, and CA-side verification failures

- **All of the are focused on executable (on-disk) files, rather than image (in-memory) files**
- **We are the first to formally document the limitations and possible solutions that memory forensics impose to the analysis of Authenticode-signed files**

Universidad Zaragoza
1542

Outline

Universidad Zaragoza

Conclusions

- **Windows PE loader and Windows memory subsystem affect to memory forensic analysis**
  - Relocation, paging memory, page smearing, and demand paging
- **Verification of digital signatures of processes captured in a memory dump becomes a difficult task**
- **Limitations**:
  - **Data incompleteness**, **data changes caused by relocation**, **catalog-signed files**, and **executable file and process inconsistencies**

Universidad Zaragoza

Conclusions

- **Windows PE loader and Windows memory subsystem affect to memory forensic analysis**
  - Relocation, paging memory, page smearing, and demand paging
- **Verification of digital signatures of processes captured in a memory dump becomes a difficult task**
- **Limitations**:
  - **Data incompleteness**, **data changes caused by relocation**, **catalog-signed files**, and **executable file and process inconsistencies**
- **Plugin `sigcheck` for Volatility: enables an analyst to verify a digitally-signed file (if feasible) retrieved from a memory dump**
  - It relies on internal OS structures (in particular, file objects)
  - Side product: `sigvalidator` (for PE files)
- The **success rate is low**, especially when the memory is acquired from a system that was running for a long time

.

Universidad Zaragoza

On Challenges in Verifying Trusted Executable Files in Memory Forensics

Daniel Uroz, **Ricardo J. Rodríguez**

Dept. of Computer Science and Systems Engineering
University of Zaragoza, Spain

June 3, 2020

DFRWS virtual EU 2020