Extraction and analysis of retrievable memory artifacts from Windows Telegram Desktop application

Pedro Fernández-Álvarez, Ricardo J. Rodríguez

Department of Computer Science and Systems Engineering

University of Zaragoza, Spain

March 30, 2022

DFRWS EU 2022

Universidad Zaragoza

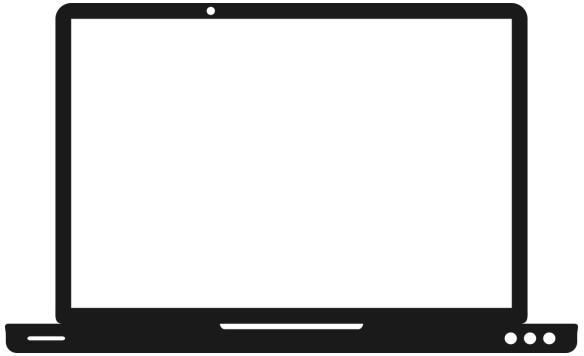
1542

Introduction (I)

- **Instant Messaging (IM)** applications → Communicate quickly and easily
  - Sometimes misused for malicious purposes
- Forensic analysis → Essential clues to solve or clarify a possible crime
  - Relevant data to an investigation
- **Encryption** (database and communications)
- Application needs to decrypt data
  - Location: **RAM**
  - **Memory forensics** important when database and communications are encrypted

Introduction (II)

- Popularity of smartphones → Most research oriented to mobile platforms
  - Less attention paid to **desktop applications**
- **Telegram** → 5 most popular IM apps globally
  - **Telegram Desktop** (Telegram official client for computers)
- Goal
  - Study the contents present in **RAM** of the Telegram Desktop process
- Scope
  - Version 2.7.1
  - Windows 10 (most popular OS for desktop devices)
- Telegram Desktop **database and communications encrypted**



Related Work

- Research in the memory forensics field focused on IM apps
  - Different **devices** (smartphones, computers…)
  - Different **operating systems** (Windows, Android…)
  - Different **IM applications** (WhatsApp, Telegram…)
  - Different **IM application versions** (mobile version, desktop version…)
- Normally, data about conversations
  - Various degrees of success
- Research not found for **Telegram Desktop** on **Windows**

Background (I)

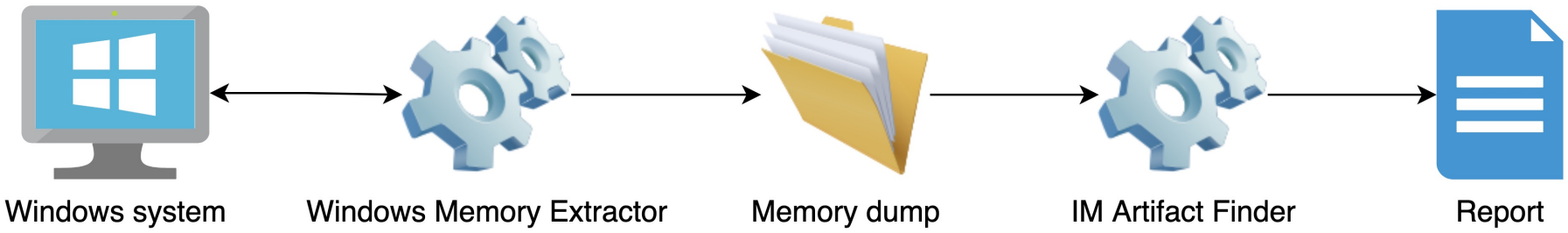
- Virtual memory
  - Each process has its own **private address space** → Divided in **pages**
  - **Relationship** between virtual and physical memory → Virtual memory manager
  - Save pages to disk (**memory paging**)
- Windows **process memory mapping**
  - Image file run → Virtual address space is created for new process
  - Stack
  - Heap
  - DLLs → Mapped into the process address space

Background (II)

- Telegram
  - **Multiplatform** IM service with **client-server** architecture
  - Necessary to have user account associated with phone number
  - **Username** support
  - **Text**, **voice** and **video**
  - Applications can be **locked** with password

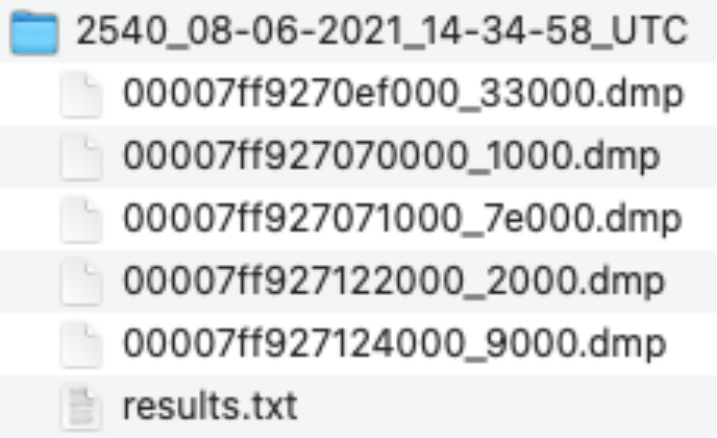
Analysis Environment

- Based on a 3-phase analysis methodology commonly used in digital forensics
  1. Extraction → Obtain data to analyze
  2. Analysis → Identify relevant digital artifacts
  3. Reporting → Generate a report with the artifacts found
- Composed of **2 command line tools** (GNU/GPLv3 license) → Facilitates **integration**
  - **Windows Memory Extractor:** https://github.com/reverseame/windows-memory-extractor
  - **IM Artifact Finder:** https://github.com/reverseame/instant-messaging-artifact-finder



Windows Memory Extractor

- Characteristics
  - C++
  - No installation needed (USB drive)
- Optional arguments
  - Specify protections
  - Specify name of module
- Nomenclature
  - Access an element stored at a virtual address
  - **Navigate** through a process dump from object A to object B




```
.\WindowsMemoryExtractor_x64.exe --pid 1234
```

Windows Memory Extractor Demo

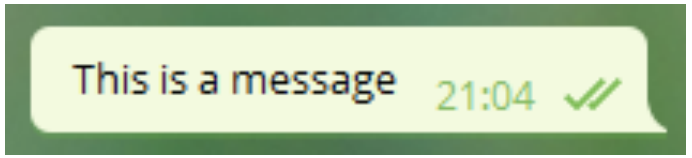
IM Artifact Finder (Framework)

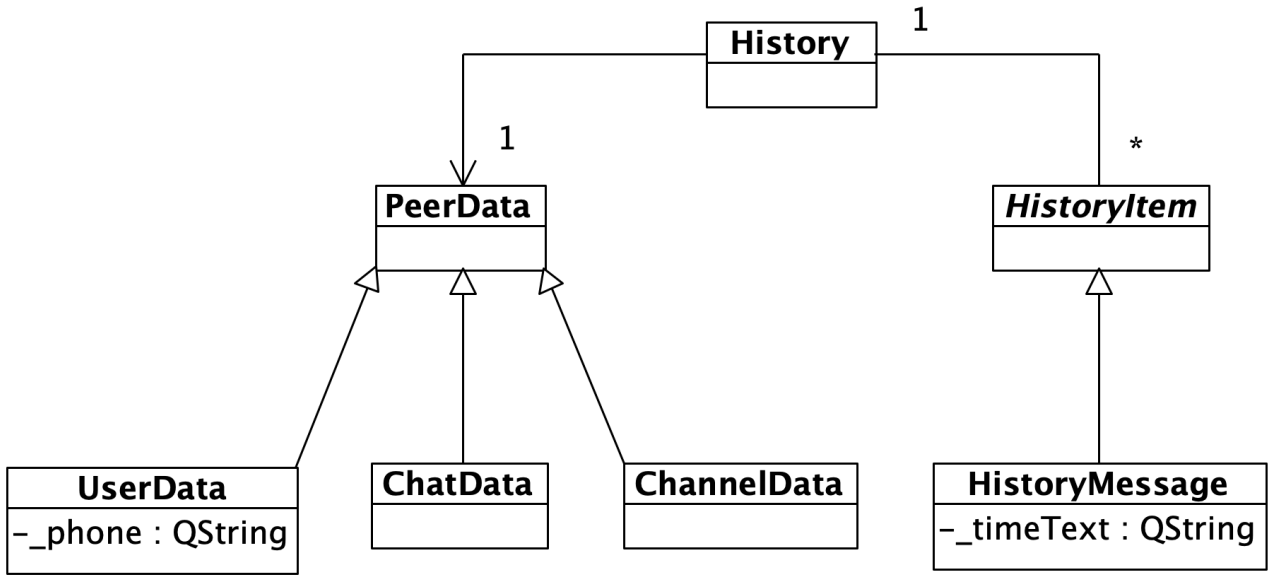
- Characteristics
  - Python
  - **Framework** → Support different IM applications, operating systems and devices
- IM Artifact Finder depends on interfaces → **Easy to add support** for new IM application
  - Implement a set of interfaces as well as the classes that represent artifacts
- Reports
  - **JSON**
  - Easy to add support for new reporting formats (CSV)

```
python3 finder.py memory_data_path TELEGRAM_DESKTOP
```

IM Artifact Finder (Telegram Desktop)

- **Source code analysis**
  - **Automatic** → General idea of structure → Identification of relevant elements
  - **Manual** analysis of relevant elements
- **Pattern matching**
  - Phone number
  - Time
- **Pointers** to other objects
  - Obtain additional related objects





Experiments and Discussion (I)

- **Owners** of accounts used in Telegram Desktop
  - ID, full name, phone number and username
- Users who **share** their phone number
  - Some users who do not share it (privacy)
  - **Distinguish** between contact, blocked and bot
  - Retrieve **deleted** contacts
- Reconstruct **accessed** conversations
  - Except edited and deleted messages
- Account to which they belong


```
{
    "user_id": 1680408400,
    "name": "UsuarioTest TFM",
    "is_contact": true,
    "is_blocked": false,
    "phone_number": null,
    "username": "UsernameTestTFM",
    "about": null,
    "is_bot": false
},
```

```
"conversations": [
    {
        "conversation_id": 916093840,
        "conversation_type": "Individual conversation",
        "name": "Pedro Fernández",
        "messages": [
```

Experiments and Discussion (II)

- Multimedia messages
  - Files → Name and type
  - Shared contacts → Name and phone number
  - Geographic locations
- Locking → **Same** information
- Log out and deleted conversations → Reliability


```
"attachments": [
    {
        "attachment_id": null,
        "latitude": 42.59849852178585,
        "longitude": -5.571721718367973,
        "title": "Museo de León",
        "description": "Pl. Santo Domingo, 8"
    }
]
```



```
{
    "message_id": null,
    "text": "This is a normal message",
    "date": "2021-06-18T18:09:14+00:00",
    "sender": {
        "user_id": 1680408400,
        "name": "UsuarioTest TFM",
        "is_contact": true,
        "is_blocked": false,
        "phone_number": null,
        "username": "UsernameTestTFM",
        "about": null,
        "is_bot": false
    },
    "attachments": null
},
```

Conclusions

- Importance of **memory forensics** when there is **encryption**
- Obtain digital artifacts **relevant** for an investigation
- **Future work**
  - Support other Telegram Desktop versions
  - Analyze Telegram Desktop in other operating systems
  - Support more IM applications

Extraction and analysis of retrievable memory artifacts from Windows Telegram Desktop application

Pedro Fernández-Álvarez (pfernandez@unizar.es), Ricardo J. Rodríguez (rjrodriguez@unizar.es)

Department of Computer Science and Systems Engineering

University of Zaragoza, Spain

March 30, 2022

DFRWS EU 2022

Universidad Zaragoza

1542