IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Decoding The Divide: Forensic Discrepancies In Windows OS Identification

M B Pathak, Emmanuel Ben T, Eswara Sai Prasad Chunduru, M Krishna, Sourav Mondal
Scientist-B, Forensic Professional, Asst. Director (Retd.), Deputy Director (Retd.), Scientist-B
Central Forensic Science Laboratory, DFSS, MHA, GOI, India.

ABSTRACT

The accurate identification of an operating system (OS) version is pivotal in forensic investigations and registry analysis. However, discrepancies in the detection results provided by different forensic tools can lead to significant challenges. This research paper explores the disparity observed when Magnet AXIOM identifies the OS version as Windows 11 Pro, whereas other registry analysis tools detect it as Windows 10 Pro. We investigate the underlying causes of this inconsistency by examining the various algorithms, registry keys, and parsing techniques employed by these tools. Our analysis reveals that differences in detection algorithms, registry key interpretation, software configuration, and data parsing methods contribute to the observed disparities. By understanding these factors, we aim to provide forensic professionals with insights to achieve consistent and reliable OS version detection across different forensic tools. This study underscores the importance of harmonizing detection methods and highlights the need for standardized approaches in forensic software development.

Index Terms: Registry, Operating System, Digital Forensic, OS Forensic

1. INTRODUCTION

1.1 Background

Digital forensics is of inevitable value in modern investigations, enabling the retrieval and analysis of digital evidence from diverse computing environments, particularly operating systems (OS). With the rapid growth of technology and the increasing reliance on digital information, OS forensic tools have become essential for forensic professionals. These tools provide access to valuable data hidden within an operating system's digital artifacts. Identifying the accurate OS version of the questioned system has significant effect in the admissibility of the report in the court [9].

We will explore methods for manually analysing the Windows registry to uncover traces of windows 11 operating system details. This analysis highlights the crucial role of digital forensic investigations by demonstrating how the Windows registry and some forensic registry analysis tools shows discrepancies in the operating system identification [12].

Manually examining the registry can be an arduous task, as its values are stored in a hexadecimal format that is difficult to interpret. To simplify the investigation process, forensic investigators use specialized tools both open source and commercial to decode registry key values and present them in a more accessible, readable format. But it is crucial to verify whether the data interpreted by the tools is reliable. We are exploring some manual ways for OS identification and will conclude which software is reliable in the case of windows 11 operating system identification.

In this paper, the basic aim is to identify the accurate operating system used in a windows system by using various forensic tools which include both open source and some commercial tools. This paper aims to help researchers, investigators and other professionals to accurately identify the operating system while some software shows discrepancy in the operating system identification [12]

1.2 Problem Statement

Forensic investigations rely on the precise identification of an operating system (OS) version, particularly during registry analysis. However, inconsistencies in detection results among forensic tools present a significant challenge. A notable discrepancy arises when Magnet AXIOM identifies the OS as Windows 11 Pro, whereas other registry analysis tools classify it as Windows 10 Pro. This inconsistency stems from variations in detection algorithms, registry key interpretations, software configurations, and data parsing methods across different tools. These disparities can lead to unreliable forensic outcomes, potentially affecting the accuracy of digital investigations. Therefore, this study seeks to analyse the factors contributing to these inconsistencies, offering forensic professionals' insights to enhance reliability in OS version detection. Additionally, the research emphasizes the need for harmonized detection methods and standardized approaches in forensic software development to ensure uniformity and accuracy in digital forensics.

2. LITERATURE REVIEW

The growing prevalence of the detailed analysis of the windows artifacts during the cyber-crime investigation is inevitable as general tools are relying on the registry data of windows which should be cross verified with other artifacts before arriving at conclusion.

2.1. Introduction to OS Version Detection in Forensic Tools

Digital forensics is a critical process for gathering, identifying, extracting, and documenting electronic evidence for use in court [7]. Registry analysis in windows forensics is utmost important discipline of cyber or network crime investigation [11]. The Windows Registry serves as a central repository or hierarchical database containing configuration data for the system [11]. It is often called the heart of Windows Operating Systems because it holds all configuration settings for specific users, groups, hardware, software, and networks. Due to this comprehensive storage, the Windows Registry is viewed as a valuable source of forensic evidence [11]. Investigating the Registry can help collect information relevant to a case [2].

Accurate OS version detection is implicitly important in digital forensic investigations because the Registry structure and the specific keys containing valuable forensic artifacts can vary between OS versions. For instance, while Windows XP and Windows 7 Registry structures are identical and share identical root keys [11], research specifically examined the registry of a Windows 10 device to identify artifacts related to cloud storage usage [3] Another study focused on the Windows 7 registry architecture for forensic analysis [10]. The digital forensics relies significantly on software tools to acquire and investigate digital evidence. Without these tools, analysing digital media would often be impossible [12]. Common forensic tools used for various digital forensics tasks, including registry analysis, are mentioned in the sources. These include Windows Reg Editor (regedit.exe or regedt32.exe) [2], AccessData's Registry Viewer [8], Forensic Tool Kit (FTK) [7], OSForensics [2], WIRESHARK [7], Autopsy, TRUECRYPT, X-WAYS, and SANS SIFT. Computer forensic tools aim to certify that extracted evidence is correct and dependable [7].

2.2. Detection Algorithms and Methods

The sources describe various methods and functionalities employed by forensic tools, which would be relevant to detecting OS versions, although specific algorithms for version detection are not detailed. Forensic investigators analyze user activities, and the details of these activities are recorded in the Windows Registry [2]. Tools are needed to extract required information from the Registry [2]. Methods discussed include integrated Registry analysis, timeline analysis, and comparing Registry hive files. Integrated analysis examines hive files stored on the seized hard disk [2]. Timeline analysis is used to correlate the sequence of

events and their timings. Comparing hive files (current vs. backup) helps identify potential locations where a user may have deleted or altered values, keys, or subkeys to hide their activity [2].

Some tools offer different scanning modes; for example, the Stellar tool can perform normal or deep scanning, with deep scanning doing a signature-based file search useful for recovering files missed by regular scanning [7]. Pattern recognition is suggested as an ideal method for the analysis step in digital forensics. Tools like Encase and FTK can handle scripts to extract information from data [7]. The proposed RegForensicTool performs integrated Registry analysis, timeline analysis, and extracts information useful for digital forensic analysis of the Registry. It can examine and extract evidence from the Registry of various operating systems like Windows XP, Windows 7, and Windows 8 [2].

Comparing the detection methods of specific tools like Magnet AXIOM (not mentioned in the provided sources) with others mentioned (FTK, OSForensics, etc.) is not possible based solely on these sources.

2.3. Registry Key Interpretation

The registry of windows operating system is a hierarchical database composed of keys and subkeys that contain configuration settings [11]. These keys and subkeys are crucial for forensic investigation. During one research process, subkeys were checked one by one of all root keys, and keys and subkeys with forensic importance were filtered and arranged. All the root keys mentioned have different functions, and consequently, their subkeys also have other functions. At the same time the sources emphasize that the Registry contains vast information about user activities, installed software, hardware, and network configurations [11]. And that specific keys/subkeys hold forensic value [10], they do not explicitly name the specific Registry keys that indicate the OS version**.** However, it is implied that the configuration data within the Registry, distributed among keys and subkeys, includes information about the installed operating system. Research has provided detailed descriptions of keys and their purpose in Windows XP Registry [2] and discussed keys helpful to forensic examiners in Windows 7 Registry. Tools are needed to interpret these keys and transform them into meaningful evidence [11]. Offered a catalog of high-value registry artifacts and explored anti-forensic strategies, calling for more robust and tamper-resistant forensic tools [1]. Developed a functional prototype tool capable of decoding hex-encoded registry values for reconstructing accurate user activity timelines [10].

2.4. Software Configuration and Data Parsing

Software configuration and data parsing are inherent to the functionality of cyber forensic tools. Tools should be capable of parsing various digital file formats and data structures to interpret digital data [12]. The analysis stage of digital forensics involves extracting information from examined data [7]. The RegForensicTool, for example, performs integrated analysis, extracts evidence and timestamps, and generates reports [2]. It also extracts information about running processes, services, and DLLs [2]. FTK is described as a tool that searches a hard disk for various types of data, like text strings, and can decrypt encrypted data using dictionary passwords. It can also analyze forensic data pictures stored in various formats (E01, DD/raw, AD1) [7]. This implies significant data parsing capabilities.

While the sources discuss the capabilities of tools in extracting and presenting data, they do not explicitly detail how software configuration settings within these tools or specific data parsing techniques directly influence or affect the detection results of the OS version. However, the reliability of tool results and the potential for errors are discussed, which could implicitly relate to parsing issues [7].

2.5. Updates and Patches

The provided sources do not contain information regarding the impact of OS updates and security patches on OS version detection in forensic tools or case studies illustrating this impact.

2.6. Challenges and Gaps in Current Research

Digital forensic practitioners face several challenges, including the increasing volume and complexity of digital forensic data [7]. The field is dependent on automated tools, and shortcomings or flaws in algorithms, instruments, and procedures can probably lead to erroneous findings. A significant challenge is ensuring the accuracy of tools and their effective use. Understanding boundaries of a tool also presents a research

opportunity, potentially leading to practitioner user-error. In legal disputes, flaws may exist in the tools used to interpret data, unbeknownst to the parties involved [12].

A significant challenge mentioned is the limited research available on digital forensic tool testing and validation [12]. While NIST's Computer Forensics Tool Testing (CFTT) Project exists, it is limited to a few core functions and platforms. Practitioners cannot simply accept a manufacturer's word regarding a tool's reliability; tools must be validated to confirm they function correctly as intended. However, undertaking this testing is burdensome, and error rates are rarely established [12].

Specifically, regarding Registry forensics tools, a key challenge is that existing tools often provide limited facilities for comprehensive analysis. Features found in one tool may not be available in another, requiring forensic examiners to use multiple tools for different purposes, which increases the time and cost of investigation. Some tools only cover specific activities (like autorun or USB devices) or merely provide a Registry view without the capability to extract specific forensic information [2]. There is a need for more robust and comprehensive Registry forensic tools that can extract data based on various user activities and integrate different analysis techniques [2]. Criminals may also employ anti-forensic techniques, such as deleting or altering Registry entries, which poses a challenge and necessitates methods for detecting such modifications and recovering deleted data [2].

A gap in the existing research, as reflected in these sources, includes the detailed analysis of how specific OS updates or patches affect the Registry structure in ways that might complicate OS version detection by forensic tools. Also, a deeper comparison of the underlying detection *algorithms* used by different commercial and open-source tools for OS version identification from the Registry is not present. While the importance of validation is stressed, specific methodologies or comparative studies on validating the OS version detection capability of different tools are not provided.

3. METHODOLOGY

3.1 Research Design and Approach

This study adopts an exploratory and applied research design to investigate discrepancies in OS identification by forensic tools. An initial qualitative analysis of registry values and tool outputs was conducted, followed by empirical testing of alternative methods such as system file analysis and build number mapping. A comparative analysis approach was used to evaluate the reliability of each identification technique in forensic context.

3.2 DATA COLLECTION METHOD

3.2.1 Registry Extraction and Analysis:

- The system and software registry hives were extracted from systems
- Key registry paths were examined:
- HKEY_LOCAL_MACHINE\SOFTWARE|Microsoft\windowsNT\CurrentVersion.
- Other supporting registry keys, such as ProductName, EditionID, and CurrentBuildNumber.
- Tools used for registry analysis included RegReport, RegRipper, F_RAT, Belkasoft Evidence Center, Magnet Axiom.

3.2.2 System File Inspection:

- Important System Files were identified and examined manually:
- Winver.exe (for Windows version information
- Ntosknrl.exe (kernel version)
- Metadata from system files retrieved from c:\windows\system32

3.2.3 Os Detection Using Forensic Tools:

- Magnet Axiom, Belkasoft Evidence Center, RegReport, RegRipper, F-Rat
- Each tool’s reported OS version was documented for comparison

3.2.4 Build Number Mapping

- The build numbers collected from registry and system files were mapped to the official Microsoft official release documentation.
- It helped to determine whether the system matched windows 10 or windows 11 regardless of the outputs provided by the tools.

3.2.5 Cross Verification Using System Commands:

Where possible, commands like systeminfo, winver, and powershell queries (Get-computerinfo) were run to obtain OS details directly from the system environment.

3.3 DATA ANALYSIS TECHNIQUE

The analysis in this study was conducted using a comparative and interpretive approach, combining both qualitative and quantitative techniques to evaluate the accuracy and reliability of different OS identification methods across forensic tools.

4. RESULTS AND ANALYSIS

The analysis is performed using multiple digital forensic tools—F-RAT, Belkasoft Evidence Center, Registry Viewer, and Magnet AXIOM—on a forensic image of a Windows system. The primary objective was to determine the operating system version based on registry data and compare the consistency across tools.

Findings:

1. RegRipper: Retrieved the OS details from the registry path: HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion. It reported the ProductName as Windows 10 Pro, with a CurrentVersion value 6.3. *(Figure. 1)*
2. F-RAT (Forensic Registry Analysis Tool): Displayed a similar result, again listing Windows 10 Pro based on the same registry key. *(Figure. 2)*
3. Belkasoft Evidence Center: Also identified Windows 10 Pro and Windows 10 Enterprise. *(Figure. 3)*
4. Magnet AXIOM: Contrarily, identified the OS as Windows 11 Professional (2009). Notably, the BuildNumber was 22000, and the version label was still 6.3. *(Figure. 4)*
5. RegReport: Retrieved the OS details from the registry path: HKLM\SOFTWARE\. It also reported ProductName as windows 10. *(Figure. 5)*

4.1 Alternative Methods for OS Identification

4.1.1 System File Version Analysis

Investigators should analyze key system files located in C:\Windows\System32\, such as:

- winver.exe: Confirms OS name and version when executed or viewed through file properties.
- ntoskrnl.exe: Kernel version analysis offers strong indicators of the actual OS build.

4.1.2 Build Number Mapping

Extract the build number from system files or by executing systeminfo/winver commands, and map it to known Windows versions:

Cross-verification ensures the build number corresponds to the correct OS version.

4.1.3 Reviewing Installation Artifacts

- Panther logs (C:\Windows\Panther\) may retain installation or upgrade histories.
- SetupDiag logs can reveal upgrade paths and original OS versions.

4.1.4 Analysing SMBIOS and System Metadata

Forensic tools can extract OS version information from BIOS/firmware metadata using commands like:

- wmic os get caption

- Get-ComputerInfo in PowerShell

  This hardware-level data reflects the current operating system more reliably than user-level registry values.

4.1.5 Recovery and Backup Analysis

Presence of Windows.old folders or recovery partitions can indicate recent upgrades and help in reconstructing the system's OS history.

5. DISCUSSION:

The core of the discrepancy lies in the ProductName field and the interpretation of the CurrentVersion and BuildNumber. Microsoft retained the internal CurrentVersion as 6.3 for both Windows 10 and Windows 11, making registry-only checks insufficient for accurate OS differentiation.

This analysis demonstrates the importance of examining additional system files, such as:

- System32\kernel32.dll version info
- winver.exe output
- BuildLabEx and UBR values

Magnet AXIOM likely uses these advanced indicators (e.g., Build number 22000 = Windows 11, per Microsoft Documentation, 2021), whereas other tools rely only on registry keys that haven't been updated for new OS branding.

Recommendations:

1. Do not rely solely on registry keys for OS identification. Include build metadata and file versioning checks.
2. Corroborate with multiple tools and OS build version mappings.
3. Reference Microsoft's official release documentation for accurate build-version interpretation.
4. Encourage tool vendors to update their detection logic to accommodate the static nature of specific registry fields.

6. CONCLUSION:

In digital forensic investigations, relying on outdated or partial indicators (like ProductName or CurrentVersion) can mislead OS identification. Advanced methods using build number correlation and executable metadata offer better accuracy and should be adopted as standard practice.

FIGURES

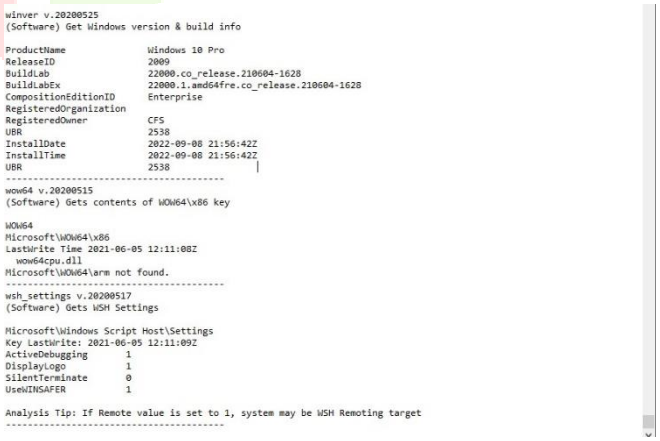

Figure 1

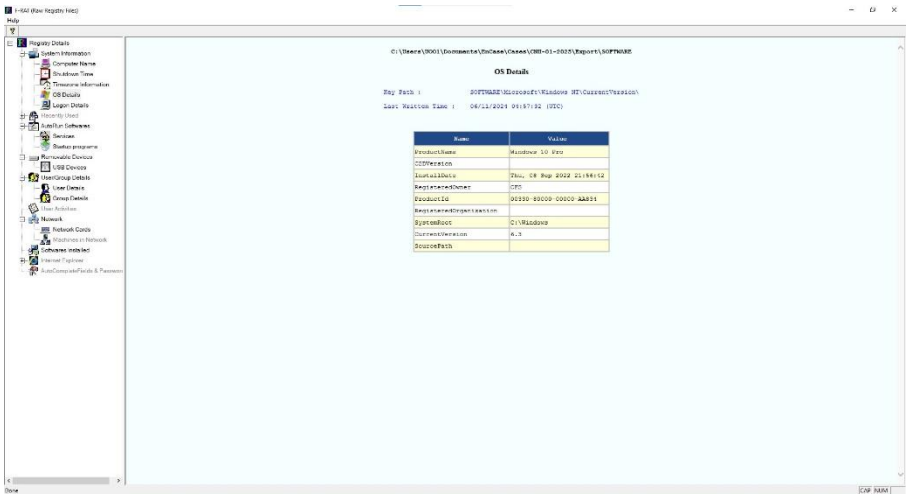

Figure 2

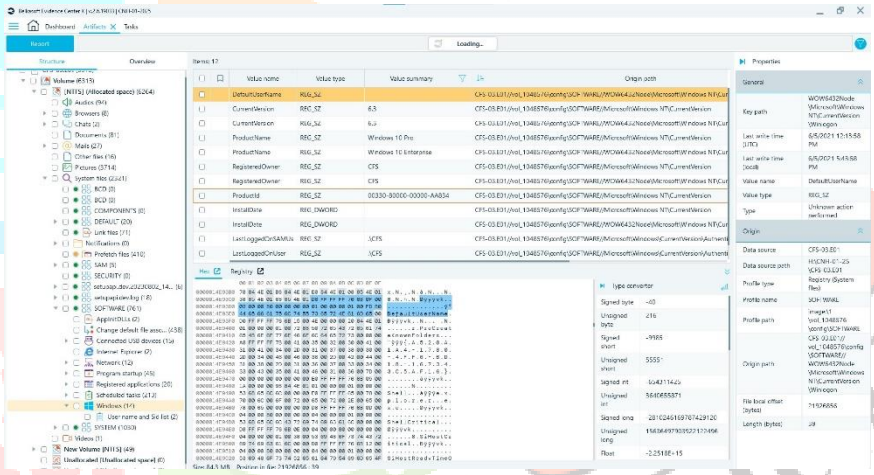

Figure 3

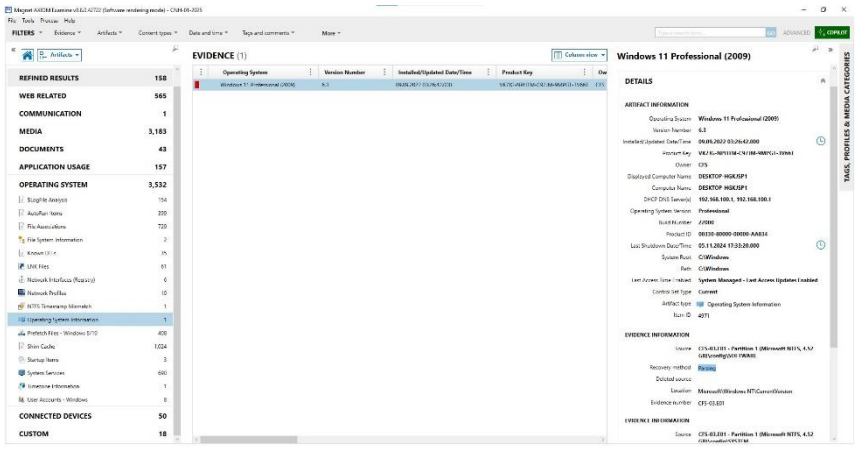

Figure 4

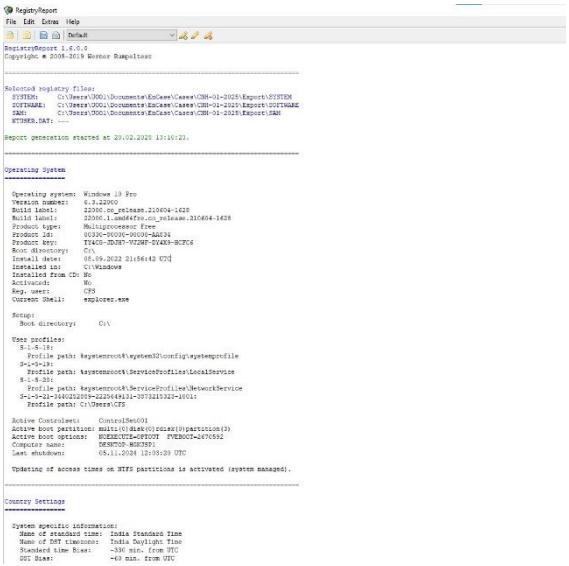

Figure 5

REFERENCE

1. Ali, H. (2018). Role of Windows Registry Forensics in Digital Forensics Investigation. LGU International Journal for Electronic Crime Investigation, 2(3). Available at: http://lgu.edu.pk/dfrsc/journal/Journal-IJECI.pdf
2. Patil, D. N. (2016). RegForensicTool: Evidence Collection and Analysis of Windows Registry. International Journal of Cyber-Security and Digital Forensics. https://doi.org/10.17781/P002064
3. Adesina, A. A., Adebiyi, A. A., & Ayo, C. K. (2022). Identification of forensic artifacts from the registry of windows 10 device in relation to idrive cloud storage usage. Bulletin of Electrical Engineering and Informatics, 11(1), 521–529. https://doi.org/10.11591/eei.v11i1.3489
4. Ali, H. (n.d.). Forensically Important Artifacts in Windows Operating systems. Digital Forensic Lead, Digital Forensic CoE. Available: https://www.academia.edu/29746363/Forensically_Important_Artifacts_in_Windows_Operating_systems
5. Albanese, M., Battista, E., & Jajodia, S. (2015). A De-ception Based Approach for Defeating OS and Service Fingerprinting. In IEEE Conference on Communications and Network Security (CNS) (pp. 317–325).
6. Elkan, C. (2001). The Foundations of Cost-Sensitive Learning. In International Joint Conference on Artificial Intelligence (IJCAI) (pp. 973–978).
7. Parveen, K., & Haider, G. (2024). Digital Investigations: Navigating Challenges in Tool Selection for Operating System Forensics. International Journal for Electronic Crime Investigation, 8(1), 173-332.
8. C. K. (2022). Forensic Foresight: A Comparative Study of Operating System Forensics Tools. International Journal of Engineering, Science and Mathematics, 11(07). https://www.ijesm.co.in/Forensicforesight

9. Aftab, P., & Haider, G. (2024). Digital Investigations: Navigating Challenges in Tool Selection for Operating System Forensics. International Journal for Electronic Crime Investigation (IJECI), 8(1). https://doi.org/10.54692/ijeci.2024.0801189
10. Ramani, A., & Dewangan, S. K. (2014). Digital Forensic Identification, Collection, Examination and Decoding of Windows Registry Keys for Discovering User Activities Patterns. International Journal of Computer Trends and Technology (IJCTT), 17(2), 109-115. Available from: http://www.ijcttjournal.org
11. Xie, H., Jiang, K., Yuan, X., & Zeng, H. (2012). FORENSIC ANALYSIS OF WINDOWS REGISTRY AGAINST INTRUSION. International Journal of Network Security & Its Applications (IJNSA), 4(2), 121–137. https://doi.org/10.5121/ijnsa.2012.4209
12. Horsman G. "I couldn't find it your honour; it mustn't be there!" - Tool errors, tool limitations and user error in digital forensics. Sci Justice. 2018 Nov;58(6):433-440. Doi: https://doi.org/10.1016/j.scijus.2018.04.001. Epub 2018 Apr 17. PMID: 30446072.