Received: 15/05/2025, Accepted: 20/06/2025

Research Article

Tampering with Truth: Designing a Tool to Undermine Digital Evidence Recovery

Himani Bansal[1], Manju Nunia[2]

[1] Department of CSE&IT, Jaypee Institute of Information and Technology, Noida, India

[2] Department of CSE, PES University, Bangalore, India

singal.himani@gmail.com, manju.nunia@gmail.com

ABSTRACT

The internet and technology advancements have given rise to too much comfort for both legitimate and malicious users or cyber criminals. Therefore, if these advancements are misused deliberately by the offenders, it can result in many harmful consequences, such as the prevention of services for benign or legitimate users. It is very difficult to investigate and prosecute any electronic crime because investigators need to build their cases or perform the investigation according to the evidence left by the computer criminals on the system. These days, computer criminals or adversaries are very much aware of the Computer Anti-forensic tactics and methods. Criminals apply such anti-forensic techniques to impede the whole digital investigation process efficiently and successfully. Such forensic investigation processes affected by Anti-forensic measures are too expensive and time-consuming to carry out. Numerous anti-forensic techniques can be used by anti-forensic practitioners to thwart the investigation process. Therefore, intruders try to hide or wipe out the evidence from the compromised system so that they cannot fall into the hands of Forensic examiners. Adversaries can try their very best to protect their evidence so that they can use any anti-forensic techniques for the same.

In this study, we have proposed an Anti-Forensic tool for destructing the evidential files stored on the hard disk. Our proposed tool works under the controlled environment of the Secondary extended file system of the Linux distribution. This tool clears out all the inodes which actually store the metadata of the files and folders on the file system, so clearing the inodes corrupts the file system structure in the internal structure of the file system. Detection of such activity performed by the anti-forensic practitioner is possible by the Forensics software or Forensic Investigator during the Investigation. Still, it will not be possible for the analyst to recover the file content as the file will no longer be accessible, as its inode entries have already been cleared out by the intruder. In this research work, we have also compared our proposed tool with similar existing tools and found that none other than the proposed one could clear out all the inode entries of the evidential files on the File System.

Keywords: *Artifacts, Anti-forensics, Linux File Systems, EXT2, Digital Forensics Investigation*

1. Introduction

The digital world is not safe from the malicious logic and smart but malefic techniques of those who desire to break them, as there are always security loopholes present, and the best minds are divided into 2 groups of people: one who wants to stay safe. They play the game from their end, and the others come up with even better methods to exploit these loopholes.

Users want their data secured, but since every piece of code is prone to some human errors, there are ways in which a crime can find its way inside. Digital security is now an even bigger issue, becoming harder to break into systems as more techniques are coming to light, due to which it is crucially important that we think from the perspective of a malicious hacker.

*Corresponding author: Himani Bansal, Jaypee Institute of Information Technology (singal.himani@gmail.com)

A digital crime investigator has the job to provide methods and come up with ways to reach the root of the trail once a crime is committed. Digital Forensic Investigator is the keyword which describes the act of deducing the source of the crime.

Investigators look for post-crime patterns, including IP tracing and electromagnetic fingerprinting, which can be as easy as checking the last modification date of a file, which can be helpful in the process. Given that all this information is out to the world, the so-called "bad-guys", the Anti-forensics, already know what an investigation may look like and can make arrangements to redirect the investigator in a direction which can leave him with fewer real clues or even stop the investigation altogether. techniques like Artifact wiping, Data Hiding, Trail obfuscation, Use of Encryption techniques and Steganography, Attack against Computer Forensic Tools etc. come under Anti-forensics.

Steps in digital forensic investigation include preparation, securing and protecting the crime scene, Evaluating and Conducting Survey, documentation, Evidence collection, Searching and Investigation of Anti-Forensic attacks, analysis of the digital artifacts and presentation of the general report.

Evidence is the most important role in the investigation, and there is no way of knowing in advance if the evidence found is left behind as an effect or is tampered with smartly, which is the central concept behind this research.

Recent studies highlight the growing role of anti-forensics in IoT environments [24]. Digital forensic intelligence is critical to bridging OSINT and traditional analysis [25]. Mobile application forensics, especially social media apps, presents unique anti-forensic challenges [26]. Secure communication in IoT forensics is a vital area of current research [27]. Forensic analysis of messaging platforms like WhatsApp reveals potential anti-forensic blind spots [28].

Linux is a Unix-like computer operating system assembled under the free and open-source software development and distribution model. It plays the role of the target operating system for this paper. File systems are required to store and access files in a storage medium like a hard disk, partitions, flash drives, etc and in any block device. Linux distributions support almost all the file systems, and the native file systems include EXT (obsolete), EXT2, EXT3, and EXT4. Among which EXT3 and EXT4 are journal and EXT2 is non-journaling file-system. EXT stands for Extended, and uses inodes structure to store and access the files. Inodes store metadata about the file, which includes file name, file name length, time of creation and modification, info about rightful owners and groups and most importantly, pointers to the location of the actual file data on the disk, which are called the block data.

2. Components of Digital Forensic Investigation

Digital Forensic Investigation has various components: Proactive, Active and Reactive as shown in Figure 1.

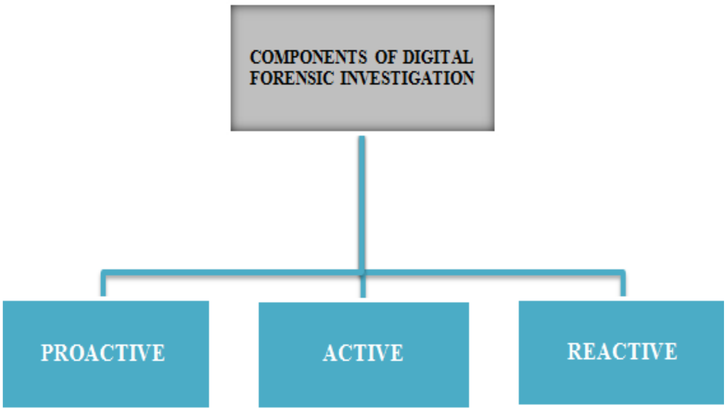


Figure 1: Various components of Digital Forensic Investigation

3. Anti-Forensic Techniques

Computer anti-forensics or counter-forensics is a newly emerging area that greatly threatens the digital forensic investigation process. So, the digital evidence becomes difficult to get admissible in court. Anti-forensics is a blend of methodologies and techniques that use technological advancements to obstruct the efforts of investigators. It is important to agree that this critical area of computer anti-forensics should have clearly defined standards and procedures.

Many challenges of computer forensics and anti-forensics exist that must be properly analysed. Anti-forensics is the practice of using a set of techniques as countermeasures to forensic analysis. Anti-forensics is more than technology. It is an approach to criminal hacking that can be summed up like this: Make it difficult for them to identify you and almost impossible for them to prove they identified you [2]. Anti-forensics or anti-investigation techniques are the techniques that aim to make the digital investigation process very slow, expensive, and unreliable. The adversary uses these methods or measures to make the whole investigation difficult and inefficient for the flesh-and-blood investigators.

Table 1: Anti-Forensic Techniques

ANTI-FORENSIC TECHNIQUES	TYPES	SUB-TYPE
Data Hiding	Steganography	Audio Steganography
		Video Steganography
		Image Steganography
		File Steganography
		Text Steganography
	Encryption	Application Encryption
		Cloud Service Encryption
		Network Protocol Encryption
		Database Encryption
		File Encryption
		Folder Encryption
		Mobile Device Encryption
		Portable Disk Encryption
	File System Manipulation	Alternate data stream Utilization
		Slack Space Hiding
	Hard Disk Manipulation	DCO Area
		HPA Area
Artifact Wiping	Disk degaussing / damaging Techniques	Disk Wiping
		File Wiping
		Log Wiping
		Metadata Wiping
		Registry Wiping
Trail obfuscation	Data Fabrication	
	Data Framing/Misleading	
	Data Obscurity	
	Log Manipulation	

4. Proposed System

The File System plays a very important role in describing the format and structure of the hard drive or how a typical hard drive stores files. The file system tells us how the data is stored on the hard drive, i.e., it provides some structure for arranging or storing the information in a storage medium. Here, we have considered the Secondary Extended File System of the Linux Distribution.

Preferred working on Linux, besides other operating systems, because Linux provides the most flexible and reliable working environment, and even host-based forensic tools often run on Linux platforms [22].

The Proposed tool answers this question of how to apply Anti-Forensic techniques on the File System of the Linux distribution and how to combat the threat of falling into the hands of Forensic Investigators by messing with the internal structure of the file system or by clearing out all the information related to the evidential files contained in the inodes of the file system. In the Linux system, the whole hard disk is divided into some partitions in order to store the data, i.e. the entire content on the hard disk is grouped and stored on the various partitions.

Every partition on the hard disk has its separate file system. The hard disk's partitions can be attached to the system through a mount point. A secondary extended file system, or Ext2fs, is a widely used file system in Linux, as it is one of the most robust file systems and has an excellent performance level. In the structure of Ex2fs, a disk or partition is divided into sectors or sections, known as blocks, which are further grouped to form block groups. Now, each block group contains important file system blocks such as Superblock, File System Descriptor, Block bitmap, Inode Bitmap, Inode Table and Data blocks, which are further described in detail in the following sections.

- In our proposed work, first, we have written an Anti-Forensic crawler in Python which automatically crawls all the possible URLs of the web pages related to Anti-Forensics tool and techniques, i.e., we now do not need to search the web pages exhaustively.
- We extracted the different Anti-Forensics tools running on various platforms from those web pages.
- Then, we created a dataset of 272 anti-forensics tools along with their corresponding Anti-forensics category and platforms. Amongst them, 149 Anti-forensic tools run on the Windows platform, 38 on the Linux platform, and 27 on the multi-platform.
- From the Dataset created, we analysed that Necrofile and Klismafile are the tools which deal with the file System of the Linux Platform, and a comparison of these tools is explained in the later part of this paper.
- Then we developed an Anti-Forensic tool for the file systems of Linux distributions, in which inodes which contain the crucial information about the file can be cleared, so that the evidential file cannot be opened, as its inodes get cleared by the tool.

In our proposed tool, first and foremost, the thing to get cleared out is that we are working in a controlled environment. So, when any of the hard disks or bootable pen drives containing the Linux distribution are inserted into the system, we create a partition of our own before installing the operating system on our machine. Then, we mount that particular partition to create a structured file system. After that, we run the Find command, which finds all the files and folders already in that file system. Then, to wipe out all the data or content in those evidential files, we have written zeroes to all the possible hard disk locations. After this, we formatted our ext2fs, and after formatting, there will no longer be any file structure in the file system. So, to obfuscate or mislead the investigator, we have mounted it to get the file structure the same as the previous one. After returning all the files and folders, we have created a mess with the inodes of those evidential files.

5. Linux File Systems

"On a Linux System, everything is a file; if something is not a file, it is a process". This statement is true because there are special files that are more than just files (for example, pipes, sockets etc.), but to keep things simple, saying that everything is a file is an acceptable generalization [16].

Linux File System has a hierarchical tree like structure. In the Linux System, all the programs, texts, images, etc., are all files, and even the Input and Output devices are considered files. So, in order to manage such different types of files on the hard disk, Linux has a tree like structure.

There are various types of files in the whole File System, such as Regular files, Directories, Special files, links, Sockets and named pipes.

- **Partitioning**

In the Linux System, the whole Hard disk is divided into some sectors or partitions to store the data securely, i.e., the entire hard disk is grouped and separated and then stored on the various partitions. Partitioning is achieved to protect the entire content or to achieve security in case of any destruction, corruption or criminal activity. For instance, if an intruder writes a malicious script or program that tries to fill the hard disk, the whole disk will stop working or functioning if it gets too full. And if the hard disk is partitioned, the partition with the malicious script will get affected, and all the other partitions can run securely and safely without any malicious impact.

- **Mount Points**

All the hard disk partitions can be attached to the system through a mount point. Mount Point tells us about the location of particular content or data in the File System. Generally, the root partition is the parent of all the partitions, and all the others are connected. Then, directories are created on this root partition, which is indicated with the slash (/). These empty directories will be the starting point of all the partitions that are attached to them [17].

Every partition on the hard disk has its separate file system. The File System we are dealing with here is the Secondary Extended File System.

- **Secondary Extended File System**

The Secondary Extended File System, or ext2 file system, was written by Remy Card, Theodore Ts'o and Stephen Tweedie as a major rewrite of the extended file system. It was released in January 1993 as part of the Linux Kernel [17].

Ext2 file system has the functionality of an extended file system, while its internal structures are also maintained. Ext3 and Ext4 are the new File Systems developed based on Ext2 and have some additional functions. Still, the Ext2 file system is the most preferred one among all the other file systems as it needs very few write operations (as it does not have a journal). This file system is the most commonly used file system in Linux distributions, providing many additional functionalities. Key features of this file system are that it is one of the robust file systems, and its performance level is also excellent. Secondary Extended File System includes standard UNIX functionalities or features and additional extensions. This file system includes regular files, directories, special files, links, etc.

- **Secondary Extended File System Structure**

A disk or partition is divided into some sectors or sections, which are known as blocks. Then these blocks are further grouped to form block groups in order to minimize or reduce the fragmentation of files on the whole disk randomly at different locations. Therefore, block groups play an important role in the file system structure shown in Figure 2.

The structure of the ext2 file system consists of the Boot Sector, Block Group 1, Block Group 2, followed by the other block groups till Block Group N-1, then finally Block Group N, as shown in Figure 3. Each block group contains important and crucial file system blocks such as Superblock, File System Descriptor table, Block Bitmap, Inode Bitmap, Inode Table and Data Blocks. All these blocks directly control the whole file system. All the block groups in the file system structure have a similar structure, i.e., all the blocks present in one block group are replicated to the other. The structure of those will always be the same, but the content of those blocks might be different, as content may vary according to which block group they belong to.

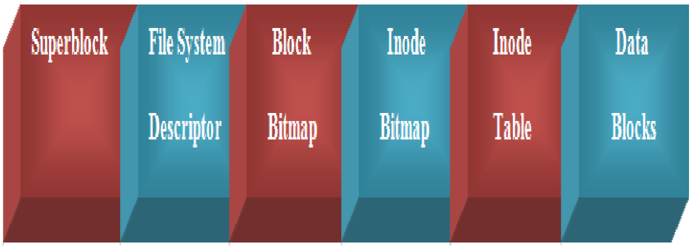


Figure 2: Structure of each Block Group

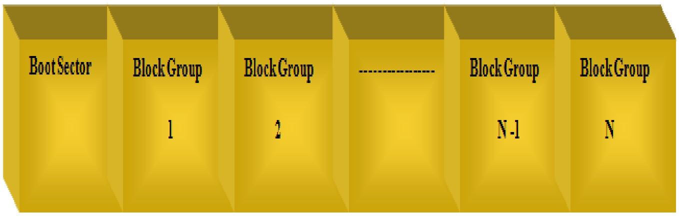


Figure 3: Structure of Ext2 File System

- **Superblock**

This block contains all the necessary information about the File System Configuration. Superblock of the block group contains information such as the shape and size of the File system.

Superblock of the very first Block Group, i.e., 0^{th} block group, is read during the mounting of the file system. Still, every block group do contain a replica of this superblock to deal with the situations when the superblock of the first block group gets corrupted.

Table 2: Information in Superblock

Info	**Description**
s_inodes_count	Gives the total inodes present on the file system
s_blocks_count	Gives the total blocks present on the file system
s_free_blocks_count	Gives the total free blocks available on the file system
s_free_inodes_count	Gives the total free inodes available on the file system
s_first_data_block	Gives the position or location of the first block on the file system
s_log_block_size	Calculates the logical size of the block in bytes
s_blocks_per_group	Gives the total blocks present in one block group

- **File System Descriptors**

File System Descriptor block immediately following the Superblock in the Block Group structure. This block contains the list, which describes that particular group. This block includes the list, which describes that specific group. Information includes total free blocks present in the group, total free inodes present in the group, total count of the inodes allocated to the directories, etc.

- **Block Bitmap**

The Secondary Extended File System uses Block Bitmaps and Inode Bitmaps to track the total number of blocks and inodes allocated. Block Bitmap refers to or points to a particular block in the whole block group, i.e., each block of the block group is represented by a bit. For instance, a 0 bit value refers to the block being free, and a 1 bit value represents the block being used.

If a bit of any particular block has to be identified, then first of all, we need to identify the block group of that block, and then the bit of that block can be identified by looking into the block bitmap table of the block group

- **Inode Bitmap**

In order to track the total number of inodes allocated to the file system, the secondary Extended File System uses Inode Bitmaps contained in the block group. Inode Bitmap refers to or points to a particular inode in the whole block group, i.e., each inode of the block group is represented by a bit, for instance, a 0 bit value refers to the fact that the inode is free, and a 1 bit value represents that the inode is being used. If a bit of any particular inode has to be identified, then first of all, we need to identify the block group of that inode. Then the bit of that inode can be identified by looking into the inode bitmap table of the block group.

- **Inodes**

You store your information in a file, and the operating system stores the information about a file in an inode (sometimes called an inode number) [23]. As users tend to search our files in the system by their names, but the file system cannot recognise a file by its name, it needs an Inode number or an Index number. An operating system can identify the file's location we are searching for with the help of an inode number. An inode is metadata of the data [23]. The Operating System first needs to search the inode number of a particular file required to be accessed by the user from the inode table, i.e., the file cannot be accessed without the inode number of that file.

Table 3: Metadata contained in the Inode of any file

Info	**Description**
i_uid	Gives the UID of the file’s owner
i_size	Gives the size of the file in bytes
i_atime	Gives the last access time of the file
i_ctime	Gives the last recent time when the inode data of the file has been changed
i_mtime	Gives the last modification time when the modification of the file content took place
i_dtime	Gives the time when the deletion of the file took place
i_gid	Gives the gid of a particular file
i_links_count	Tells about the particular file that has been pointed to by how many total links
i_blocks	Gives the total count of blocks a particular file has been allocated with
i_block[EXT2_N_BLOCKS]	Tells about the 15 pointers to blocks (see below)
i_version	Gives the file version
i_file_acl	Gives the file’s access control list
i_dir_acl	Gives the directory’s access control list

6. Results and Analysis

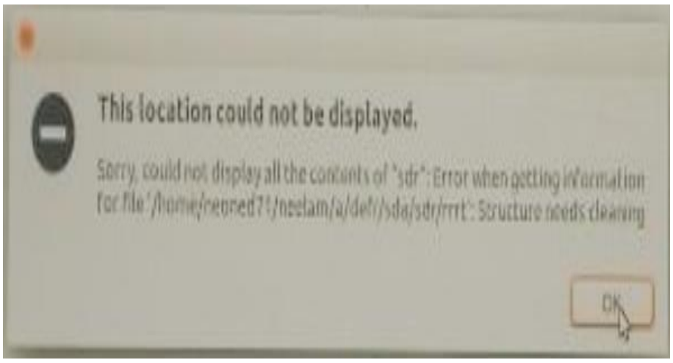
Table 4 provides a comparative overview of various anti-forensic tools, highlighting their platforms, techniques, and effectiveness. This contextualizes the performance of the proposed tool.

Table 4: Comparative Overview of Anti-Forensic Tools

Tool Name	**Platform**	**Anti-Forensic Technique**	**Effectiveness**
Necrofile	Linux	Inode Metadata Clearing	High
Klismafile	Linux	File Metadata Obfuscation	Moderate
Eraser	Windows	File Shredding	Low

BleachBit	Cross-Platform	Log and Cache Cleaning	Moderate
Proposed Tool	Linux	Inode & Data Wiping	Very High

- Cleared out all the inode information of evidential files.
- Also wiped out the content of those files.
- During Forensic examination, the investigator will be able to detect that something is wrong when they try to analyse the file system. Still, it will not be possible for him to open those files; he will get the error whenever he tries to open them, as the inodes of that file have already been cleared out by the anti-forensic practitioner.
- Moreover, he will not be able to recover the content of those files again as the intruder has already overwritten those files before formatting the Secondary Extended File System.
- Hence, through our proposed tool, it is possible to efficiently clear out all the inodes of evidential files. When the inodes of those files get cleared, it is impossible to open those files as their inode information is no longer present on the system. We can also wipe out the data of those files before formatting the Ex2fs by overwriting zeroes on all the possible locations of the hard disk, due to which the content of the evidential files gets wiped out.



This location could not be displayed.

Sorry, could not display all the contents of "sdr": Error when getting information for file '/home/neoned71/neelam/a/def//sda/sdr/rrrt': Structure needs cleaning

Figure 4: Error Message

Upon opening the file, an error was displayed in Figure 4, and we could not open the file, i.e. after clearing out Inodes of this file state, “The authors declare no conflict of interest.”

7. Conclusion

From the research work conducted specifically in the field of Anti-Forensics and the developed proposed tool, it can be concluded that as the advancements in the field of Forensic Sciences is increasing day by day, field of Anti-Forensics is also becoming technically strong consecutively to thwart the practice of Forensic Science and to hinder the whole Digital Forensic Investigation Process, hence making it more complex and time consuming for the investigators to investigate. Most of the time, Forensics software or a forensics toolkit doesn’t look into the internal structures of the file system of any operating system to find the evidence. So, it becomes very easy for the Anti-Forensics people to perform their malicious activities on such parts of the system. In our proposed framework, it is possible to clear out all the inodes of evidential files very efficiently. When the inodes of those files get cleared, it is impossible to open those files as their inode information is no longer present on the system. We can also wipe out the data of those files before formatting the Ex2fs by overwriting zeroes on all the possible locations of the hard disk, due to which the content of the evidential files gets wiped out.

Funding source

No funding was received for this study.

Conflict of Interest

There is no conflict of interest.

References

[1] Yusuf, Y., Ismail, R., & Hassan, Z. (2011). Common phases of computer forensics investigation models. *International Journal of Computer Science & Information Technology (IJCSIT), 3*(3), 17-31.

[2] Agarwal, A., Gupta, M., Gupta, S., & Gupta, S. C. (2011). Systematic digital forensic investigation model. *International Journal of Computer Science and Security (IJCSS), 5*(1), 118- 131.

[3] Slay, J., Lin, Y. C., Turnbull, B., Beckett, J., & Lin, P. (2009, January). Towards a formalization of digital forensics. In *IFIP International Conference on Digital Forensics* (pp. 37-47). Springer Berlin Heidelberg.

[4] Rekhis, S., & Boudriga, N. (2012). A system for formal digital forensic investigation aware of anti-forensic attacks. *IEEE transactions on Information Forensics and Security*, *7*(2), 635-650.

[5] Barske, D., Stander, A., & Jordaan, J. (2010, August). A digital forensic readiness framework for South African SME's. In *2010, Information Security for South Africa* (pp. 1-6). IEEE.

[6] Grobler, C. P., Louwrens, C. P., & von Solms, S. H. (2010, February). A multicomponent view of digital forensics. In *Availability, Reliability, and Security, 2010. ARES'10 International Conference on* (pp. 647-652). IEEE.

[7] Alharbi, S., Weber-Jahnke, J., & Traore, I. (2011, August). The proactive and reactive digital forensics investigation process: A systematic literature review. In *International Conference on Information Security and Assurance* (pp. 87-100). Springer Berlin Heidelberg.

[8] Jain, A., & Chhabra, G. S. (2014, August). Anti-forensics techniques: an analytical review. In *Contemporary Computing (IC3), 2014 Seventh International Conference on* (pp. 412-418). IEEE.

[9] Kessler, G. C. (2007, March). Anti-forensics and the digital investigator. In *Australian Digital Forensics Conference* (p. 1).

[10] Pajek, P., & Pimenidis, E. (2009, September). Computer anti-forensics methods and their impact on computer forensic investigation. In *International Conference on Global Security, Safety, and Sustainability* (pp. 145-155). Springer Berlin Heidelberg.

[11] Rekhis, S., & Boudriga, N. (2010, May). Formal digital investigation of antiforensic attacks. In *Systematic Approaches to Digital Forensic Engineering (SADFE), 2010 Fifth IEEE International Workshop on* (pp. 33-44). IEEE.

[12] Stamm, M. C., Lin, W. S., & Liu, K. R. (2012, March). Forensics vs. antiforensics: A decision and game theoretic framework. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 1749- 1752). IEEE.

[13] Dahbur, K., & Mohammad, B. (2011, April). The anti-forensics challenge. In *Proceedings of the 2011 International Conference on Intelligent Semantic Web- Services and Applications* (p. 14). ACM.

[14] Geiger, M. (2006, June). Counter-forensic tools: Analysis and data recovery. In *18th Annual FIRST Conference, Maltimore, Maryland* (pp. 25-30).

[15] http://e2fsprogs.sourceforge.net/ext2intro.html

[16] http://www.science.unitn.it/~fiorella/guidelinux/tlk/node95.html

[17] http://teaching.csse.uwa.edu.au/units/CITS2002/fs-ext2/

[18] Carrier, Brian. *File system forensic analysis*. Addison Wesley Professional, 2005.

[19] Liu, Dale. *Cisco router and switch forensics: Investigating and analyzing malicious network activity*. Syngress, 2009.

[20] Bilby, Darren. "Low down and dirty: anti-forensic rootkits." *Proceedings of Ruxcon* 2006 (2006).

[21] Botas, Álvaro, et al. "Counterfeiting and Defending the Digital Forensic Process." *Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM), 2015 IEEE International Conference on*. IEEE, 2015.

[22] https://www.slideshare.net/santoshkhadsare/linux- forensics-15854317
[23] http://www.slashroot.in/inode-and-its-structure- linux
[24] Conti, M., Dehghantanha, A., Franke, K., & Watson, S. (2018). 'Internet of Things security and forensics: Challenges and opportunities.' Future Generation Computer Systems, 78, 544–546. doi:10.1016/j.future.2017.07.060
[25] Quick, D., & Choo, K. K. R. (2018). 'Digital forensic intelligence: Data subsets and open source intelligence (DFINT+OSINT): A timely and cohesive mix.' Future Generation Computer Systems, 79, 584–595. doi:10.1016/j.future.2017.06.043
[26] Al Mutawa, N., Baggili, I., & Marrington, A. (2016). 'Forensic analysis of social networking applications on mobile devices.' Digital Investigation, 9, S24–S33. doi:10.1016/j.diin.2016.06.004
[27] Faheem, M., Gungor, V. C., & Koçak, T. (2020). 'A survey on secure communication and authentication in IoT forensics and anti-forensics.' Journal of Network and Computer Applications, 162, 102630. doi:10.1016/j.jnca.2020.102630
[28] Karpisek, F., Baggili, I., & Breitinger, F. (2019). 'WhatsApp network forensics: Decrypting and understanding the WhatsApp call signaling messages.' Digital Investigation, 29, S66–S76. doi:10.1016/j.diin.2019.04.008