УДК 343.98 DOI: 10.18698/2541-8009-2021-05-696

ОСОБЕННОСТИ КРИМИНАЛИСТИЧЕСКОГО ИССЛЕДОВАНИЯ ФАЙЛОВОЙ СИСТЕМЫ NTFS

А.С. Коваленко

kovalenkoas@student.bmstu.ru
SPIN-код: 1225-0528

МГТУ им. Н.Э. Баумана, Москва, Российская Федераци

Аннотация

Статья посвящена поиску и анализу криминалистически значимой информации на уровне файловой системы. Рассмотрены некоторые основные концепции, структуры данных и принципы работы одной из наиболее распространенных файловых систем New Technology File System (NTFS), являющейся стандартом для операционных систем семейства Windows. Проанализирована структура основных атрибутов, таких как $STANDARD_INFORMATION, $FILE_NAME, $OBJECT_ID и $DATA. Описан эксперимент по декодированию файловой записи в главной файловой таблице (Master File Table, MFT), в ходе которого была получена информация, позволяющая идентифицировать файл. Данный способ также может быть применен при восстановлении удаленных или поврежденных данных.

Ключевые слова

Файловая система, NTFS, MFT, атрибуты файла, файловая запись, экспертиза, криминалистическое исследование, судебная компьютерно-техническая экспертиза

Поступила в редакцию 20.04.2021

Актуальность статьи обусловлена тем, что исследование файловых систем является одним из важнейших направлений судебной компьютерно-технической экспертизы. В настоящее время в условиях развития науки и техники судебная компьютерно-техническая экспертиза становится все более востребована, поскольку именно экспертные технологии являются основным каналом применения современных достижений науки и техники в судебном процессе [1, 2]. Распространенной задачей эксперта при производстве судебной компьютерно-технической экспертизы является поиск и анализ цифровых следов, большинство из которых, как правило, содержится именно в файловой системе и в файлах. Одной из наиболее распространенных на сегодняшний день файловых систем является New Technology File System (NTFS), ставшая стандартом для операционных систем семейства Windows.

Поиск и анализ цифровых следов включает в себя несколько областей, которые могут быть представлены в виде основанной на архитектуре цифровых данных иерархии уровней анализа [3, с. 250]. Данная работа посвящена одному из верхних уровней такого анализа — анализу на уровне файловой системы.

Прежде всего необходимо раскрыть само понятие файловой системы. В соответствии с ГОСТ Р 5729–2017, под файловой системой следует понимать описа-

ние способа хранения, распределения, наименования и обеспечения доступа к информации, хранящейся на машинном носителе информации [4]. Иными словами, файловая система представляет собой набор структур данных, которые позволяют приложениям создавать, записывать и читать файлы [3, с. 251]. В свою очередь, файл — это поименованный набор данных, расположенный на машинном носителе информации [4].

Все служебные структуры в NTFS представлены файлами [5, с. 73]. В основе файловой системы NTFS лежит главная файловая таблица (Master File Table, MFT), которая содержит массив записей типа FILE Record, описывающих соответствующий файл или каталог. Главная файловая таблица (Master File Table) — база данных, хранящая информацию обо всех файлах раздела (в том числе о самой MFT). Именно в MFT содержится вся криминалистически значимая информация о файлах и каталогах [6, с. 22]. Описание файла или каталога может быть как полностью представлено одной записью в MFT, так и содержаться в нескольких таких записях. Структура подобной файловой записи представлена заголовком и атрибутами, размер файловой записи — 1024 байт [7, с. 101]. Данные об MFT хранятся в служебном файле $MFT.

Помимо $MFT в NTFS присутствуют другие файлы метаданных файловой системы, которые могут содержать значимую для проведения исследования информацию.

Важная особенность исследования состоит в том, что в файловой системе NTFS любой файл представлен как совокупность атрибутов (также называемых потоками), которые состоят из заголовка и тела атрибута. Эксперту также следует учитывать возможность наличия сжатых, зашифрованных или разреженных атрибутов [5, с. 75]. О таком свойстве атрибута сигнализирует ненулевое значение 16-разрядного поля флагов по смещению 0Ch от начала атрибутного заголовка [5].

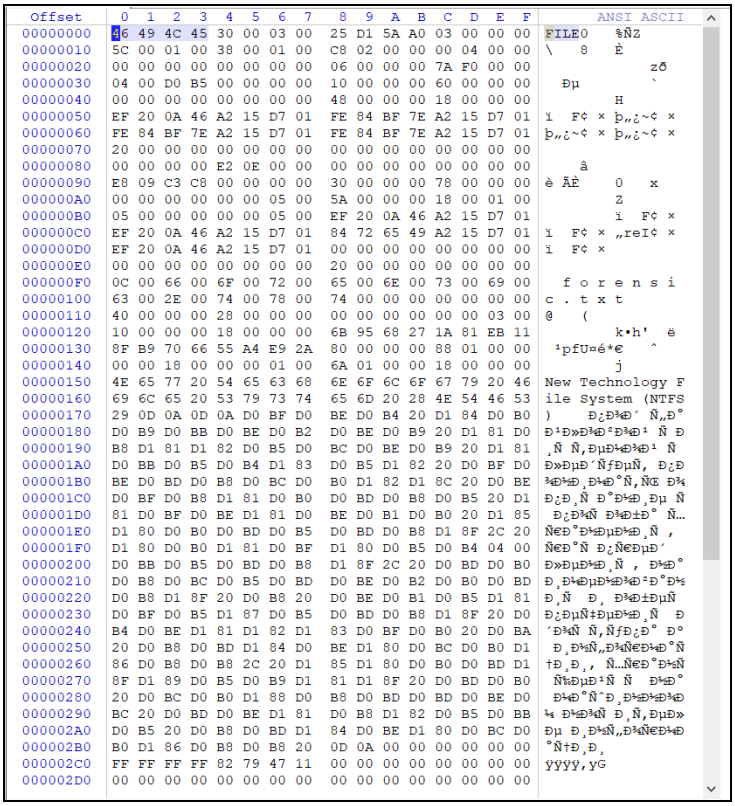
При производстве анализа на уровне файловой системы эксперту стоит принимать во внимание тот факт, что служебные структуры NTFS могут претерпевать определенные изменения в зависимости от версии файловой системы. В данной работе рассмотрена файловая система NTFS 5.0 в операционной системе Windows 10.

Неотъемлемой частью анализа на уровне файловой системы является умение эксперта работать с низкоуровневыми данными. Также немаловажной составляющей служит наличие навыков по проведению экспертного эксперимента, знание методических основ производства экспертизы [8, 9]. В этой связи в рамках данной работы был проведен эксперимент по декодированию файловой записи в MFT. В основе эксперимента лежал алгоритм декодирования файловой записи, предложенный Крисом Касперски [5].

Описанный Крисом Касперски алгоритм применялся для исследования файловой системы NTFS версии младше 3.0. В данной работе экспериментально проверена возможность работы данного способа для версии NTFS 5.0.

С помощью шестнадцатеричного редактора WinHex (версия 19.9) была просмотрена главная файловая таблица и найден один из секторов, содержащих

сигнатуру FILE (46 49 4C 45). Файловая запись в MFT занимает 1024 байт (2 сектора по 512 байт). Для удобства дальнейшего исследования соответствующие сектора были скопированы в отдельный блок (рис. 1). Расположенное по смещению 16h от начала исследуемого сектора 16-разрядное поле содержит значение 01h из чего можно сделать вывод, что данная файловая запись описывает файл и данный файл не был удален. Следующее, что необходимо проанализировать, — является ли данная файловая запись базовой или же представляет собой продолжение другой файловой записи. За это отвечает 64-разрядное поле по смещению 20h, которое для исследуемой файловой записи равно нулю, следовательно, файловая запись является базовой [10].



Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ANSI ASCII
00000000	46	49	4C	45	30	00	03	00	25	D1	5A	A0	03	00	00	00	FILE0 %ÑZ
00000010	5C	00	01	00	38	00	01	00	C8	02	00	00	00	04	00	00	\ 8 È
00000020	00	00	00	00	00	00	00	00	06	00	00	00	7A	F0	00	00	zð
00000030	04	00	D0	B5	00	00	00	00	10	00	00	00	60	00	00	00	Ðµ `
00000040	00	00	00	00	00	00	00	00	48	00	00	00	18	00	00	00	H
00000050	EF	20	0A	46	A2	15	D7	01	FE	84	BF	7E	A2	15	D7	01	ï F¢ × þ„¿~¢ ×
00000060	FE	84	BF	7E	A2	15	D7	01	FE	84	BF	7E	A2	15	D7	01	þ„¿~¢ × þ„¿~¢ ×
00000070	20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000080	00	00	00	00	E2	0E	00	00	00	00	00	00	00	00	00	00	â
00000090	E8	09	C3	C8	00	00	00	00	30	00	00	00	78	00	00	00	è ÃÈ 0 x
000000A0	00	00	00	00	00	00	05	00	5A	00	00	00	18	00	01	00	Z
000000B0	05	00	00	00	00	00	05	00	EF	20	0A	46	A2	15	D7	01	ï F¢ ×
000000C0	EF	20	0A	46	A2	15	D7	01	84	72	65	49	A2	15	D7	01	ï F¢ × „reI¢ ×
000000D0	EF	20	0A	46	A2	15	D7	01	00	00	00	00	00	00	00	00	ï F¢ ×
000000E0	00	00	00	00	00	00	00	00	20	00	00	00	00	00	00	00	
000000F0	0C	00	66	00	6F	00	72	00	65	00	6E	00	73	00	69	00	f o r e n s i
00000100	63	00	2E	00	74	00	78	00	74	00	00	00	00	00	00	00	c . t x t
00000110	40	00	00	00	28	00	00	00	00	00	00	00	00	00	03	00	@ (
00000120	10	00	00	00	18	00	00	00	6B	95	68	27	1A	81	EB	11	k•h' ë
00000130	8F	B9	70	66	55	A4	E9	2A	80	00	00	00	88	01	00	00	¹pfU¤é*€ ^
00000140	00	00	18	00	00	00	01	00	6A	01	00	00	18	00	00	00	j
00000150	4E	65	77	20	54	65	63	68	6E	6F	6C	6F	67	79	20	46	New Technology F
00000160	69	6C	65	20	53	79	73	74	65	6D	20	28	4E	54	46	53	ile System (NTFS
00000170	29	0D	0A	0D	0A	D0	BF	D0	BE	D0	B4	20	D1	84	D0	B0	) Ð¿Ð¾Ð´ Ñ„Ð°
00000180	D0	B9	D0	BB	D0	BE	D0	B2	D0	BE	D0	B9	20	D1	81	D0	Ð¹Ð»Ð¾Ð²Ð¾Ð¹ Ñ Ð
00000190	B8	D1	81	D1	82	D0	B5	D0	BC	D0	BE	D0	B9	20	D1	81	¸Ñ Ñ‚Ðµ Ð¾Ð¹ Ñ
000001A0	D0	BB	D0	B5	D0	B4	D1	83	D0	B5	D1	82	20	D0	BF	D0	Ð»ÐµÐ´ÑƒÑ‚ Ð¿Ð
000001B0	BE	D0	BD	D0	B8	D0	BC	D0	B0	D1	82	D1	8C	20	D0	BE	¾Ð½Ð¸Ð°Ñ‚ÑŒ Ð¾
000001C0	D0	BF	D0	B8	D1	81	D0	B0	D0	BD	D0	B8	D0	B5	20	D1	Ð¿Ð¸Ñ Ð°Ð½Ð¸Ðµ Ñ
000001D0	81	D0	BF	D0	BE	D1	81	D0	BE	D0	B1	D0	B0	20	D1	85	Ð¿Ð¾Ñ Ð¾Ð±Ð° Ñ…
000001E0	D1	80	D0	B0	D0	BD	D0	B5	D0	BD	D0	B8	D1	8F	2C	20	Ñ€Ð°Ð½Ðµ Ð¸Ñ ,
000001F0	D1	80	D0	B0	D1	81	D0	BF	D1	80	D0	B5	D0	B4	04	00	Ñ€Ð°Ñ Ð¿Ñ€ÐµÐ´
00000200	D0	BB	D0	B5	D0	BD	D0	B8	D1	8F	2C	20	D0	BD	D0	B0	Ð»ÐµÐ½Ð¸Ñ , Ð½Ð°
00000210	D0	B8	D0	BC	D0	B5	D0	BD	D0	BE	D0	B2	D0	B0	D0	BD	Ð¸Ð¼ÐµÐ½Ð¾Ð²Ð°Ð½Ð½
00000220	D0	B8	D1	8F	20	D0	B8	20	D0	BE	D0	B1	D0	B5	D1	81	Ð¸Ñ Ð¸ Ð¾Ð±ÐµÑ
00000230	D0	BF	D0	B5	D1	87	D0	B5	D0	BD	D0	B8	D1	8F	20	D0	Ð¿ÐµÑ‡Ð°Ñ‚Ð¸Ñ Ð
00000240	B4	D0	BE	D1	81	D1	82	D1	83	D0	BF	D0	B0	20	D0	BA	´Ð¾Ñ Ñ‚ÑƒÐ¿Ð° Ðº
00000250	20	D0	B8	D0	BD	D1	84	D0	BE	D1	80	D0	BC	D0	B0	D1	Ð¸Ð½Ñ„Ð¾Ñ€Ð¼Ð°Ñ
00000260	86	D0	B8	D0	B8	2C	20	D1	85	D1	80	D0	B0	D0	BD	D1	†Ð¸Ð¸, Ñ…Ñ€Ð°Ð½Ñ
00000270	8F	D1	89	D0	B5	D0	B9	D1	81	D1	8F	20	D0	BD	D0	B0	Ñ‰ÐµÐ¹Ñ Ñ Ð½Ð°
00000280	20	D0	BC	D0	B0	D1	88	D0	B8	D0	BD	D0	BD	D0	BE	D0	Ð½Ð°Ñ^Ð¸Ð½Ð½Ð¾Ð
00000290	BC	20	D0	BD	D0	BE	D1	81	D0	B8	D1	82	D0	B5	D0	BB	¼ Ð½Ð¾Ñ Ð¸Ñ,ÐµÐ»
000002A0	D0	B5	20	D0	B8	D0	BD	D1	84	D0	BE	D1	80	D0	BC	D0	Ðµ Ð¸Ñ„Ð¾Ñ€Ð¼Ð
000002B0	B0	D1	86	D0	B8	D0	B8	20	0D	0A	00	00	00	00	00	00	°Ñ†Ð¸Ð¸
000002C0	FF	FF	FF	FF	82	79	47	11	00	00	00	00	00	00	00	00	ÿÿÿÿ‚yG
000002D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	

Рис. 1. Фрагмент окна программы WinHex. Исследуемая файловая запись

Данные о смещении первого атрибута файловой записи расположены в 16-разрядном поле по смещению 14h [11]. Значение данного поля для исследуемой записи равно 38h, следовательно, заголовок первого атрибута начинается со смещения 38h от начала сектора.

По смещению 38h находим значение 10h, из чего можно заключить, что тип первого атрибута $STANDARD_INFORMATION [11]. Для того чтобы вычислить смещение для следующего атрибута, необходимо установить длину атрибута $STANDARD_INFORMATION. За длину отвечает 32-разрядное поле, расположенное по смещению 04h от начала атрибута, значение которого составляет 60h. Смещение следующего атрибута вычисляют путем сложения смещения

предыдущего атрибута с его длиной: 38h + 60h = 98h. Следовательно, заголовок следующего атрибута начинается по смещению 98h от начала сектора.

По смещению 98h обнаружено значение 30h, что соответствует типу атрибута $FILE_NAME [11]. Как и в предыдущем случае, длина атрибута хранится в 32-разрядном поле по смещению 04h от начала атрибута. Для атрибута $FILE_NAME длина составляет 78h. Соответственно, смещение следующего атрибута: 98h + 78h = 110h.

По смещению 110h обнаружено значение 40h, следовательно, тип атрибута — $OBJECT_ID [11]. Длина атрибута находится аналогично первым двум случаям и составляет 28h. Смещение следующего атрибута 110h + 28h = 138h.

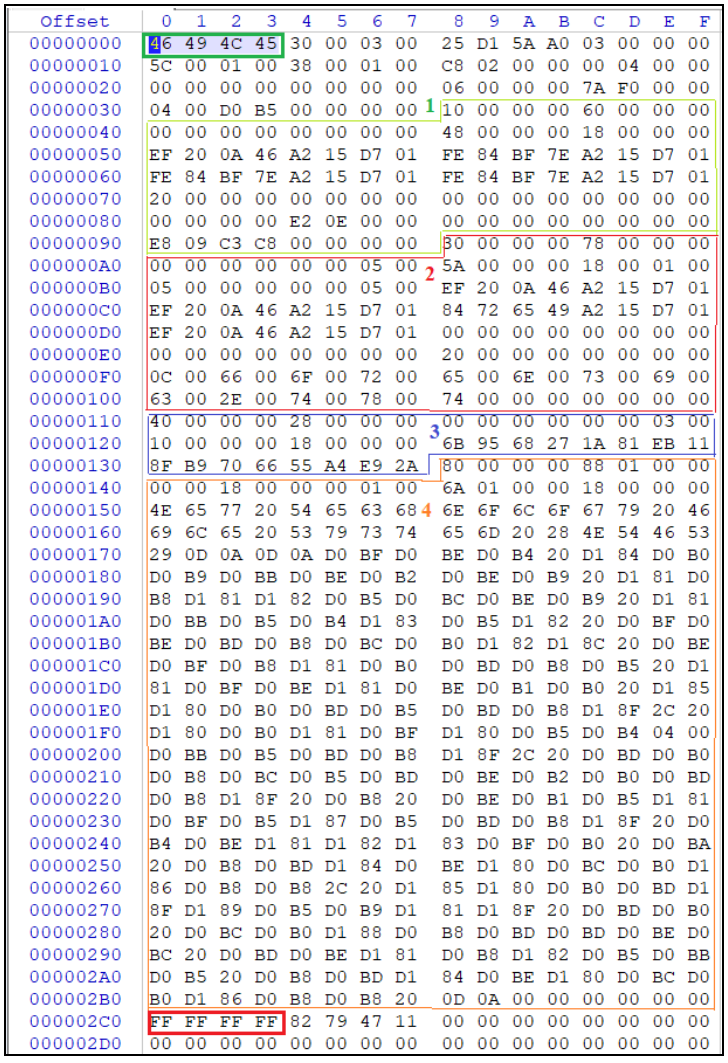


Рис. 2. Фрагмент окна программы WinHex. Атрибуты в исследуемой файловой записи:

1 — $STANDARD_INFORMATION; 2 — $FILE_NAME; 3 — $OBJECT_ID; 4 — $DATA

По смещению 138h обнаружено значение 80h, что соответствует типу $DATA [11]. По смещению 04h в 32-разрядном поле содержится значение длины атрибута, которое составляет 0188h. Смещение следующего атрибута 138h + 188h = 2C0h. По смещению 2C0h находим последовательность FF FF FF FF, из чего следует, что атрибут $DATA последний в списке [12].

После того как в исследуемой файловой записи выделены все атрибуты (рис. 2), рассмотрим структуру каждого из них.

Атрибут $STANDARD_INFORMATION. Данный атрибут хранит общую информацию, такую как временные штампы, флаги, информацию о владельце [5, с. 258]. По смещению 08h от начала заголовка атрибута хранится значение 00h, следовательно, данный атрибут является резидентным, а тело атрибута полностью хранится в MFT. Отвечающее за значение флагов 32-разрядное поле по смещению 0Ch от начала атрибута имеет нулевое значение, следовательно, данный атрибут не является сжатым, зашифрованным или разреженным [11].

Атрибут $FILE_NAME. Также является резидентным (нулевое значение поля по смещению 08h от начала заголовка атрибута) и не является сжатым, зашифрованным или разреженным (поле флагов имеет нулевое значение). Сведения о длине имени файла находятся в 8-разрядном поле по смещению 40h от начала тела атрибута (E8h от начала сектора). Для исследуемого файла длина имени составляет 20h. В свою очередь, сведения об имени файла расположены по смещению 42h от начала тела атрибута (EAh от начала сектора) [11]. Следовательно, имя исследуемого файла forensic.txt.

Атрибут $OBJECT_ID. Аналогично рассмотренным ранее атрибутам является резидентным, не является зашифрованным, сжатым или разреженным. Поле данного атрибута, расположенное по смещению 0h от начала тела атрибута и 18h от заголовка атрибута, занимает 16 байт и содержит уникальный идентификатор, присвоенный файлу (B9 8F 11 EB 81 1A 27 68 95 6B 00 00 00 18 00 00) [11].

Атрибут $DATA. Данный атрибут хранит непосредственно само содержимое файла, не имеет фиксированного формата или заранее определенных значений [5, с. 288]. Для исследуемого файла атрибут является резидентным (значение поля по смещению 08h от начала заголовка атрибута имеет нулевое значение), следовательно, содержимое исследуемого файла полностью хранится в MFT. Тело атрибута не является зашифрованным, сжатым или разреженным (16-разрядное поле флагов по смещению 0Ch от начала атрибута имеет нулевое значение). Тело атрибута (т. е. содержащиеся в исследуемом файле данные) начинается со смещения 18h от начала заголовка атрибута [11] (рис. 3).

Отметим, что в случае нерезидентного атрибута для установления расположения содержимого файла необходимо декодировать список отрезков, расположенный по смещению 20h от начала атрибутного заголовка в 16-разрядном поле [11].

При работе с данным атрибутом эксперту также стоит учитывать возможность сокрытия криминалистически значимой информации в скрытых потоках — дополнительных атрибутах $DATA.

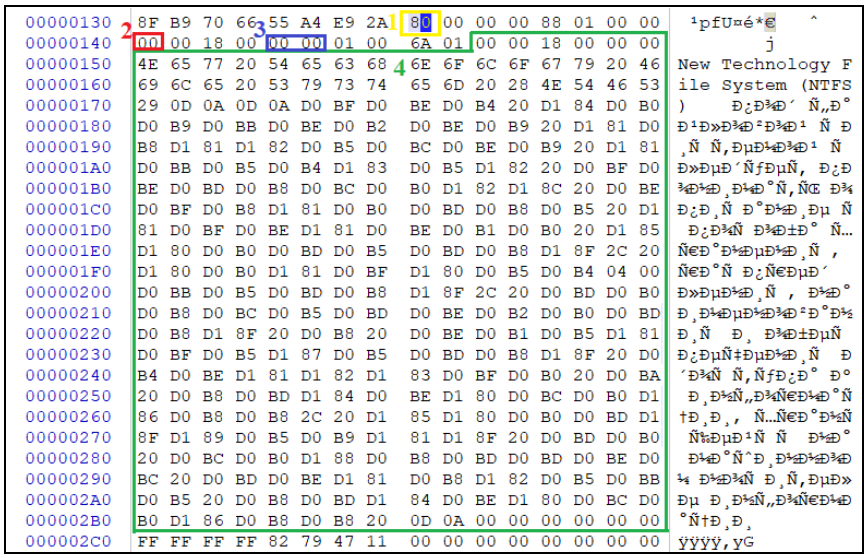


Рис. 3. Фрагмент окна программы WinHex. Атрибут $DATA:

1 — тип атрибута; 2 — флаг, указывающий, что атрибут резидентный; 3 — флаг (нулевое значение указывает на то, что атрибут не является зашифрованным, сжатым или разреженным; 4 — тело атрибута (содержимое файла)

Таким образом, в ходе декодирования файловой записи установлено, что данная запись соответствует файлу с именем forensic.txt, содержимое которого полностью расположено в самой таблице $MFT и не является зашифрованным, сжатым или разреженным.

Исследования на уровне файловой системы играют важную роль при получении криминалистически значимой информации. Умение выполнять декодирование файловой записи может пригодиться эксперту для восстановления удаленных и поврежденных файлов, а также для получения информации, идентифицирующей файл. Кроме того, при использовании прикладных программ анализа файловой системы NTFS эксперт должен понимать принцип их работы. В связи с этим эксперту в области судебной компьютерно-технической экспертизы необходимо владеть знаниями об основных концепциях и структурах данных файловых систем, в частности, одной из наиболее распространенных на сегодняшний день файловых систем — NTFS.

Литература

[1] Омельянюк Г.Г, Усов А.И. Тенденции развития судебно-экспертной деятельности: вызовы времени и решения. Фундаментальные и прикладные исследования в сфере судебно-экспертной деятельности и ДНК-регистрации населения РФ. Мат. Всерос. науч.-практ. конф. Уфа, БГУ, 2019, с. 205–212.

[2] Сафонова Н.А. К вопросу о применении цифровых технологий в гражданском процессе. Сб. науч. тр. конф. Юридическая наука в XXI веке: актуальные проблемы и перспективы их решений. М., Конверт, 2020, с. 89–91.

[3] Кэрриэ Б. Криминалистический анализ файловых систем. СПб., Питер, 2007.
[4] ГОСТ Р 57429-2017. Судебная компьютерно-техническая экспертиза. Термины и определения. М., Стандартинформ, 2018.
[5] Касперски К. Файловая система NTFS извне и изнутри. Системный администратор, 2004, № 11. URL: http://samag.ru/archive/article/375
[6] Волкова С.В., Карлова А.В. Криминалистический анализ атрибутов даты и времени информационных объектов. Modern Science, 2020, № 4-2, с. 21–27.
[7] Shaaban A., Sapronov K. Practical Windows forensics. Packt Publishing, 2016.
[8] Нехорошев А.Б., Шухнин М.Н., Юрин И.Ю. и др. Практические основы компьютерно-технической экспертизы. Саратов, Научная книга, 2007.
[9] Вехов В.Б., Ковалев С.А. Компьютерное моделирование при расследовании преступлений в сфере компьютерной информации. Волгоград, ВА МВД России, 2014.
[10] Касперски К. Восстановление данных на NTFS-разделах. Системный администратор, 2004, № 9. URL: http://samag.ru/archive/article/342
[11] NTFS documentation. ftp.kolibrios.org: веб-сайт. URL: http://ftp.kolibrios.org/users/Asper/ docs/NTFS/ntfsdoc.html (дата обращения: 10.03.2021).
[12] Файловая система NTFS. intuit.ru: веб-сайт. URL: https://intuit.ru/studies/professional_skill_improvements/ 10808/courses/1078/lecture/16586 (дата обращения: 10.03.2021).

Коваленко Анна Сергеевна — студентка кафедры «Цифровая криминалистика», МГТУ им. Н.Э. Баумана, Москва, Российская Федерация.

Научный руководитель — Купин Алексей Федорович, кандидат юридических наук, доцент кафедры «Цифровая криминалистика», МГТУ им. Н.Э. Баумана, Москва, Российская Федерация.

Ссылку на эту статью просим оформлять следующим образом:
Коваленко А.С. Особенности криминалистического исследования файловой системы NTFS. *Политехнический молодежный журнал*, 2021, № 05(58). http://dx.doi.org/ 10.18698/2541-8009-2021-05-696

FEATURES OF FORENSIC INVESTIGATION OF THE NTFS FILE SYSTEM

A.S. Kovalenko
kovalenkoas@student.bmstu.ru
SPIN-code: 1225-0528

Bauman Moscow State Technical University, Moscow, Russian Federation

Abstract

The article is devoted to the search and analysis of forensic information at the file system level. The paper considers some basic concepts, data structures and principles of operation of one of the most common file systems New Technology File System (NTFS), which is the standard for operating systems of the Windows family. Authors analyzed the structure of the main attributes, such as $STANDARD_INFORMATION, $FILE_NAME, $OBJECT_ID and $DATA. An experiment on decoding a file record in the Master File Table (MFT) is described, during which information was obtained that allows identifying a file. This method can also be applied when recovering deleted or damaged data.

Keywords

File system, NTFS, MFT, file attributes, file recording, forensics, forensic research, forensic computer-technical examination

Received 20.04.2021

References

[1] Omel'yanyuk G.G, Usov A.I. [Development tendencies of forensics: time issues and solutions]. Fundamental'nye i prikladnye issledovaniya v sfere sudebno-ekspertnoy deyatel'nosti i DNK-registratsii naseleniya RF. Mat. Vseros. nauch.-prakt. konf. [Fundamental and Applied Study in the field of forensic and DNA-registration of the Russian population. Proc. Russ. Sci.-Tech. Conf.]. Ufa, BGU, 2019, pp. 205–212 (in Russ.).

[2] Safonova N.A. [On the application of digital technologies in the civil process]. Sb. nauch. tr. conf. Yuridicheskaya nauka v XXI veke: aktual'nye problemy i perspektivy ikh resheniy. [Proc. Conf. Legal Science in XXI Century: Actual Problems and Prospects of Their Solution]. Moscow, Konvert Publ., 2020, pp. 89–91 (in Russ.).

[3] Carrier B. File system forensic analysis. Addison-Wesley, 2005. (Russ. ed.: Kriminalisticheskiy analiz faylovykh system. Sankt-Petersburg, Piter Publ., 2007.)

[4] GOST R 57429-2017. Sudebnaya komp'yuterno-tekhnicheskaya ekspertiza. Terminy i opredeleniya [State standard R 57429-2017. Forensic information technology examination. Terms and definitions]. Moscow, Standartinform Publ., 2018 (in Russ.).

[5] Kasperski K. Faylovaya sistema NTFS izvne i iznutri [NTFS file system inside and outside]. Sistemnyy administrator, 2004, no. 11. URL: http://samag.ru/archive/article/375 (in Russ.).

[6] Volkova S.V., Karlova A.V. Forensic analysis of date and time attributes for information objects. Modern Science, 2020, no. 4-2, pp. 21–27 (in Russ.).

[7] Shaaban A., Sapronov K. Practical Windows forensics. Packt Publishing, 2016.

[8] Nekhoroshev A.B., Shukhnin M.N., Yurin I.Yu., et al. Prakticheskie osnovy komp'yuterno-tekhnicheskoy ekspertizy [Practical basis of cyber forensics]. Saratov, Nauchnaya kniga Publ., 2007 (in Russ.).

[9] Vekhov V.B., Kovalev S.A. Komp'yuternoe modelirovanie pri rassledovanii prestupleniy v sfere komp'yuternoy informatsii [Computer modelling at cyber crime investigation]. Volgograd, VA MVD Rossii Publ., 2014 (in Russ.).
[10] Kasperski K. Restoring data in NFTS-sections. Sistemnyy administrator, 2004, no. 9. URL: http://samag.ru/archive/article/342 (in Russ.).
[11] NTFS documentation. ftp.kolibrios.org: website. URL: http://ftp.kolibrios.org/users/ Asper/ docs/ NTFS/ntfsdoc.html (accessed: 10.03.2021).
[12] Faylovaya sistema NTFS [NTFS file system]. intuit.ru: website (in Russ.). URL: https://intuit.ru/studies/professional_skill_improvements/10808/courses 1078/ lecture/16586 (accessed: 10.03.2021).

Kovalenko A.S. — Student, Department of Digital Forensics, Bauman Moscow State Technical University, Moscow, Russian Federation.

Scientific advisor — Kupin A.F., Cand. Sc. (Jur.), Assoc. Professor, Department of Digital Forensics, Bauman Moscow State Technical University, Moscow, Russian Federation.

Please cite this article in English as:
Kovalenko A.S. Features of forensic investigation of the NTFS file system. *Politekhnicheskiy molodezhnyy zhurnal* [Politechnical student journal], 2021, no. 05(58). http://dx.doi.org/ 10.18698/ 2541-8009-2021-05-696.html (in Russ.).