О целостности компьютерной информации во время и после изъятия ее носителя

Суханов Максим Андреевич *Версия документа: 0.96*

Роль целостности компьютерной информации

Исследование компьютерной информации, в общем случае, требует обеспечения ее целостности с целью сохранения технической возможности воспроизведения (повторения) проведенного исследования, а также проведения дополнительных исследований. Некоторые виды исследований компьютерной информации неизбежно сопряжены с нарушением целостности исследуемой компьютерной информации, вызванного процессом получения доступа к ней и ее чтением (например, исследования компьютерной информации в мобильных телефонах и иных мобильных устройствах).

Согласно действующему законодательству в области судебной экспертизы, эксперт вправе проводить разрушающие исследования только с разрешения органа или лица, назначившего судебную экспертизу. Поскольку целостность является одним из основных свойств компьютерной информации, то ее нарушение является достаточным основанием для признания соответствующего способа (метода) исследования разрушающим[1].

Разрешение на нарушение целостности компьютерной информации, хранимой в объектах исследования, в т. ч. в форме перечисления в постановлении о назначении судебной экспертизы конкретных разрешаемых эксперту действий («прямое включение»[2] объектов исследования и т. п.), часто дается следователями автоматически, без оценки практической необходимости такого нарушения целостности для исследования и без оценки возможных последствий такого нарушения целостности для доказывания. С практической точки зрения, такой подход позволяет, в одном случае, ускорить производство судебной экспертизы за счет исключения необходимости рассмотрения ходатайства эксперта о разрешении проведения разрушающего исследования и, в другом случае, предупредить отказ эксперта от ответа по существу на поставленные вопросы в отношении объектов, для которых требуется проведение подобного разрушающего исследования (если эксперт решит не заявлять указанное ходатайство и, тем самым, уменьшить объем своей работы). Вместе с тем, наличие такого разрешения в весьма широкой формулировке позволяет «узаконить» отклонение от общепринятых методик исследования компьютерной информации и применить разрушающее исследование там, где это объективно не требуется.

---

1 В частности, статья 57 Уголовно-процессуального кодекса РФ запрещает эксперту «проводить без разрешения дознавателя, следователя, суда исследования, могущие повлечь полное или частичное уничтожение объектов либо изменение их внешнего вида или основных свойств». Похожее положение есть и в статье 16 Федерального закона «О государственной судебно-экспертной деятельности в РФ».

2 Под «прямым включением» понимается включение объекта исследования без принятия мер по обеспечению целостности хранимой в нем компьютерной информации (примером такой меры является запуск объекта исследования с использованием клона носителя информации, а не оригинала).

Дискуссионными являются пределы необходимости давать разрешение на нарушение целостности компьютерной информации. По мнению автора, такое разрешение требуется для исследования мобильных телефонов во всех случаях (получение компьютерной информации из таких устройств почти всегда требует их «прямого включения»[3], либо «перепрошивки»[4], либо извлечения модулей памяти[5],а заявленные некоторыми производителями средств исследования «криминалистически правильные» способы копирования компьютерной информации из некоторых таких устройств требуют предварительных манипуляций, которые в случае неудачи часто приводят к «прямому включению»), но не требуется для исследования исправных накопителей на жестких магнитных дисках (включение и работа таких накопителей все же приводят к некоторым изменениям диагностических данных S.M.A.R.T.[6], однако такие изменения являются, в целом, несущественными, предсказуемыми и не попадающими под формулировки указанного ранее положения статьи 57 Уголовно-процессуального кодекса РФ и аналогичного положения в статье 16 Федерального закона «О государственной судебно-экспертной деятельности в РФ»).

Рекомендуется, по мнению автора, получение указанного разрешения на нарушение целостности компьютерной информации и при исследовании флеш-накопителей типа solid-state drive (известно, что в некоторых старых моделях таких флеш-накопителей на уровне контроллера реализована поддержка самостоятельной, без участия операционной системы компьютера, обработки структур файловой системы с целью поиска свободных (незанятых) блоков для их дальнейшего высвобождения в интересах процесса выравнивания изнашивания ячеек флеш-памяти, т. е. простое включение флеш-накопителя может привести к стиранию хранимых в нем удаленных данных; кроме того, не исключено отложенное исполнение команд высвобождения блоков данных[7], переданных на флеш-накопитель до его изъятия, или «обнуление» видимых операционной системе компьютера блоков данных в произвольный момент времени в случае реализации во флеш-накопителе недетерминированного чтения после высвобождения[8]). Например, флеш-накопитель Corsair P64 (CMFSSD-64GBG2D, 64GB, rev. VBM19C1Q), согласно результатам обратной разработки его микропрограммы автором, а также согласно опубликованному в 2010 году исследованию [1], имеет реализацию алгоритма обработки файловой системы NTFS.

3 Следует признать ошибочным мнение, что штатное включение современного мобильного телефона никакой компьютерной информации не модифицирует.

4 Записи на мобильное устройство другого образа операционной системы с целью преодоления некоторых мер защиты компьютерной информации.

5 Что может повредить модули памяти. В этом случае речь идет не только о нарушении целостности компьютерной информации, но и об изменении свойств объекта исследования как аппаратного устройства.

6 Например, данных о продолжительности работы накопителя и о количестве циклов включения-выключения электропитания.

7 Речь идет про команды ATA TRIM, SCSI UNMAP и т. п.

8 После того, как блок данных стал свободным (незанятым) и об этом было указано флеш-накопителю (до его изъятия), попытки дальнейшего чтения этого блока данных (в т. ч. после изъятия) могут возвращать в течение некоторого времени старые (ранее хранимые в этом блоке) данные, а после (без какой-либо записи в этот блок) – нулевые или иные байты.

```
ROM:0001EDF0 ; =============== S U B R O U T I N E =======================================
ROM:0001EDF0
ROM:0001EDF0
ROM:0001EDF0 ; signed int __fastcall CheckFileRecord(int a1, int FileRecord_Ptr, int a3)
ROM:0001EDF0 CheckFileRecord                         ; CODE XREF: sub_1F224+DA↓p
ROM:0001EDF0                 PUSH            {R4,LR}
ROM:0001EDF2                 MOVS            R4, R1
ROM:0001EDF4                 LDR             R1, [R2,#0xC]
ROM:0001EDF6                 LSLS            R0, R0, #1
ROM:0001EDF8                 ADDS            R0, R1, R0
ROM:0001EDFA                 MOVS            R2, R4
ROM:0001EDFC                 MOVS            R1, #2
ROM:0001EDFE                 BL              sub_1DD90
ROM:0001EE02                 CMP             R0, #0
ROM:0001EE04                 BNE             loc_1EE0A
ROM:0001EE06
ROM:0001EE06 loc_1EE06                               ; CODE XREF: CheckFileRecord+30↓j
ROM:0001EE06                 MOVS            R0, #0
ROM:0001EE08                 POP             {R4,PC}
ROM:0001EE0A ; ---------------------------------------------------------------------------
ROM:0001EE0A
ROM:0001EE0A loc_1EE0A                               ; CODE XREF: CheckFileRecord+14↑j
ROM:0001EE0A                 LDRB            R0, [R4]
ROM:0001EE0C                 CMP             R0, #0x46 ; 'F'
ROM:0001EE0E                 BEQ             loc_1EE14
ROM:0001EE10                 ADR             R0, aSDelMftDataIsW ; "\n[S-DEL] MFT Data is wrong.[Signature "...
ROM:0001EE12                 B               loc_1EE1C
ROM:0001EE14 ; ---------------------------------------------------------------------------
ROM:0001EE14
ROM:0001EE14 loc_1EE14                               ; CODE XREF: CheckFileRecord+1E↑j
ROM:0001EE14                 LDRH            R0, [R4,#0x16]
ROM:0001EE16                 CMP             R0, #1
ROM:0001EE18                 BEQ             loc_1EE22
ROM:0001EE1A                 ADR             R0, aSDelMftDataIsW_0 ; "\n[S-DEL] MFT Data is wrong. [It's not "...
ROM:0001EE1C
ROM:0001EE1C loc_1EE1C                               ; CODE XREF: CheckFileRecord+22↑j
ROM:0001EE1C                 BL              DebugPrint
ROM:0001EE20                 B               loc_1EE06
ROM:0001EE22 ; ---------------------------------------------------------------------------
ROM:0001EE22
ROM:0001EE22 loc_1EE22                               ; CODE XREF: CheckFileRecord+28↑j
ROM:0001EE22                 MOVS            R0, #1
ROM:0001EE24                 POP             {R4,PC}
ROM:0001EE24 ; End of function CheckFileRecord
```

Илл. 1: Функция проверки файловой записи NTFS в микропрограмме флеш-накопителя типа solid-state drive

Видна проверка на сигнатуру файловой записи (по адресу 0x1EE0C) и проверка на флаг «Is in use» (по адресу 0x1EE16).

Известно о существовании накопителей типа USB Flash, которые для блоков данных, в которые еще ни разу не производилась запись, возвращают в качестве их содержимого байты поступившей команды чтения, что приводит к тому, что попытки чтения такого блока данных каждый раз возвращают разные «данные» (либо, если операционная система компьютера использует одинаковые по своему содержимому команды чтения для одного и того же блока, «данные» будут различаться после переподключения накопителя)[9]; для таких накопителей получение разрешения на нарушение целостности компьютерной информации не требуется (если для противного нет иных причин), поскольку реального нарушения целостности (т. е. модификации хранимой компьютерной информации) не происходит.

9 Подобные «недетерминированные блоки» характеризуются наличием символов «USBC» (сигнатура блока команды, передаваемого по интерфейсу USB) в начале каждого прочитанного блока данных [2].

```
fuf@WS0008:~$ sudo dd if=/dev/sdd skip=100070 count=1 status=none | hexdump -C
00000000  55 53 42 43 6f 00 00 00  00 10 00 00 80 00 0a 28  |USBCo..........(|
00000010  00 00 01 86 e0 00 00 08  00 00 00 00 00 00 00 e9  |................|
00000020  00 e9 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000030  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000200
fuf@WS0008:~$ sudo dd if=/dev/sdd skip=100070 count=1 status=none | hexdump -C
00000000  55 53 42 43 72 00 00 00  00 10 00 00 80 00 0a 28  |USBCr..........(|
00000010  00 00 01 86 e0 00 00 08  00 00 00 00 00 00 00 e8  |................|
00000020  00 e8 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000030  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000200
fuf@WS0008:~$
```

Илл. 2. Пример чтения данных из одного и того же сектора в составе блока, в который еще ни разу не производилась запись (накопитель типа USB Flash)

При даче разрешения эксперту на нарушение целостности компьютерной информации и при проведении разрушающих исследований самим экспертом (а также при проведении осмотра с участием специалиста и иных подобных процессуальных действий) рекомендуется учитывать следующее:

- нарушение целостности компьютерной информации целесообразно допускать в разумных (но не только в минимально возможных[10]) пределах, обусловленных необходимостью получения доступа к компьютерной информации и ее чтения (копирования);
- всегда существует вероятность «поворота дела» (обнаружение признаков преступления при производстве судебной экспертизы по гражданскому или арбитражному делу, обнаружение признаков другого преступления при производстве судебной экспертизы по уголовному делу и т. д.), что повлечет смещение акцента на компьютерную информацию, которая ранее представлялась незначимой или малозначимой;
- всегда существует вероятность возникновения в деле вопросов о природе компьютерной информации, использованной в обоснование позиции одной из сторон (в особенности, если были установлены процессуальные нарушения, связанные с носителем этой компьютерной информации; кроме того, возможны ситуации, когда необходимо показать, что на носителе, целостность опечатанной упаковки которого была нарушена по какой-либо причине или упаковка которого не соответствовала криминалистическим требованиям[11], отсутствуют файлы, записанные после изъятия);

10 По мнению автора, прежде всего следует исходить из применимых в конкретном случае методов исследования компьютерной информации и имеющихся для реализации этих методов технических средств. Снижение количества модифицируемых байтов исследуемой компьютерной информации, при соблюдении установленных юридических и методических требований, не должно быть самоцелью.

11 В части исключения возможности незаметного доступа к упакованному и опечатанному объекту, а также в части подмены опечатанной упаковки объекта. Более детально вопросы упаковки и опечатывания объектов рассмотрены в статье Шапошникова А. Ю. [10].

- даже «простые» операции вроде подключения носителя к компьютеру в режиме «чтение-запись» могут повлечь утрату важных информационных следов (в частности, может происходить автоматическое антивирусное сканирование с удалением детектируемых файлов, перезапись временных меток последнего доступа к файлам).

Факторы, нарушающие целостность оригинала компьютерной информации

Факторы, нарушающие целостность компьютерной информации на носителе при его изъятии или после его изъятия, можно разделить на физические (направленные на носитель как на аппаратное устройство, вещь) и логические (направленные на содержимое носителя — компьютерную информацию).

К физическим факторам можно отнести повреждение носителя, влекущее невозможность чтения (копирования) компьютерной информации в полном объеме, вследствие нарушения правил хранения или эксплуатации (падение носителя с высоты, его повреждение электрическим током и т. п.).

К логическим факторам можно отнести передачу носителю команды записи (или иной команды, влекущей модификацию компьютерной информации: например, команды высвобождения блока данных или команды стирания данных) или самостоятельное выполнение носителем операций, влекущих модификацию компьютерной информации.

В ряде случаев нельзя рассматривать компьютерную информацию, хранимую на носителе, отдельно от компьютера, в котором функционирует этот носитель. Например, компьютерная информация, хранимая на носителе, может быть зашифрована, а ключ шифрования может быть «запечатан» доверенным платформенным модулем, который «распечатает»[12] этот ключ по запросу загрузчика или операционной системы только в случае, если аппаратная и программная конфигурация компьютера не была изменена. Изъятие одного лишь носителя в таком случае приведет к потере доступа к зашифрованной компьютерной информации, хотя ее целостность нарушена не будет (вместо этого необходимо изымать компьютер целиком, что не всегда возможно в силу статьи 164.1 Уголовно-процессуального кодекса РФ, либо искать ключ восстановления, предназначенный для получения доступа к зашифрованной файловой системе в случае сбоя). Такие факторы влекут потерю доступа к компьютерной информации при сохранении ее целостности, однако при этом можно говорить о нарушении целостности системы «данные — компьютер».

Описанные выше логические факторы, влияющие на оригинал компьютерной информации, в свою очередь, могут быть умышленными (фальсификация компьютерной информации или уклонение от применения мер для обеспечения целостности компьютерной информации),

12 Слова «запечатан» и «распечатает» характеризуют способ, при котором ключ шифрования файловой системы, как набор байтов, шифруется («запечатывается») доверенным платформенным модулем и сохраняется в таком виде на носитель. Для использования этого ключа шифрования необходимо его расшифровать («распечатать»), что осуществляется тем же самым доверенным платформенным модулем при условии, что заданные при «запечатывании» параметры аппаратной и программной конфигурации компьютера не изменились (ключ, с помощью которого осуществляются операции «запечатывания» и «распечатывания», доверенный платформенный модуль не покидает).

неосторожными (например, «прямое включение» ноутбука из-за неудавшейся попытки вызвать меню выбора загрузочного носителя, если изначально планировалось произвести загрузку ноутбука в криминалистический дистрибутив[13] на сменном носителе) и случайными (например, выход ноутбука из спящего режима при открытии его крышки, если факт нахождения ноутбука во включенном состоянии не был задокументирован при назначении судебной экспертизы, а с момента изъятия ноутбука не прошло достаточно времени для потери заряда аккумуляторной батареи, или выход ноутбука из спящего режима для дальнейшего перехода в режим гибернации[14] при низком заряде аккумуляторной батареи[15]). Следует особо отметить, что признание логического фактора умышленным или неосторожным не равно признанию соответствующих действий незаконными (например, является законным просмотр компьютерной информации, хранимой на носителе, для решения вопроса о необходимости изъятия этого носителя в ходе обыска, даже если во время такого просмотра целостность компьютерной информации не обеспечивалась[16]); кроме того, как уже было сказано ранее, некоторые исследования компьютерной информации неизбежно сопряжены с нарушением ее целостности.

Некоторые нарушения целостности оригинала компьютерной информации могут происходить не от ответственного лица (например, эксперта или специалиста), а от производителя средства исследования. Например, если блокиратор записи, аппаратный дупликатор или криминалистический дистрибутив позволяют внести изменения (без согласия или уведомления пользователя) или сами (без какого-либо участия пользователя) вносят изменения в оригинальную компьютерную информацию в силу ошибки (недостатка).

Считается, что такие ошибки (недостатки) должны выявляться в ходе валидации средства исследования (перед его применением на практике) экспертом, экспертной или иной организацией. Однако реальные возможности по валидации средств исследования экспертами и экспертными организациями, как правило, ограничиваются проведением тестирования на базе концепции «черного ящика» (когда внутреннее устройство объекта исследования (валидации) не изучается, вместо этого проверяется корректность работы этого объекта с разным набором входных данных: в практике валидации блокираторов записи и аппаратных дупликаторов это заключается в проверке неизменности компьютерной информации после применения блокиратора записи или аппаратного дупликатора, при этом указанная компьютерная информация представлена в виде различного набора файловых систем, записанных на тестовые носители). Вместе с тем, автор считает, что реальное

13 Криминалистический дистрибутив – это операционная система и набор прикладных программ для решения основных задач исследования компьютерной информации. Криминалистические дистрибутивы могут использоваться для копирования компьютерной информации (с обеспечением ее целостности) из компьютеров без их разборки (путем загрузки в данный дистрибутив).

14 Сохранение содержимого оперативной памяти на энергонезависимый носитель в специальный файл (файл гибернации) с последующим выключением питания компьютера, при включении компьютера содержимое оперативной памяти восстанавливается из файла гибернации.

15 В практике автора был случай, когда изъятый ноутбук, находящийся в опечатанной упаковке, вышел из спящего режима (в котором находился на момент изъятия) и перешел в режим гибернации (из-за низкого уровня заряда аккумуляторной батареи). Это повлекло появление в файловой системе временных меток от даты после изъятия, что потребовало объяснения.

16 При условии, что не было иного нарушения закона.

обнаружение ошибок (недостатков), влияющих на целостность исследуемой компьютерной информации, возможно лишь с обязательной обратной разработкой объекта исследования (валидации), что существенного повышает требования к специалисту, который проводит валидацию.

Так, руководство FSR-G-218 устанавливает, что для тестирования аппаратного блокиратора записи достаточно попытаться записать данные на подключенный к нему носитель [3, пункт 8.1.4]. При этом не учитывается:

- что данное устройство может самостоятельно (без команды от операционной системы компьютера) отправить на носитель команду записи (например, если подключенный носитель содержит файловую систему в определенном состоянии, которое «провоцирует» операционную систему аппаратного блокиратора записи на запись каких-либо данных на подключенный носитель);
- что данное устройство может «скрыть» факт успешного изменения данных на носителе из-за применения кеша для операций чтения (если на носителе данные изменились, а в кеше — нет; таким образом, перезагрузка аппаратного блокиратора записи перед проверкой возможного факта изменения данных является обязательной операцией, если существование такого кеша не было опровергнуто);
- что данное устройство может блокировать не все команды, влекущие изменение данных на носителе (например, всегда необходимо проверять блокировку команд высвобождения блоков данных для флеш-накопителей, а не ограничиваться тестированием накопителей на жестких магнитных дисках), либо переставать блокировать некоторые команды при наступлении определенных условий (например, обнаружение нечитаемого сектора[17]).

Вышеперечисленные замечания (в особенности, первое) исключают полное и всестороннее тестирование аппаратного блокиратора записи только на базе концепции «черного ящика».

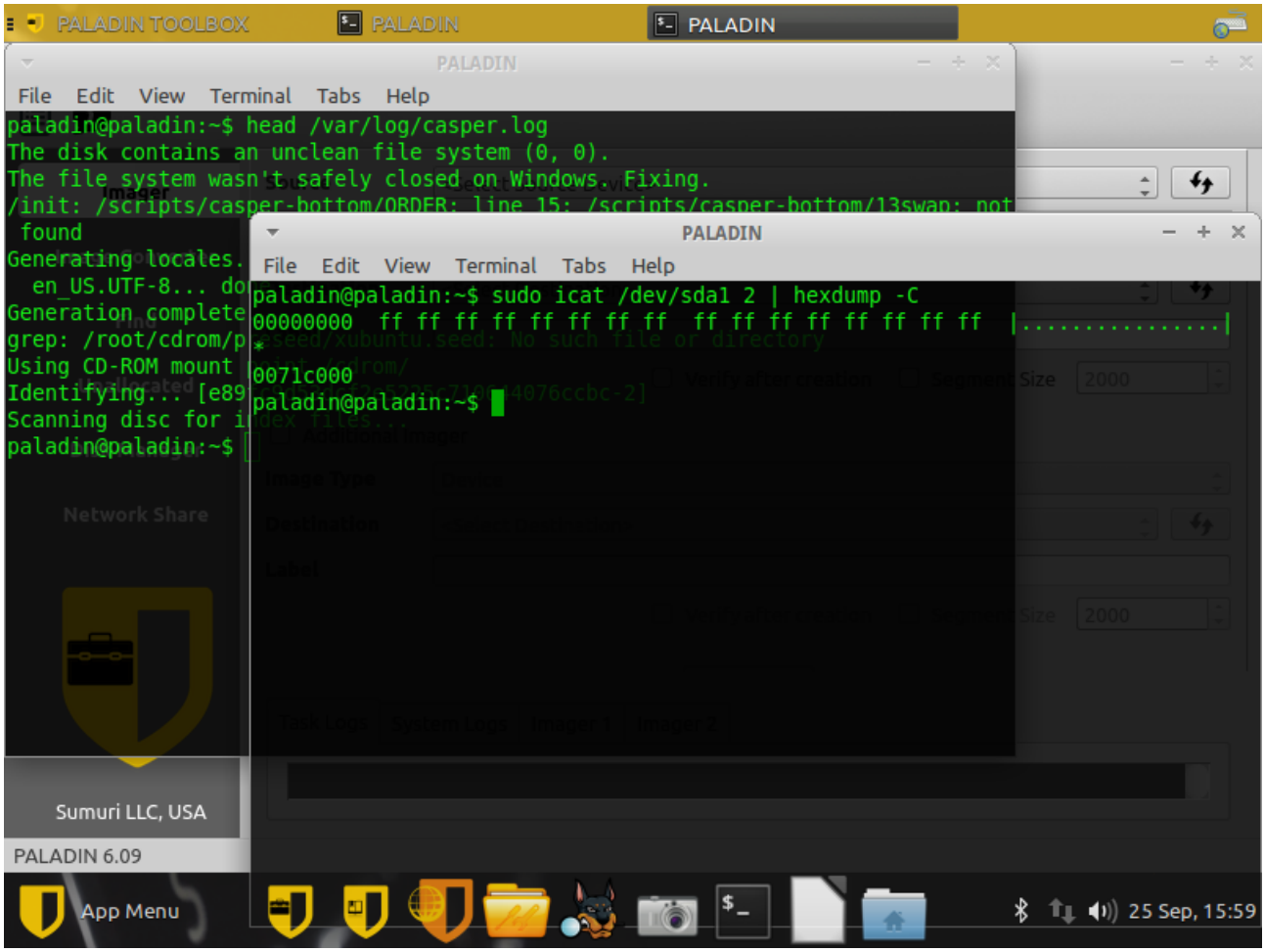
Кроме того, известны случаи, когда попытки протестировать иные средства исследования на базе концепции «черного ящика» оказывались неполными.

Например, в отчете о тестировании программы «SMART» (версия: 2010-11-03) в составе криминалистического дистрибутива «SMART Linux live CD» (версия: 2011-01) написано, что этот дистрибутив модифицировал на тестируемом носителе-источнике[18] секторы, используемые журналом NTFS [4, раздел 3.2]; при этом не указываются причины такого поведения тестируемого средства, но отмечается, что тесты с другими носителями с файловыми системами NTFS аналогичных результатов не дали (по мнению автора, причиной этому является некорректное отключение одного из носителей после форматирования в ходе подготовки тестовых данных, что не было предусмотрено планом тестирования и по факту привело к случайному выявлению описанного недостатка). В то же время результаты

17 В ядре Linux обнаружение нечитаемого сектора может привести к повторному опросу носителя, что, в свою очередь, может нарушить работу некоторых реализаций блокировки записи (за счет неожиданного сброса флага «только чтение», который был присвоен носителю ранее).

18 Тестировались функции копирования данных с одного носителя на другой.

тестирования аналогичных средств (криминалистических дистрибутивов), имеющих тот же недостаток[19] (по результатам их тестирования автором), подобных фактов модификации журнала NTFS не выявили [5][6][7]. Тесты, проведенные при подготовке описанных только что отчетов, были проведены Национальным институтом стандартов и технологий США (NIST).



Илл. 3. Криминалистический дистрибутив «PALADIN» (версия: 6.09), который прошел валидацию NIST

В терминале слева показаны журнальные записи, свидетельствующие о подключении «грязной» (отключенной некорректно) файловой системы. В терминале справа показан стертый журнал файловой системы NTFS («/$LogFile»), который до этого содержал различные записи. Данный недостаток не был обнаружен и задокументирован NIST.

Модификации, подобные описанной выше, характерны для дистрибутивов на основе Ubuntu, загружаемых со сменных носителей, поскольку на раннем этапе их загрузки, после передачи управления программе init в стартовой (начальной) файловой системе, загруженной в память загрузчиком, осуществляется поиск основной (корневой) файловой системы, представленной в файловой системе сменного носителя в виде файла-образа (обычно в формате SquashFS).

19 А именно, кратковременное подключение (монтирование; с последующим отключением) файловых систем NTFS (в том числе размещенных не на загрузочном носителе) в режиме «чтение-запись» в процессе загрузки дистрибутива. При этом драйвер NTFS (ntfs-3g) стирает незавершенный журнал файловой системы, чтобы его записи не конфликтовали с модификациями данных после подключения файловой системы.

Такой поиск происходит путем перебора подключенных носителей, при этом поддерживаемые файловые системы на каждом из них последовательно подключаются (монтируются), проверяются на наличие файла-образа основной (корневой) файловой системы (и иных признаков, указывающих на то, что данная файловая система должна использоваться для дальнейшей загрузки, — например, наличие файла со случайным идентификатором, генерируемым при создании загрузочного образа дистрибутива), после проверки файловая система либо используется для продолжения загрузки, либо отключается (и происходит переход к другой файловой системе носителя или к следующему носителю). Необходимость такого поиска обусловлена тем, что загрузчик не передает ядру операционной системы идентификатор носителя, с которого происходит загрузка, а потому загружаемая операционная система должна (при переходе на собственные драйверы вместо интерфейса BIOS/UEFI, используемого для чтения данных с носителей на самом начальном этапе загруки) инициировать поиск загрузочного носителя самостоятельно (а поскольку любой подключенный носитель может быть загрузочным, возникает необходимость в проверке каждого из них). Такой подход может нарушать целостность не только файловых систем NTFS, но и файловых систем Ext3, Ext4 (если они не были корректно отключены). Похожий процесс поиска реализован и в дистрибутивах на основе Debian, однако в них сменные носители проверяются до остальных, что приводит к нарушению целостности в случаях, если загрузка происходит не со сменного носителя (таким, в зависимости от версии дистрибутива, может считаться и накопитель на жестких магнитных дисках, подключенный по интерфейсу USB) или в ходе загрузки требуемый сменный носитель не был опознан драйверами дистрибутива.

Известны и другие случаи модификации компьютерной информации на исследуемых носителях (со стороны средства исследования). Так, аппаратный дупликатор «Tableau TD3» (протестированные версии программного обеспечения: 2.0.0, 2.1.0 и 2.1.1) самостоятельно модифицирует компьютерную информацию на носителе-источнике, если этот носитель содержит файловую систему Ext4, в журнале которой зафиксирована ошибка (например, ошибка ввода-вывода): модификация заключается в переносе кода ошибки из журнала в суперблок (заголовок) файловой системы.

Илл. 4. Различия в данных на носителе-источнике после применения аппаратного дупликатора «Tableau TD3» (начало)

Сверху показаны модифицированные данные (на накопителе-источнике, который был подключен к «защищенному от записи» порту устройства ранее), снизу показаны исходные данные (из образа файловой системы, который был записан на накопитель источник перед его подключением к аппаратному дупликатору).

Илл. 5. Различия в данных на носителе-источнике после применения аппаратного дупликатора «Tableau TD3» (конец)

Указанная модификация происходит из-за того, что названный дупликатор работает под управлением операционной системы на основе Linux и подключает (монтирует) файловые системы на носителях, подключенных к портам, якобы имеющих защиту от записи, при этом подлинный режим «только чтение» этим дупликатором не обеспечивается.

На практическую невозможность полноценной валидации средств исследования их производителями и поставщиками указывает и то обстоятельство, что в составе одного из предлагаемых в России «чемоданов эксперта» находится аппаратный блокиратор записи (интерфейс: PATA/SATA-USB) «AgeStar 3FBCP». Данное устройство блокирует команды записи, получаемые по интерфейсу USB, кроме команд внутри команд ATA pass-through[20].

Факторы, влияющие на компьютерную информацию при ее чтении

Необходимо отметить и некоторые вопросы целостности компьютерной информации в рамках процесса ее чтения (копирования), поскольку во многих случаях исследование компьютерной информации производится в отношении ее копии, а не оригинала (с целью уменьшения рисков, связанных с нарушением целостности оригинала компьютерной информации, а в некоторых случаях и с целью проведения нескольких исследований в одно время), при этом считается, что копия либо целиком, либо частично (в каких-либо известных пределах[21]) соответствует оригиналу.

Во время копирования компьютерной информации или, в общем случае, во время ее чтения могут происходить искажения, затрагивающие лишь копию этой компьютерной информации (но не оригинал): программы для копирования данных могут «обнулять» (записывать в копию нулевые байты вместо реального содержимого) секторы, расположенные вокруг нечитаемого сектора (в пределах границ копируемого за одну итерацию блока данных); программы для копирования данных могут произвольно «сдвигать» секторы, следующие за нечитаемым, т. е. записывать их в файл по неправильному смещению (например, очень старые версии программы «dd»[22] и программа «dcfldd» (версия: 1.3.4-1) неправильно определяют смещение, по которому следует записать в файл-получатель сектор, следующий за нечитаемым, что приводит к «сдвигу» копируемых данных и, как следствие, к их логическому повреждению [8]); аппаратные блокираторы записи могут возвращать ошибки ввода-вывода при попытках чтения секторов, расположенных вокруг нечитаемого сектора, исключая таким образом возможность чтения прилегающих к нечитаемому сектору данных операционной системой компьютера.

Например, криминалистический мост «Tableau T356789iu» (версия программного обеспечения: 1.3.0; в последующих версиях проблема была исправлена) сообщает о 128 нечитаемых секторах вместо одного (128 нечитаемых секторов включают в себя тот, который

20 Такие команды обычно разрешаются для получения диагностических данных S.M.A.R.T. от носителя. Тем не менее, таким же способом на носитель могут быть переданы и другие команды. Например, утилита «wiper.sh» (в составе программы «hdparm») использует команды ATA pass-through для передачи на носитель команд ATA TRIM.

21 В частности, может исследоваться точная копия одного раздела, а не всего содержимого носителя. В некоторых случаях (например, при копировании данных из работающей системы) можно говорить о точной копии каждого блока данных (сектора) на момент его копирования (после его копирования, до завершения процесса копирования всех блоков данных в целом, оригинал содержимого какого-либо скопированного блока может измениться).

22 Вышедшие до октября 2003 года.

действительно не читается, и 127 прилегающих к нему секторов, которые в действительности могут быть прочитаны, если подключить носитель напрямую).

Такая ошибка происходит из-за того, что названный криминалистический мост работает под управлением операционной системы на основе Linux и чтение с подключенного носителя происходит с участием кеша ядра, который заполняется чтением с носителя блоков по 128 секторов, при этом, в случае ошибки чтения, такой большой блок не перечитывается посекторно (и потому весь блок, а не конкретный сектор, считается нечитаемым). Данная ошибка была исправлена в актуальных версиях программного обеспечения криминалистического моста.

```
trying to recover sector 4148
trying to recover sector 4149
trying to recover sector 4150
trying to recover sector 4151
trying to recover sector 4152
trying to recover sector 4153
trying to recover sector 4154
trying to recover sector 4155
trying to recover sector 4156
trying to recover sector 4157
trying to recover sector 4158
trying to recover sector 4159
  1426915328 bytes ( 1,3 G ) copied (  2% ),    6 s, 222 M/s

input results for device `/dev/sdc':
   2786944 sectors in
   1 bad sectors replaced by zeros

output results for file `image.raw':
   2786944 sectors out

dc3dd aborted at 2018-08-10 22:21:50 +0300
```

Илл. 6: Пример чтения данных из носителя с одним нечитаемым сектором (без аппаратного блокиратора записи)

```
trying to recover sector 4215
trying to recover sector 4216
trying to recover sector 4217
trying to recover sector 4218
trying to recover sector 4219
trying to recover sector 4220
trying to recover sector 4221
trying to recover sector 4222
trying to recover sector 4223
     2162688 bytes ( 2,1 M ) copied (  0% ),   56 s, 38 K/s
[!!] 128 occurences while reading `/dev/sdd' from sector 4096 to sector 4223 : О
шибка ввода/вывода
  1387954176 bytes ( 1,3 G ) copied (  2% ),   72 s, 18 M/s

input results for device `/dev/sdd':
   2710848 sectors in
   128 bad sectors replaced by zeros

output results for file `image_wrtblk.raw':
   2710848 sectors out

dc3dd aborted at 2018-08-10 22:28:05 +0300
```

Илл. 7: Пример чтения данных из носителя с одним нечитаемым сектором (с использованием аппаратного блокиратора записи: «Tableau T356789iu», версия программного обеспечения: 1.3.0)

Видна информация о 128 нечитаемых секторах (вместо одного).

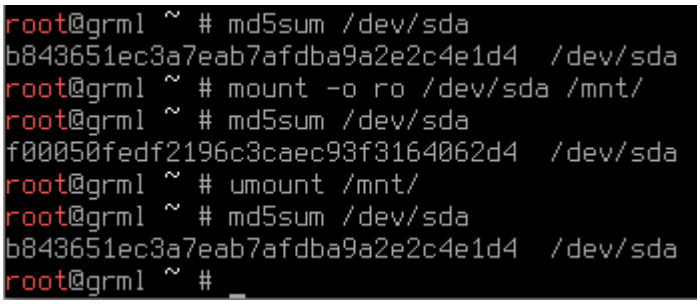
```
image.raw
001F FFB0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ........ ........
001F FFC0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ........ ........
001F FFD0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ........ ........
001F FFE0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ........ ........
001F FFF0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ........ ........
0020 0000: 22 00 00 04 00 00 00 00  04 00 00 08 00 00 00 00  "....... ........
0020 0010: 08 00 76 04 0C 0A 2F 13  00 02 00 00 01 00 00 01  ..v.../. ........
0020 0020: 01 00 22 00 00 04 00 00  00 00 04 00 00 08 00 00  ..".....  ........
0020 0030: 00 00 08 00 76 04 0C 0A  36 15 00 02 00 00 01 00  ....v... 6.......
image_wrtblk.raw
001F FFB0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ........ ........
001F FFC0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ........ ........
001F FFD0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ........ ........
001F FFE0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ........ ........
001F FFF0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ........ ........
0020 0000: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ........ ........
0020 0010: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ........ ........
0020 0020: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ........ ........
0020 0030: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ........ ........
```

Илл. 8: Сравнение прочитанных данных

Сверху показаны данные, полученные без аппаратного блокиратора записи, снизу — данные (нулевые байты), полученные с подключением носителя к аппаратному блокиратору записи: «Tableau T356789iu» (версия программного обеспечения: 1.3.0).

Кроме того, если блокиратор записи заблокировал полученную команду записи, но не сообщил об этом операционной системе (например, работая в режиме подавления ошибок

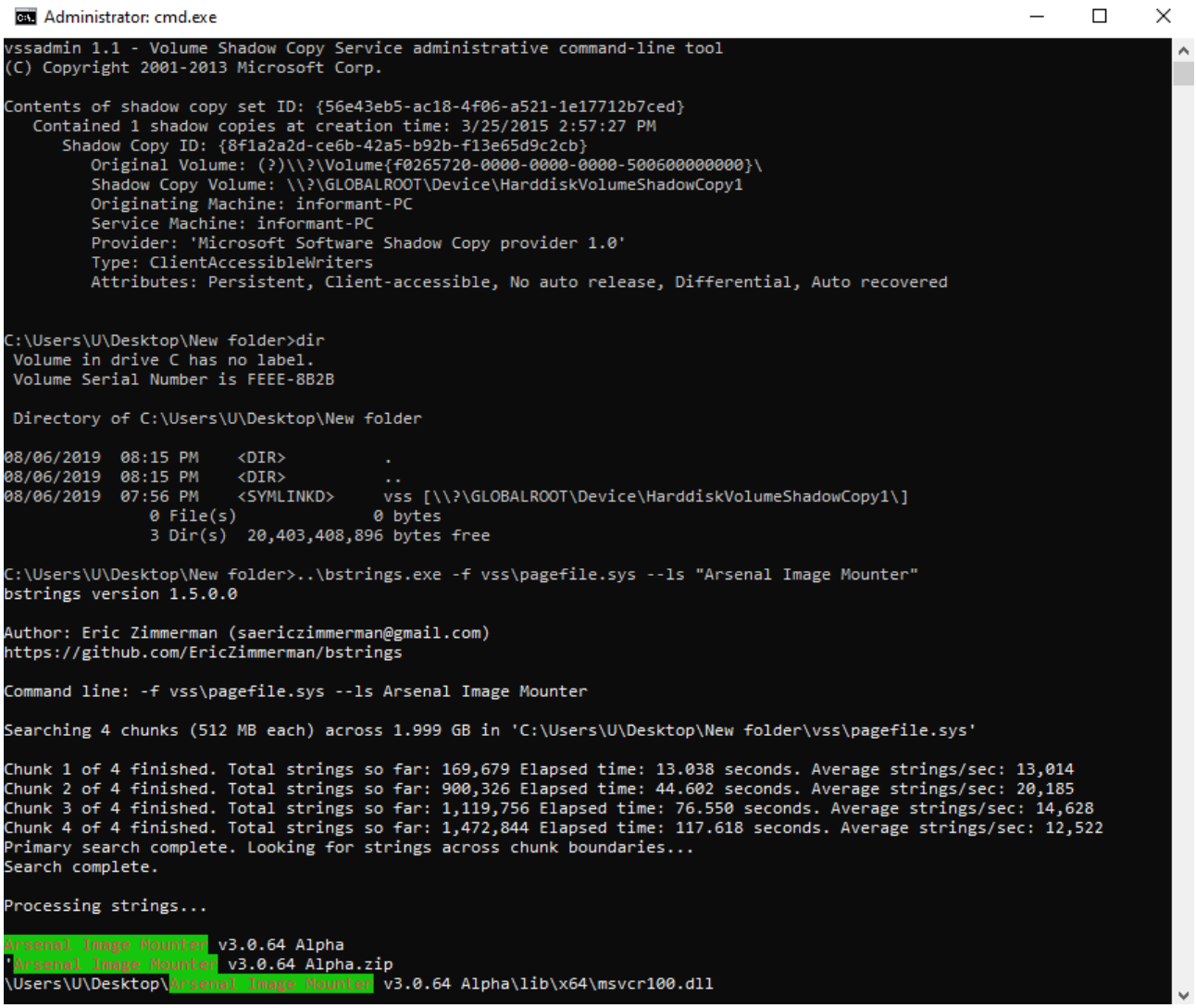
записи) или сообщил, а операционная система компьютера проигнорировала эту ошибку, то операционная система компьютера может сохранить модифицированные данные в кеше, а при дальнейшей попытке чтения якобы модифицированных блоков данных вернуть содержимое из кеша (а не прочитать его с носителя, где данные не были изменены).



Илл. 9: Пример искажения читаемых данных при использовании программного блокиратора записи

Показано изменение хеша от содержимого носителя до и после подключения (монтирования) файловой системы, а также после ее отключения. Изменения затрагивали только кеш в оперативной памяти компьютера (криминалистический дистрибутив «grml», версия: 2014.11).

Искажения могут иметь место и в процессе исследования созданной копии компьютерной информации. Например, одно из наиболее опасных искажений возможно при проведении поиска по ключевым словам в файлах из теневых копий операционных систем семейства Windows — драйвер теневого копирования при чтении некоторых файлов, исключенных из теневой копии при ее создании (например, файла подкачки: «/pagefile.sys»), может в некоторых (не во всех) случаях возвращать в качестве прочитанного содержимого фрагменты данных из оперативной памяти стендового компьютера (компьютера, на котором проводится исследование). Это обусловлено тем, что выделенный фрагмент памяти, предназначенный для записи в него прочитанных из теневой копии данных, не очищается и не изменяется драйвером теневого копирования, а возвращается пользовательской программе как есть (с остаточными данными).



Илл. 10: Поиск по ключевым словам в содержимом файла «/pagefile.sys» в теневой копии

В результате этого поиска были обнаружены данные, отсутствующие в исследуемом образе, которые были взяты из оперативной памяти стендового компьютера (речь идет про строки со словами «Arsenal Image Mounter», на момент создания тестового образа версия «3.0.64 Alpha» программного обеспечения «Arsenal Image Mounter» отсутствовала, а в третьей обнаруженной строке указан путь на стендовом компьютере).

Такие искажения при чтении (копировании) компьютерной информации и ее исследовании могут негативно влиять на достоверность выводов по результатам проведенного исследования, а также приводить к разночтениям при оценке фактических обстоятельств дела.

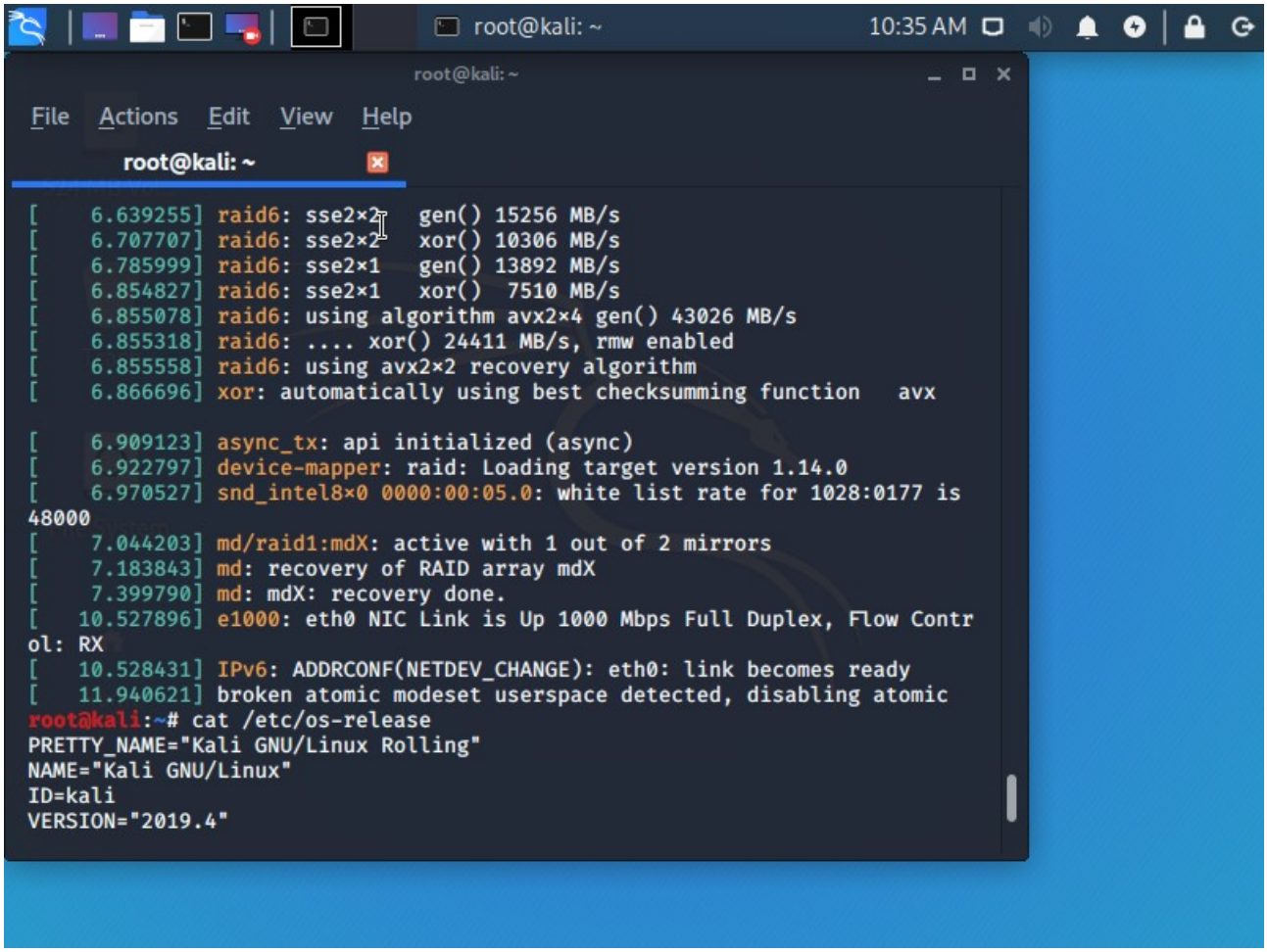
Виды нарушений целостности оригинала компьютерной информации

Нарушения целостности оригинала компьютерной информации можно условно разделить на два вида:

1. позволяющие по модифицированной компьютерной информации установить факт ее модификации после определенного момента времени;
2. не позволяющие по модифицированной компьютерной информации установить факт ее модификации после определенного момента времени или не позволяющие убедительно установить сам факт такой модификации.

К первому виду можно отнести модификацию (или создание) файла с фиксацией временной метки такой модификации (такого создания) в метаданных файловой системы (при условии, что показания системных часов соответствуют, точно или с незначительным отклонением, текущему времени на момент модификации).

Ко второму виду можно отнести показанные на илл. 4 и 5 изменения, а также синхронизацию компьютерной информации на носителях в составе «зеркального» RAID. Например, при использовании «зеркальной» конфигурации менеджера логических томов (Logical Volume Manager, сокращенно — LVM) Linux, когда двум и более физическим носителям соответствует одна группа логических томов, данные которой хранятся на каждом физическом носителе в полном объеме («зеркалирование» данных), часть данных на одном физическом носителе может быть устаревшей (в сравнении с данными на другом физическом носителе) из-за сбоя или действий администратора (замена, отключение носителя на уровне операционной системы и т. п.). Если для копирования данных из такой конфигурации используется криминалистический дистрибутив и этот дистрибутив допускает синхронизацию данных на подключенных физических носителях, то по синхронизированным данным (т. е. после их модификации) нельзя сказать, что эти данные на одном из физических носителей были другими ранее, до запуска криминалистического дистрибутива (т. е. сам факт модификации компьютерной информации является неочевидным).



Илл. 11: Журнал работы ядра дистрибутива «Kali Linux» (версия: 2019.4), который имеет криминалистический режим работы

В данном режиме допускается синхронизация данных драйвером Linux LVM, о чем в конкретном случае свидетельствуют журнальные записи, показанные в терминале (начинаются с временной метки «7.» в квадратных скобках). После перезагрузки компьютера сведения о синхронизации данных будут утрачены, а по самим данным нельзя будет судить об их синхронизации.

Надежное обнаружение второго вида нарушений целостности возможно лишь путем подробного документирования всех действий как при изъятии носителя, так и после (в том числе путем документирования содержимого системных журналов криминалистического дистрибутива, если он применялся). Хотя не исключаются ситуации, когда даже такое документирование не позволит дать однозначный ответ о факте нарушения целостности оригинала компьютерной информации в некоторый период времени.

Некоторые технические детали о файловой системе NTFS

Файловая система NTFS — это основная файловая система, используемая операционными системами семейства Windows. По состоянию на 2020 год, для системного тома (т. е. для хранения основных файлов операционной системы, за исключением менеджера загрузки, его

конфигурационных файлов и, в некоторых случаях, образа восстановления системы) может быть использована только файловая система NTFS.

Несмотря на попытки корпорации Microsoft внедрить файловую систему ReFS, файловая система NTFS до сих пор является активно поддерживаемой в плане разработки (в частности, одними из последних нововведений стали поддержка очень больших кластеров — более 64 КиБ, а также включение обновления временных меток последнего доступа к файлам).

Реализации (драйверы) файловой системы NTFS можно разделить на официальные, присутствующие в операционных системах семейства Windows, и неофициальные, созданные сторонними разработчиками (например, драйвер ntfs-3g). Неофициальные драйверы базируются на ограниченном понимании внутреннего устройства файловой системы NTFS (поскольку открытая официальная спецификация не существует) и не содержат реализации некоторой функциональности, не критичной для пользователя (например, отсутствует журналирование операций записи).

Файловая система NTFS является журналируемой — перед изменением структур файловой системы на накопителе происходит запись описаний этих изменений в журнал, т. е. каждое изменение сначала сохраняется в специальный файл журнала и лишь затем, после успешной записи в журнал, оно сохраняется в целевой области накопителя. В случае сбоя во время записи в журнал, структуры файловой системы на накопителе (вне файла журнала) останутся целыми (поскольку неудачная запись их не касалась), а в случае сбоя во время записи в целевую область накопителя, на базе описаний изменений из файла журнала можно повторить или отменить (откатить) неудачную операцию изменения структур файловой системы. Следует отметить, что журналирование в файловой системе NTFS затрагивает только изменения структур этой файловой системы (файловые записи, индексы директорий и некоторые другие типы метаданных), но не данные вроде содержимого файлов (таким образом, официальная реализация файловой системы NTFS гарантирует защиту ее внутренних структур от сбоев вроде прерывания электропитания или внезапного отключения накопителя от компьютера, но содержимое пользовательских файлов такой защиты не имеет). Описанный файл журнала находится в корневой директории и имеет имя «$LogFile». В нем каждая запись, описывающая изменение каких-либо служебных данных файловой системы, имеет уникальный номер («Log file sequence number»), который обладает следующим свойством: у каждой новой записи этот номер всегда больше, чем у предыдущей.

Неофициальные драйверы, поддерживающие режим «чтение-запись», по состоянию на 2020 год не имеют поддержки восстановления файловой системы NTFS по записям в журнале (в случае сбоев, описанных выше). Поэтому, если файловая система NTFS требует такого восстановления, производится либо ее подключение (монтирование) в режиме «только чтение», либо очистка файла журнала во время подключения (монтирования). Таким образом, исключается конфликт между записями в журнале (которые применит официальный драйвер) и модификациями от неофициального драйвера (например, записи в журнале могут описывать создание файла, который так и не был создан из-за сбоя, а неофициальный драйвер создаст на его месте другой файл, проигнорировав записи в журнале; в ходе

восстановления этот созданный файл будет утерян, поскольку его файловая запись будет использована для того файла, создание которого описано в журнале). Таким образом, в ущерб возможности восстановления файловой системы NTFS после сбоя обеспечивается бесконфликтность работы неофициального драйвера.

Для функционирования сторонних программ, обеспечивающих резервное копирование или индексацию (поиск) данных, официальная реализация файловой системы NTFS предоставляет журнал с краткими описаниями многих действий в отношении файловой системы (например, создание или удаление файла, переименование файла, изменение содержимого файла). Такой журнал может использоваться, например, для быстрого поиска файлов, подлежащих резервному копированию, если есть необходимость исключить из создаваемой резервной копии файлы, уже скопированные ранее (иными словами, речь идет о задаче быстрого поиска файлов, созданных или измененных после предыдущего резервного копирования). Такой журнал, если он включен, находится в директории «/$Extend» и имеет имя «$UsnJrnl:$J» (имя файла — «$UsnJrnl», данные журнала находятся в альтернативном потоке с именем «$J»), сам журнал называется «Update Sequence Number (USN) change journal». В этом журнале каждая запись, описывающая файловое событие, имеет уникальный номер («Update sequence number»), который у всякой новой записи больше, чем у предыдущей.

Кроме того, в современных официальных реализациях файловой системы NTFS (в Windows 8, 8.1 и 10) имеются два журнала, в которых фиксируются подтвержденные и предполагаемые ошибки в структурах файловой системы на носителе, — оба журнала находятся в директории «/$Extend/$RmMetadata» и имеют имена «$Repair:$Corrupt», «$Repair:$Verify» соответственно.

Файловые системы NTFS могут иметь теневые копии, которые являются разновидностью резервных копий многих частей файловой системы (включая содержимое почти всех файлов). Теневые копии не являются частью файловой системы, хотя и существуют внутри нее, поскольку за сохранение и чтение теневых копий отвечает драйвер, работающий на уровень ниже драйвера NTFS. Драйвер теневого копирования выделяет некоторое количество пространства файловой системы для размещения информации о теневых копиях, а также для размещения блоков данных, составляющих каждую теневую копию.

Каждая теневая копия представляет собой слепок состояния файловой системы на момент создания этой теневой копии. Такое состояние может быть представлено как самостоятельная файловая система (которая в части представленных в явном (неудаленном) виде данных будет той же файловой системой в какой-то момент времени в прошлом; при этом в операционных системах Windows 8, 8.1 и 10 могут создаваться «целевые» теневые копии — в них включаются только те блоки данных, которые необходимы для отката системных файлов). При создании теневой копии ядру операционной системы и ряду прикладных программ передается команда «заморозить» данные, т. е. обеспечить их целостное состояние на носителе, чтобы в теневую копию не попали файлы в момент активной записи в них (т. е. когда процесс записи в файл был начат, но еще не завершен, что может повлечь

рассогласование «новых» (перезаписанных) данных из начала файла со «старыми» (еще не перезаписанными) данными из конца файла, что, в свою очередь, снижает качество резервного копирования).

На низком уровне теневые копии реализованы в виде списка с элементами «исходное смещение — новое смещение», где «исходное смещение» — это смещение блока данных, которое драйвер NTFS пытается прочитать, а «новое смещение» — это смещение блока данных, которое фактически должен прочитать драйвер теневого копирования, чтобы вернуть драйверу NTFS данные по состоянию на момент создания теневой копии. Таким образом, при чтении виртуального устройства, содержащего теневую копию, драйвер теневого копирования перенаправляет запросы чтения к другим областям данных согласно указанному списку, а при отсутствии нужного элемента в списке запрос чтения перенаправляется согласно списку для следующей (созданной позднее) теневой копии (если и в том списке нет нужного элемента, то запрос перенаправляется дальше аналогичным способом) или, если следующей теневой копии нет (текущая теневая копия является последней), с носителя читаются данные по «исходному смещению».

Иными словами, последняя из созданных теневых копий содержит разницу между текущими данными файловой системы и данными файловой системы на момент создания этой теневой копии, предпоследняя из созданных теневых копий содержит разницу между данными последней теневой копии и данными файловой системы на момент создания предпоследней теневой копии. И далее по аналогии.

Современные операционные системы семейства Windows имеют функцию быстрой загрузки (иногда ее называют гибридным выключением). Если аппаратное обеспечение компьютера поддерживает режим гибернации (большинство современных конфигураций компьютеров имеют такую поддержку), то выключение компьютера через меню «Пуск» по умолчанию завершает пользовательский сеанс (как при обычном выключении компьютера) и переводит ядро операционной системы в режим гибернации. Отличие от стандартного режима гибернации заключается в том, что в последнем случае пользовательский сеанс не завершается.

Поскольку драйвер NTFS работает в режиме ядра, то при таком выключении компьютера часть служебных структур, описывающих файловую систему и ее текущее состояние, сохраняется в файле гибернации, а затем, при включении компьютера, восстанавливается из этого файла. Если структуры файловой системы NTFS были изменены в период гибернации (например, из другой операционной системы путем создания новых файлов или изменения существующих), то возникает противоречие между служебными данными на носителе и в оперативной памяти (приоритет для драйвера имеют служебные данные в оперативной памяти).

По этой причине драйвер ntfs-3g не стирает незавершенный журнал файловой системы, если этот журнал имеет версию 2.0, что указывает на подключение (монтирование) файловой системы в операционной системе семейства Windows с поддержкой быстрой загрузки (в конфигурации по умолчанию эта версия снижается до 1.1 при отключении файловой

системы, в том числе при полноценном выключении компьютера). В таком случае файловая система всегда подключается (монтируется) в режиме «только чтение» (а пользователю предлагается загрузиться в операционную систему Windows и полноценно выключить компьютер).

Современные операционные системы семейства Windows имеют поддержку файловых транзакций, в рамках которых высокоуровневые операции над файлами и директориями (создание файла, запись в него данных, удаление другого файла, создание директории и т. д.) могут быть объединены в одну сущность, которая обеспечивает единовременное выполнение всех этих операций в полном объеме (т. е. можно, например, создать 100 файлов таким образом, что в произвольный момент времени какая-либо программа будет «видеть» либо 0 созданных файлов — до или во время исполнения транзакции, либо 100 созданных файлов — после исполнения транзакции; это соответствует концепции транзакций, существующей во многих системах управления базами данных). Файловые транзакции являются частью файловой системы, однако их поддержка не является обязательной. Служебные файлы, относящиеся к файловым транзакциям, находятся в директориях «/$Extend/$RmMetadata/$Txf» и «/$Extend/$RmMetadata/$TxfLog».

Методика обнаружения нарушений целостности компьютерной информации в файловой системе NTFS

В рамках настоящей работы были протестированы следующие сценарии нарушения целостности компьютерной информации в файловой системе NTFS:

1. Подключение носителя к компьютеру в режиме «чтение-запись» без совершения пользователем действий, направленных на запись на этот носитель.
2. Подключение носителя к компьютеру в режиме «чтение-запись» и копирование на этот носитель (с компьютера) файла способом, который имитирует фальсификацию компьютерной информации:

   2.1. Системные часы компьютера установлены «задним числом» (на время до изъятия).

   2.2. Системные часы компьютера установлены в текущее время, однако временные метки скопированного файла изменяются программой timestomp.

Предполагаемые результаты тестов следующие:

1. Простое подключение носителя к компьютеру под управлением операционной системы семейства Windows должно изменять, как минимум, служебные файлы, относящиеся к функциональности файловых транзакций.
2. Простое подключение носителя к компьютеру под управлением операционной системы на базе Linux может приводить к обновлению временных меток последнего доступа к файлам и директориям. В случае, если подключаемая (монтируемая)

файловая не была корректно отключена ранее, ожидается стирание журнала файловой системы (в случаях, когда журнал файловой системы имеет версию ниже 2.0).

3. Подключение носителя с установленной операционной системой Windows 7, который никогда ранее не подключался к компьютеру с операционной системой Windows 8, 8.1 или 10, оставит следы такого подключения в виде создания журналов ошибок в структурах файловой системы NTFS (обычно пустых).
4. Подключение носителя к компьютеру под управлением операционной системы семейства Windows, у которого системные часы установлены «задним числом», с последующим копированием на этот носитель файлов не должно влиять на порядок файловых операций, определяемый по значениям полей «Log file sequence number» и «Update sequence number» (что позволяет определить последние файловые операции над файловой системой вне зависимости от временных меток). *Этот результат ранее нашел подтверждение в одной из работ [9]*.
5. Подключение носителя к компьютеру под управлением операционной системы на базе Linux, у которого системные часы установлены «задним числом», с последующим копированием на этот носитель файлов должно оставлять характерный след — как минимум, нулевое значение поля «Log file sequence number» у соответствующей файловой записи.
6. Подключение носителя к компьютеру под управлением операционной системы семейства Windows, у которого системные часы установлены в текущее время, с последующим копированием на этот носитель файлов, за которым следует изменение их временных меток программой timestomp, должно оставлять явный след изменения временных меток в журнале файловой системы.
7. При наличии теневых копий файловой системы, копирование в эту файловую систему файлов не должно изменять содержимое ее теневых копий.
8. Просмотр (например, в программе Notepad или WordPad) файлов на внешнем несъемном носителе (например, на накопителе на жестких магнитных дисках, который подключен через интерфейс USB и имеет файловую систему NTFS) приведет к появлению у этих файлов атрибутов $OBJECT_ID, которые содержат MAC-адрес (основного сетевого интерфейса) компьютера, на котором происходил просмотр файлов, и временную метку (которая соответствует дате и времени генерации уникального идентификатора в атрибуте $OBJECT_ID, которые обычно примерно соответствуют времени загрузки операционной системы).
9. Подключение носителя, содержащего операционную систему в состоянии гибернации или в состоянии быстрой загрузки, к компьютеру с последующим копированием на этот носитель файлов должно создавать противоречие между служебными данными файловой системы на этом носителе и в оперативной памяти (в файле гибернации операционной системы на том же носителе).

Результаты тестов, проведенных в отношении операционный систем Windows 7 и Windows 10, в целом подтверждают справедливость ожидаемых результатов. Тесты были проведены с использованием следующих операционных систем:

Физический компьютер Windows 10 Домашняя 64-разрядная операционная система Версия драйвера NTFS: 10.0.18362.145 Часть тестов проведена после обновления драйвера NTFS до версии: 10.0.18362.449
Виртуальная машина (VMWare) Windows 10 Профессиональная 64-разрядная операционная система Версия драйвера NTFS: 10.0.18362.145
Физический компьютер Windows 7 Домашняя расширенная Service Pack 1 64-разрядная операционная система Версия драйвера NTFS: 6.1.7601.24382 Версия антивирусной программы: Kaspersky Anti-Virus 19 Некоторые тесты дублировались с отключенной антивирусной программой.
Физический компьютер Ubuntu 19.04 (64-bit), в режиме без установки Версия драйвера ntfs-3g: 2017.3.23AR.3 *(та же версия используется в Ubuntu 19.10)*

Таблица 1: Использованные в тестах операционные системы

Эксперименты проводились в отношении двух накопителей, каждый с одной файловой системой NTFS: накопитель типа USB Flash (идентифицирует себя как съемный) и флеш-накопитель типа solid-state drive, подключаемый по интерфейсу USB через переходник (идентифицирует себя как несъемный). Образы тестируемых файловых систем были получены в результате форматирования соответствующих накопителей как из операционной системы Windows 7, так и из операционной системы Windows 10.

Простое подключение накопителя (любого из названных) к компьютеру под управлением операционной системы семейства Windows приводит к изменениям в файловой системе. Иногда эти изменения не могут быть идентифицированы по временным меткам, т. е. факт подключения может остаться неочевидным (изменение данных не может быть датировано). Такие неочевидные изменения в тестах оказались ограничены ситуациями, когда накопитель отключается некорректно (без выбора опции извлечения накопителя в интерфейсе операционной системы семейства Windows), за исключением случая, когда обновления временных меток последнего доступа оказываются включены.

Идентификация простого подключения возможна по временным меткам следующих файлов:

1. /$Extend/$RmMetadata/$TxfLog/$TxfLog.blf
2. /$Extend/$RmMetadata/$TxfLog/$TxfLogContainer00000000000000000001
3. /System Volume Information/WPSettings.dat
4. /System Volume Information/IndexerVolumeGuid

В первых двух случаях следует обращать внимание на временную метку изменения файла, а в последних двух — на временную метку последнего доступа к файлу.

Следует отметить, что операционная система семейства Windows может синхронизировать временные метки между атрибутом $STANDARD_INFORMATION в файловой записи и атрибутом $FILE_NAME в индексе родительской директории (в некоторых случаях временные метки в индексе родительской директории могут отставать от таковых в файловой записи), а потому простое подключение накопителя может привести к записи временных меток «из прошлого» (т. е. не соответствующих текущему времени) в индексы директорий. Так, в одном из тестов, где накопитель не был корректно отключен, были обновлены временная метка модификации файла и временная метка изменения MFT-записи (файловой записи) для файлов с именами «$TxfLog.blf» и «$TxfLogContainer00000000000000000001» в индексе директории «/$Extend/$RmMetadata/$TxfLog» (указанные временные метки получили значения из атрибута $STANDARD_INFORMATION в файловых записях двух названных файлов). Подобное изменение затрагивает только индекс соответствующей директории, временная метка в этом индексе получает известное значение, а потому такое изменение легко идентифицируется (признак: отсутствие сопутствующего изменения атрибута $STANDARD_INFORMATION в файловой записи, которое уже имело место до подключения накопителя).

Таким образом, первый ожидаемый результат подтвержден с некоторыми исключениями (в части того, что обновления временных меток могут не наблюдаться при некорректном отключении накопителя).

Простое подключение несъемного накопителя к компьютеру под управлением операционной системы Ubuntu 19.04 приводит к изменениям в файловой системе, а именно — к обновлениям временных меток последнего доступа к различным пользовательским файлам.

В случае, если несъемный накопитель был отформатирован в операционной системе Windows 7 и не был корректно отключен, происходит стирание содержимого файла «/$LogFile» (его перезапись байтами 0xFF). В случае, если несъемный накопитель был отформатирован в операционной системе Windows 10 (версия журнала: 2.0) и не был корректно отключен, файловая система не изменяется, поскольку монтируется в режиме «только чтение».

При этом некорректное отключение накопителя сразу после его подключения не приводит к изменениям в файловой системе, поскольку операционная система не успевает синхронизировать данные с накопителем.

Следует отметить, что по умолчанию операционные системы на основе Linux подключают (монтируют) файловые системы с параметром «relatime». В таком случае драйвер ntfs-3g, согласно исследованию его исходного текста и проведенному тесту, не обновляет временную метку последнего доступа на накопителе, если каждое из трех условий выполняется:

1. имело место только событие доступа (а не событие изменения файла или подобное);
2. дата и время последнего доступа позднее даты и времени модификации файла;
3. дата и время последнего доступа позднее даты и времени изменения MFT-записи.

Данное обстоятельство сужает перечень файлов, временные метки последнего доступа которых могут быть обновлены.

Таким образом, второй ожидаемый результат подтвержден.

Простое подключение несъемного накопителя, отформатированного в операционной системе Windows 7, к компьютеру под управлением операционной системы Windows 10 приводит к созданию журналов ошибок: «$Extend/$RmMetadata/$Repair:$Corrupt» и «/$Extend/$RmMetadata/$Repair:$Verify». Эти журналы представляют собой альтернативный потоки данных файла «$Extend/$RmMetadata/$Repair».

Простое подключение съемного накопителя, отформатированного в операционной системе Windows 7, к компьютеру под управлением операционной системы Windows 10 к созданию журналов ошибок не приводит.

Следовательно, третий ожидаемый результат подтвержден только в отношении несъемных накопителей.

Для проверки дальнейших предполагаемых результатов использовались два образа установленной операционной системы Windows 10 (два разных экземпляра, один из которых активно использовался в течение длительного периода времени до начала тестов), записанные на несъемный накопитель.

В ходе тестов на такой накопитель копировался файл с именем, содержащим «planted_evidence». Этот файл не является пустым (в нем есть данные в формате, который

соответствует расширению файла). Копирование производилось с другого компьютера, у которого системные часы установлены «задним числом».

Процитированные ниже фрагменты данных были получены с помощью программы dfir_ntfs[23].

В случае, когда копирование производилось из операционной системы Windows 10, зафиксировано следующее.

Последние события в журнале «USN change journal»:

Значение USN	Причина генерации события	Временная метка	Имя файла
10918043248	USN_REASON_DATA_EXTEND | USN_REAS	2019-09-12 21:03:09.958524	swapfile.sys
10918043336	USN_REASON_DATA_OVERWRITE | USN_RE	2019-09-12 21:03:09.958524	NTUSER.DAT
10918043416	USN_REASON_DATA_OVERWRITE | USN_RE	2019-09-12 21:03:09.958524	IconCache.db
10918043504	USN_REASON_DATA_OVERWRITE | USN_RE	2019-09-12 21:03:09.958524	EtwRTEventLog-System.etl
10918043648	USN_REASON_DATA_OVERWRITE | USN_RE	2019-09-12 21:03:09.958524	BootPerfDiagLogger.etl
10918043752	USN_REASON_DATA_EXTEND | USN_REAS	2019-09-12 21:03:09.958524	StateRepository-Machine.srd-wal
10918043880	USN_REASON_DATA_EXTEND | USN_REAS	2019-09-12 21:03:09.958524	WdiContextLog.etl.002
10918043984	USN_REASON_FILE_CREATE	2019-09-01 15:58:53.001310	planted_evidence_.rtf
10918044088	USN_REASON_FILE_CREATE | USN_REASC	2019-09-01 15:58:53.001310	planted_evidence_.rtf
10918044192	USN_REASON_DATA_EXTEND	2019-09-01 15:58:53.004308	planted_evidence_.rtf
10918044296	USN_REASON_DATA_OVERWRITE | USN_R	2019-09-01 15:58:53.005304	planted_evidence_.rtf
10918044400	USN_REASON_DATA_OVERWRITE | USN_RE	2019-09-01 15:58:53.012378	planted_evidence_.rtf
10918044504	USN_REASON_DATA_OVERWRITE | USN_RE	2019-09-01 15:58:53.013424	planted_evidence_.rtf
10918044608	USN_REASON_DATA_OVERWRITE	2019-09-01 15:59:04.253954	$TxfLog.blf
10918044696	USN_REASON_DATA_OVERWRITE | USN_R	2019-09-01 15:59:04.257196	$TxfLog.blf

Илл. 12: Последние события в журнале «USN change journal» (второй столбец обрезан), модификация из операционной системы Windows 10 (системные часы установлены «задним числом»)

Первое событие в указанном журнале от 2019-09-12 17:36:09.055186. Таким образом, записи журнала «USN change journal» идут за 12 сентября 2019 года, начиная с 17:36:09 и заканчивая 21:03:09. За ними следуют записи от 1 сентября 2019 года, что является аномалией, свидетельствующей о сдвиге показаний системных часов компьютера (компьютеров), записывающего (записывающих) данные в файловую систему.

В журнале «$LogFile» первой записью, относящейся к копированию файла с указанным ранее именем и содержащей временную метку, является следующая:

```
LSN: 49745043011
Transaction ID: 24
Log record, redo operation: AddIndexEntryAllocation, undo operation:
DeleteIndexEntryAllocation
Target (file reference number): 23643898043698760
Target (attribute name): $I30
Target path (from $MFT): /Users/U/Desktop
Offset in tagret: 7016
LCN(s): 1578189
Redo data:
00000000  20 05 00 00 00 00 7E 00-80 00 6C 00 00 00 00 00   .....~...l.....
00000010  48 0E 00 00 00 00 54 00-B3 CF F6 26 DE 60 D5 01  H.....T....&.`..
00000020  B3 CF F6 26 DE 60 D5 01-B3 CF F6 26 DE 60 D5 01  ...&.`.....&.`..
00000030  B3 CF F6 26 DE 60 D5 01-00 80 03 00 00 00 00 00  ...&.`..........
00000040  00 00 00 00 00 00 00 00-20 00 00 00 00 00 00 00  ........ .......
00000050  15 01 70 00 6C 00 61 00-6E 00 74 00 65 00 64 00  ..p.l.a.n.t.e.d.
```

23 URL: https://github.com/msuhanov/dfir_ntfs

```
00000060  5F 00 65 00 76 00 69 00-64 00 65 00 6E 00 63 00  _.e.v.i.d.e.n.c.
00000070  65 00 5F 00 2E 00 72 00-74 00 66 00 02 00 00 00  e._...r.t.f.....

Undo data:
-

$FILE_NAME in index:
 * M timestamp: 2019-09-01 15:58:53.001310
 * A timestamp: 2019-09-01 15:58:53.001310
 * C timestamp: 2019-09-01 15:58:53.001310
 * E timestamp: 2019-09-01 15:58:53.001310
 * File name: planted_evidence_.rtf
 * Parent (file reference number): 23643898043698760
 * Parent path (from $MFT): /Users/U/Desktop
```

Данная запись журнала добавляет в индекс родительской директории («/Users/U/Desktop») сведения о файле с именем «planted_evidence_.rtf».

Перед этой записью имеются другие записи, последняя из которых (имеющая в своем составе временную метку) имеет более позднюю временную метку:

```
LSN: 49745040762
Transaction ID: 24
Log record, redo operation: UpdateNonresidentValue, undo operation: Noop
Target (file reference number): 562949953475627
Target (attribute name): $J
Target path (from $MFT): /$Extend/$UsnJrnl
Offset in tagret: 10918043880
LCN(s): 3538814
Redo data:
00000000  68 00 00 00 02 00 00 00-8A 07 00 00 00 00 2D 68  h.............-h
00000010  A4 7C 00 00 00 00 19 00-E8 20 C4 8A 02 00 00 00  .|....... ......
00000020  D6 A6 80 7B AD 69 D5 01-06 80 00 80 00 00 00 00  ...{.i..........
00000030  00 00 00 00 00 20 00 00-2A 00 3C 00 57 00 64 00  ..... ..*.<.W.d.
00000040  69 00 43 00 6F 00 6E 00-74 00 65 00 78 00 74 00  i.C.o.n.t.e.x.t.
00000050  4C 00 6F 00 67 00 2E 00-65 00 74 00 6C 00 2E 00  L.o.g...e.t.l...
00000060  30 00 30 00 32 00 00 00                          0.0.2...

Undo data:
-

USN record:
 Number: 10918043880
 Source:
 Reason: USN_REASON_DATA_EXTEND | USN_REASON_DATA_TRUNCATION |
USN_REASON_BASIC_INFO_CHANGE | USN_REASON_CLOSE
 File reference number: 7506656153896486794
 Parent file reference number: 7036874417798308
 Timestamp: 2019-09-12 21:03:09.958524
 File name: WdiContextLog.etl.002
```

Таким образом, аномалия с системными часами компьютера подтверждается еще раз.

Если отсортировать MFT-записи по полю «$STANDARD_INFORMATION USN value» (по возрастанию), то следующие файлы будут в конце списка:

Значение USN	Путь к файлу
10918043336	/Windows/ServiceProfiles/NetworkService/NTUSER.DAT
10918043416	/Users/U/AppData/Local/IconCache.db
10918043416	/Users/U/AppData/Local/ICONCA~1.DB
10918043504	/Windows/System32/LogFiles/WMI/RtBackup/EtwRTEventLog-System.etl
10918043504	/Windows/System32/LogFiles/WMI/RtBackup/ETWRTE~3.ETL
10918043648	/Windows/System32/WDI/LogFiles/BootPerfDiagLogger.etl
10918043648	/Windows/System32/WDI/LogFiles/BOOTPE~1.ETL
10918043752	/ProgramData/Microsoft/Windows/AppRepository/StateRepository-Machine.srd-wal
10918043752	/ProgramData/Microsoft/Windows/AppRepository/STATER~2.SRD
10918043880	/Windows/System32/WDI/LogFiles/WdiContextLog.etl.002
10918043880	/Windows/System32/WDI/LogFiles/WDICON~1.002
10918044504	/Users/U/Desktop/planted_evidence_.rtf
10918044504	/Users/U/Desktop/PLANTE~1.RTF
10918044696	/$Extend/$RmMetadata/$TxfLog/$TxfLog.blf

Илл. 13: Последние операции с файлами (по значению поля «$STANDARD_INFORMATION USN value»), модификация из операционной системы Windows 10 (системные часы установлены «задним числом»)

Если отсортировать MFT-записи по полю «Log file sequence number» (по возрастанию), то следующие файлы будут в конце списка:

Значение «Log file sequence number»	Путь к файлу
49745040528	/Windows/System32/LogFiles/WMI/RtBackup/EtwRTEventLog-System.etl
49745040528	/Windows/System32/LogFiles/WMI/RtBackup/ETWRTE~3.ETL
49745040652	/Windows/System32/WDI/LogFiles/BootPerfDiagLogger.etl
49745040652	/Windows/System32/WDI/LogFiles/BOOTPE~1.ETL
49745040730	/ProgramData/Microsoft/Windows/AppRepository/StateRepository-Machine.srd-wal
49745040730	/ProgramData/Microsoft/Windows/AppRepository/STATER~2.SRD
49745040805	/Windows/System32/WDI/LogFiles/WdiContextLog.etl.002
49745040805	/Windows/System32/WDI/LogFiles/WDICON~1.002
49745043934	/Users/U/Desktop/planted_evidence_.rtf
49745043934	/Users/U/Desktop/PLANTE~1.RTF
49745044650	/$Extend/$RmMetadata/$TxfLog/$Tops
49745044757	/$Extend/$RmMetadata/$TxfLog/$TxfLogContainer00000000000000000002
49745044957	/$Extend/$UsnJrnl
49745044976	/$Extend/$RmMetadata/$TxfLog/$TxfLog.blf
49745045047	/Users/U/Desktop

Илл. 14: Последние операции с файлами (по значению поля «Log file sequence number»), модификация из операционной системы Windows 10 (системные часы установлены «задним числом»)

Видно, что операции с файлом с именем «planted_evidence_.rtf» (в том числе создание записи о нем в индексе родительской директории) были одними из последних, хотя временные метки этого файла противоречат данному выводу.

Кроме того, данная файловая система имеет две теневых копии — от 8 сентября 2019 года и от 12 сентября 2019 года. Запись о файле с именем «planted_evidence_.rtf» в перечисленных теневых копиях отсутствует (т. е. файл не был создан 1 сентября 2019 года, иначе бы он попал в эти теневые копии).

В случае, когда копирование производилось из операционной системы Windows 7 (с отключенной антивирусной программой), зафиксировано следующее.

Последние события в журнале «USN change journal»:

Значение USN	Причина генерации события	Временная метка	Имя файла
10918043248	USN_REASON_DATA_EXTEND | USN_F	2019-09-12 21:03:09.958524	swapfile.sys
10918043336	USN_REASON_DATA_OVERWRITE | US	2019-09-12 21:03:09.958524	NTUSER.DAT
10918043416	USN_REASON_DATA_OVERWRITE | US	2019-09-12 21:03:09.958524	IconCache.db
10918043504	USN_REASON_DATA_OVERWRITE | US	2019-09-12 21:03:09.958524	EtwRTEventLog-System.etl
10918043648	USN_REASON_DATA_OVERWRITE | US	2019-09-12 21:03:09.958524	BootPerfDiagLogger.etl
10918043752	USN_REASON_DATA_EXTEND | USN_	2019-09-12 21:03:09.958524	StateRepository-Machine.srd-wal
10918043880	USN_REASON_DATA_EXTEND | USN_F	2019-09-12 21:03:09.958524	WdiContextLog.etl.002
10918043984	USN_REASON_FILE_CREATE	2019-08-15 22:26:06.834892	planted_evidence.docx
10918044088	USN_REASON_FILE_CREATE | USN_R	2019-08-15 22:26:06.834892	planted_evidence.docx
10918044192	USN_REASON_DATA_EXTEND	2019-08-15 22:26:06.835892	planted_evidence.docx
10918044296	USN_REASON_DATA_OVERWRITE | U	2019-08-15 22:26:06.836892	planted_evidence.docx
10918044400	USN_REASON_DATA_OVERWRITE | US	2019-08-15 22:26:06.836892	planted_evidence.docx
10918044504	USN_REASON_DATA_OVERWRITE | US	2019-08-15 22:26:06.836892	planted_evidence.docx
10918044608	USN_REASON_DATA_OVERWRITE	2019-08-15 22:26:19.064592	$TxfLog.blf
10918044696	USN_REASON_DATA_OVERWRITE | U	2019-08-15 22:26:19.068592	$TxfLog.blf

Илл. 15: Последние события в журнале «USN change journal» (второй столбец обрезан), модификация из операционной системы Windows 7 (системные часы установлены «задним числом»)

Первое событие в указанном журнале от 2019-09-12 17:36:09.055186 (то же время, что и в предыдущем случае).

В журнале «$LogFile» первой записью, относящейся к копированию файла с указанным ранее именем и содержащей временную метку, является следующая:

```
LSN: 49745043321
Transaction ID: 24
Log record, redo operation: AddIndexEntryAllocation, undo operation:
DeleteIndexEntryAllocation
Target (file reference number): 23643898043698760
Target (attribute name): $I30
Target path (from $MFT): /Users/U/Desktop
Offset in tagret: 7016
LCN(s): 1578189
Redo data:
00000000  20 05 00 00 00 00 7E 00-80 00 6C 00 00 00 00 00   .....~...l.....
00000010  48 0E 00 00 00 00 54 00-FB B7 62 6E B8 53 D5 01  H.....T...bn.S..
00000020  FB B7 62 6E B8 53 D5 01-FB B7 62 6E B8 53 D5 01  ..bn.S....bn.S..
00000030  FB B7 62 6E B8 53 D5 01-00 80 00 00 00 00 00 00  ..bn.S..........
00000040  00 00 00 00 00 00 00 00-20 00 00 00 00 00 00 00  ........ .......
00000050  15 01 70 00 6C 00 61 00-6E 00 74 00 65 00 64 00  ..p.l.a.n.t.e.d.
00000060  5F 00 65 00 76 00 69 00-64 00 65 00 6E 00 63 00  _.e.v.i.d.e.n.c.
00000070  65 00 2E 00 64 00 6F 00-63 00 78 00 02 00 00 00  e...d.o.c.x.....

Undo data:
-

$FILE_NAME in index:
 * M timestamp: 2019-08-15 22:26:06.834892
 * A timestamp: 2019-08-15 22:26:06.834892
 * C timestamp: 2019-08-15 22:26:06.834892
 * E timestamp: 2019-08-15 22:26:06.834892
 * File name: planted_evidence.docx
 * Parent (file reference number): 23643898043698760
 * Parent path (from $MFT): /Users/U/Desktop
```

Перед этой записью имеются другие записи, последняя из которых (имеющая в своем составе временную метку) имеет более позднюю временную метку:

```
LSN: 49745040747
Transaction ID: 24
Log record, redo operation: UpdateNonresidentValue, undo operation: Noop
Target (file reference number): 562949953475627
Target (attribute name): $J
Target path (from $MFT): /$Extend/$UsnJrnl
Offset in tagret: 10918043880
LCN(s): 3538814
Redo data:
00000000  68 00 00 00 02 00 00 00-8A 07 00 00 00 00 2D 68  h.............-h
00000010  A4 7C 00 00 00 00 19 00-E8 20 C4 8A 02 00 00 00  .|....... ......
00000020  D6 A6 80 7B AD 69 D5 01-06 80 00 80 00 00 00 00  ...{.i..........
00000030  00 00 00 00 00 20 00 00-2A 00 3C 00 57 00 64 00  ..... ..*.<.W.d.
00000040  69 00 43 00 6F 00 6E 00-74 00 65 00 78 00 74 00  i.C.o.n.t.e.x.t.
00000050  4C 00 6F 00 67 00 2E 00-65 00 74 00 6C 00 2E 00  L.o.g...e.t.l...
00000060  30 00 30 00 32 00 00 00                          0.0.2...

Undo data:
-

USN record:
 Number: 10918043880
 Source:
 Reason: USN_REASON_DATA_EXTEND | USN_REASON_DATA_TRUNCATION | USN_REASON_BASIC_
 File reference number: 7506656153896486794
 Parent file reference number: 7036874417798308
 Timestamp: 2019-09-12 21:03:09.958524
 File name: WdiContextLog.etl.002
```

Следовательно, аномалия с системными часами компьютера подтверждается таким же образом.

Если отсортировать MFT-записи по полю «$STANDARD_INFORMATION USN value» (по возрастанию), то следующие файлы будут в конце списка:

Значение USN	Путь к файлу
10918043336	/Windows/ServiceProfiles/NetworkService/NTUSER.DAT
10918043416	/Users/U/AppData/Local/IconCache.db
10918043416	/Users/U/AppData/Local/ICONCA~1.DB
10918043504	/Windows/System32/LogFiles/WMI/RtBackup/EtwRTEventLog-System.etl
10918043504	/Windows/System32/LogFiles/WMI/RtBackup/ETWRTE~3.ETL
10918043648	/Windows/System32/WDI/LogFiles/BootPerfDiagLogger.etl
10918043648	/Windows/System32/WDI/LogFiles/BOOTPE~1.ETL
10918043752	/ProgramData/Microsoft/Windows/AppRepository/StateRepository-Machine.srd-wal
10918043752	/ProgramData/Microsoft/Windows/AppRepository/STATER~2.SRD
10918043880	/Windows/System32/WDI/LogFiles/WdiContextLog.etl.002
10918043880	/Windows/System32/WDI/LogFiles/WDICON~1.002
10918044504	/Users/U/Desktop/planted_evidence.docx
10918044504	/Users/U/Desktop/PLANTE~1.DOC
10918044696	/$Extend/$RmMetadata/$TxfLog/$TxfLog.blf

Илл. 16: Последние операции с файлами (по значению поля «$STANDARD_INFORMATION USN value»), модификация из операционной системы Windows 7 (системные часы установлены «задним числом»)

Если отсортировать MFT-записи по полю «Log file sequence number» (по возрастанию), то следующие файлы будут в конце списка:

Значение «Log file sequence number»	Путь к файлу
49745040509	/Windows/System32/LogFiles/WMI/RtBackup/EtwRTEventLog-System.etl
49745040509	/Windows/System32/LogFiles/WMI/RtBackup/ETWRTE~3.ETL
49745040637	/Windows/System32/WDI/LogFiles/BootPerfDiagLogger.etl
49745040637	/Windows/System32/WDI/LogFiles/BOOTPE~1.ETL
49745040715	/ProgramData/Microsoft/Windows/AppRepository/StateRepository-Machine.srd-wal
49745040715	/ProgramData/Microsoft/Windows/AppRepository/STATER~2.SRD
49745040790	/Windows/System32/WDI/LogFiles/WdiContextLog.etl.002
49745040790	/Windows/System32/WDI/LogFiles/WDICON~1.002
49745043526	/Users/U/Desktop
49745044328	/Users/U/Desktop/planted_evidence.docx
49745044328	/Users/U/Desktop/PLANTE~1.DOC
49745044716	/$Extend/$RmMetadata/$TxfLog/$Tops
49745044823	/$Extend/$RmMetadata/$TxfLog/$TxfLogContainer00000000000000000002
49745045009	/$Extend/$UsnJrnl
49745045028	/$Extend/$RmMetadata/$TxfLog/$TxfLog.blf

Илл/ 17: Последние операции с файлами (по значению поля «Log file sequence number»), модификация из операционной системы Windows 7 (системные часы установлены «задним числом»)

Таким образом, тест привел к тем же результатам, что и предыдущий.

Четвертый ожидаемый результат подтвержден. Также подтвержден седьмой результат.

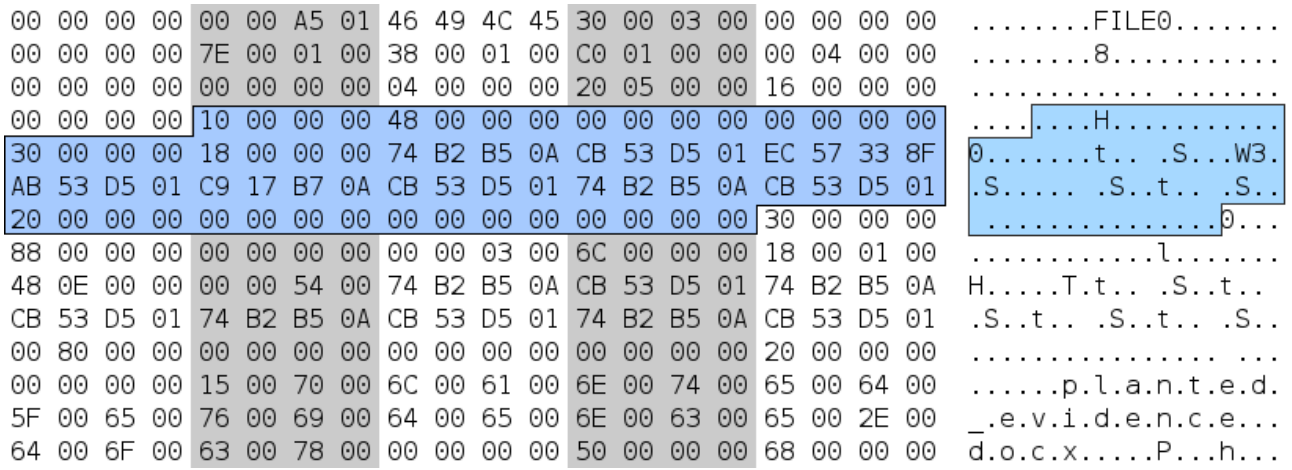
Также был проведен тест, аналогичный двум только что описанным, однако копирование осуществлялась из операционной системы Ubuntu 19.04.

После копирования установлено следующее.

Содержимое журналов «USN change journal» и «$LogFile» не было изменено (альтернативным исходом, при иных исходных данных, могло быть стирание журнала «$LogFile», что уже было описано выше).

Если отфильтровать MFT-записи по нулевому значению поля «Log file sequence number», то будут обнаружены следующие файлы: «/$Boot» и «/Users/U/Desktop/planted_evidence.docx».

Скопированный файл явно выделяется среди остальных. Кроме того, данный файл не имеет установленного значения поля «$STANDARD_INFORMATION USN value», что является еще одним признаком; это обусловлено тем, что драйвер записал укороченную (устаревшую) версию атрибута $STANDARD_INFORMATION, которая не содержит данное поле.



Илл. 18: Атрибут $STANDARD_INFORMATION (выделен), не содержащий поля «USN value» (это поле должно быть по смещению, которое выходит за границы выделения)

Таким образом, пятый ожидаемый результат подтвержден.

В случае, если копирование производится с использованием операционной системы Windows 10 на компьютере, системные часы которого установлены в текущее время, с дальнейшим изменением (подделкой) временных меток программой timestomp, результаты следующие (в тесте временные метки изменялись на 8 августа 2019 года).

Последние события в журнале «USN change journal»:

Значение USN	Причина генерации события	Временная метка	Имя файла
10918043504	USN_REASON_DATA_OVERWRITE | USN_	2019-09-12 21:03:09.958524	EtwRTEventLog-System.etl
10918043648	USN_REASON_DATA_OVERWRITE | USN_	2019-09-12 21:03:09.958524	BootPerfDiagLogger.etl
10918043752	USN_REASON_DATA_EXTEND | USN_REA	2019-09-12 21:03:09.958524	StateRepository-Machine.srd-wal
10918043880	USN_REASON_DATA_EXTEND | USN_REA	2019-09-12 21:03:09.958524	WdiContextLog.etl.002
10918043984	USN_REASON_FILE_CREATE	2019-09-21 11:12:37.620596	planted_evidence.rtf
10918044088	USN_REASON_FILE_CREATE | USN_REA	2019-09-21 11:12:37.620596	planted_evidence.rtf
10918044192	USN_REASON_DATA_EXTEND	2019-09-21 11:12:37.620596	planted_evidence.rtf
10918044296	USN_REASON_DATA_OVERWRITE | USN_	2019-09-21 11:12:37.620596	planted_evidence.rtf
10918044400	USN_REASON_DATA_OVERWRITE | USN_	2019-09-21 11:12:37.636128	planted_evidence.rtf
10918044504	USN_REASON_DATA_OVERWRITE | USN_	2019-09-21 11:12:37.636128	planted_evidence.rtf
10918044608	USN_REASON_BASIC_INFO_CHANGE	2019-09-21 11:13:40.312708	planted_evidence.rtf
10918044712	USN_REASON_BASIC_INFO_CHANGE | U	2019-09-21 11:13:40.312708	planted_evidence.rtf
10918044816	USN_REASON_DATA_OVERWRITE	2019-09-21 11:13:58.643172	$TxfLog.blf
10918044904	USN_REASON_DATA_OVERWRITE | USN_	2019-09-21 11:13:58.647452	$TxfLog.blf

Илл. 19: Последние события в журнале «USN change journal» (второй столбец обрезан), модификация из операционной системы Windows 10 (показания системных часов соответствуют текущим)

Данные события датируют копирование файлы действительным временем совершения этого действия.

Также в журнале «$LogFile» была обнаружена запись об изменений временных меток скопированного файла (с подлинных на поддельные):

```
LSN: 49745047819
Transaction ID: 24
Log record, redo operation: UpdateResidentValue, undo operation: UpdateResidentValue
Target (file number): 1312
Target path (from $MFT, likely wrong if the file was deleted later):
/Users/U/Desktop/planted_evidence.rtf
Offset in tagret: 80
LCN(s): 786760
```

```
Redo data:
00000000  80 65 D9 55 F3 4D D5 01-80 65 D9 55 F3 4D D5 01  .e.U.M...e.U.M..
00000010  80 65 D9 55 F3 4D D5 01-80 65 D9 55 F3 4D D5 01  .e.U.M...e.U.M..
00000020  20 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ...............
00000030  00 00 00 00 A4 4C 00 00-00 00 00 00 00 00 00 00  .....L..........
00000040  58 23 C4 8A 02 00 00 00                          X#......

Undo data:
00000000  89 B2 E6 79 6D 70 D5 01-7C A9 DC E4 65 70 D5 01  ...ymp..|...ep..
00000010  7C A9 DC E4 65 70 D5 01-36 11 E9 79 6D 70 D5 01  |...ep..6..ymp..
00000020  20 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ...............
00000030  00 00 00 00 A4 4C 00 00-00 00 00 00 00 00 00 00  .....L..........
00000040  58 23 C4 8A 02 00 00 00                          X#......

Possible update to $STANDARD_INFORMATION (redo data):
 * M timestamp: 2019-08-08 14:12:39.000000
 * A timestamp: 2019-08-08 14:12:39.000000
 * C timestamp: 2019-08-08 14:12:39.000000
 * E timestamp: 2019-08-08 14:12:39.000000

Possible update to $STANDARD_INFORMATION (undo data):
 * M timestamp: 2019-09-21 10:18:21.097202
 * A timestamp: 2019-09-21 11:12:37.636128
 * C timestamp: 2019-09-21 11:12:37.620596
 * E timestamp: 2019-09-21 10:18:21.097202
```

Кроме того, дробная часть поддельных временных меток равна нулю, что является известным наглядным признаком их подделки. В данном случае использование программы timestomp не имело целью установку временных меток с правдоподобной дробной частью.

Таким образом, шестой ожидаемый результат подтвержден.

Возможны ситуации, когда скопированный файл открывается на компьютере, с использованием которого производилось копирование. При этом системные часы этого компьютера установлены «задним числом».

Был проведен тест, аналогичный двум ранее описанным (при подтверждении четвертого и седьмого результатов); копирование производилось с использованием операционных систем Windows 10 и Windows 7 (со включенной антивирусной программой). Результаты следующие.

Скопированному файлу (в каждом тесте), после его открытия в связанной программе для редактирования, был назначен новый атрибут $OBJECT_ID. Этот атрибут имеет уникальный идентификатор, генерируемый на основе MAC-адреса основного сетевого интерфейса компьютера (если таковой имеется, в противном случае используется произвольное значение MAC-адреса) и временной метки (которая устанавливается в текущее время на начальном этапе загрузки операционной системы, а затем обновляется по мере исчерпания запаса кешированных уникальных идентификаторов; в большинстве случаев эта временная метка указывает на примерное время загрузки операционной системы, поскольку запас кешированных уникальных идентификаторов исчерпывается редко).

Так, в случае с операционной системой Windows 10, файл был скопирован якобы 1 января 2019 года, а временная метка в уникальном идентификаторе в атрибуте $OBJECT_ID указывала на реальный день, когда производилось копирование, — 21 сентября 2019 года. Такая временная метка показывает, что перевод системных часов компьютера в прошлое (на

1 января 2019 года) произошел после загрузки операционной системы этого компьютера (когда время еще не было переведено в прошлое). MAC-адрес внутри указанного идентификатора соответствовал MAC-адресу сетевой карты компьютера, на котором производилось копирование.

В случае с операционной системой Windows 7, файл был скопирован якобы 30 декабря 2018 года, а временная метка в уникальном идентификаторе в атрибуте $OBJECT_ID указывала на реальный день, когда производилось копирование, — 21 сентября 2019 года. MAC-адрес внутри указанного идентификатора был установлен произвольно (поскольку в операционной системе отсутствовал драйвер для сетевой карты). Следует отметить, что в произвольном MAC-адресе нет однозначных признаков, указывающих на его генерацию (вместо использования MAC-адреса основного сетевого интерфейса компьютера): хотя стандартизированный алгоритм создания уникальных идентификаторов на основе метки времени и MAC-адреса предусматривает установку мультикаст-бита (младший бит первого октета) в единицу, если MAC-адрес не был известен и его пришлось сгенерировать случайно или использовать предварительно заданный набор байтов, в реализации в ядре операционных систем семейства Windows устанавливается старший бит первого октета (что является ошибкой, поскольку некоторые корректные MAC-адреса имеют этот бит установленным в единицу).

Восьмой ожидаемый результат подтвержден.

Копирование нового файла в файловую систему, в которой установлена операционная система, находящаяся в режиме гибернации, может, как было сказано ранее, создать противоречия между данными на накопителе и данными в оперативной памяти (которые записаны в файл гибернации).

К сожалению, по состоянию на начало 2020 года, какие-либо программные средства, позволяющие проанализировать память драйвера NTFS на предмет наличия указанных противоречий, отсутствуют (существующие программные средства ограничиваются, например, обнаружением в памяти файловых записей, что не дает точного представления о состоянии файловой системы). Такие средства могли бы извлекать битовую карту, содержащую указатели на свободные файловые записи в файловой системе, а затем сравнивать ее с битовой картой на накопителе; результатом был бы перечень файлов, отсутствовавших на момент входа в режим гибернации.

В качестве рабочего метода исследования можно предложить эксперимент с выходом из режима гибернации, который заключается в загрузке с клона накопителя на оригинальном (конфигурация которого соответствует той, что была на момент входа в режим гибернации) аппаратном обеспечении. В некоторых случаях может потребоваться загрузка с оригинала накопителя, а не с клона (в таком случае перед попыткой загрузки целесообразно сделать точную копию этого накопителя). После успешной загрузки необходимо проверить «видимость» и возможность чтения (открытия) файлов, в отношении которых имеются сомнения, штатными средствами операционной системы (например, программой Explorer). Затем можно перезагрузить компьютер и проверить, требуется ли во время загрузки запуск

сканирования файловой системы на ошибки и не возникает ли «синий экран смерти» с кодом, соответствующим ошибке в файловой системе; подобное указывает на обнаруженные противоречия в служебных структурах (и, напротив, отсутствие необходимости сканирования или «синего экрана смерти» еще не означает, что таких противоречий нет). Если первая тестовая попытка загрузки операционной системы не удалась (такое наблюдалось при включенном шифровании BitLocker для системного тома, в который затем копировались файлы), то проверка «видимости» и возможности чтения (открытия) файлов может быть произведена (после неуспешной попытки загрузки) в другой операционной системе.

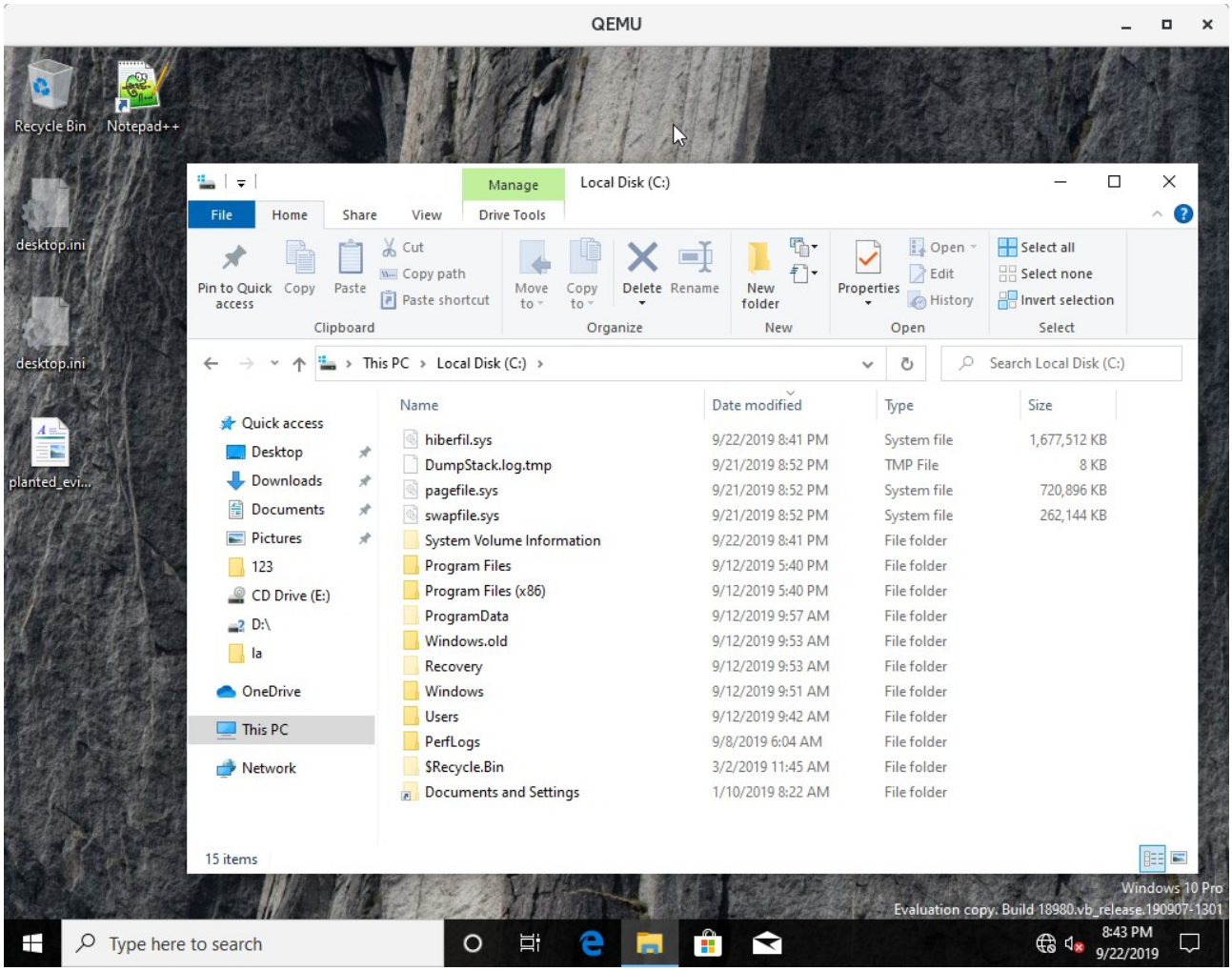
В ходе многочисленных тестов установлено, что копирование нового файла при описанных обстоятельствах приводит к тому, что этот файл часто отсутствует в списке файлов, отображаемых программой Explorer. А после выключения компьютера файловая запись, ранее использованная скопированным файлом, оказывается используемой другим файлом. Если же файл присутствует в списке файлов, отображаемых программой Explorer, то иногда попытка чтения этого файла завершается ошибкой.

Так, в ходе одного из тестов, проведенного в виртуальной машине, на накопитель были скопированы следующие файлы: «/Users/U/Desktop/planted_evidence_.rtf» и «/planted_evidence_.rtf».

После загрузки последний файл оказался отсутствующим в числе отображаемых программой Explorer (в то же время первый файл присутствовал в этом списке). А после выключения компьютера MFT-запись о файле «/planted_evidence_.rtf» «исчезла» (была выделена другому файлу).

Путь к файлу	**Временная метка модификации**	**Временная метка создания**
/Users/U/Desktop/planted_evidence_.rtf	2019-09-01 15:58:15.715706	2019-09-02 19:53:11.689448
/planted_evidence_.rtf	2019-09-01 15:58:15.715706	2019-09-02 19:53:13.989178

Илл. 20: Сведения о скопированных файлах, полученные до эксперимента с загрузкой



Илл. 21: Видимые в загруженной операционной системе файлы (присутствует скопированный на рабочий стол файл, отсутствует файл в корне файловой системы)

Девятый ожидаемый результат подтвержден (с тем условием, что предложенный эксперимент не всегда дает точный в плане обнаружения всех скопированных файлов результат).

В ходе небольшого количества тестов (из числа описанных выше, описанное далее выявлялось не во всех тестах), когда несъемный накопитель подключался к компьютеру под управлением операционной системы Windows 10, был выявлен факт создания на этом накопителе (в подключенной файловой системе, относящейся к системному тому) директорий «корзины» для учетной записи пользователя указанного компьютера. При этом путь к создаваемой директории содержал идентификатор безопасности этой учетной записи, например (идентификатор безопасности выделен жирным): «/$Recycle.Bin/**S-1-5-21-1738794144-21243219-2192724015-1001**»; этот идентификатор отсутствует у учетных записей пользователей в операционной системе, установленной на несъемном накопителе. Данное обстоятельство является еще одним, десятым, признаком подключения накопителя к другому компьютеру, однако этот признак не всегда проявляется:

10. Подключение носителя к компьютеру под управлением операционной системы семейства Windows может в некоторых случаях приводить к созданию директории

«корзины» для учетной записи пользователя, идентификатор безопасности которой отсутствует в установленной на этом носителе операционной системе, при этом данный идентификатор присутствует (в текстовом виде) в имени директории «корзины».

Таким образом, перечисленные выше результаты тестов, будучи дополненными действиями по получению сведений об известных временных интервалах работы операционной системы, об известных событиях синхронизации системных часов с серверами точного времени, а также о показаниях системных часов (их отклонения от текущего времени) на момент тестового включения изъятого (изымаемого) компьютера (без подключенных накопителей), формируют методику обнаружения изменений, внесенных в файловые системы NTFS после даты изъятия соответствующего носителя (т. е. методику обнаружения нарушений целостности компьютерной информации в файловой системе NTFS).

Методика определения давности форматирования накопителя в файловую систему NTFS

В ситуациях, когда необходимо определить давность форматирования накопителя в файловую систему NTFS, например, в случаях, когда имеющие отношение к делу файлы были скопированы на «чистый» накопитель, уникальные номера, которые показывают порядок действий, совершенных над файловой системой (вне зависимости от показаний системных часов компьютера), имеют малую значимость, поскольку, даже в случае установки системных часов компьютера «задним числом», порядок действий будет ожидаемым.

В качестве признака, позволяющего установить время, не ранее которого было произведено форматирование накопителя в файловую систему NTFS, можно использовать номер сборки операционной системы семейства Windows, который записывается в каждую файловую систему NTFS при форматировании (в Windows 8, 8.1 и 10). Наиболее информативными являются номера сборок в Windows 10, поскольку эта операционная система регулярно обновляется с увеличением номера сборки.

Номер сборки хранится в файле «/$UpCase:$Info» (имя файла — «$UpCase», данные находятся в альтернативном потоке с именем «$Info»), пример данных из этого файла:

```
00000000  20 00 00 00 00 00 00 00  0c 69 1b 6b 77 7e dc da  | ........i.kw~..|
00000010  0a 00 00 00 00 00 00 00  ba 47 00 00 00 00 00 00  |.........G......|
```

Таблица 2: Пример данных в файле «/$UpCase:$Info»

Формат данных:

Смещение поля (в байтах)	**Длина поля (в байтах)**	**Поле (числа в представлении «Little Endian»)**	**Интерпретированные данные (из примера выше)**
0	4	Размер данных в байтах	32 *(фиксированное значение)*
4	4	Не используется	-
8	8	Контрольная сумма CRC-64	0c 69 1b 6b 77 7e dc da

		от содержимого таблицы UpCase	
16	4	Старшая версия операционной системы	10
20	4	Младшая версия операционной системы	0
24	4	Номер сборки операционной системы	18362 *(название: «May 2019 Update»)*
28	4	Номер сервис-пака операционной системы	0

Таблица 3: Формат данных в файле «/$UpCase:$Info»

Данные в вышеописанном файле используются для идентификации и контроля целостности таблицы перевода символов в верхний регистр, которая хранится в файле «/$UpCase» и используется в конкретном экземпляре файловой системы NTFS (сама таблица применяется, в том числе, для сравнения без учета регистра символов имен файлов в ходе их размещения в дереве индекса директории — для такого сравнения имена необходимо перевести в верхний регистр). Разные версии операционных систем семейства Windows могут использовать разные таблицы перевода символов в верхний регистр, а потому драйвер файловой системы NTFS должен использовать именно ту таблицу, которая хранится в самой файловой системе, иначе результаты сравнения двух имен файлов могут различаться в зависимости от версии применяемой таблицы (что, в свою очередь, приведет к тому, что одна из версий операционной системы посчитает дерево индекса директорий поврежденным из-за нарушения порядка сортировки имен файлов).

Файл «/$UpCase:$Info» используется только в операционных системах Windows 8, 8.1 и 10. Тем не менее, если файловая система NTFS, созданная (отформатированная) в более ранней версии операционной системы семейства Windows, окажется подключена к компьютеру под управлением одной из только что перечисленных версий операционных систем, то у файла «/$UpCase» будет создан альтернативный поток данных с именем «$Info» и содержимым, соответствующим формату (в нем указывается версия операционной системы, определяемая по алгоритму, описанному ниже).

В ходе тестирования установлено, что вне зависимости от применяемого типа накопителя (съемный или несъемный) в ходе его форматирования в указанный альтернативный поток данных записываются сведения об операционной системе (в т. ч. номер сборки), из которой осуществляется это форматирование.

В ходе обратной разработки официального драйвера NTFS и сопутствующего тестирования установлено, что алгоритм создания (обновления) содержимого файла «/$UpCase:$Info» у существующих файловых систем NTFS следующий:

1. Если альтернативный поток данных с именем «$Info» не существует, то переходим к пункту 3.

2. Если контрольная сумма содержимого таблицы перевода символов в верхний регистр совпадает с указанной в альтернативном потоке данных с именем «$Info», то алгоритм завершается (т. е. существующие сведения о версии операционной системы не изменяются).
3. Происходит проверка соответствия содержимого таблицы перевода символов в верхний регистр возвращаемым значениям функции *RtlUpcaseUnicodeChar()* (для всех поддерживаемых таблицей кодов символов). Перевод символов в верхний регистр указанной функцией осуществляется согласно глобальной таблице перевода символов в верхний регистр (которая используется операционной системой, в т. ч. для вызовов прикладных программ). Если содержимое таблицы перевода символов в верхний регистр полностью соответствует возвращаемым значениям функции *RtlUpcaseUnicodeChar()*, то в альтернативный поток данных с именем «$Info» записываются сведения о версии текущей операционной системы, затем алгоритм завершается.
4. Проверяется соответствие вычисленной контрольной суммы известным значениям, хранимым в драйвере файловой системы NTFS. Каждому известному значению контрольной суммы соответствуют сведения о версии операционной системы, в которой используется именно такая таблица перевода символов в верхний регистр. Если совпадение по вычисленной контрольной сумме найдено, то в альтернативный поток данных с именем «$Info» записываются сведения о такой операционной системе, затем алгоритм завершается.
5. Алгоритм завершается, файловая система NTFS считается поврежденной.

Как видно, обновление уже имеющихся в файловой системе NTFS сведений о версии операционной системы возможно лишь в случае, когда таблица перевода символов в верхний регистр повреждена.

В версии драйвера NTFS 10.0.19551.1001 используется следующая таблица известных значений (в формате: «контрольная сумма (байтовая строка): старшая версия.младшая версия.номер сборки номер сервис-пака»):

- `0C 69 1B 6B 77 7E DC DA: 6.1.7600 0;`
- `02 25 5D 96 4C 87 10 BB: 6.0.6001 1;`
- `69 EA B7 DD E9 0D 45 A8: 5.1.2600 0.`

По состоянию на начало 2020 года, поддерживаемые операционные системы семейства Windows и драйвер ntfs-3g используют таблицу с контрольной суммой «0c 69 1b 6b 77 7e dc da». При этом драйвер ntfs-3g устанавливает все значения, кроме размера данных и контрольной суммы, в нулевые байты.

Таким образом, исходя из имеющейся в файловой системе NTFS информации о номере сборки операционной системы семейства Windows можно установить время, раньше которого соответствующий накопитель не мог быть отформатирован (по причине того, что

такая версия операционной системы еще не была выпущена). Для номера сборки 18362 (из примера выше) это будет 20 марта 2019 года (дата выпуска соответствующей сборки по программе «Insider Preview» в канале «Fast Ring»).

Номера сборок для разных версий операционной системы Windows 10 приведены в Википедии [11].

Заключение

В настоящей работе рассмотрены факторы, влияющие на целостность компьютерной информации во время судебной экспертизы и иных действий, в том числе в ходе копирования компьютерной информации.

Предложена и обоснована методика обнаружения изменений, внесенных в файловые системы NTFS после даты изъятия носителя. Данная методика может быть использована для относительной датировки файлов, имеющих значение для уголовного или иного дела, при возникновении каких-либо сомнений в их происхождении.

Также предложена и обоснована методика определения давности форматирования накопителя в файловую систему NTFS, которая может использоваться для примерной датировки события записи файлов, имеющих значение для уголовного или иного дела, при возникновении сомнений в том, что эта запись имела место в заявленное время.

Концепция относительной датировки может быть расширена и на другие типы файловых систем. Например, в практике автора служебные файлы сервиса Spotlight, функционирующего в операционных системах семейства macOS, помогли установить обстоятельства (прежде всего, границу временного интервала, не ранее которой произошло событие) удаления некоторых файлов на накопителе типа USB Flash, отформатированном в файловую систему семейства FAT: указанные служебные файлы хранятся на самом съемном накопителе, а первое событие подключения этого накопителя к компьютеру под управлением названной операционной системы (которое совпало с датой изъятия) оставило следы в виде базы данных (в файле с именем «psid.db») со списком файлов, представленных (на момент подключения) в явном (неудаленном) виде (сам список в указанной базе данных представлен в неявном виде, т. е. в виде фрагмента удаленной записи). В дальнейшем сравнение этого списка с перечнем файлов, полученным из структур файловой системы, позволило выявить файлы, удаленные после изъятия накопителя.

Список литературы

1. Bell, Graeme B. and Boddington, Richard (2010) "Solid State Drives: The Beginning of the End for Current Practice in Digital Forensic Recovery?," Journal of Digital Forensics, Security and Law: Vol. 5 : No. 3 , Article 1.

   DOI: https://doi.org/10.15394/jdfsl.2010.1078

   Available at: https://commons.erau.edu/jdfsl/vol5/iss3/1

2. Digital Corpora. "nps-2014-usb-nondeterministic." URL: https://digitalcorpora.org/corpora/disk-images/nps-2014-usb-nondeterministic (accessed February 1, 2020).
3. Forensic Science Regulator. "Method validation in digital forensics." URL: https://www.gov.uk/government/publications/method-validation-in-digital-forensics (accessed February 1, 2020).
4. U.S. Department of Justice. "Test Results for Digital Data Acquisition Tool: ASR Data SMART version 2010-11-03." URL: https://www.dhs.gov/sites/default/files/publications/508_Test%20Report_NIST_Digital%20Data_ASR_Data_SMART_September%202012.pdf (accessed February 1, 2020).
5. Department of Homeland Security. "Test Results for Digital Data Acquisition Tool: Paladin 3.0." URL: https://www.dhs.gov/sites/default/files/publications/508_Test%20Report_Paladin%203%200_%20October%202015_Final.pdf (accessed February 1, 2020).
6. U.S. Department of Homeland Security. "Test Results for Digital Data Acquisition Tool: Paladin 4.0." URL: https://www.dhs.gov/sites/default/files/publications/508_Test%20Report_NIST_Paladin%204%200_August%202015_Final_0.pdf (accessed February 1, 2020.
7. U.S. Department of Homeland Security. "Test Results for Disk Imaging Tool: Paladin Version 6.09 (ewfacquire 20160403)." URL: https://www.dhs.gov/sites/default/files/publications/1496_508_Test%20Report_NIST_Disk%20Imaging_Paladin%20v6.09_October_14_2016.pdf (accessed February 1, 2020).
8. U.S. Department of Homeland Security. "Test Results for Digital Data Acquisition Tool: DCFLDD 1.3.4-1." URL: https://www.dhs.gov/sites/default/files/publications/DCFLDD%201%203%204-1%20Test%20Report_updated.pdf (accessed February 1, 2020).
9. Spencer, Mark. "APPLYING ANCHORS IN RELATIVE TIME." URL: https://arsenalexperts.com/persistent/resources/news/DFM_Applying_Anchors_in_Relative_Time.pdf (accessed February 1, 2020).
10. Шапошников А. Ю. Проблема обеспечения подлинности объектов, направляемых для экспертного исследования: системный подход // Юридический вестник СамГУ. 2015. Т. 1. №4. С. 92–102.
11. Wikipedia contributors. "Windows 10 version history." URL: https://en.wikipedia.org/wiki/Windows_10_version_history (accessed February 2, 2020).