Forensische Informatik
Zusammenfassung des Wichtigsten

Sommersemester 2016

Prof. Dr.-Ing. Felix Freiling

Autor:

- Christian Strate

Inhaltsverzeichnis

0 Organisatorisches

- Materialien: Studon
- Pro Aufgabe ein 10-15 Seitiges Protokoll
- Die Protokolle mit in die Prüfung nehmen, sollten sich nach Abgabe Dinge geändert haben, kann trotzdem die neuere Version mitgenommen werden und dann sollte man ansprechen warum sich was geändert hat und warum man da nicht früher drauf gekommen ist.
- Jo geil, Primärteil der Prüfung wird der dritte Bericht sein, die anderen beiden nur eingeschränkt.

1 Intro

- explizite Spuren: Existenz zum Zweck des Blicks in die Vergangenheit
- implizite Spuren: Ursprünglich zu einem anderen Zweck
- Persönlichkeitsrechte der Betroffenen sind bei der Forensic zu schützen. Bsp: unbekannte DNA darf lediglich verwendet werden um das Geschlecht der Person zu bestimmen, da alle weiteren Informationen zu tiefe Einschnitte in die Intimität wären.

2 Dateisystemanalyse-Methodik

Referezmodell nach Carrier:

- Dateisystemdaten
  - Allgemeine Dateisysteminformationen
  - Sollen eine Orientierung im Datenträger ermöglichen
  - Bietet einen Ansatz zum Finden von Inhaltsdaten, Metadaten, Anwendungsdaten
  - Wie groß ist das Dateisystem, wo liegen welche Daten, wo liegen Kopien
  - Liegen meistens am Anfang der Partition (also des Dateisystems), sonst müsste man wissen wo diese liegen $\mapsto$ fsstat mit sleuth kit
- Inhaltsdaten
  - Inhalt von Dateien
  - Sind das Elementare zu speichernde.
  - Werden in Blöcken gespeichert
- Metadaten
  - Daten über die Inhaltsdaten (Inodes in Unix)
- Dateinamensdaten
  - Benutzerschnitstellendaten
- Anwendungsdaten
  - Werden für den korrekten Dateisystembetrieb eigentlich nicht benötigt
  - Quota
  - Statistken

Platz am Ende einer Partition, die nicht vom Dateisystem verwendet wird. Also partitionierter, ungenutzter Platz. Dieser nennt sich Volume Slack. Dort liegen häufig Daten von früheren Benutzungen.

Essentielle Daten nach Carrier - Strikt essentielle Daten:

Essentielle Daten sind diese, die notwendig sind, um Daten speichern und wieder abrufen zu können. Diese sind vertrauenswürdiger als *nicht-essentielle Daten*, da die Konsistenz notwendig ist, um das Dateisystem nutzen zu können. Beispiel: Dateinamen, oder Verweise auf die Festplattenblöcke. Beispiel für nicht-essentielle: Zeitstempel etc.

Sind unabhängig vom Betriebssystem essentielle Daten

Partiell essentielle Daten:

Können abhängig vom Betriebssystem essentielle Daten sein. Z.B. kann Windows ein Fat-Dateisystem nicht mehr mounten, wenn die Bitfolge zur Signalisierung eines Fat-Dateisystems, fehlt - Linux kann das trotzdem.

Fragmentierte Dateien bestehen selten aus mehr als zwei Fragmenten. Die allermeisten bestehen aus nur einem Fragment.

Slack Space:

Gegeben sei ein Dateisystemblock, der aus mehreren Festplattenblöcken besteht, dann gibt es Slack Space verschiedener Arten.

- restlicher Sektor, restlicher Block - Der Bereich eines Blocks der nicht mehr durch Dateiinhalt ausgefüllt wird - werden heutzutage meist ausgenullt
- restlicher Cluster - nicht mit Dateiinhalten befüllte Blöcke - hier sind häufig noch Daten früherer Benutzungen auffindbar (Dieser Slack Space existiert in SSDs unter Umständen nur ausgenullt)

Dateiwiederherstellung auf Basis von Metadaten:

Unter der Annahme, dass die Daten sich noch physikalisch auf der Platte befinden. Es können die Referenzen der Metadaten auf die ursprünglichen Daten noch vorhanden sein. Fehlen diese so muss gecarved werden. Es kann auch eine weitere Metadatei geben, die die selben Blöcke der Datei verwenden, wie es eine vorherige Metadatei tat, wenn diese nun auch gelöscht wird, wird eine Zuordnung der Blöcke und Metadaten (Dateisystemeinträgen) schwierig. Daten können auch in badclustern versteckt werden. Also Blöcke, die vom Dateisystem als beschädigt markiert werden, die Festplatte weiß von der Beschädigung aber nichts.

Analyse von Metadaten:

Dateiname verweist typischerweise auf Metadaten - Metadaten eines Eintrags lassen sich mithilfe von *istat* (Sleuthkit) auslesen. Slack Space kann auch mit *icat -s* betrachtet werden. Unbenutzte Metadaten können helfen Dateien zu verstecken, diese können mit *ils* untersucht werden.

Analyse von Dateinamen:

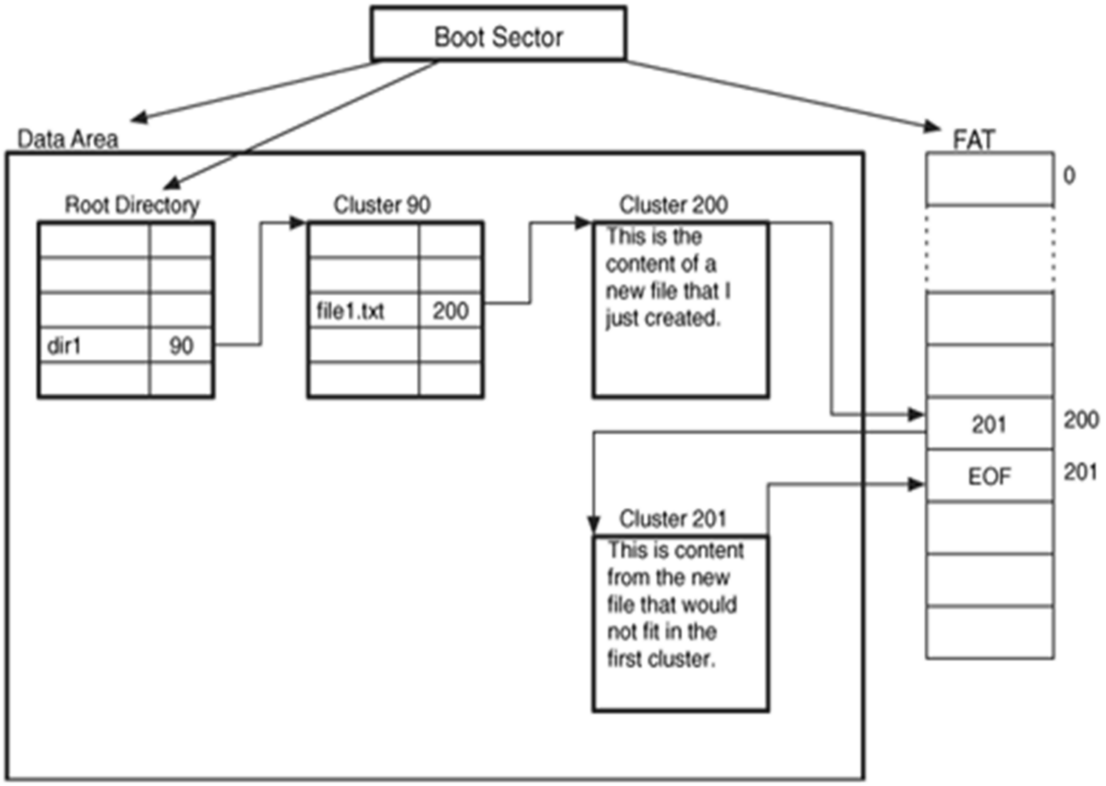
Löschen einer Datei in FAT passiert indem der Dateiname gelöscht wird. Dateinamen lassen sich mit *fls* auflisten.

3 FAT - File Allocation Table

- Jede Datei und jedes Verzeichnis besitzt einen Verzeichniseintrag (directory entry)
- Verzeichniseintrag: Metadaten: Dateiname, Dateigröße, Verweis auf den ersten Cluster der Festplatte, der Inhaltsdaten dieser Datei speichert. Der nächste Clustereintrag, der zu der Datei gehört lässt sich aus der FAT-Tabelle auslesen. Man schaut also in der FAT-Tabelle nach dem Eintrag, der in dem Verzeichniseintrag genannt wurde und der dort stehende Clusterblock ist der nächste der Datei etc. Bei dem letzten Cluster einer Datei steht EOF in der FAT-Tabelle.
- FAT-Dateisystem-Layout
  - reservierter Bereich (reserved area) - Boot-Sektor (BPB - BIOS Parameter Block), Dateisystemdaten - liegt an Sektor 0, Größe steht im Boot-Sektor
  - FAT-Bereich (FAT area)
  - Datenbereich (data area)
  - Wurzelverzeichnis kann in FAT32 überall im Datenbereich liegen (Verweis darauf im Boot-Sektor), bei allen anderen liegt es an dessen Anfang
  - Im Boot-Sektor steht häufig FAT12, FAT16, FAT32, was nicht der Wahrheit entsprechen muss
  - Häufig hinterlassen die Tools, die das Dateisystem erzeugen Spuren, unter Linux erstelltes Windows erzeugt den String mkdosfs, das kann aber händisch nachträglich geändert werden
  - Clusterzählung in der FAT-Tabelle beginnen bei 2, denn 0 ist die Markierung für unbelegt.
  - Eine beschädigt-Markierung wird bei heutiger Firmware nur noch selten genutzt, d.h. eine solche Markierung ist ungewöhnlich. Die Markierung ist abhängig von der FAT-Version FAT12: 0xFF7, FAT16: 0xFFF7, FAT32: 0x0FFFFFF7
  - Größenfeld im Verzeichniseintrag sollte immer 0 sein, sonst auffällig.

Erstellen und Löschen einer Datei - figs. 1a and 2a:

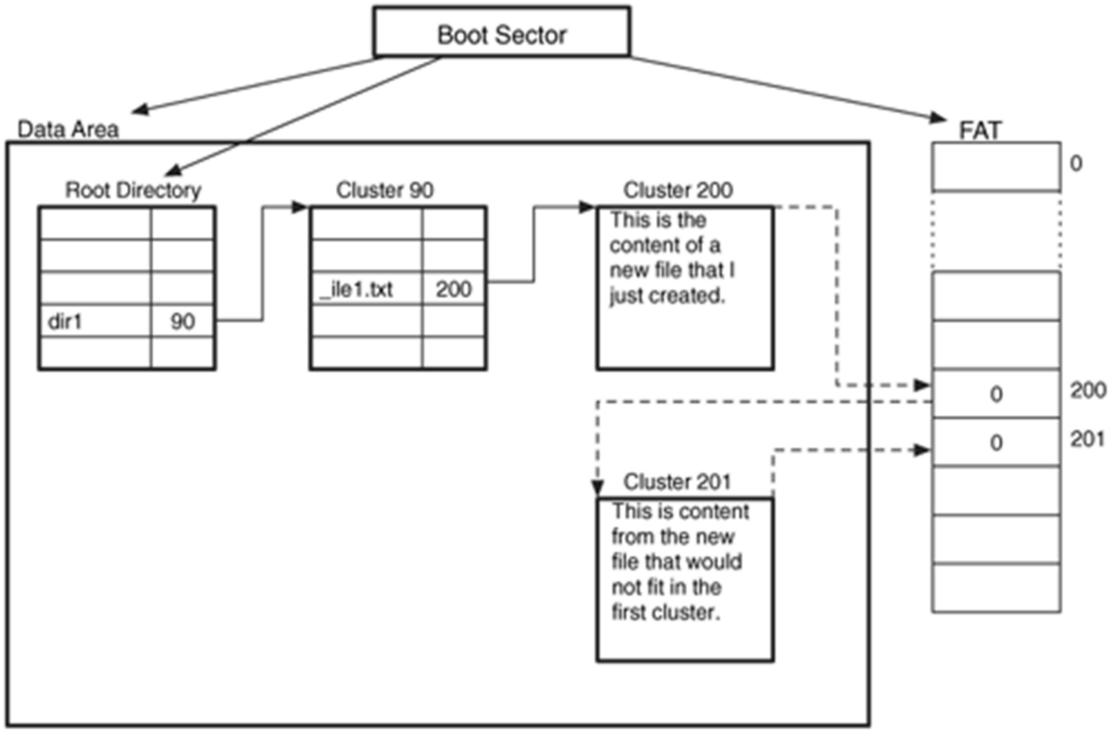
- Dateien können wiederhergestellt werden, verlieren dabei jedoch den ersten Buchstaben ihres Namens (weil dieser beim Löschen überschrieben wurde) - d.h. hier kann es einen Namenskonflikt geben, wenn es zwei gelöschte Verzeichnisse gibt, die sich nur im ersten Buchstaben unterscheiden. Einige Tools können das aber vielleicht vollständig rekonstruieren, vermutlich dank der langen und kurzen Dateinamen, die es in FAT gibt.
- Wiederhergestellte Verzeichnisse, die auf ihren Ursprünglichen Cluster verweisen, müssen damit nicht auf den ursprünglichen Inhalt verweisen, denn eine andere Datei könnte diese Cluster in der Zwischenzeit verwendet haben.
- Gelöschte Verzeichnisse lassen sich mit carving nach . bzw .. finden



(a) Erstellung der Datei dir1\file1.txt

1. Lese Boot-Sektor, identifiziere FAT-Position, Datenbereich und Wurzelverzeichnis
2. Suche nach dir1 im Wurzelverzeichnis (Cluster 90)
3. Lese Inhalt von Cluster 90 und suche nach unbelegtem Verzeichniseintrag
4. Belege Verzeichniseintrag mit *file1.txt*
5. Belege Cluster für den Dateiinhalt im FAT
6. Belege Cluster schrittweise mit Dateninhalt
7. Erzeuge entsprechende Verkettung der FAT-Einträge

Abbildung 1: Erstellung der Datei dir1\file1.txt



(a) Löschen der Datei dir1\file1.txt

1. Erstes Zeichen von *file1.txt* auf _ setzen
2. FAT-Einträge auf 0 setzen

Abbildung 2: Löschen der Datei dir1\file1.txt

4 NTFS - New Technologies File System

Proprietäres Format, ohne offizielle Dokumentation.

- Alles ist eine Datei, d.h. alles liegt im Datenbereich
- MFT (Master File Table) - Tabelle in der die Verweise für jede Datei und jedes Verzeichnis zu finden sind und auch die liegt als Datei im Datenbereich - auf diese wird im Boot-Sektor (Anfang) verwiesen.
- Die Einträge sind alle gleich groß. Insbesondere enthält die MFT einen Eintrag über sich selbst.
- Die einzelnen Einträge speichern Attribute und sind 1KB groß, müssen jedoch nicht ausgefüllt sein, also ist Platz für MFT-Slack.
- Die ersten 42 Byte eines Eintrags sind meist der String $FILE$. Alternativ $BAAD$ für unbenutzbare Einträge.
- Die Einträge enthalten eine Sequenznummer, also wie oft wurde ein Eintrag geändert (verwendet). Dieser Counter ist 16 Bit groß. Diese Nummer ist auch Teil der Adressierungsnummer einer Datei.
- Eintrag vermerkt die Cluster einer Datei mit der zugehörigen Länge
- Einträge können beliebig viele verschiedene Attribute haben (weshalb gelegentlich ein MFT-Eintrag nicht genügt). Da wird dann $ATTRIBUTE_LIST verwendet, um eine Liste aller Attribute der Datei und deren MFT-Adressen zu finden
- Es ist nicht spezifiziert wie häufig welche Attribute vorhanden sein müssen, so kann es also beliebig viele Dateinamen-Attribute für eine Datei geben. Es kann auch mehrere Dateiinhalte geben (lol)
- Standardattribute
  - $FILE_NAME: Dateiname und Zeitstempel
  - $STANDARD_INFORMATION: Besitzer, Zeitstempel, Zugriffsinformationen
  - $DATA: Inhalt der Datei
  - $INDEX_ROOT: Basisknoten eines Indexbaums von Verzeichniseinträgen, möglicherweise durch $INDEX_ALLOCATION erweitert
- Metadateien
  - $MFT: Eintrag für die MFT selbst
  - $MFTMirr: Eintrag für die MFT-Kopie
  - $Boot: Boot-Sektor und Boot-Code
  - $Volume: Laufwerksinformationen

  - $VOLUME_NAME und $VOLUME_INFORMATION sollten nur in $Volume stehen, sonst ist irgendetwas komisch
  - $BadClus: beschädigte Cluster-Dateien
  - $Bitmap: Belegtstatus aller Cluster

- der MFT-Modified Time-Zeitsempel ist schwieriger zu manipulieren, als andere Zeitstempel. Dieser besagt wann der zugehörige MFT-Eintrag zuletzt manipuliert wurde.
- Über die Elternverzeichnisse, die in $FILE_NAME gespeichert sind, lassen sich auch Teile des Dateisystems rekonstruieren.
- Organisation durch B-Baum, d.h. Löschen kann die Datei-Suche erschweren.

Erstellung einer Datei:

Eine Datei *file1.dat* der Größe 4000 Byte soll im bereits existierenden Verzeichnis *dir1* angelegt werden. Und zwar bei einer Clustergröße von 2048 Bytes.

1. Boot-Sektor lesen, Beginn der MFT finden (erster Eintrag $MFT), Größe eines MFT-Eintrags lesen, dort befindet sich auch die MFT-Entry-Bitmap.
2. $MFT nach unbelegtem Eintrag durchsuchen, indem das $BITMAP-Attribut der $MFT betrachtet wird
3. Anhand der $BITMAP einen unbelegten Eintrag finden und diesen Eintrag belegen und den betreffenden Eintrag in der MFT ausnullen
4. Attribute des Eintrags setzen: $STANDARD_INFORMATION, $FILE_NAME erzeugen und Zeitstempel setzen ($DATA-Feld ausfüllen)
5. Zwei neue Cluster in $DATA-Attribut von $Bitmap-Datei (MFT-Eintrag 6) finden und belegen - Also in der Cluster-Bitmap auf belegt setzen
6. Dateiinhalt in die frisch belegten Cluster schreiben
7. Indexstruktur im Wurzelverzeichnis \(MFT-Eintrag 5) absuchen nach Eintrag für *dir1*
8. Dort neuen Eintrag für *file1.dat* anlegen, Baum ggf. reorganisieren
9. Zeitstempel des Verzeichnisses aktualisieren

Löschen einer Datei:

1. Boot-Sektor lesen, MFT finden, Größe eines MFT-Eintrags auslesen
2. $MFT auslesen um das Layout des Dateisystems zu bestimmen

3. \(MFT-Eintrag 5) analysieren, Eintrag für dir1 im Index finden, Zeitstempel (Zugriff) aktualisieren
4. In dir1 den Eintrag für *file1.dat* im Index finden
5. Eintrag aus Index entfernen, Baum ggf. reorganisieren
6. MFT-Eintrag von *file1.dat* deallokieren, d.h. MFT-Eintrag in $BITMAP von $MFT und Cluster in $DATA von $Bitmap-Datei als unbelegt markieren
7. Nicht-residente Attribute von *file1.dat* deallokieren

5 Ext

Designziele sind Zuverlässigkeit und Geschwindigkeit. Das Image wird in viele kleine Bereiche (Blöcke) unterteilt, die eigenständig voneinander bestehen können. Damit wirken sich zerstörte Teile auf möglichst wenig andere negativ aus. Es sind Äquivalente zu Hardlinks möglich. Metainformationen über das Dateisystem befinden sich am Anfang im reservierten Bereich. Das Standardlayout beginnt mit einem Superblock, auf den eine Blockgruppenmenge folgt.

Typen von Zusatzfunktionen des Dateisystems:

- Compatible features - Unproblematisch, wenn das Betriebssystem dieses Feature nicht unterstützt
- Incompatible features - Wenn das Betriebssystem dieses Feature nicht unterstützt, dann sollte das Dateisystem auch nicht von diesem Betriebssystem gemountet werden
- Read only features - Wenn das Betriebssystem dieses Feature nicht unterstützt, dann sollte das Dateisystem nur Read-only gemountet werden.

Gruppendeskriptor einer Blockgruppe:

- Der Superblock wird in jedem Gruppendeskriptor einer Blockgruppe kopiert. D.h. Carven nach dem Superblock liefert sämtliche Gruppendeskriptoren
- Gruppendeskriptortabelle - verweist auf andere Blockgruppen
- Block-Belegt-Bitmap
- Blockgruppen-Inode-Belegt-Bitmap
- Inodes

Es wird versucht neue Dateien eines Verzeichnisses in die selbe Blockgruppe zu packen, um ein gewisses Maß an Lokalität hervorzurufen. Ist diese Gruppe voll, so wird nach Platz in einer anderen gesucht. Dies ist relativ leicht möglich, indem Statistiken zu jeder Gruppe geführt werden.

Löschen von Einträgen:

- In Ext3 und Ext4 wird die Dateigröße auf 0 gesetzt und die Blockverweise gelöscht, auch die in den indirekten Blöcken - d.h. zur Rekonstruktion muss man sich die Lokalität zu nutze machen
- In Ext2 werden all diese Werte nicht gelöscht, Zeitstempel werden sogar aktualisiert und die Rekonstruktion ist massiv einfacher.

Analyse:

Zur Analyse muss immer erst der Indexknoten der Datei gefunden werden. Also alle (un)belegten Indexknoten durchsuchen, den belegt Status der Inode anhand der Bitmap analysieren und die Inode auslesen. Interessant ist, dass Dateien, die unlinked werden erst dann verschwinden, wenn der Prozess, der diese Datei öffnete sich beendet. D.h. Prozesse können Dateien verstecken, indem sie diese nach dem Öffnen unlinken und weiterlaufen. Gelöschte Dateien werden sowieso nicht gelöscht, sondern nur aus der verketteten Liste des Verzeichnisses entfernt.

Gelöschte Dateinamen suchen, indem man das komplette Verzeichnis durchgeht und die Zwischenräume zwischen den Dateinamen analysiert. Gelöschte kurznamige Einträge werden länger erhalten bleiben, als lange, da die Wiederverwendung nach dem First-Fit-Verfahren erfolgt.

Im Journal lassen sich unter Umständen alte Verzeichnisstrukturen finden, da beim Aktualisieren von Indexknoten der gesamte zugehörige Block ins Journal geschrieben wird.

Erstellung einer Datei:

1. Superblock lesen, Identifiziere Beginn, Ort, Größe der Blockggruppen
2. Über diesen Supberlock in die Gruppendeskriptortabelle in die Block-Gruppe und Inode-Tabelle vom Root-Verzeichnis springen. Suche über den Indexknoten des Wurzelverzeichnisses den Indexknoten zum Verzeichnis *dir1*. Deren Inode ist in der Gruppendeskriptortabelle nachzuschlagen, um in die Block-Gruppe und Inode-Table des gesuchten Verzeichnisses zu gelangen.
3. Dort wird in der Content-Tabelle und in der Inode-Tabelle ein Eintrag erstellt. Belege Platz im Inhalt von *dir1* für den Dateinamen *file1.dat*
4. Belege einen Indexknoten für *file1.dat* in derselben Blockggruppe
5. Belege Blöcke für *file1.dat* und verknüpfe die Blöcke mit Blockverweisen im Indexknoten von *file1.dat*
6. Schreibe Daten von *file1.dat* in diese Blöcke

Löschen einer Datei:

1. Finden des Eintrags *file1.dat* durch Ablaufen der Verzeichnisse und Verzeichniseinträge
2. Durch Ändern der Next-Zeiger des vorhergehenden Verzeichniseintrags wird der Name *file1.dat* gelöscht
3. Falls Verweiszeiger im Indexknoten von *file1.dat* 0 ist (Ref-Counter), wird
   a) Inode in Inode Bitmap auf unbelegt gesetzt - also aus der Inode-Tabelle entfernt
   b) Datenblöcke werden auf unbelegt gesetzt
   c) Verweise auf Blöcke werden in EXT3/4 ausgenullt
   d) Löschzeitpunkt im Inode aktualisiert

6 Klassische Forensik

Einführungs-Foo mit viel sehr allgemein gehaltenen Begrifflichkeiten.

7 Digitale Forensik

Der Größte Unterschied zwischen einer physischen Spur und einer digitalen Spur ist wohl, dass digitale Spuren häufig eine Überführung der Daten zu den Informationen anhand einer Interpretation erfolgt, also die korrekte Interpretation von Bitmustern. Spuren scheinen prinzipiell immer mit einer Tat in Verbindung zu stehen, was mir nicht ganz eingängig ist.

- Authentizität: im Tresor gelagerte original gesicherte Festplatte
- Integrität: perfekte Kopie des authentischen Beweismittels. Also die Bedeutung hat sich nicht geändert.
- Technisch vermeidbare Spuren - Spuren, die um ihrer selbst Willen erzeugt wurden
- Technisch unvermeidbare Spuren - unweigerlich anfallende Spuren, die nicht/nur schwer zu verwischen sind und sich nicht wegkonfigurieren lassen

8 Festplattentechnik

- Cylinder Head Sector (CHS) - System zur eindeutigen Koordinatenangabe von Daten. Cylinder - die Spur, Head - welcher Kopf (also welche Scheibe), Sektor - welcher Abschnitt auf der Scheibe
- Umrechnung in LBA: Mappen auf ein zusammenhängenden Speicher. 0,0,0 - 0,0,1 - ... - 0,0,X - 0,1,0 - ... - 0,1,X - ... - 0,Y,X - 1,0,0, ...
- Am Ende einer Festplatte befindet sich die Host Protected Area (HPA), der normalerweise nicht mehr gelesen werden kann. Die Grenze wo dieser Bereich beginn lässt sich über ATA-Befehle manipulieren und dann kann auch auf den Bereich zugegriffen werden.
- in neueren Standards liegt hinter dem HPA noch der Device Configuration Overlay (DCO), um defekte Sektoren ausbessern zu können. Die defekten werden dann in die DCO verlagert und mit den ursprünglichen getauscht

SSD:

- Page - kleinste Einheit, die geschrieben werden kann 2-16 KiB
- Block - kleinste Einheit, die gelöscht werden kann 4-8 MiB - Löschen ist aufwendig und sorgt für den Verschleiß der Festplatte
- Strategien um veraltete Daten zu entfernen bzw. Speicherplatz zurückzugewinnen

Update einer Page:

Szenario: Page 3 enthält Daten, die überschrieben werden sollen. Also ein Update auf Page 3.

- Version 1:
  1. Auslesen des gesamten Blocks (alle seine Pages) in den DRAM (Buffer)
  2. Modifikation der Daten im DRAM
  3. Löschen des gesamten Blocks
  4. zurückschreiben des gesamten Blocks aus dem DRAM
- Version 2:
  1. Entnehmen des gesamten Blocks (alle seine Pages) und schreiben dieser in den DRAM
  2. Änderung der Daten im DRAM
  3. den kompletten Block ungültig markieren
  4. Schreiben des aktualisierten Inhalts des alten Blocks in einen anderen Block

Hierbei wird also verglichen mit der ersten Version kein Rücksetzzyklus verbraucht, dafür hat man einen gesamten Block als Garbage, wovon in der ersten Version am Ende keiner vorlag. Also ist das äquivalent zur Version 1, wobei das Löschen auf später verschoben wird.

- Version 3:
  1. Auslesen der Page 3 (nicht des gesamten Blocks) in den DRAM
  2. Verändern des Wertes im DRAM
  3. Das Ergebnis wird in eine freie Page geschrieben
  4. Markieren der ursprünglichen Page 3 als ungültig

  Der Garbage-Anteil ist reduziert, dafür muss nun expliziter Buch darüber geführt werden, wo sich welche Daten befinden, da sich die Page-Zuordnung innerhalb eines Blocks ändern kann.

- Version 4:
  1. Auslesen der Page 3 (nicht des gesamten Blocks) in den DRAM
  2. Verändern des Wertes im DRAM
  3. Das Ergebnis wird in eine freie Page eines beliebigen Blocks geschrieben
  4. Markieren der ursprünglichen Page 3 als ungültig

Interessant ist, dass der Slackspace im restlichen Cluster Aussage darüber treffen kann, ob die vorliegende Festplatte eine SSD ist oder nicht. Denn durch das verschieben der Page bzw der Blöcke wird in einen zwangsweise frischen Block geschrieben, wodurch der hintere Slack-Space ausgenullt ist.

Regeln für den forensischen Zugriff:

1. So tief wie nötig und so hoch wie möglich. Also nur die Daten sichern die fallrelevant sind, aber dabei keine fallrelevanten Spuren übersieht. D.h. man möchte eigentlich selektiv Daten Sichern

2. Dokumentation des Weges durch die Hierarchiestufen, also man gibt an in welcher Partition und in welchem physikalischen Block sich die entsprechende Datei befand.

3. Keine Daten verändern. Falls man doch etwas verändert, dann muss das sauber dokumentiert werden. - Dafür kann man write blocker in Hardware verwenden, die verhindern, dass Schreibzugriffe die Festplatte erreichen. Dieser Punkt ist bei SSDs aber praktisch nicht zu realisieren, da sie insbesondere bei eingeschaltetem Trim-Befehl permanent Garbage aufräumt. D.h. sobald die SSD läuft zerstört sie höchstwahrscheinlich Daten. Und auch die interne Reorganisation ist ein Problem.

Dead vs. Live Acquisition:

- *Dead Acquisition* ist die Sicherung der Daten ohne Hilfe des darauf befindlichen Betriebssystems

- *Live Acquisition* ist die Sicherung der Daten mit Hilfe des darauf befindlichen Betriebssystems - ist selten sinnvoll. Dadurch vertraut man dem verdächtigen System. Bei Gefahr im Verzug womöglich eine Anlaufstelle.

9 Partitionssysteme

Die Firmware von Festplatten liegt auf der Festplatte selbst in Zylindern negativer Nummer. Die Manipulation dieser ist möglich. Allerdings ist das überspielen mit einer vertrauenswürdigen Firmware schwierig, da man einer gewissen Basis-Code-Menge der Festplatte vertrauen muss.

Laufwerkslayout:

- Sektor 0
  - ersten 446 Byte sind Boot-Code - dient zur Analyse der Partitionstabellen etc. Wenn sich hier bösartiger Code einnistet hat man verloren. Die Ausführung bei jedem Systemstart ist garantiert
  - am Ende des Sektors liegt die Partitionstabelle, wobei jeder Eintrag 16 Byte groß ist
  - Ganz hinten befindet sich die Signatur 0xAA
- weitere Sektoren
- erste Partition
- weitere Partitionen

9.1 MBR

MBR spezifiziert ursprünglich nur 4 Partitionen, deswegen wurden primäre erweiterte Partitionen eingeführt, die selbst auch noch eine Partitionstabelle beinhalten, um auf weitere Partitionen verweisen zu können. Ein sehr großes Problem ist die mangelnde Dokumentation und Spezifikation. Man hält sich oft an das was Carrier rausgefunden hat.

Partitionstypen unter MBR:

- Primäre Partitionen
  - Primäre Partition - Primäre Dateiystempartition (primary file system partition)
    - Partition im MBR mit Dateisystem (normale Partition)
  - Primäre erweiterbare Partition (primary extended partition)
    - Partition im MBR, an deren Beginn eine weitere Partitionstabelle steht
    - Erweiterte Partition, um mehr als vier DOS-Partitionen verwalten zu können
    - Die Idee ist, dass nicht alle Einträge in der MBR normale Partitionseinträge sind, sondern eine erweiterte Partition existiert, in der sich selbst weitere Partitionen befinden.

    - befindet sich am Ende der Platte

- Sekundäre Partitionen
  - Sekundäre Dateisystempartition (secondary file system partition)
    - Partition innerhalb einer primären erweiterten Partition mit Dateisystem
    - unter Windows logische Partitionen
    - Aufbau ist wie bei primären Dateisystempartitionen, befinden sich allerdings innerhalb einer weiteren Partition
  - Sekundäre erweiterte (secondary extended partition)
    - Partition mit einer Partitionstabelle und einer sekundären Dateisystempartition
    - Dient als Wrapper für sekundäre Dateisystempartitionen

- Festplatte mit 12 GB und 6 Partitionen à 2 GB

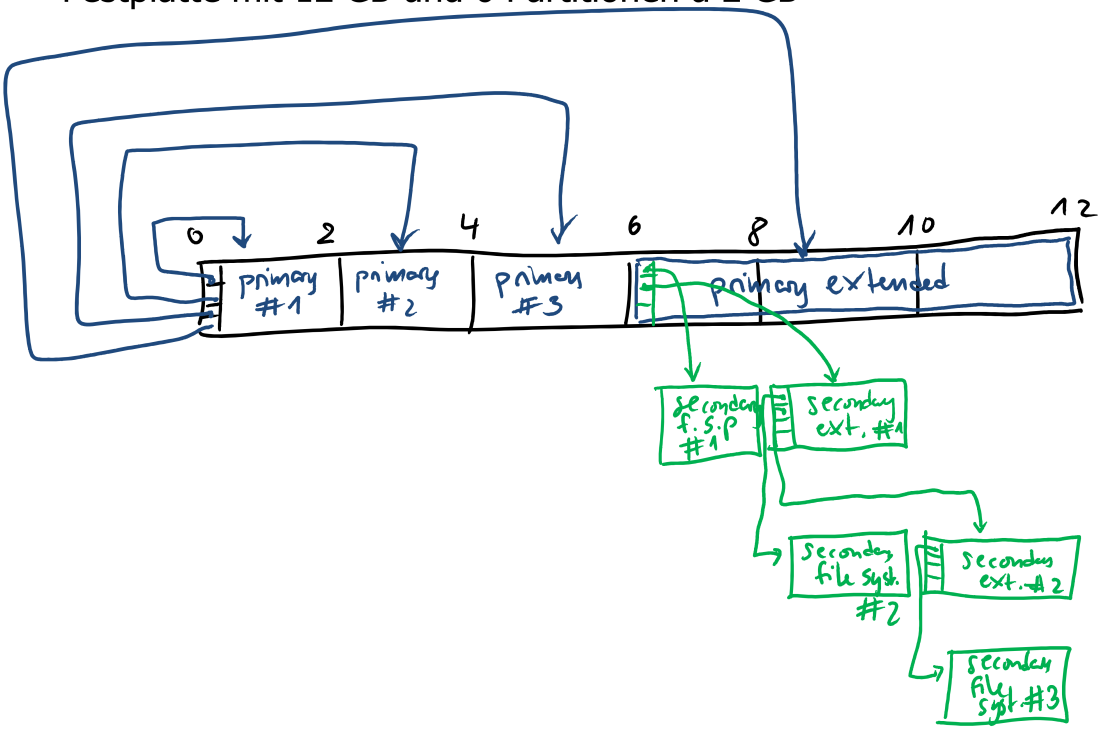


Abbildung 3: MBR Hack für größere Festplatten

Kann in primary extended und in secondary ext nur jeweils ein sec. file system partition verlinkt werden? Oder warum braucht man so viele, ich dachte man hat 4 Einträge?

9.2 GPT

- der Magic String zum Carven lautet *EFI PART*
- 64-Bit-LBA
- redundanz der protective MBR und anderer wichtiger Daten
- Aufbau
  - Sektor 0 - Protective MBR - Pseudo MBR, enthält genau einen Eintrag
  - Sektor 1 - GPT Header - Größe + Lage der Partitionstabelle, Enthält CRC-Checksum
  - partition Tabelle
  - Partitionen
  - Backup-Area
    - Backup Partition Table
    - Backup GPT Header

10 Recht

Ermächtigungsgrundlage:

Ist notwendig um Beweismittel zu sichern. Dies hat von einem unabhängigen Richter geprüft zu werden. Diese wird untergliedert in folgende teilweise vermischende Punkte. Der Grund für eine Vermengung ist, dass teilweise das Vorbereiten von Straftaten bereits eine Straftat sein kann.

- Strafverfolgung (Repression) - konkrete Straftat als Anlass
  - TKÜ - 100a, 100b StPO - Eingriff in das Fernmeldegeheimnis, nur mit richterlicher Genehmigung, auf maximal drei Monate befristet (kann mit Begründung verlängert werden)
  - Durchsuchung - 102-110 StPO - richterliche Genehmigung - ist örtlich begrenzt, es muss also genau angegeben werden wo durchsucht wird - Der Berschuldigte darf der Durchsuchung beiwohnen
  - Großer Lauschangriff 100c, 100d StPO - verwanzen der Wohnung - richterliche Genehmigung - stärkere Beschränkungen als bei der TKÜ
- Gefahrenabwehr (Prävention) - konkrete Gefahr

TKÜ-Paragraph - 100a StPO:

(1) Überwachung der Betroffenen, ohne dessen Wissen möglich, wenn
  a) jemand verdächtigt wird eine in Absatz 2 versuchte Straftat beging oder dies bei gegebener Strafbarkeit versuchte
  b) die Tat im Einzelfall schwer wiegt
  c) die Überprüfung des Sachverhalts oder die Aufenthaltsermittlung des Verdächtigen wäre auf eine andere Weise wesentlich schwerer oder aussichtslos

(2) Schwere Straftaten im Sinne des Absatzes 1 Nr. 1 sind. - Da scheint jede Menge drin zu stehen und das ändert sich recht häufig

Durchsuchung-Paragraph - 102 StPO:

- Verdächtiger als Täter oder Teilnehmer einer Straftat, wenn dadurch der Verdächte ergriffen werden kann oder dadurch wahrscheinlich Beweismittel aufzufinden sind.
- Eingeschränkt durch 105 StPO:
  - Anordnung durch einen Richter und bei Gefahr im Verzug auch durch die Staatsanwaltschaft
  - Durchsuchungsbeschluss muss enthalten:
    * Bezeichnung der Straftat

    - Zweck, Ziel, Ausmaß der Durchsuchung
    - Angaben über die Form der gesuchten Beweismittel

- Weiterhin darf eine Sicherstellung zur Durchsicht erfolgen. Also Gegenstände oder elektronische Speichermedien werden mitgenommen, und durchsucht, um festzustellen welche man denn nun tatsächlich sicherstellt.
- Daten die sich im Ausland befinden sind nicht uneingeschränkt abgegriffen werden. Also Daten auf im Ausland liegender Server. Praktisch wird das gemacht, aber dagegen kann man klagen.

Gefahrenabwehr am Beispiel BKAG:

- 20l BKAG - Quellen-TKÜ. Spähprogramm auf dem Endgerät ist erlaubt, um Daten mitzuschneiden
- 20l BKAG - Quellen-TKÜ. Spähprogramm auf dem Endgerät ist erlaubt, um Daten zu erheben
- Dies ist erlaubt zur Abwehr einer dringenden Gefahr für den Bestand oder die Sicherheit des Staates oder für Leib, Leben oder Freiheit einer Person oder Sachen von bedeutendem Wert, deren Erhaltung im öffentlichen Interesse liegt und wenn kein milderes Mittel existiert.

Es folgen einige Begrifflichkeiten, wie Zusatzfunde etc. die ich nicht mit in diese Zusammenfassung aufnehme.

11 Theorie

System zur Bestimmung der Zustände der Vergangenheit:

Hierfür wird kein Graph verwendet, da dieser bei Nichtdeterminismus unpraktisch ist. Stattdessen wird ein programmierbares Systemmodell verwendet, aus dem sich Graphen konstruieren lassen.

- Besteht aus $S = (V, \Sigma, q_0)$, also einer Variablenmenge, einer Aktionsmenge und einem Initialzustand, wobei ein Zustand $\sigma = (g, c)$ aus einem Wächter (g) (boolean) und einem Befehl (c) besteht.
- Die Befehle werden durch einen Wächter bewacht. Es wird betrachtet welche Wächter sind wahr. Ist dies der Fall, so ist der Befehl scharf geschaltet. Von all diesen scharf geschalteten Befehlen wählt der Scheduler einen nichtdeterministisch aus und kehrt zum Anfangsprogramm zurück.
- Beispiel auf Seite 12.

Spurenmenge - evidence (E):

Ist die Menge aller Teilmengen von Zuweisungen, die von der Aktion ausgeführt werden. Beispiel Seite 25

$$E(\sigma) = \mathcal{P}(\{[v = d] | [v = d] \in \sigma\})$$

Kombinierte Spurenmengen - merged evidence (ME):

Vereinigung der Spurenmengen aller Aktionen in der Menge

$$ME(\Sigma) = \bigcup_{\sigma \in \Sigma} \bigcup_{e \in E(\sigma)} e$$

Charakteristische Spuren - characteristic evidence (CE):

Wenn eine solche Spur gefunden wird, dann weiß man eine bestimmte Aktion hat stattgefunden, sonst gäbe es diese Spur nicht. Es sind also alle Spuren, die nicht durch andere Aktionen erzeugt werden können oder im Initialzustand - zero evidence (ZE) enthalten sind. Beispiel Folie 31

$$CE(\sigma, \Sigma') = E(\sigma) \setminus (\mathcal{P}(ME(\Sigma') \cup ZE))$$

Kontraspuren - counter evidence (XE):

Sind Spuren die darauf hindeuten, dass eine bestimmte Aktion nicht stattgefunden hat. Auch hier gibt es weitere Typen: charakteristische Kontraspuren (Aktion hat sicher nicht stattgefunden), gemeinsame charakteristische Kontraspuren - Beispiel Folie 37. Nicht-charakteristische Kontraspuren besagen nur, dass die betrachtete Aktion nicht die letzte war. Lediglich die charakteristische Kontraspur verrät, wenn eine Aktion überhaupt nicht ausgeführt wurde.

$$XE(\sigma) = \mathcal{P}\left(\bigcup_{[v=d]|\exists d' \neq d:[v=d'] \in \sigma} \{[v=d]\}\right)$$

charakteristische Kontraspuren - characteristic counter evidence (CXE) - Beispiel 41

$$CXE(\sigma, \Sigma') = XE(\sigma) \setminus \mathcal{P}(ME(\Sigma'))$$

Folgende Folien zeigen, dass es immer Daten gibt, die von nur einer Aktion angefasst werden. Halte ich für kaum praxisrelevant.

12 Vorgehensmodelle

Alles dokumentieren und falls es wichtig ist auch händisch in gebundenen Büchern. Was man also für das Tool script macht ist die Aufzeichnung ganz normal zu starten und die Hashsummen der entsprechenden Dateien mit Datum versehen händisch zu notieren. Dadurch lässt sich Manipulation ausschließen.

13 Abbildungsverzeichnis

Abbildungsverzeichnis

Liste der noch zu erledigenden Punkte