Data-Recovery und forensische Methoden zur Wiederherstellung von Daten[1]

Timo Minartz

Seminar Speichermedien WS 2008/2009

13. Januar 2009

[1] betreut von Olga Mordvinova, Julian Kunkel

Inhalt

Motivation

- Wie kann man versehentlich gelöschte Daten wieder herstellen?
- Wie können Daten von einer defekten CD/Festplatte gesichert werden?
- Welche Daten können von einer USB-Disk gesichert werden?
- Wie können forensische Daten gesichert werden?
- Welche Tools können genutzt werden?
- Welche Chance auf Erfolg kann garantiert werden?
- Wie können sensible Daten geschützt werden?
- Wie können Daten sicher gelöscht werden?

Einleitung / Relevanz

Datensicherheit

- gebrauchte Festplatten werden weiterverkauft
- Studien zeigen ca. 60% nicht ausreichend gelöscht
- z.B. militärische Zugangscodes wurden versteigert [Onl]

Hardwarefehler

- Hardware ist generell langlebig
- aber: Fehler möglich
  - Materialermüdung
  - falsche Lagerung
  - Disk fallenlassen, CD zerkratzen, ...

Datensicherheit
- gebrauchte Festplatten werden weiterverkauft
- Studien zeigen ca. 60% nicht ausreichend gelöscht
- z.B. militärische Zugangscodes wurden versteigert [Onl]

Hardwarefehler
- Hardware ist generell langlebig
- aber: Fehler möglich
  - Materialermüdung
  - falsche Lagerung
  - Disk fallenlassen, CD zerkratzen, ...

Datensicherheit:

- einfaches Formatieren nicht ausreichend, um Daten zu löschen
- oft Datenträger noch nicht einmal formatiert

Unterschied Recovery - forensische Methoden

Recovery

- Wiederherstellung gelöschter Daten
- Aufgrund von Defekten oder Benutzerfehlern

Forensik

- Wiederherstellung gelöschter Dateien
- Untersuchung von Benutzerverhalten
- Durchsuchen nach versteckten Dateien

Folgerung

Recovery ist Speziallfall von Forensik, da mehr Wissen über wiederherzustellende Dateien vorhanden

Unterschied Recovery - forensische Methoden

Recovery
- Wiederherstellung gelöschter Daten
- Aufgrund von Defekten oder Benutzerfehlern

Forensik
- Wiederherstellung gelöschter Dateien
- Untersuchung von Benutzerverhalten
- Durchsuchen nach versteckten Dateien

Folgerung
Recovery ist Speziallfall von Forensik, da mehr Wissen über wiederherzustellende Dateien vorhanden

Die forensische Informatik beschäftigt sich mit der Rekonstruktion von Daten mit dem Ziel, Aussagen über Benutzerverhalten zu finden. Um auch vor beispielsweise einem Gericht eine Beweiskette vorlegen zu können ist es unabdingbar, dass der Verlauf der Untersuchung dokumentiert wird und die Datenintegrität nicht gefährdet wird. Generell wird versucht, vom Benutzer gelöschte oder versteckte Dateien zu finden, die Aufschluss über die konkrete Tat geben (z.B. Bilder auf einem Handy, Chatprotokolle auf der Festplatte, ...).

Recovery ist somit ein vereinfachter Prozess mit forensischen Methoden, da keine Beweiskette aufrechterhalten werden muss. Weiterhin sind in der Regel mehr Informationen darüber vorhanden, was wann von wo gelöscht wurde (Dateiname, Dateityp, Löschzeitpunkt, ...).

Defekte (1)

Hardwaredefekt

- Ursache: Head-Crash, Kratzer, Verschmutzung, ...
- viele spezialisierte Firmen zur Datenrettung
- von eigenem Öffnen von Festplatten ist aufgrund von Staub, Unwissen etc. abzuraten
- bei CD's / DVD's können Hausmittel durchaus helfen

Hardwaredefekt

- Ursache: Head-Crash, Kratzer, Verschmutzung, ...
- viele spezialisierte Firmen zur Datenrettung
- von eigenem Öffnen von Festplatten ist aufgrund von Staub, Unwissen etc. abzuraten
- bei CD's / DVD's können Hausmittel durchaus helfen

- Kosten für Datenrettung: CD/DVD's für einige Euros
- Oft erfolgsabhängige Bezahlung für Recovery bestimmter Dateien (z.B. Abschlussarbeit)

Defekte (2)

Logischer Defekt

- Ursache: Daten versehentlich gelöscht, Folge von Hardwaredefekt
- Image von Datenträger erstellen
  - um Überschreiben und weitere Beschädigung der Daten zu verhindern
  - z.B. mit *dd* oder *ddrescue*, Auslesen von (defekten) Byteblöcken ohne Interpretation möglich
  - es existieren Hardwaretools für Festplatten, um Schreibzugriffe zu unterdrücken

Logischer Defekt

- Ursache: Daten versehentlich gelöscht, Folge von Hardwaredefekt
- Image von Datenträger erstellen
  - um Überschreiben und weitere Beschädigung der Daten zu verhindern
  - z.B. mit *dd* oder *ddrescue*, Auslesen von (defekten) Byteblöcken ohne Interpretation möglich
  - es existieren Hardwaretools für Festplatten, um Schreibzugriffe zu unterdrücken

- *dd* für Flash-Speicher erstmal nicht brauchbar, da Blöcke durch Mapping des Flash-Controllers in der Regel nicht zusammenhängend ausgelesen werden können
- Schreibzugriffe zu unterdrücken ist wichtig, um die Beweiskette aufrecht zu erhalten (kein Modifizieren der Daten beim Erstellen des Abbildes). Für die Recovery ist dies weniger relevant, hier reicht in der Regel das Benutzen eines 2. Betriebsystem (Live-Distribution, z.B. Ubuntu, Knoppix, BartPE, ...).

Recovery Techniken

einfacher Fall

- Backup nutzen

Normalfall

- Konsistenzüberprüfung (consistency checking)
- Sezieren der Daten (data carving)

Alternative zum “normalen” Backup: *continuous data protection*. Um zu verhindern das falsche Daten vom Backup erfasst und überschrieben werden (z.B. bei Virenbefall), werden die einzelnen Backupinformationen wie in einer Versionsverwaltungsoftware (SVN, CVS, . . .) abgespeichert, um verschiedene Versionen der Dateien zu erhalten. Dieses Konzept ist in Microsofts Windows Vista bereits als “Schattenkopie” integriert.

Consistency checking

Vorausetzungen

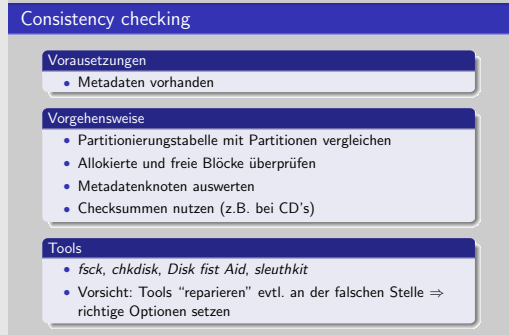
- Metadaten vorhanden

Vorgehensweise

- Partitionierungstabelle mit Partitionen vergleichen
- Allokierte und freie Blöcke überprüfen
- Metadatenknoten auswerten
- Checksummen nutzen (z.B. bei CD's)

Tools

- *fsck*, *chkdisk*, *Disk fist Aid*, *sleuthkit*
- Vorsicht: Tools "reparieren" evtl. an der falschen Stelle $\Rightarrow$ richtige Optionen setzen



Suche in Metadaten nach Informationen wie

- Dateiname
- Änderungszeitpunkt, ...

Vorgestellte Tools zur Konsistenzüberprüfung sind eigentlich Tools zur Konsistenzsicherung, d.h. die Programme versuchen das Dateisystem wieder in einen konsistenten Zustand zu überführen. Dies kann so geschehen, dass die beschriebenen Blöcke, die allokiert aber nicht referenziert sind, entgültig deallokiert werden. Zweck der Recovery ist es aber, die Allokierung aufrecht zu erhalten und Referenzierung wieder herzustellen. Da beide Verfahren in einen konsistenten Zustand führen ist zu überprüfen, welches Verfahren vom jeweiligen Tool genutzt wird (in der Regel ist es das erste Verfahren, da einfacher zu realisieren. Also nur nutzen, um die inkonsistente Stelle zu finden, nicht zur Reparatur).

Data carving

Vorausetzungen

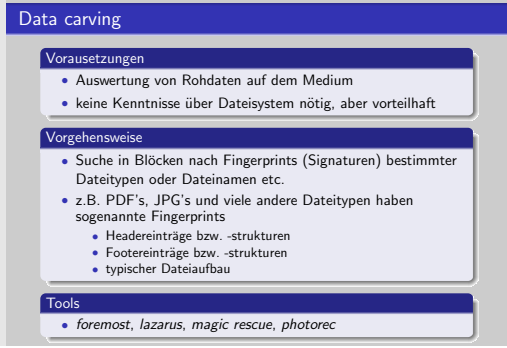
- Auswertung von Rohdaten auf dem Medium
- keine Kenntnisse über Dateisystem nötig, aber vorteilhaft

Vorgehensweise

- Suche in Blöcken nach Fingerprints (Signaturen) bestimmter Dateitypen oder Dateinamen etc.
- z.B. PDF's, JPG's und viele andere Dateitypen haben sogenannte Fingerprints
  - Headereinträge bzw. -strukturen
  - Footereinträge bzw. -strukturen
  - typischer Dateiaufbau

Tools

- *foremost*, *lazarus*, *magic rescue*, *photorec*



eine gute Dokumentation der Tools und Vorgehensweise zur Nutzung findet sich unter [com09]. Auch hier ist es durchaus sinnvoll, mehrere Tools zu probieren, da mit unterschiedlichen Methoden gearbeitet wird um Daten wiederherzustellen. Generell ist je nach Recoveryfall Motivation gefragt, bis die Daten wieder hergestellt sind.

Hardwaredefekt

Lesefehler durch Kratzer / Verschmutzung

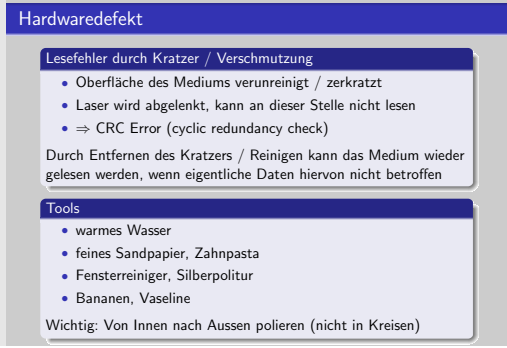
- Oberfläche des Mediums verunreinigt / zerkratzt
- Laser wird abgelenkt, kann an dieser Stelle nicht lesen
- $\Rightarrow$ CRC Error (cyclic redundancy check)

Durch Entfernen des Kratzers / Reinigen kann das Medium wieder gelesen werden, wenn eigentliche Daten hiervon nicht betroffen

Tools

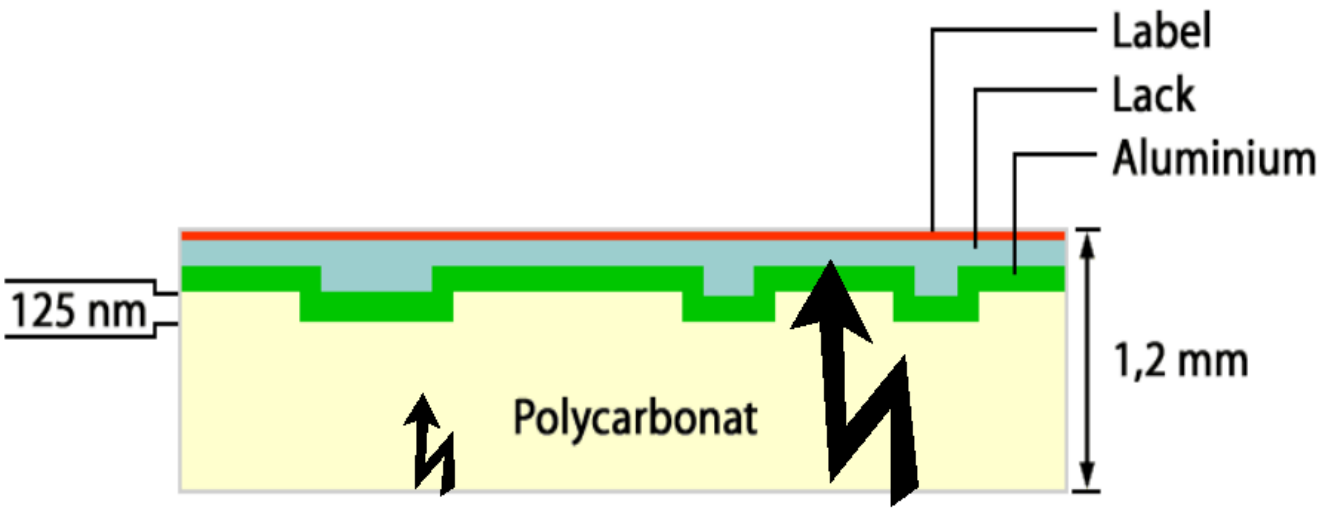
- warmes Wasser
- feines Sandpapier, Zahnpasta
- Fensterreiniger, Silberpolitur
- Bananen, Vaseline

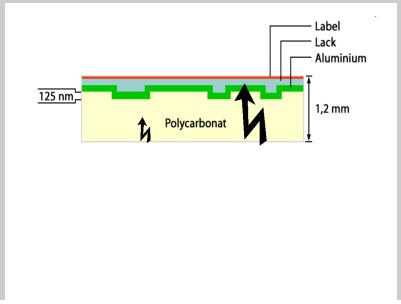
Wichtig: Von Innen nach Aussen polieren (nicht in Kreisen)



Kombination der Tools zur Optimierung: Zahnpasta für tiefe Risse, feines Sandpapier für feinere Risse. In einigen Foren wurde die Benutzung von Alkohol zur Reinigung propagiert, allerdings sollte Alkohol aufgrund seiner aggressiven Wirkung nicht zur Reinigung eingesetzt werden, da eine weitere Beschädigung der Disk hierdurch nicht auszuschliessen ist. Generell sollte der spezielle Fall entschieden werden, ob sich eine eigene Reperatur lohnt. Handelt es sich hierbei um wichtige Daten, die nur auf diesem Medium vorhanden sind, ist eine professionelle Recovery (da kostengünstig) wahrscheinlich der bessere Weg. Weiterhin werden Medien von vielen Herstellern kostenlos erneuert (z.B. Softwaredatenträger), was auch vor der Reparatur zu erwägen ist.

Hardwaredefekt (2)





- Kratzer kann nicht repariert werden, wenn Pits beschädigt
- Bruchstellen können in der Regel nicht repariert werden
- Problem bei DVD's: 2 Seiten können zerkratzt werden
- Professionelle Reparatur
  - Abschleifen der CD mit speziellen feingranularen Abschleifgeräten
  - Gerätkosten zwischen 300 und 5000 Dollar
  - oft bei CD-Läden durchführbar

Fehlererkennung und Fehlerkorrektur

CIRC

- cross-interleaved Reed-Solomon code
- zwei verschiedene Reed-Solomon-Code Konfigurationen verschachtelt
- Fehlererkennung und Fehlerkorrektur möglich
- nicht-lesbare Sektoren können wiederhergestellt werden
- 2 Byteblöcke pro 32 Byteblöcke wiederherstellbar

Funktionsweise Reed-Solomon-Code

- Modellierung der Daten als Koeffizienten für ein Polynom
- Division des Polynoms durch einen festen Term
- Speicherung des Divisionsergebnisses als redundante Information

CIRC

- cross-interleaved Reed-Solomon code
- zwei verschiedene Reed-Solomon-Code Konfigurationen verschachtelt
- Fehlererkennung und Fehlerkorrektur möglich
- nicht-lesbare Sektoren können wiederhergestellt werden
- 2 Byteblöcke pro 32 Byteblöcke wiederherstellbar

Funktionsweise Reed-Solomon-Code

- Modellierung der Daten als Koeffizienten für ein Polynom
- Division des Polynoms durch einen festen Term
- Speicherung des Divisionsergebnisses als redundante Information

Verschachtelung = Interleaving. Mehr Informationen unter [Var09]

Softwarelösungen

Funktionsweise

- Fehlerkorrektur der Hardware unterschiedlich ausgeprägt
- Lesen der defekten Blöcke und Korrektur “per Hand”
- Interpolation der defekten Blöcke falls nicht mehr rekonstruierbar (idR. nur bei Audio und Video)

Tools

- Testen mit mehreren Tools/Laufwerken sinnvoll
- *Recover Disk*,*CD Recovery Toolbox*, . . .

Funktionsweise
- Fehlerkorrektur der Hardware unterschiedlich ausgeprägt
- Lesen der defekten Blöcke und Korrektur "per Hand"
- Interpolation der defekten Blöcke falls nicht mehr rekonstruierbar (idR. nur bei Audio und Video)

Tools
- Testen mit mehreren Tools/Laufwerken sinnvoll
- *Recover Disk,CD Recovery Toolbox, ...*

Zur Funktionsweise der Softwarelösungen konnten leider keine Quellen gefunden werden, insofern handelt es sich hier um eine Annahme zur Funktionsweise. Letztendlich heisst das, dass die Daten von der CD auf einem möglichst niedrigem Level ausgelesen werden (z.B. mit *ddrescue*) und dann mit den bereits bekannten Methoden zur Recovery weiter verarbeitet werden können. Hierfür lässt sich dann der Reed-Solomon Code und weitere Informationen wie der Dateiname etc. nutzen.

Löschen von Daten auf der CD-RW

schnelles Löschen

- Nur Inhaltsverzeichnis (TOC) gelöscht
- Daten sind potentiell wiederherstellbar

normales Löschen

- komplette CD gelöscht
- Daten nicht wiederherstellbar

Unter Vorbehalt, keine Quellen gefunden die Verfahren aufzeigen, um schnell-gelöschte CD-RW Medien wieder herzustellen. Jedoch sind die Daten definitiv noch auf dem Medium vorhanden, d.h. wahrscheinlich durch irgendwelche Verfahren auch wieder herstellbar. Für das normale Löschen wird allgemein angenommen, dass die Daten nicht wieder ausgelesen werden können. Allerdings sind auch hierfür keine Quellen mit eindeutiger Aussage gefunden worden.
Zum sicheren Löschen von CD's bzw. CD-RW's wird vom NIST (National Institute of Standards and Technology) empfohlen:

- Pulverisieren oder
- Shreddern

Zusammenfassung Compact Disk

- Daten können von defekten CD's wieder hergestellt werden
- gute eingebaute Fehlererkennung und Korrektur
- Hausmittel haben gute Erfolgswahrscheinlichkeiten
- professionelle Datenrettung erschwinglich
- zum sicheren Löschen CD shreddern oder verbrennen
- Prävention: Backup, bessere Rohlinge

Hardwaredefekt

Beschädigungen

- Gehäuse (insbesondere Stecker) beschädigt
- Disk beschädigt
- Controller beschädigt

Lösungen

Bauteiletausch: defekten Controller austauschen, Gehäuse austauschen, ...

Beschädigungen
- Gehäuse (insbesondere Stecker) beschädigt
- Disk beschädigt
- Controller beschädigt

Lösungen
Bauteiletausch: defekten Controller austauschen, Gehäuse austauschen, ...

Die vorgestellten Lösungen sind nicht immer empfehlenswert, da sie potentiell nicht einfach durchzuführen sind. Je nach Wichtigkeit der Daten auf dem Medium ist eine professionelle Recovery die bessere Wahl, da sonst die Gefahr besteht, durch den Reparaturversuch die Disk weiter zu beschädigen (z.B. durch Staub im Gehäuse oder Unwissen).

Logische Defekte

Datenfehler

- Boot Sektor defekt ⇒ Boot Sektor neu schreiben
- Betriebsystem defekt ⇒ Recovery Funktion des Betriebssystems oder Herstellers, Neuinstallation
- Virenbefall ⇒ Integrität der Daten nicht mehr gewährleistet
- unabsichtliches Löschen Dateien / Ordnern
- unabsichtliches Löschen von Partitionen
  - reversibel durch Recoverytechniken
  - Tools: parted, sleuthkit, photorec, testdisk

Im Folgenden

Am Beispiel von Dateisystem *ext3*, für andere Dateisysteme ähnliche Vorgehensweise

Logische Defekte

Datenfehler

- Boot Sektor defekt ⇒ Boot Sektor neu schreiben
- Betriebsystem defekt ⇒ Recovery Funktion des Betriebssystems oder Herstellers, Neuinstallation
- Virenbefall ⇒ Integrität der Daten nicht mehr gewährleistet
- unabsichtliches Löschen Dateien / Ordnern
- unabsichtliches Löschen von Partitionen
  - reversibel durch Recoverytechniken
  - Tools: parted, sleuthkit, photorec, testdisk

Im Folgenden

Am Beispiel von Dateisystem *ext3*, für andere Dateisysteme ähnliche Vorgehensweise

Auch bei denn logischen Defekten ist eine Vorabeinschätzung des Nutzen des Aufwandes für die Recovery zu treffen. So ist bei gängigen Betriebssystemen schnell der Bootsektor neu geschrieben und konfiguriert, Recoverymassnahmen sind in der Regel überflüssig. Bei einem defekten Betriebssystem lässt sich meist nicht absehen, an welchen Stellen / Dateien der Fehler auftritt. Wenn also Nutzer- und Systemdaten getrennt sind, bzw. ein Backup der Nutzerdaten erstellt wurde, ist es meist sinnvoller, eine neues, sauberes System zu installieren. Das gleiche gilt bei Virenbefall, da die Integrität der Dateien nicht mehr gewährleistet ist. Um eine erneute Infektion auszuschliessen, sollte die Wichtigkeit der Daten gegen eine Neuinstallation abgewägt werden.

Dateisystem *ext3*

Generell

- Nachfolger von *ext2*
- Blockbasiertes Dateisystem
- unterstützt Journaling

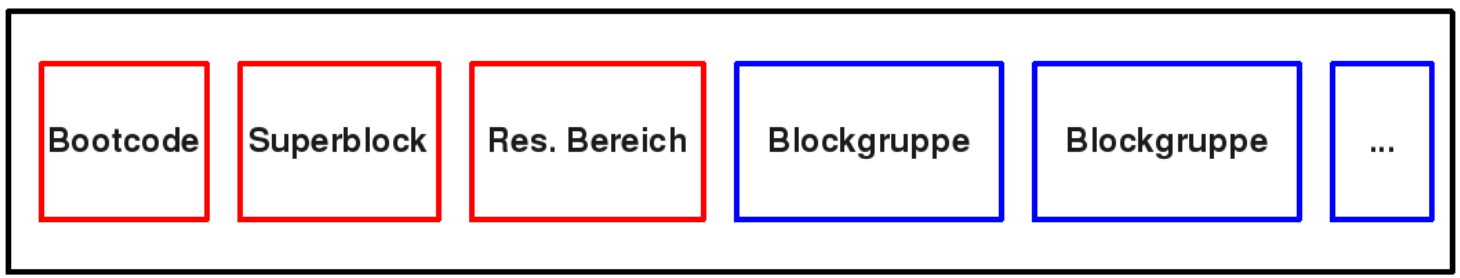
Journaling

- Konsistenzsicherung
- jedes Update als Transaktion
- gesamter Block wird ins Journal geschrieben

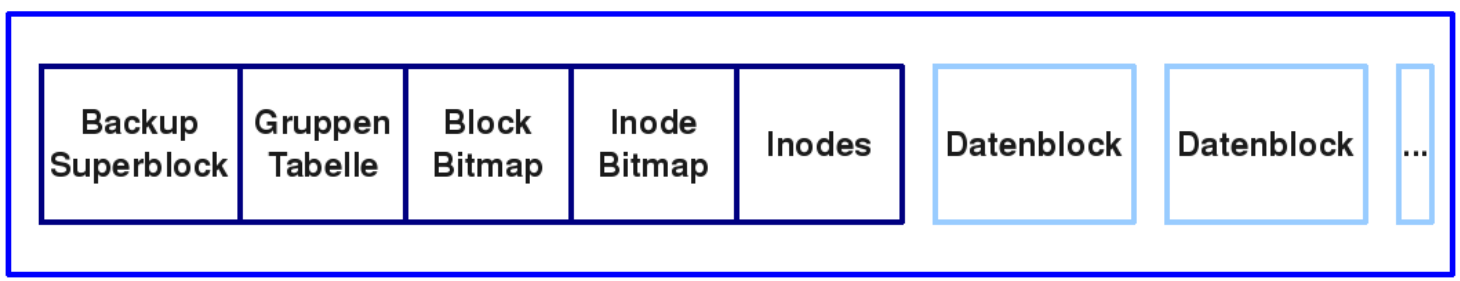
$\Rightarrow$ Bei Abbruch des Änderungsvorgangs kann die inkonsistente Stelle gefunden werden

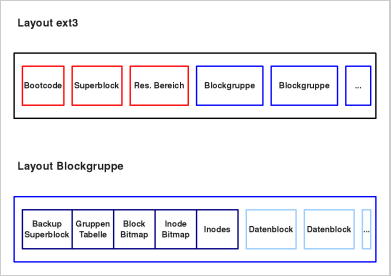
Aufbau ext3

Layout ext3



Layout Blockgruppe



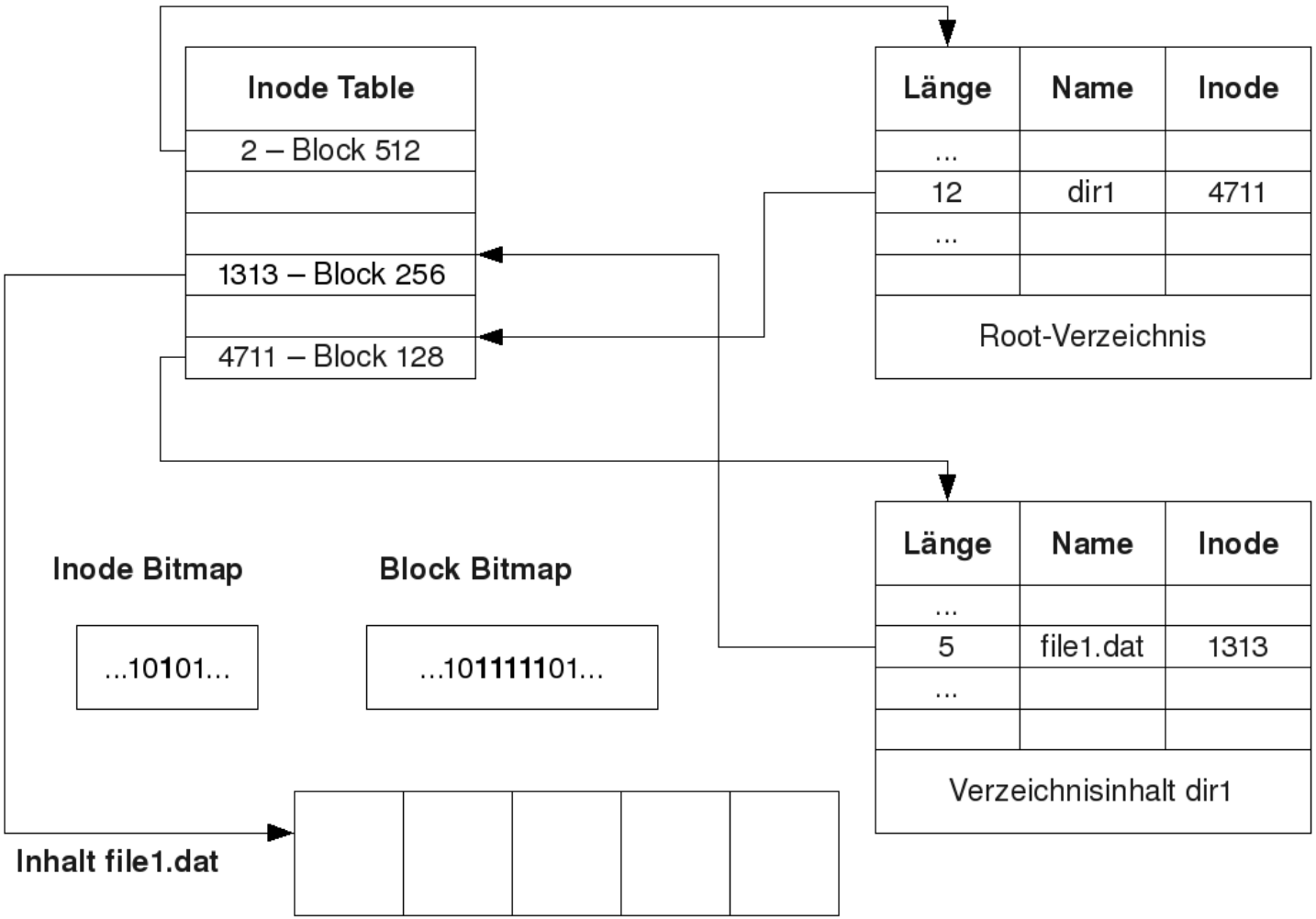


- Superblock: Konfigurationsdaten (Anzahl Blöcke pro Blockgruppe, Gesamtzahlblöcke, Größe reservierter Bereich)
- Blockgruppe: aufgeteilt in Gruppendeskriptor und Datenblöcke
- Gruppentabelle: Verweise auf alle anderen Blockgruppen
- Block Bitmap: Belegtstatus der Datenblöcke
- Inode Bitmap: Belegtstatus der Inodeblöcke
- Inodes: Blöcke mit Metainformationen zu den Datenblöcken (Verzeichniseinträge, Dateinamen, etc.)

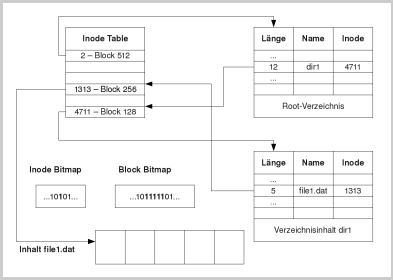
Möglichkeiten um Daten zu verstecken:

- Bootcode-Block (wird meist von Betriebssystemen nicht mehr genutzt)
- Res. Vereich, Superblock und seine Kopien
- ungenutze Datenblöcke $\Rightarrow$ Gefahr das Daten überschrieben werden
- Block Bitmap editieren $\Rightarrow$ Daten werden nicht überschrieben und sind nicht in Verzeichnissstruktur eingebunden, somit nicht mit einfachen Mitteln zu finden, aber durch Methoden des *Data Carving*

Erstellen einer Datei



Erstellen einer Datei



Die Datei */dir1/file1.dat* soll erstellt werden.
Im Inode Table wird zu Zuordnung Inode - Block des Inodes gespeichert.
Um die Datei anzulegen werden die Verzeichniseinträge des Root-Verzeichnisses nach dem Ordner *dir1* durchsucht. Dieser Eintrag beinhaltet die Nummer des Inodes, der die Metainformationen für dieses Verzeichnis verwaltet (4711). In den Verzeichniseinträgen des Inodes 4711 wird dann die Datei hinzugefügt und ein neuer Inode zur Verwaltung der Dateimetadaten der neuen Datei erstellt (1313). Dieser Inode enthält dann einen Verweis auf die Datenblöcke, die angelegt werden. In den Bitmaps werden die zugehörigen Einträge der Datenblöcke und des Inodes auf 1 gesetzt.

Löschen einer Datei

Verzeichniseinträge

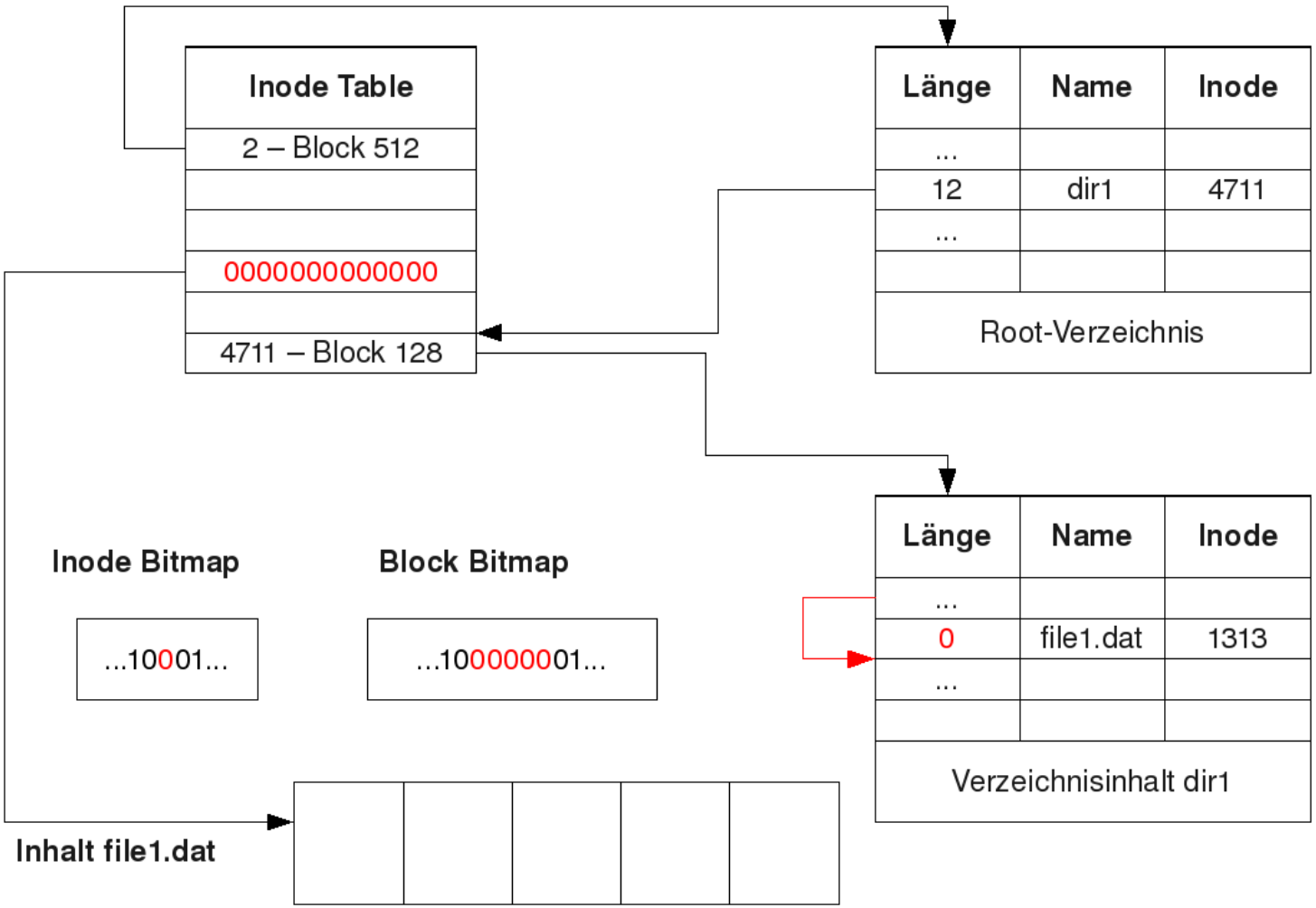
- liegen in Datenblöcken von Verzeichnissen
- bilden eine Liste (ext2) oder einen Baum (ext3)
- enthalten Dateiname und Verweis auf Indexknoten mit Daten

Löschen

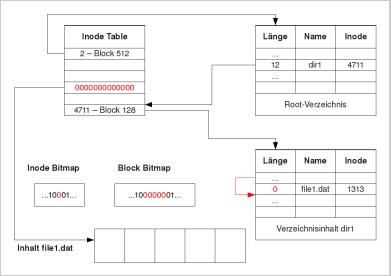
- Dateigröße wird auf 0 gesetzt und alle Blockverweise gelöscht
- Zeiger in linearer Liste bzw. Baum der Verzeichniseinträge wird einfach weitergeschaltet
- kompletter Datenblock bleibt erhalten

$\Rightarrow$ Dateiname und Inodenummer noch vorhanden

Löschen einer Datei (2)



Löschen einer Datei (2)



Der Verweis auf den Inode 1313 wird aus dem Inode Table gelöscht (genullt), der Zeiger in deem Baum der Verzeichniseinträge für das Verzeichnis */dir* wird einfach weitergeschaltet. In den Bitmaps werden die zugehörigen Einträge für die Datenblöcke des Dateiinhalts und des Inodes 1313 auf 0 gesetzt. Also weiterhin vorhanden (aber dereferenziert) sind:

- der Inode 1313 mit den Metadaten zur Datei *file1.dat*
- die Datenblöcke zur Datei
- der Verzeichniseintrag */dir1/file1.dat*

Löschen einer Datei (3)

Zustand

- Inode in Inode Bitmap als unbelegt markiert
- Datenblöcke in Block Bitmap als unbelegt markiert
- Verweise auf Blöcke genullt
- Löschzeitpunkt im Inode aktualisiert

Zitat von Andreas Dilger, Entwickler von *ext3*

How can I recover (undelete) deleted files from my ext3 partition?

"In order to ensure that ext3 can safely resume an unlink after a crash, it actually zeros out the block pointers in the inode, whereas ext2 just marks these blocks as unused in the block bitmaps and marks the inode as deleted and leaves the block pointers alone."

Wiederherstellen

Suche nach gelöschtem Dateinamen

- gehe komplettes Verzeichnis durch
- Betrachtung der Zwischenräume zwischen den Einträgen
- ein kurzer Dateiname ist eher vorhanden als ein langer Dateiname

Suche nach Blockverweisen

- Inodes mit korrekten Verweisen können noch im Journal stehen
- Vorgehen evtl. wiederholen
- Suche nach Inodes mit Löschzeitpunkt

Tool

- *ext3grep*

ext3grep: Mischansatz Data Carving und Consistency Checking

Das generelle Vorgehen ist abhängig vom konkreten Dateiystem und dessen Belegungsstrategie.

Wiederherstellen (2)

Problem

- Blöcke bereits wieder überschrieben
- Journal wieder überschrieben

weitere Möglichkeiten

- aufgrund der Belegungstrategie liegen die Blöcke einer Datei meist in einer Blockgruppe
- evtl. weitere Blöcke in der Blockgruppe als gelöscht markiert

Daten sicher gelöscht durch Überschreiben?

Zusammenfassung normales Löschen

- eigentliche Daten nicht gelöscht
- nur Verweise auf Daten gelöscht

Überschreiben

praktizierter Ansatz: Blöcke in mehreren Durchläufen mit Zufallsbits überschreiben

nach Gutmann [Gut96]

Durch Ansatz “scanning transmission electron microscopy” überschriebene Daten wiederherstellbar

Zusammenfassung normales Löschen
- eigentliche Daten nicht gelöscht
- nur Verweise auf Daten gelöscht

Überschreiben
praktizierter Ansatz: Blöcke in mehreren Durchläufen mit Zufallsbits überschreiben

nach Gutmann [Gut96]
Durch Ansatz "scanning transmission electron microscopy" überschriebene Daten wiederherstellbar

Unter Unix: Tool *shred* zum Überschreiben

Löschen durch Überschreiben (1)

"scanning transmission electron microscopy"

- Vorherige gespeicherte Bits hinterlassen elektromagnetische Felder, die die neuen Felder überlagern. Durch Auslesen des analogen Signals und Berechnung der Differenz zum optimalen digitalen Signal (selbstberechnet) kann das vorherige Feld bestimmt werden $\Rightarrow$ Wiederherstellung von überschriebenen Daten
- Keine praktische Verifikation für Funktionalität des Verfahrens
- Wissenschaft ist über Theorie strittig

Löschen durch Überschreiben (2)

Tatsachen

- einmaliges Überschreiben auf jedem Fall sinnvoll, da Daten nicht gelöscht werden
- aber: Überschreiben auf Low-Level Ebene; auf Dateisystemebene ist nicht garantiert, dass die eigentlichen Blöcke überschrieben werden
- durch Festplattencontroller werden defekte Sektoren für den Benutzer nicht sichtbar versteckt $\Rightarrow$ Abschalten dieser Funktionalität

Löschen durch Überschreiben (3)

sicheres Löschen

- Tools überschreiben Blöcke mehrfach mit Zufallsbits
- nicht sicherer als einmaliges Überschreiben mit Zufallsbits nach einem bestimmten Muster um Felderdifferenz unbrauchbar zu machen [Gut96]

Zusammenfassung Festplatten

Recovery Aspekte

- generell können Daten wiederhergestellt werden, aber: Zeitintensiv, Wissen erforderlich
- Erfolg kann nicht garantiert werden, bei zeitnaher Recovery wahrscheinlich

forensische Aspekte

- Daten hinterlassen Spuren auf der Festplatte
- Rückschluss auf Benutzerverhalten möglich

Prävention

- Datensicherung (Recovery)
- Nutzung kryptographischer Ansätze (Forensik)
  - wichtige Dateien verschlüsseln
  - Datenträger verschlüsseln

Verwendung

NAND-Flash

- Datenspeicherung in USB-Flash Disks, digitalen Kameras etc.

NOR-Flash

- Speicherung und Ausführung von Firmware

forensische Problematik

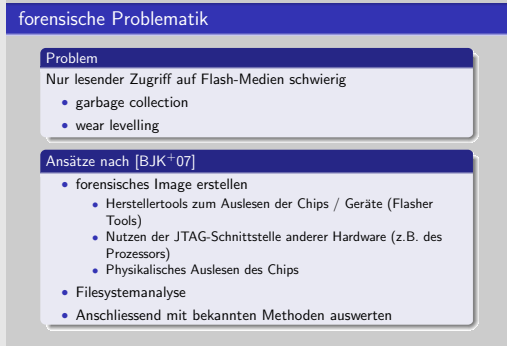
Problem

Nur lesender Zugriff auf Flash-Medien schwierig

- garbage collection
- wear levelling

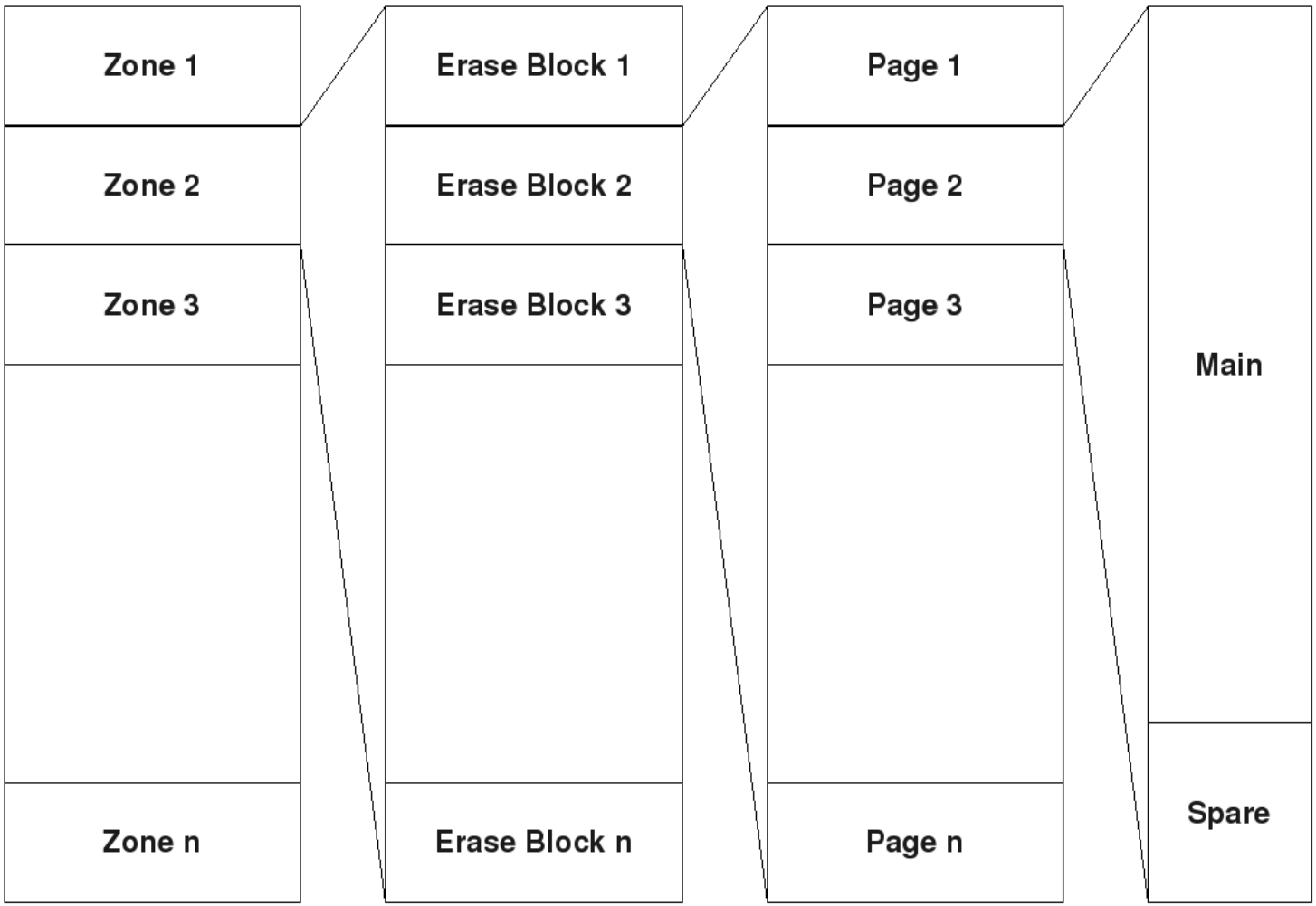
Ansätze nach [BJK+07]

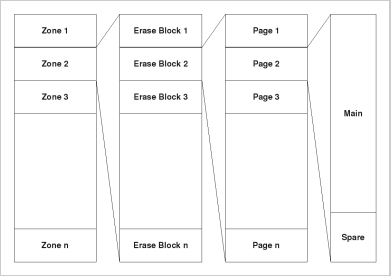
- forensisches Image erstellen
  - Herstellertools zum Auslesen der Chips / Geräte (Flasher Tools)
  - Nutzen der JTAG-Schnittstelle anderer Hardware (z.B. des Prozessors)
  - Physikalisches Auslesen des Chips
- Filesystemanalyse
- Anschliessend mit bekannten Methoden auswerten



Die Algorithmen für *wear levelling* und *garbage collection* sind in der Regel nicht öffentlich und somit ist nicht feststellbar, durch welche Kriterien Operationen auf der Flashdisk ausgelöst werden. Dies erschwert den nur-lesenden Zugriff auf die Disk, was wichtig ist um die Beweiskette aufrecht zu erhalten. Um diese Problematik zu lösen werden 3 Ansätze vorgestellt. Nach der erfolgreichen Erstellung der Imagedatei der Flashdisk ist ein weiteres Problem zu lösen: Das Mapping der System-LBA zur Flash-Disk-LBA muss gelöst werden (LBA = Logical Byte Address, s. folgende Folien). Wenn diese beiden Schritte abgeschlossen sind, kann das konvertierte Image mit den bekannten Methoden ausgewertet werden. Zur Recovery (wo das Schreiben der Daten zwar nicht egal ist, aber nicht generell unterdrückt werden muss) existieren bereits Tools, die die Blöcke des Flash-Dateisystems mit Methoden des *Data Carving* durchsuchen.

Aufbau


Zone 1
Zone 2
Zone 3
Zone n
Erase Block 1
Erase Block 2
Erase Block 3
Erase Block n
Page 1
Page 2
Page 3
Page n
Main
Spare

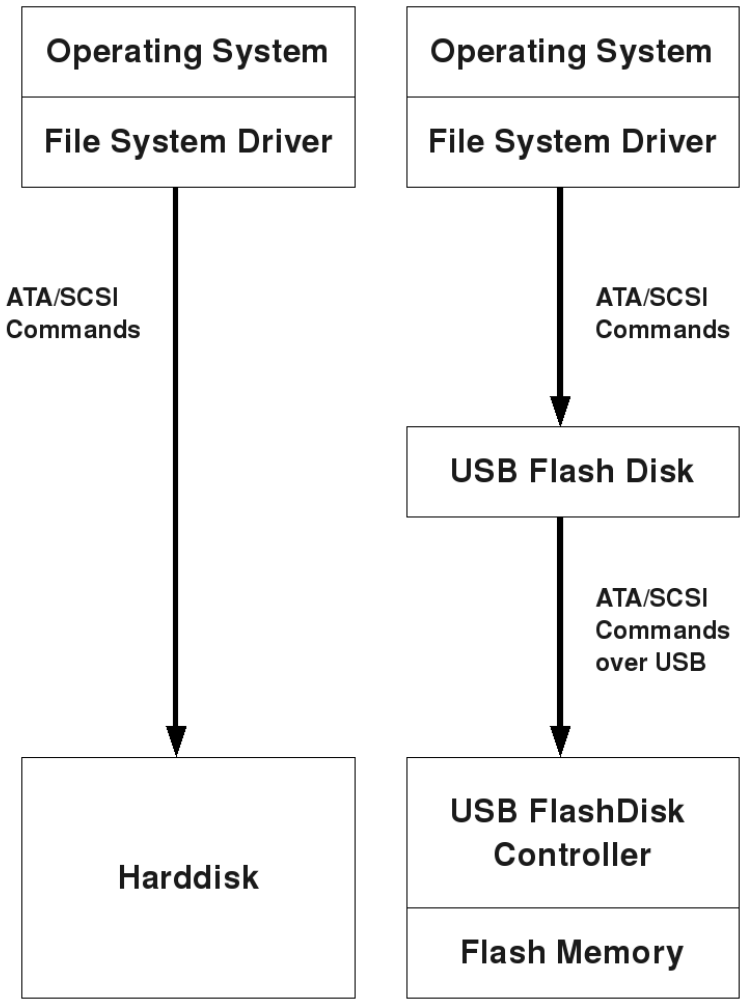


spare-Area:

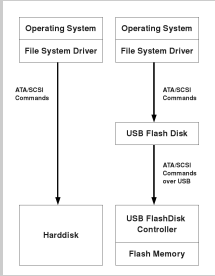
- Status des Blocks, der Page
- Error Correction Codes
- Mapping Informationen
- ...

Löschprozess: Erase Block als gelöscht markiert $\Rightarrow$ wird vom *garbage collector* erfasst, kompletter Block mit '1' gefüllt. Der Zeitpunkt des wirklichen Löschens ist somit nicht bekannt.

Warum ist lesender Zugriff schwierig?



Warum ist lesender Zugriff schwierig?



- USB Flash Disk übersetzt LBA des Systems zu einer anderen LBA auf der FlashDisk
- Mapping ist im Allgemeinen unbekannt, im Flash Memory abgelegt
- **Problem:** Auslesen physikalisch zusammenhängender Blöcke nicht möglich

Flasher Tools

Prinzip

- Hardware mit verschiedenen Adapter für verschiedene Medien (Handys, Kameras, Kartenleser)
- Softwareschnittstelle zur Erstellung eines Images auf ein anderes Medium (z.B. Festplatte)

Vorteil

- zu untersuchendes Medium kann einfach angeschlossen werden

Nachteile

- einige Tools bilden nicht den gesamten Flash-Speicher ab (z.B. ohne “spare-area”)
- es ist ungewiss, ob Schreiboperationen auf dem Medium angestossen werden

JTAG Schnittstelle (1)

Prinzip

- Zugriff über "JTAG test access port" (JTAG: Joint Test Action Group)
- Beim Zugriff über den Prozessor (Extest Mode):
  - Nutzung eines Testlesebefehls, dem eine Adresse übergeben werden kann
  - Durch Wiederholung mit allen Adressen kann gesamtes Medium ausgelesen werden
- weiterer Zugriff über Debug Mode möglich (Prozessoren unterstützen mehrere oder nur einen Modus)

JTAG Schnittstelle (2)

Vorteile

- keine Schreibzugriffe im Extest bzw. Debug Mode
- aber: Bis der Mode erreicht wird, können theoretisch Schreibzugriffe angestossen werden
- Image kann erstellt werden, ohne Flash-Speicher von der Platine zu lösen
- komplettes forensisches Image inklusive “spare area” und “bad blocks”

Nachteile

- langsame Kommunikation
- Finden der JTAG Zugriffstellen
- nicht jede Hardware unterstützt JTAG

Physikalisches Auslesen

Prinzip

- Auslösen des Flash-Chips aus der Platine
- Chip mit “flash memory chip programmer” / “reader” auslesen

Vorteile

- kein Schreibzugriff, da kein Strom
- defekte Systeme können ausgelesen werden
- komplettes forensisches Image

Nachteile

- Schwieriges Auslösen des Chips
- Hardware evtl. beschädigt
- Gefahr den Chip selbst zu zerstören

Filesystemanalyse (1)

Fragestellung

Image vorhanden, doch wie ist die Zuordnung der physikalischen Adressen zu den logischen Blöcken?

Probleme

- Zuordnung unterschiedlich
- Verwendbarkeit des Images
  - bad blocks

Fragestellung

Image vorhanden, doch wie ist die Zuordnung der physikalischen Adressen zu den logischen Blöcken?

Probleme

- Zuordnung unterschiedlich
- Verwendbarkeit des Images
  - bad blocks

Um das Mapping der Adressen zu bestimmen, muss das komplette Image inkusive bad blocks vorliegen, da ansonsten Verschiebungen stattfinden.

Filesystemanalyse (2)

USB Memory Sticks

- NAND-Speicher
- meist kein wear-leveling
- dann Blockgröße des Flash-Dateisystems = Erase-Block Größe
- Erase-Block Größen: 16 kByte, 128 kByte
- Page Größen: 528 Byte, 2112 Byte

Konvertierung des Images zu Filesystem

1. Granularität des Flash Dateisystems?
2. Wo sind die Metadaten?
3. Wie können die Metadaten interpretiert werden?

Filesystemanalyse (3)

Smart media flash file system

- Beispiel zur Speicherung eines FAT-Clusters in einem Flash Speicher
- “End Of Life”
- Zuordnung FAT-Cluster - Erase-Block
- Informationen hierzu in den Metadaten in der spare-Area für jede Page

Konvertierung eines Smart media flash Filesystems

- Sortierung der Erase-Blöcke nach ihrer logischen Block Nummer (innerhalb jeder Zone)
- Bei jeder Page die spare-Area abschneiden

Filesystemanalyse (4)

Unbekanntes Dateisystem

- Reverse-Engineering
- Auswertung der Metadaten in den spare-Areas
- je nach Granularität: Page/Erase-Block
- Suche nach identischen Blöcken in den Metadaten, die Counter innerhalb einer Zone indizieren (Blocknummer)

Mobiltelefone

- neue Problematik: nicht alle Pages werden für Dateisystem genutzt
- z.B. Nutzung einzelner Abschnitte zur Speicherung / Ausführung von Firmware
- wieder andere Vorgehensweise [BJK+07]

Zusammenfassung Flash

Komplexität

- liegt (noch) in der Interpretation des physikalischen Speichers
- wird im Laufe der Zeit aber besser erforscht / dokumentiert sein
- momentan kaum Möglichkeiten für Privatanwender, Daten wiederherzustellen

Zusammenfassung

- forensische Methoden weiterhin anwendbar
- Daten und Nutzerverhalten kann rekonstruiert werden
- *garbage collection* und *wear leveling* bringen zusätzliche Komplexität
- keine Aussage darüber, wann Daten wirklich gelöscht werden

Dynamic Random Access Memory

Verwendung

- Schneller elektronischer Speicher
- Verwendung z.B. als Arbeitsspeicher im PC
- muss in kurzen Zeitabständen aufgrund von Leckströmen elektronisch “aufgefrischt” werden, um Daten zu erhalten

Motivation Recovery

- eigentlich kein Szenario vorstellbar
- aber: forensischer Nutzen
- Passwörter, Schlüssel und andere geheime Informationen im Arbeitsspeicher abgelegt

forensische Methoden DRAM (1)

Szenario Cold-Boot-Attack [HSH+08]

- DRAM bis zu einigen Sekunden nach Abschalten noch lesbar (bei Raumtemperatur)
- bei entsprechender Kühlung bis zu mehreren Minuten / einzelnen Stunden haltbar (-50 Grad)
- 2. Betriebssystem booten (Coldboot) und Image des Speichers erstellen
- Alternativ: RAM kühlen, dann ausbauen und spezielle Hardware zum Auslesen nutzen
- beide Methoden einfach durchführbar, benötigen aber physikalischen Zugang zum Rechner
- auch bei Rechnern im Hibernate- oder Sleep-Modus möglich

forensische Methoden DRAM (2)

Nutzen

- Auslesen von Betriebssystempasswörtern (Mac OS X)
- RSA-Schlüssel auslesen (Apache)
- Key für verschlüsselte Dateisysteme auslesen

Warum bleiben Speicherzellen ohne Refresh erhalten?

Leckströme

- treten im Kondensator auf
- Elektronensprünge zwischen den beiden dotierten Schichten
- werden im Betrieb durch Refresh kompensiert (32-64 ms, herstellerabhängig)
- temperaturabhängig

Eigenschaften DRAM

- ohne Refresh der Speicherzellen nicht mehr erkennbar welches Bit gespeichert wurde
- bei niedriger Temperatur mit geringer Fehlerrate auch nach Minuten noch auslesbar

DRAM-Chip im Laptop

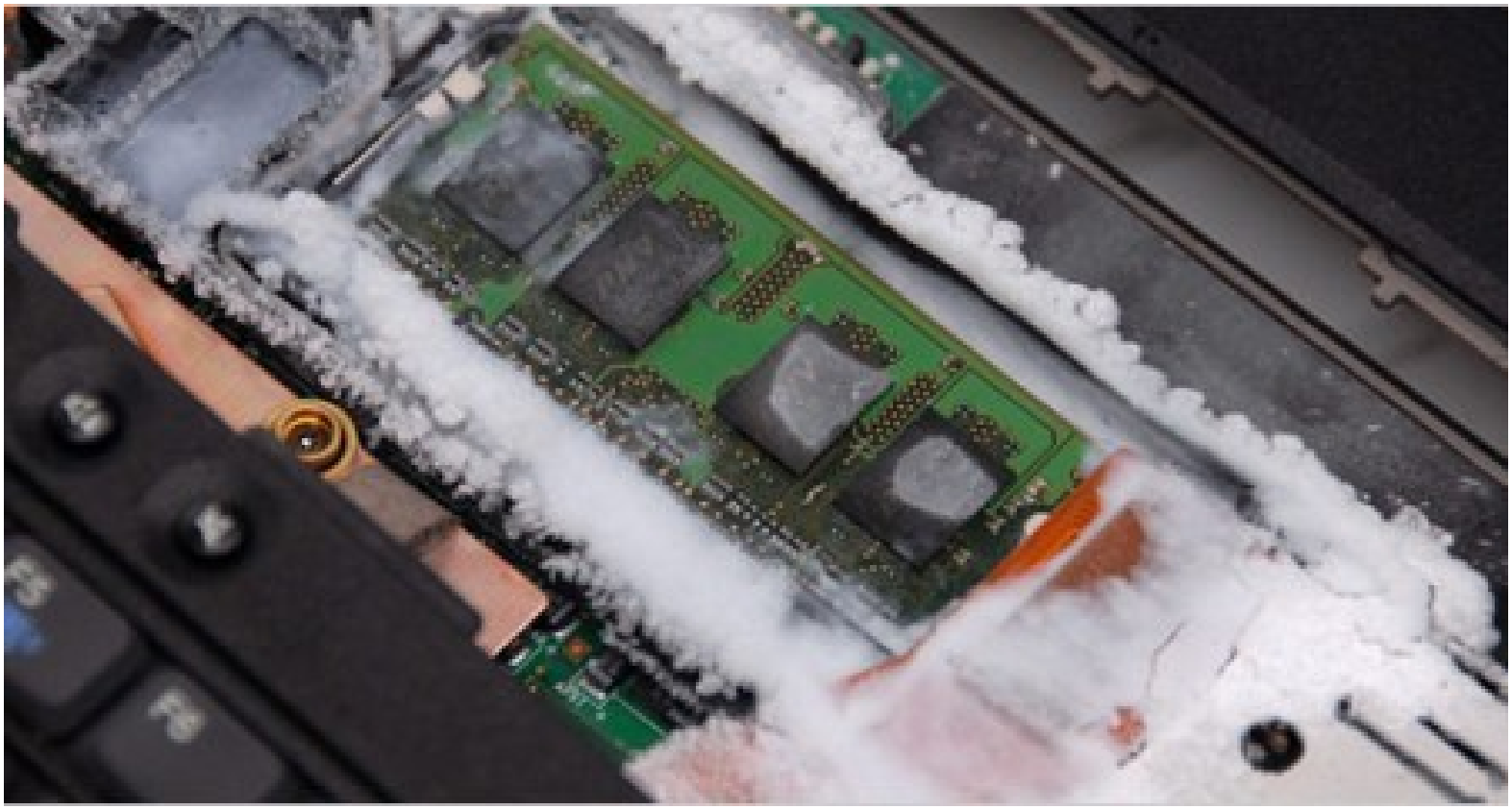
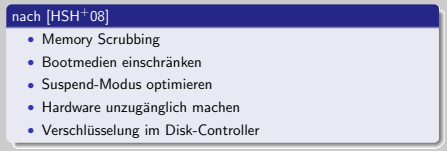


Abbildung: http://www.bizzntech.com/2008/02/24/freeze-memory-chips-steal-encrypted-data

Gegenmaßnahmen

nach [HSH+08]

- Memory Scrubbing
- Bootmedien einschränken
- Suspend-Modus optimieren
- Hardware unzugänglich machen
- Verschlüsselung im Disk-Controller



- **Memory Scrubbing**: Daten im RAM überschreiben, sobald nicht mehr benötigt
- **Bootmedien einschränken**: reversibel
- **Suspend-Modus optimieren**: RAM-Inhalt sichern, z.B. Bildschirmsperre lässt Hauptspeicher unverändert, Suspend-Modus kopiert Arbeitsspeicher auf Festplatte
- **Hardware unzugänglich machen**: RAM-Riegel kann nicht ausgebaut oder gekühlt werden, Sensoren zum Schutz
- **Verschlüsselung im Disk-Controller**: Write-Only-Memory im Diskcontroller, Key wird in den Speicher geschrieben und mit diesem Key verschlüsselt bzw. entschlüsselt bei jeder Schreib- oder Leseoperation der Festplatte. Dieser Speicher muss dann natürlich vom Diskcontroller gelesen werden, insofern verschiebt sich nur der Angriffspunkt.

Zusammenfassung

- gelöschte Daten wiederherstellbar
- für Festplatten und CD's viele Tools vorhanden
- Erfolgswahrscheinlichkeit abhängig vom Recoveryzeitpunkt
- einfacher: Prävention
- Datenverschlüsselung ist sinnvoll, aber auch angreifbar


Breeuwsma, Marcel ; Jongh, Martien de ; Klaver, Coert ; Knijff, Ronald van d. ; Roeloffs, Mark:
Forensic Data Recovery from Flash Memory.
In: *Small Scale Digital Device Forensics Journal* (2007), Juni.
http://www.ssddfj.org/papers/SSDDFJ_V1_1_Breeuwsma_et_al.pdf


Carrier, Brian:
Why Recovering a Deleted Ext3 File Is Difficult... / linux.sys-con.com.
Version: August 2005.
http://linux.sys-con.com/node/117909.
2005. –
Forschungsbericht


community, Ubuntu:
DataRecovery / www.ubuntu.com.
Version: 2009.

https://help.ubuntu.com/community/DataRecovery. 2009. –
Forschungsbericht

Freiling, Prof. Dr. F.:
Computerforensik Vorlesung / Hochschule Mannheim - Lehrstuhl für praktische Informatik 1.
Version: 2007.
http://pi1.informatik.uni-mannheim.de. 2007. –
Forschungsbericht

Getchell, Abe:
Data Recovery on Linux and ext3 / www.securityfocus.com.
Version: August 2008.
http://www.securityfocus.com/infocus/1902/2. 2008. –
Forschungsbericht


Gutmann, Peter:
Secure Deletion of Data from Magnetic and Solid-State Memory.
In: *Sixth USENIX Security Symposium Proceedings* (1996), Juli.
http://www.cs.auckland.ac.nz/~pgut001/pubs/secure_del.html


Halderman, J. A. ; Schoen, Seth D. ; Heninger, Nadia ; Clarkson, William ; Paul, William ; Calandrino, Joseph A. ; Feldman, Ariel J. ; Appelbaum, Jacob ; Felten, Edward W.:
Lest We Remember: Cold Boot Attacks on Encryption Keys.
In: *USENIX Security Symposium* (2008), Februar.
http://citp.princeton.edu.nyud.net/pub/coldboot.pdf


Online, Welt:

Militärische Zugangscodes bei EBAY versteigert.
In: *www.welt.de* http://www.welt.de/webwelt/article1155141/Militaerische_Zugangscodes_bei_Ebay_versteigert.html


UNKNOWN:
Recover Deleted Files with Foremost,scalpel in Ubuntu / www.ubuntugeek.com.
Version: September 2008.
http://www.ubuntugeek.com/recover-deleted-files-with-foremostscalpel-in-ubuntu.html.
2008. –
Forschungsbericht. –
User Post


VARIOUS:
Data recovery.

In: *Wikipedia.org* (2008).
http://en.wikipedia.org/wiki/Data_recovery

VARIOUS:
Gutmann method.
In: *Wikipedia.org* (2008).
http://en.wikipedia.org/wiki/Gutmann_method

VARIOUS:
Reed–Solomon error correction.
In: *Wikipedia.org* (2009).
http://en.wikipedia.org/wiki/Reed\OT1\
textendashSolomon_error_correction

WOOD, Carlo:
Howto recover deleted files on an ext3 file system / www.xs4all.nl.
Version: März 2008.

http://www.xs4all.nl/~carlo17/howto/undelete_ext3.html.

2008. –
Forschungsbericht