HOCHSCHULE MITTWEIDA
University of Applied Sciences

Betriebssysteme

macOS – Teil 5

Leander Hoßfeld, B.Sc.

Autor: Prof. Ronny Bodach

Bundeskriminalamt

macOS Agenda

1. Einführung in macOS
2. macOS Bedienung
3. macOS Lab & Image Einbindung
4. Bootcamp Besonderheiten (& Parallels)
5. Mac FHS und Speicherstrukturen
6. Datenformate SQLite und Plist
7. Zuletzt genutzte Elemente & Nutzeraktivitäten
8. Spotlight und erweiterte Metadaten
9. Gelöschte Dateien
10. Schlüsselbund
11. Logdateien
12. Mac Disk Images
13. Time Machine und lokale Backups
14. Kommunikations-Apps
15. Browser Artefakte
16. Cloud
17. iOS Backups

(C) 07.07.2023 Hochschule Mittweida

macOS
Agenda

11. Logdateien
12. Mac Disk Images
13. Time Machine und lokale Backups

(C) 07.07.2023 Hochschule Mittweida

BETRIEBSSYSTEM macOS

Logdateien

Logdateien

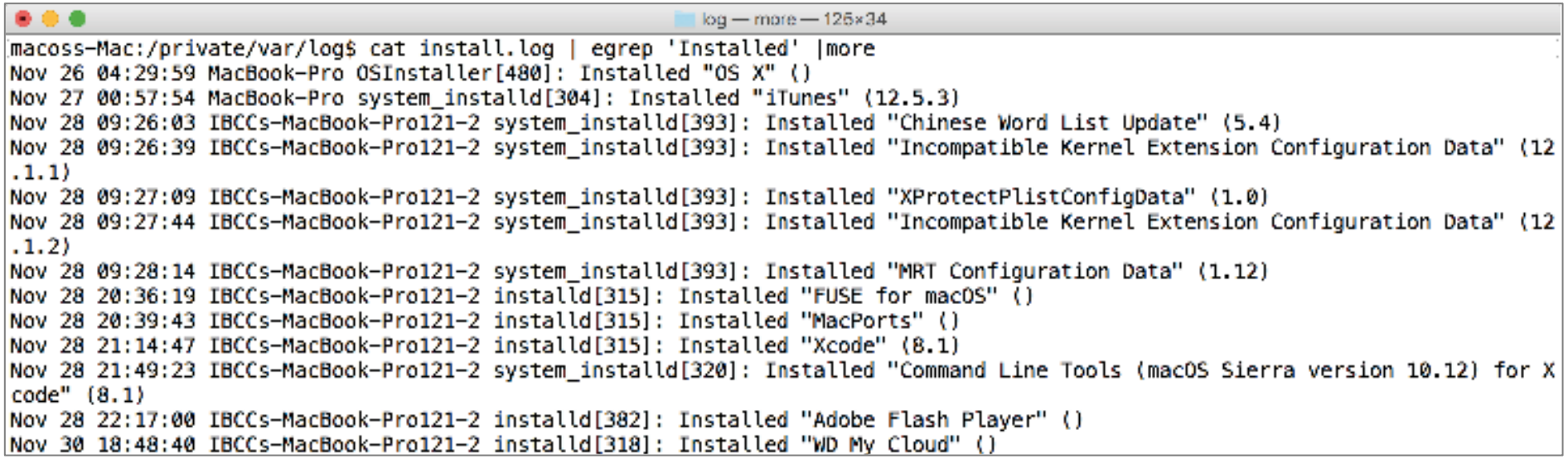
- **Log-Dateien des Betriebssystems**
  - Befinden sich im Verzeichnis **/private/var/log**
  - MacOS hat bis zur Version macOS 10.12 / 11 BigSur verschiedene „überlieferte“ Logdateiformate genutzt
  - Die Log Formate stammen aus dem Betriebssystem Sun Solaris und wurde in verschiedenen Formen von BSD Unix, einschließlich Mac OS übernommen

- **Nutzer-/Account-Informationen**
  - Informationen zu eingeloggten Nutzern bzw. Nutzer-Accounts können in der Datei **/private/var/log/accountpolicy.log** eingesehen werden

(C) 07.07.2023 Hochschule Mittweida

Logdateien

- **Software-Installationen**
  - Installationen von Betriebssystem-Versionen, Updates und Apps lassen sich in der Datei */private/var/log/install.log* ermitteln
  - Die Logdatei kann mit dem Terminalbefehl cat install.log | grep "Suchbegriff" analysiert werden.



(C) 07.07.2023 Hochschule Mittweida

Logdateien

- **Filesystem Check**
  - Die Log-Dateien **/private/var/log/fsck_hfs.log** oder **/private/var/log/fsck_apfs.log** zeigen an, wann eine Dateisystemprüfung für die Dateisysteme HFS+ und Apple File System durchgeführt wurde.
  - Neben dem Character Device wird das Ergebnis der Prüfung (CLEAN, ERROR) ausgegeben.

(C) 07.07.2023 Hochschule Mittweida

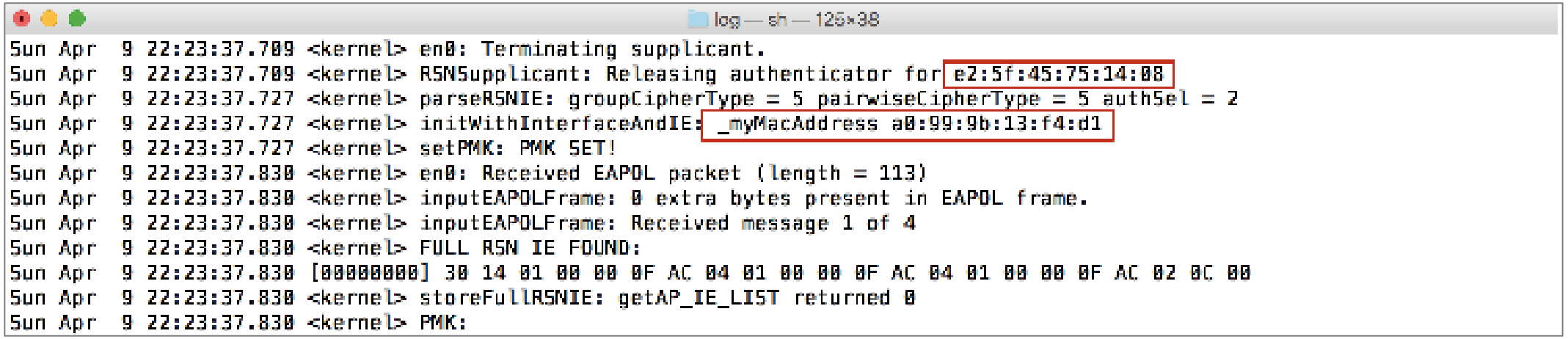
Logdateien

- **Storage Manager**
  - Die Log-Datei **/private/var/log/com.apple.revisiond/revisiond.log** zeigt Informationen zum Mac-OS-Storage-Manager *revisiond*
  - der Hintergrunddienst verwaltet unterschiedliche Fassungen von Dokumenten, die von Applikationen oder Mac-OS-Systemdiensten erzeugt wurden
  - in das Verzeichnis private/var/log/com.apple.revisiond/ kann nur mit Root-Rechten navigiert werden
  - die Log-Datei kann u. a. Hinweise auf Volumes geben, beispielsweise bei der Löschung von Cache-Speicher

(C) 07.07.2023 Hochschule Mittweida

Logdateien

- **WIFI**
  - Von Mac OS hergestellte WiFi-Verbindungen lassen sich in der Log-Datei /private/var/log/wifi.log auffinden
  - die Log-Datei speichert WiFi-Verbindungen der letzten 24 Stunden und zeigt detaillierte Informationen zum Verbindungsaufbau mit WiFi-Access- Points
  - WiFi-Log-Dateien von vergangenen Tagen werden bzip2-komprimiert abgespeichert



Verbindungsaufbau mit der BSSID E2:5F:45:75:14:08 von der eigenen Mac-Adresse.

(C) 07.07.2023 Hochschule Mittweida

Logdateien

- **Periodische Log-Dateien**
  - Die Log-Dateien werden von Mac OS in periodischen Abständen angelegt und protokollieren systemspezifische Vorgänge.
  - Die Log-Dateien befinden sich unter /private/var/log und sind mit einem Texteditor oder im Terminal beispielsweise mit dem Kommando cat lesbar.
  - Sie beinhalten die folgenden Informationen:
    - **daily.out:** Disk-Status, Network-Status, System-Status
    - **weekly.out**: Protokollierung des Neuaufbaus der Whatis-Datenbank
    - **monthly.out**: Statistik zu Nutzer-Logins
  - Die Whatis-Datenbank ist eine BSD-Komponente zur Ausgabe von Informationen zu Schlüsselbegriffen. Zum Beispiel können mit dem Befehl whatis grep Informationen zum UNIX-Kommando grep abgerufen werden.

(C) 07.07.2023 Hochschule Mittweida

Logdateien

- **Periodische Log-Dateien**
  - Die Log-Dateien werden von Mac OS in periodischen Abständen angelegt und protokollieren systemspezifische Vorgänge.

log — -bash — 118×33

```
IBCCs-MacBook-Pro121:log ibcc$ cat daily.out | grep "\Volumes"
/dev/disk2s1   1.9Gi   13Mi  1.9Gi     1%       512          0  100%   /Volumes/ITB LIVE
/dev/disk2s2   465Gi  465Gi    0Bi   100% 122012666          0  100%   /Volumes/iacis
/dev/disk2s2   7.2Gi  5.4Gi  1.8Gi    75%  1405151     476949   75%   /Volumes/Install OS X Yosemite
/dev/disk4s2    46Gi   17Gi   29Gi    38%  4570055    7511050   38%   /Volumes/OSX 10.11
/dev/disk4s8    46Gi  149Mi   46Gi     1%    38236   12104244    0%   /Volumes/OSX 10.6
/dev/disk4s4    47Gi  426Mi   46Gi     1%   109031   12097998    1%   /Volumes/OSX 10.10
/dev/disk4s5    47Gi  416Mi   46Gi     1%   106605   12085026    1%   /Volumes/OSX 10.9
/dev/disk4s9    47Gi  289Mi   46Gi     1%    73913   12117718    1%   /Volumes/OSX 10.5
/dev/disk4s12   46Gi   42Gi  3.9Gi    92% 11109944    1016152   92%   /Volumes/OSX Software I
/dev/disk4s6    46Gi  149Mi   46Gi     1%    38194   12120670    0%   /Volumes/OSX 10.8
```

Auszug aus der Log-Datei daily.out, grep-Suche nach eingehängten Volumes

(C) 07.07.2023 Hochschule Mittweida

Logdateien

- **System Log Dateien im Text Format**
  - bis zur macOS Version 10.8 gab es drei Log Dateien im Text Format die Systemnachrichten speicherten:
    - **secure.log**, **kernel.log** und **system.log**
  - ab macOS Version 10.8 wurden diese zum **system.log** zusammengefasst
  - ab macOS Version 10.12 und mit Version 11 BigSure werden Systemnachrichten im **Apple Unified Log (AUL)** zusammengefasst aufgezeichnet

(C) 07.07.2023 Hochschule Mittweida

Logdateien

- **Binary Apple System Log (ASL):**
  - Mac OS hat als BSD-System Zugang zum unter UNIX-Systemen weit verbreiteten Log-Mechanismus System Log
  - System Log (syslog) ist ein Hintergrundprozess, der von verschiedenen Komponenten des Betriebssystems Meldungen entgegennimmt und aufzeichnet
  - ab Mac OS X 10.4 wurde der UNIX-Mechanismus in Apple System Log (ASL) überführt
  - ASL Grundlage ist syslog und daher mit älteren Standards kompatibel

(C) 07.07.2023 Hochschule Mittweida

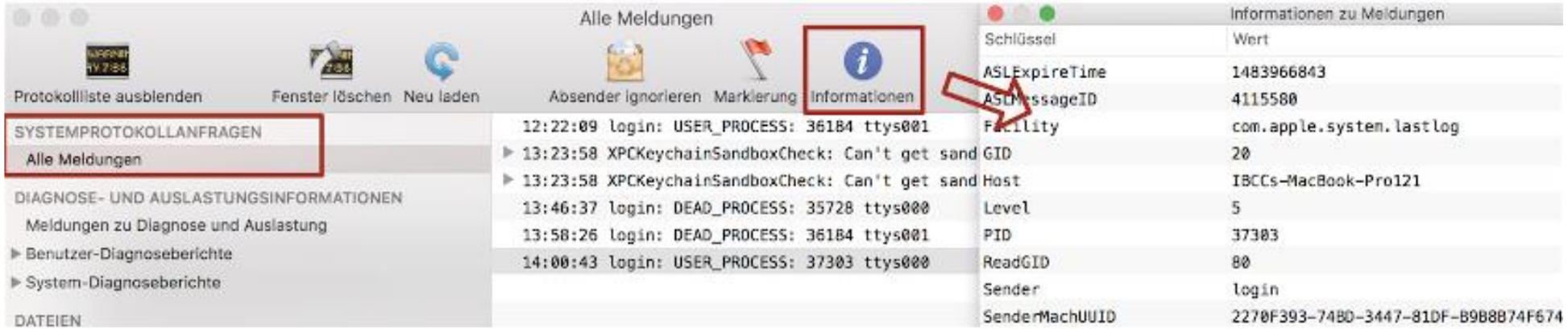
Logdateien

- **Binary Apple System Log (ASL):**
  - ASL-Logfiles befinden sich im Verzeichnis **/private/var/log/asl**
  - haben ein binäres Format mit der Signatur *ASL DB*
  - Grundsätzlich gibt es zwei verschiedene Arten von ASL-Log-Dateien:
    - Dateien, die mit Zeitstempeln im Format **YYYY.MM.DD.*.asl**
    - Dateien mit der Bezeichnung **AUX.YYYY.MM.DD.asl**
  - Beide speichern Informationen für einen Zeitraum von 7Tagen bevor sie in die Log-Dateien mit der Bezeichnung BB.*.asl überführt werden
  - BB.*.asl Dateien beinhalten eine monatliche Speicherung der protokollierten Aktionen und werden für ein Jahr gespeichert

(C) 07.07.2023 Hochschule Mittweida

Logdateien

- **Binary Apple System Log (ASL):**
  - ASL-Log-Dateien können mit der Console.app unter Auswahl von Systemprotokollanfragen/ Alle Meldungen intuitiv betrachtet werden.
  - Jede ASL-Meldung enthält weitere Informationen, die in Schlüsseln (ASL Keys) abgelegt sind welche durch Auswahl der Schaltfläche Informationen verfügbar sind

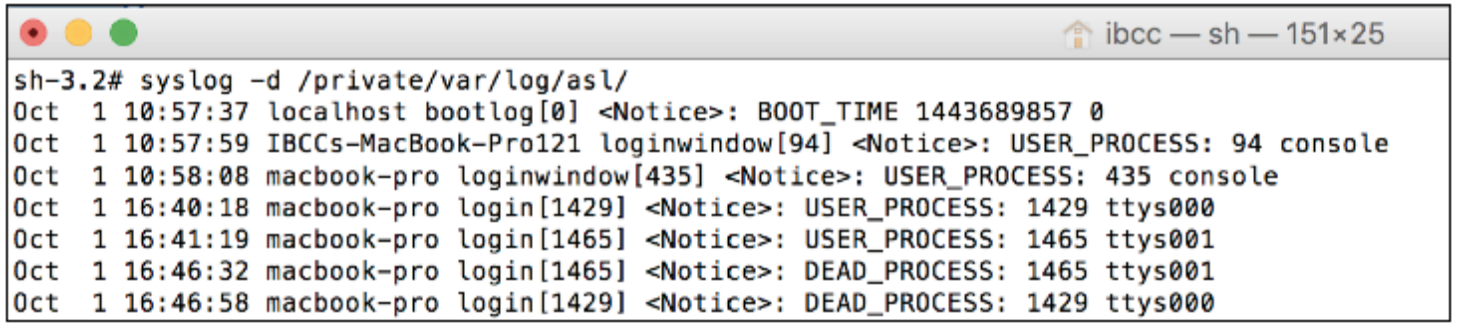


ASL-Meldung eines Bash-Login-Vorgangs

(C) 07.07.2023 Hochschule Mittweida

Logdateien

- **Binary Apple System Log (ASL):**
  - Alternativ kann eine Ausgabe auch unter Zuhilfenahme des Kommandos **syslog** erfolgen
  - Dabei können einzelne Dateien oder ganze Verzeichnisse mit entsprechenden ASLLog- Dateien betrachtet werden.
  - ➢ der Befehl **syslog -d /private/var/log/asl/** gibt die Inhalte des gesamten ASL-Log-Verzeichnisses wieder



Auszug aus der Ausgabe der ASL-Log-Dateien mit syslog

(C) 07.07.2023 Hochschule Mittweida

Logdateien

- **Basic Security Modules Audit Logs (BSM):**
  - das Dateiformat Basic Security Module (BSM) stammt aus dem Betriebssystem Sun Solaris und wurde in verschiedenen Formen von BSD Unix, einschließlich Mac OS X, übernommen.
  - Audit-Logs basieren auf einer Apple-eigenen Implementierung des Basic-Security-Module (BSM) in Darwin
  - Audit-Logs ermöglichen die Verfolgung und Prüfung von Aktionen, die von Nutzern oder Prozessen durchgeführt werden

(C) 07.07.2023 Hochschule Mittweida

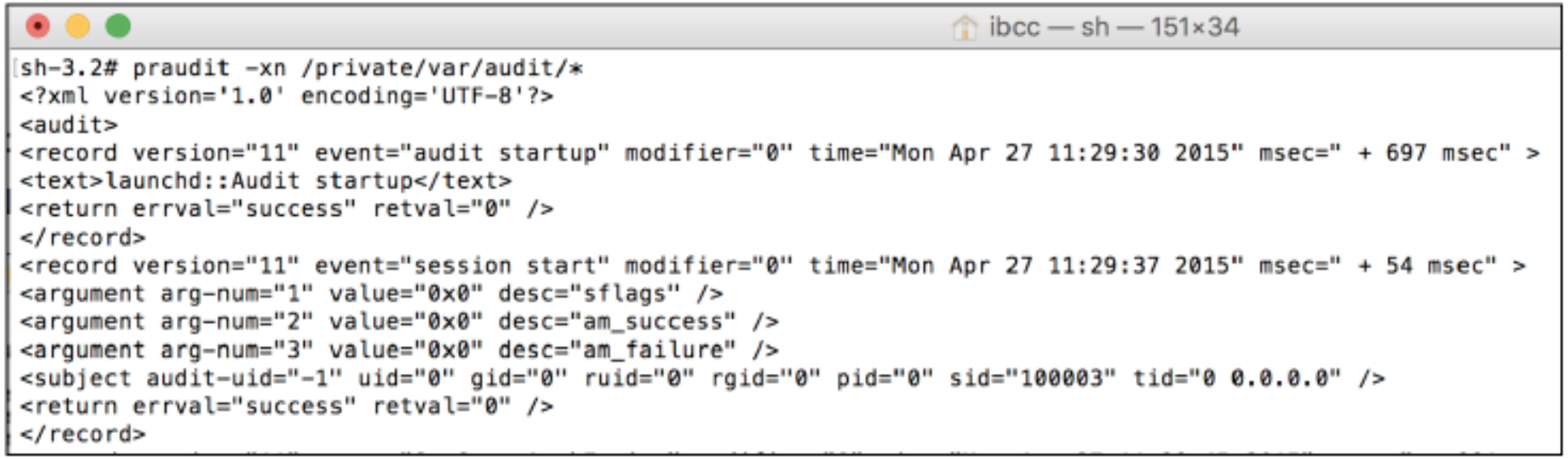
Logdateien

- **Basic Security Modules Audit Logs (BSM):**
  - Audit-Logs sind im Verzeichnis **/private/var/audit/** abgelegt welches nur mit Root-Rechten geöffnet werden kann
  - Log-Dateien liegen in einem binären Format vor und haben das Format **[Start-Zeitstempel].[End-Zeitstempel]**.
  - die Log-Datei **[Start-Zeitstempel].not terminated** ist die aktuell beschriebene Audit-Log-Datei
  - Zeitstempel haben das Format YYYYMMDDHHMMSS

(C) 07.07.2023 Hochschule Mittweida

Logdateien

- **Basic Security Modules Audit Logs (BSM):**
  - um Audit-Log-Dateien auszuwerten, verwendet man das Kommando **praudit -xn**
  - dabei stehen die Parameter -x für eine XML-Ausgabe und -n für eine Konvertierung der Nutzer- und Gruppen-IDs



Auszug der Ausgabe der Audit-Logfiles mit praudit –xn

(C) 07.07.2023 Hochschule Mittweida

Logdateien

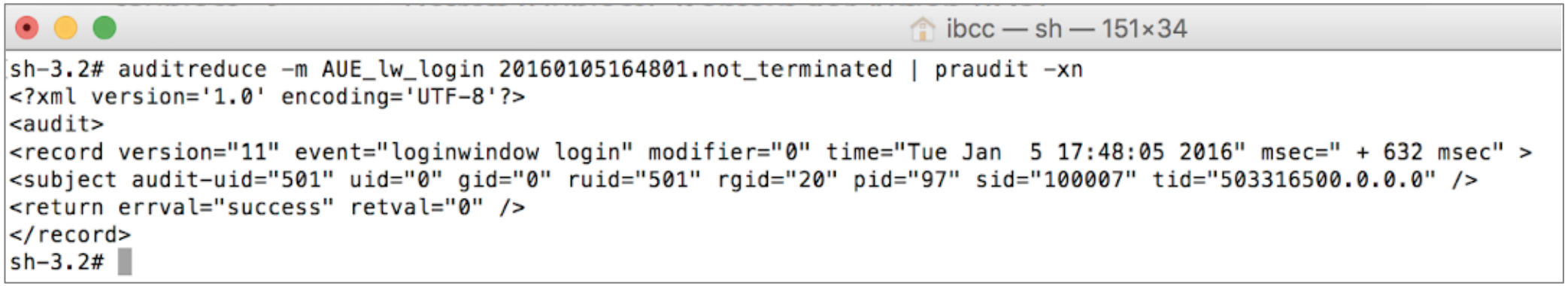
- **Basic Security Modules Audit Logs (BSM):**
  - Audit-Einträge sind durch Tags (ähnlich wie ein XML-/ HTML-Dokument) nach folgendem Muster hierarchisch strukturiert:

Struktur eines Audit-Eintrags	
‹record...›	Beginn des Eintrags (Header)
‹subject...›	Betreff (Subject), Kontext der Aktion (IDs)
‹text›	String mit Beschreibung zur Aktion
‹return›	Rückgabewert
‹/record›	Ende des Eintrags

(C) 07.07.2023 Hochschule Mittweida

Logdateien

- **Basic Security Modules Audit Logs (BSM):**
  - einzelne Aktionen aus den Audit-Log-Dateien können mit dem Befehl **auditreduce** gefiltert werden.
  - Filterung nur nach bestimmten Audit-Events, die in der Konfigurationsdatei **/etc/security/audit_event** beschrieben sind



Filtern nach Nutzer-Logins (AUE_lw_login) in der Audit-Datei 20160105164801.not_terminated

(C) 07.07.2023 Hochschule Mittweida

Logdateien

- **Apple Unified Log (ALU)**
  - mit macOS 10.12 Sierra ersetzte Apple im September 2016 das traditionelle textbasierte Unix-Protokollsystem durch sein neues einheitliches Protokoll, dem **Apple Unified Logging**.
  - der Log-Mechanismus ersetzt die bisherigen Log-Dateien system.log und Apple System Logs (ASL) bzw. führt sie zusammen
  - das neue Unified Logging wird plattformübergreifend auch in iOS (ab iOS-Version 10), watchOS und tvOS eingesetzt.
  - mit macOS Sierra 10.12 wird das Unified Logging parallel zu den bekannten Log-Dateien betrieben, ab macOS 11 BigSure ausschließlich

(C) 07.07.2023 Hochschule Mittweida

Logdateien

- **Apple Unified Log (ALU)**
  - es wurden jedoch nicht alle Protokolle im einheitlichen Protokoll gebündelt.
  - zu den verbleibenden traditionellen textbasierten Protokolldateien gehören:
    - daily.out, monthly.out und wifi.log sind noch aktiv
    - /var/log/install.log führt immer noch ein wertvolles Protokoll der Installation von Softwareupdates
    - CUPS führt immer noch seine eigenen Protokolle in /var/log/cups aus
    - Apps von Drittanbietern wie die von Adobe führen immer noch ihre eigenen Textprotokolle aus
    - system.log existiert noch, wird nur noch von veralteter Software besucht, z.B. dem Google Software Update.

(C) 07.07.2023 Hochschule Mittweida

Logdateien

- **Apple Unified Log (ALU)**
  - es gibt drei Hauptgruppen von Dateien, die Protokolleinträge speichern:
    1. die in **/var/db/diagnostics/Persist/** in Form von tracev3-Dateien, die reguläre Protokolleinträge enthalten
    2. die in **/var/db/diagnostics/Special/** in tracev3 Dateien, die zusätzliche Einträge mit kürzerer Lebensdauer enthalten
    3. das Logging-System speichert Daten zudem im Verzeichnis **/var/db/uuidtext**
       - Dieses enthält diverse Unterverzeichnisse mit Dateien, die mit GUIDs benannt sind und Log-Mitteilungen in Textform enthalten, welche von Unified Logging zur Darstellung von Log-Mitteilungen verwandt werden

(C) 07.07.2023 Hochschule Mittweida

Logdateien

- **Apple Unified Log (ALU)**
  - Log-Dateien des Unified Logging sind nach dem folgenden Muster abgespeichert:

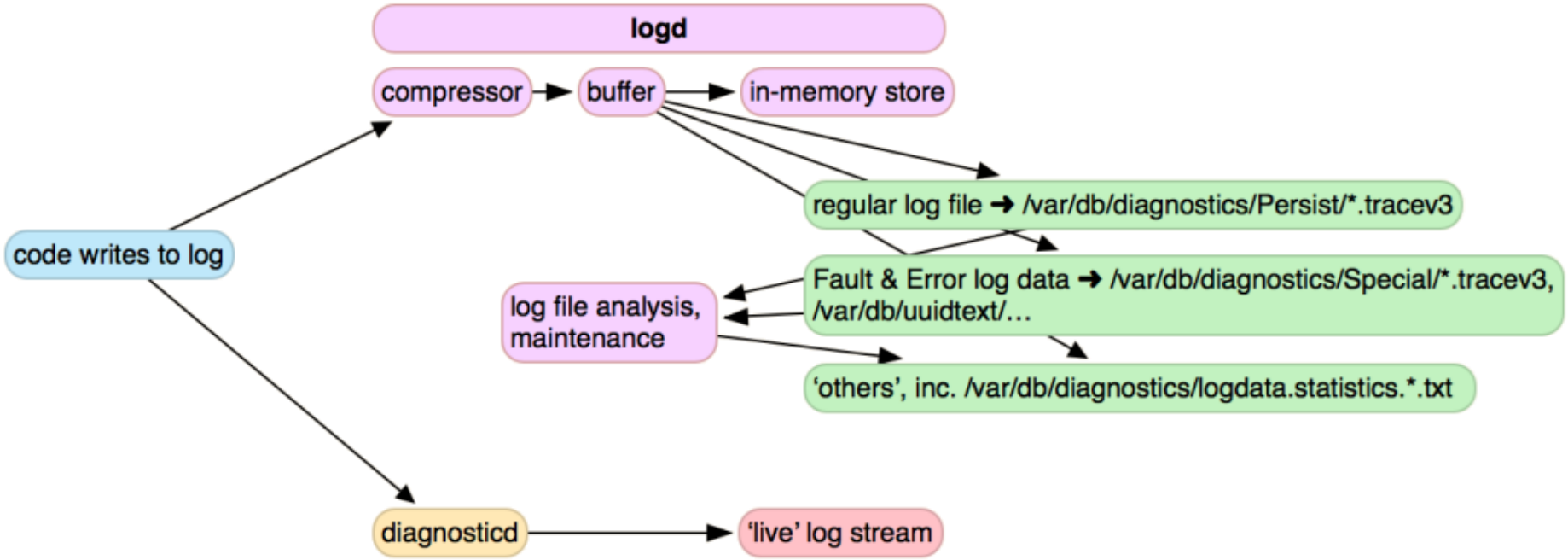
logdata.Persistent.YYYYMMDDTHHMMSS.tracev3

  - das Unified-Logging-System nutzt die Hintergrundprozesse **logd** und **diagnosticd**
  - zur Laufzeit einer Anwendung verarbeitet **logd** Mitteilungen, komprimiert sie und speichert sie in eine normale **Log-Datei**
  - der Hintergrunddienst **diagnosticd** nimmt Mitteilungen entgegen und erzeugt einen **Live-Log-Stream**

(C) 07.07.2023 Hochschule Mittweida

Logdateien

- **Apple Unified Log (ALU) -**



Quelle: eclecticlight.co/2018/03/19/macos-unified-log-1-why-what-and-how/

(C) 07.07.2023 Hochschule Mittweida

Logdateien

- **Apple Unified Log (ALU)**
  - Log-Mitteilungen des Unified-Logging-Systems können durch die API (os_log) als Default-, Info- oder Debug-Mitteilungen definiert werden
  - für jede Art kann definiert werden, ob ein Logging aktiviert ist und wo Log-Dateien gespeichert werden (Festplatte oder Hauptspeicher).
  - ➢ Die folgende Tabelle zeigt die Standardeinstellungen für Mitteilungen:

Standardeinstellungen für Mitteilungen im Unified-Logging-System		
Mitteilungsart	Status	Speicherort
Debug-Level	nicht aktiviert	-
Info-Level	aktiviert	Hauptspeicher
Default-Level	immer	Festplatte
Error	immer	Festplatte
Fault	immer	Festplatte

(C) 07.07.2023 Hochschule Mittweida

Logdateien

- **Apple Unified Log (ALU)**
  - Um die erstellten Log-Dateien oder den Live-Log-Stream betrachten zu können, kann die in Mac OS integrierte Konsole-App oder das Terminalkommando log eingesetzt werden
  - Zum **Exportieren** und Öffnen von **Log-Mitteilungen** auf weiteren Mac-OS-Systemen kann das **neues Speicherformat .logarchive** eingesetzt werden
  - Unified Logging ermöglicht eine Kategorisierung von Log-Mitteilungen sowie verbesserte Filtermöglichkeiten bei der Analyse

(C) 07.07.2023 Hochschule Mittweida

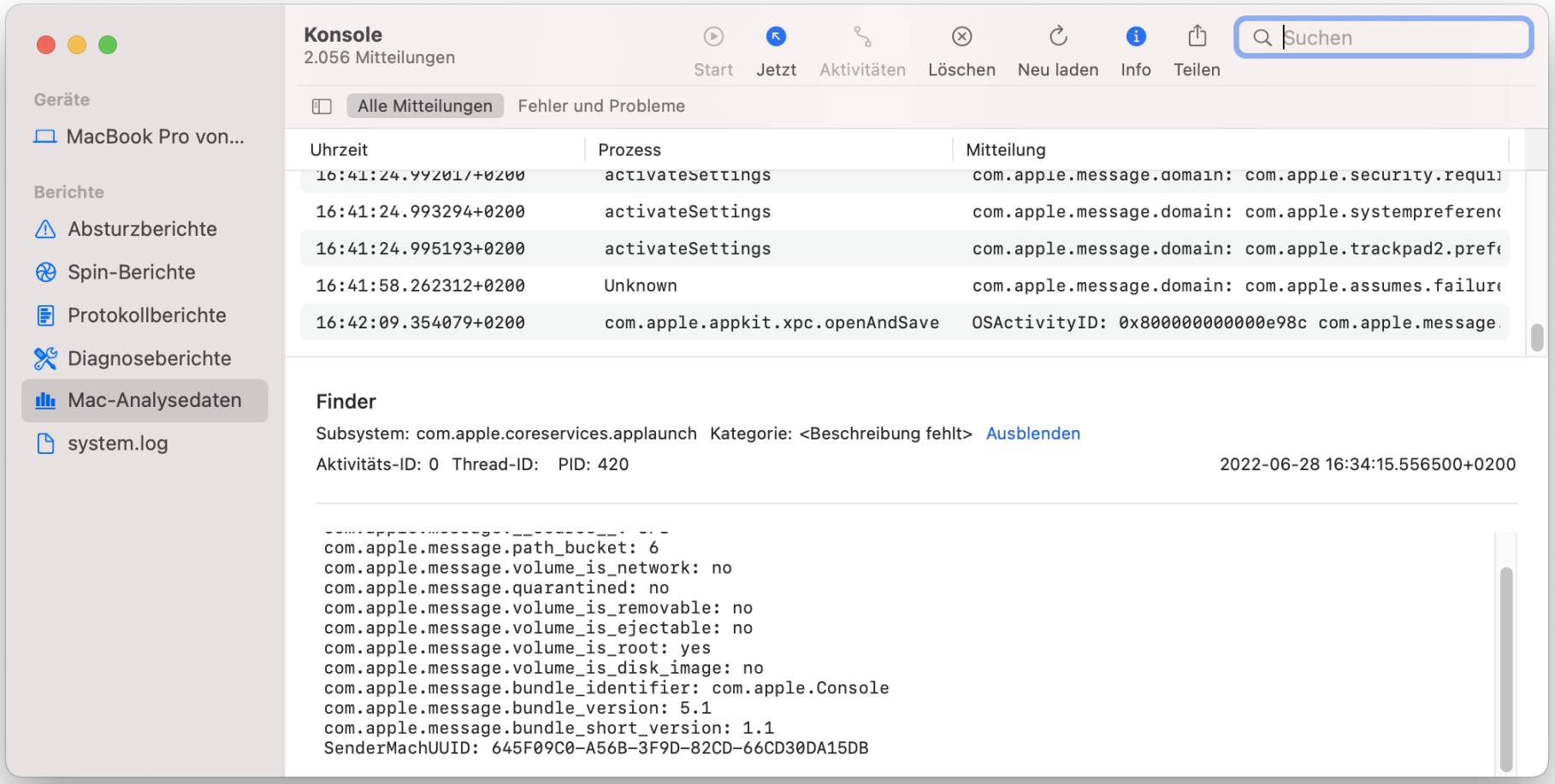
Logdateien

- **Apple Unified Log (ALU)**
  - in der Konsole-App werden unter Auswahl des Symbols des Mac-Computers die Mitteilungen des Unified Logging ausgewertet.
  - Live-Sicht ermöglicht aktuelle Betriebssystem-Meldungen in Echtzeit zu verfolgen (Live Log Stream)
  - Fehler (Fault und Error) werden mit einem gelben Punkt gekennzeichnet und können explizit gefiltert werden
  - Info erlaubt die Anzeige von erweiterten Informationen für eine geloggte Meldung (Unter anderem die Kategorisierung einer Mitteilung nach Subsystem und Kategorie)
  - mit Details kann die Info- Sicht nochmals erweitert werden

(C) 07.07.2023 Hochschule Mittweida

Logdateien

- **Apple Unified Log (ALU)**



(C) 07.07.2023 Hochschule Mittweida

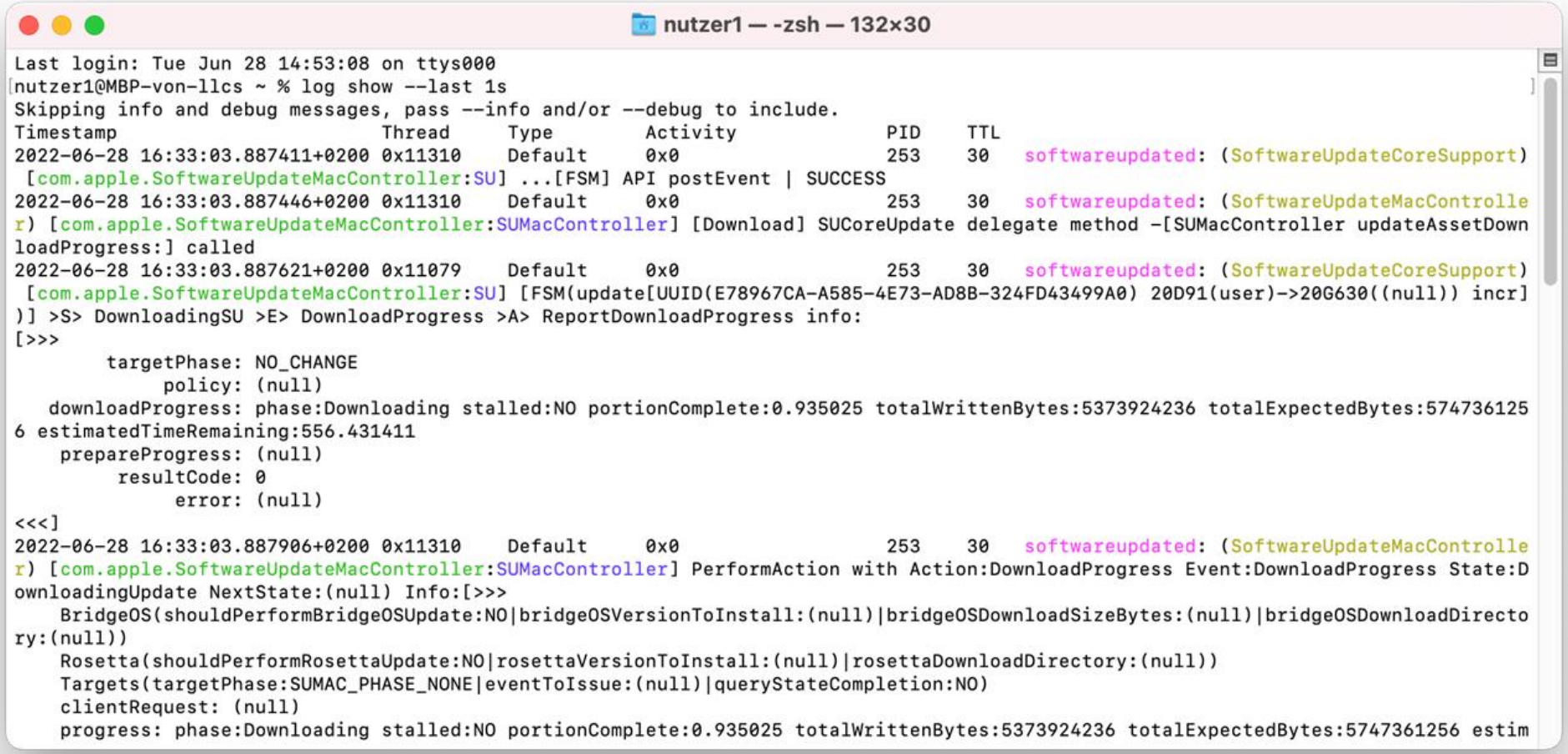
Logdateien

- **Apple Unified Log (ALU)**
  - es kann auch das Terminalkommando log eingesetzt werden, um den Live-Log-Stream oder vom Unified-Logging-System erstellte Log-Dateien darzustellen.
  - das **log-Kommando** bietet dabei erweiterte Möglichkeiten zur Filterung und Suche:
    - **log show:** Zeigt den Inhalt von .tracev3 Log-Dateien oder .logarchive-Dateien an
    - **log collect:** Sammelt Informationen in einer Archivdatei (.logarchive) zur späteren Verwendung mit log oder der Konsole
    - **log stream:** Zeigt den Live-Log-Stream
    - **log config:** Konfiguriert das Unified-Logging-System

(C) 07.07.2023 Hochschule Mittweida

Logdateien

- **Apple Unified Log (ALU)**



(C) 07.07.2023 Hochschule Mittweida

Logdateien

- **Apple Unified Log (ALU)**
  - **log show** zeigt den Inhalt von .tracev3 Log-Dateien / .logarchive-Dateien an

Syntax: log show	
Befehl	Bedeutung
$ log show --archive Datei	Zeigt archivierte .logarchive-Dateien an.
$ log show --file Datei	Zeigt .tracev3-Dateien an (die Datei muss sich in .logarchive oder in einem System-Log-Verzeichnis befinden.
$ log show --predicate Filterausdruck	Ermöglicht die Angabe eines Filterausdrucks (predicate).
$ log show --start Datum/Zeit --end Datum/Zeit	Grenzt Mitteilungen nach Start- und Endzeit zeitlich ein (akzeptiert sind die Formate „YYYY-MM-DD“, „YYYY-MM-DD HH:MM:SS“ oder „YYYY-MM-DD HH:MM:SSZZZZZ“).

(C) 07.07.2023 Hochschule Mittweida

Logdateien

- **Apple Unified Log (ALU)**
  - **log collect** eignet sich zum sammeln von Informationen in einer Archivdatei (.logarchive)

Syntax: log collect	
Befehl	Bedeutung
$ log collect --output Pfad	Setzt den Ausgabepfad für die .logarchive-Datei.
$ log collect --start Datum/Zeit	Grenzt Mitteilungen nach Anfangszeit ein (akzeptiert sind die Formate „YYYY-MM-DD“, „YYYY-MM-DD HH:MM:SS“ oder „YYYY-MM-DD HH:MM:SSZZZZZ“).
$ log collect --last Num [m|h|d]	Grenzt Mitteilungen zeitlich ein. Ausgehend von der aktuellen Zeit wird eine bestimmte Periode zurückgerechnet, z. B. last 2m (die letzten 2 Minuten) oder last 3h (die letzten 3 Stunden).

(C) 07.07.2023 Hochschule Mittweida

Logdateien

- **Apple Unified Log (ALU)**
  - Das Kommando log erlaubt es, sogenannte **Predicate-Filter** zu nutzen:

Wichtige Predicate-Filter	
Predicate-Filter	Bedeutung
eventMessage contains ´string´	Durchsucht Log-Mitteilungen nach einem bestimmten String. Der String muss in der Ausgabe der Mitteilung enthalten sein.
messageType == error	Sucht nach Mitteilungen des Typs Error.
processID == 100	Sucht nach Mitteilungen einer ProzessID, hier mit PID 100.
subsystem == "com. apple.TimeMachine"	Sucht nach einem konkreten Subsystem/App Bundle, im Beispiel Time Machine.

  - Predicate-Filter können durch logische Operatoren wie &&, and, or, >, <, !=, between, contains, like u. a. miteinander verknüpft werden

(C) 07.07.2023 Hochschule Mittweida

Logdateien

- **Apple Unified Log (ALU)**
  - um die Log-Dateien mit log decodieren zu können, müssen .logarchive-Bundle-Dateien vorliegen oder sich die .traveV3-Dateien im Unified-Logging-Verzeichnis des laufenden Systems befinden.
  - zur Analyse bietet es sich daher entweder an, das zu untersuchende System zu virtualisieren oder die Verzeichnisse /var/db/diagnostics und /var/db/uuidtext auf einen Analyse-Mac-Computer zu übernehmen
  - das log-Kommando gibt seine Ausgaben grundsätzlich auf dem Bildschirm aus, daher ist es sinnvoller, die Ausgabe in eine Textdatei umzuleiten

```
$ touch ~/Desktop/meineLogDatei.txt
$ log show --last 5m --info > ~/Desktop/meineLogDatei.txt
```


(C) 07.07.2023 Hochschule Mittweida

Logdateien

- **Apple Unified Log (ALU)**
  - es existiert derzeit eine Alpha Version eines Offline Datei Parsers:

https://github.com/ydkhatri/**UnifiedLogReader**

```
G:\>c:\Python37-32\python.exe c:\Github\UnifiedLogReader\UnifiedLogReader.py -h
usage: UnifiedLogReader.py [-h] [-f OUTPUT_FORMAT] [-l LOG_LEVEL]
                           uuidtext_path timesync_path tracev3_path
                           output_path

UnifiedLogReader is a tool to read macOS Unified Logging tracev3 files.
This is version 0.3 tested on macOS 10.12.5 - 10.15 and iOS 12.
```


(C) 07.07.2023 Hochschule Mittweida

BETRIEBSSYSTEM macOS

Mac Disk Images

Mac Disk Images

- **Mac OS Disk Image Formate**

– Mac OS kann mit verschiedenen Arten von Disk Images umgehen. Diese beinhalten vollständige Dateisysteme (unterstützt werden HFS+, FAT oder ExFAT):
  - ➢ in einer einzigen Datei (.dmg)
  - ➢ in einer mitwachsenden Datei (.sparseimage)
  - ➢ in einem mitwachsenden Bundle (.sparsebundle).

– DMG-Images haben im Gegensatz zu Sparse Disk Images eine feste Größe

– ein Sparse Disk Image kann seinen Inhalten entsprechend mitwachsen.

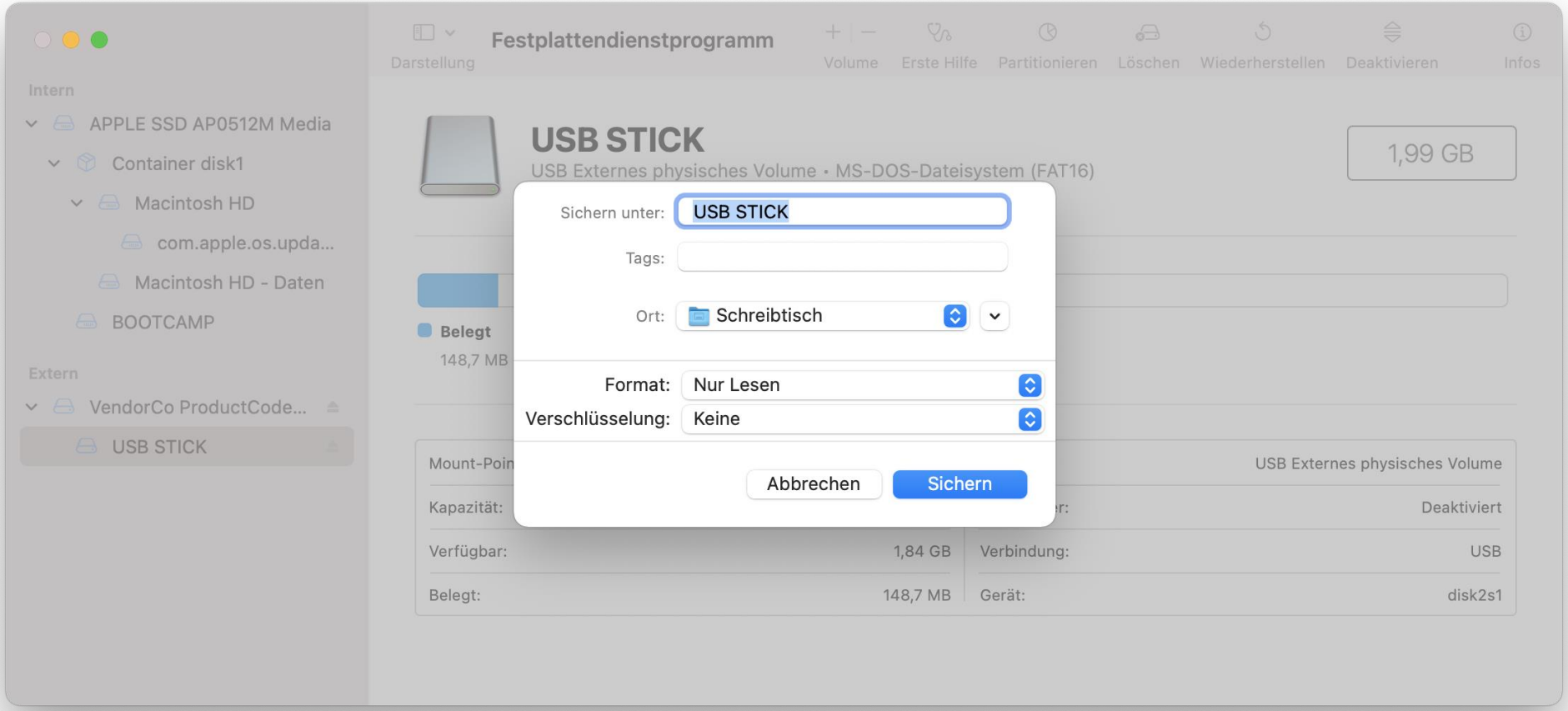
– ein Sparse Bundle wird für den Nutzer transparent in kleinen Einheiten gespeichert, so dass beispielsweise eine inkrementelle Sicherung von Teilen des Images möglich ist

➢ die angegebene Größe von gemounteten Sparse Disk Images oder Sparse Bundles kann aus diesem Grund wesentlich größer sein, als der tatsächlich belegte Speicherplatz auf dem Datenträger

(C) 07.07.2023 Hochschule Mittweida

Mac Disk Images

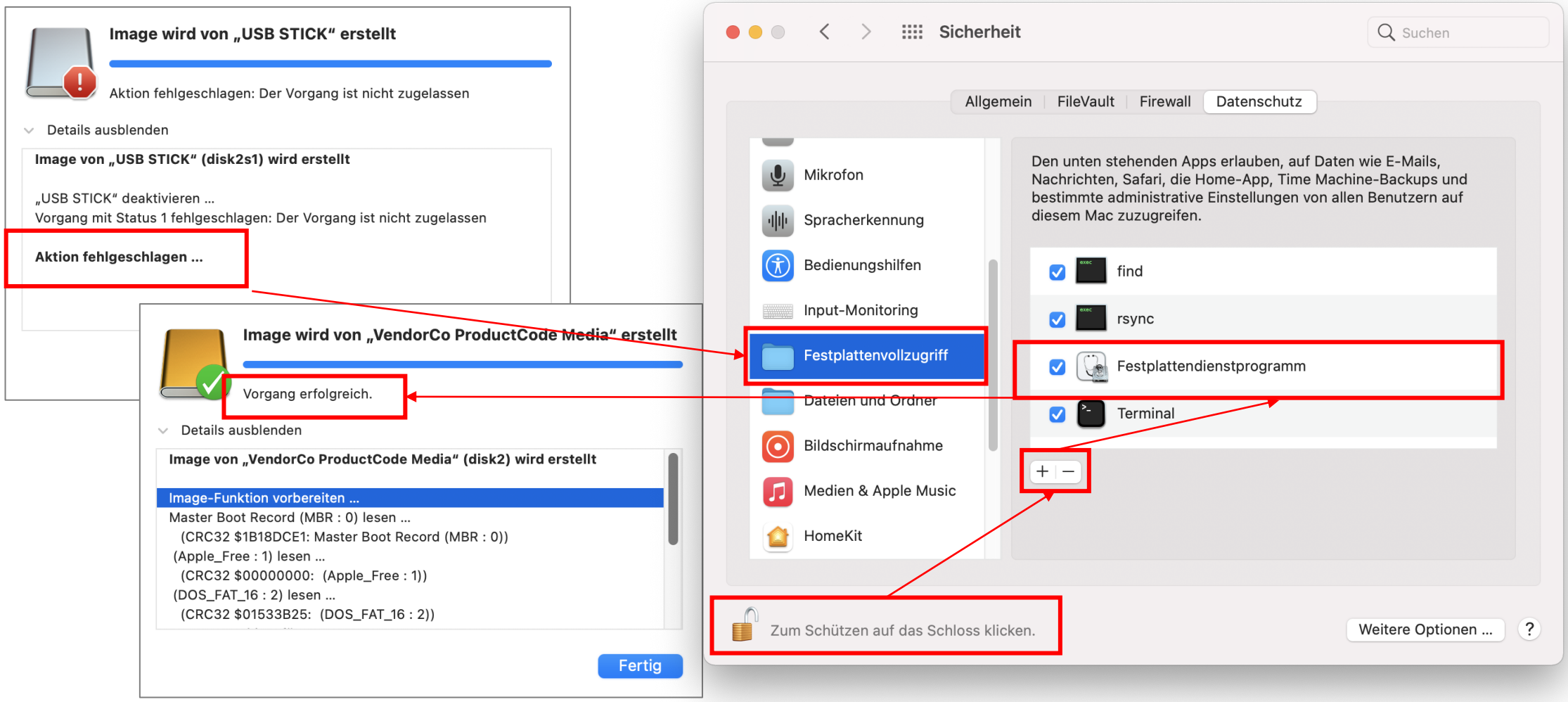
- **Mac OS Disk Image erstellen**



(C) 07.07.2023 Hochschule Mittweida

Mac Disk Images

- **Mac OS Disk Image erstellen**

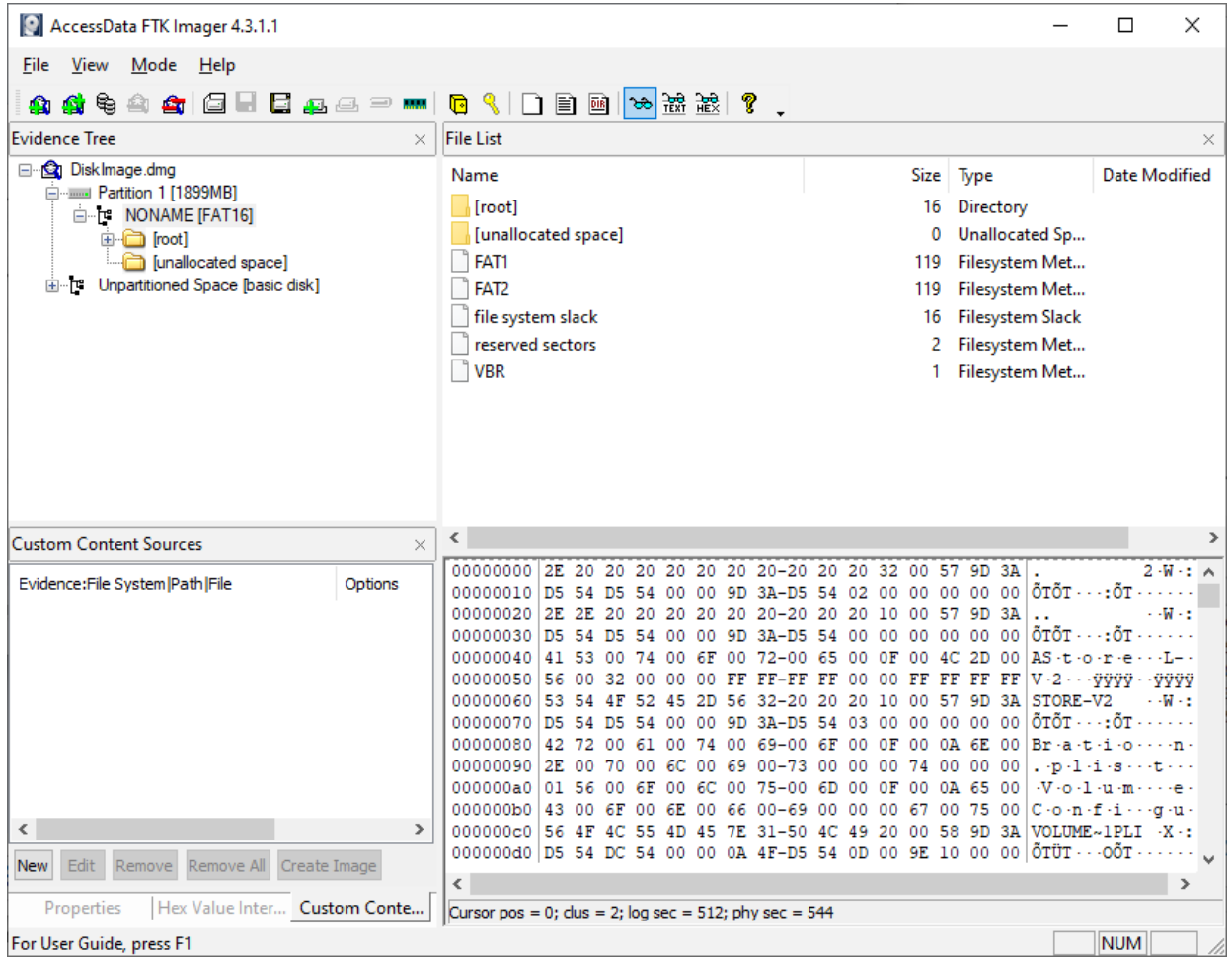


(C) 07.07.2023 Hochschule Mittweida

Mac Disk Images

- **Mac OS Disk Image erstellen**

Das Ergebnis der Sicherung im FTK eingelesen

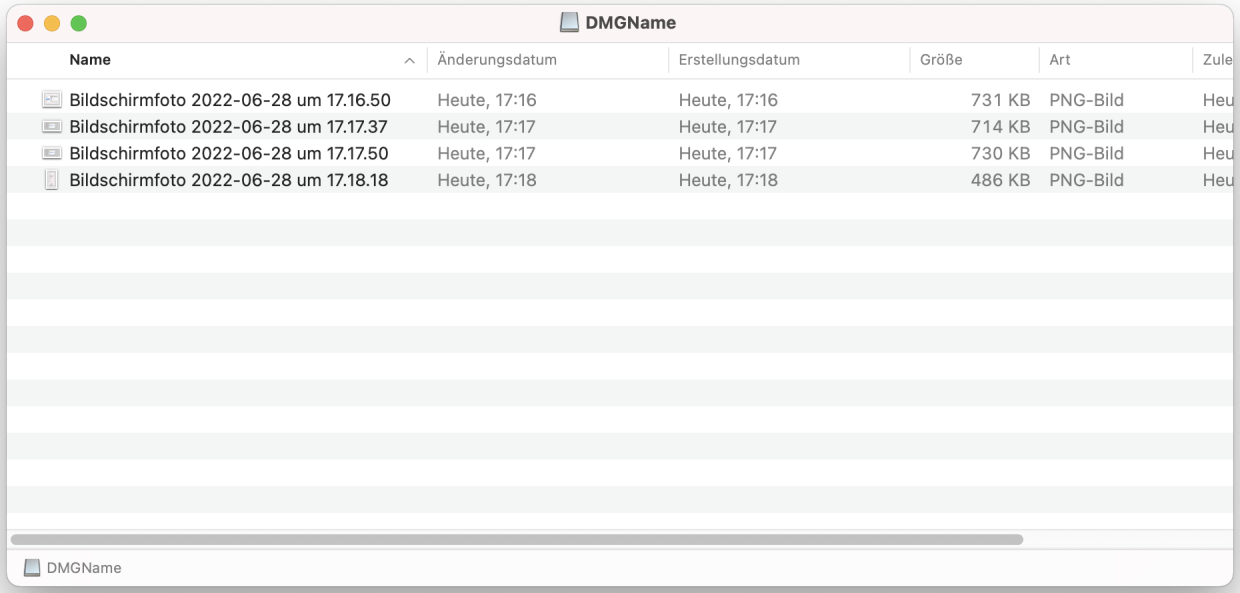


(C) 07.07.2023 Hochschule Mittweida

Mac Disk Images

- **UDIF (Universal Disk Image Format)**

Disk Images nutzen das von Apple entwickelte proprietäres Format UDIF (Universal Disk Image Format). Unter Mac OS können sie bei aktiviertem Disk Arbitration per Doppelklick im Finder gemountet werden.



(C) 07.07.2023 Hochschule Mittweida

Mac Disk Images

UDIF (Universal Disk Image Format)

Der Befehl hdiutil kann genutzt werden, um Informationen zu Disk Images zu erhalten bzw. um sie in das System einzubinden oder auszuhängen:

$ hdiutil info - zeigt Informationen zu gemounteten Disk Images

$ hdiutil attach /Desktop/Disk_Image.dmg - mountet ein Disk Image

$ hdiutil detach /Desktop/Disk_Image.dmg - entfernt ein Disk Image

(C) 07.07.2023 Hochschule Mittweida

Mac Disk Images

- **forensische Abbilder von Mac-Computern**
- Disk Images können auch dazu genutzt werden, forensische Abbilder von Mac-Computern zu erstellen.
- Das DMG-Format wird insbesondere von Produkten, die unter Mac OS lauffähig sind, unterstützt.
- Es bietet den Vorteil, dass die Mac-eigene Technologie Spotlight benutzt werden kann, um Abbilder im DMG-Format zu indexieren und damit einfach und vor allem schnell durchsuchbar zu machen.

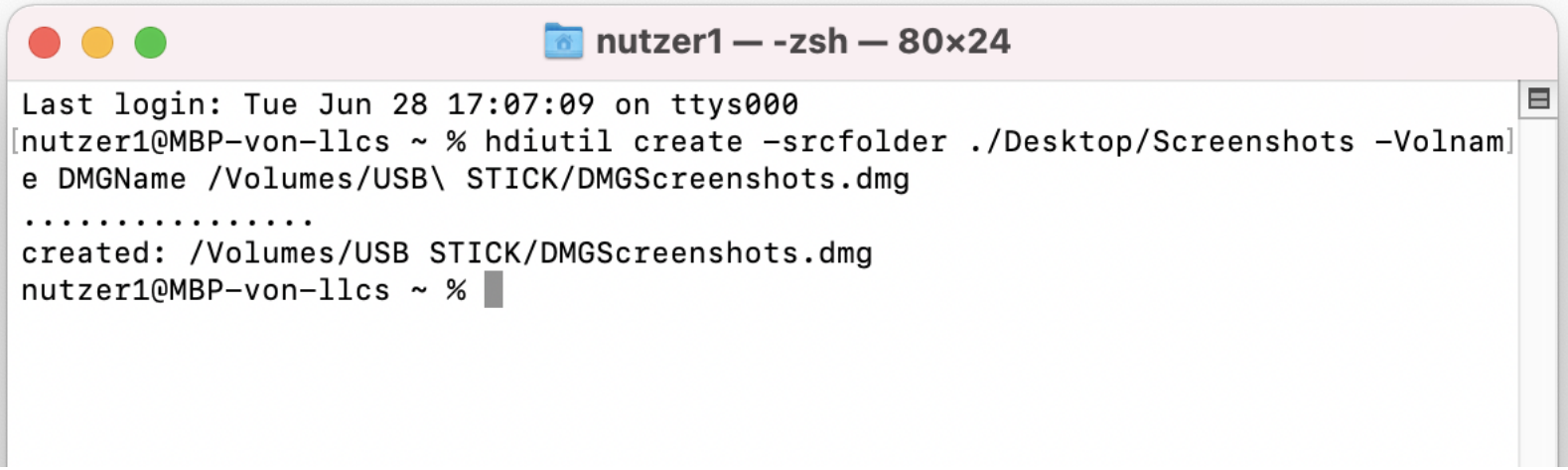
(C) 07.07.2023 Hochschule Mittweida

Mac Disk Images

- **forensische Abbilder von Mac-Computern**

hdiutil create -srcfolder /Users **-volname** DMGNAME /mountpoint/NAME.dmg

– Erzeugt DMG Dateien mit APFS Container Strukturen:



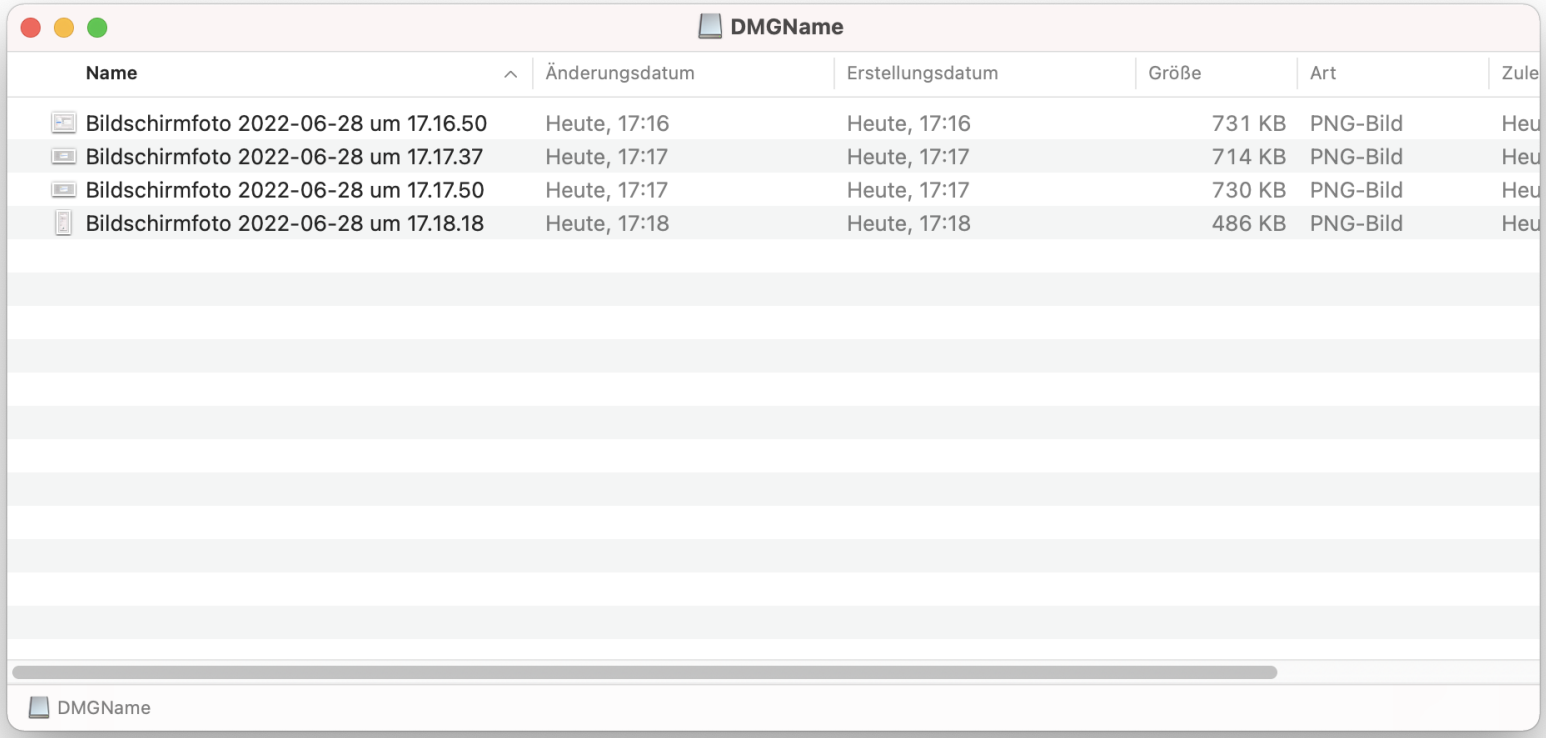
(C) 07.07.2023 Hochschule Mittweida

Mac Disk Images

- **forensische Abbilder von Mac-Computern**

hdiutil create -srcfolder /Users **-volname** DMGNAME /mountpoint/NAME.dmg

– Erzeugt DMG Dateien mit APFS Container Strukturen:



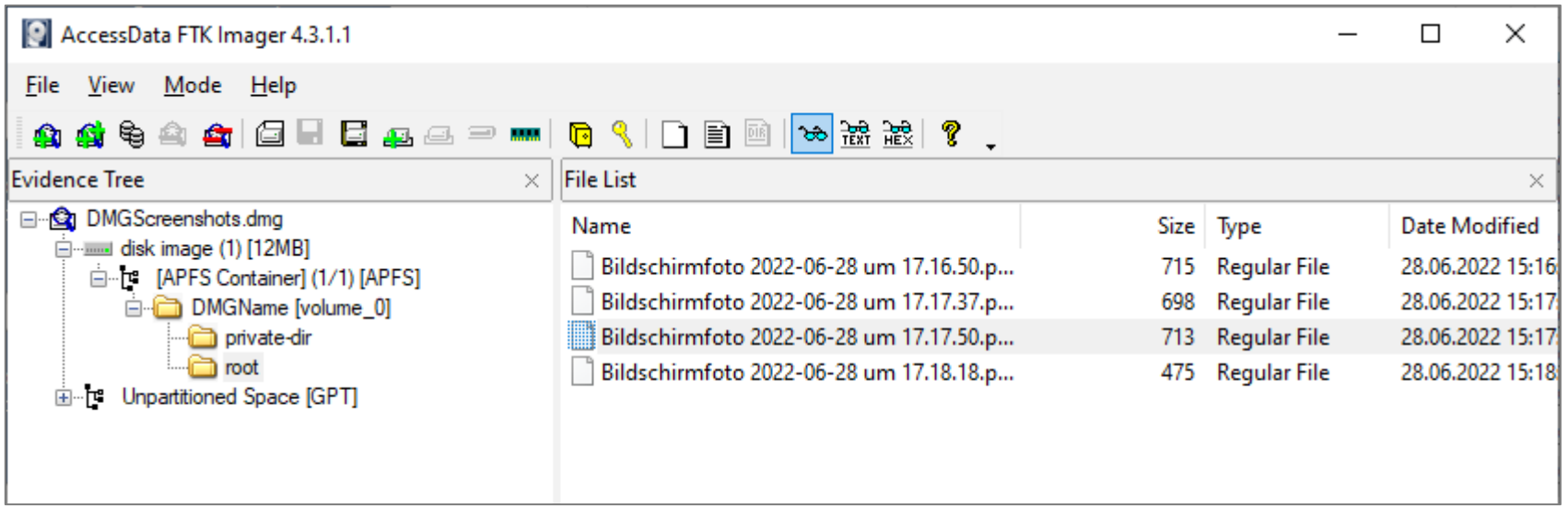
(C) 07.07.2023 Hochschule Mittweida

Mac Disk Images

- **forensische Abbilder von Mac-Computern**

hdiutil create -srcfolder /Users **-volname** DMGNAME /mountpoint/NAME.dmg

– Erzeugt DMG Dateien mit APFS Container Strukturen:



(C) 07.07.2023 Hochschule Mittweida

Mac Disk Images

- Schloss-Icon markiert.

Falls ein forensisches Abbild im DMG-Format eingebunden werden soll, ist es sinnvoll, die Datei schreibgeschützt einzubinden, um Veränderungen auszuschließen.

Das kann erreicht werden, indem man die DMG-Datei im Finder als geschützt markiert. Die Datei wird, wenn geschützt, mit einem kleinen Schloss-Icon markiert.

(C) 07.07.2023 Hochschule Mittweida

Mac Disk Images

- **schreibgeschütztes Mounten von DMG-Dateien**

Damit Spotlight die Datei indexieren kann, wird der Befehl **hdiutil attach** mit der Option **-shadow** benutzt. Dieser Befehl erzeugt eine Shadow-Datei, in der alle von Spotlight durchgeführten Änderungen an der DMG-Datei gespeichert werden.

Spotlight Indexing kann mit

mdutil -i on

aktiviert werden.

Der Befehl

mdutil -s

überprüft, ob Spotlight Indexing aktiv oder nicht aktiv ist.

(C) 07.07.2023 Hochschule Mittweida

Mac Disk Images

- **schreibgeschütztes Mounten von DMG-Dateien**

Neben dem schreibgeschützten Mounten von DMG-Dateien durch Setzen des Hakens *geschützt* ist ein zweiter Weg möglich, der allerdings kein Spotlight Indexing unterstützt.

Hierzu wird das Disk Image zunächst mit dem Befehl

hdiutil attach -nomount

als Block-Device eingebunden und anschließend

mit mount_hfs -j -o rdonly, noexec, noowners

schreibgeschützt gemountet.

(C) 07.07.2023 Hochschule Mittweida

Mac Disk Images

- **Forensische Abbilder mounten**
  - Die Möglichkeit, DMG-Dateien schreibgeschützt einzubinden, kann auch zum Mounten von forensischen Abbildern im RAW- oder E01-Format genutzt werden.
  - Hierzu wird das forensische Abbild mit dem Befehl **xmount** (alternativ **ewfmount**) in eine DMG-Datei konvertiert und anschließend, wie zuvor beschrieben, schreibgeschützt eingebunden.

(C) 07.07.2023 Hochschule Mittweida

Mac Disk Images

- **Forensische Abbilder mounten**

– RAW-Abbilder können neben dem beschriebenen Weg auch direkt in Mac OS eingebunden werden:

$ **hdiutil attach –nomount –imagekey** diskimageclass=CRawDiskImage [Pfad zum RAW Abbild]

– Alternativ können forensische Abbilder unter Mac OS auch mit den grafischen Tools *EWMounter* von Blacklight oder mit *Sumuri Recon* in das System eingebunden und anschließend untersucht werden.

(C) 07.07.2023 Hochschule Mittweida

BETRIEBSSYSTEM macOS

Time Machine und lokale Backups

Time Machine und lokale Backups

- **Time Machine**

– seit der Mac-OS-X-Version 10.5 ist Time Machine die Apple-eigene Technologie zur Erstellung von Backups

– Bis macOS 11 BigSure wurden Time Machine Backups auf ein virtuelles HFS+-Dateisystem auf einem Sparse-Bundle geschrieben

  ➢ Verschlüsselung erfolgt gegebenenfalls im Sparseimage

– MacOS 11 Big Sur ist die erste Version von macOS, die Time Machine-Backups auf APFS-Volumes erstellt, ohne ein virtuelles HFS+-Dateisystem auf einem Sparse-Bundle zu verwenden

  ➢ Verschlüsselung erfolgt im APFS Volume

(C) 07.07.2023 Hochschule Mittweida

Time Machine und lokale Backups

- **Time Machine Backups im Sparsebundle Format**
- Time Machine erstellt in virtuelles HFS+-Dateisystem auf einem Sparse-Bundle
- darin wird in einen Ordner eine Kopie des Dateisystems des zu sichernden Volumes abgelegt
- Dateien und Ordner, die sich seit der letzten Sicherung nicht geändert haben, werden dort durch feste Links dargestellt.
  - Überprüfen man die Inode-Nummer von zwei entsprechenden „Kopien“ eines Ordners, der sich zwischen den Sicherungen nicht geändert hat, und Sieht man identischen Inodes, genau wie bei normalen Datei-Hardlinks

(C) 07.07.2023 Hochschule Mittweida

Time Machine und lokale Backups

- **Time Machine Backups auf APFS Datenträgern**
- APFS unterstützt keine Verzeichnis-Hardlinks und kann daher beim Speichern von Time Machine-Backups nicht denselben Mechanismus verwenden.
- APFS verwendet die aus dem Dateisystem stammenden Snapshots
- Ziellaufwerk muss APFS formatiert werden
- Snapshots werden innerhalb des Dateisystems mit dem Copy on Write Prinzip erstellt
  - ist das Ziellaufwerk nicht verschlüsselt, so ist auch das Backup unverschlüsselt
  - Theoretisch ist mit T2 und M1 eine peer File Verschlüsselung im Backup möglich

(C) 07.07.2023 Hochschule Mittweida

Time Machine und lokale Backups

- **Time Machine Sicherungen**

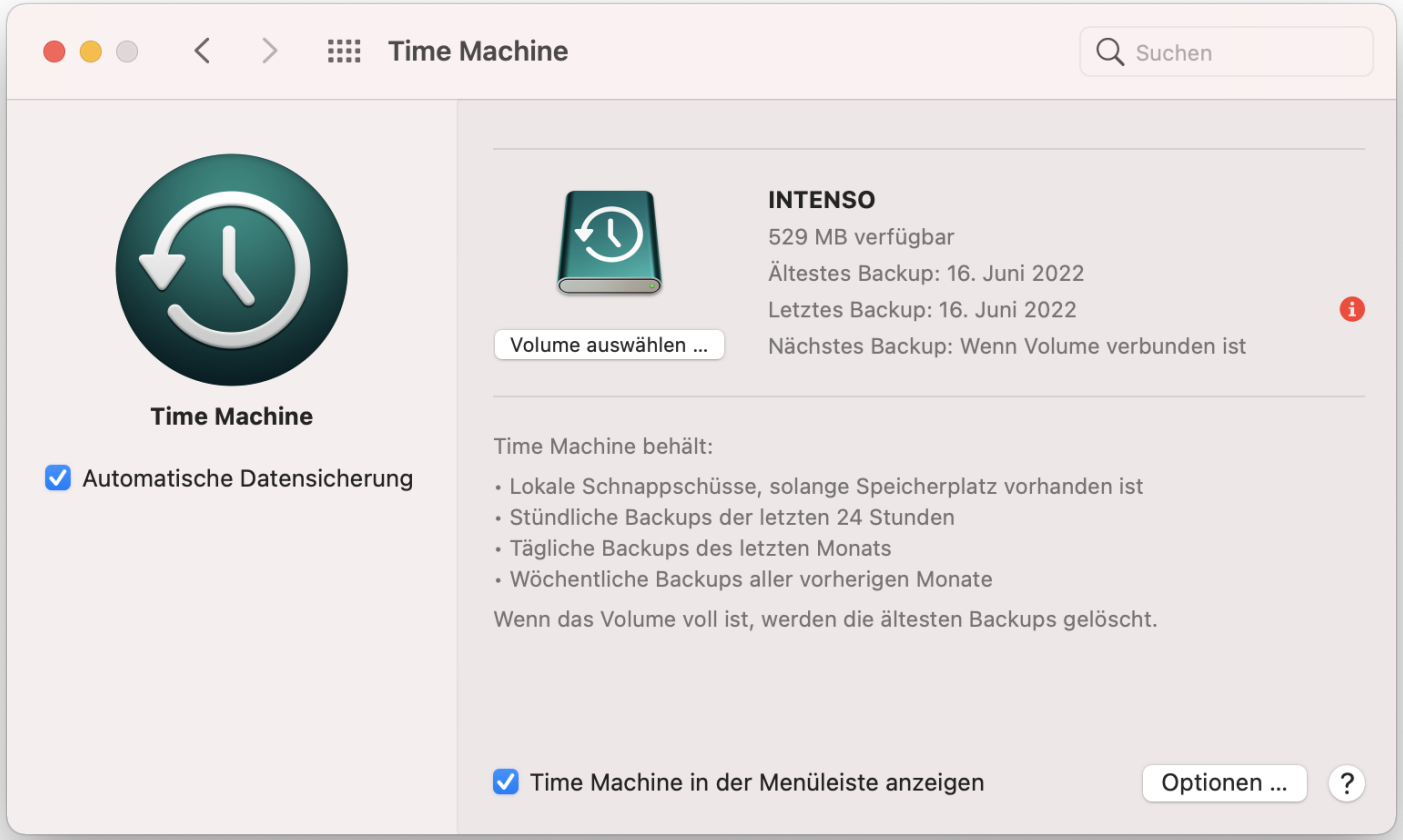
Werden nach einem bestimmten Zeitschema durchgeführt:

- Local Snapshots (Offline), wenn das Time-Machine-Volume nicht präsent ist
- Stündliche Backups werden nach 24 Stunden verworfen
- Tägliche Backups werden nach einem Monat verworfen
- Wöchentliche Backups werden nicht verworfen (Ende, wenn das Time-Machine-Volume voll ist).

(C) 07.07.2023 Hochschule Mittweida

Time Machine und lokale Backups

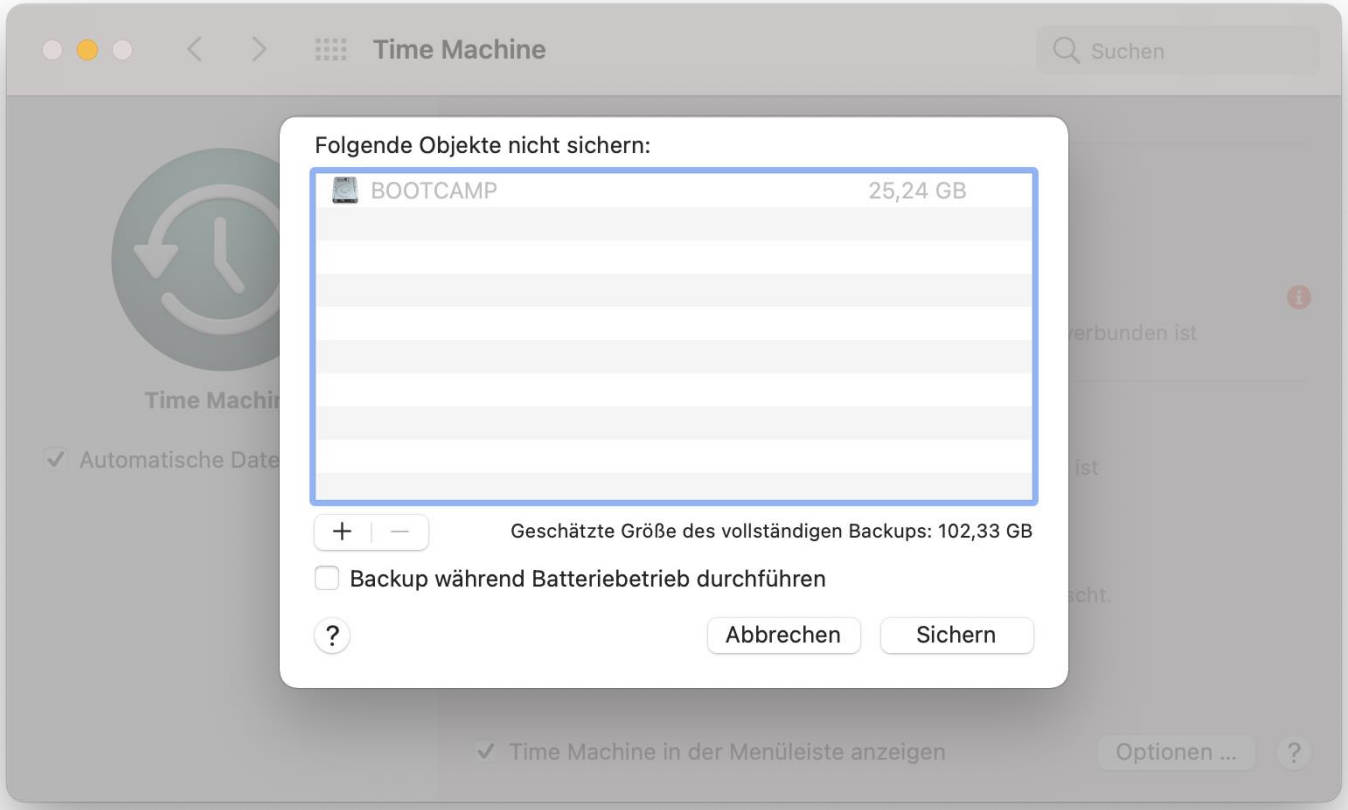
- **TimeMachine**
  - Ansicht letzte Backups in der Anwendung



(C) 07.07.2023 Hochschule Mittweida

Time Machine und lokale Backups

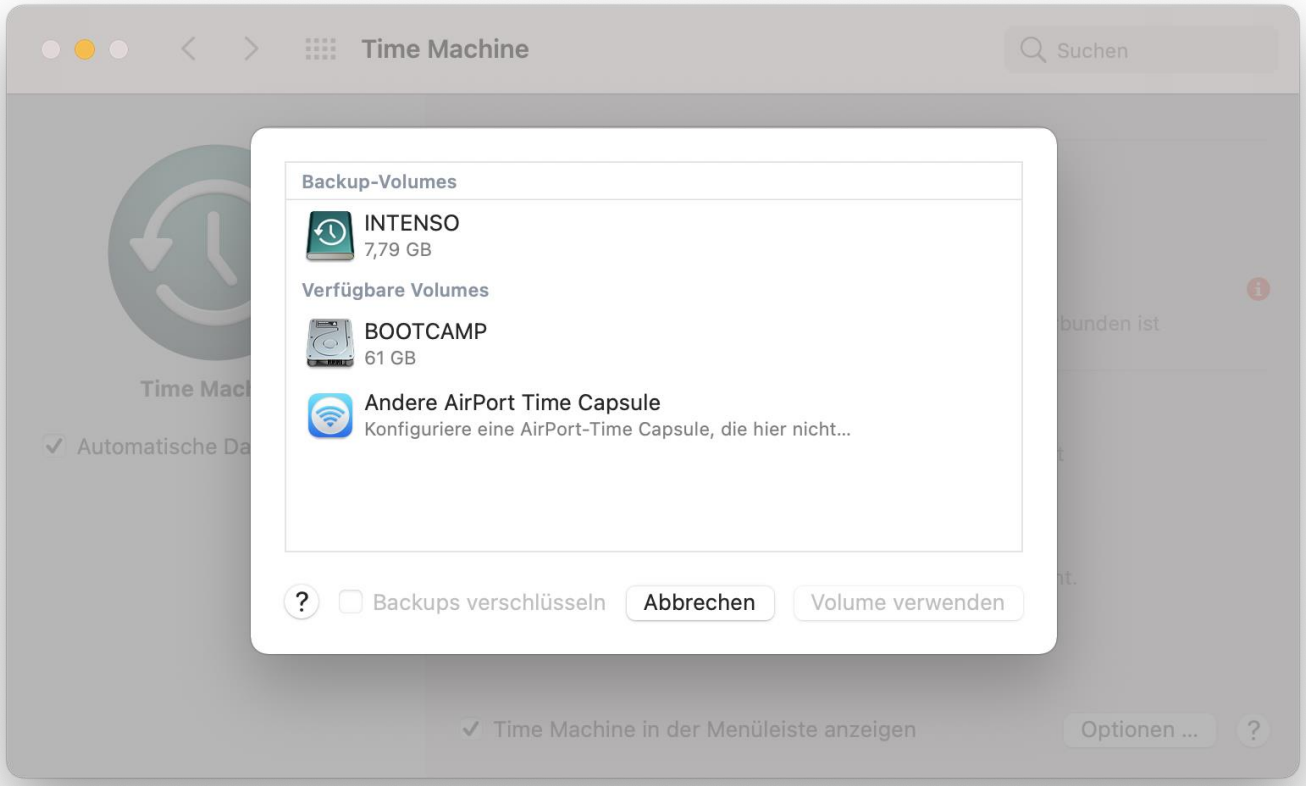
- **TimeMachine**
  - Von der Sciherung ausgenommene Objekteanzeigen in der Anwendung



(C) 07.07.2023 Hochschule Mittweida

Time Machine und lokale Backups

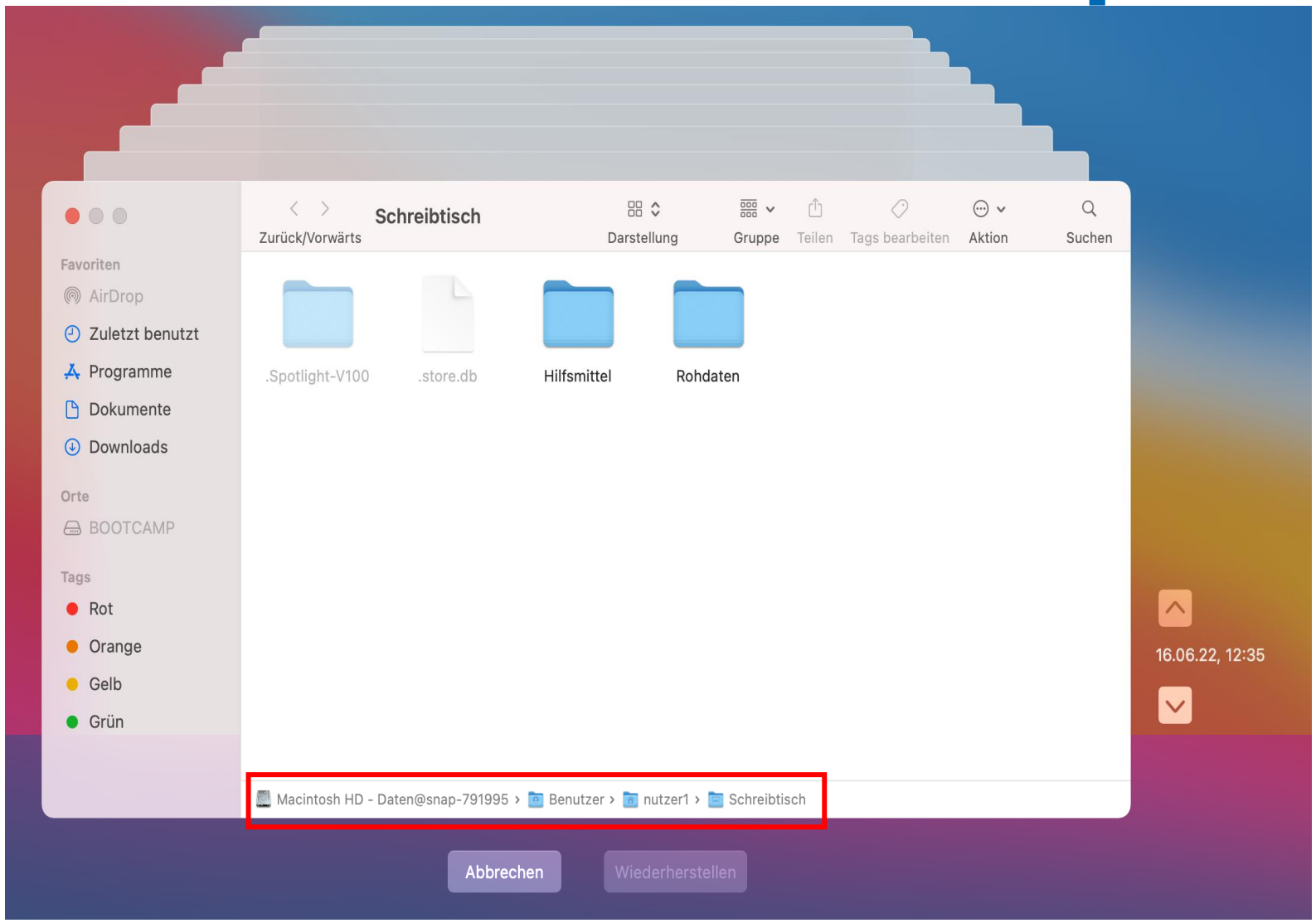
- **TimeMachine**
  - Backuplaufwerk anzeigen in der Anwendung



(C) 07.07.2023 Hochschule Mittweida

Time Machine und lokale Backups

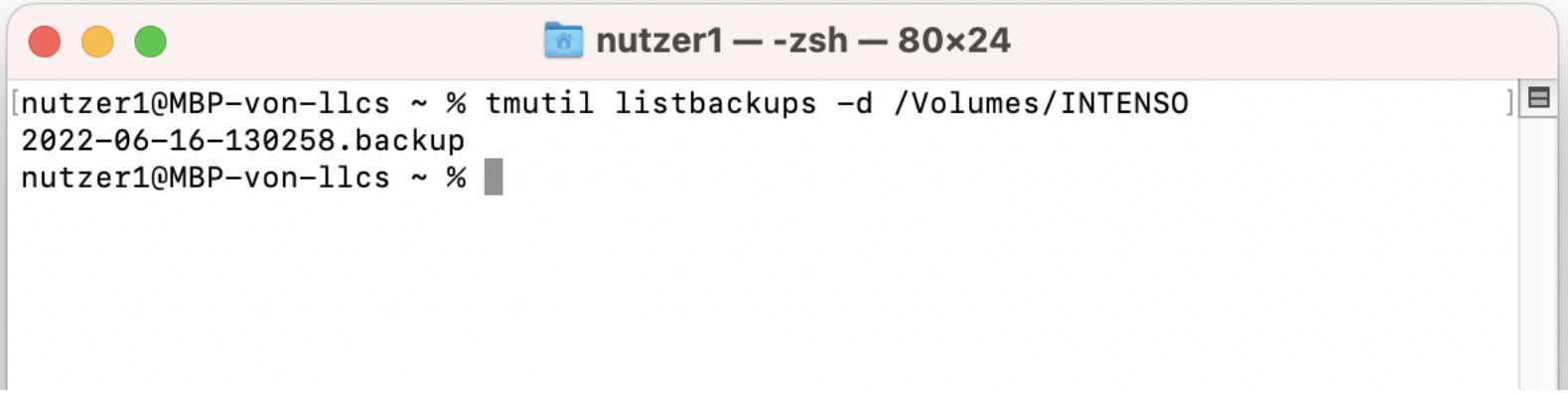
- **TimeMaschine**
  - Vom Finder aufgerufen erhält man die Ansicht der jeweiligen backupstände des geöffneten Verzeichnis



(C) 07.07.2023 Hochschule Mittweida

Time Machine und lokale Backups

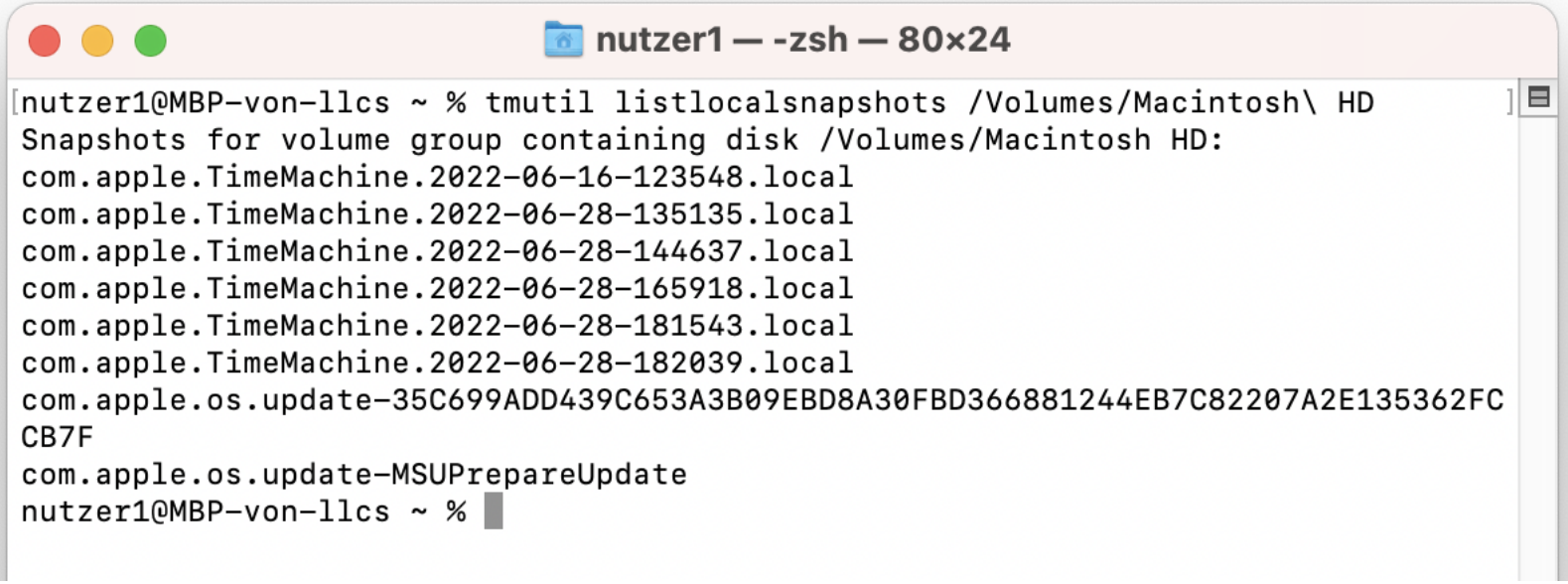
- **tmutil listbackups –d** [/mountpoint/BackupLaufwerk]
  - gibt die Backups im Backuplaufwerk mit Datum aus



(C) 07.07.2023 Hochschule Mittweida

Time Machine und lokale Backups

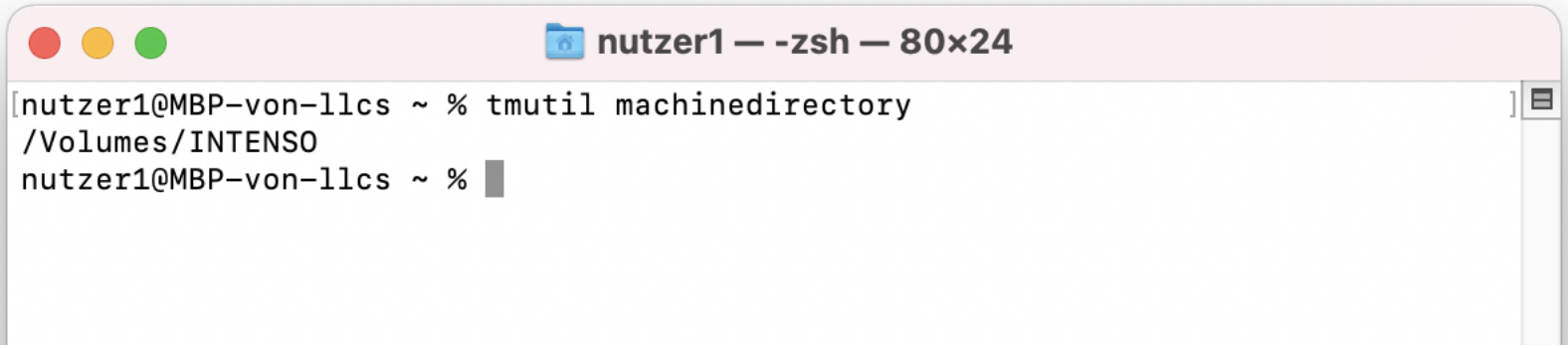
- **tmutil listlocalsnapshots** [/mountpoint]
  - gibt die gespeicherten lokalen Backups aus



(C) 07.07.2023 Hochschule Mittweida

Time Machine und lokale Backups

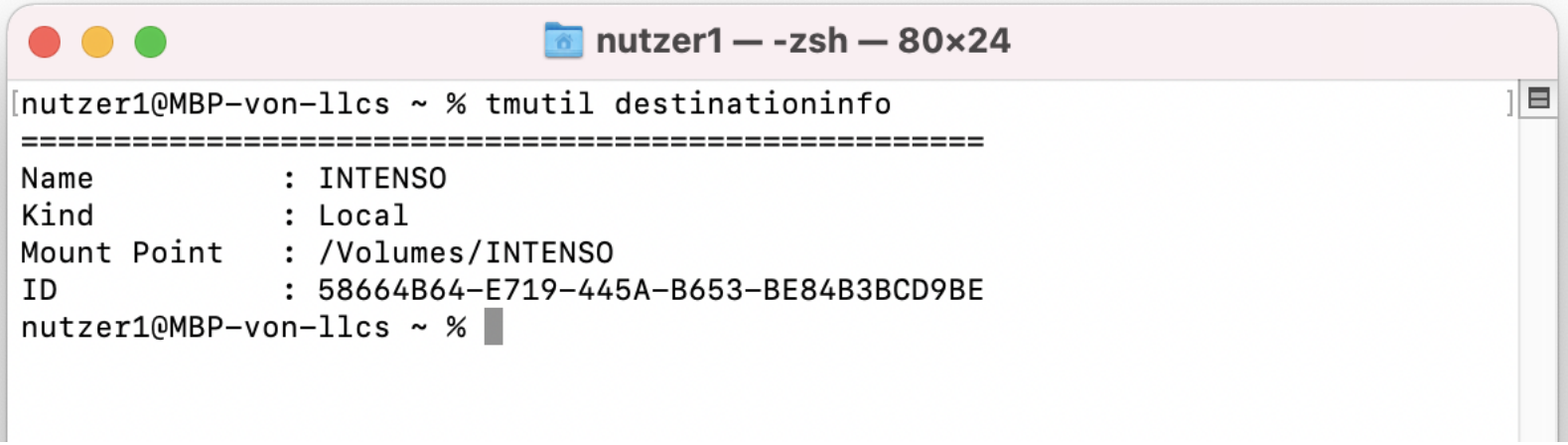
- **tmutil machinedirectory**
  - gibt den Mountpoint der Backups aus



(C) 07.07.2023 Hochschule Mittweida

Time Machine und lokale Backups

- **tmutil destinationinfo**
  - Gibt den Speicherort der Backups aus



(C) 07.07.2023 Hochschule Mittweida

Time Machine und lokale Backups

- Das Backup Laufwerk enthält die einzelnen Backup Stände als Verzeichnisstruktur:

\INTENSO — vor 7 Min.

Name	Pfad	Beschreibung	Erw.	Typ	Größe	Erzeugung
.. = (Stammverzeichnis)		existierend			5,9 GB	
. = INTENSO (176.317)	\	existierend			5,5 GB	16.06.2022 10:35:02
2022-06-21-071648.inprogress (3)	\INTENSO	existierend	inprogress		253 KB	21.06.2022 05:16:49
2022-06-16-183911.interrupted (5)	\INTENSO	existierend	interrupted		196 KB	16.06.2022 16:39:12
2022-06-16-182156.interrupted (160.452)	\INTENSO	existierend	interrupted		3,7 GB	16.06.2022 16:21:57
2022-06-16-174012.interrupted (4.721)	\INTENSO	existierend	interrupted		524 MB	16.06.2022 15:40:13
2022-06-16-130258.previous (11.050)	\INTENSO	existierend	previous		1,1 GB	16.06.2022 10:35:49
.Spotlight-V100 (80)	\INTENSO	existierend			79,5 MB	16.06.2022 10:35:02
private-dir (0)	\INTENSO	existierend			0 B	16.06.2022 10:35:02
backup_manifest.plist (1)	\INTENSO	existierend	plist	plist	0,6 KB	16.06.2022 11:03:19
.DS_Store (1)	\INTENSO	existierend			6,0 KB	16.06.2022 10:35:16
com.apple.fs.cow-exempt-file-count	\INTENSO	existierend	cow-exempt-file-count	cow-exe...	12 B	
Catalog	\INTENSO	virtuell (für Untersuchungszwecke)			76,0 MB	

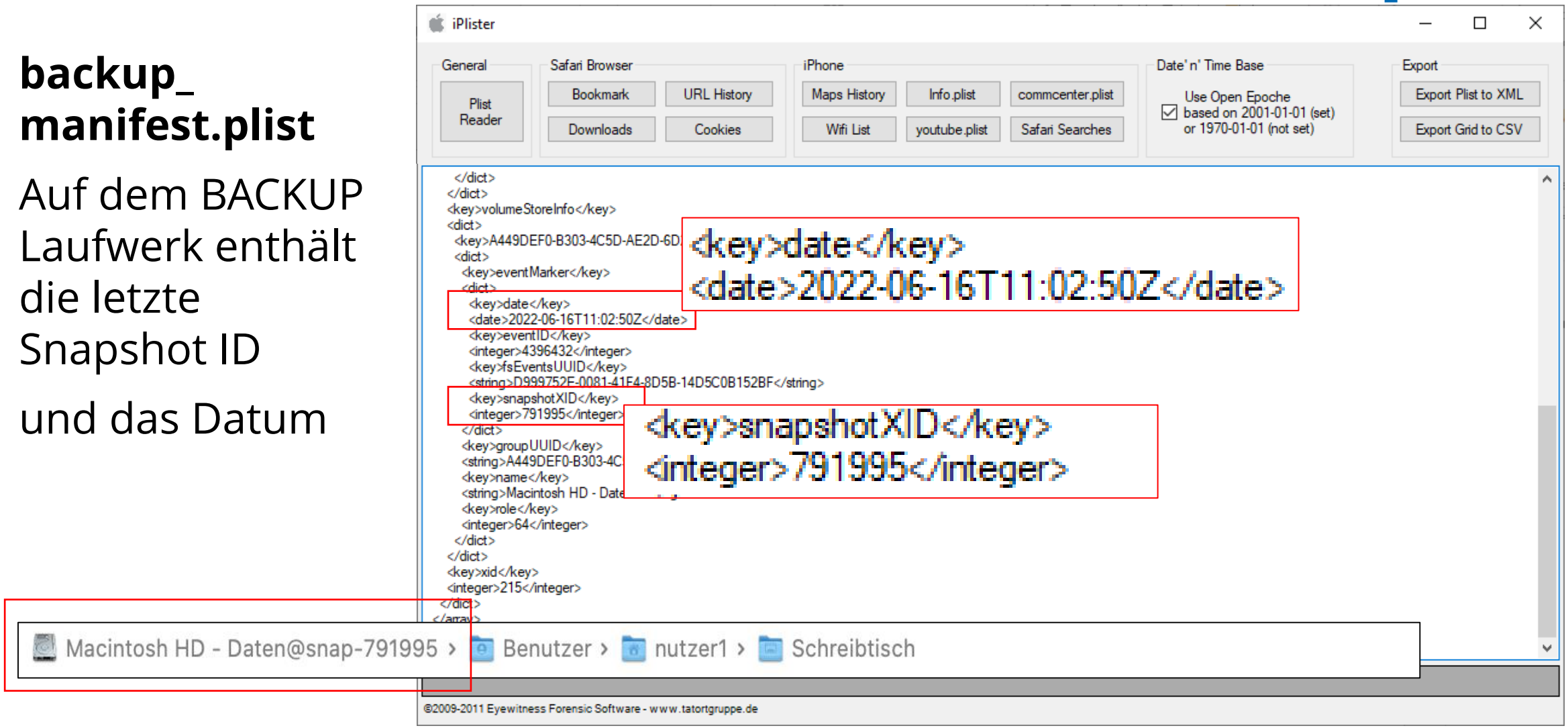
(C) 07.07.2023 Hochschule Mittweida

Time Machine und lokale Backups

backup_ manifest.plist

Auf dem BACKUP Laufwerk enthält die letzte Snapshot ID

und das Datum



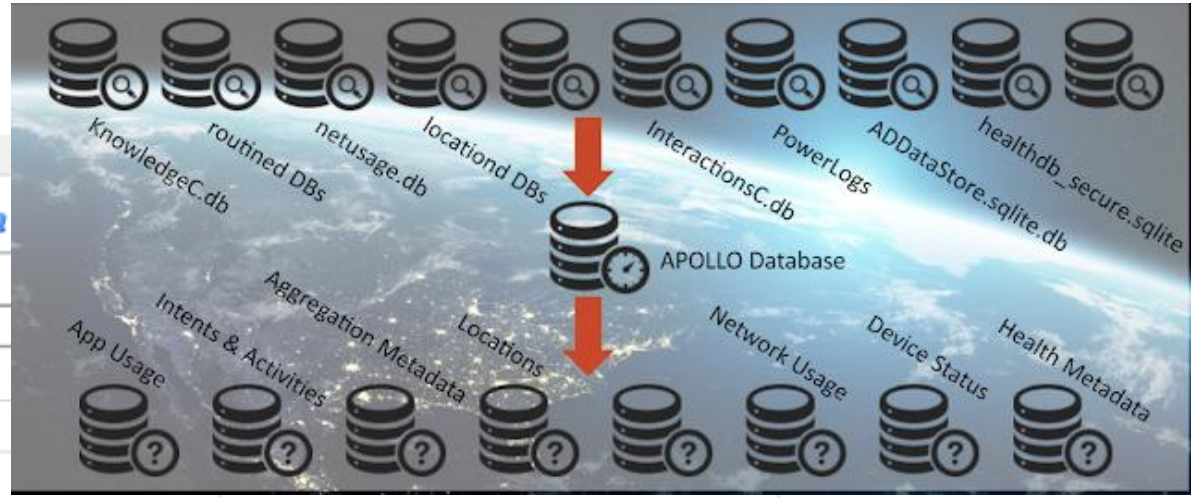
(C) 07.07.2023 Hochschule Mittweida

BETRIEBSSYSTEM macOS

Weitere Artefaktanalysen

Weitere Artefaktanalysen

- **Apple Pattern of Life Lazy Output'er (APOLLO)**
  https://github.com/mac4n6/**APOLLO**



Database Structure | Browse Data | Edit Pragmas | Execute SQL

Table: APOLLO

	Key	Activity	Output		
	Filter	safari	Filter		
1	2020-04-12 17:29:05	Safari Activity	{...		
2	2020-04-12 17:29:15	Safari Activity	{...		
3	2020-04-12 17:30:30	Safari Activity	{...		
4	2020-04-12 17:30:40	Safari Activity	{...	/mnt/w/Practice Images/MacOS/c18-spotlight/...	./modules/...
5	2020-04-12 17:30:45	Safari Activity	{...	/mnt/w/Practice Images/MacOS/c18-spotlight/...	./modules/...
6	2020-04-12 17:30:55	Safari Activity	{...	/mnt/w/Practice Images/MacOS/c18-spotlight/...	./modules/...
7	2020-04-12 17:31:10	Safari Activity	{...	/mnt/w/Practice Images/MacOS/c18-spotlight/...	./modules/...

(C) 07.07.2023 Hochschule Mittweida

Weitere Artefaktanalysen

- **mac_apt - macOS (and iOS) Artifact Parsing Tool**
  https://github.com/ydkhatri/**mac_apt**

```
usage: mac_apt_artifact_only.exe [-h] [-i INPUT_PATH [INPUT_PATH ...]] [-o OUTPUT_PATH] [-x] [-c] [-t]
                                 [-l LOG_LEVEL] [--plugin_help]
                                 plugin

mac_apt is a framework to process macOS forensic artifacts
You are running macOS Artifact Parsing Tool - Artifact Only mode version 1.5.0.dev (20220614)

Note: The default output is now sqlite, no need to specify it now

positional arguments:
  plugin                Plugin to run

optional arguments:
  -h, --help            show this help message and exit
  -i INPUT_PATH [INPUT_PATH ...], --input_path INPUT_PATH [INPUT_PATH ...]
                        Path to input file(s)
  -o OUTPUT_PATH, --output_path OUTPUT_PATH
                        Path where output files will be created
  -x, --xlsx            Save output in excel spreadsheet(s)
  -c, --csv             Save output as CSV files
  -t, --tsv             Save output as TSV files (tab separated)
  -l LOG_LEVEL, --log_level LOG_LEVEL
                        Log levels: INFO, DEBUG, WARNING, ERROR, CRITICAL (Default is INFO)
  --plugin_help         Plugin usage info
```


(C) 07.07.2023 Hochschule Mittweida

Vielen Dank

HOCHSCHULE MITTWEIDA
University of Applied Sciences

Prof. Ronny Bodach

Hochschule Mittweida | University of Applied Sciences
Technikumplatz 17 | 09648 Mittweida
Fakultät Angewandte Computer- und Biowissenschaften

T +49 (0) 3727 58-1011
F +49 (0) 3727 58-21011
@ bodach@hs-mittweida.de
www.cb.hs-mittweida.de

Haus 8 | Richard-Stücklen Bau| Raum 8-205
Am Schwanenteich 6b| 09648 Mittweida

Tim Wetterau B.Sc., Leander Hoßfeld B.Sc.

T +49 (0) 3727 58-1752
+49 (0) 3727 58-1752
@ wetterau@hs-mittweida.de
hossfeld@hs-mittweida.de

Haus 6 | Grunert de Jacome Bau | Raum 6-031
Am Schwanenteich 4b | 09648 Mittweida

hs-mittweida.de