



Betriebssysteme

Praktikum macOS

Inhaltsverzeichnis

1. Aufgabenstellung Vorbereitung	3
2. Kennenlernen von macOS aus Bedienersicht	7
3. Untersuchung der macOS Artefakte und digitaler Spuren in externem Betriebssystem	21
3.1. Spotlight Untersuchung mit Python Parser	22
3.2. FSEvent Untersuchung mit FSEvent Parser	27
3.3. Untersuchung der Safari History mit SQLite	31
3.4. Untersuchung der Datenbankdatei KnowledgeC.db mit SQLite	37
4. Erstellung einer macOS Lab VM in VirtualBox (unter Windows)	43
5. Deaktivierung der SIP in VirtualBox	51

1. AUFGABENSTELLUNG VORBEREITUNG

In diesem Praktikum lernen Sie die Einrichtung einer macOS VM kennen. Die Einrichtung dieses Betriebssystems ist durch verschiedene Kniffe nicht trivial und erfordert einige Umwege und besondere Maßnahmen. Zusätzlich sollen Sie erste Schritte in dem Betriebssystem machen und dessen Oberfläche kennenlernen. Auch in diesem Praktikum verwenden wir die Virtualisierungssoftware Oracle VirtualBox, welche Sie schon aus den vorigen Praktika kennen sollten.

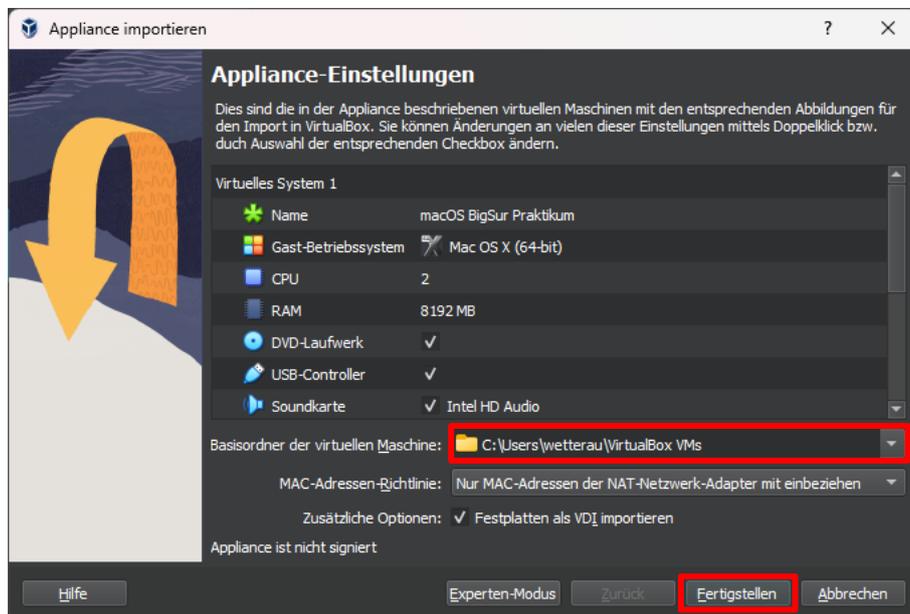
Zur Durchführung des Praktikums nutzen wir das auf den PCs der HSMW installierte Betriebssystem Windows. Weiterhin benötigen wir einige Dateien, welche Ihnen im Laufwerk R: unter folgendem Pfad zur Verfügung stehen:

R:\CB\Bodach\BKA Studiengang\Betriebssysteme\Praktikum Blockwochen\macOS\

Laden Sie sich aus diesem Verzeichnis für dieses Praktikum die Dateien „macOS BigSur Praktikum Installiert.ova“ und „macOS BigSur Praktikum\macOS BigSur Praktikum.nvram“ herunter und speichern Sie diese Dateien an einen geeigneten Ort auf Ihrem lokalen System. (An den PC der HSMW optimalerweise unter D:\)

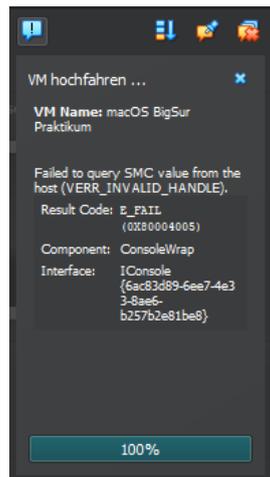
Importieren der OVA-Datei

Nachdem Sie alle Dateien heruntergeladen haben, starten Sie die vorbereitete virtuelle Maschine mit einem Doppelklick auf die OVA-Datei. Daraufhin öffnet sich VirtualBox und fragt nach der Importierung der virtuellen Maschine. Stellen Sie bitte den Basisordner der Datei unten auf einen von Ihnen gewählten Ordner auf Laufwerk



D:\ um und bestätigen Sie den Dialog mit dem Button „Fertigstellen“.

Danach wird Ihnen die VM importiert, wie Sie es bereits kennen. Sollten Sie nun versuchen die VM ohne Weiteres zu starten, werden Sie höchstwahrscheinlich einen Fehler in VirtualBox bekommen:



Sollte das bei Ihnen nicht der Fall sein, dann haben Sie Glück und können mit dem Kennenlernen der Oberfläche fortfahren. Ansonsten gehen Sie das Praktikum weiter durch und versuchen die Fehler, wie beschrieben zu behandeln. Ein weiterer Fehler kann sein, dass die im EFI-Menü gefangen sind. Dann gehen Sie auch weiter.

Zusätzliche Maßnahmen vornehmen (Fehlerbehandlung)

Damit das Gastbetriebssystem reibungslos auf den Hostsystem laufen kann, müssen noch einige Einstellungen getätigt werden, wenn dies bis hierher nicht der Fall ist. Dazu starten Sie bitte mit einem Rechtsklick auf den Windowsstartbutton eine neue PowerShell (Terminal unter Windows 11). Anschließend geben Sie bitte den folgenden Befehl in die die Befehlszeile ein:

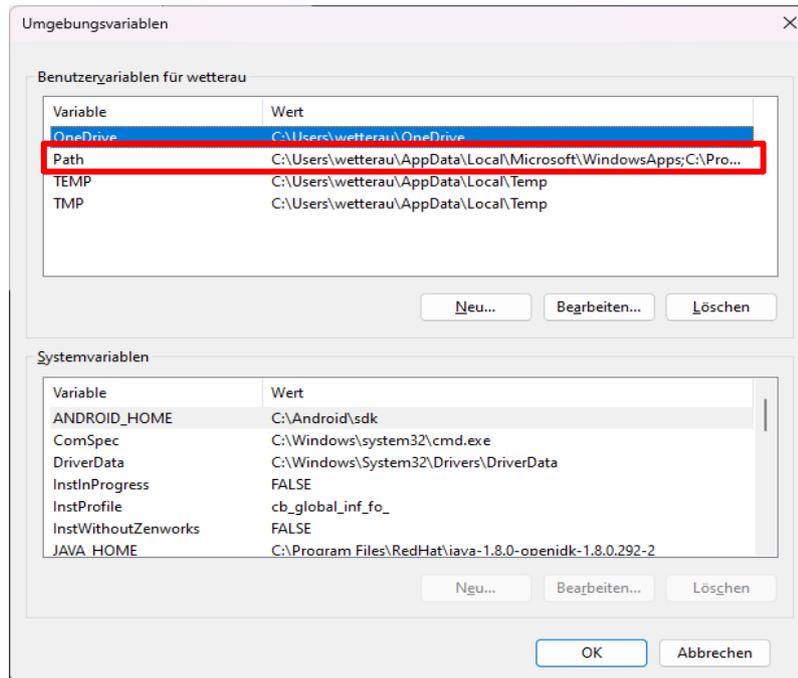
```
$ VBoxManage setextradata "macOS BigSur Praktikum"  
"VBoxInternal/Devices/smc/0/Config/GetKeyFromRealSMC" 0
```

Sollte dieser Befehl bei Ihnen einen Fehler erzeugen, dass der Befehl VBoxManage nicht als Cmdlet registriert ist, fahren Sie bitte mit dem Hinzufügen von VirtualBox zu den Umgebungsvariablen fort (nächstes Kapitel).

Sollten Sie im EFI Bios gefangen sein und macOS nicht starten, so kopieren Sie anschließend die *.nvram-Datei in das Verzeichnis, in dem Sie die VM abgelegt haben, dann sollte dieser Fehler behoben sein.

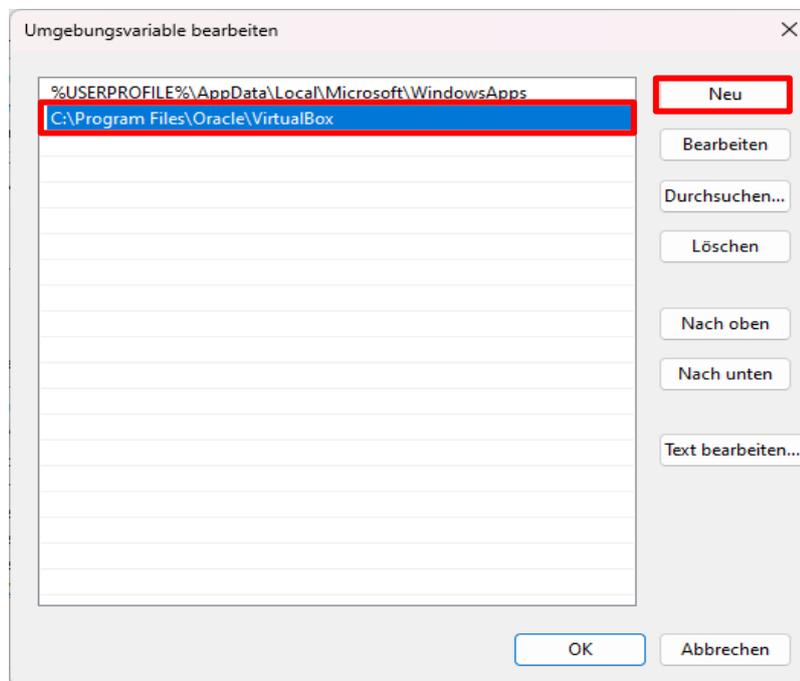
VirtualBox-Pfad zu den Umgebungsvariablen hinzufügen (Optional)

Sollte bei Ihnen der Befehl mit VBoxManage nicht funktionieren, kann eine Abhilfe geschaffen werden, indem Sie den Pfad zu der ausführbaren Datei in die Systemvariable PATH einfügen. Dazu drücken Sie bitte den Windows-



Button und geben den Begriff Umgebungsvariablen ein und wählen den Ergebniseintrag „Umgebungsvariablen für dieses Konto bearbeiten“ aus. Im Folgenden sich öffnenden Fenster klicken Sie bitte doppelt auf die Variable PATH im oberen Fenster.

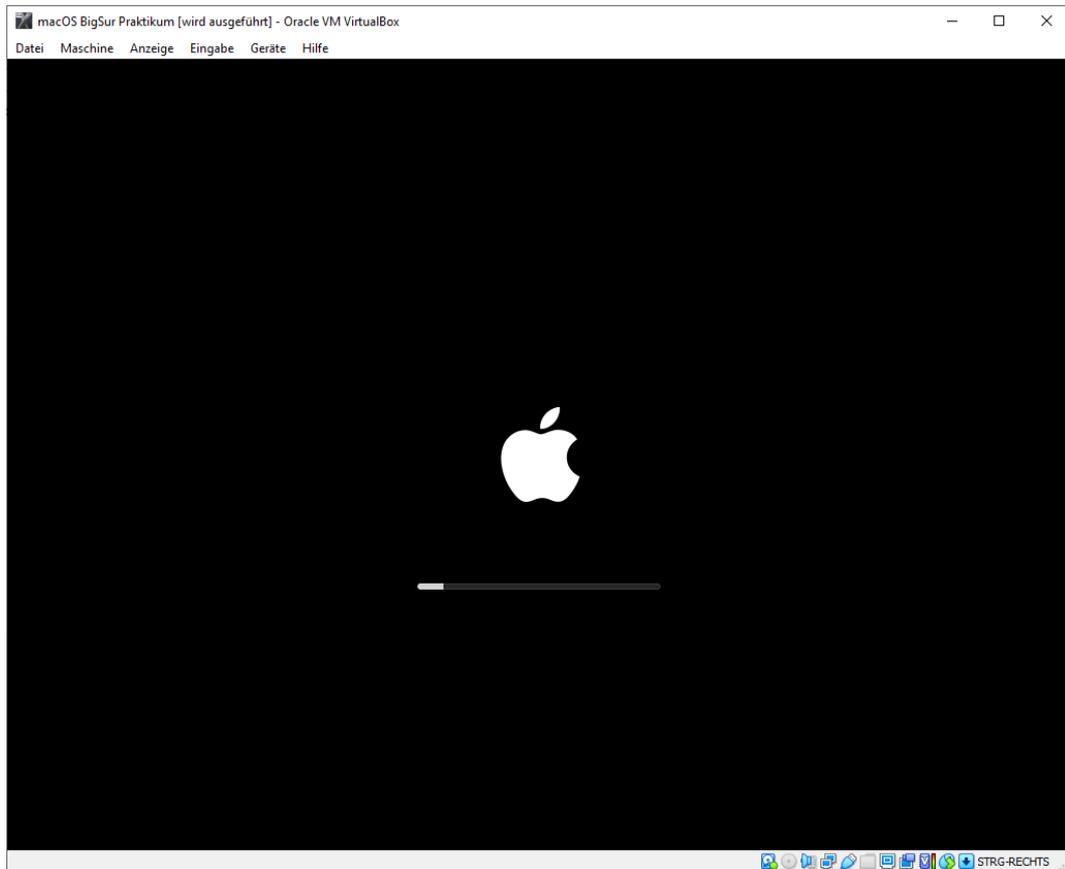
Im sich nun öffnenden Fenster klicken Sie bitte auf „Neu“ und fügen den Pfad „C:\Program Files\Oracle\VirtualBox“ ein:



Anschließend klicken Sie bitte auf „OK“, um Ihre Änderungen zu bestätigen. Nun sollten Sie das Kommando VBoxManage, wie oben beschrieben ausführen können.

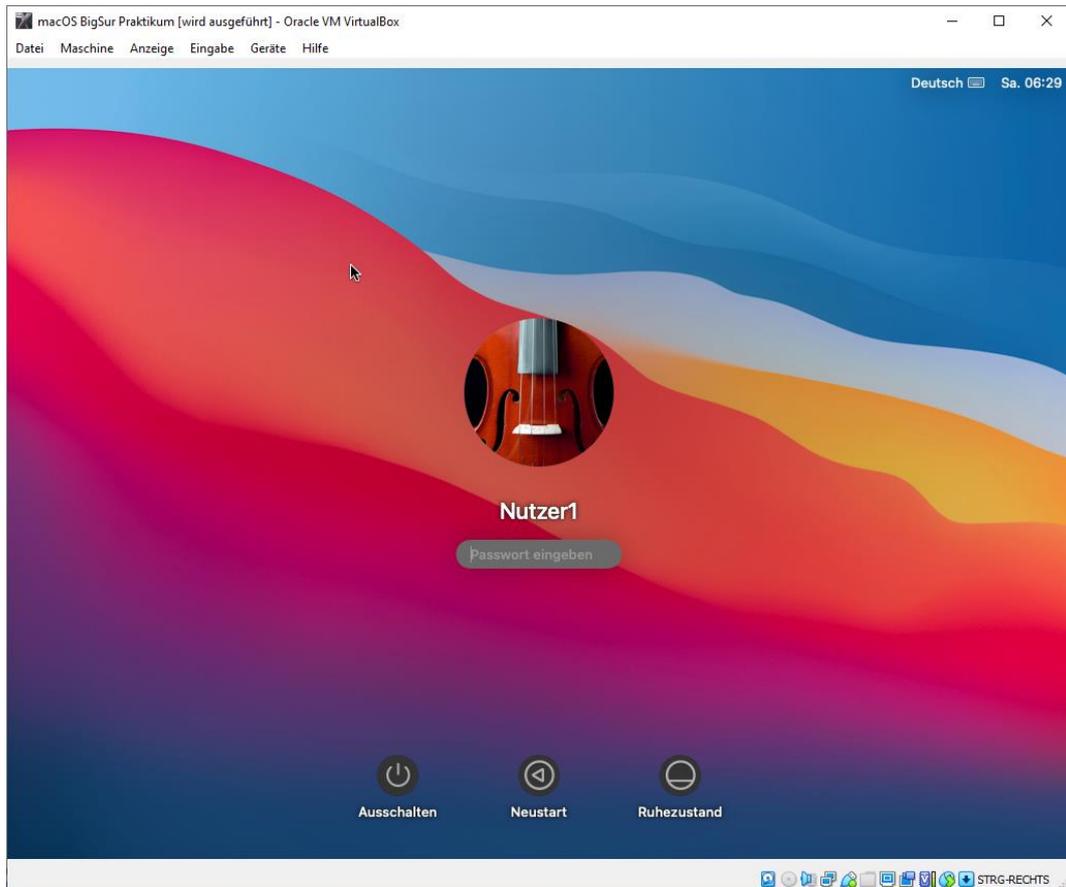
Starten der VM zum weiteren arbeiten

Starten Sie die **macOS Big Sure Praktikum** VM durch Doppelklick oder Starten.

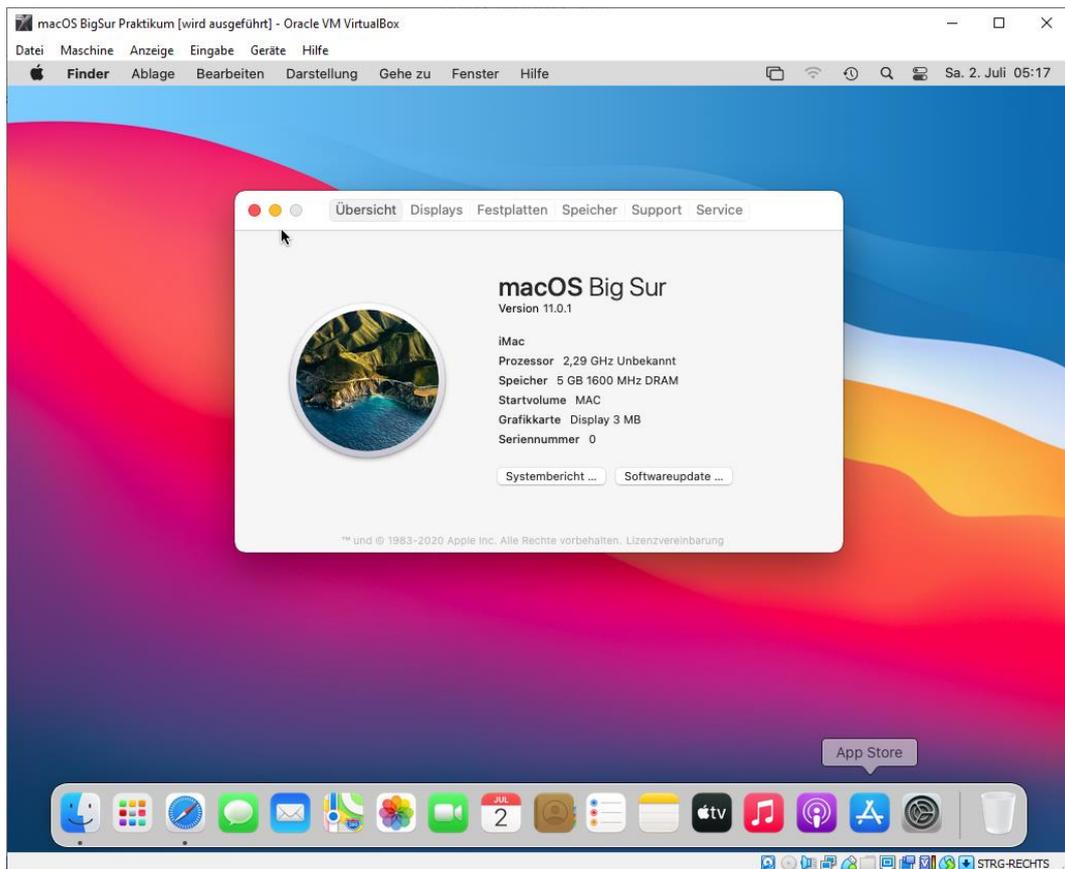
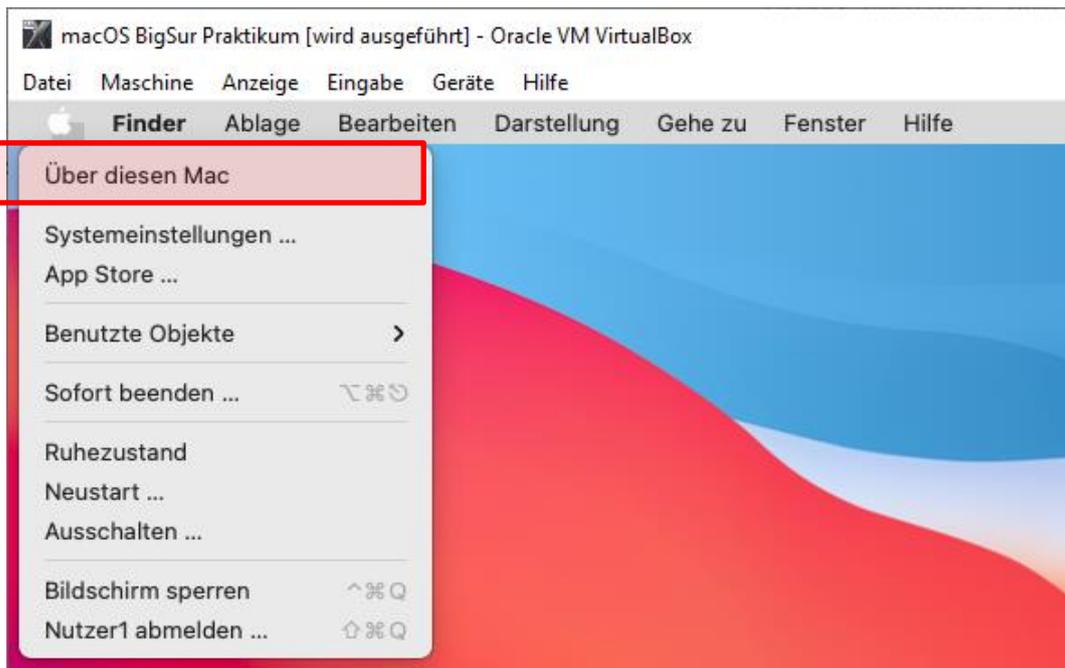


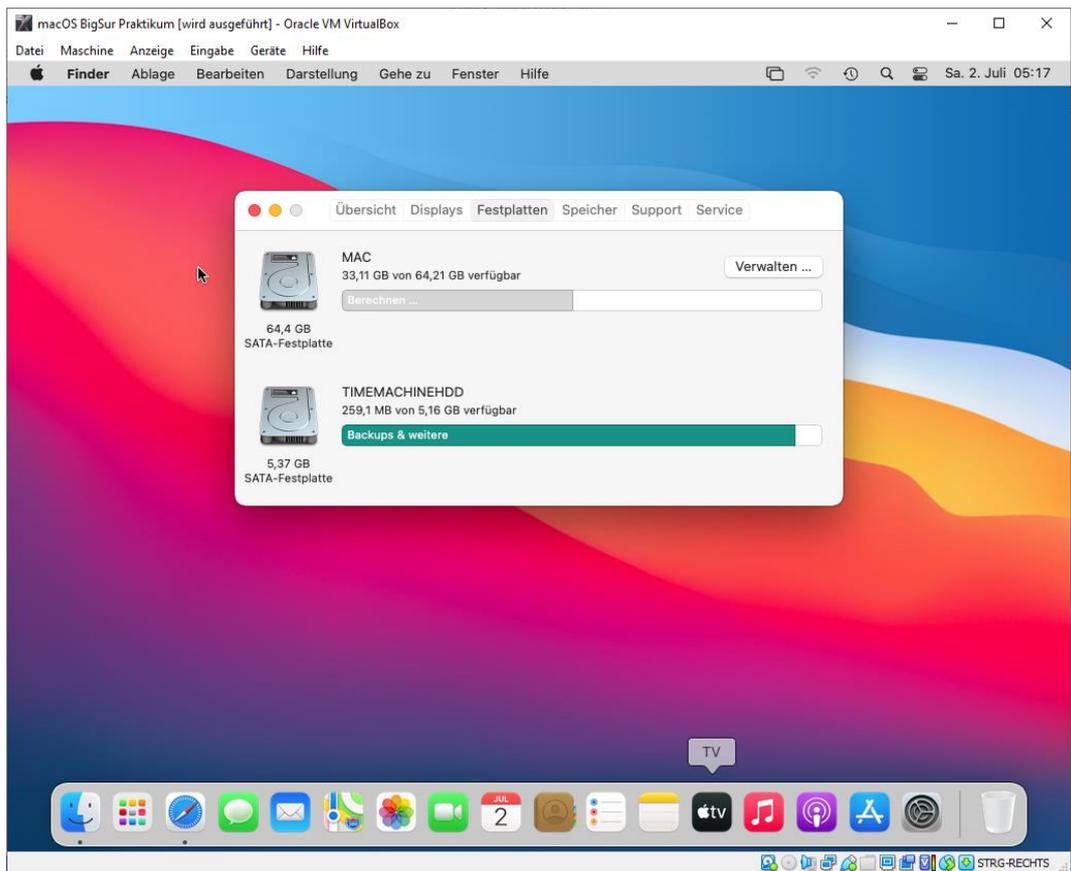
2. KENNENLERNEN VON MACOS AUS BEDIENERSICHT

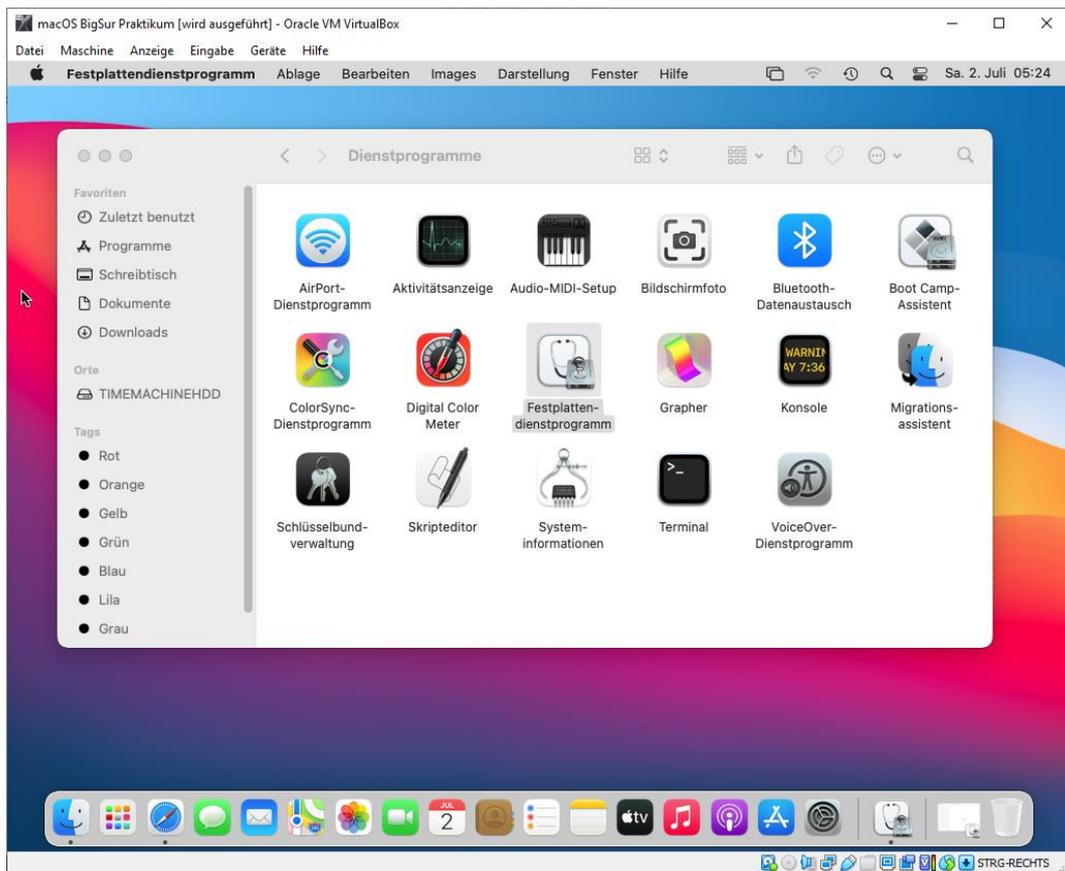
Melden Sie sich als **Nutzer1** mit **Kennwort1** an.



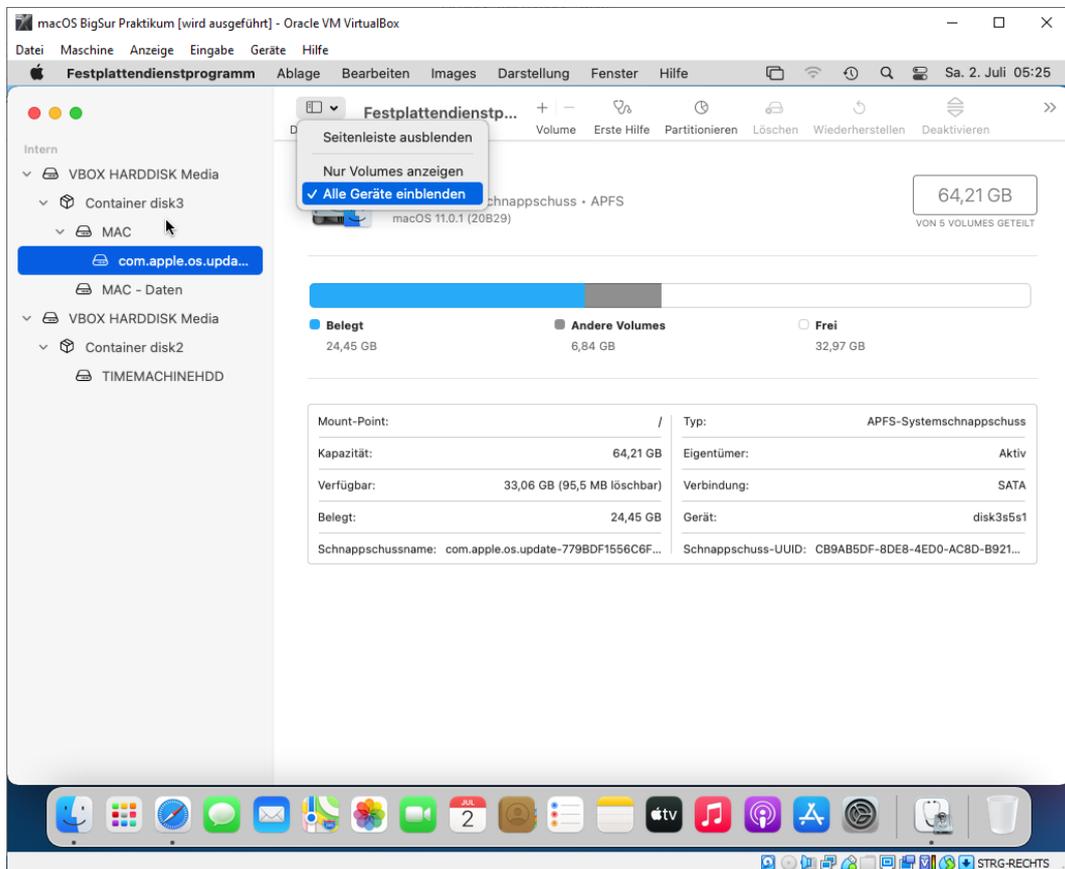
Rufen Sie **Über diesen Mac** auf und schauen Sie sich an, um was für ein System es sich handelt und welches Startvolumen eingebunden ist.







Öffnen Sie das **Festplattendienstprogramm**. Überprüfen Sie ob die Ansicht auf **Alle Geräte anzeigen** eingestellt ist.

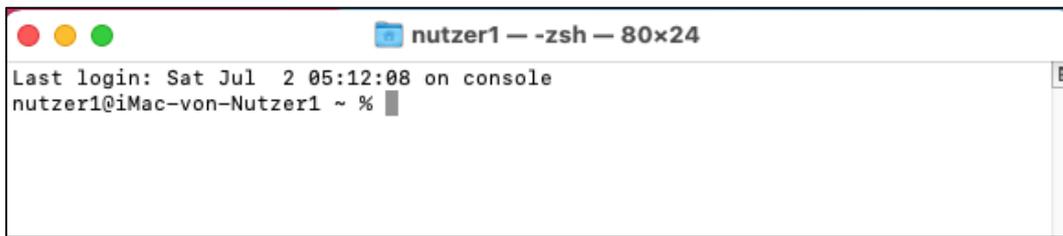


Wählen Sie den Container **disk3** aus und schauen Sie sich die Übersicht der tatsächlich vorhandenen Volumes an.

Öffnen Sie das **Terminal** unter **Finder > Programme > Dienstprogramme**.

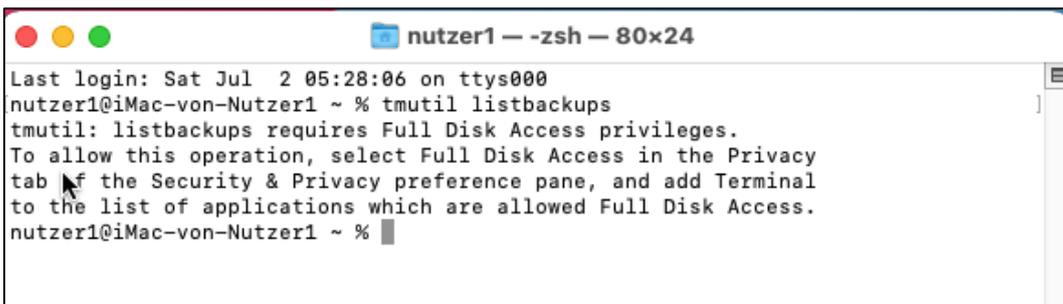
Schließen Sie das Terminal Fenster und öffnen Sie **Spotlight** über die **Menüleiste**.

Geben Sie hier **Terminal** ein und öffnen Sie das Terminal Fenster erneut.



```
nutzer1 — -zsh — 80x24
Last login: Sat Jul 2 05:12:08 on console
nutzer1@iMac-von-Nutzer1 ~ %
```

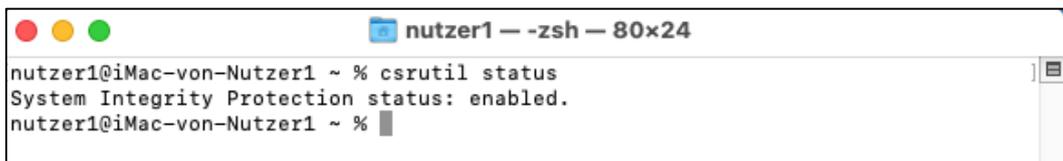
Listen Sie sich alle Backups auf.



```
nutzer1 — -zsh — 80x24
Last login: Sat Jul 2 05:28:06 on ttys000
nutzer1@iMac-von-Nutzer1 ~ % tutil listbackups
tutil: listbackups requires Full Disk Access privileges.
To allow this operation, select Full Disk Access in the Privacy
tab of the Security & Privacy preference pane, and add Terminal
to the list of applications which are allowed Full Disk Access.
nutzer1@iMac-von-Nutzer1 ~ %
```

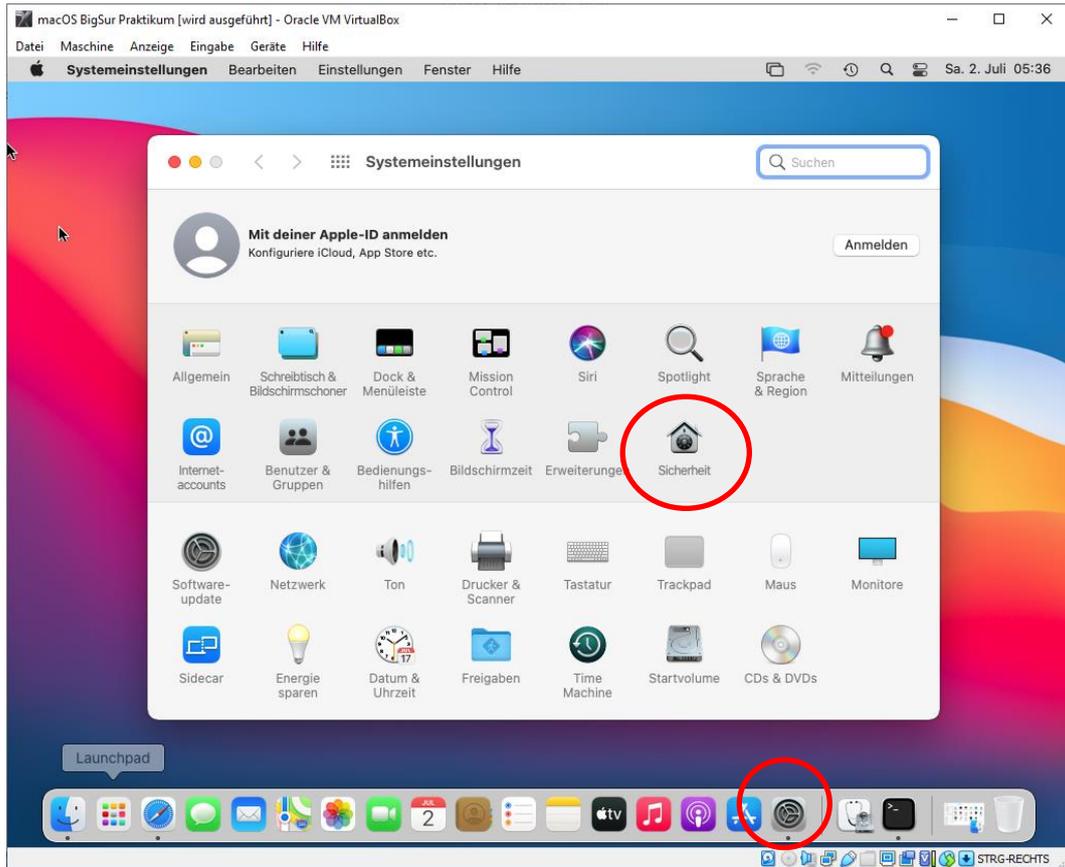
Hier fehlen ihnen die Berechtigungen, da die System Integrity Protection SIP einiges blockiert.

Überprüfen Sie dies mit dem Befehl **csrutil status** im Terminal.

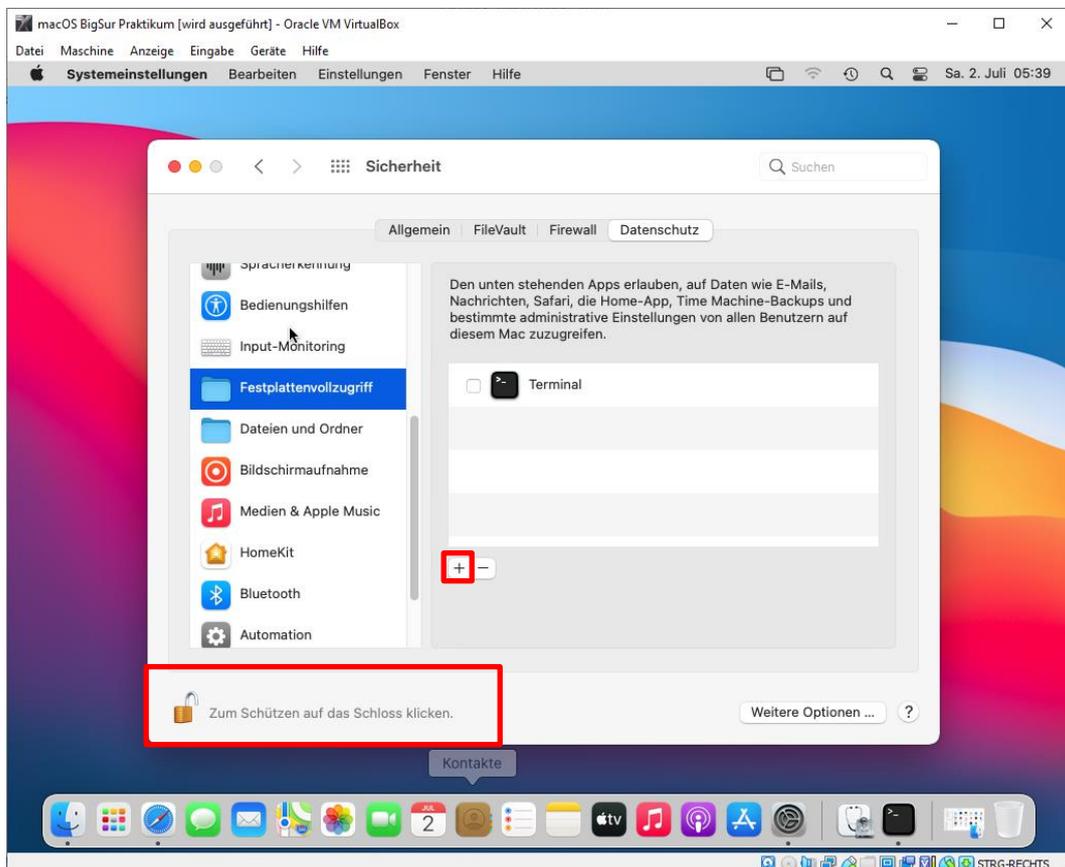


```
nutzer1 — -zsh — 80x24
nutzer1@iMac-von-Nutzer1 ~ % csrutil status
System Integrity Protection status: enabled.
nutzer1@iMac-von-Nutzer1 ~ %
```

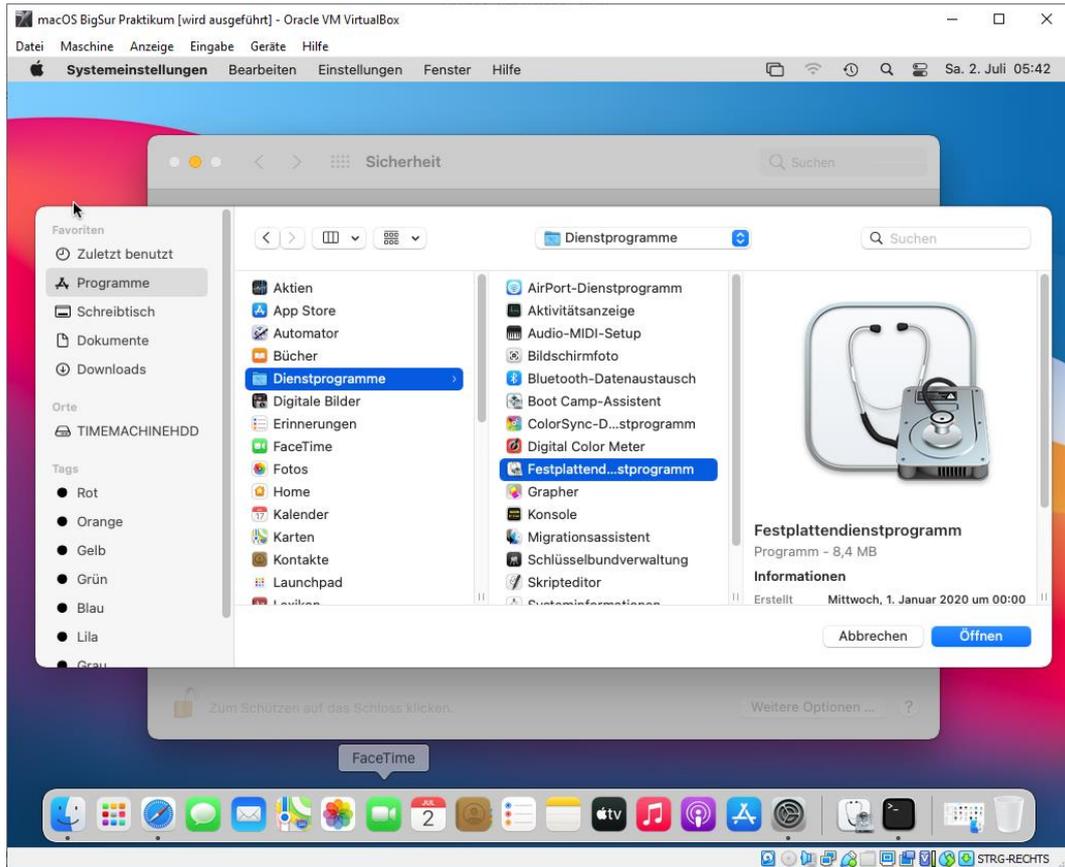
Schließen Sie das Terminal und Öffnen Sie die Systemeinstellungen im Dock.



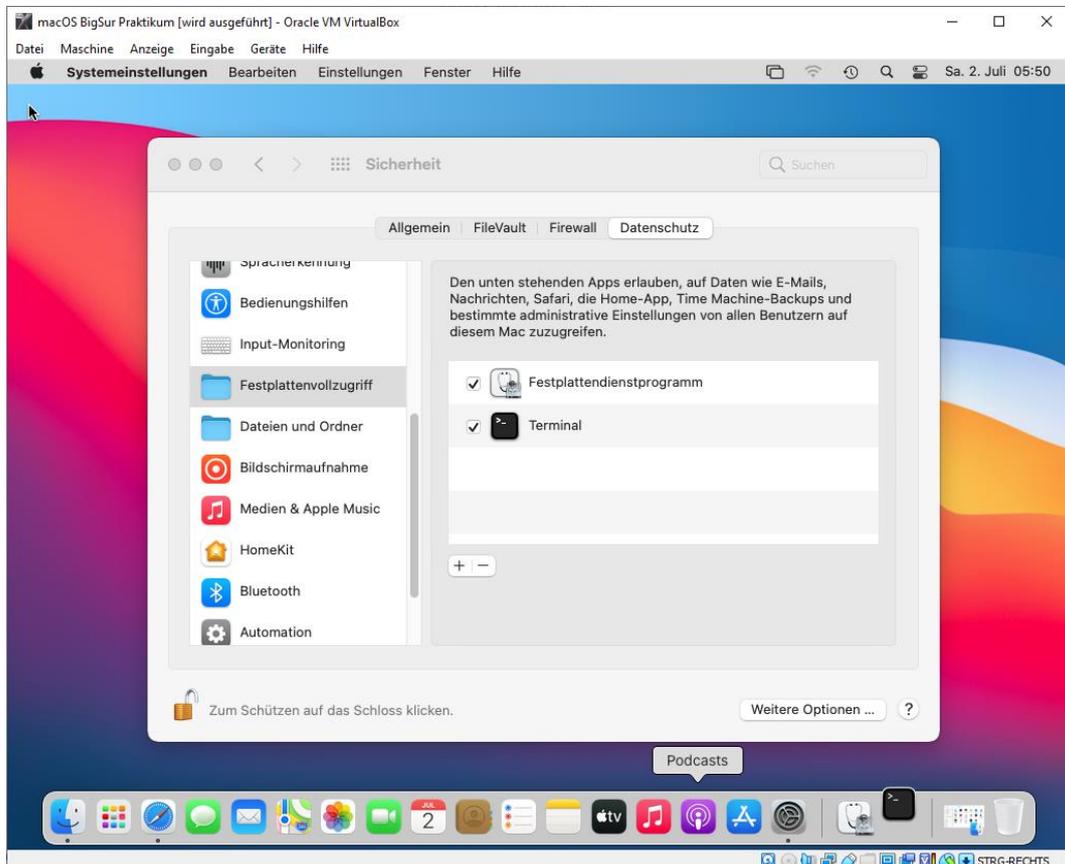
Wählen Sie danach den Punkt Sicherheit und wechseln Sie zu Datenschutz. Hier Wählen Sie bitte Festplattenvollzugriff aus.



Klicken Sie auf das Schlosssymbol und fügen Sie hier das Festplattendienstprogramm hinzu über das (+).



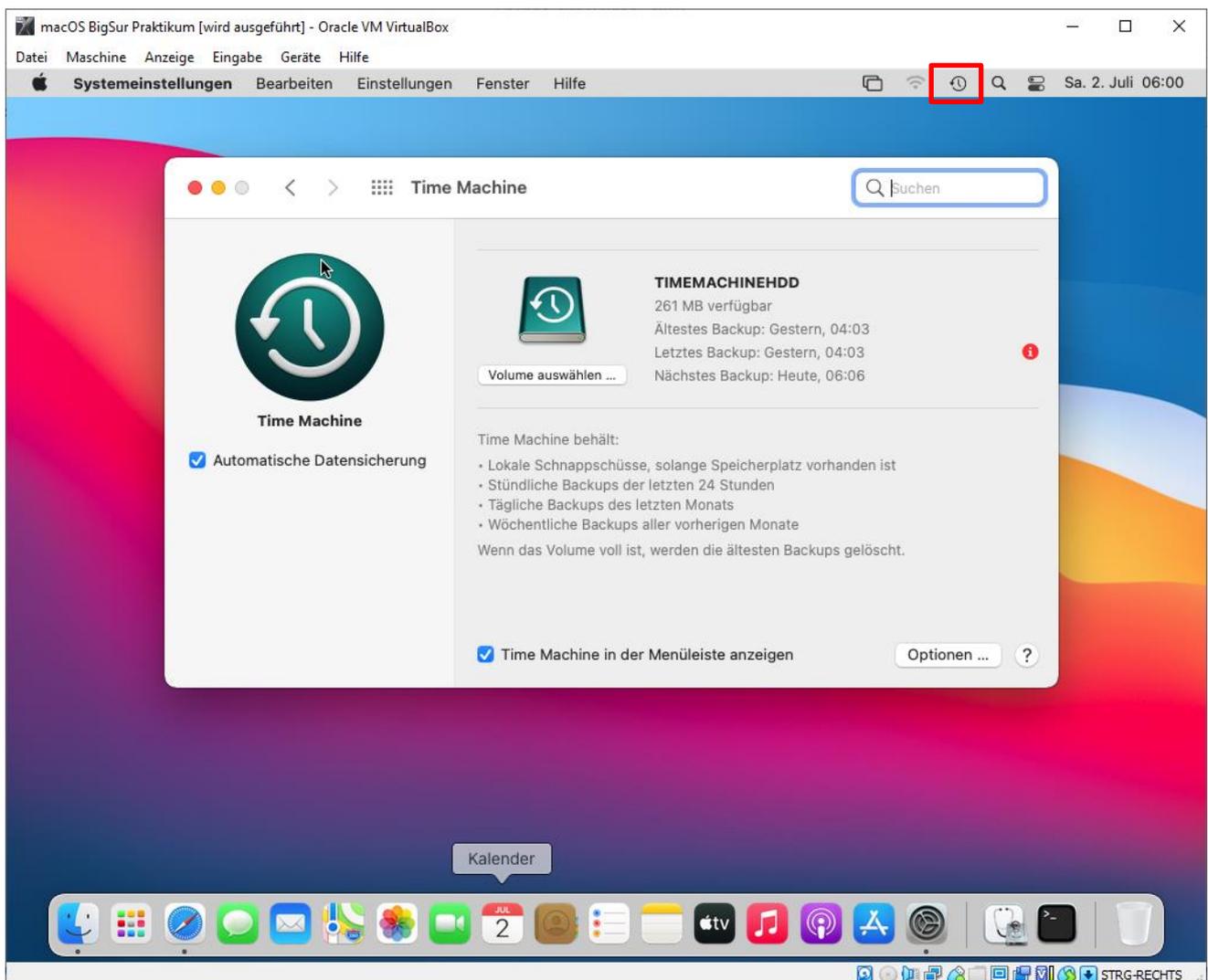
Setzen Sie zudem den **Haken** bei **Terminal**.



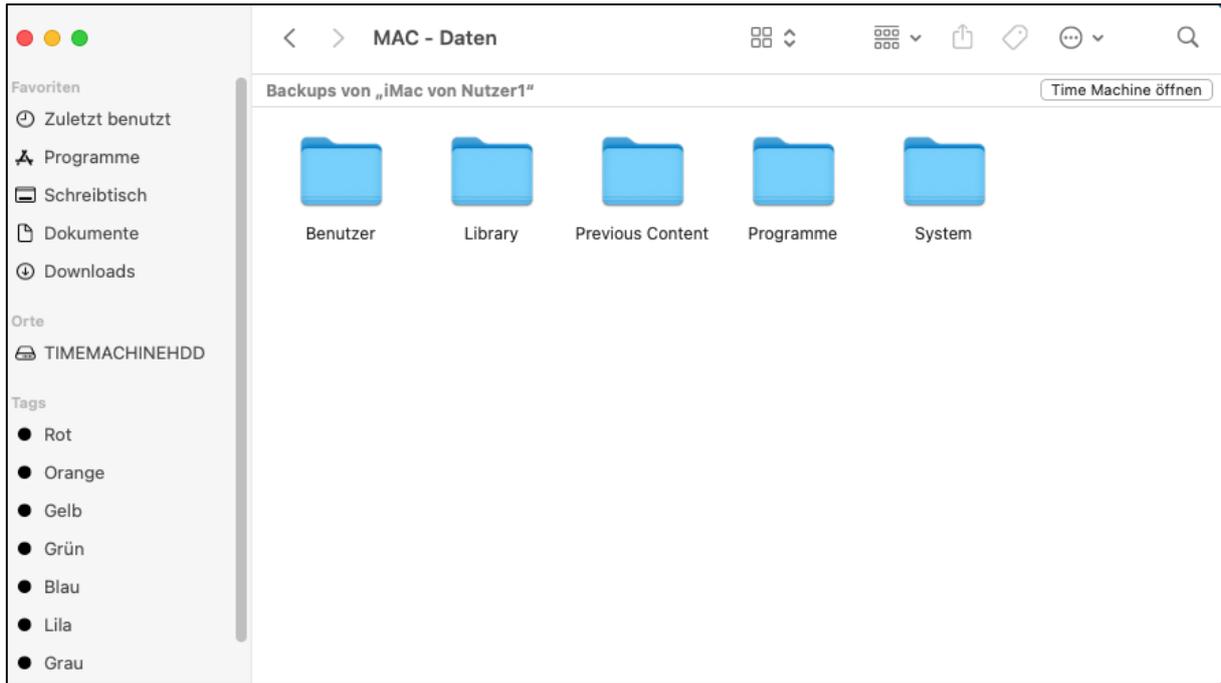
Öffnen Sie das **Terminal** Listen Sie sich das Time Machine Laufwerk, alle Backups und die lokalen Backups auf.

```
nutzer1 — -zsh — 80x24
[nutzer1@iMac-von-Nutzer1 ~ % tutil machinedirectory
/Volumes/TIMEMACHINEHDD
[nutzer1@iMac-von-Nutzer1 ~ % tutil destinationinfo
=====
Name       : TIMEMACHINEHDD
Kind       : Local
Mount Point : /Volumes/TIMEMACHINEHDD
ID         : 4550FC46-B6F5-4DB2-B8B4-AC5FD59AF5A4
[nutzer1@iMac-von-Nutzer1 ~ % tutil listbackups
2022-07-01-040352.backup
[nutzer1@iMac-von-Nutzer1 ~ % tutil listlocalsnapshots /Volumes/MAC
Snapshots for volume group containing disk /Volumes/MAC:
com.apple.TimeMachine.2022-07-01-035222.local
com.apple.TimeMachine.2022-07-02-050734.local
com.apple.os.update-779BDF1556C6F688504E24FB29C75AFFABFCB91E701806FFFF35235E1991
4F1E
com.apple.os.update-MSUPPrepareUpdate
nutzer1@iMac-von-Nutzer1 ~ %
```

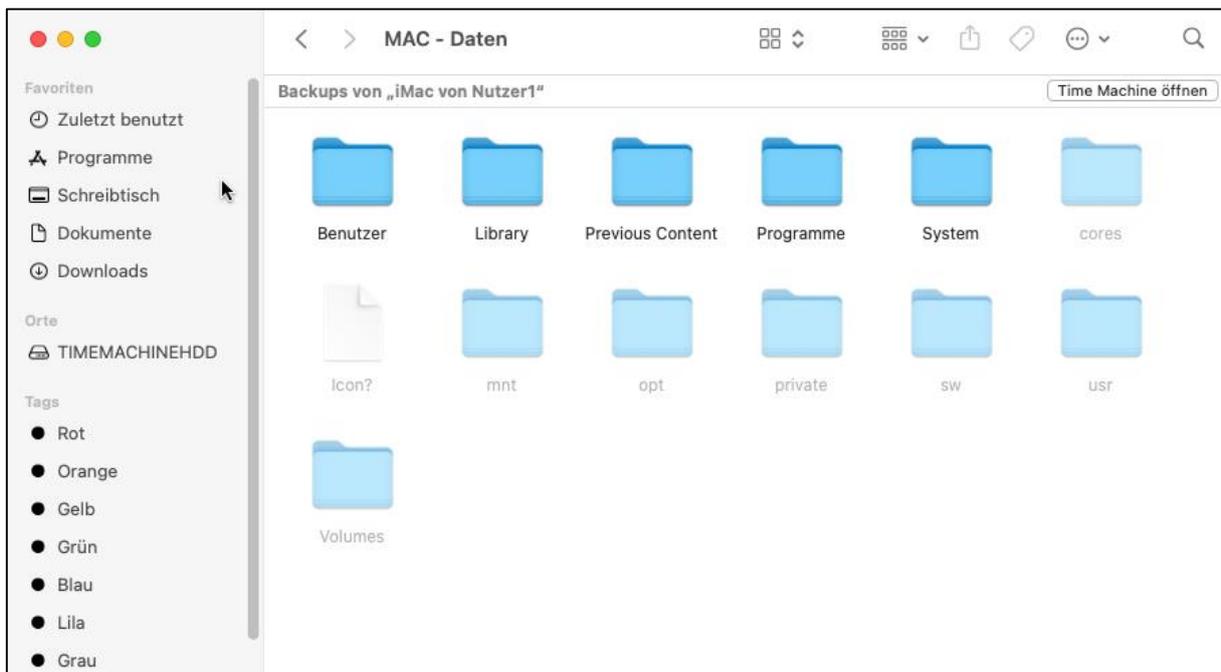
Öffnen Sie die **TimeMachine Systemeinstellungen** über das Icon in der Menüleiste.



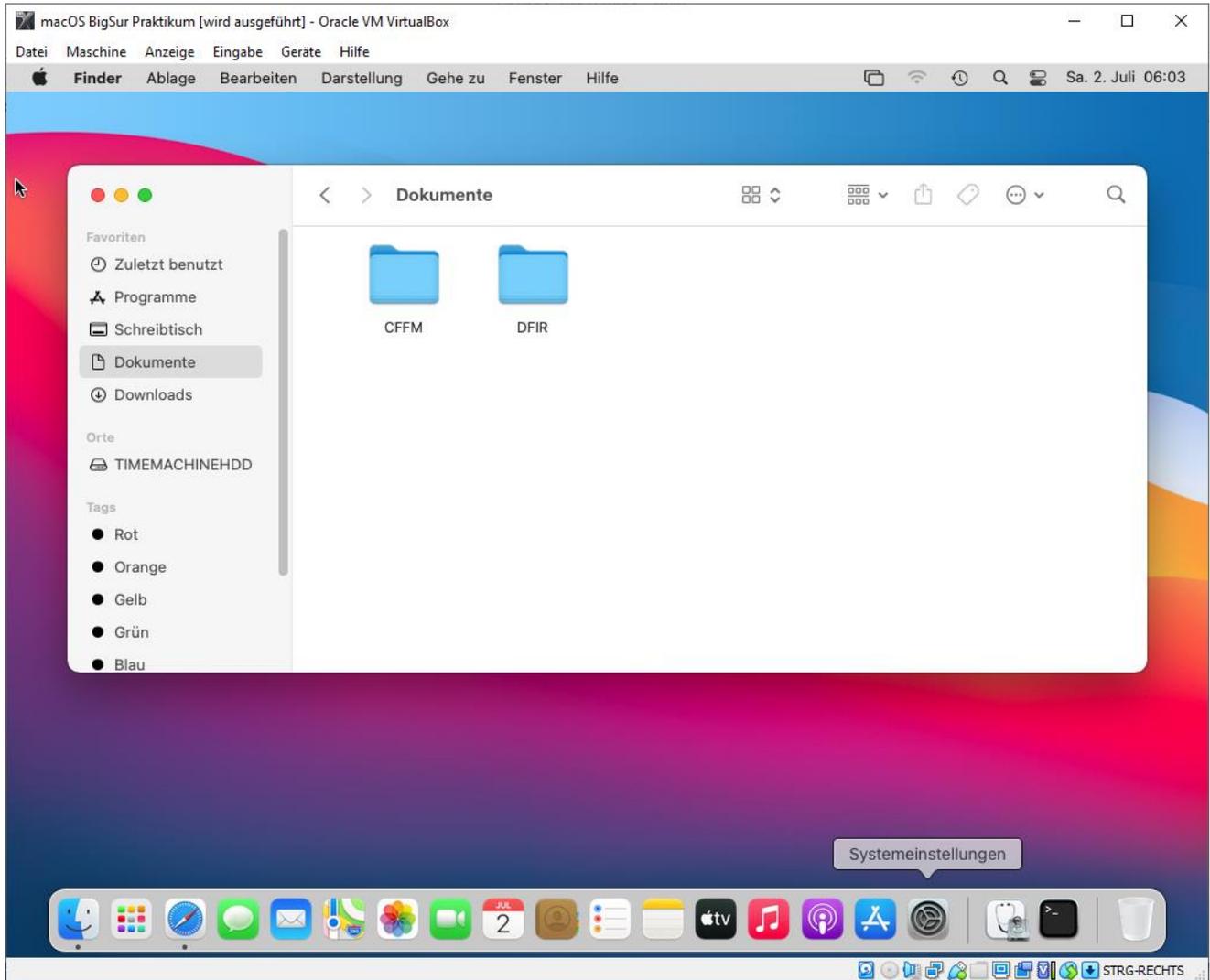
Schauen Sie auf dem Time Machine Laufwerk die Speicherstruktur an (Finder TIMMACHINEHDD).



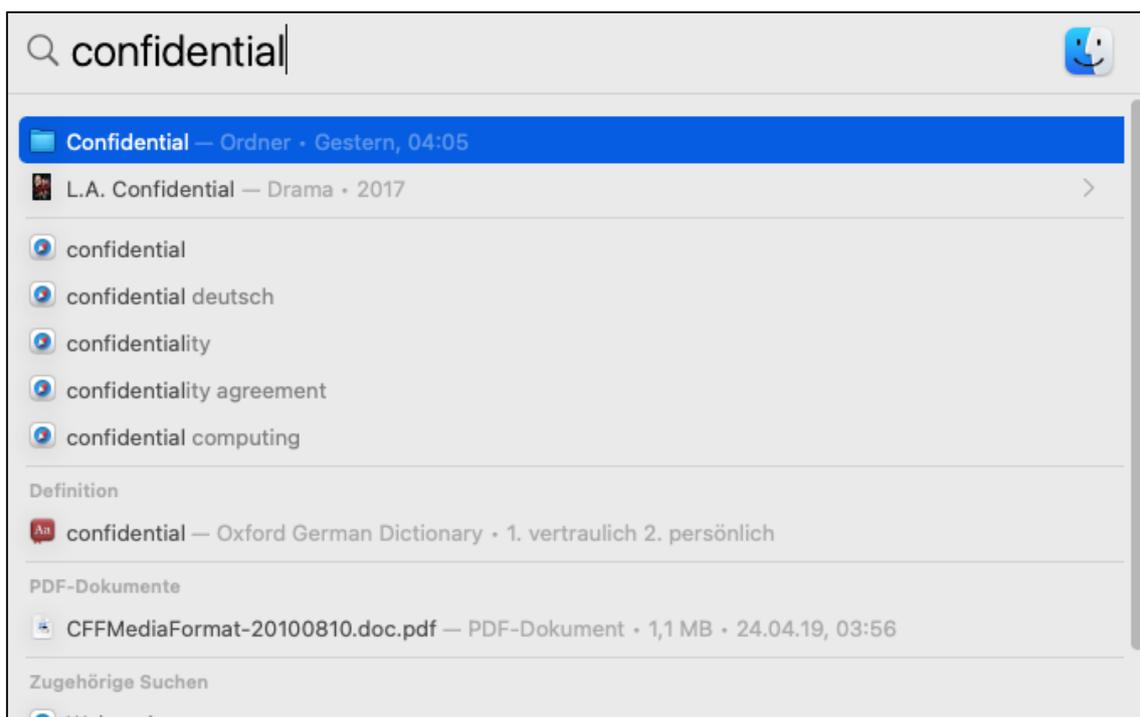
Aktivieren Sie die Ansicht der versteckten Dateien und Ordner durch **SHIFT+WINDOWS Taste + [.]**.



Öffnen Sie den Finder und schauen Sie sich in den **Dokumenten** des **Nutzers1** um.



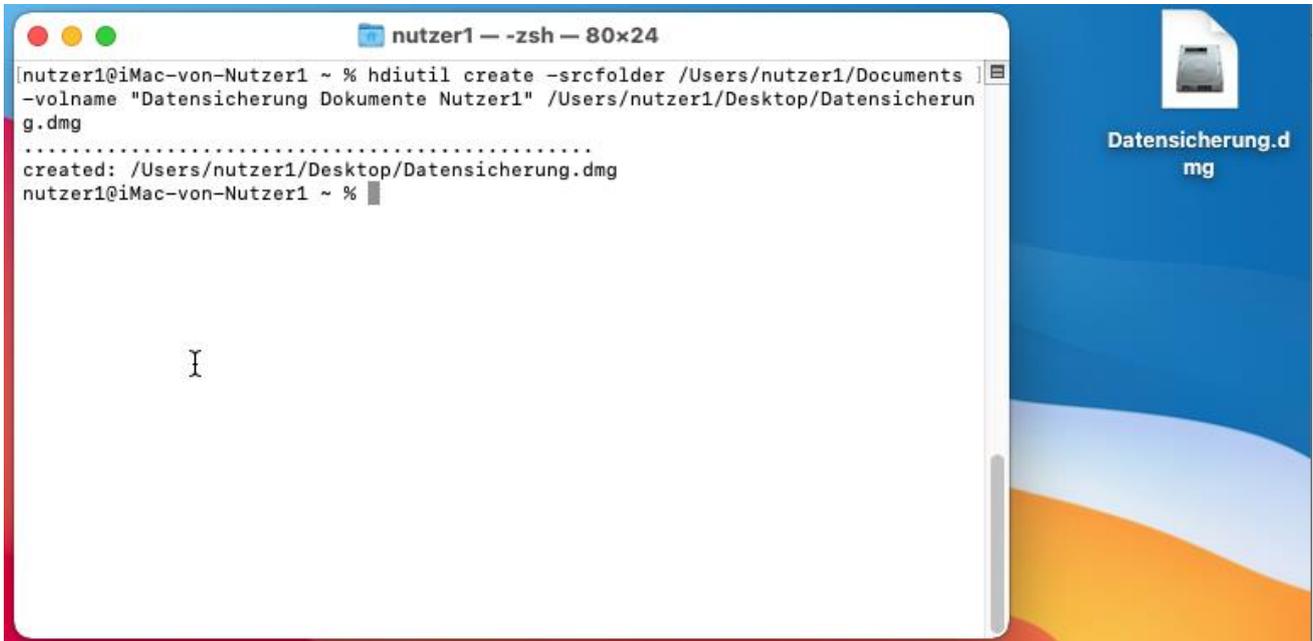
Sie suchen eine Datei die das Wort **Confidential** enthält. Nutzen Sie dazu **Spotlight** über die **Menüleiste**.



Sichern Sie die Dokumente von Nutzer1 in einem DMG Container.

Öffnen Sie dazu das Terminal und nutzen Sie den Befehl

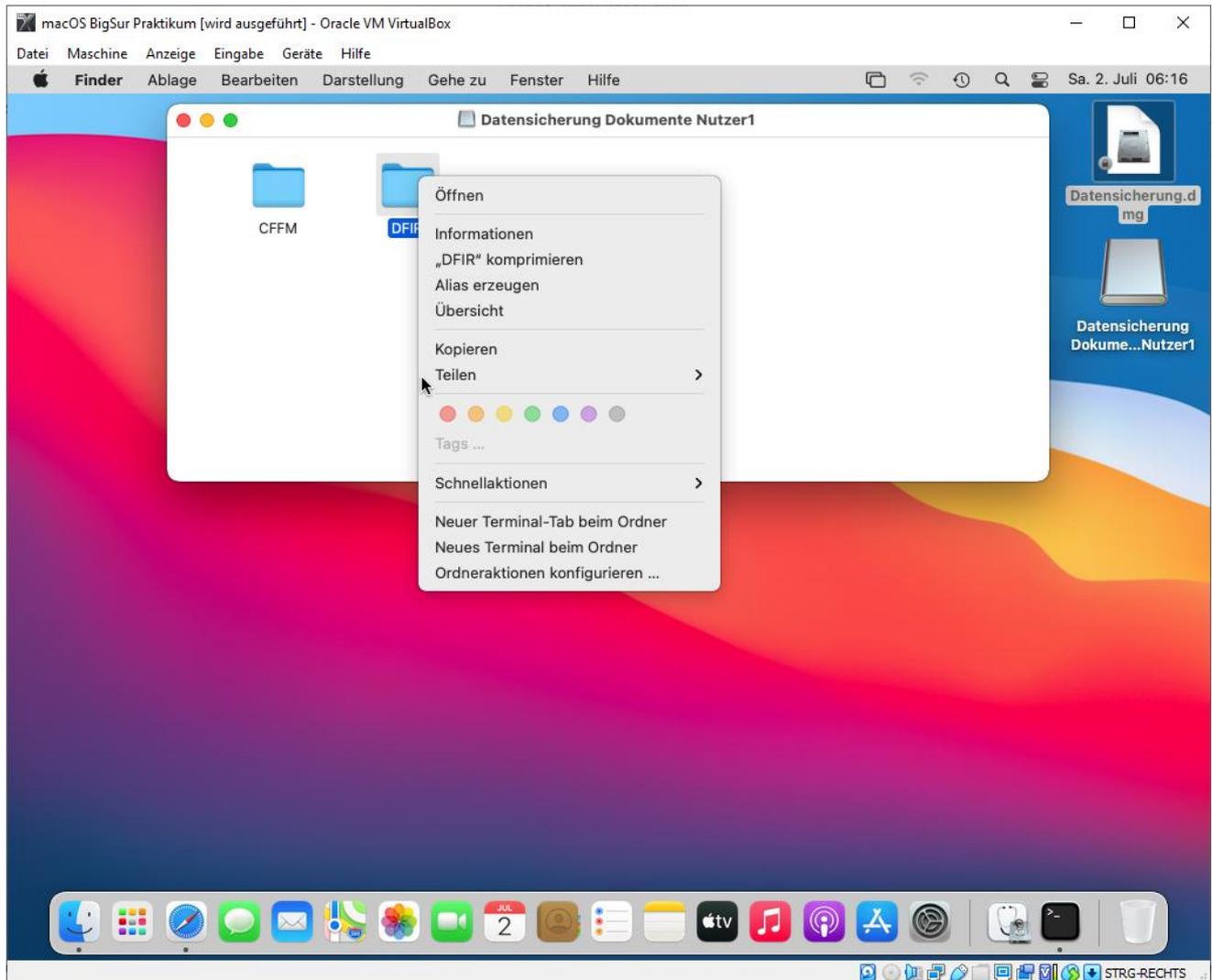
```
hdiutil create -srcfolder /Users/nutzer1/Dokumente -volname „Datensicherung Dokumente Nutzer1“ /Users/nutzer1/Desktop/Datensicherung.dmg
```



Zeigen Sie sich die Informationen zur Datei **Datensicherung.dmg** an.



Binden Sie das Image **Datensicherung.dmg** im System ein und greifen Sie auf die Dateien zu.

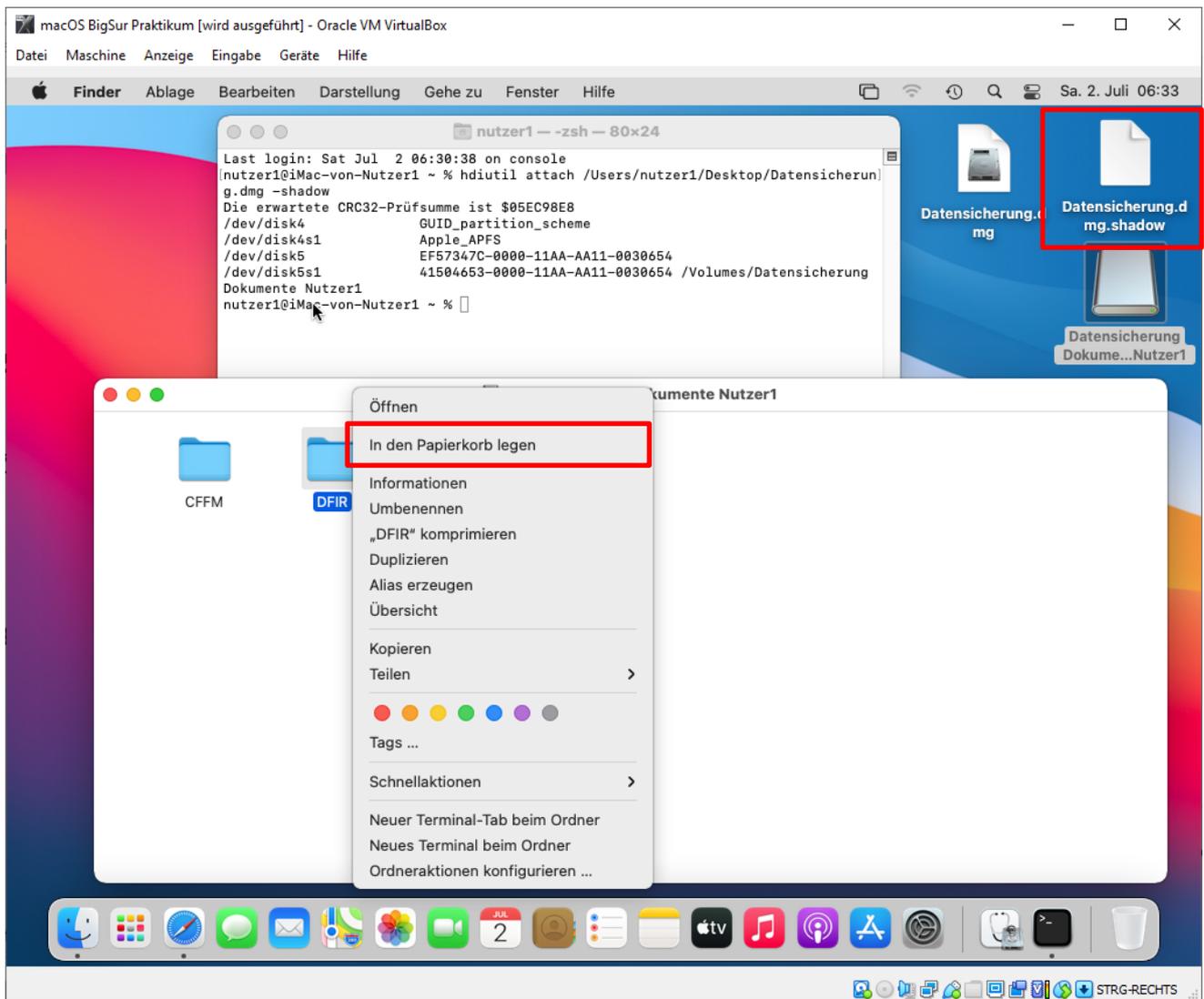


Wird das Image Schreibgeschützt eingebunden oder ist von Haus aus Schreibgeschützt, können Sie über das Kontext Menü keine Daten löschen!

Werfen Sie das gemountete Volume aus und binden Sie es über das Terminal erneut ein mit dem folgenden Befehl:

```
hdiutil attach /Users/nutzer1/Desktop/Datensicherung.dmg -shadow
```

```
nutzer1 -- -zsh -- 80x24
Last login: Sat Jul 2 06:30:38 on console
nutzer1@iMac-von-Nutzer1 ~ % hdiutil attach /Users/nutzer1/Desktop/Datensicherung.dmg -shadow
Die erwartete CRC32-Prüfsumme ist $05EC98E8
/dev/disk4          GUID_partition_scheme
/dev/disk4s1       Apple_APFS
/dev/disk5          EF57347C-0000-11AA-AA11-0030654
/dev/disk5s1       41504653-0000-11AA-AA11-0030654 /Volumes/Datensicherung
Dokumente Nutzer1
nutzer1@iMac-von-Nutzer1 ~ %
```



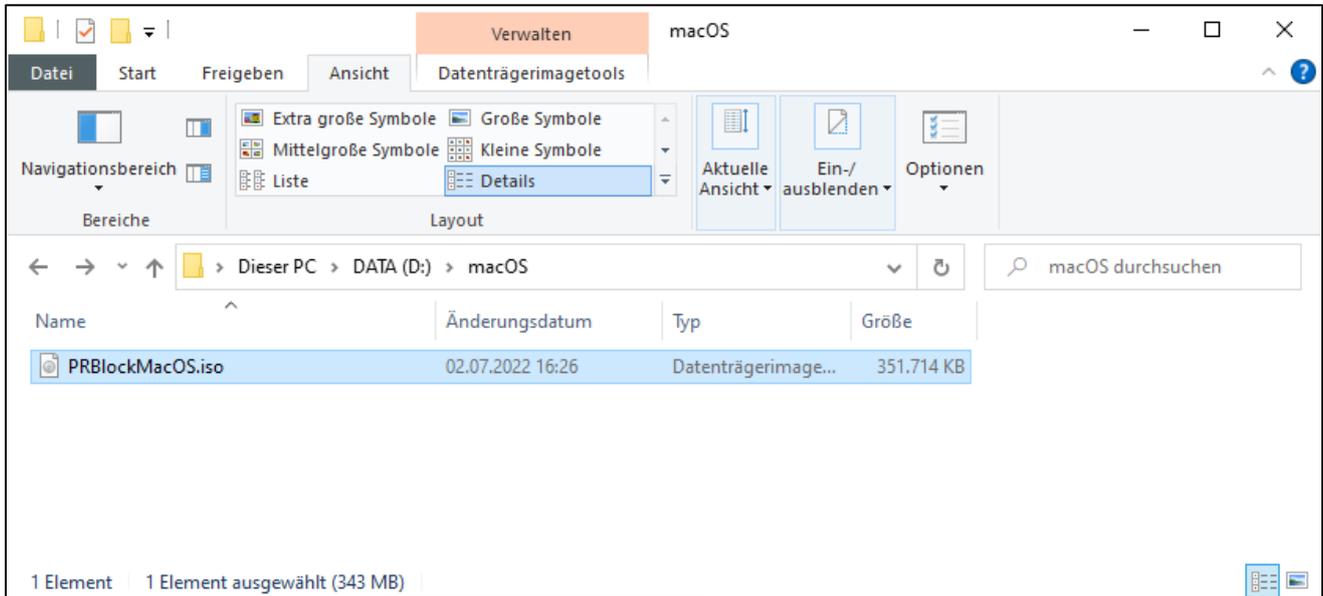
Jetzt wird zusätzlich eine **Shadow Datei** erzeugt, welche die Änderungen aufnimmt.

Dies sieht man daran, dass Dateien auch gelöscht werden können.

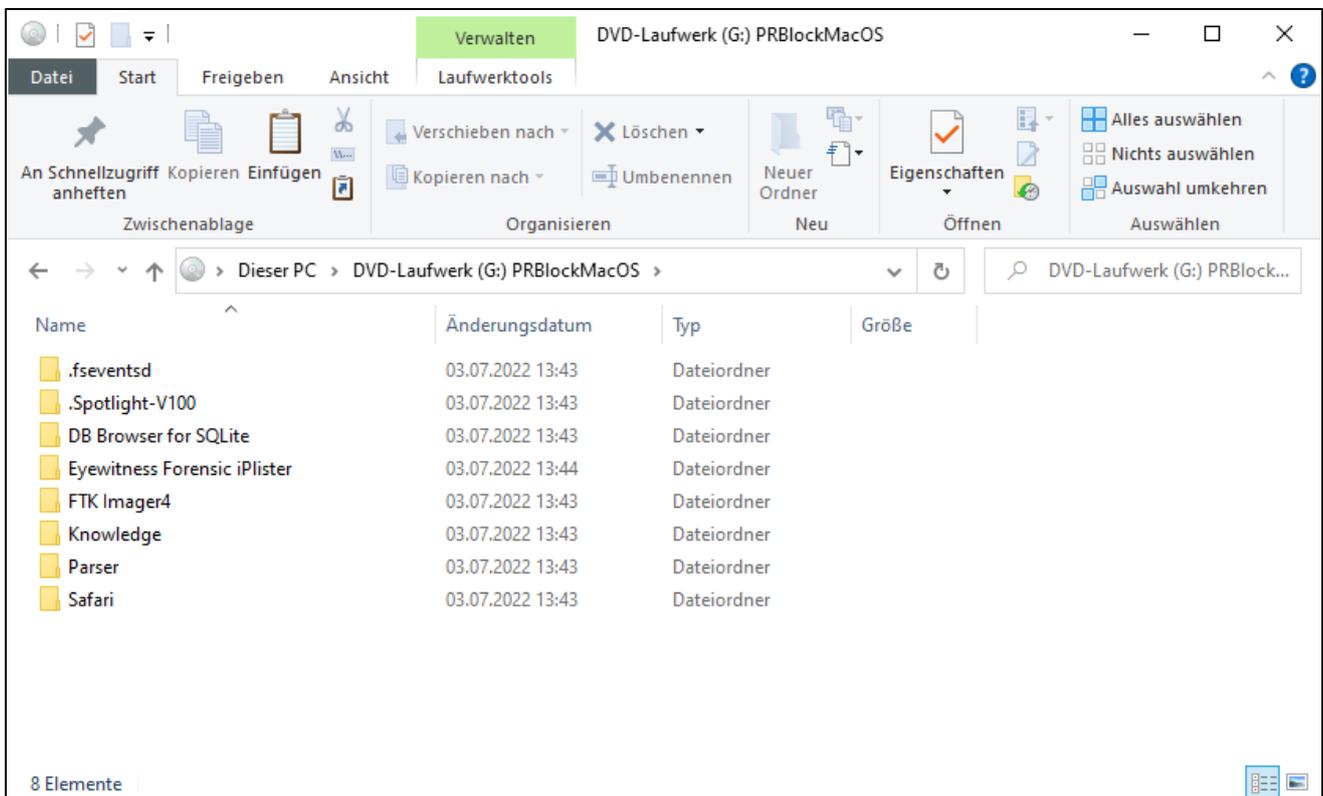
3. UNTERSUCHUNG DER MACOS ARTEFAKTE UND DIGITALER SPUREN IN EXTERNEM BETRIEBSSYSTEM

Zur Vorbereitung erstellen Sie auf Laufwerk D:\ ein Verzeichnis **macOS** und kopieren dahin die **PRBlockMacOS.iso** Datei.

Binden Sie zum Bearbeiten der Daten die auf die lokale Festplatte kopierte ISO Datei **PRBlockMacOS.iso** durch Doppelklick ein.

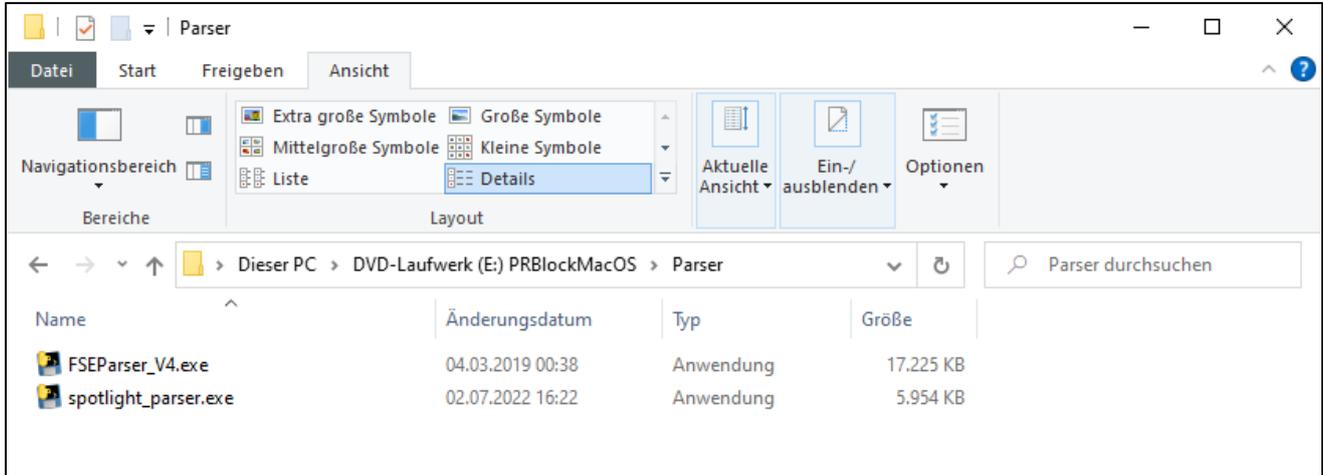


Auf der eingebundenen CD finden Sie alle nötigen Tools und auch die bereits gesicherten Daten des macOS.

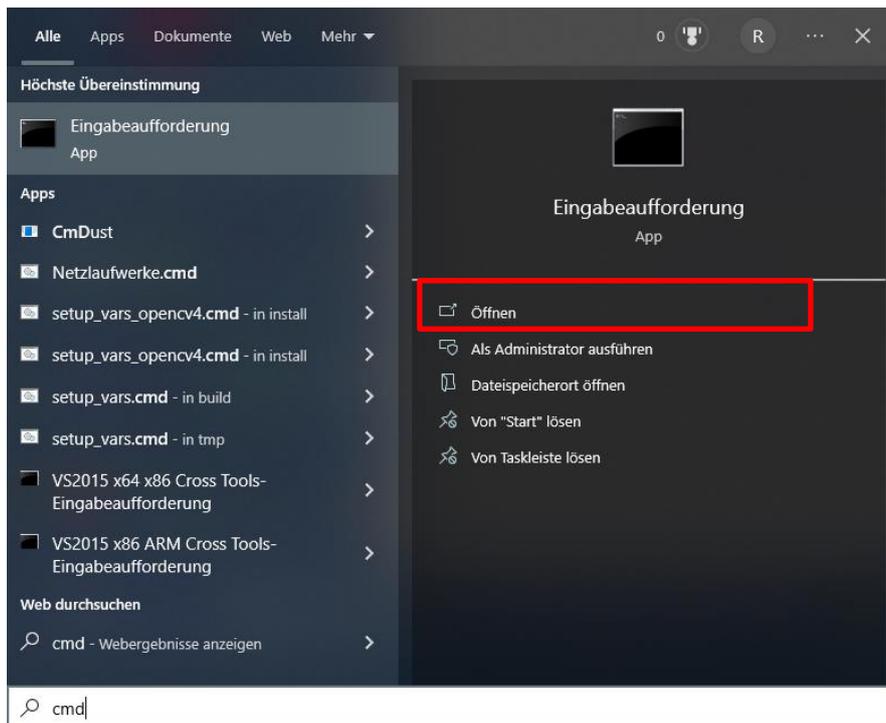


3.1. SPOTLIGHT UNTERSUCHUNG MIT PYTHON PARSER

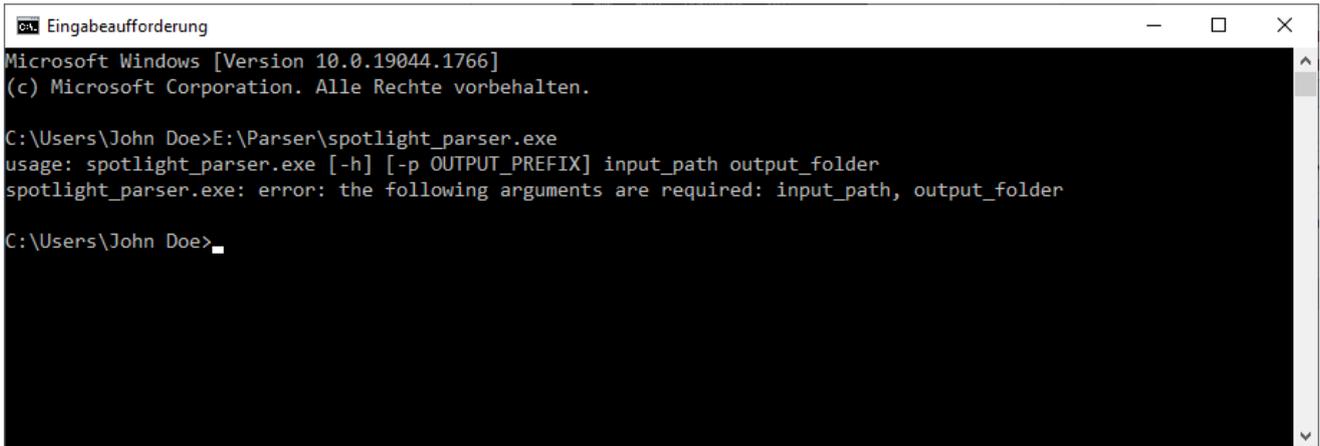
Rufen Sie den das Verzeichnis Parser auf der CD auf. Her befinden sich zwei in Python geschriebene Parser für Spotlight und FSEvents.



Öffnen Sie eine Eingabeaufforderung durch Klicken auf den Start Button und Eingabe von CMD. Es reicht eine Eingabeaufforderung ohne Admin Rechte.



Starten Sie den SpotlightParser in der Eingabeaufforderung.

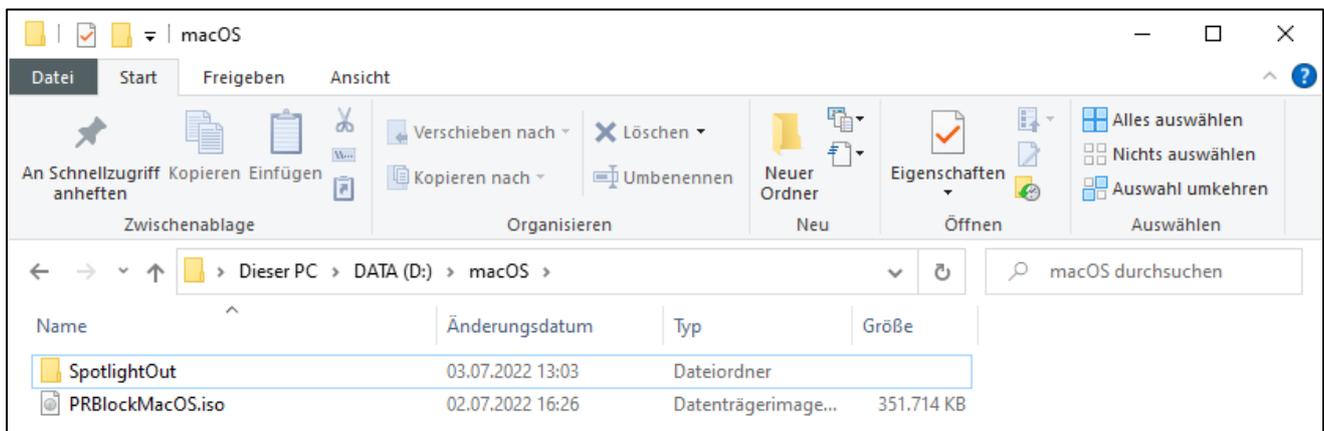


```
Microsoft Windows [Version 10.0.19044.1766]
(c) Microsoft Corporation. Alle Rechte vorbehalten.

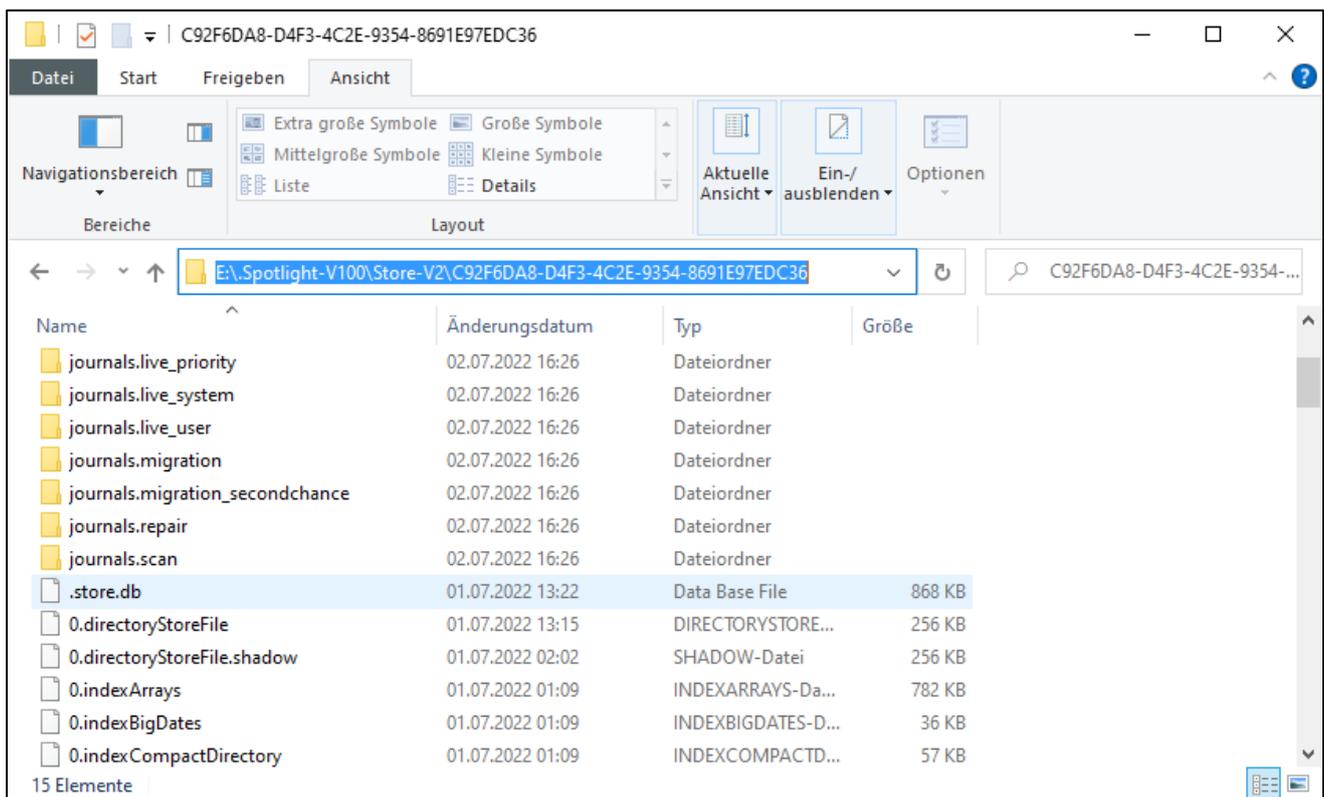
C:\Users\John Doe>E:\Parser\spotlight_parser.exe
usage: spotlight_parser.exe [-h] [-p OUTPUT_PREFIX] input_path output_folder
spotlight_parser.exe: error: the following arguments are required: input_path, output_folder

C:\Users\John Doe>
```

Der SpotlightParser benötigt zur Ausgabe ein Verzeichnis, welches vor Nutzung angelegt werden muss.

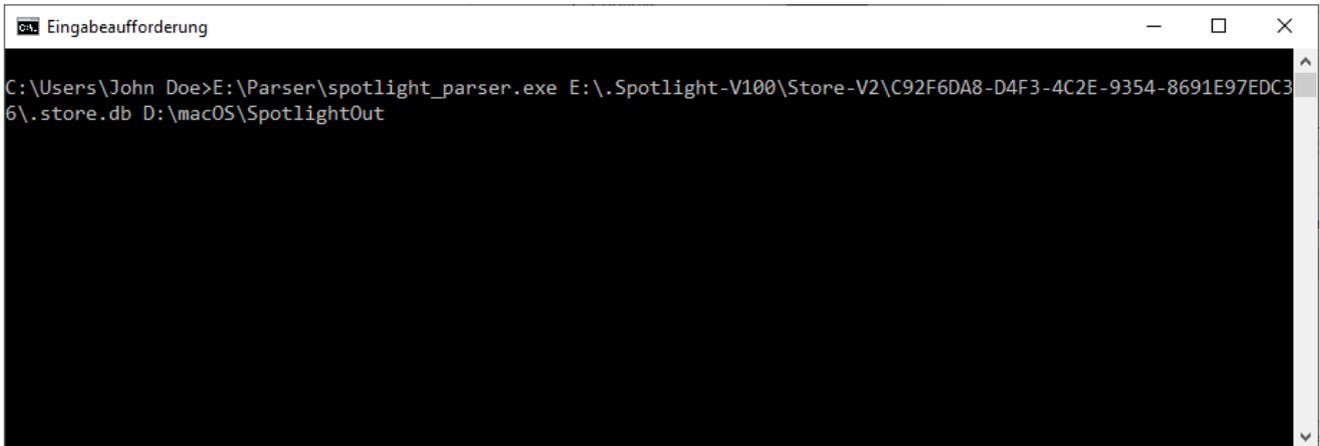


Die zu untersuchende Spotlight Datenbank Datei finden Sie unter `\\.Spotlight-V100\Store-V2\C92F6DA8-D4F3-4C2E-9354-8691E97EDC36\.store.db`

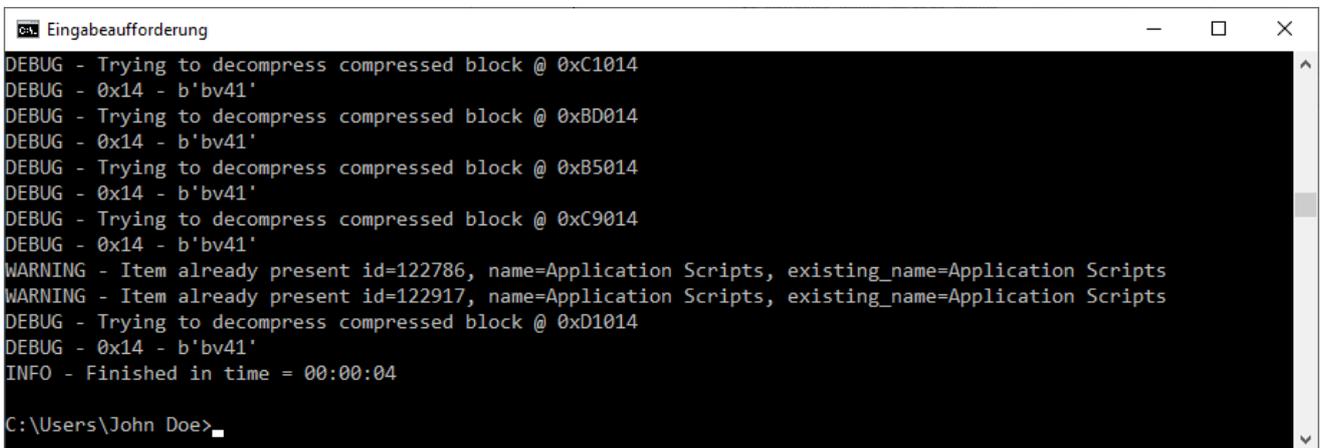


Mit folgendem Befehl kann der Spotlight Parser gestartet werden:

```
\Parser\spotlight_parser.exe G:\.Spotlight-V100\Store-V2\C92F6DA8-D4F3-4C2E-9354-8691E97EDC36\store.db D:\macOS\SpotlightOut
```



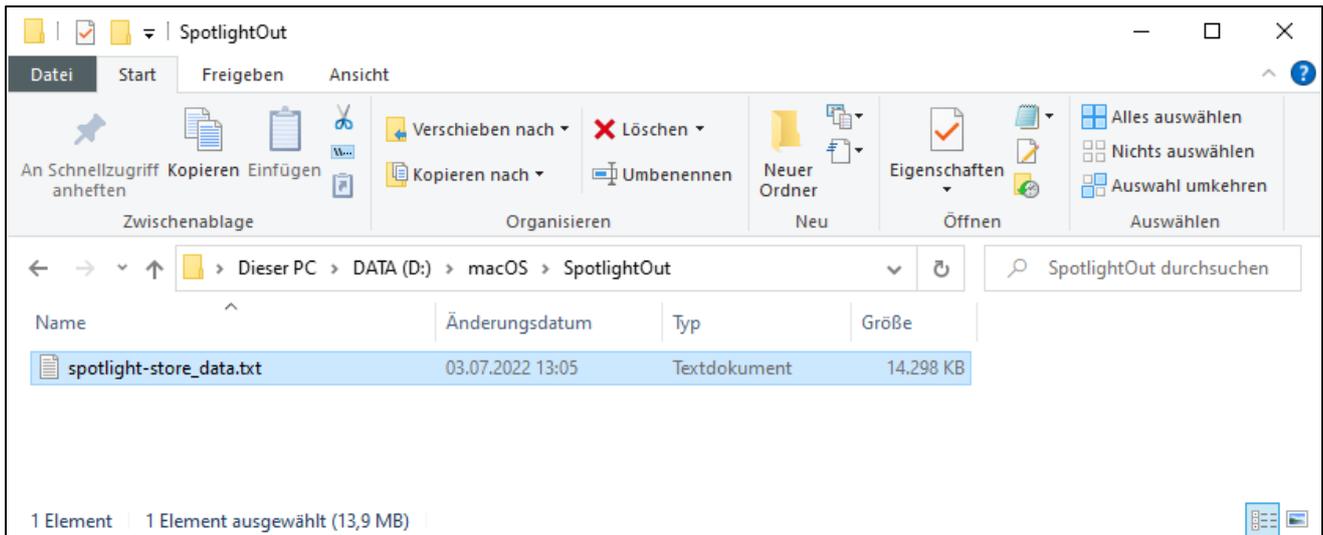
```
Eingabeaufforderung
C:\Users\John Doe>E:\Parser\spotlight_parser.exe E:\.Spotlight-V100\Store-V2\C92F6DA8-D4F3-4C2E-9354-8691E97EDC36\store.db D:\macOS\SpotlightOut
```



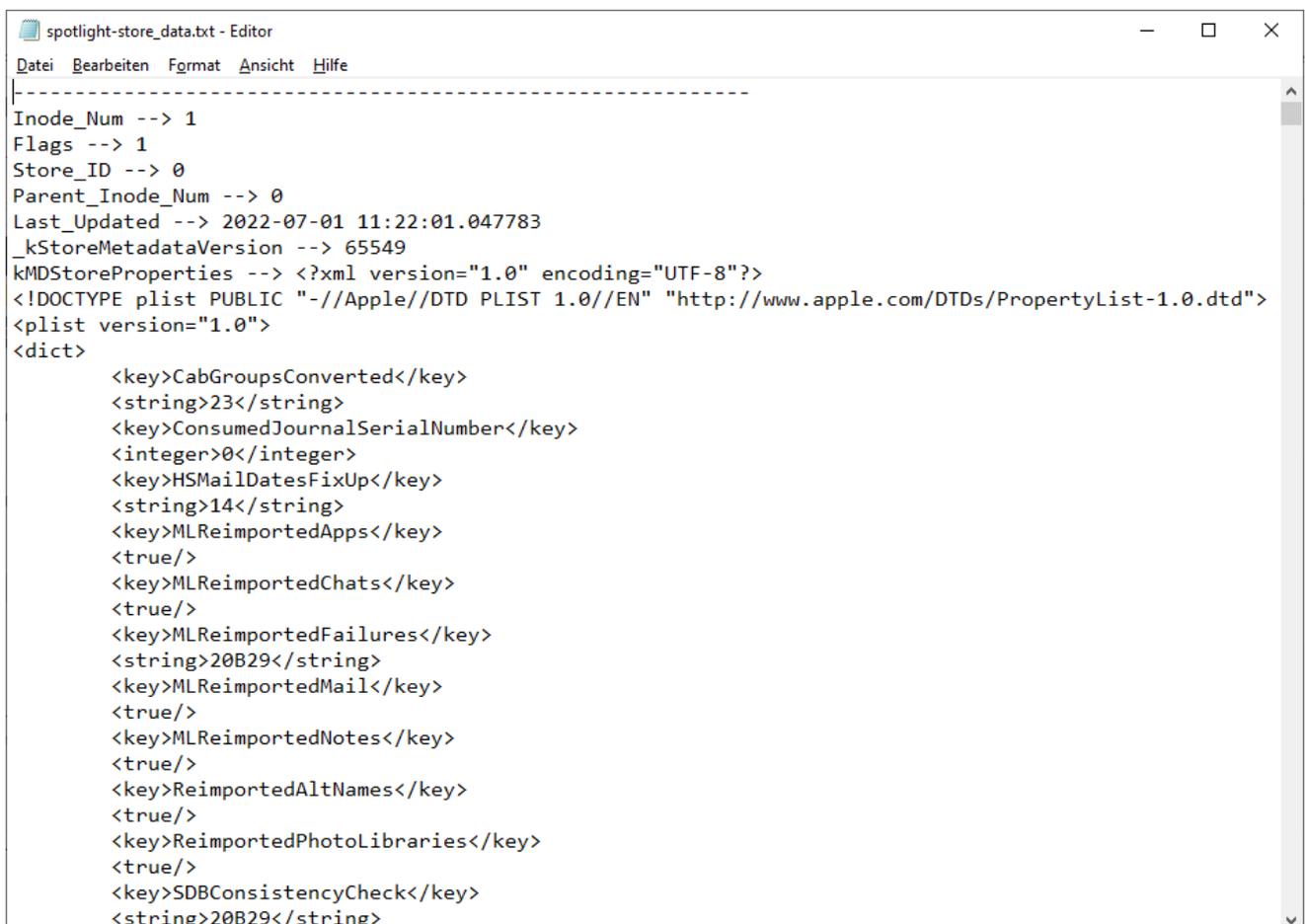
```
Eingabeaufforderung
DEBUG - Trying to decompress compressed block @ 0xC1014
DEBUG - 0x14 - b'bv41'
DEBUG - Trying to decompress compressed block @ 0xBD014
DEBUG - 0x14 - b'bv41'
DEBUG - Trying to decompress compressed block @ 0xB5014
DEBUG - 0x14 - b'bv41'
DEBUG - Trying to decompress compressed block @ 0xC9014
DEBUG - 0x14 - b'bv41'
WARNING - Item already present id=122786, name=Application Scripts, existing_name=Application Scripts
WARNING - Item already present id=122917, name=Application Scripts, existing_name=Application Scripts
DEBUG - Trying to decompress compressed block @ 0xD1014
DEBUG - 0x14 - b'bv41'
INFO - Finished in time = 00:00:04

C:\Users\John Doe>
```

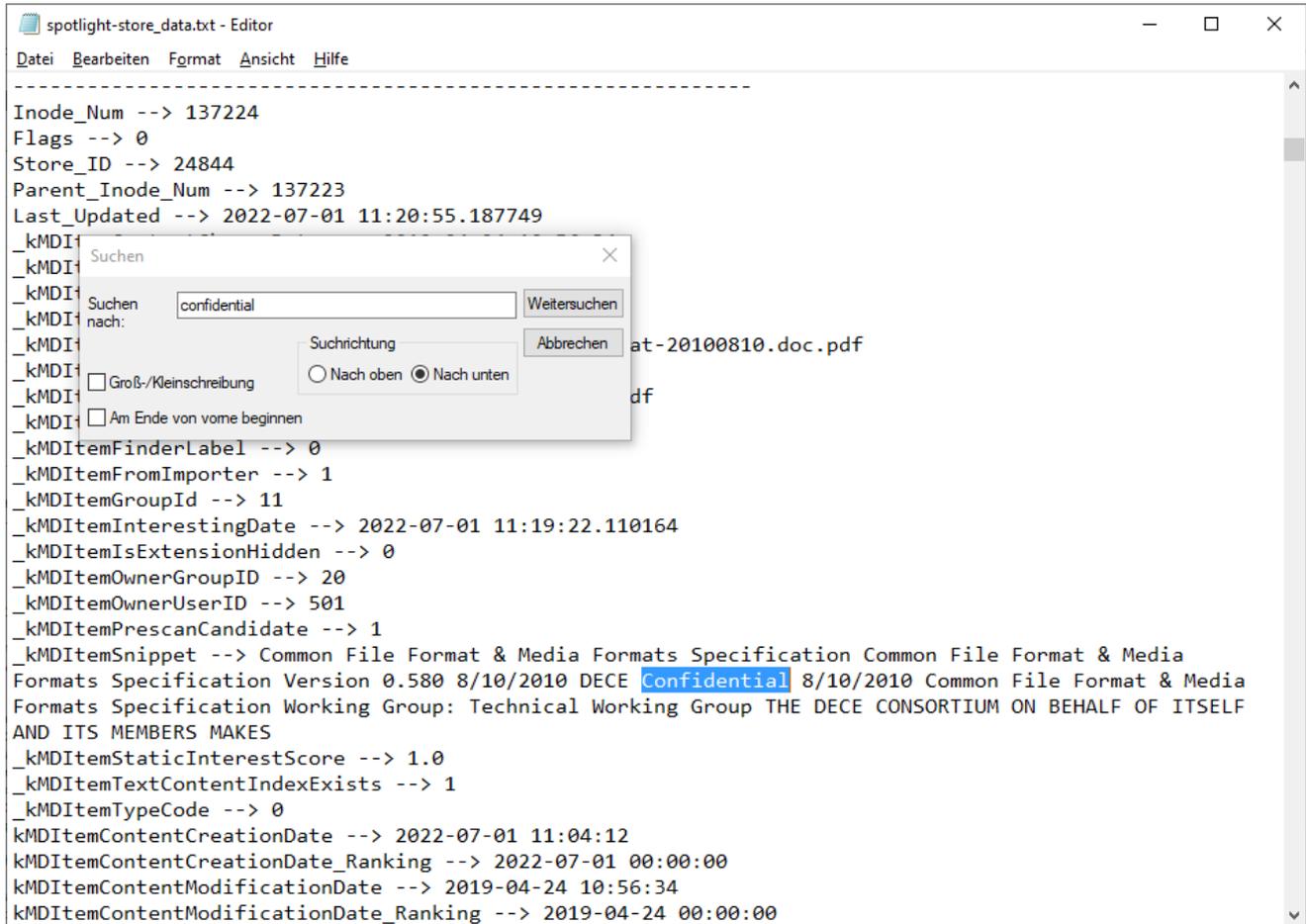
Im Ausgabeverzeichnis erstellt der Parser eine Textdatei mit den aufbereiteten Spotlight Meta-Informationen.



Hier finden Sie alle ausgelesenen Daten der Spotlight Datenbank.



Suchen Sie nach dem Schlüsselwort **Confidential!**



spotlight-store_data.txt - Editor

Datei Bearbeiten Format Ansicht Hilfe

Inode_Num --> 137224
Flags --> 0
Store_ID --> 24844
Parent_Inode_Num --> 137223
Last_Updated --> 2022-07-01 11:20:55.187749

_kMDIItemFinderLabel --> 0
_kMDIItemFromImporter --> 1
_kMDIItemGroupId --> 11
_kMDIItemInterestingDate --> 2022-07-01 11:19:22.110164
_kMDIItemIsExtensionHidden --> 0
_kMDIItemOwnerGroupId --> 20
_kMDIItemOwnerUserId --> 501
_kMDIItemPrescanCandidate --> 1
_kMDIItemSnippet --> Common File Format & Media Formats Specification Common File Format & Media
Formats Specification Version 0.580 8/10/2010 DECE Confidential 8/10/2010 Common File Format & Media
Formats Specification Working Group: Technical Working Group THE DECE CONSORTIUM ON BEHALF OF ITSELF
AND ITS MEMBERS MAKES
_kMDIItemStaticInterestScore --> 1.0
_kMDIItemTextContentIndexExists --> 1
_kMDIItemTypeCode --> 0
kMDIItemContentCreationDate --> 2022-07-01 11:04:12
kMDIItemContentCreationDate_Ranking --> 2022-07-01 00:00:00
kMDIItemContentModificationDate --> 2019-04-24 10:56:34
kMDIItemContentModificationDate_Ranking --> 2019-04-24 00:00:00

Suchen

Suchen nach: confidential

Suchrichtung: Nach oben Nach unten

Groß-/Kleinschreibung

Am Ende von vorne beginnen

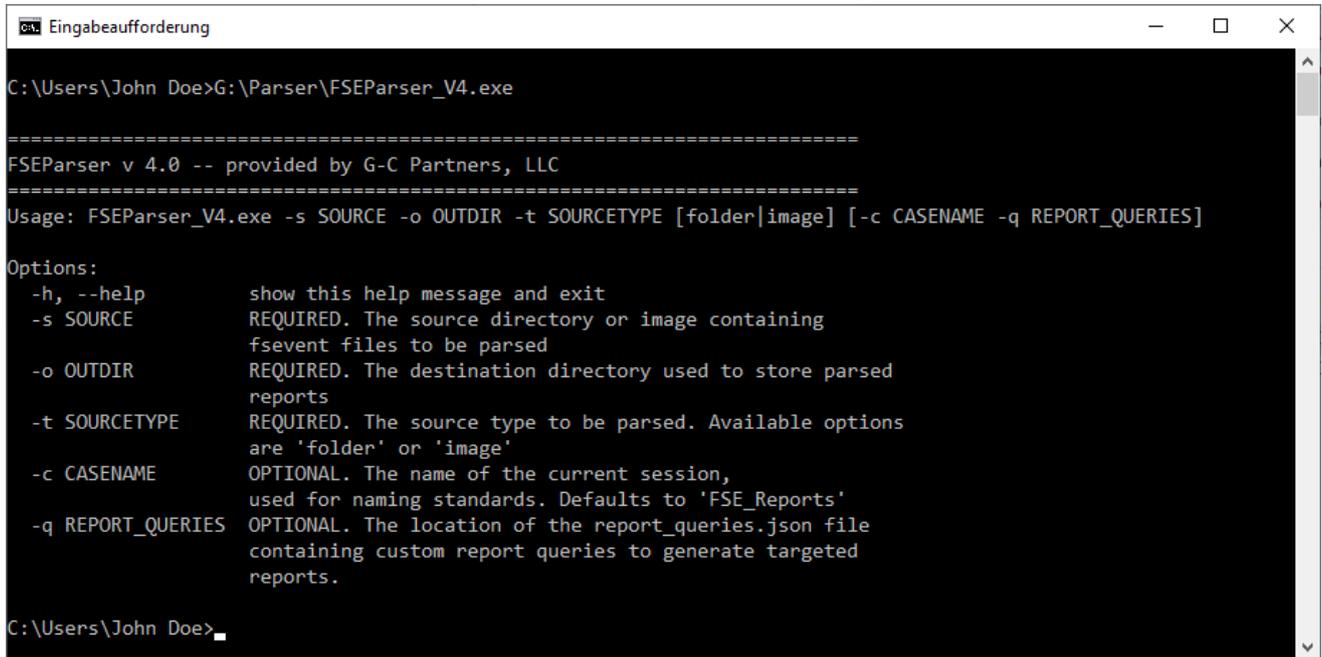
Weitersuchen

Abbrechen

Sie erhalten sogar die indizierten Meta Informationen die aus der PDF Datei gelesen und indiziert wurden!

3.2. FSEVENT UNTERSUCHUNG MIT FSEVENT PARSER

Für die Untersuchung der FSEvent Daten wird der FSEvent Parser verwendet der als Executable im Verzeichnis Parser auf der CD vorhanden ist.



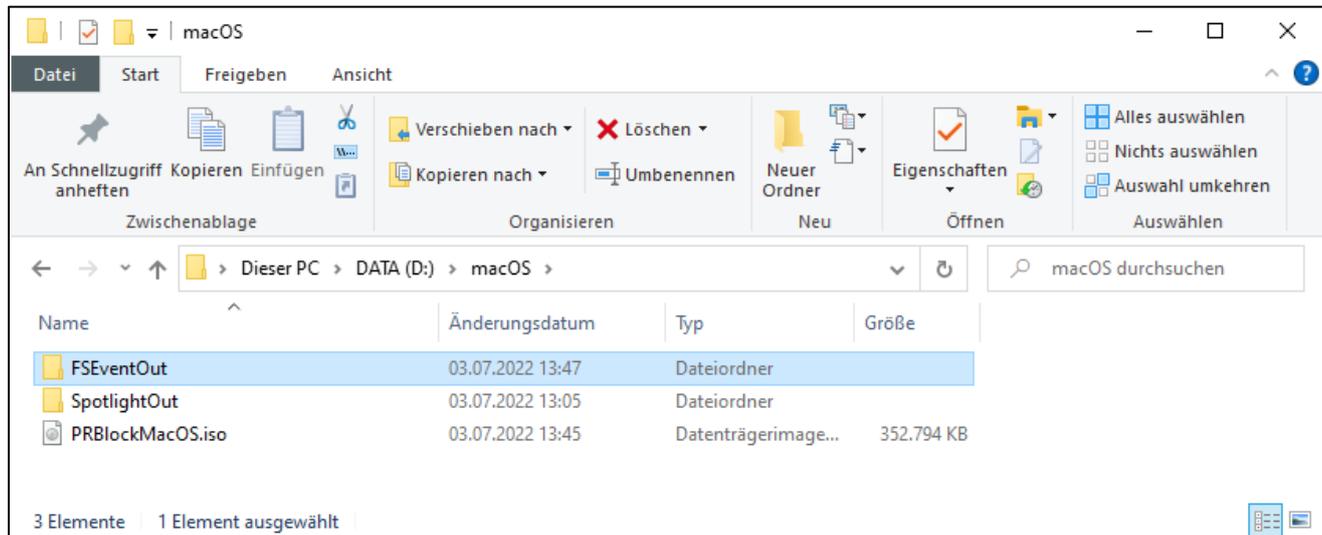
```
C:\Users\John Doe>G:\Parser\FSEParser_V4.exe

=====
FSEParser v 4.0 -- provided by G-C Partners, LLC
=====
Usage: FSEParser_V4.exe -s SOURCE -o OUTDIR -t SOURCETYPE [folder|image] [-c CASENAME -q REPORT_QUERIES]

Options:
-h, --help          show this help message and exit
-s SOURCE           REQUIRED. The source directory or image containing
                   fsevent files to be parsed
-o OUTDIR           REQUIRED. The destination directory used to store parsed
                   reports
-t SOURCETYPE       REQUIRED. The source type to be parsed. Available options
                   are 'folder' or 'image'
-c CASENAME         OPTIONAL. The name of the current session,
                   used for naming standards. Defaults to 'FSE_Reports'
-q REPORT_QUERIES  OPTIONAL. The location of the report_queries.json file
                   containing custom report queries to generate targeted
                   reports.

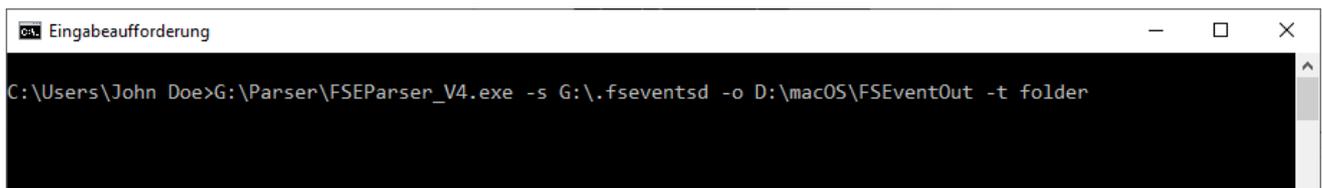
C:\Users\John Doe>
```

Der FSEventParser benötigt zur Ausgabe ein Verzeichnis, welches vor Nutzung angelegt werden muss.



Mit folgendem Befehl kann der FSEvent Parser gestartet werden:

```
\Parser\FSEParser_V4.exe -s G:\.fseventsd -o D:\macOS\FSEventOut\FSE_Reports -t folder
```



```
C:\Users\John Doe>G:\Parser\FSEParser_V4.exe -s G:\.fseventsd -o D:\macOS\FSEventOut -t folder
```

```
Eingabeaufforderung
C:\Users\John Doe>G:\Parser\FSEParser_V4.exe -s G:\.fsevents -o D:\macOS\FSEventOut -t folder

=====
FSEParser v 4.0 -- provided by G-C Partners, LLC
=====
[Info]: Report queries file not specified using the -q option. Custom reports will not be generated.
[Info]: No casename specified using -c. Defaulting to "FSE_Reports".

[STARTED] 07/03/2022 11:48:52 UTC Parsing files.
  File 9 of 9 [=====] 100.0%
  All Files Attempted: 9
  All Parsed Files: 9
  Files with Errors: 0
  All Records Parsed: 20260
[FINISHED] 07/03/2022 11:49:02 UTC Parsing files.

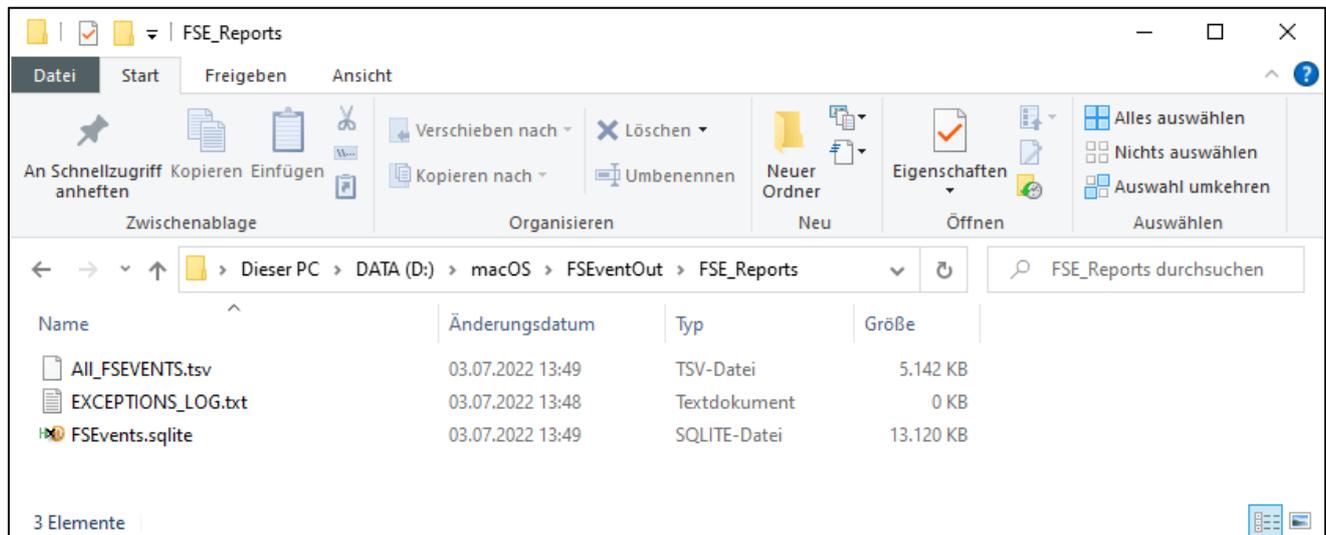
[STARTED] 07/03/2022 11:49:02 UTC Sorting fsevents table in Database.
[FINISHED] 07/03/2022 11:49:03 UTC Sorting fsevents table in Database.

[STARTED] 07/03/2022 11:49:03 UTC Exporting fsevents table from Database.
[FINISHED] 07/03/2022 11:49:03 UTC Exporting fsevents table from Database.

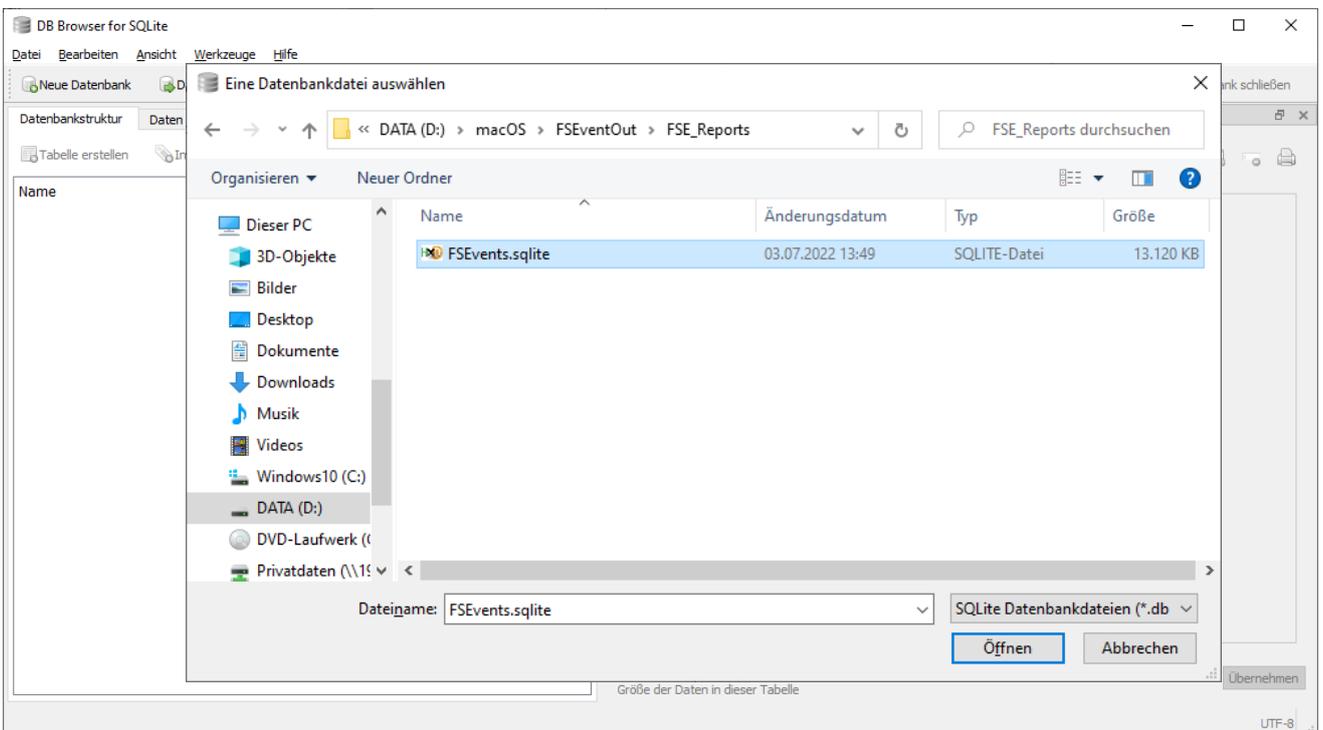
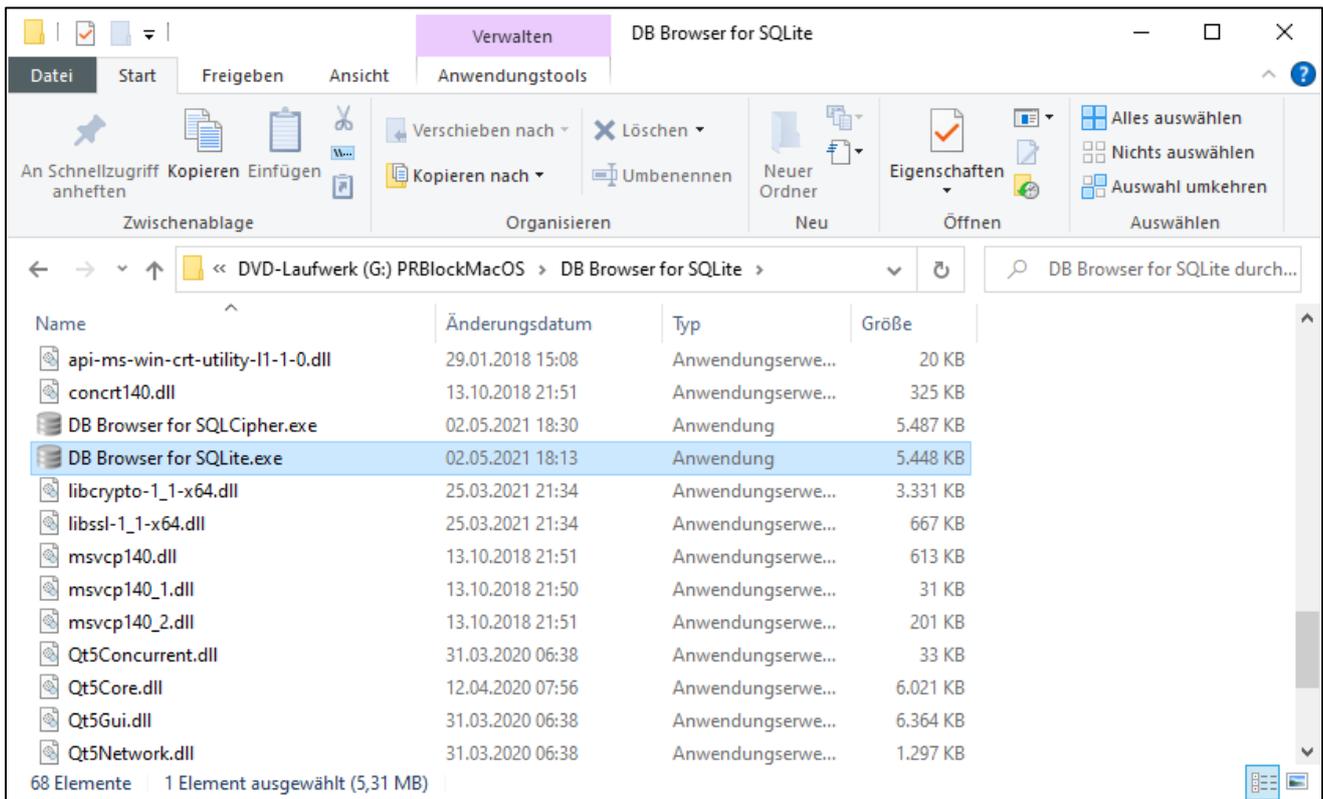
Exception log and Reports exported to:
'D:\macOS\FSEventOut\FSE_Reports'

C:\Users\John Doe>
```

Im Ausgabeverzeichnis erstellt der Parser drei Dateien. Ein Errorlog mit Fehleraufzeichnung während der Verarbeitung der Daten und eine CSV Datei sowie eine SQLite Datenbank mit den aufbereiteten FS-Events.



Zur Untersuchung der FSEvent Eintragungen kann nun die Datei **FSEvents.sqlite** herangezogen werden. Diese kann im **DB Browser for SQLite** geöffnet und eingesehen werden.



Hier können alle aufgezeichneten Dateisystem Events nachverfolgt werden.

The screenshot shows the DB Browser for SQLite interface. The main window displays a table named 'fsevents_sorted_by_event_id'. The table has five columns: 'id', 'id_hex', 'fullpath', 'filename', and 'type'. The data is sorted by event ID. The 'fullpath' column contains various system and user-related paths. The 'filename' column shows files like 'com.apple.runningboard', 'A0F838B4-54DC-48B...', 'KextPolicy-shm', '.fi8E6EFC6C', 'mdsDirectory.db', 'mds.lock', '.f1CF600F4B', 'mdsObject.db', and 'sqlindex-shm'. The 'type' column indicates the event type, such as 'Folder' and 'FileEv'. The interface includes a menu bar, a toolbar, and a right-hand pane for editing the selected cell.

id	id_hex	fullpath	filename	type
1	2	private/var/folders/zz/...	com.apple.runningboard	Folde
2	5	private/var/db/systemstats/A0F838B4-54DC-48B...	A0F838B4-54DC-48BC-...	FileEv
3	8	private/var/db/SystemPolicyConfiguration/...	KextPolicy-shm	FileEv
4	11	private/var/db/mds/system/.fi8E6EFC6C	.fi8E6EFC6C	FileEv
5	14	private/var/db/mds/system/mdsDirectory.db	mdsDirectory.db	FileEv
6	17	private/var/db/mds/system/mds.lock	mds.lock	FileEv
7	20	private/var/db/mds/system/.f1CF600F4B	.f1CF600F4B	FileEv
8	23	private/var/db/mds/system/mdsObject.db	mdsObject.db	FileEv
9	26	private/var/db/mds/system/.fi8E6EFC6C	.fi8E6EFC6C	FileEv
10	53	private/var/db/mds/system/...	mdsObject.db.sb-34c71ff2-finnYb	FileEv
11	55	private/var/db/mds/system/mdsObject.db	mdsObject.db	FileEv
12	58	private/var/db/mds/system/.f1CF600F4B	.f1CF600F4B	FileEv
13	67	private/var/db/dslocal/nodes/Default/sqlindex-shm	sqlindex-shm	FileEv
14	88	private/var/db/mds/system/...	mdsDirectory.db.sb-34c71ff2-rUWuin	FileEv

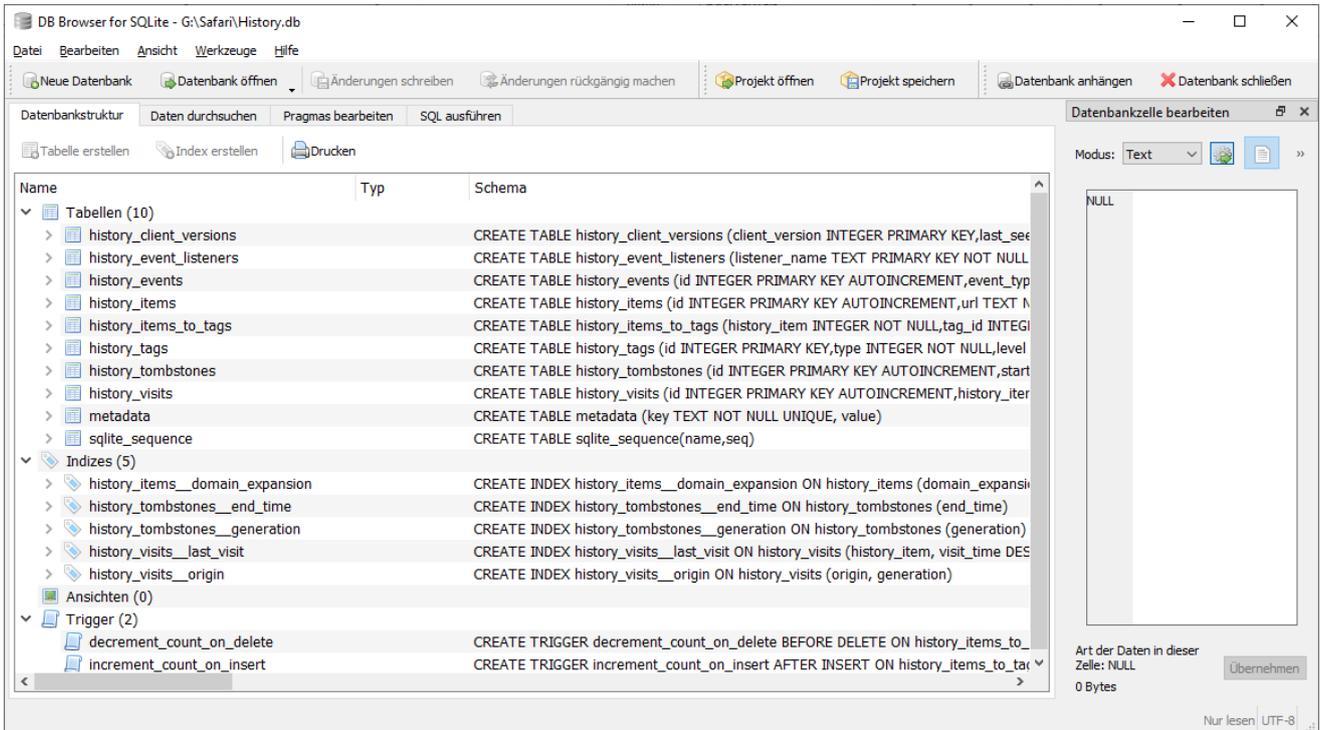
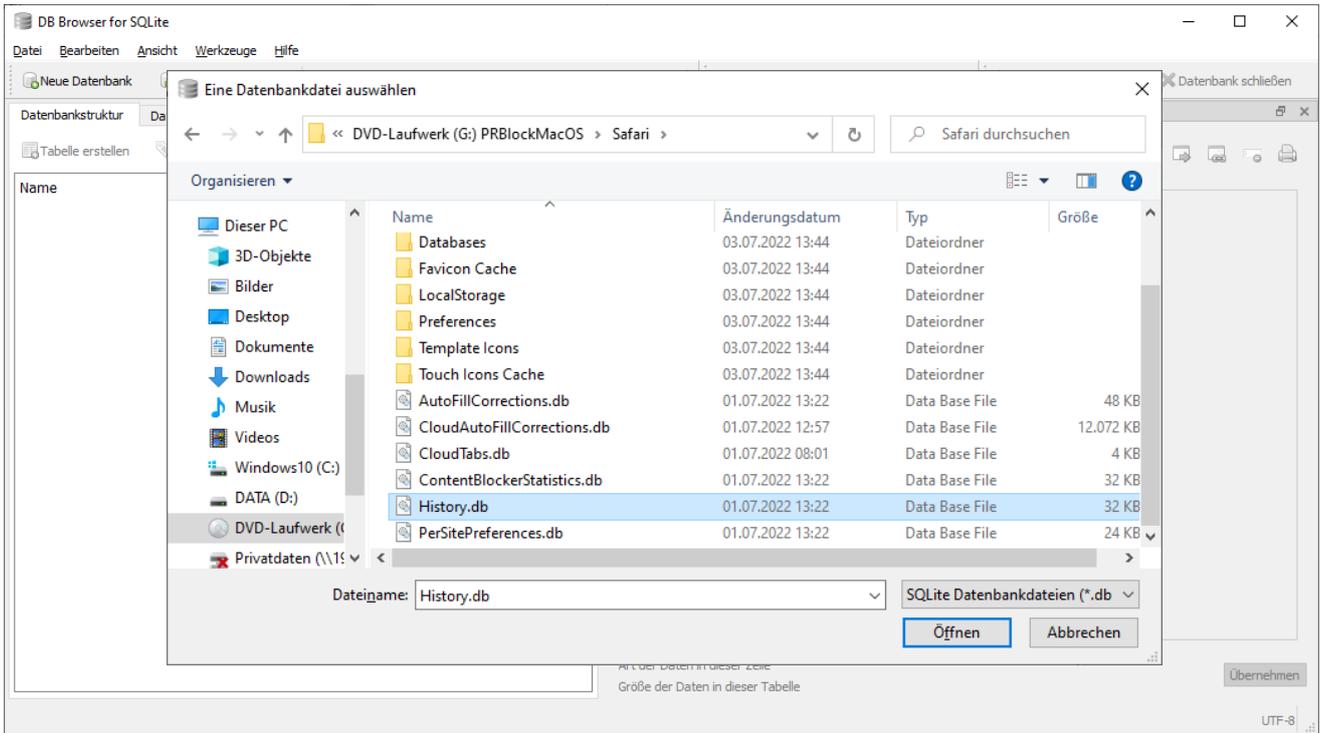
Es gibt sogar die Möglichkeit zu filtern, wie im Beispiel nach **Confidential**.

The screenshot shows the DB Browser for SQLite interface with a filter applied to the 'fullpath' column. The filter text 'confidential' is entered in the 'In allen Spalten filtern' field. The table now displays only three rows where the 'fullpath' contains 'Users/nutzer1/Documents/CFMM/Research/...'. The 'filename' column shows 'Confidential' and 'CFFMediaFormat-20100810.doc.pdf'. The 'type' column indicates 'FolderEv' and 'FileEven'. The interface includes a menu bar, a toolbar, and a right-hand pane for editing the selected cell.

id	id_hex	fullpath	filename	type
1	856...	Users/nutzer1/Documents/CFMM/Research/...	Confidential	FolderEv
2	865...	Users/nutzer1/Documents/CFMM/Research/...	CFFMediaFormat-20100810.doc.pdf	FileEven
3	866...	Users/nutzer1/Documents/CFMM/Research/...	CFFMediaFormat-20100810.doc.pdf	FileEven

3.3. UNTERSUCHUNG DER SAFARI HISTORY MIT SQLITE

Zur Untersuchung des Safari Browserverlaufs kann die Datei **History.db** herangezogen werden. Diese kann im **DB Browser for SQLite** geöffnet und eingesehen werden.



Angabe der Tabellen Einträge mit **SELECT** mit wenig verwertbaren forensischen Inhalten.

```
select * from history_items;
```

The screenshot shows the DB Browser for SQLite interface. The main window displays the following SQL query in the editor:

```
1 select * from history_items;
```

Below the editor, the results of the query are shown in a table with 7 columns: id, url, domain_expansion, visit_count, daily_visit_counts, weekly_visit_counts, and autoc. The table contains 6 rows of data:

	id	url	domain_expansion	visit_count	daily_visit_counts	weekly_visit_counts	autoc
1	1	https://www.google.de/?...	NULL	2	BLOB	NULL	NULL
2	2	http://mac4n6.com/	mac4n6	1	BLOB	NULL	NULL
3	3	http://www.mac4n6.com/	mac4n6	1	BLOB	NULL	NULL
4	4	http://hs-mittweida.de/	NULL	1	BLOB	NULL	NULL
5	5	https://www.hs-mittweida.de/	NULL	2	BLOB	NULL	NULL
6	6	http://bka.bund.de/	NULL	1	BLOB	NULL	NULL

Below the table, a status message reads: "Ausführung wurde ohne Fehler beendet. Ergebnis: 20 Zeilen in 12ms zurückgegeben. In Zeile 1: select * from history_items;".

On the right side, the "Datenbankzelle bearbeiten" panel shows the selected cell content as "NULL". Below this, it indicates "Art der Daten in dieser Zeile: NULL" and "0 Bytes".

Eine Ausgabe die mehrere Tabellen zusammengefasst (URL, Titel und Abrufdatum) erhält man mit **INNER JOIN**.

```
SELECT  visit_time, title, url
FROM
    history_visits
INNER JOIN
    history_items ON
    history_items.id = history_visits.history_item;
```

The screenshot shows the DB Browser for SQLite interface. The SQL editor contains the following query:

```
1 SELECT
2   visit_time, title, url
3 FROM
4   history_visits
5 INNER JOIN
6   history_items ON
7   history_items.id = history_visits.history_item;
```

The results are displayed in a table with the following columns: visit_time, title, and url. The data is as follows:

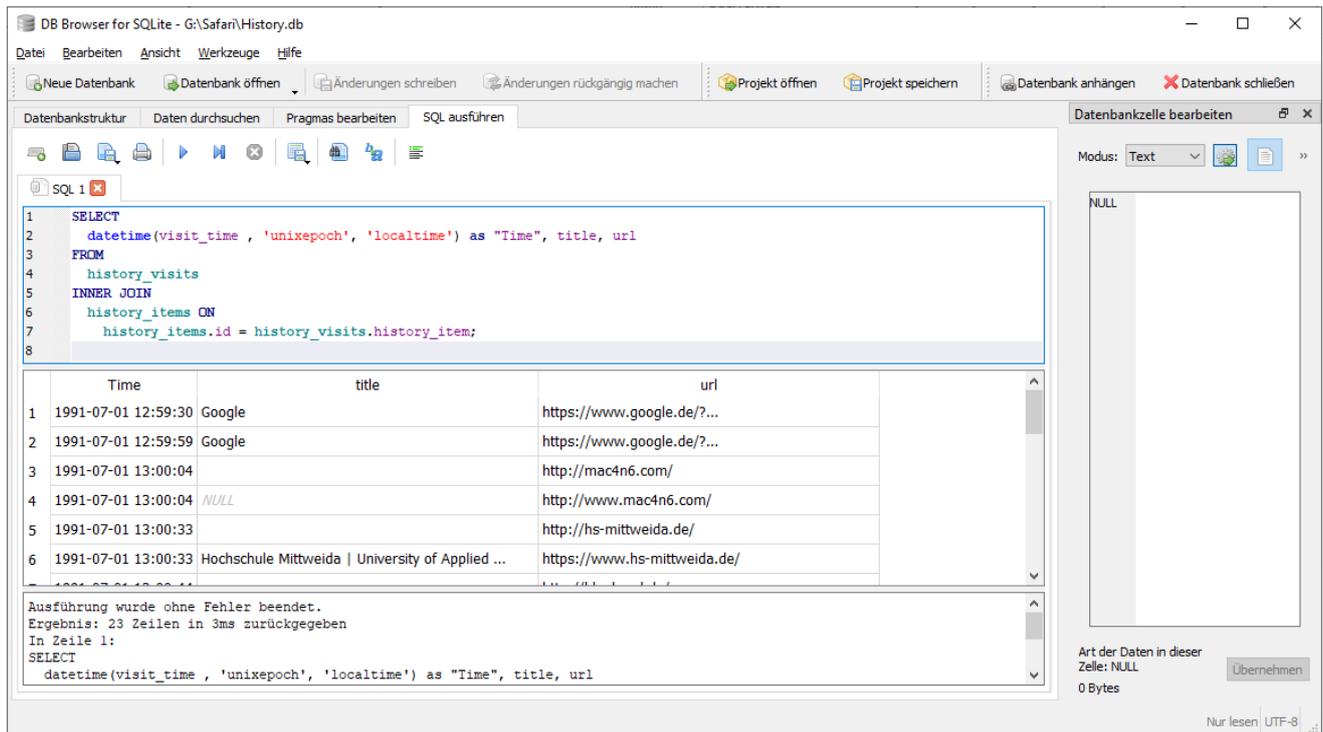
	visit_time	title	url
1	678365970.452298	Google	https://www.google.de/?...
2	678365999.00924	Google	https://www.google.de/?...
3	678366004.470999		http://mac4n6.com/
4	678366004.561832	NULL	http://www.mac4n6.com/
5	678366033.962511		http://hs-mittweida.de/
6	678366033.977679	Hochschule Mittweida University of Applied ...	https://www.hs-mittweida.de/

Below the table, a status message reads: "Ausführung wurde ohne Fehler beendet. Ergebnis: 23 Zeilen in 4ms zurückgegeben. In Zeile 1: SELECT visit_time, title, url".

On the right side, the "Datenbankzelle bearbeiten" panel shows "Modus: Text" and "Art der Daten in dieser Zelle: NULL" with "0 Bytes".

Eine Ausgabe der Datumsformate erreicht man mittels `datetime()` Funktion.

```
SELECT
    datetime(visit_time , 'unixepoch', 'localtime') as "Time", title, url
FROM
    history_visits
INNER JOIN
    history_items ON
        history_items.id = history_visits.history_item;
```



The screenshot shows the DB Browser for SQLite interface. The SQL editor contains the following query:

```
1 SELECT
2     datetime(visit_time , 'unixepoch', 'localtime') as "Time", title, url
3 FROM
4     history_visits
5 INNER JOIN
6     history_items ON
7     history_items.id = history_visits.history_item;
8
```

The results table displays the following data:

	Time	title	url
1	1991-07-01 12:59:30	Google	https://www.google.de/?...
2	1991-07-01 12:59:59	Google	https://www.google.de/?...
3	1991-07-01 13:00:04		http://mac4n6.com/
4	1991-07-01 13:00:04	NULL	http://www.mac4n6.com/
5	1991-07-01 13:00:33		http://hs-mittweida.de/
6	1991-07-01 13:00:33	Hochschule Mittweida University of Applied ...	https://www.hs-mittweida.de/

The status bar at the bottom indicates: "Ausführung wurde ohne Fehler beendet. Ergebnis: 23 Zeilen in 3ms zurückgegeben. In Zeile 1: SELECT datetime(visit_time , 'unixepoch', 'localtime') as "Time", title, url".

Die Zeitstempel scheinen nicht ganz zu stimmen, betrachtet man das Erscheinungsdatum von macOS BigSur.

Datetime liefert eine **unixepoch Zeit** als Sekunden seit dem 01.01.1970. Die **macepoch Zeit** basiert auf dem 01.01.2001 sozusagen **978.307.200** Sekunden später als die unixepoch Zeit.

Daher lässt sich der korrekte Zeitstempel leicht berechnen:

$$\text{macepoch (cocoa touch / apple base time)} = \text{'unixepoch'} + 978307200$$

```
SELECT
    datetime(visit_time + 978307200, 'unixepoch', 'localtime') as "Time", title, url
FROM
    history_visits
INNER JOIN
    history_items ON
    history_items.id = history_visits.history_item;
```

The screenshot shows a SQLite database browser window titled "DB Browser for SQLite - G:\Safari\History.db". The main window contains a SQL editor with the following query:

```
1 SELECT
2   datetime(visit_time + 978307200, 'unixepoch', 'localtime') as "Time", title, url
3 FROM
4   history_visits
5 INNER JOIN
6   history_items ON
7   history_items.id = history_visits.history_item;
8
```

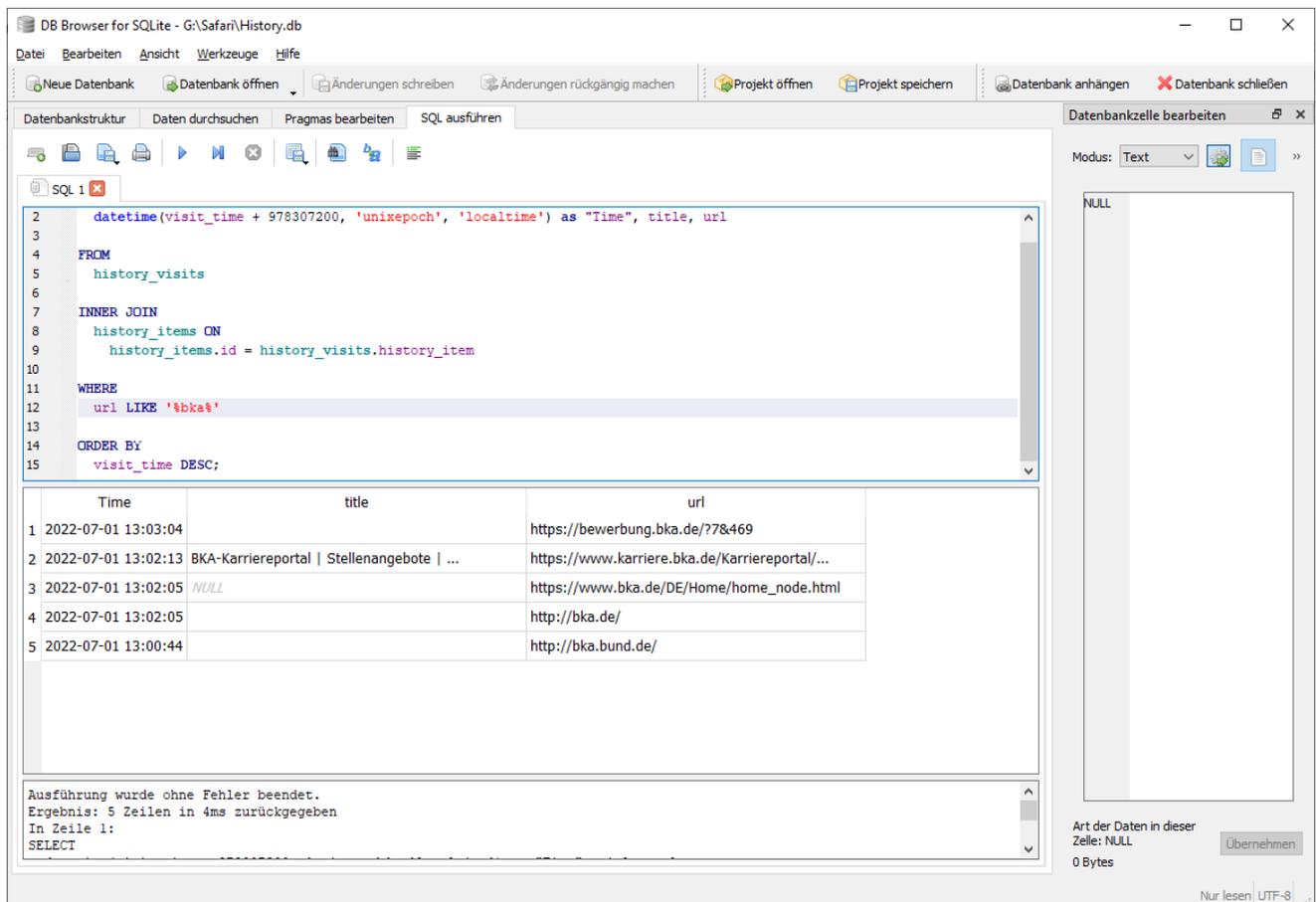
Below the editor is a table with the following data:

	Time	title	url
1	2022-07-01 12:59:30	Google	https://www.google.de/?...
2	2022-07-01 12:59:59	Google	https://www.google.de/?...
3	2022-07-01 13:00:04		http://mac4n6.com/
4	2022-07-01 13:00:04	NULL	http://www.mac4n6.com/
5	2022-07-01 13:00:33		http://hs-mittweida.de/
6	2022-07-01 13:00:33	Hochschule Mittweida University of Applied ...	https://www.hs-mittweida.de/

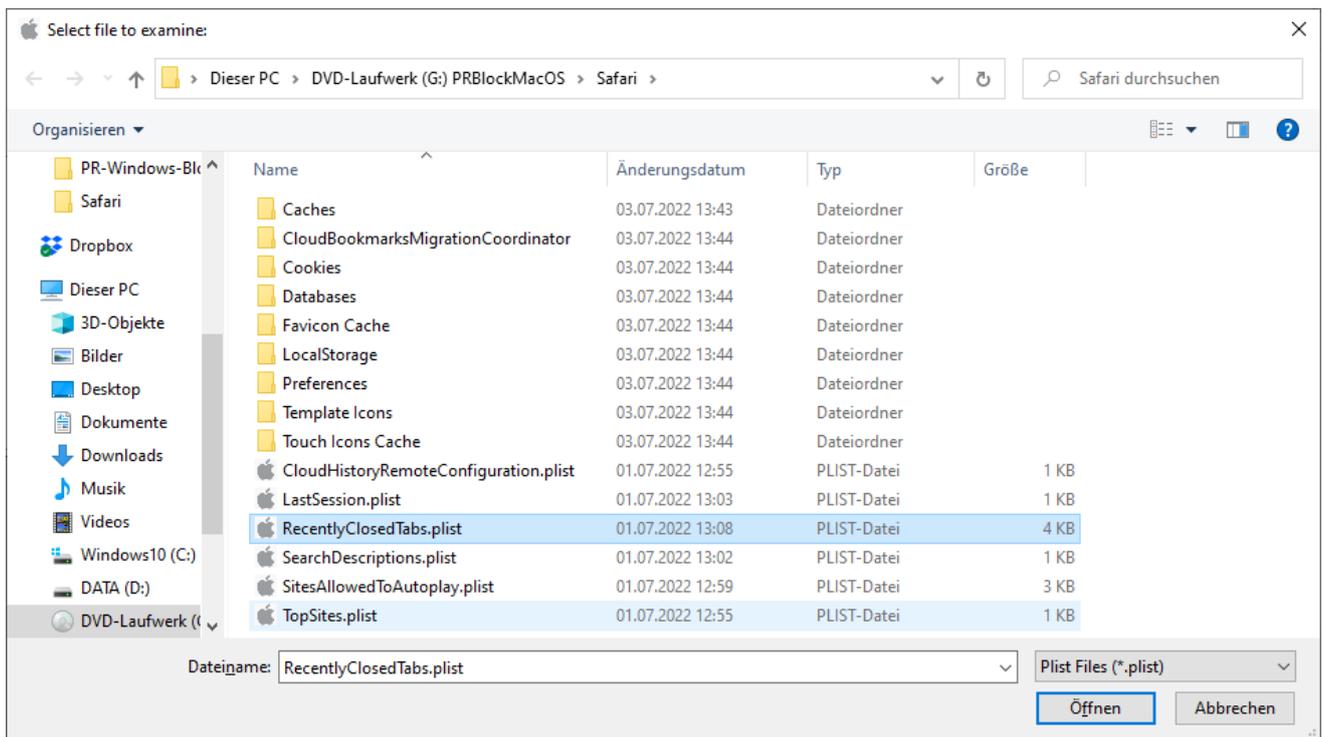
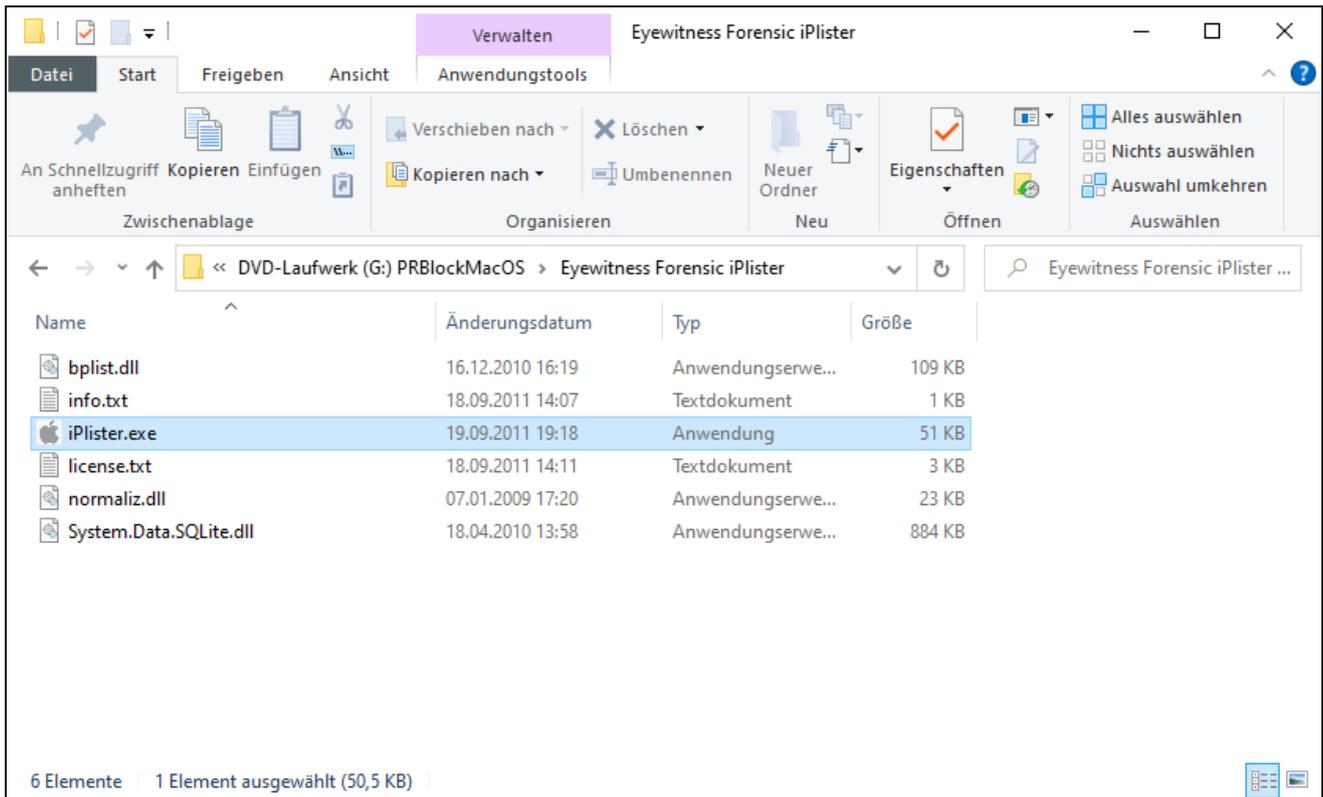
At the bottom of the window, a status bar indicates: "Ausführung wurde ohne Fehler beendet. Ergebnis: 23 Zeilen in 4ms zurückgegeben. In Zeile 1: SELECT datetime(visit_time + 978307200, 'unixepoch', 'localtime') as "Time", title, url".

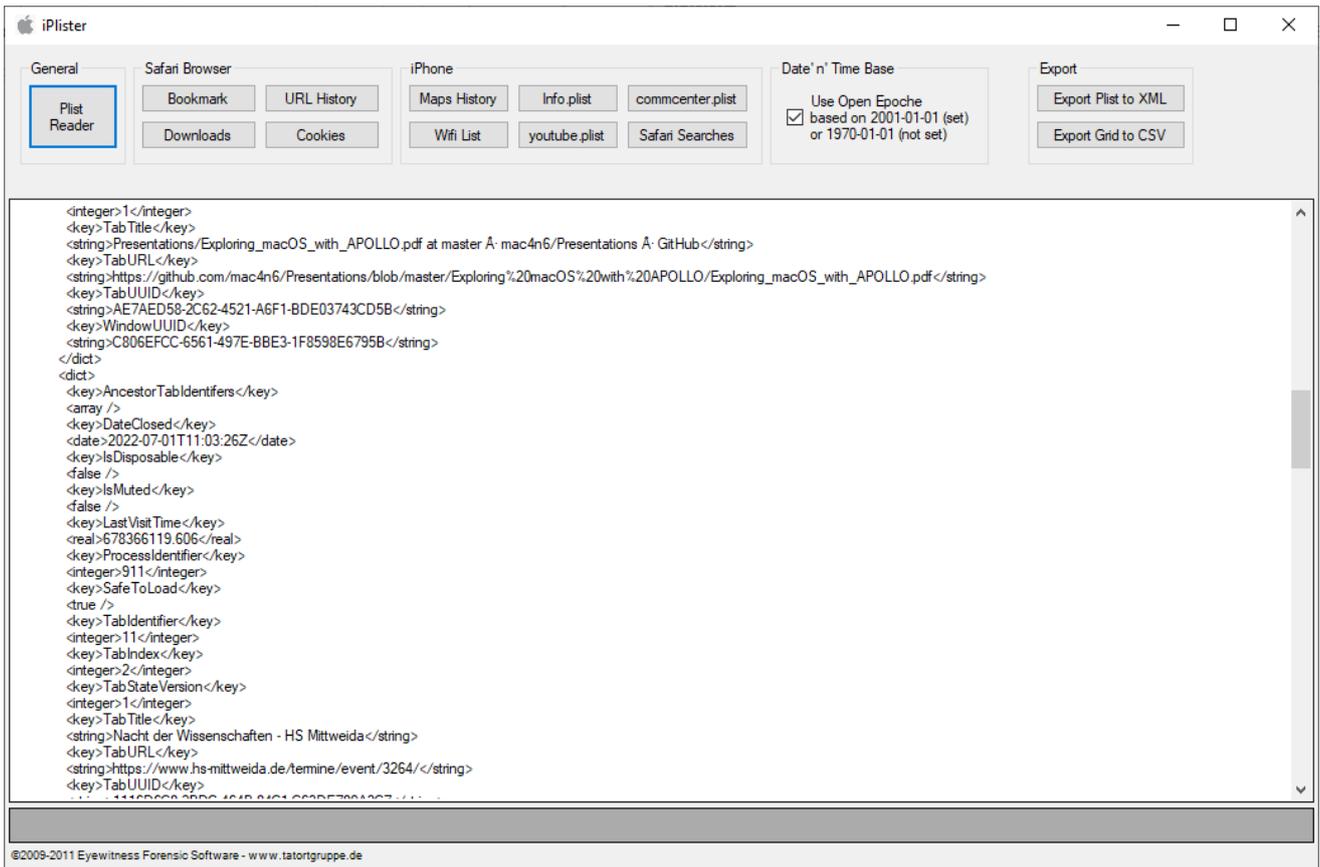
Mit den Abfragen ist auch eine **gezielte Suche** nach Inhalten möglich, im Beispiel **nach BKA** zeitlich geordnet:

```
SELECT
    datetime(visit_time + 978307200, 'unixepoch', 'localtime') as "Time", title, url
FROM
    history_visits
INNER JOIN
    history_items ON
        history_items.id = history_visits.history_item
WHERE
    url LIKE '%bka%'
ORDER BY
    visit_time DESC;
```

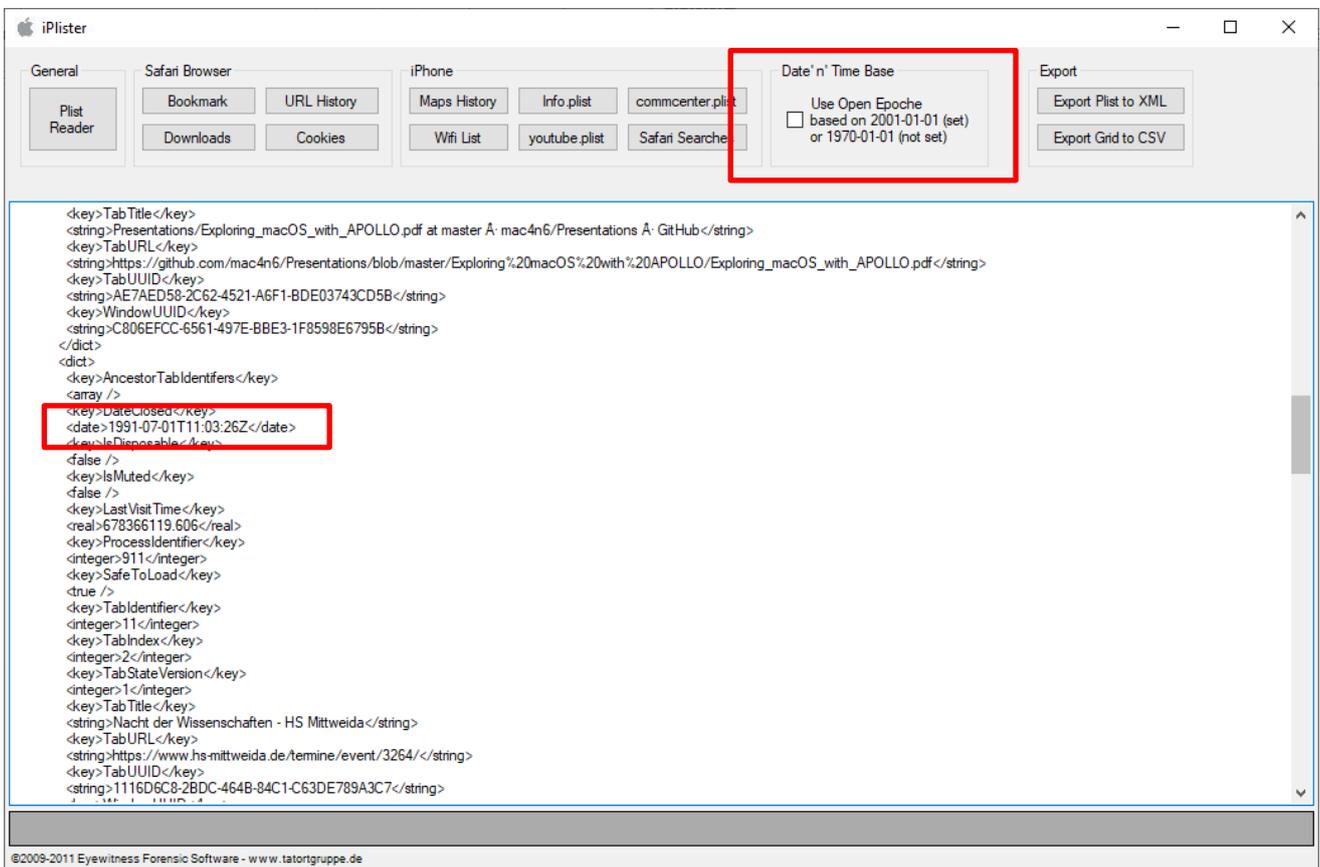


Zudem können mit dem iPlister die Binary Plist Dateien im Safari Verzeichnis ebenfalls noch eingesehen werden:



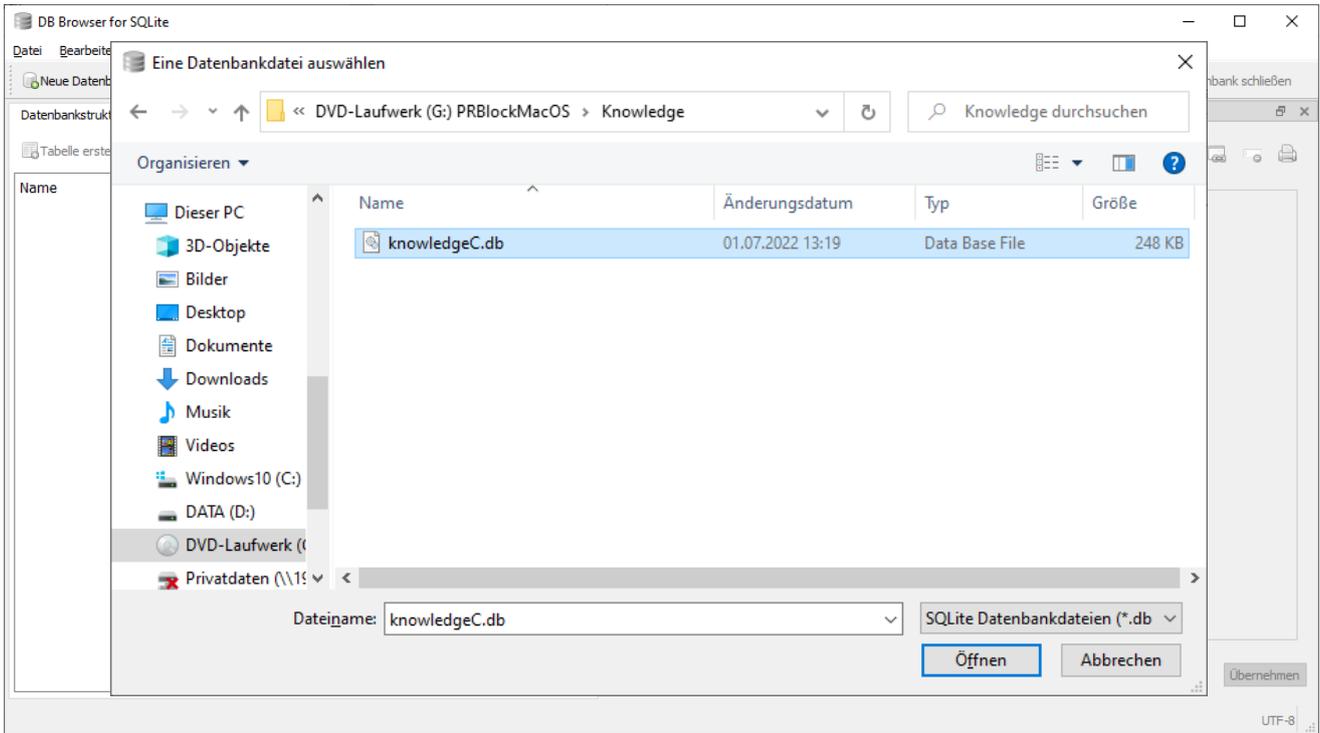


Eine Umschaltung der Datumsformate zwischen Unix Zeit und Mac Base Zeit ist möglich (Datei erneut einlesen erforderlich).

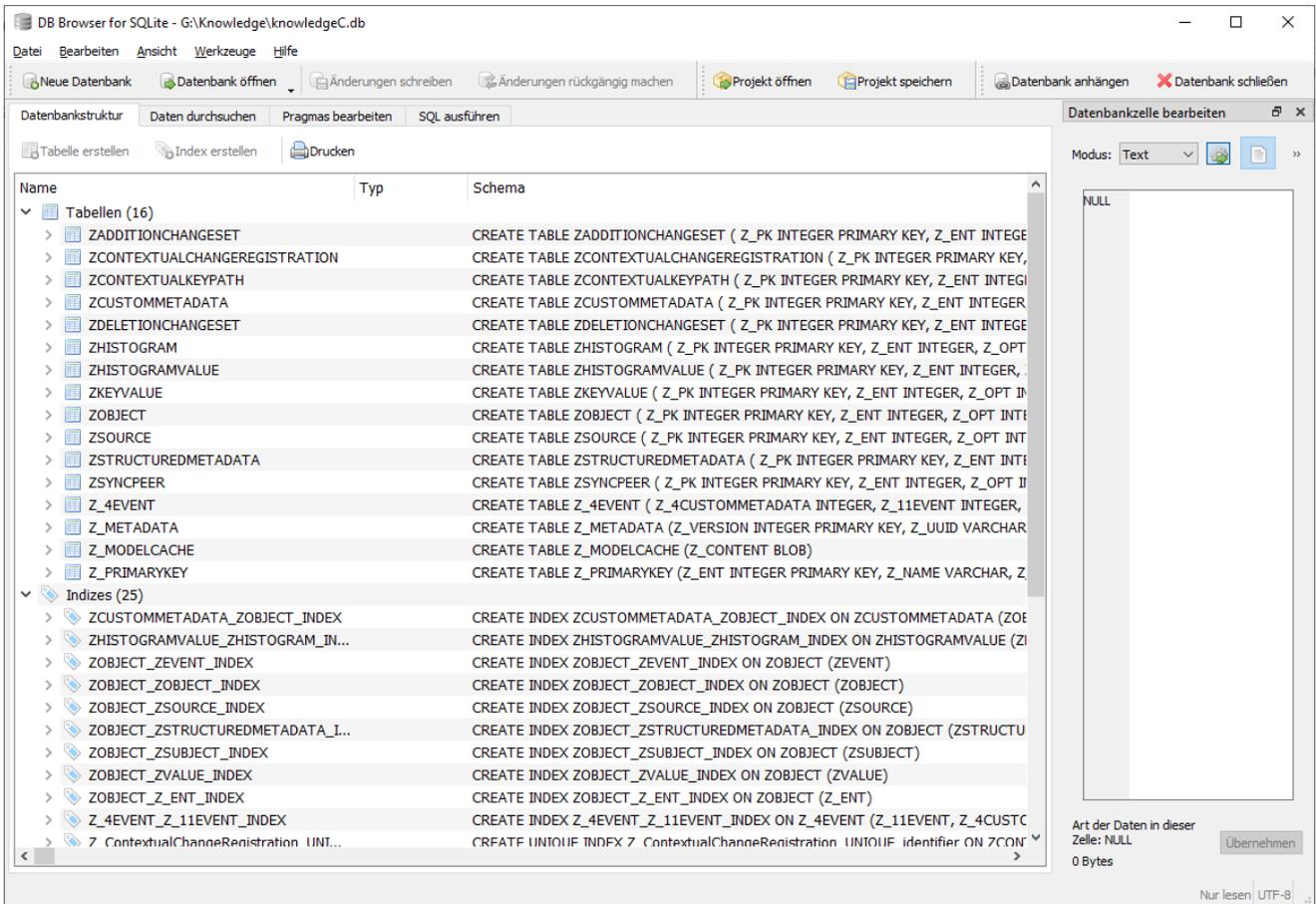


3.4. UNTERSUCHUNG DER DATENBANKDATEI KNOWLEDGE.CDB MIT SQLITE

Zur Untersuchung der Pattern of Life Informationen kann die Datei **KnowledgeC.db** herangezogen werden. Diese kann im **DB Browser for SQLite** geöffnet und eingesehen werden.



Die Datenbank beinhaltet sehr viele zusammenzuführende Tabellen.



Folgender SQL Befehl stellt die **wichtigsten Eintragungen im Überblick** dar:

```
SELECT
datetime(ZOBJECT.ZCREATIONDATE+978307200,'UNIXEPOCH','LOCALTIME') as "ENTRY
CREATION",
CASE ZOBJECT.ZSTARTDAYOFWEEK
WHEN "1" THEN "Sunday"
WHEN "2" THEN "Monday"
WHEN "3" THEN "Tuesday"
WHEN "4" THEN "Wednesday"
WHEN "5" THEN "Thursday"
WHEN "6" THEN "Friday"
WHEN "7" THEN "Saturday"
END "DAY OF WEEK",
datetime(ZOBJECT.ZSTARTDATE+978307200,'UNIXEPOCH','LOCALTIME') as "START",
datetime(ZOBJECT.ZENDDATE+978307200,'UNIXEPOCH','LOCALTIME') as "END",
(ZOBJECT.ZENDDATE-ZOBJECT.ZSTARTDATE) as "USAGE IN SECONDS",
ZOBJECT.ZSTREAMNAME,
ZOBJECT.ZVALUESTRING
FROM ZOBJECT
ORDER BY "START"
```

The screenshot shows a SQLite database browser window with the following SQL query executed:

```
1 SELECT
2 datetime(ZOBJECT.ZCREATIONDATE+978307200,'UNIXEPOCH','LOCALTIME') as "ENTRY CREATION",
3 CASE ZOBJECT.ZSTARTDAYOFWEEK
4 WHEN "1" THEN "Sunday"
5 WHEN "2" THEN "Monday"
6 WHEN "3" THEN "Tuesday"
7 WHEN "4" THEN "Wednesday"
8 WHEN "5" THEN "Thursday"
9 WHEN "6" THEN "Friday"
10 WHEN "7" THEN "Saturday"
11 END "DAY OF WEEK",
12 datetime(ZOBJECT.ZSTARTDATE+978307200,'UNIXEPOCH','LOCALTIME') as "START",
13 datetime(ZOBJECT.ZENDDATE+978307200,'UNIXEPOCH','LOCALTIME') as "END",
14 (ZOBJECT.ZENDDATE-ZOBJECT.ZSTARTDATE) as "USAGE IN SECONDS",
15 ZOBJECT.ZSTREAMNAME,
16 ZOBJECT.ZVALUESTRING
17 FROM ZOBJECT
18 ORDER BY "START"
```

The results are displayed in a table with the following columns: ENTRY CREATION, DAY OF WEEK, START, END, USAGE IN SECONDS, ZSTREAMNAME, and ZVALUESTRING. The table contains 9 rows of data.

ENTRY CREATION	DAY OF WEEK	START	END	USAGE IN SECONDS	ZSTREAMNAME	ZVALUESTRING
2022-07-01 08:03:54	Thursday	2022-07-01 08:03:54	2022-07-01 08:03:54	0	/notification/usage	Receive
2022-07-01 08:57:19	Thursday	2022-07-01 08:55:24	2022-07-01 08:57:16	112	/display/isBacklit	NULL
2022-07-01 08:58:35	Thursday	2022-07-01 08:57:16	2022-07-01 08:58:32	76	/display/isBacklit	NULL
2022-07-01 09:04:46	Thursday	2022-07-01 08:58:32	2022-07-01 09:04:44	372	/display/isBacklit	NULL
2022-07-01 09:04:45	Thursday	2022-07-01 08:58:39	2022-07-01 09:04:45	366	/app/usage	com.apple.finder
2022-07-01 09:05:21	Friday	2022-07-01 09:04:44	2022-07-01 09:05:20	36	/display/isBacklit	NULL
2022-07-01 09:06:15	Friday	2022-07-01 09:05:20	2022-07-01 09:06:11	51	/app/usage	com.apple.finder
2022-07-01 09:13:04	Friday	2022-07-01 09:06:11	2022-07-01 09:13:03	412	/app/usage	com.apple.DiskUtility
2022-07-01 09:13:21	Friday	2022-07-01 09:13:03	2022-07-01 09:13:12	9	/app/usage	com.apple.finder

The execution status at the bottom indicates: "Ausführung wurde ohne Fehler beendet. Ergebnis: 56 Zeilen in 15ms zurückgegeben. In Zeile 1: SELECT".

Folgender SQL Befehl zeigt die **Nutzung der Apps** an:

```

SELECT
datetime(ZOBJECT.ZCREATIONDATE+978307200,'UNIXEPOCH', 'LOCALTIME') as "ENTRY
CREATION",
CASE ZOBJECT.ZSTARTDAYOFWEEK
WHEN "1" THEN "Sunday"
WHEN "2" THEN "Monday"
WHEN "3" THEN "Tuesday"
WHEN "4" THEN "Wednesday"
WHEN "5" THEN "Thursday"
WHEN "6" THEN "Friday"
WHEN "7" THEN "Saturday"
END "DAY OF WEEK",
datetime(ZOBJECT.ZSTARTDATE+978307200,'UNIXEPOCH', 'LOCALTIME') as "START",
datetime(ZOBJECT.ZENDDATE+978307200,'UNIXEPOCH', 'LOCALTIME') as "END",
(ZOBJECT.ZENDDATE-ZOBJECT.ZSTARTDATE) as "USAGE IN SECONDS",
ZOBJECT.ZSTREAMNAME,
ZOBJECT.ZVALUESTRING,
ZSTRUCTUREDMETADATA.Z_DKAPPLICATIONACTIVITYMETADATAKEY__ACTIVITYTYPE AS "ACTIVITY
TYPE",
ZSTRUCTUREDMETADATA.Z_DKAPPLICATIONACTIVITYMETADATAKEY__TITLE as "TITLE",
ZSTRUCTUREDMETADATA.Z_DKAPPLICATIONACTIVITYMETADATAKEY__USERACTIVITYREQUIREDSTRING as
"ACTIVITY STRING",
datetime(ZSTRUCTUREDMETADATA.Z_DKAPPLICATIONACTIVITYMETADATAKEY__EXPIRATIONDATE+978307
200,'UNIXEPOCH', 'LOCALTIME') as "EXPIRATION DATE"
FROM ZOBJECT
left join ZSTRUCTUREDMETADATA on ZOBJECT.ZSTRUCTUREDMETADATA =
ZSTRUCTUREDMETADATA.Z_PK
WHERE ZSTREAMNAME is "/app/usage"
ORDER BY "START"

```

The screenshot shows the DB Browser for SQLite interface. The SQL query is entered in the editor and has been executed. The results are displayed in a table with 11 columns: ENTRY CREATION, DAY OF WEEK, START, END, USAGE IN SECONDS, ZSTREAMNAME, ZVALUESTRING, ACTIVITY TYPE, TITLE, and ACTIVITY STRING. The results show 8 rows of data for the date 2022-07-01, with various activities like 'finder', 'DiskUtility', and 'systempreferences' being used.

	ENTRY CREATION	DAY OF WEEK	START	END	USAGE IN SECONDS	ZSTREAMNAME	ZVALUESTRING	ACTIVITY TYPE	TITLE	ACTIVITY STRING
1	2022-07-01 09:04:45	Thursday	2022-07-01 08:58:39	2022-07-01 09:04:45	366	/app/usage	com.apple.finder	NULL	NULL	NULL
2	2022-07-01 09:06:15	Friday	2022-07-01 09:05:20	2022-07-01 09:06:11	51	/app/usage	com.apple.finder	NULL	NULL	NULL
3	2022-07-01 09:13:04	Friday	2022-07-01 09:06:11	2022-07-01 09:13:03	412	/app/usage	com.apple.DiskUtility	NULL	NULL	NULL
4	2022-07-01 09:13:21	Friday	2022-07-01 09:13:03	2022-07-01 09:13:12	9	/app/usage	com.apple.finder	NULL	NULL	NULL
5	2022-07-01 09:16:53	Friday	2022-07-01 09:13:12	2022-07-01 09:16:53	221	/app/usage	com.apple.systempreferences	NULL	NULL	NULL
6	2022-07-01 09:16:55	Friday	2022-07-01 09:16:53	2022-07-01 09:16:54	1	/app/usage	com.apple.DiskUtility	NULL	NULL	NULL
7	2022-07-01 09:17:53	Friday	2022-07-01 09:16:54	2022-07-01 09:17:52	58	/app/usage	com.apple.systempreferences	NULL	NULL	NULL
8	2022-07-01 09:17:58	Friday	2022-07-01 09:17:52	2022-07-01 09:17:58	6	/app/usage	com.apple.DiskUtility	NULL	NULL	NULL

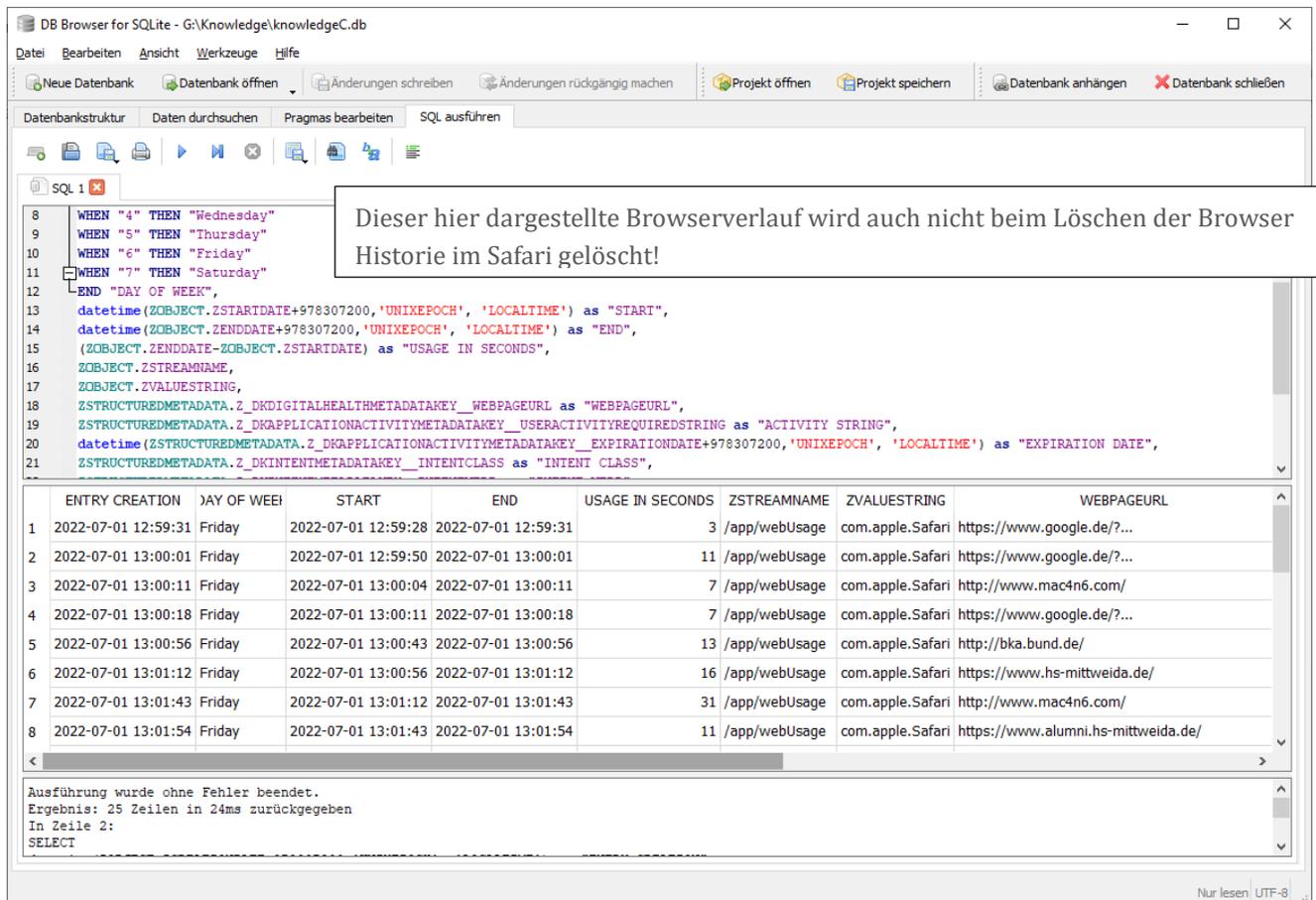
Ausführung wurde ohne Fehler beendet.
Ergebnis: 20 Zeilen in 10ms zurückgegeben
In Zeile 1:
SELECT

Folgender SQL Befehl zeigt die **Nutzung des Safari Browsers** und deren **aufgerufener URL** an:

```

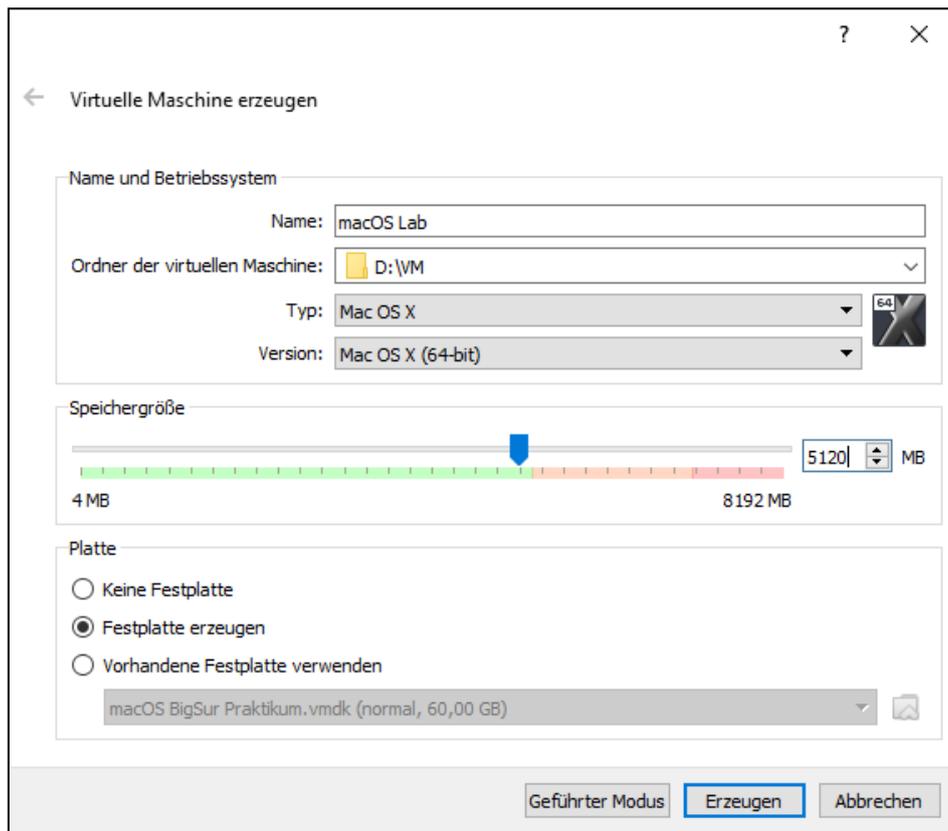
SELECT
datetime(ZOBJECT.ZCREATIONDATE+978307200,'UNIXEPOCH', 'LOCALTIME') as "ENTRY
CREATION",
CASE ZOBJECT.ZSTARTDAYOFWEEK
WHEN "1" THEN "Sunday"
WHEN "2" THEN "Monday"
WHEN "3" THEN "Tuesday"
WHEN "4" THEN "Wednesday"
WHEN "5" THEN "Thursday"
WHEN "6" THEN "Friday"
WHEN "7" THEN "Saturday"
END "DAY OF WEEK",
datetime(ZOBJECT.ZSTARTDATE+978307200,'UNIXEPOCH', 'LOCALTIME') as "START",
datetime(ZOBJECT.ZENDDATE+978307200,'UNIXEPOCH', 'LOCALTIME') as "END",
(ZOBJECT.ZENDDATE-ZOBJECT.ZSTARTDATE) as "USAGE IN SECONDS",
ZOBJECT.ZSTREAMNAME,
ZOBJECT.ZVALUESTRING,
ZSTRUCTUREDMETADATA.Z_DKDIGITALHEALTHMETADATAKEY__WEBPAGEURL as "WEBPAGEURL",
ZSTRUCTUREDMETADATA.Z_DKAPPLICATIONACTIVITYMETADATAKEY__USERACTIVITYREQUIREDSTRING as
"ACTIVITY STRING",
datetime(ZSTRUCTUREDMETADATA.Z_DKAPPLICATIONACTIVITYMETADATAKEY__EXPIRATIONDATE+978307
200,'UNIXEPOCH', 'LOCALTIME') as "EXPIRATION DATE",
ZSTRUCTUREDMETADATA.Z_DKINTENTMETADATAKEY__INTENTCLASS as "INTENT CLASS",
ZSTRUCTUREDMETADATA.Z_DKINTENTMETADATAKEY__INTENTVERB as "INTENT VERB",
ZSTRUCTUREDMETADATA.Z_DKINTENTMETADATAKEY__SERIALIZEDINTERACTION as "SERIALIZED
INTERACTION",
ZSOURCE.ZBUNDLEID
FROM ZOBJECT
left join ZSTRUCTUREDMETADATA on ZOBJECT.ZSTRUCTUREDMETADATA =
ZSTRUCTUREDMETADATA.Z_PK
left join ZSOURCE on ZOBJECT.ZSOURCE = ZSOURCE.Z_PK
WHERE ZSTREAMNAME is "/app/webUsage"
ORDER BY "START"

```

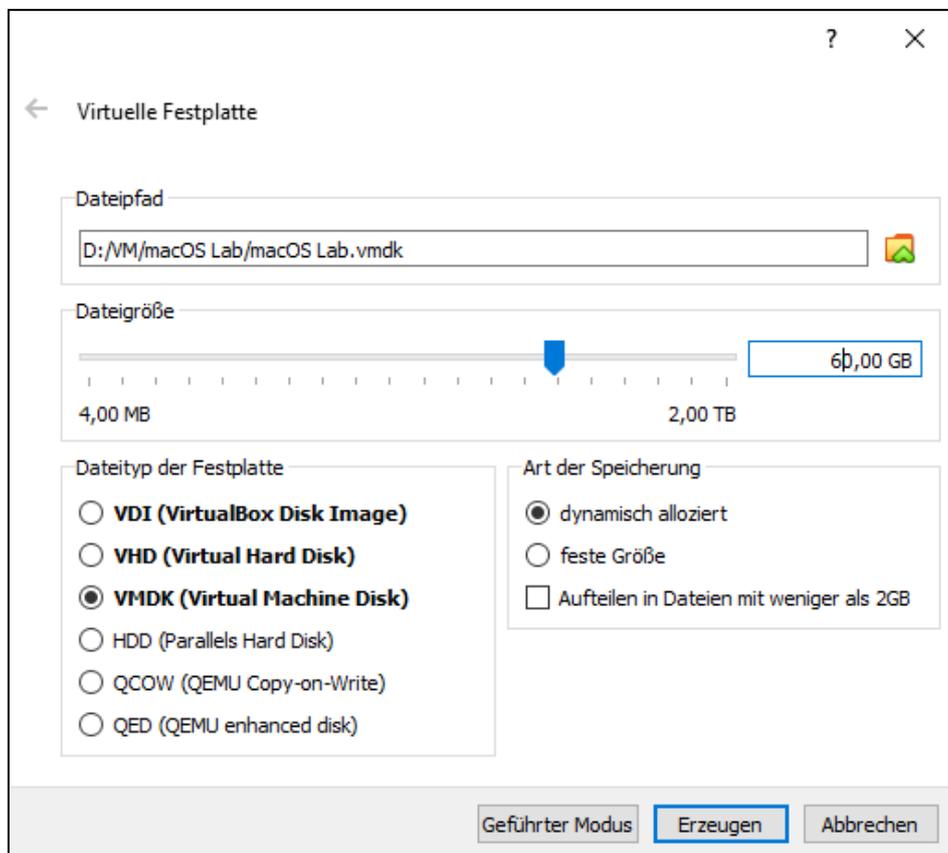


4. ERSTELLUNG EINER MACOS LAB VM IN VIRTUALBOX (UNTER WINDOWS)

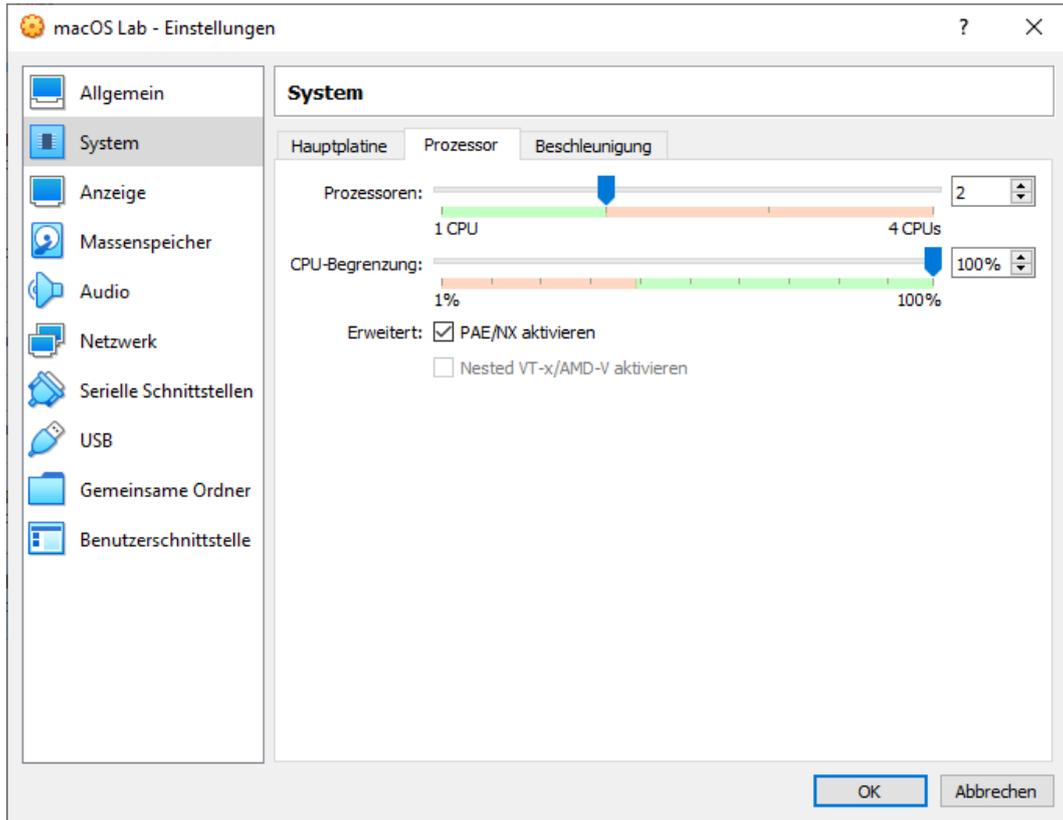
In VirtualBox **Neu** wählen und folgende Einstellungen vornehmen (wichtig mindestens 4GB RAM).



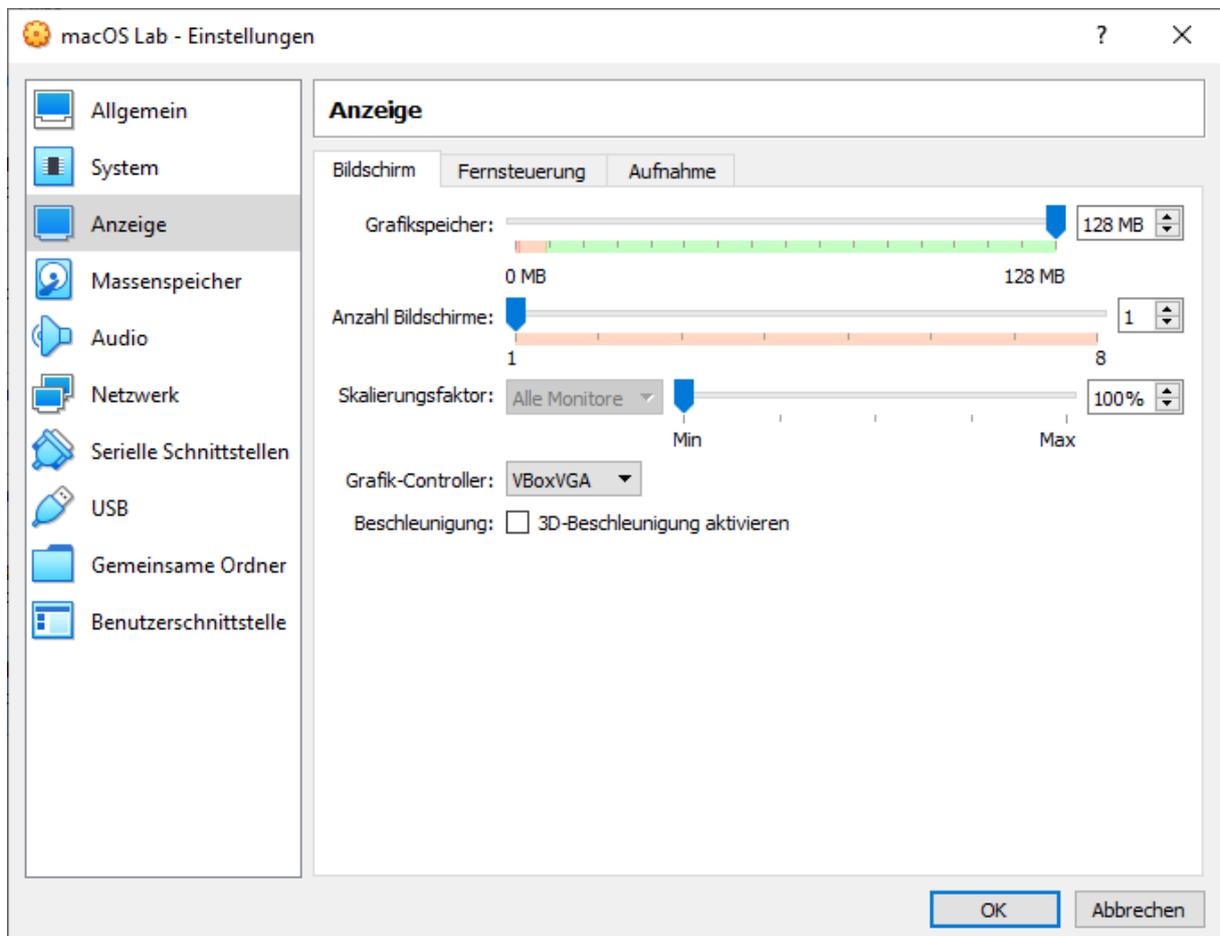
Festplatte erzeugen als VMDK und dynamische Größe mindestens 60 GB.



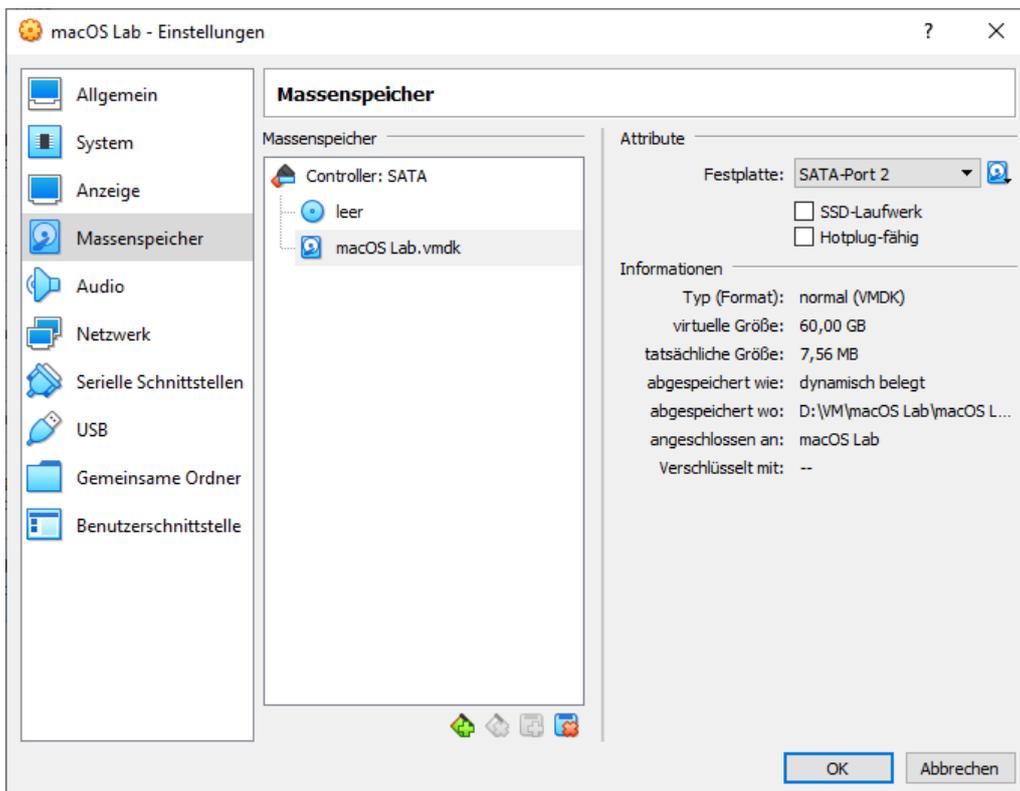
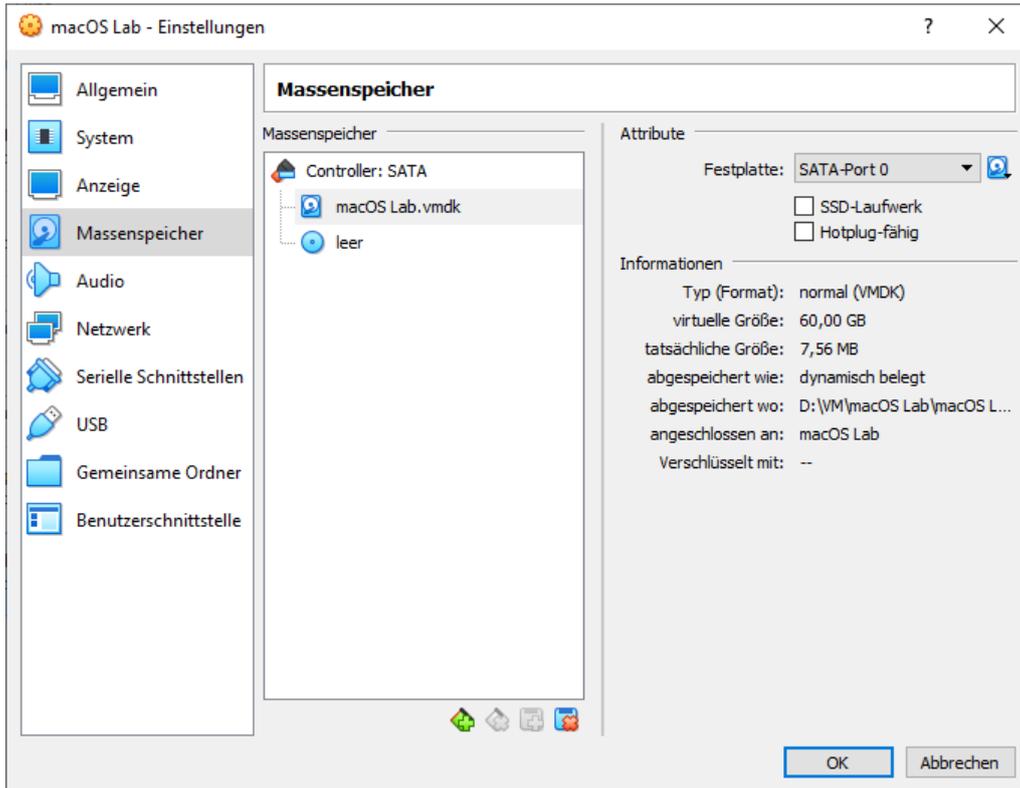
System Einstellung minimal 2 CPUs



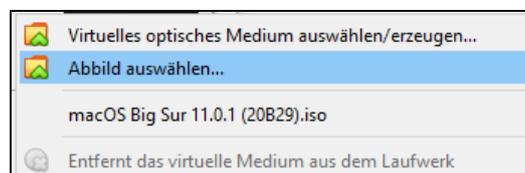
Anzeige Einstellunge Grphik Speicher 128 MB.

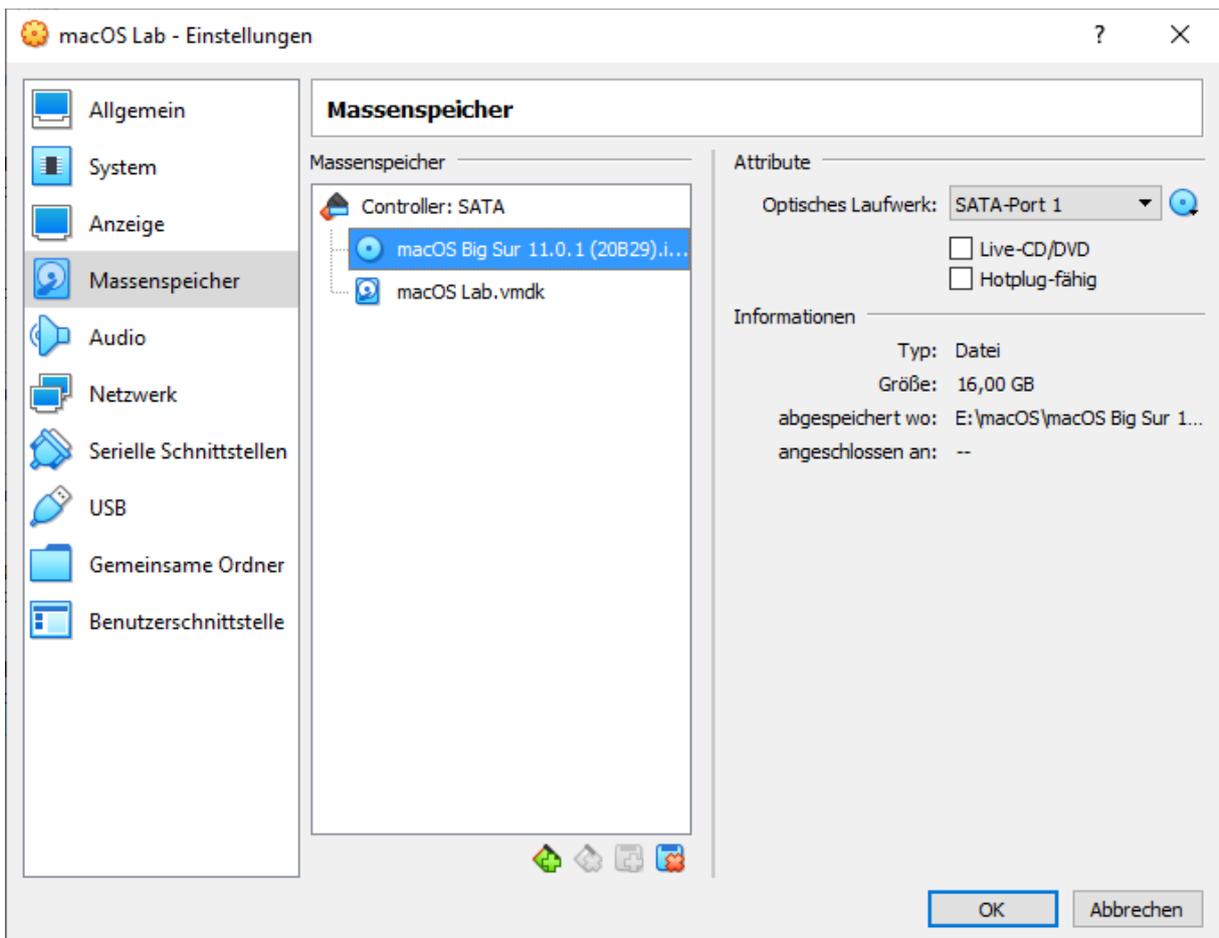
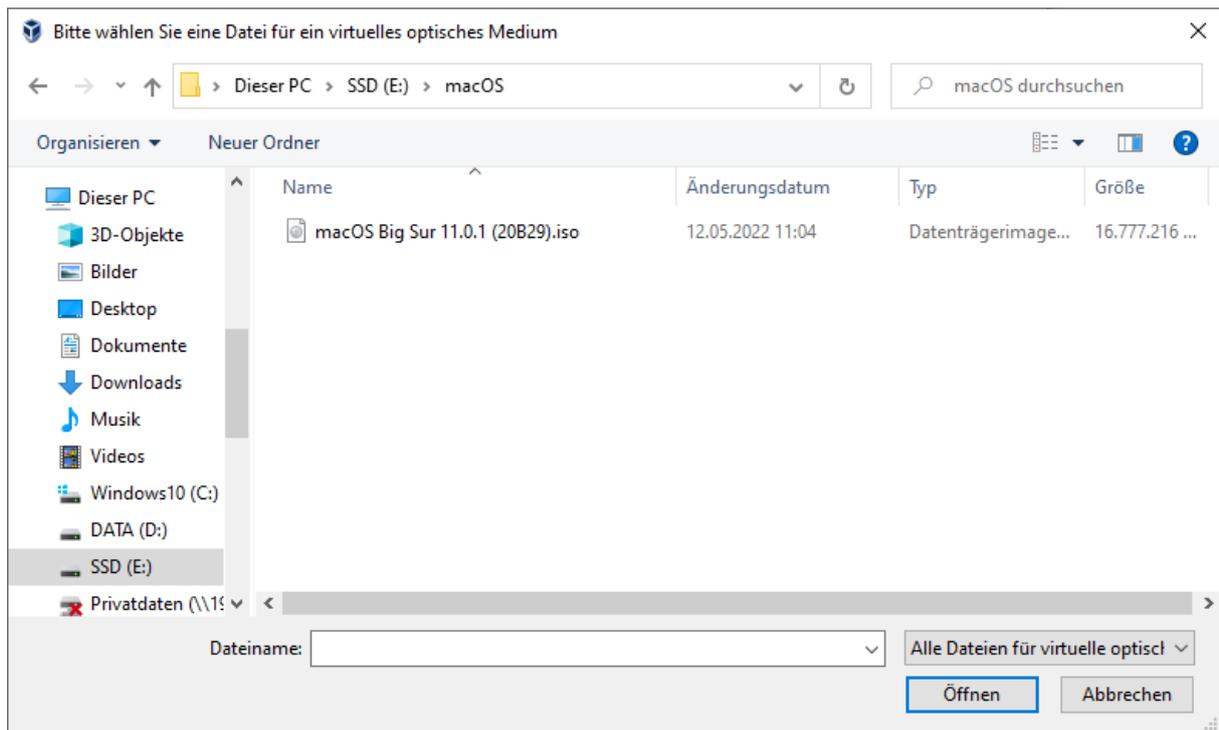


Massenspeicher **macOS Lab.vmdk** von **SATA-Port 0** auf 2 oder 3 ändern.



CD Image **macOS Big Sur....ISO** oder neuer auswählen für die CD.





Danach Öffnen einer Eingabeaufforderung (auch ohne Admin Rechte) und Ausführung folgender Befehle mit angepasstem VM Name „macOS Lab“ in einer Kommandozeile.

```
cd "C:\Program Files\Oracle\VirtualBox\"
VBoxManage.exe modifyvm "macOS Lab" --cpuidset 00000001 000106e5 00100800
0098e3fd bfebfbff
VBoxManage setextradata "macOS Lab"
"VBoxInternal/Devices/efi/0/Config/DmiSystemProduct" "iMac19,1"
VBoxManage setextradata "macOS Lab"
"VBoxInternal/Devices/efi/0/Config/DmiSystemVersion" "1.0"
VBoxManage setextradata "macOS Lab"
"VBoxInternal/Devices/efi/0/Config/DmiBoardProduct" "Mac-AA95B1DDAB278B95"
VBoxManage setextradata "macOS Lab" "VBoxInternal/Devices/smc/0/Config/DeviceKey"
"ourhardworkbythesewordsguardedpleasedontsteal(c)AppleComputerInc"
VBoxManage setextradata "macOS Lab"
"VBoxInternal/Devices/smc/0/Config/GetKeyFromRealSMC" 1
```



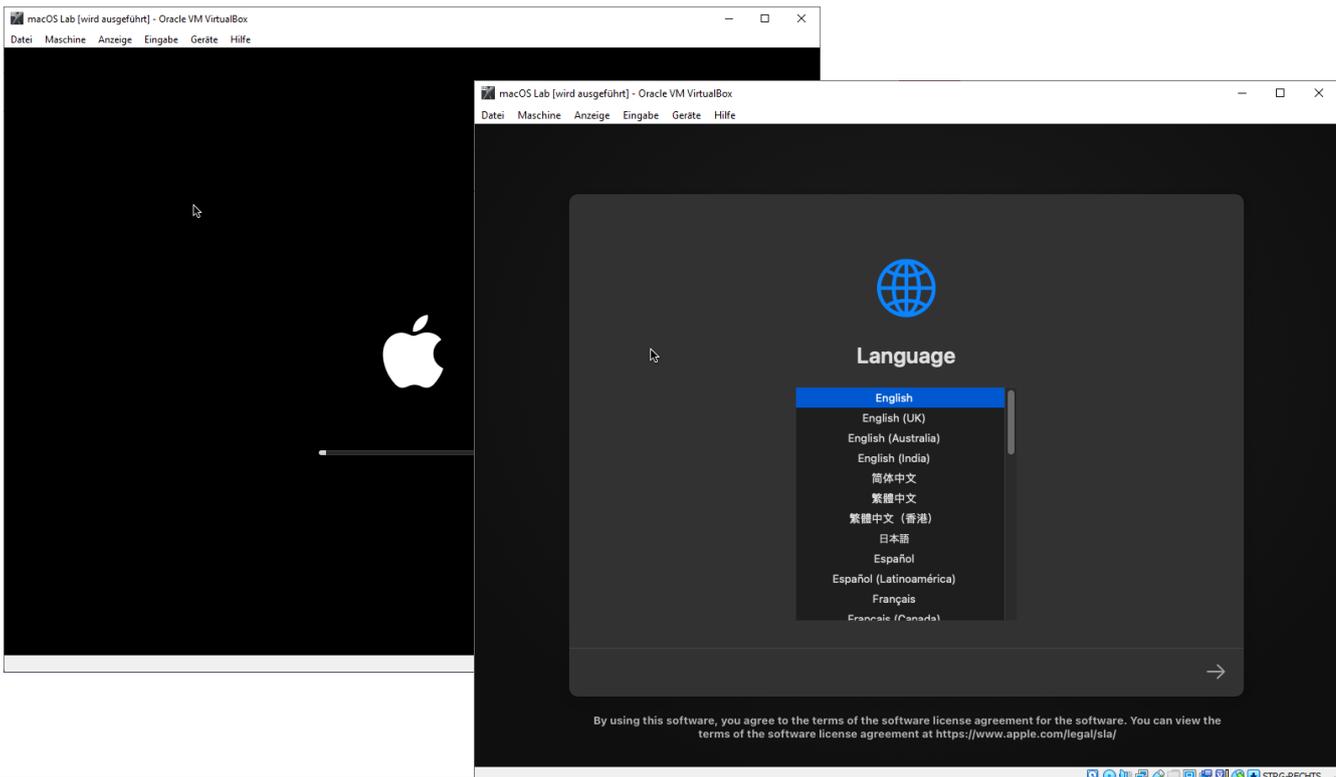
```
Eingabeaufforderung
Microsoft Windows [Version 10.0.19044.1766]
(c) Microsoft Corporation. Alle Rechte vorbehalten.

C:\Users\John Doe>cd "C:\Program Files\Oracle\VirtualBox\"
C:\Program Files\Oracle\VirtualBox>VBoxManage.exe modifyvm "macOS Lab" --cpuidset 00000001 000106e5 00100800 0098e3fd bfebfbff
C:\Program Files\Oracle\VirtualBox>VBoxManage setextradata "macOS Lab" "VBoxInternal/Devices/efi/0/Config/DmiSystemProduct" "iMac19,1"
C:\Program Files\Oracle\VirtualBox>VBoxManage setextradata "macOS Lab" "VBoxInternal/Devices/efi/0/Config/DmiSystemVersion" "1.0"
C:\Program Files\Oracle\VirtualBox>VBoxManage setextradata "macOS Lab" "VBoxInternal/Devices/efi/0/Config/DmiBoardProduct" "Mac-AA95B1DDAB278B95"
C:\Program Files\Oracle\VirtualBox>VBoxManage setextradata "macOS Lab" "VBoxInternal/Devices/smc/0/Config/DeviceKey" "ourhardworkbythesewordsguardedpleasedontsteal(c)AppleComputerInc"
C:\Program Files\Oracle\VirtualBox>VBoxManage setextradata "macOS Lab" "VBoxInternal/Devices/smc/0/Config/GetKeyFromRealSMC" 1
C:\Program Files\Oracle\VirtualBox>
```

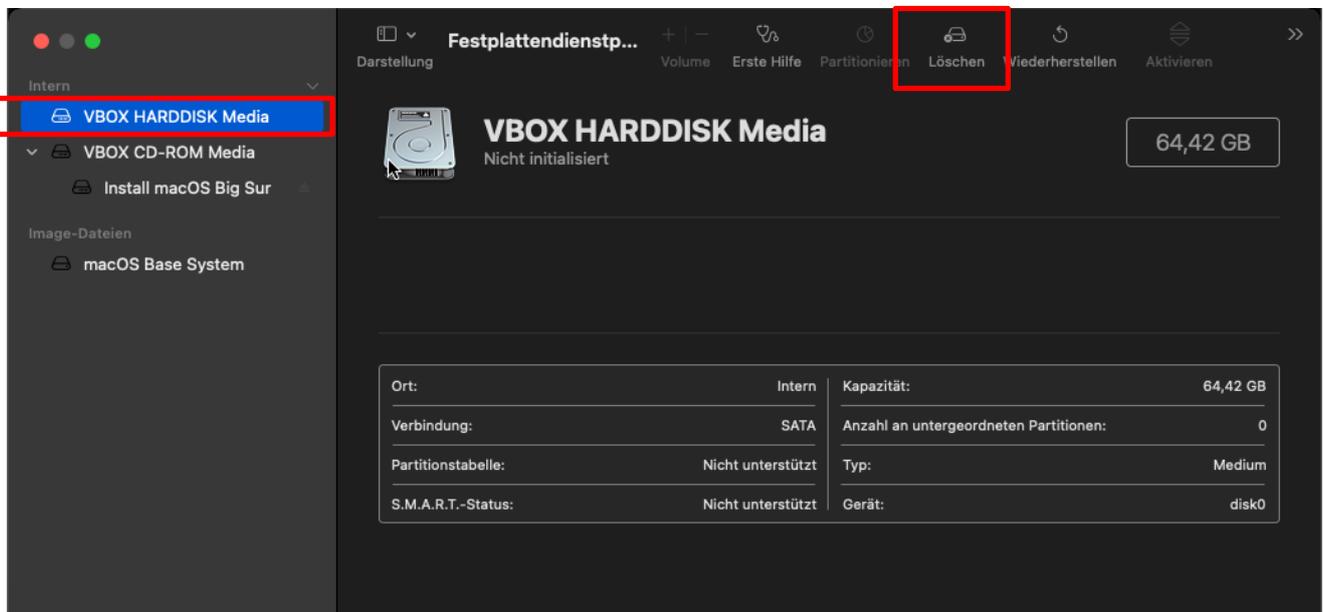
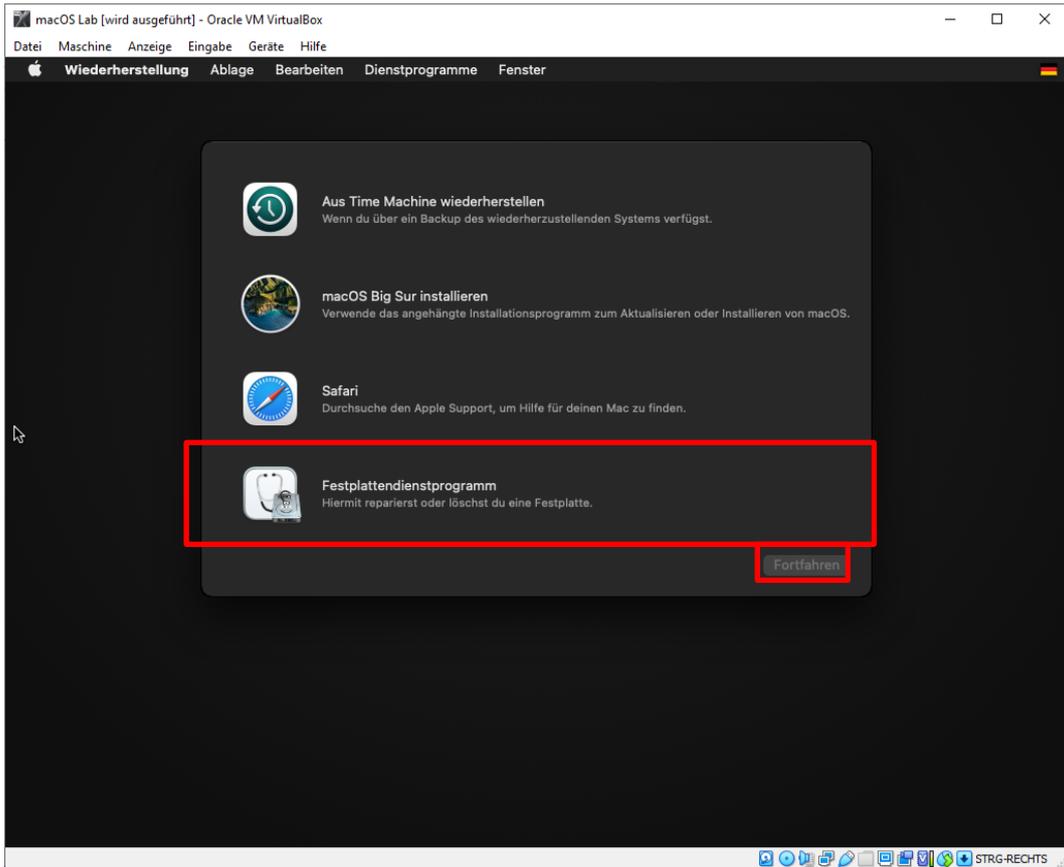
Eine vorgefertigte und eingerichtete VM finden Sie unter:

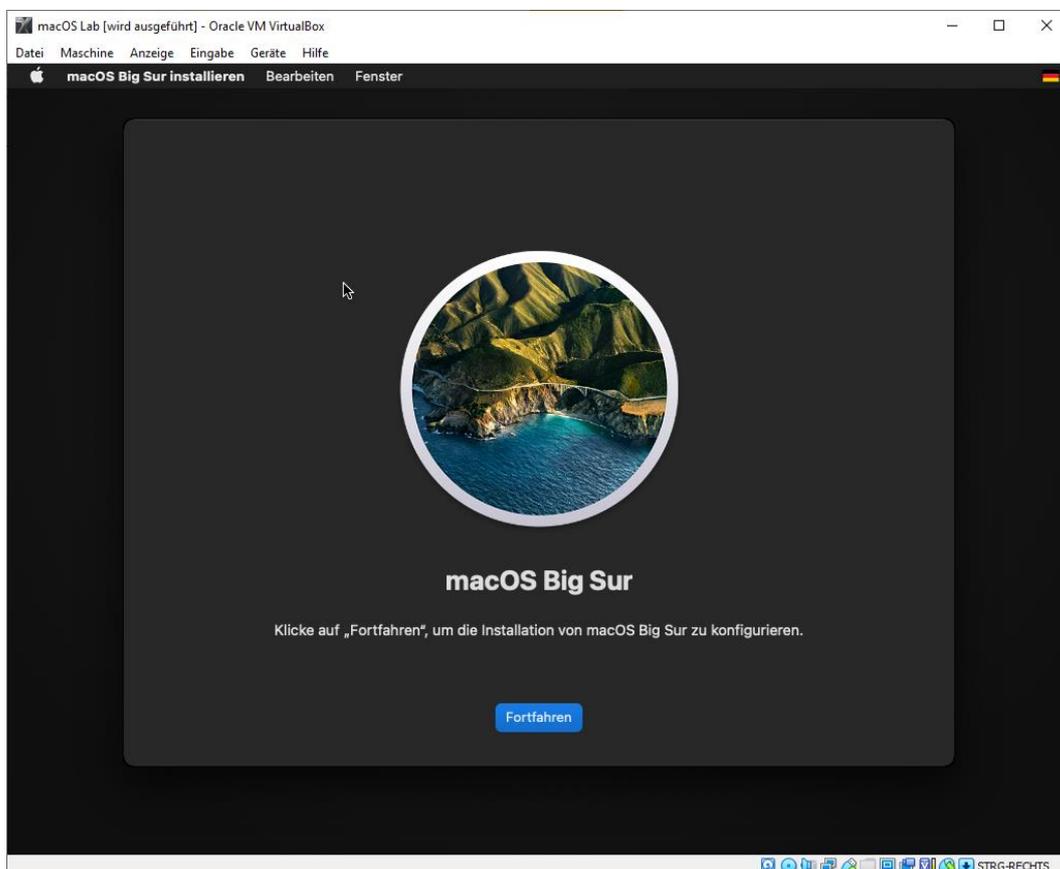
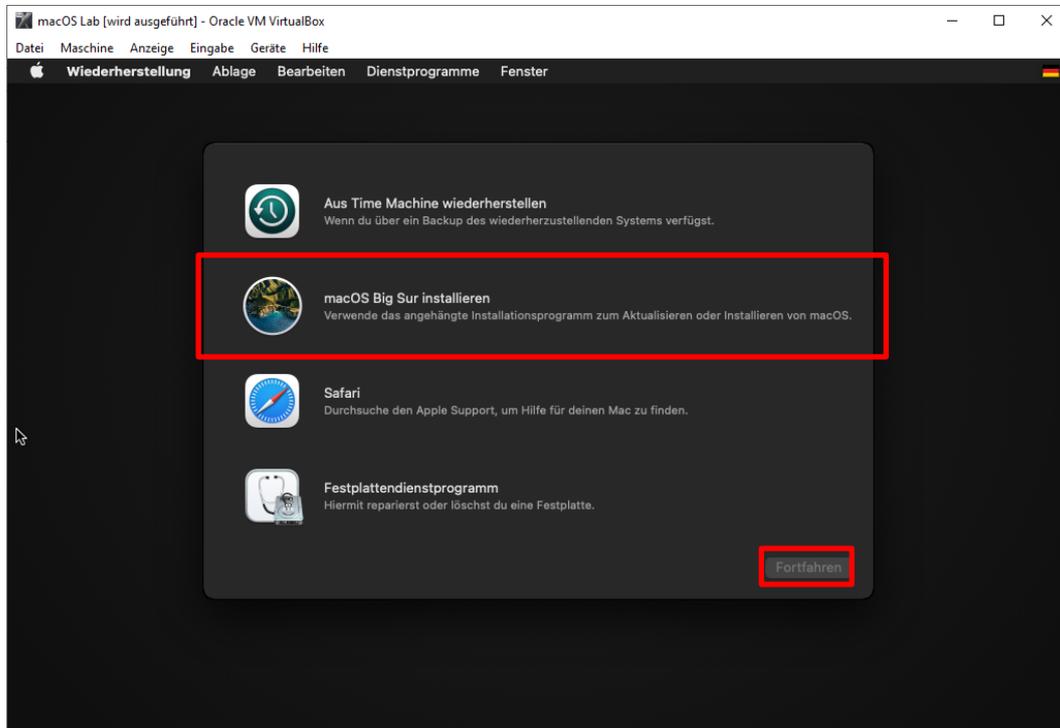
<https://download.hs-mittweida.de/intranet/Lehre/CB/Bodach/BKA%20Studiengang/Betriebssysteme/Praktikum%20Blockwochen/macOS/macOS Lab.ova>

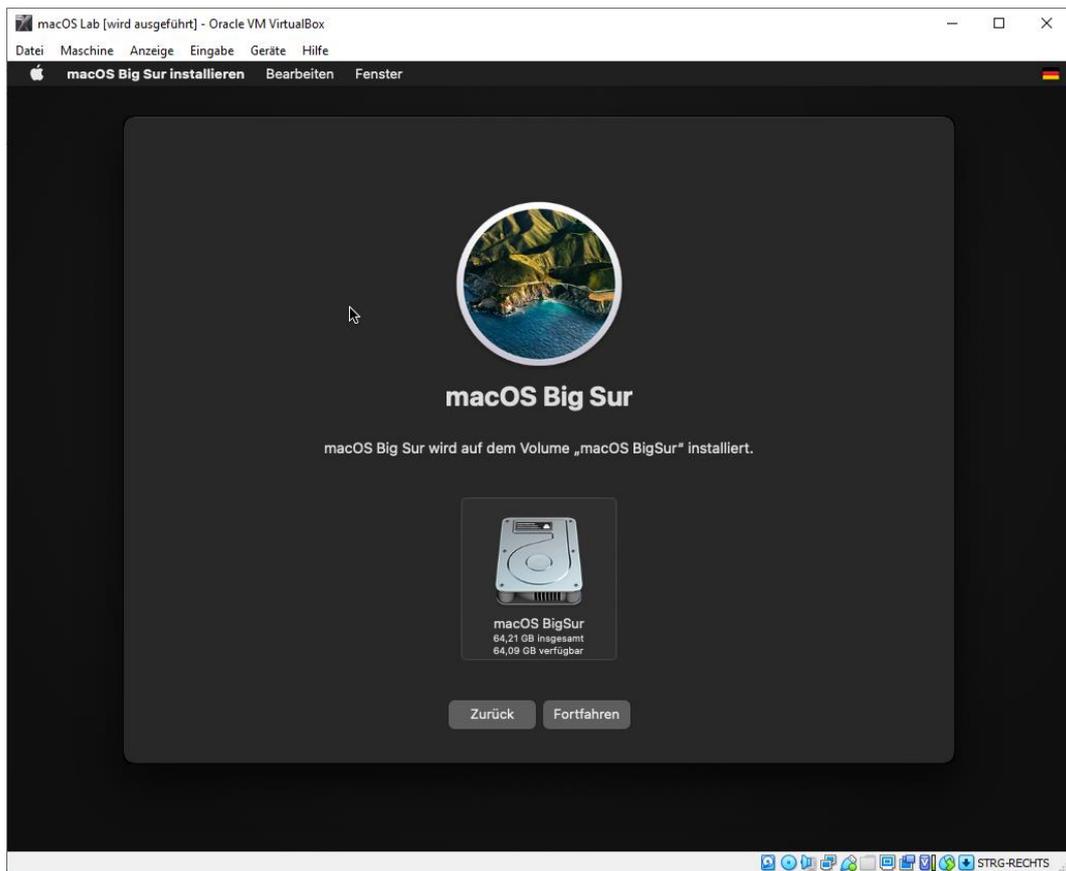
Danach Starten der VM und Boot Vorgang der macOS Installations CD abwarten. Jetzt kann macOS eingerichtet werden.



Für die Installation muss die HDD manuell formatiert (gelöscht) werden um macOS zu installieren.





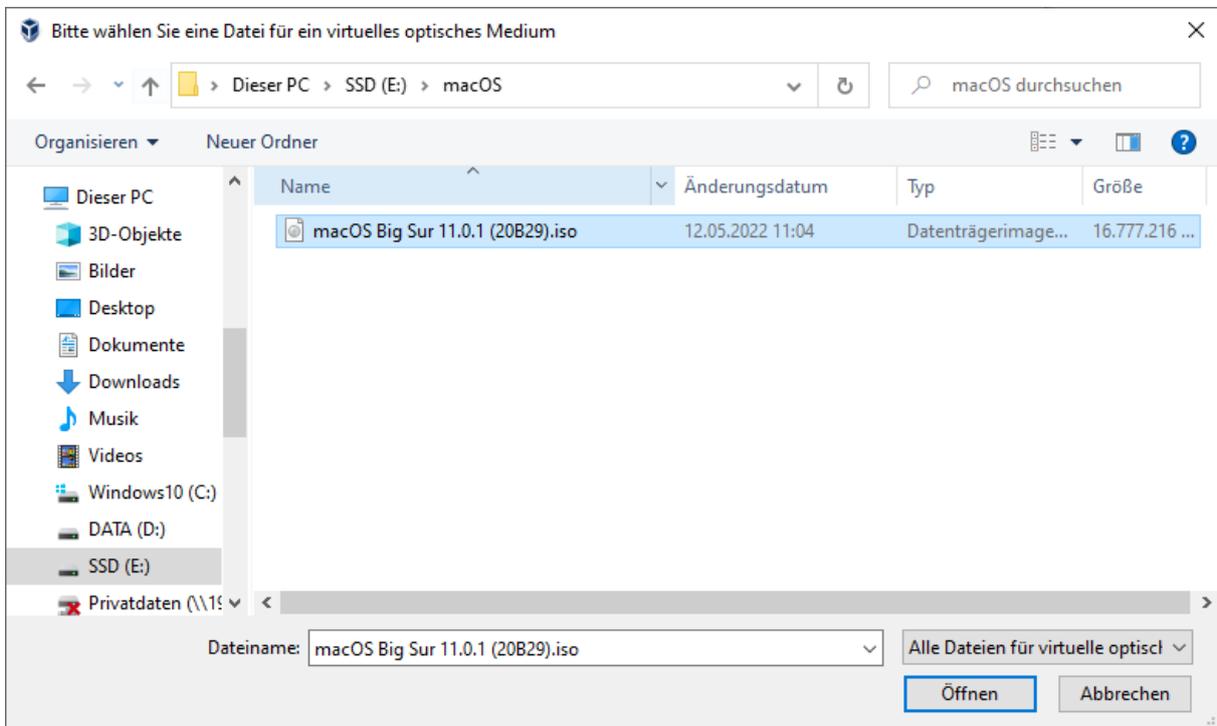
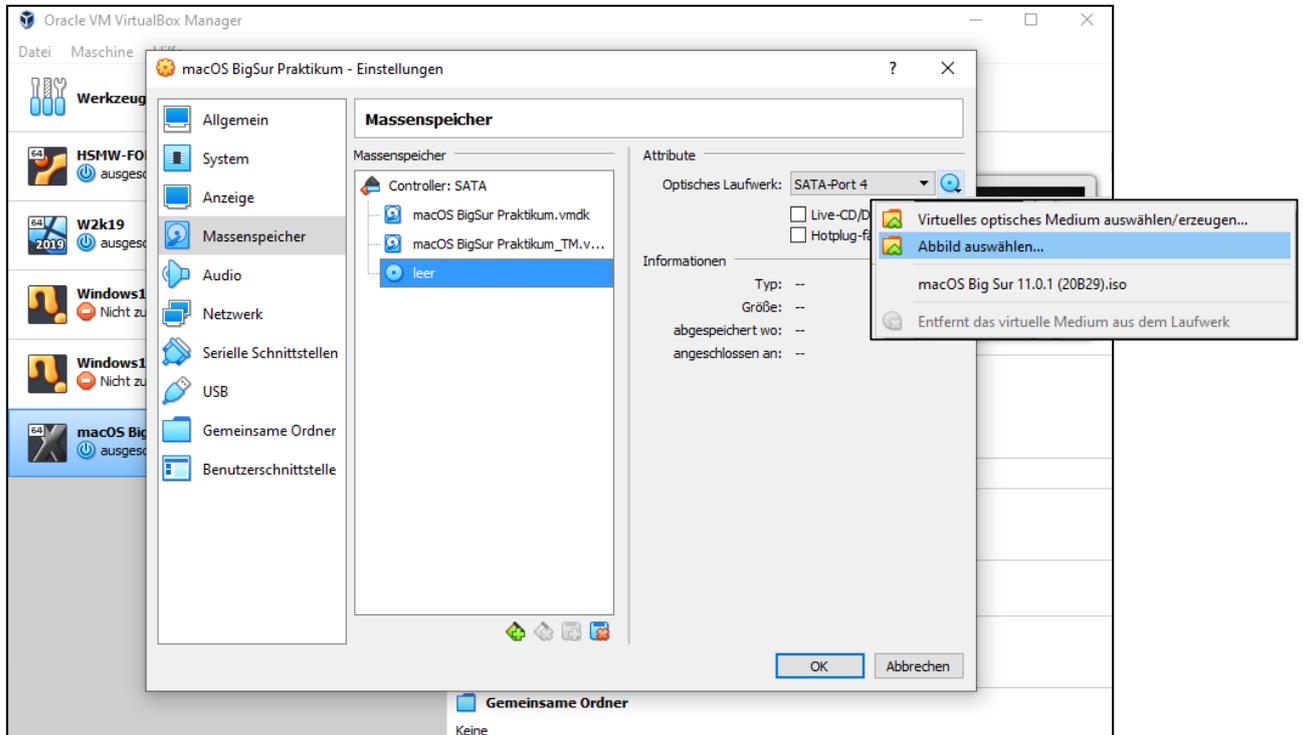


Jetzt kann die weitere Installation erfolgen, was bis zu 1h andauern kann.

5. DEAKTIVIERUNG DER SIP IN VIRTUALBOX

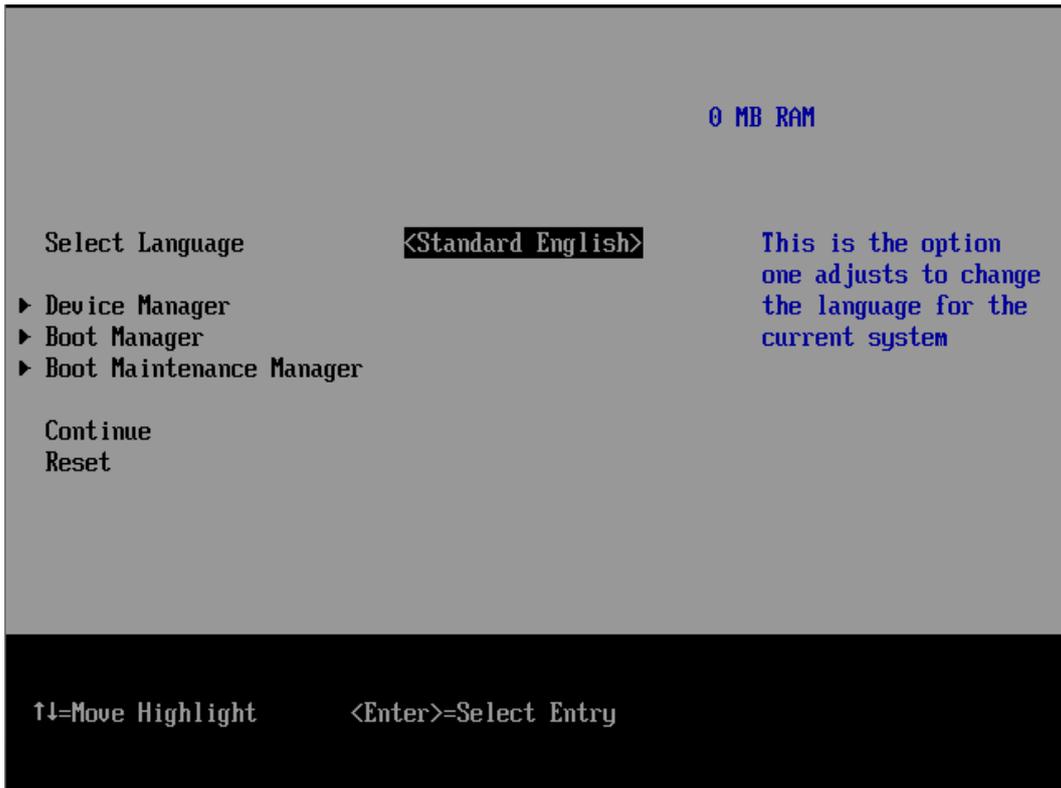
Konfiguration der VM aufrufen über Ändern.

Massenspeicher auswählen und der leeren CD ein Abbild der **macOS Big Sur 11.0.1 (20B29).iso** Image Datei hinzufügen.



Danach Starten Sie die VM mit eingelegter macOS BigSur Installations CD / ISO Datei.

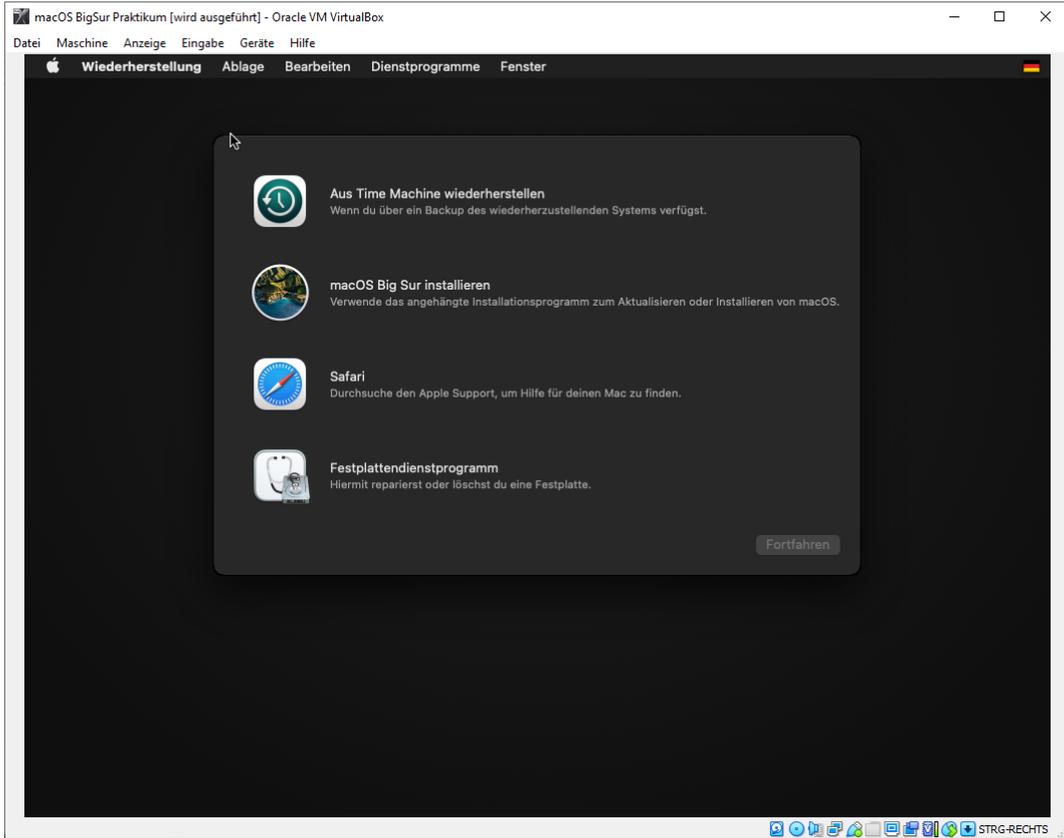
Direkt beim Starten F8/F12/ESC drücken (neueste VBox Version ESC ältere F8 oder F12!)



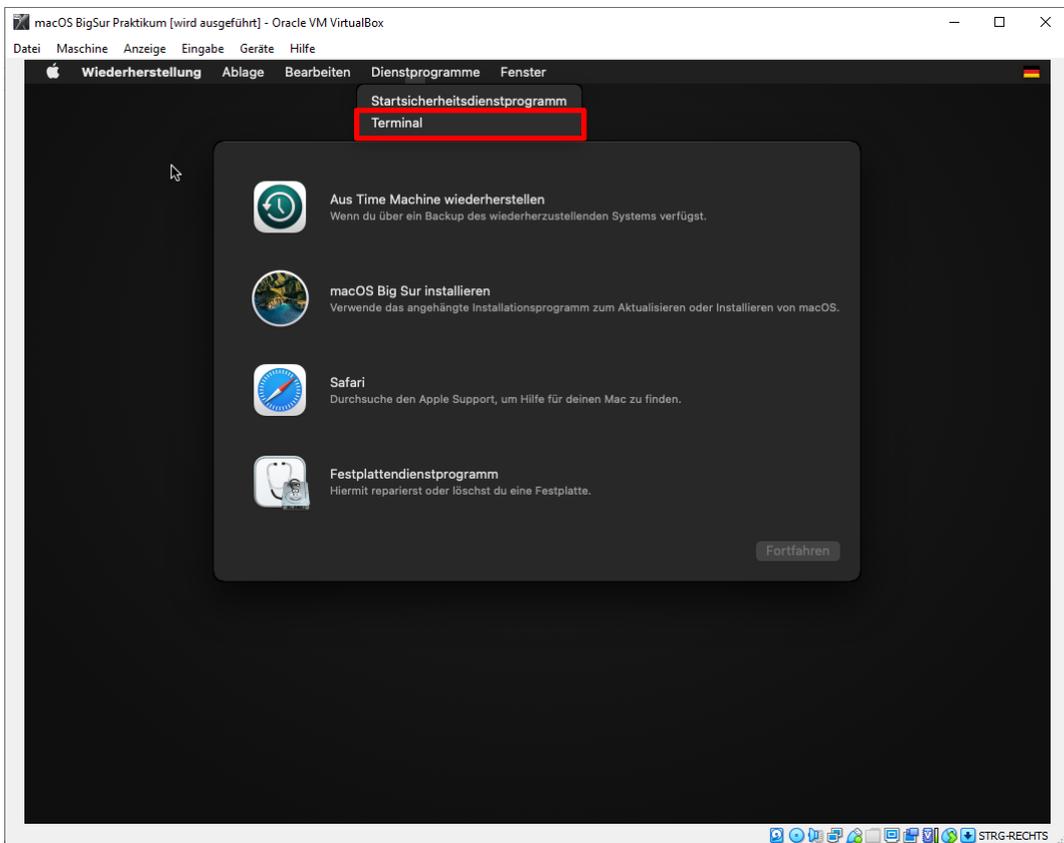
Boot Manager auswählen und CD Booten.



Wiederherstellungskonsole wird gestartet.



Terminal aus der Menüleiste aufrufen.



Im Terminal SIP überprüfen:

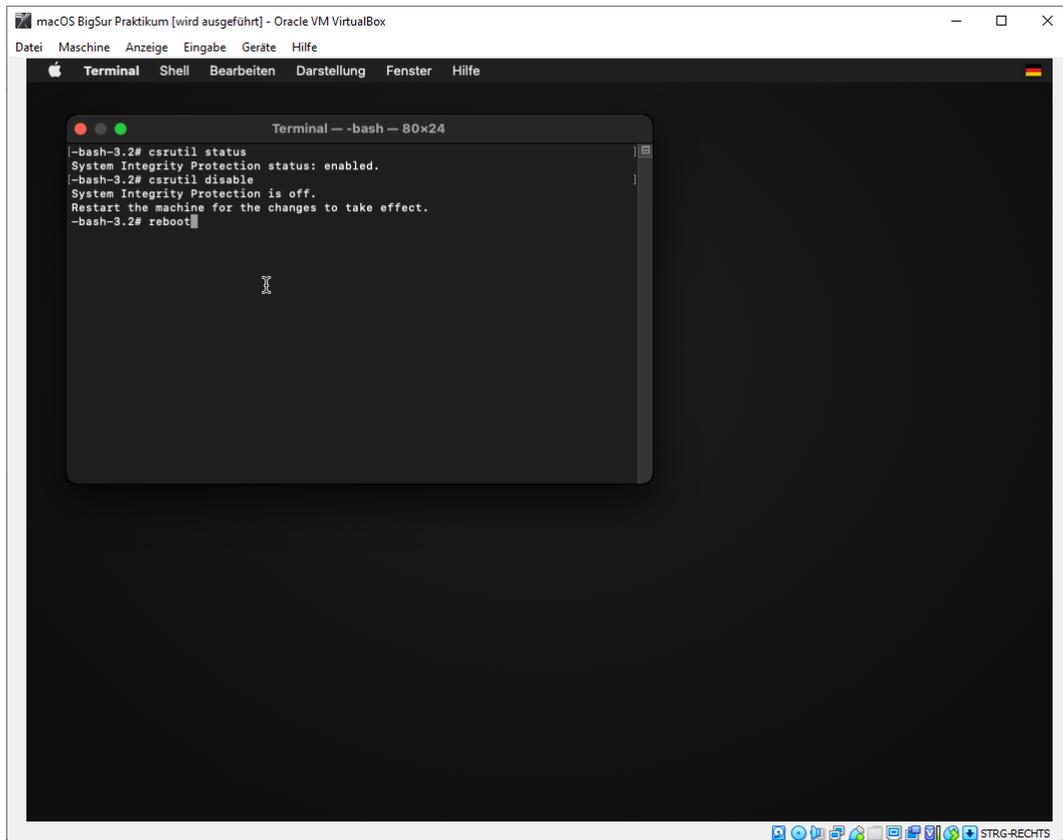
csrutil status

SIP deaktivieren mit:

csrutil disable

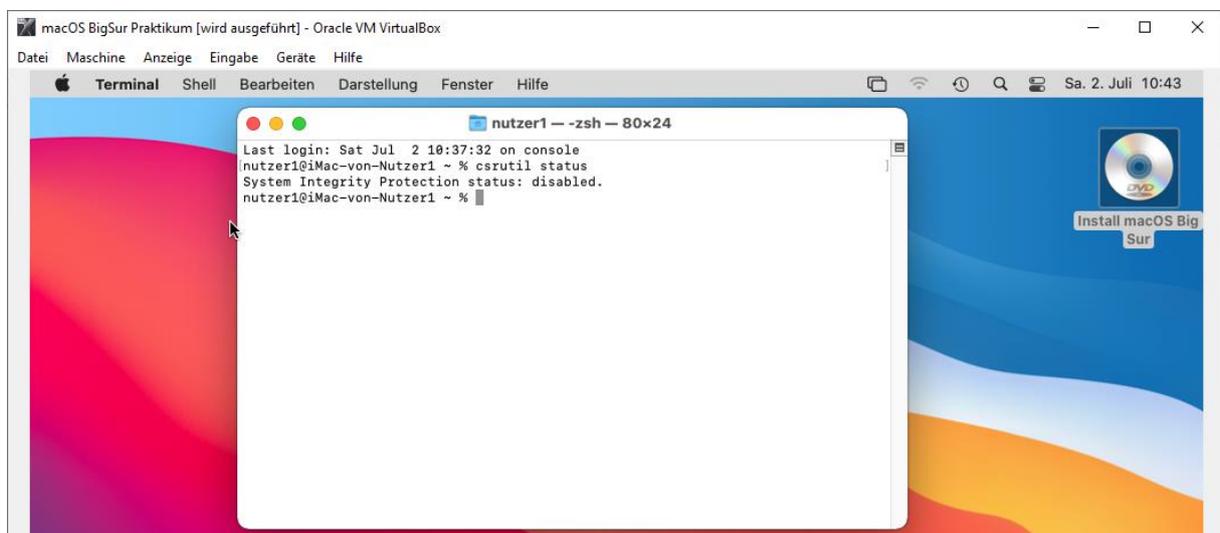
Die Deaktivierung erfolgt erst nach Neustart:

reboot



Im macOS Anmelden und das Terminal aufrufen und die Deaktivierung überprüfen mit:

csrutil status



Der System Integritätsschutz SIP ist deaktiviert.

