8.5 macOS – Prozesse, Dienste, Protokolle, Ressourcen und Speicher analysieren

Diese Seite beschreibt die systematische Fehleranalyse auf macOS. Im Mittelpunkt stehen Prozesse, `launchd`-Dienste, Protokolle, CPU, Arbeitsspeicher, APFS-Datenträger, Dateiberechtigungen, Datenschutzfreigaben und Startprobleme.

8.5.1 Ziel und Abgrenzung

Nach dieser Seite soll beurteilt werden können:

- ob eine Störung nur eine App, einen Benutzer oder das gesamte System betrifft;
- ob ein Prozess ausgeführt wird, blockiert ist, abstürzt oder ungewöhnlich viele Ressourcen verbraucht;
- ob ein Dienst im richtigen `launchd`-Bereich registriert ist;
- ob Protokolle, Absturzberichte oder Systemdiagnosen eine Ursache belegen;
- ob CPU, Arbeitsspeicher, Datenträger oder Dateisystem den Engpass bilden;
- ob Unix-Berechtigungen, ACLs oder macOS-Datenschutzfreigaben den Zugriff verhindern;
- ob Anmeldeobjekte, Hintergrunddienste oder Systemerweiterungen beteiligt sind;
- ob ein Software-, Konfigurations- oder Hardwareproblem vorliegt.

Eine laufende App, ein vorhandener Prozess oder ein registrierter Dienst beweist noch nicht, dass die ursprüngliche Benutzerfunktion funktioniert.

8.5.2 Besonderheiten der macOS-Systemarchitektur

Für die Fehleranalyse sind insbesondere folgende Komponenten wichtig:

Komponente	Aufgabe	Bedeutung für die Diagnose
Kernel	Verwaltet Hardware, Speicher, Prozesse und Systemressourcen	Kernel-Panics, Treiber- oder Hardwareprobleme können das gesamte System betreffen
`launchd`	Startet und verwaltet Dienste, Agents und XPC-Dienste	Ein Dienst kann dauerhaft oder nur bei Bedarf gestartet werden
LaunchDaemon	Systemweiter Hintergrunddienst	Wird gewöhnlich im `system`-Bereich verwaltet
LaunchAgent	Benutzerbezogener Hintergrunddienst	Wird gewöhnlich im `gui/<UID>`-Bereich verwaltet
Unified Logging	Zentrales Protokollsystem von macOS	Wird über „Konsole“ oder den Befehl `log` ausgewertet

Komponente	Aufgabe	Bedeutung für die Diagnose
APFS	Standarddateisystem moderner macOS-Systeme	Container, Volumes, Snapshots und gemeinsam genutzter Speicher müssen unterschieden werden
TCC	Datenschutz- und Zugriffssteuerung	Kann Zugriffe trotz korrekter Unix-Berechtigungen verhindern
Systemerweiterungen	Erweiterungen für Netzwerk, Sicherheit, Treiber und andere Funktionen	Fehlerhafte Erweiterungen können systemweite Auswirkungen verursachen
Aktivitätsanzeige	Grafische Prozess- und Ressourcenanalyse	Zeigt CPU, Speicher, Energie, Datenträger und Netzwerkaktivität

`launchd`-Dienste können bedarfsgesteuert arbeiten. Ein registrierter Dienst ohne aktuelle PID ist deshalb nicht automatisch fehlerhaft.

8.5.3 Typische Symptome

- Eine App startet nicht oder beendet sich sofort.
- Eine App reagiert nicht mehr.
- Ein Hintergrunddienst funktioniert nach der Anmeldung oder einem Neustart nicht.
- Das System ist dauerhaft oder zeitweise langsam.
- Der Lüfter läuft stark oder der Energieverbrauch ist ungewöhnlich hoch.
- Der Speicherdruck steigt auf Gelb oder Rot.
- Der verfügbare Datenträgerspeicher nimmt unerwartet ab.
- Ein Prozess verursacht dauerhaft hohe CPU- oder Datenträgeraktivität.
- Eine App kann bestimmte Dateien, Ordner, Geräte oder Netzwerkziele nicht verwenden.
- Eine Funktion arbeitet nur unter einem bestimmten Benutzerkonto.
- Der Fehler verschwindet im sicheren Modus.
- Der Mac startet wiederholt neu oder meldet einen Kernel-Panic.
- Ein externes Speichermedium wird nicht oder nur schreibgeschützt eingebunden.
- Eine App funktioniert nach einem macOS- oder App-Update nicht mehr.

8.5.4 Kennzeichnung der Befehle

Kennzeichnung	Bedeutung
`[LESEND]`	Liest Informationen aus und verändert normalerweise keine Konfiguration
`[LIVE]`	Beobachtet laufende Ereignisse bis zum Abbruch
`[PRIVILEGIERT]`	Benötigt möglicherweise `sudo` oder Administratorrechte
`[BELASTEND]`	Kann CPU, Datenträger oder Protokollsystem zusätzlich belasten

Kennzeichnung	Bedeutung
[VERÄNDERND]	Verändert einen Prozess, Dienst oder eine Systemeinstellung
[NEUSTART]	Unterbricht laufende Sitzungen oder Funktionen

Vor verändernden Maßnahmen müssen Befund, Konfiguration und Rückweg dokumentiert werden.

8.5.5 Sicherheits- und Betriebswarnungen

- Flüchtige Informationen müssen vor dem Beenden einer App oder vor einem Neustart gesichert werden.
- `kill -9`, erzwungenes Ausschalten und erzwungenes Aushängen eines Datenträgers können zu Datenverlust führen.
- Apple-Systemdienste dürfen nicht versuchsweise deaktiviert oder aus `/System/Library` entfernt werden.
- SIP, Gatekeeper, FileVault und Datenschutzfunktionen dürfen nicht als dauerhafte Fehlerlösung deaktiviert werden.
- Protokolle, Systemberichte und `sysdiagnose`-Archive können Benutzernamen, Pfade, Seriennummern, Netzwerkdaten und andere vertrauliche Informationen enthalten.
- `diskutil` besitzt sowohl lesende als auch destruktive Unterbefehle. Datenträgerkennung und Unterbefehl müssen vor der Ausführung kontrolliert werden.
- Eine Dateisystemreparatur ersetzt kein Backup.
- Das manuelle Löschen unbekannter Dateien aus `/Library`, `/System`, `/private` oder APFS-Systembereichen ist keine sichere Standardmaßnahme.
- Bei verwalteten Macs können MDM-Richtlinien Einstellungen absichtlich erzwingen.
- Befehlsoptionen können sich zwischen macOS-Versionen ändern. Maßgeblich ist die lokale Manpage der tatsächlich installierten Version.

Beispiel:

```
man launchctl
man log
man diskutil
```

8.5.6 Benötigte Informationen

Vor der Diagnose sollten mindestens folgende Angaben erfasst werden:

- genaue macOS-Version und Buildnummer;
- Apple-Chip oder Intel-Prozessor;
- Zeitpunkt und Häufigkeit der Störung;
- betroffene App, Funktion, Benutzer und Geräte;
- letzte Updates, Installationen oder Konfigurationsänderungen;

- angemeldeter Benutzer und vorhandene Administratorrechte;
- MDM- oder Unternehmensverwaltung;
- angeschlossene Datenträger, Docks und USB-Geräte;
- freier Speicherplatz;
- genaue Fehlermeldung;
- Verhalten nach Ab- und Anmeldung oder Neustart;
- Verhalten unter einem anderen Benutzerkonto;
- Verhalten im sicheren Modus;
- vorhandene Absturz-, Spin- oder Panic-Berichte.

8.5.7 Systemzustand und Basisdaten sichern

Die folgenden Befehle sind lesend. Systeminformationen können Seriennummern und andere vertrauliche Daten enthalten und müssen vor einer Weitergabe geprüft werden.

```
date
sw_vers
uname -a
uname -m
uptime
who
last reboot | head -n 10
system_profiler SPHardwareDataType SPSoftwareDataType
```

Wichtige Fragen:

- Stimmt die Systemzeit?
- Welche macOS-Version und welcher Build sind installiert?
- Wurde das System kurz vor der Störung neu gestartet?
- Betrifft das Problem einen Mac mit Apple-Chip oder einen Intel-Mac?
- Ist die problematische App für die vorhandene Prozessorarchitektur geeignet?
- Trat der Fehler unmittelbar nach einem Update auf?

Die Installationshistorie kann ergänzend geprüft werden:

```
system_profiler SPInstallHistoryDataType
```

Die zeitliche Übereinstimmung zwischen einer Installation und dem Beginn der Störung ist ein Hinweis, aber noch kein Ursachenbeweis.

8.5.8 Ungefährliche Schnellprüfung

```
df -h
df -ih
diskutil list
vm_stat
sysctl vm.swapusage
ps -axo pid,ppid,user,state,%cpu,%mem,etime,command | head -n 30
log show --last 15m --style compact \
  --predicate 'messageType == error OR messageType == fault'
```

Zusätzlich in der grafischen Oberfläche prüfen:

1. „Aktivitätsanzeige“ öffnen.
2. CPU, Speicher, Energie, Festplatte und Netzwerk kontrollieren.
3. Nach ungewöhnlich belastenden oder nicht reagierenden Prozessen suchen.
4. „Konsole“ öffnen und den Fehlerzeitraum untersuchen.
5. Unter „Systemeinstellungen → Allgemein → Speicher“ den verfügbaren Speicher prüfen.
6. Unter „Systemeinstellungen → Allgemein → Anmeldeobjekte & Erweiterungen“ auffällige Hintergrundobjekte erfassen.
7. Unter „Datenschutz & Sicherheit“ prüfen, ob die betroffene App die tatsächlich benötigten Freigaben besitzt.

Noch keine Prozesse beenden, Dienste neu starten oder Dateien löschen.

8.5.9 Prozesse systematisch untersuchen

Prozesse können über die Aktivitätsanzeige oder das Terminal untersucht werden.

Alle Prozesse anzeigen:

```
ps -axo pid,ppid,user,state,%cpu,%mem,etime,command
```

Mehrere Messungen der aktiven Prozesse durchführen:

```
top -l 3 -s 2 -o cpu \
  -stats pid,command,cpu,mem,threads,state,time
```

Nach einem Prozess suchen:

```
pgrep -fl "BeispielApp"
```

Einen bestimmten Prozess untersuchen, wobei `1234` durch die tatsächliche PID ersetzt wird:

```
ps -p 1234 \
  -o pid,ppid,user,state,%cpu,%mem,etime,command
```

Geöffnete Dateien und Verbindungen des Prozesses anzeigen:

```
lsof -nP -p 1234 | head -n 100
```

Einen Prozess für zehn Sekunden analysieren:

```
sample 1234 10 -file "$HOME/Desktop/BeispielApp-sample.txt"
```

Alternativ kann in der Aktivitätsanzeige ein Prozess ausgewählt und über „Mehr → Prozess analysieren“ untersucht werden.

Zu prüfen sind:

- Elternprozess;
- ausführender Benutzer;
- Laufzeit;
- CPU- und Speicherverbrauch;
- geöffnete Dateien;
- Netzwerkverbindungen;
- wiederholte Neustarts;
- nicht reagierende Threads;
- zugehörige Protokoll- und Absturzmeldungen.

Ein hoher Messwert während eines kurzen Startvorgangs kann normal sein. Aussagekräftiger ist eine wiederholte oder dauerhaft hohe Auslastung im Fehlerzeitraum.

8.5.10 Prozesse kontrolliert beenden

Zuerst sollte die App regulär beendet werden. Reagiert sie nicht, kann die Aktivitätsanzeige verwendet werden.

Ein Prozess kann im Terminal zunächst mit `SIGTERM` zum geordneten Beenden aufgefordert werden:

```
kill -TERM 1234
```

Kennzeichnung: `[VERÄNDERND]`

Risiken:

- nicht gespeicherte Daten können verloren gehen;
- verbundene Anwendungen können ihre Verbindung verlieren;
- ein von `launchd` verwalteter Prozess kann automatisch erneut gestartet werden.

`SIGKILL` beziehungsweise `kill -9` beendet einen Prozess ohne geordnetes Aufräumen und ist nur die letzte Eskalationsstufe:

```
kill -KILL 1234
```

Kennzeichnung: `[VERÄNDERND] [HOHES RISIKO]`

Vorher müssen PID, Prozessname, Benutzer und Auswirkungen eindeutig geprüft werden.

8.5.11 Dienste und Agents mit launchd analysieren

Systemweite Dienste und benutzerbezogene Agents liegen in unterschiedlichen `launchd`-Bereichen.

Bereich	Typische Verwendung
`system`	Systemweite Daemons
`user/<UID>`	Benutzerbereich ohne zwingenden Bezug zur grafischen Anmeldung
`gui/<UID>`	Grafische Anmeldesitzung eines Benutzers

Systembereich anzeigen:

```
launchctl print system
```

Bereich des aktuell angemeldeten Benutzers anzeigen:

```
launchctl print gui/$(id -u)
```

Einen konkreten systemweiten Dienst prüfen:

```
launchctl print system/com.example.service
```

Einen konkreten Benutzer-Agent prüfen:

```
launchctl print gui/$(id -u)/com.example.agent
```

Deaktivierungszustände anzeigen:

```
launchctl print-disabled system
launchctl print-disabled gui/$(id -u)
```

Typische Speicherorte:

```
~/Library/LaunchAgents
/Library/LaunchAgents
/Library/LaunchDaemons
/System/Library/LaunchAgents
/System/Library/LaunchDaemons
```

Eine Drittanbieter-Property-List syntaktisch prüfen:

```
plutil -lint "/Library/LaunchDaemons/com.example.service.plist"
```

Zu untersuchen sind:

- korrekter `launchd`-Bereich;
- exaktes Dienstlabel;
- Programmpfad und Argumente;
- ausführender Benutzer;
- letzte Exit-Codes;
- Startbedingungen;
- Standardausgabe und Fehlerausgabe;
- Dateiberechtigungen der Property-List;
- Existenz und Ausführbarkeit des hinterlegten Programms;
- Abhängigkeiten und benötigte Dateien.

Ein nicht laufender On-Demand-Dienst kann sich im Normalzustand befinden. Entscheidend ist, ob der vorgesehene Auslöser den Dienst startet und die Benutzerfunktion anschließend funktioniert.

8.5.12 Dienst kontrolliert neu starten

Ein eindeutig identifizierter Drittanbieterdienst kann zu Diagnosezwecken neu gestartet werden.

Systemweiter Dienst:

```
sudo launchctl kickstart -k system/com.example.service
```

Benutzer-Agent:

```
launchctl kickstart -k gui/$(id -u)/com.example.agent
```

Kennzeichnung: `[VERÄNDERND]`

Vorher dokumentieren:

- Dienstlabel;
- aktueller Status;
- PID;
- offene Verbindungen;
- aktuelle Protokollmeldungen;
- mögliche Benutzerunterbrechung;
- erwartetes Ergebnis.

Ein erfolgreicher Neustart beweist nur, dass der Dienst erneut gestartet werden konnte. Danach muss die ursprüngliche Funktion getestet werden.

8.5.13 Unified Logging und Konsole

Die App „Konsole“ und der Befehl `log` greifen auf das zentrale macOS-Protokollsystem zu.

Fehler und Fault-Meldungen der letzten 30 Minuten:

```
log show --last 30m --style compact \
  --predicate 'messageType == error OR messageType == fault'
```

Meldungen eines bestimmten Prozesses:

```
log show --last 30m --style compact \
  --predicate 'process == "BeispielApp"'
```

Bestimmten Zeitraum untersuchen:

```
log show \
  --start "2026-08-02 13:00:00" \
  --end "2026-08-02 13:30:00" \
```

```
  --style compact
```

Protokollmeldungen live verfolgen:

```
log stream --style compact --level info \
  --predicate 'process == "BeispielApp"'
```

Kennzeichnung: `[LIVE]`

Die Live-Ausgabe wird mit `Ctrl-C` beendet.

Ein Protokollarchiv der letzten 30 Minuten erstellen:

```
sudo log collect --last 30m \
  --output "$HOME/Desktop/macOS-30m.logarchive"
```

Kennzeichnung: `[PRIVILEGIERT]` `[DATENSCHUTZRELEVANT]`

Bei der Auswertung beachten:

- Fehlerzeitpunkt möglichst genau eingrenzen;
- Prozess, Subsystem und Kategorie berücksichtigen;
- erste Fehlermeldung einer Fehlerkette suchen;
- wiederholte Folgefehler nicht automatisch als Ursache interpretieren;
- Meldungen vor und nach dem Fehler vergleichen;
- normale Warnungen von reproduzierbaren Fehlern unterscheiden;
- fehlende Protokollmeldungen sind kein Beweis für Fehlerfreiheit;
- Info- und Debug-Meldungen werden nicht immer dauerhaft gespeichert.

8.5.14 Absturz-, Spin- und Diagnoseberichte

Die App „Konsole“ zeigt unter anderem:

- Absturzberichte mit der Erweiterung `.ips`;
- Spin-Berichte nicht reagierender Prozesse;
- Protokollberichte;
- Diagnoseberichte;
- ältere Einträge aus `system.log`;
- System- und Benutzerberichte.

Benutzerbezogene Diagnoseberichte können zusätzlich in folgendem Verzeichnis liegen:

```
ls -lt "$HOME/Library/Logs/DiagnosticReports" | head -n 20
```

Systemweite Diagnoseberichte:

```
sudo ls -lt "/Library/Logs/DiagnosticReports" | head -n 20
```

Wichtige Inhalte eines Absturzberichts:

- betroffener Prozess und Version;
- Zeitpunkt;
- Prozessorarchitektur;
- Exception Type;
- Termination Reason;
- auslösender Thread;
- verwendete Bibliotheken;
- wiederkehrende Funktionsnamen oder Module;
- Übereinstimmung mit dem tatsächlichen Fehlerzeitpunkt.

Ein einzelner Absturzbericht beweist nicht automatisch einen dauerhaften Defekt. Entscheidend sind Reproduzierbarkeit, Häufigkeit und Übereinstimmung mit der beobachteten Störung.

8.5.15 CPU-Auslastung analysieren

Grafisch:

1. Aktivitätsanzeige öffnen.
2. Bereich „CPU“ auswählen.
3. Nach `% CPU` sortieren.
4. Messung während der Störung beobachten.
5. Prozess analysieren oder Spindump erstellen.

Im Terminal:

```
top -l 5 -s 2 -o cpu \
  -stats pid,command,cpu,mem,threads,state,time
```

Mögliche Befunde:

Befund	Mögliche Bedeutung
Ein Prozess dauerhaft stark ausgelastet	Endlosschleife, fehlerhafte Verarbeitung oder außergewöhnliche Arbeitslast

Befund	Mögliche Bedeutung
Viele Prozesse gleichzeitig ausgelastet	Systemweite Last, Indizierung, Update oder konkurrierende Aufgaben
Hohe CPU mit normaler Benutzerfunktion	Möglicherweise erwartete Verarbeitung
Niedrige CPU bei langsamer App	Warten auf Datenträger, Netzwerk, Sperre oder externen Dienst
App reagiert nicht und CPU ist nahezu null	Blockierter Thread oder Warten auf eine Ressource
Wiederkehrende kurze Lastspitzen	Regelmäßiger Agent, Synchronisation oder geplanter Hintergrundprozess

Die CPU-Auslastung muss immer zusammen mit Prozesszustand, Benutzerfunktion, Protokollen und I/O-Aktivität bewertet werden.

8.5.16 Arbeitsspeicher analysieren

Apple empfiehlt, den Speicherdruck und nicht nur den freien Arbeitsspeicher zu bewerten.

Grafisch:

1. Aktivitätsanzeige öffnen.
2. Bereich „Speicher“ auswählen.
3. Speicherdruck beobachten.
4. Speicher, komprimierten Speicher und Swap-Nutzung prüfen.
5. Nach Speicherverbrauch sortieren.

Terminalprüfung:

```
vm_stat
sysctl vm.swapusage
```

Mehrere Messungen im Abstand von zwei Sekunden:

```
vm_stat -c 5 2
```

Wichtige Werte:

- freie und aktive Speicherseiten;
- komprimierter Speicher;
- Page-ins und Page-outs;
- Swap-ins und Swap-outs;
- Entwicklung der Werte während der Störung.

Interpretation:

Befund	Bewertung
Speicherdruck Grün	Speicherverwaltung arbeitet derzeit ausreichend
Speicherdruck Gelb	Ressourcen werden knapp; Verlauf und verursachende Prozesse prüfen
Speicherdruck Rot	Akuter Speicherengpass wahrscheinlich
Hohe Komprimierung ohne Störung	Kann normal sein
Wachsende Swap-Nutzung und Leistungseinbruch	Speicherengpass oder übermäßiger Prozessverbrauch möglich
Einzelner Prozess wächst kontinuierlich	Mögliches Speicherleck
Hohe historische Swap-Zahl ohne aktuelle Änderung	Kein ausreichender Ursachenbeweis

Ein einmaliger Gesamtwert ist weniger aussagekräftig als die Veränderung während des reproduzierten Fehlers.

8.5.17 Datenträger, APFS und Speicherplatz analysieren

Dateisystembelegung:

```
df -h
df -h /System/Volumes/Data
df -ih
```

Datenträgerstruktur:

```
diskutil list
diskutil info /
diskutil apfs list
```

Lokale Time-Machine-Snapshots anzeigen:

```
tmutil listlocalsnapshots /
```

Speicherbelegung im Benutzerordner untersuchen:

```
du -x -h -d 1 "$HOME" 2>/dev/null | sort -h
```

Datenträgeraktivität beobachten:

```
iostat -w 2 -c 5
```

Dateisystem oder Datenträger-I/O für 15 Sekunden verfolgen:

```
sudo fs_usage -w -f diskio -t 15
```

Kennzeichnung: `[PRIVILEGIERT] [LIVE] [BELASTEND] [DATENSCHUTZRELEVANT]`

Ein Volume prüfen:

```
diskutil verifyVolume /
```

Kennzeichnung: `[PRÜFEND] [POTENZIELL BELASTEND]`

Bei APFS beachten:

- mehrere Volumes können Speicher innerhalb eines Containers gemeinsam nutzen;
- lokale Snapshots können Speicher belegen;
- löschbarer Speicher wird nicht in jeder Anzeige identisch dargestellt;
- System- und Datenvolume bilden gemeinsam das startfähige macOS-System;
- ein voller Datenträger kann Updates, Protokollierung, Swap, App-Starts und Datenbanken beeinträchtigen;
- freie Inodes und freier Speicherplatz sind unterschiedliche Größen;
- externe Datenträger können durch Kabel, Stromversorgung, Adapter oder Dateisystemfehler ausfallen.

Für Reparaturen sollte das Festplattendienstprogramm verwendet werden. Apple empfiehlt, „Erste Hilfe“ zuerst für die Volumes, anschließend für die Container und zuletzt für das physische Speichergerät auszuführen. Bei Problemen mit dem Startvolume kann die macOS-Wiederherstellung erforderlich sein.

8.5.18 Dateiberechtigungen, ACLs und erweiterte Attribute

Identität des aktuellen Benutzers prüfen:

```
id
groups
```

Datei oder Ordner untersuchen:

```
ls -ldeO@ "/Pfad/zum/Ordner"
stat -x "/Pfad/zum/Ordner"
```

Dabei werden unter anderem sichtbar:

- Eigentümer;
- Gruppe;
- klassische Unix-Berechtigungen;
- ACL-Einträge;
- Dateiflags;
- erweiterte Attribute.

Zu prüfen ist der gesamte benötigte Pfad. Schreibrechte auf eine Datei reichen beispielsweise nicht aus, wenn ein übergeordneter Ordner nicht betreten oder verändert werden darf.

Berechtigungen dürfen nicht pauschal mit `chmod -R 777` geöffnet werden. Dadurch entstehen Sicherheitsprobleme, und bestehende ACLs oder Eigentümerfehler werden nicht zuverlässig behoben.

8.5.19 Datenschutzfreigaben und TCC

macOS schützt bestimmte Daten und Geräte zusätzlich zu den Unix-Berechtigungen.

Mögliche Freigaben sind unter anderem:

- Dateien und Ordner;
- Festplattenvollzugriff;
- Kamera;
- Mikrofon;
- Bildschirm- und Systemaudioaufnahme;
- Bedienungshilfen;
- Automation;
- Eingabeüberwachung;
- Bluetooth;
- lokales Netzwerk.

Ein Zugriff kann deshalb fehlschlagen, obwohl `ls -l` korrekte Unix-Berechtigungen anzeigt.

Prüfung:

1. „Systemeinstellungen → Datenschutz & Sicherheit“ öffnen.
2. Nur die für die App tatsächlich benötigte Kategorie auswählen.
3. Prüfen, ob die richtige App oder das richtige Terminalprogramm eingetragen ist.
4. Nach einer Änderung die betroffene App gegebenenfalls vollständig beenden und neu starten.

5. Funktion erneut testen.

Festplattenvollzugriff sollte nicht pauschal vergeben werden. Die kleinste benötigte Freigabe ist vorzuziehen.

8.5.20 App-Signatur, Gatekeeper und Quarantäne prüfen

Eine App kann wegen einer beschädigten Signatur, einer nicht vertrauenswürdigen Herkunft oder einer unvollständigen Installation nicht starten.

Signatur einer Beispiel-App prüfen:

```
codesign --verify --deep --strict --verbose=2 \
  "/Applications/Beispiel.app"
```

Gatekeeper-Bewertung prüfen:

```
spctl --assess --type execute --verbose=4 \
  "/Applications/Beispiel.app"
```

Erweiterte Attribute anzeigen:

```
xattr -l "/Applications/Beispiel.app"
```

Diese Befehle sind lesend.

Ein negatives Ergebnis sollte durch eine unveränderte, signierte Installationsdatei des Herstellers behoben werden. Das Entfernen von Quarantäneattributen, das Ad-hoc-Signieren oder das Deaktivieren von Gatekeeper ist keine allgemeine Fehlerlösung.

8.5.21 Anmeldeobjekte, Hintergrunddienste und Erweiterungen

Grafische Prüfung:

1. „Systemeinstellungen → Allgemein → Anmeldeobjekte & Erweiterungen“ öffnen.
2. „Bei der Anmeldung öffnen“ erfassen.
3. „Im Hintergrund erlauben“ erfassen.
4. installierte Erweiterungen prüfen.
5. unbekannte oder zeitlich zur Störung passende Drittanbieterkomponenten dokumentieren.

Systemerweiterungen anzeigen:

```
systemextensionsctl list
```

MDM-Registrierungsstatus prüfen:

```
profiles status -type enrollment
```

Sicherheitszustände können ergänzend gelesen werden:

```
fdesetup status
csrutil status
spctl --status
```

Diese Zustände dürfen nicht ohne begründete Anforderung verändert werden.

Wenn Anmeldeobjekte als Ursache vermutet werden:

1. vollständige Liste sichern;
2. verdächtige Drittanbieterobjekte kontrolliert deaktivieren;
3. ab- und wieder anmelden oder neu starten;
4. ursprüngliche Funktion testen;
5. Objekte einzeln wieder aktivieren;
6. den tatsächlich verursachenden Eintrag bestätigen.

8.5.22 Netzwerkabhängigkeiten kurz prüfen

Wenn eine App langsam ist oder einen Dienst nicht erreicht, muss zwischen lokalem Prozessproblem und externer Abhängigkeit unterschieden werden.

```
scutil --dns
route -n get default
netstat -rn
lsof -nP -iTCP -sTCP:LISTEN
```

Fragen:

- Ist die lokale App blockiert oder wartet sie auf DNS?
- Ist das Ziel per IP und Name erreichbar?
- Ist ein erforderlicher lokaler Listener vorhanden?
- Verhindern Proxy, VPN, Firewall oder lokaler Netzwerkzugriff die Verbindung?
- Wartet die App auf einen Server, eine API, eine Freigabe oder einen Cloud-Dienst?

Die ausführliche Netzwerkdiagnose erfolgt nach den Abläufen aus Kapitel 3.

8.5.23 Fehler auf Benutzer, System oder Hardware eingrenzen

Test	Erkenntnis
Nur eine App betroffen	App-Konfiguration, App-Daten oder Abhängigkeit wahrscheinlich
Mehrere Apps eines Benutzers betroffen	Benutzerprofil, TCC, LaunchAgent oder Anmeldeobjekt möglich
Alle Benutzer betroffen	Systemdienst, systemweite Konfiguration, Datenträger oder Hardware möglich
Fehler nur mit einem externen Gerät	Gerät, Kabel, Adapter, Stromversorgung oder Treiber untersuchen
Fehler verschwindet nach App-Neustart	Prozesszustand wahrscheinlich, Ursache aber noch nicht bewiesen
Fehler verschwindet nach Abmeldung	Benutzerbezogener Agent oder Sitzungszustand möglich
Fehler verschwindet im sicheren Modus	Drittanbietererweiterung, Anmeldeobjekt oder zusätzlich geladene Komponente möglich
Fehler bleibt im sicheren Modus bestehen	Grundsystem, Benutzerdateien, Hardware oder weiterhin aktive Abhängigkeit untersuchen
Fehler tritt auch in der Wiederherstellung auf	Hardware oder Datenträger wird wahrscheinlicher
Apple Diagnose meldet Referenzcode	Hardwarebefund dokumentieren und nach Apple-Vorgabe weiterbearbeiten

---

8.5.24 Sicherer Modus

Der sichere Modus hilft zu prüfen, ob beim normalen Start zusätzlich geladene Software beteiligt ist.

Mac mit Apple-Chip:

1. Mac vollständig ausschalten.
2. Ein-/Ausschalter gedrückt halten, bis die Startoptionen erscheinen.
3. Startvolume auswählen.
4. Umschalttaste gedrückt halten.
5. „Im gesicherten Modus fortfahren“ auswählen.

Intel-Mac:

1. Mac einschalten oder neu starten.
2. Sofort die Umschalttaste gedrückt halten.
3. Taste loslassen, wenn das Anmeldefenster erscheint.

Der sichere Modus ist ein Kreuztest. Wenn der Fehler dort nicht auftritt, ist damit noch keine einzelne Ursache bewiesen. Anschließend müssen Anmeldeobjekte, Agents, Erweiterungen und Drittanbietersoftware einzeln geprüft werden.

Ein normaler Neustart beendet den sicheren Modus.

8.5.25 Apple Diagnose und Wiederherstellung

Apple Diagnose prüft Hardwarekomponenten.

Mac mit Apple-Chip:

1. Mac ausschalten.
2. Nicht benötigte externe Geräte trennen.
3. Ein-/Ausschalter gedrückt halten, bis die Startoptionen erscheinen.
4. Befehlstaste und `D` gedrückt halten, bis der Mac neu startet.
5. Diagnoseanweisungen befolgen.
6. angezeigte Referenzcodes dokumentieren.

Intel-Mac:

1. Mac einschalten.
2. Sofort `D` gedrückt halten.
3. Falls erforderlich, beim Start `Option-D` verwenden.
4. Diagnoseanweisungen befolgen.
5. Referenzcodes dokumentieren.

Die macOS-Wiederherstellung wird benötigt, wenn beispielsweise das Startvolume nicht im laufenden System repariert werden kann.

- Apple-Chip: Ein-/Ausschalter bis zu den Startoptionen gedrückt halten und „Optionen“ auswählen.
- Intel-Mac: beim Einschalten `Command-R` gedrückt halten.

Eine erfolgreiche Apple Diagnose schließt nicht jede denkbare Hardwarestörung aus. Sporadische Fehler, Kabelprobleme oder externe Geräte müssen weiterhin durch Kreuztests untersucht werden.

8.5.26 Systematischer Diagnoseablauf

1. **Störung aufnehmen**
   Benutzer, App, Funktion, Zeitpunkt, Häufigkeit und genaue Meldung erfassen.
2. **Umfang bestimmen**
   Eine App, ein Benutzer, alle Benutzer oder der gesamte Mac?
3. **Basiszustand sichern**
   macOS-Version, Build, Architektur, Laufzeit, Installationshistorie und Speicherplatz

dokumentieren.

4. **Fehler reproduzieren**
   Zeitpunkt notieren und nur den erforderlichen Ablauf ausführen.
5. **Prozess prüfen**
   Existenz, Benutzer, Elternprozess, Zustand, CPU, Speicher und offene Dateien untersuchen.
6. **Dienstzuordnung prüfen**
   Richtiges `launchd`-Label und richtigen Bereich bestimmen.
7. **Protokolle korrelieren**
   Meldungen unmittelbar vor, während und nach dem Fehler auswerten.
8. **Ressourcen untersuchen**
   CPU, Speicherdruck, Swap, Datenträgerbelegung und I/O messen.
9. **Zugriff prüfen**
   Unix-Berechtigungen, ACLs, TCC-Freigaben und App-Signatur unterscheiden.
10. **Abhängigkeiten prüfen**
    Netzwerk, DNS, Server, API, Freigabe, Cloud-Dienst oder externes Gerät kontrollieren.
11. **Fehlerdomäne isolieren**
    Anderer Benutzer, sicherer Modus, getrennte Peripherie oder Wiederherstellung verwenden.
12. **Hypothese formulieren**
    Erwartetes Prüfergebnis und Gegenbeweis festlegen.
13. **Eine kontrollierte Maßnahme durchführen**
    Risiko, Rückweg und Messkriterium dokumentieren.
14. **Ursprüngliche Funktion verifizieren**
    Nicht nur den Prozessstatus, sondern den realen Benutzerablauf testen.
15. **Nachkontrolle durchführen**
    Protokolle und Ressourcen erneut prüfen sowie Rückfall ausschließen.

8.5.27 Befundmatrix

Befund	Mögliche Erklärung	Nächster Nachweis
App-Prozess fehlt	App wurde nicht gestartet oder beendet sich sofort	Absturzbericht und zeitlich passende Logs prüfen
Prozess läuft, App reagiert nicht	Thread blockiert oder wartet auf Ressource	Prozessanalyse oder Spindump erstellen
Dienstlabel im `system`-Bereich nicht gefunden	Falsches Label, falscher Bereich oder Dienst nicht geladen	Property-List und tatsächliches Label prüfen
Agent fehlt im `gui/<UID>`-Bereich	Falscher Benutzer oder keine grafische Sitzung	UID und Anmeldesitzung prüfen
Dienst besitzt keine PID	On-Demand-Zustand oder Startfehler	Auslöser betätigen und Logs beobachten
Hohe CPU bei einem Prozess	Schleife, hohe Last oder fehlerhafte Verarbeitung	Mehrfach messen und Prozess analysieren

Befund	Mögliche Erklärung	Nächster Nachweis
Niedrige CPU, App trotzdem langsam	Warten auf I/O, Netzwerk oder Sperre	`fs_usage`, Netzwerk und Prozessanalyse
Speicherdruck Gelb oder Rot	Speicherengpass	Speicherverbrauch und zeitliche Entwicklung prüfen
Swap wächst während der Störung	Arbeitsspeicher reicht für aktuelle Last nicht aus	verursachende Prozesse bestimmen
Startvolume fast voll	Apps, Updates, Logs und Swap beeinträchtigt	große Verzeichnisse und Snapshots prüfen
Viele lokale Snapshots	Speicher wird durch Snapshots mitbelegt	Time-Machine-Zustand und Richtlinie prüfen
Unix-Rechte korrekt, Zugriff trotzdem verweigert	TCC- oder Sandbox-Einschränkung	Datenschutzfreigabe der richtigen App prüfen
Signaturprüfung schlägt fehl	App beschädigt oder verändert	Originaldatei des Herstellers vergleichen
Nur ein Benutzer betroffen	Benutzerprofil, LaunchAgent oder TCC	anderes Benutzerkonto als Kreuztest
Fehler verschwindet im sicheren Modus	zusätzliche Software beteiligt	Drittanbieterkomponenten einzeln prüfen
Kernel-Panics treten wiederholt auf	Treiber-, Erweiterungs- oder Hardwareproblem	Panic-Berichte, Peripherietest und Apple Diagnose
Prozess und Port sind vorhanden, Funktion scheitert	Abhängigkeit oder Anwendungsebene fehlerhaft	vollständigen Benutzerpfad Ende zu Ende prüfen

---

8.5.28 Mögliche Ursachen und erforderliche Nachweise

Mögliche Ursache	Erforderlicher Nachweis
Fehlerhafte App-Konfiguration	Fehler ist mit gesicherter Standardkonfiguration reproduzierbar beziehungsweise verschwindet kontrolliert ohne diese Konfiguration
Beschädigte App-Installation	Signatur- oder Integritätsprüfung schlägt fehl und eine unveränderte Herstellerinstallation funktioniert
Fehlerhafter LaunchAgent	Fehler tritt nur im zugehörigen Benutzerkontext auf und verschwindet nach kontrollierter Deaktivierung
Fehlerhafter LaunchDaemon	Systemweiter Fehler korreliert mit Dienststatus, Logs und reproduzierbarem Neustartverhalten
CPU-Engpass	CPU ist während der Störung dauerhaft ausgelastet und der verursachende Prozess ist bestimmt
Speicherengpass	Speicherdruck und Swap wachsen während der reproduzierten Störung
Voller Datenträger	Freier Speicher ist kritisch niedrig und die Funktion arbeitet nach kontrollierter Freigabe wieder

Mögliche Ursache	Erforderlicher Nachweis
APFS- oder Dateisystemfehler	`diskutil`, Erste Hilfe oder Systemprotokolle melden konkrete Fehler
Berechtigungsfehler	Zugriff scheitert mit konkretem Berechtigungsbefund und funktioniert nach minimaler Korrektur
TCC-Verweigerung	Benötigte Datenschutzfreigabe fehlt und die Funktion arbeitet nach gezielter Freigabe
Netzwerkabhängigkeit	Lokaler Prozess arbeitet, aber DNS, Verbindung oder Zielsystem scheitert reproduzierbar
Drittanbietererweiterung	Fehler verschwindet im sicheren Modus und kehrt mit der einzeln aktivierten Erweiterung zurück
Hardwarefehler	Apple Diagnose, wiederkehrende Panic-Berichte oder unabhängige Kreuztests belegen den Fehler

---

8.5.29 Kontrollierte Maßnahmen, Risiko und Rückweg

Maßnahme	Risiko	Rückweg
App regulär beenden und neu öffnen	Nicht gespeicherte Daten	App erneut starten und gesicherte Daten öffnen
Prozess mit `SIGTERM` beenden	Funktionsunterbrechung	App oder Dienst kontrolliert neu starten
Drittanbieterdienst mit `kickstart` neu starten	Kurzzeitiger Dienstausfall	dokumentierten Ausgangszustand und Konfiguration wiederherstellen
Anmeldeobjekt vorübergehend deaktivieren	Funktion steht nach Anmeldung nicht bereit	Eintrag anhand der gesicherten Liste wieder aktivieren
TCC-Freigabe gezielt erteilen	zusätzlicher Datenzugriff	Freigabe nach Test wieder entziehen
Drittanbieter-App aktualisieren	neue Version kann Konfiguration verändern	Installationsdatei und Konfiguration der vorherigen Version sichern
App aus Originalquelle neu installieren	lokale App-Bestandteile werden ersetzt	Konfiguration und Benutzerdaten vorher sichern
Nicht benötigte Benutzerdaten verschieben	Datei steht am alten Ort nicht mehr bereit	Datei aus dem Sicherungsort zurückverschieben
Sicherer Modus	eingeschränkte Funktionen	normal neu starten
Erste Hilfe ausführen	zusätzliche Datenträgerbelastung	aktuelles Backup bereithalten; bei Fehler abbrechen und Befund sichern
Peripheriegerät trennen	zugehörige Funktion steht nicht bereit	Gerät nach Test wieder anschließen

Pro Maßnahme darf möglichst nur eine relevante Variable verändert werden.

8.5.30 Verifikation

Nach einer Maßnahme müssen mindestens folgende Prüfungen erfolgen:

- ursprünglicher Benutzerablauf funktioniert vollständig;
- App startet und reagiert;
- benötigter Dienst wird im richtigen Kontext ausgeführt;
- erwartete Datei- und Netzwerkzugriffe funktionieren;
- keine neuen Fehler oder Fault-Meldungen erscheinen;
- CPU-Auslastung normalisiert sich;
- Speicherdruck bleibt im normalen Bereich;
- Swap wächst nicht weiter ungewöhnlich;
- ausreichender Datenträgerspeicher ist vorhanden;
- temporär erteilte Rechte wurden geprüft oder zurückgenommen;
- deaktivierte Sicherheitsfunktionen wurden nicht als Dauerlösung belassen;
- Funktion bleibt nach Ab- und Anmeldung stabil;
- falls relevant, Funktion bleibt nach einem kontrollierten Neustart stabil;
- keine andere Benutzerfunktion wurde beeinträchtigt.

8.5.31 Präventionsmaßnahmen

- macOS und Anwendungen kontrolliert aktuell halten;
- vor Updates Kompatibilität geschäftskritischer Software prüfen;
- ausreichend freien Datenträgerspeicher vorhalten;
- Backups regelmäßig durchführen und Wiederherstellung testen;
- Anmeldeobjekte und Hintergrunddienste dokumentieren;
- nicht mehr benötigte Drittanbietererweiterungen entfernen;
- Datenschutzfreigaben nach dem Minimalprinzip vergeben;
- Installationsquellen und Signaturen prüfen;
- Baseline-Werte für CPU, Speicher, Datenträger und Startzeit erfassen;
- zentrale oder regelmäßige Protokollauswertung für wichtige Systeme einrichten;
- Änderungen mit Zeitpunkt, Version und Rückweg dokumentieren;
- Systemdiagnosen und Protokolle datenschutzgerecht behandeln;
- bei wiederkehrenden Kernel-Panics Hardware und Erweiterungen frühzeitig isolieren.

8.5.32 Typische Fehler bei der Diagnose

- Einen Prozess allein wegen eines hohen Einzelwerts beenden.
- Einen fehlenden PID-Wert automatisch als Dienstfehler interpretieren.
- Einen Agent im `system`-Bereich suchen, obwohl er im Benutzerkontext läuft.
- Nur nach dem Wort „error“ suchen und Zeit, Prozess oder Subsystem ignorieren.
- Alte Protokollmeldungen mit dem aktuellen Fehler verwechseln.
- Freien RAM als einziges Speicherkriterium verwenden.
- APFS-Container, Volume, Snapshot und physisches Laufwerk gleichsetzen.
- Unix-Berechtigungen und TCC-Freigaben verwechseln.
- Gatekeeper oder SIP zum Testen dauerhaft deaktivieren.
- `chmod -R 777` als allgemeine Berechtigungslösung verwenden.

- unbekannte Systemdateien oder lokale Snapshots manuell löschen.
- sofort `kill -9` verwenden.
- nach einem Neustart den vorherigen flüchtigen Zustand nicht mehr rekonstruieren können.
- nur prüfen, ob ein Prozess läuft, statt die ursprüngliche Benutzerfunktion zu testen.

8.5.33 Typische Prüfungsfragen

Warum beweist ein laufender Prozess noch keine funktionierende Anwendung?

Der Prozess kann blockiert sein, auf eine Ressource warten oder eine benötigte Abhängigkeit nicht erreichen. Entscheidend ist die vollständige Benutzerfunktion.

Was ist der Unterschied zwischen einem LaunchDaemon und einem LaunchAgent?

Ein LaunchDaemon arbeitet systemweit. Ein LaunchAgent arbeitet im Kontext eines Benutzers beziehungsweise einer Benutzeranmeldung.

Warum kann ein registrierter launchd-Dienst keine PID besitzen, ohne fehlerhaft zu sein?

Bedarfsgesteuerte Dienste werden möglicherweise erst gestartet, wenn ein definierter Auslöser eintritt.

Warum ist der Speicherdruck aussagekräftiger als nur der freie Arbeitsspeicher?

macOS nutzt freien Speicher unter anderem für Caches und Komprimierung. Der Speicherdruck zeigt besser, ob die Speicherverwaltung die aktuelle Last noch ausreichend bewältigt.

Warum sollte kill -9 nicht als erste Maßnahme verwendet werden?

Der Prozess erhält keine Möglichkeit, Dateien zu schließen, Daten zu speichern oder seinen Zustand geordnet zu bereinigen.

Warum können korrekte Unix-Berechtigungen trotzdem zu einer Zugriffsverweigerung führen?

Zusätzliche macOS-Schutzmechanismen wie TCC können den Zugriff auf Dateien, Geräte oder geschützte Daten verhindern.

Was bedeutet es, wenn ein Fehler im sicheren Modus nicht auftritt?

Zusätzlich geladene Software, Anmeldeobjekte oder Erweiterungen werden wahrscheinlicher. Eine konkrete Ursache ist dadurch noch nicht bewiesen.

Warum müssen Protokolle zeitlich mit der Störung korreliert werden?

Ein System erzeugt auch im Normalbetrieb Warnungen und Fehler. Erst die zeitliche und funktionale Übereinstimmung macht eine Meldung für den konkreten Vorfall relevant.

Warum ersetzt eine erfolgreiche Erste-Hilfe-Prüfung kein Backup?

Die Prüfung schützt nicht vor späterem Hardwareausfall, versehentlichem Löschen oder bereits beschädigten beziehungsweise fehlenden Benutzerdaten.

8.5.34 Checkliste

- [ ] Störung und ursprüngliche Benutzerfunktion dokumentiert
- [ ] betroffene App, Benutzer und Geräte bestimmt
- [ ] Zeitpunkt und Häufigkeit erfasst
- [ ] macOS-Version, Build und Architektur gesichert
- [ ] letzte Installationen und Änderungen geprüft
- [ ] Systemlaufzeit und Neustarts erfasst
- [ ] Prozessstatus und Ressourcen gemessen
- [ ] Elternprozess und ausführender Benutzer geprüft
- [ ] richtiges `launchd`-Label bestimmt
- [ ] richtiger `launchd`-Bereich geprüft
- [ ] Protokolle auf den Fehlerzeitraum begrenzt
- [ ] Absturz-, Spin- oder Panic-Berichte geprüft
- [ ] CPU-Auslastung mehrfach gemessen
- [ ] Speicherdruck und Swap geprüft
- [ ] Datenträgerspeicher und APFS-Struktur geprüft
- [ ] Snapshots und große Verzeichnisse berücksichtigt
- [ ] Unix-Berechtigungen und ACLs geprüft
- [ ] TCC-Freigaben getrennt geprüft
- [ ] App-Signatur bei Startproblemen geprüft
- [ ] Anmeldeobjekte und Systemerweiterungen erfasst
- [ ] Netzwerk- und externe Abhängigkeiten geprüft
- [ ] Kreuztest mit anderem Benutzer durchgeführt
- [ ] sicherer Modus bei Bedarf verwendet
- [ ] Hardwarediagnose bei entsprechendem Verdacht durchgeführt
- [ ] vor Änderung Risiko und Rückweg dokumentiert
- [ ] nur eine kontrollierte Änderung durchgeführt
- [ ] ursprüngliche Benutzerfunktion verifiziert

- [ ] Protokolle und Ressourcen nachkontrolliert
- [ ] temporäre Diagnoseänderungen zurückgenommen
- [ ] Ergebnis und Präventionsmaßnahme dokumentiert

8.5.35 Schnellreferenz

Aufgabe	Befehl oder Werkzeug
macOS-Version	`sw_vers`
Prozessorarchitektur	`uname -m`
Systemlaufzeit	`uptime`
Hardware- und Softwarebericht	`system_profiler SPHardwareDataType SPSoftwareDataType`
Prozesse	`ps -axo pid,ppid,user,state,%cpu,%mem,etime,command`
Live-Ressourcen	`top` oder Aktivitätsanzeige
Prozess suchen	`pgrep -fl "Prozessname"`
offene Dateien eines Prozesses	`lsof -nP -p 1234`
Prozess analysieren	`sample 1234 10 -file Bericht.txt`
Systemweite launchd-Dienste	`launchctl print system`
Benutzer-Agents	`launchctl print gui/$(id -u)`
Property-List prüfen	`plutil -lint Datei.plist`
letzte Protokolle	`log show --last 30m`
Protokolle live	`log stream`
Arbeitsspeicher	`vm_stat`
Swap	`sysctl vm.swapusage`
Speicherbelegung	`df -h`
Datenträgerstruktur	`diskutil list`
APFS-Struktur	`diskutil apfs list`
lokale Snapshots	`tmutil listlocalsnapshots /`
Datenträger-I/O	`iostat -w 2 -c 5`
Dateiaktivität	`sudo fs_usage`
Berechtigungen und ACLs	`ls -ldeO@ Pfad`
Systemerweiterungen	`systemextensionsctl list`
MDM-Status	`profiles status -type enrollment`
App-Signatur	`codesign --verify`

Aufgabe	Befehl oder Werkzeug
Gatekeeper-Bewertung	`spctl --assess`
grafische Protokolle	Konsole
CPU, Speicher und I/O	Aktivitätsanzeige
Dateisystemprüfung	Festplattendienstprogramm
Hardwareprüfung	Apple Diagnose

8.5.36 Quellen

Offizielle Apple-Dokumentation

- Aktivitätsanzeige – Benutzerhandbuch
- Mac-Prozesse in der Aktivitätsanzeige anzeigen
- CPU-Aktivität in der Aktivitätsanzeige anzeigen
- Speichernutzung in der Aktivitätsanzeige anzeigen
- Systemdiagnose in der Aktivitätsanzeige ausführen
- Konsole – Benutzerhandbuch
- Protokollmeldungen in der App „Konsole“ anzeigen
- Berichte in der App „Konsole“ anzeigen
- Speichermedium mit dem Festplattendienstprogramm reparieren
- Mac im sicheren Modus starten
- Apple Diagnose verwenden
- Zugriff auf Dateien und Ordner steuern
- Einstellungen für Datenschutz und Sicherheit
- Anmeldeobjekte und Erweiterungen verwalten
- Systeminformationen über den Mac abrufen

Befehlsreferenzen

Die maßgebliche Befehlsreferenz ist die jeweilige lokale macOS-Manpage:

```
man launchd
man launchctl
man log
man vm_stat
man diskutil
```

```
man fs_usage
man system_profiler
```

Der folgende Spiegel stellt aus Xcode extrahierte Manpages bereit, ist jedoch keine offizielle Apple-Supportseite. Bei Abweichungen gilt die lokale Manpage der installierten macOS-Version:

- launchctl(1)
- launchd(8)
- log(1)
- vm_stat(1)
- diskutil(8)
- fs_usage(1)
- system_profiler(8)
