

Server Logs konfigurieren

Der Log Level und die Formatierung von BFFH können über folgende Wege konfiguriert werden.

Umgebungsvariable

Über die Umgebungsvariable `BFFH_LOG=debug` - zum Beispiel eingebunden per systemd Service

Unterstützte Level für `BFFH_LOG` sind:

- info
- warn
- error
- debug
- trace

Parameter in bffhd

Über `bffhd` Parameter kann das Kommando beliebig angepasst werden (und ebenso im systemd Service verwendet werden):

`--log-format`

- Full (Standard)
- Compact
- Pretty

Full

```
2024-12-07T11:10:48.198579Z DEBUG bffh:tls: difluoroborane::tls: TLS secret logging is disabled. keylog=false
2024-12-07T11:10:48.198685Z DEBUG bffh:tls: difluoroborane::tls: reading certificates
path=/etc/ssl/fablabchemnitz.de.cert.pem
2024-12-07T11:10:48.198959Z DEBUG bffh:tls: difluoroborane::tls: reading private key
path=/etc/ssl/fablabchemnitz.de.privkey.pem
2024-12-07T11:10:48.199608Z DEBUG difluoroborane::actors::process: Process actor updating state
```

```
name=actor-process-test cmd=/opt/fabinfra/actor-process-test.sh state=SendState(ArchivedState { inner:
ArchivedMachineState { state: Free, previous: Some(ArchivedUserRef { id: "local_lab_admin" }) } })
2024-12-07T11:10:48.200451Z DEBUG difluoroborane::actors::process: Process actor updating state
name=Tasmota_Mjolnir cmd=/opt/fabinfra/adapters/tasmota/main.py state=SendState(ArchivedState { inner:
ArchivedMachineState { state: InUse(ArchivedUserRef { id: "local_lab_admin" }), previous:
Some(ArchivedUserRef { id: "local_lab_admin" }) } })
2024-12-07T11:10:48.203817Z INFO bffh:binding API listen sockets: difluoroborane::capnp: Opened listen
socket on 127.0.0.1:5961
2024-12-07T11:10:48.204079Z INFO bffh:binding API listen sockets: difluoroborane::capnp: Opened listen
socket on 192.168.1.192:5961
```

Compact

Das kompakte Layout ist identisch zu Full

```
2024-12-07T11:10:48.198579Z DEBUG bffh:tls: difluoroborane::tls: TLS secret logging is disabled. keylog=false
2024-12-07T11:10:48.198685Z DEBUG bffh:tls: difluoroborane::tls: reading certificates
path=/etc/ssl/fablabchemnitz.de.cert.pem
2024-12-07T11:10:48.198959Z DEBUG bffh:tls: difluoroborane::tls: reading private key
path=/etc/ssl/fablabchemnitz.de.privkey.pem
2024-12-07T11:10:48.199608Z DEBUG difluoroborane::actors::process: Process actor updating state
name=actor-process-test cmd=/opt/fabinfra/actor-process-test.sh state=SendState(ArchivedState { inner:
ArchivedMachineState { state: Free, previous: Some(ArchivedUserRef { id: "local_lab_admin" }) } })
2024-12-07T11:10:48.200451Z DEBUG difluoroborane::actors::process: Process actor updating state
name=Tasmota_Mjolnir cmd=/opt/fabinfra/adapters/tasmota/main.py state=SendState(ArchivedState { inner:
ArchivedMachineState { state: InUse(ArchivedUserRef { id: "local_lab_admin" }), previous:
Some(ArchivedUserRef { id: "local_lab_admin" }) } })
2024-12-07T11:10:48.203817Z INFO bffh:binding API listen sockets: difluoroborane::capnp: Opened listen
socket on 127.0.0.1:5961
2024-12-07T11:10:48.204079Z INFO bffh:binding API listen sockets: difluoroborane::capnp: Opened listen
socket on 192.168.1.192:5961
```

Pretty

```
2024-12-07T11:09:18.093419Z DEBUG difluoroborane::tls: reading certificates, path:
/etc/ssl/fablabchemnitz.de.cert.pem
  at bffhd/tls.rs:113
  in difluoroborane::tls::tls
  in bffh::bffh
```

```
2024-12-07T11:09:18.093835Z DEBUG difluoroborane::tls: reading private key, path:
/etc/ssl/fablabchemnitz.de.privkey.pem
```

```
at bffhd/tls.rs:123
in difluoroborane::tls::tls
in bffh::bffh
```

```
2024-12-07T11:09:18.097839Z INFO difluoroborane::capnp: Opened listen socket on 127.0.0.1:5961
```

```
at bffhd/capnp/mod.rs:99
in difluoroborane::capnp::binding API listen sockets
in bffh::bffh
```

```
2024-12-07T11:09:18.098140Z INFO difluoroborane::capnp: Opened listen socket on 192.168.1.192:5961
```

```
at bffhd/capnp/mod.rs:99
in difluoroborane::capnp::binding API listen sockets
in bffh::bffh
```

--log-level

Die Level sind die gleichen wie `BFFH_LOG`

Siehe auch [Cheat Sheet - Wichtigste Befehle \(Übersicht\)](#)

--verbose (-v)

Dieser Parameter kann bis zu drei mal als Argument angegeben werden und erhöht die Log-Ausgabe zusätzlich. Beispiel:

```
opt/fabinfra/bffh/target/release/bffhd --config /opt/fabinfra/bffh-data/config/bffh.dhall --verbose --verbose --
verbose
```

oder

```
opt/fabinfra/bffh/target/release/bffhd --config /opt/fabinfra/bffh-data/config/bffh.dhall -vvv
```

--tls-key-log

Dieser Parameter wird nur für Entwickler benötigt. Wenn für Debug Zwecke der Inhalt der verschlüsselten Verbindungen eingesehen werden soll, werden in der angegebenen Datei

`<PATH>` die Schlüssel für jede Verbindung gespeichert und können z.B. von [Wireshark](#)

geladen werden.

Log File schreiben

Wer `bffhd` nicht über `systemd` startet und deshalb auch keine Logs mit `journalctl` auslesen möchte, der kann den Output auch konventiell in eine Log-Datei schreiben - hier im Beispiel `bffh.log`. Wir fangen dabei die Output-Streams `stdout` und `stderr` gemeinsam in einem Ausgabestrom ab (`2>&1`). Den Parameter `--log-level` verwenden wir nicht, da er bei der normalen Systemausgabe ignoriert wird (siehe [Issue #83](#)) und führen stattdessen mit `BFFH_LOG` an:

```
BFFH_LOG=debug /opt/fabinfra/bffh/target/release/bffhd --config /opt/fabinfra/bffh-data/config/bffh.dhall --log-format Pretty > bffh.log 2>&1
```

Audit Log

Der Audit Log ist das Log File, was `bffhd` schreibt und in `bffh.dhall` konfiguriert wird. Es gibt Aufschluss über "Wer hat wann welche Ressource genutzt oder zurückgegeben?". Details finden sich in [Audit Log \(Revisionsprotokoll\)](#)

Version #18

Erstellt: 23 Oktober 2024 22:53:07 von Mario Voigt (Stadtfabrikanten e.V.)

Zuletzt aktualisiert: 14 Dezember 2024 22:03:38 von Mario Voigt (Stadtfabrikanten e.V.)