

Valkey Audit Module

This module provides comprehensive auditing capabilities for Valkey servers. It allows logging of various event types to different output protocols with configurable formatting.

Features

- **Multiple logging protocols:** File system and Syslog support
- **Configurable formats:** Text, JSON, and CSV output formats
- **Selective event auditing:** Configure which events to audit
- **Command payload options:** Control whether and how much of command payloads to include

Build Instructions

ValkeyAudit uses CMake for building the Valkey module.

```
mkdir build
cmake -S . -B build
cmake --build build --target all
```

Installation

Loading the Module

Add the following line to your Valkey configuration file or issue a MODULE LOAD command. As part of the loadmodule, any of the configuration parameters can be set. Examples:

```
loadmodule /path/to/libvalkeyaudit.so
loadmodule /path/to/libvalkeyaudit.so protocol file /var/log/audit/valkey_audit.log
```

Configuring the Module

The module uses the standard Valkey configuration facility which means that parameters can be accessed with CONFIG SET and GET as well as written to the valkey.conf file with CONFIG REWRITE or manually.

Available parameters:

- `audit.enable [yes|no]` : Enable or disable auditing.
- `audit.always_audit_config [yes|no]` : Enable or disable the auditing of config commands regardless of per user events setting. This allows the logging of config commands for a user even if the user is in the exclusion list.
- `audit.protocol [file|syslog]` : Logging protocol to use. When using the file protocol it should be followed by the filepath. When using the syslog protocol it should be followed by the syslog facility.

- `audit.format [text|json|csv]` : Log format.
- `audit.events [event1,event2,...]` : Event categories to audit (connections,auth,config,keys).
- `audit.payload_disable` : Disable logging command payloads.
- `audit.payload_maxsize [size]` : Maximum payload size to log in bytes.
- `audit.excluderules` : Specific usernames and/or IP addresses to exclude from auditing.

Example Usage

enable/disable

To enable/disable auditing:

```
CONFIG SET AUDIT.ENABLE yes
CONFIG SET AUDIT.ENABLE no
```

To enable/disable always auditing of config commands:

```
CONFIG SET AUDIT.ALWAYS_AUDIT_CONFIG yes
CONFIG SET AUDIT.ALWAYS_AUDIT_CONFIG no
```

protocol

To set the audit logging protocol, use one of:

```
CONFIG SET AUDIT.PROTOCOL file /path/to/logfile
CONFIG SET AUDIT.PROTOCOL syslog local0
```

format

To sets the audit log format:

```
CONFIG SET AUDIT.FORMAT text
CONFIG SET AUDIT.FORMAT json
CONFIG SET AUDIT.FORMAT csv
```

events

To set the which event categories to audit, use one of the below:

```
CONFIG SET AUDIT.EVENTS all           # Enable all events
CONFIG SET AUDIT.EVENTS none          # Disable all events
CONFIG SET AUDIT.EVENTS connections,auth # Enable only connection and auth events
```

Available event categories:

- `connections` : Client connections and disconnections
- `auth` : Authentication attempts (with password redaction)
- `config` : Configuration commands
- `keys` : Key operations

payload

Configure options for payload logging:

```
CONFIG SET AUDIT.PAYLOAD_DISABLE yes|no
CONFIG SET AUDIT.PAYLOAD_MAXSIZE 1024
```

retrieve the current configuration

To retrieve the current complete audit configuration:

```
CONFIG GET AUDIT.*
```

To retrieve the current specific audit configuration parameter:

```
CONFIG GET AUDIT.FORMAT
```

excluding users and/or IP addresses

Rules to set usernames and/or IP addresses to be excluded from auditing through a comma-separated list. The rule formats are :

```
username           # for username-only exclusion
@ipaddress         # for IPaddress-only exclusion
username@ipaddress # combination exclusion
```

Example

```
CONFIG SET AUDIT.EXCLUDERULES "un1,@192.168.1.12,un2@192.168.1.22"
```

To remove the current list of exclusion rules

```
CONFIG SET AUDIT.EXCLUDERULES ""
```

Manual Module Testing

The project has a collection of scripts to start a Valkey server using docker-compose to easily test the module.

To start a Valkey CLI shell to test the module commands, run:

```
./scripts/run_test_cli.sh
```

The above command will start the Valkey server, and opens the valkey CLI shell. When the shell closes, it also stops the Valkey server.

If you just want to start the Valkey server, run:

```
./scripts/start_valkey.sh
```

You can connect to the Valkey server from the localhost address.

To stop the servers, run:

```
./scripts/stop_valkey.sh
```

Unit Tests

The unit tests are written in python and can be found in the test/unit directory. They will start a local valkey server with the module loaded.

Requirements:

- valkey installation, environment variable VALKEY_SERVER if not in the path
- environment variable AUDIT_MODULE_PATH to point at the module shared library libvalkeyaudit.so

To do: automation

Logged Events

Connection Events

Example in text format:

```
[2025-04-15 14:30:22] [CONNECTION] CONNECTED client_id=0x7f8a1c003a40  
[2025-04-15 14:35:15] [CONNECTION] DISCONNECTED client_id=0x7f8a1c003a40
```

Authentication Events

Example in text format (password is always redacted):

```
[2025-04-15 14:30:25] [AUTH] ATTEMPT password=<REDACTED>
```

Configuration Commands

Example in text format:

```
[2025-04-15 14:32:10] [CONFIG] GET param=port  
[2025-04-15 14:33:05] [CONFIG] SET param=maxclients
```

Key Operations

Example in text format:

```
[2025-04-15 14:31:18] [KEY_OP] SET key=user:1001 payload={"name":"John","email":"john@example.c
```

License

This module is licensed under the same terms as Valkey itself.