August 28

KEYSPACE

Amsterdam

Deploying Valkey at Enterprise level with LDAP authentication and auditing

Ricardo Dias

Principal Software Engineer @ Percona and Valkey Maintainer

Martin Visser

Valkey Tech Lead

Valkey

Intro Ricardo and Martin

Ricardo J. Dias

@rjd15372

Ricardo is a principal software engineer at Percona where he works as contributor to the Valkey project. Ricardo has been working in distributed storage systems for many years, but his interests are not limited to distributed systems, he also enjoys designing and implementing lock-free data structures, as well as, developing static code analyzers. In his free time, he's a family guy and also manages a roller hockey club.

Martin Visser

@martinrvisser

Martin is Percona's tech lead for Valkey. A long term database geek from analytics, OLTP to in-memory, he is also an open source enthusiast at heart. In his spare time, Martin is a family man and enjoys learning, tinkering and building.

Enterprise Valkey Deployments

Requirements

1. Availability
2. Monitoring and Management
3. Security

Enterprise Security & Accountability

- Ensure data integrity and confidentiality
- Mitigate risks and prevent unauthorized access
- Maintain compliance with regulations and standards
- Use of centralized authentication systems
- Foster trust and transparency with stakeholders

Authentication

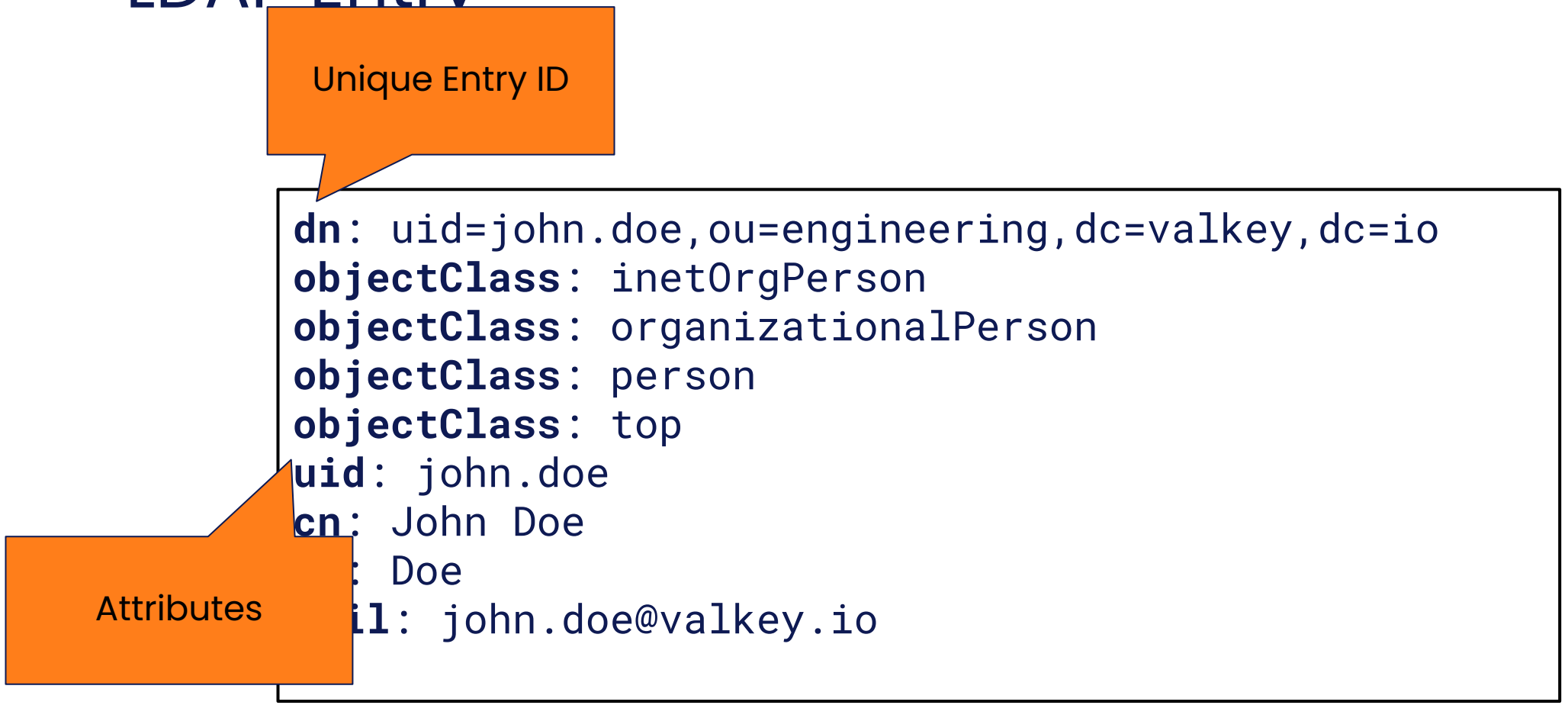
Valkey Authentication

- Valkey stores each user data internally
  - password
  - permissions (aka ACLs)
- User/Password authentication
  - AUTH command
- No integration with external authentication systems in the core
  - Valkey modules can be used for this purpose

Why LDAP support?

- LDAP is extensively utilized as an authentication system in enterprise environments
- LDAP is an industry-standard protocol, meaning it's widely supported by various applications and operating systems, promoting interoperability.
- LDAP offers significant flexibility in schema and structure, allowing organizations to tailor the directory to their specific needs

LDAP Entry



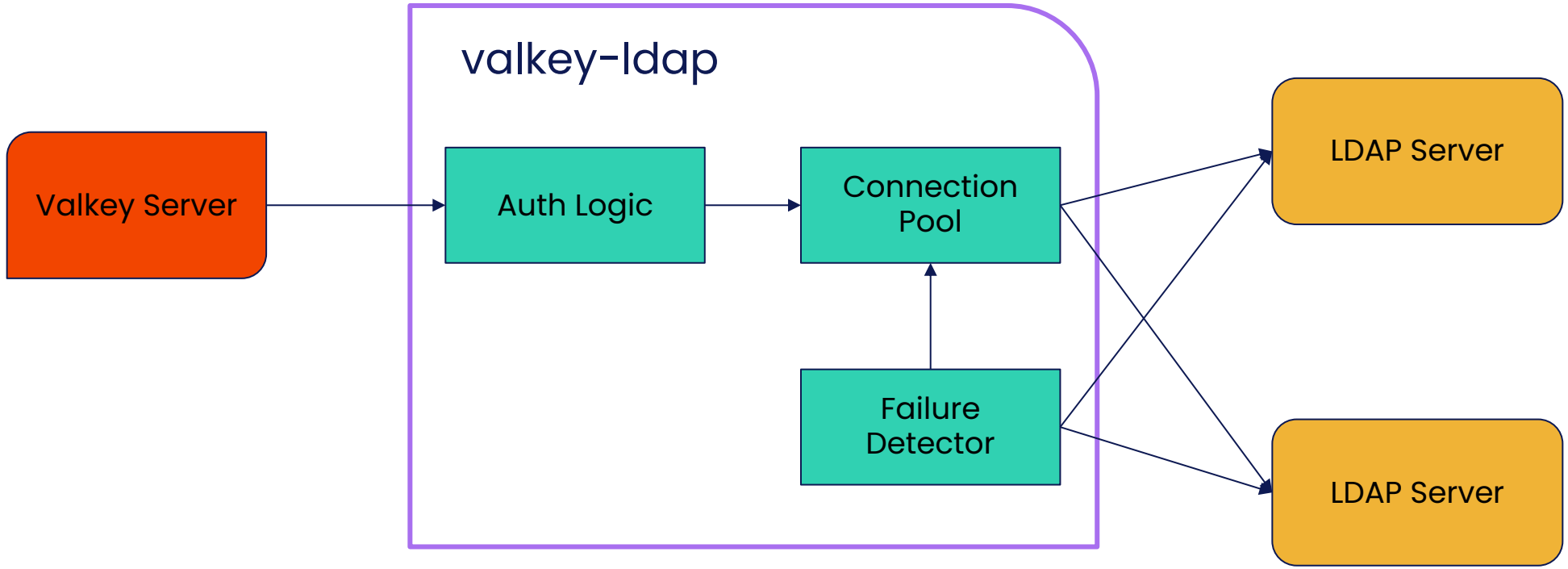
LDAP Authentication

- Authentication is done with the pair DN and password
- The authentication operation is called BIND
  - If DN/password is correct BIND succeeds; fails otherwise

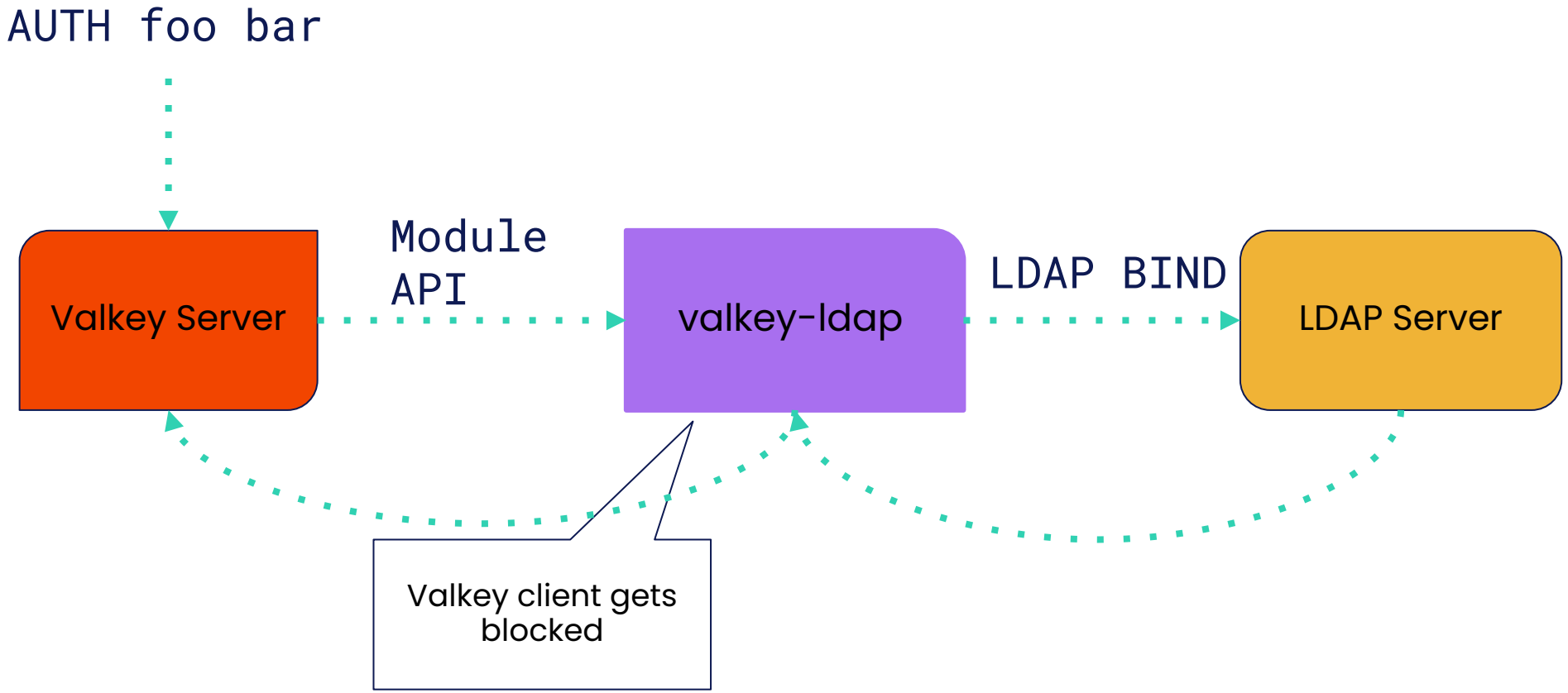
Valkey LDAP module

- Allows Valkey users to authenticate against an external identity provider using the LDAP protocol, such as Active Directory
- Provides flexible configuration that support a wide range of LDAP directory structures
- Supports HA setups of LDAP servers
- Secure connections between Valkey <-> LDAP using TLS
- Developed with Rust Valkey SDK

Valkey LDAP Module Architecture



Valkey LDAP Module Authentication Flow



Steps to setup Valkey LDAP module

1. Setup user entries in Valkey that match LDAP user accounts
2. Configure the LDAP servers location
3. Configure the LDAP server connection properties, like TLS
4. Configure the LDAP authentication mode
5. Configure the options specific to the mode chosen above

Valkey LDAP Authentication Modes

- Supports two LDAP authentication modes
  - **bind** mode
  - **search+bind** mode

Valkey LDAP: **bind** mode

- Used when the username provided in the Valkey AUTH command matches a substring of the distinguished name (DN) of user entries in the LDAP directory
- The distinguished name (DN) is constructed by adding a configurable prefix and suffix to the username.
- The prefix is set to `CN=` or `DOMAIN\`
- The suffix defines the remainder of the DN

Valkey LDAP: **bind** mode configuration

```
dn: uid=ben,ou=engineering,dc=valkey,dc=io
objectclass: person
objectclass: inetOrgPerson
cn: Ben Alex
sn: Alex
uid: ben
```

Valkey User:
ben

```
CONFIG SET ldap.bind_dn_prefix "uid="

CONFIG SET ldap.bind_dn_suffix ",ou=engineering,dc=valkey,dc=io"
```

Valkey LDAP: **search+bind** mode

- Offers greater flexibility, allowing the username to match any attribute value of an LDAP user entry, making it suitable for more complex directory structures.
- Requires 3 RTT to the LDAP server per authentication request
  a. Bind with admin user DN
  b. Search for the user entry
  c. Bind with the user DN

Valkey LDAP: **search+bind** mode configuration

```
dn: cn=Ben Alex,ou=engineering,dc=valkey,dc=io
objectClass: person
objectClass: inetOrgPerson
entryDN: cn=Ben Alex,ou=engineering,dc=valkey,dc=io
cn: Ben Alex
sn: Alex
uid: ben
```

Valkey User:
ben

```
CONFIG SET ldap.search_attribute "uid"

CONFIG SET ldap.search_dn_attribute "entryDN"

CONFIG SET ldap.search_filter "objectClass=person"
```

Valkey LDAP Observability

```
# INFO ldap
ldap_server_0:host=<hostname>,status=unhealthy,error=<some error message>
ldap_server_1:host=<hostname>,status=healthy,ping_time_ms=1.645
```

Valkey LDAP Configuration

Config Name	Type	Default	Description
ldap.search_bind_dn	string	""	DN of the user account used to perform LDAP searches.
ldap.search_bind_passwd	string	""	Password for the search bind user.
ldap.search_base	string	""	Base DN where the search for the user entry begins.
ldap.search_filter	string	"objectClass=*"	LDAP search filter to apply when searching for user entries.
ldap.search_attribute	string	"uid"	LDAP attribute to match against the username provided in the AUTH command.
ldap.search_scope	Enum(base, one, sub)	sub	Scope of the LDAP search.
ldap.search_dn_attribute	string	"entryDN"	Attribute containing the DN of the user entry.

Config Name	Type	Default	Description
ldap.bind_dn_prefix	string	"cn="	String to prepend to the username from the AUTH command when forming the DN for LDAP bind.
ldap.bind_dn_suffix	string	""	String to append to the username from the AUTH command when forming the DN for LDAP bind.

Config Name	Type	Default	Description
ldap.auth_mode	Enum(bind, search+bind)	bind	The authentication method.
ldap.servers	string	""	Space-separated list of LDAP URLs in the form ldap[s]://<domain>:<port>.

Config Name	Type	Default	Description
ldap.use_starttls	boolean	no	Whether to upgrade to a TLS-encrypted connection using the STARTTLS operation (RFC 4513).
ldap.tls_ca_cert_path	string	""	Filesystem path to the CA certificate for validating the LDAP server certificate.
ldap.tls_cert_path	string	""	Filesystem path to the client certificate for TLS connections to the LDAP server.
ldap.tls_key_path	string	""	Filesystem path to the client certificate key for TLS connections to the LDAP server.

Bind Mode Options

Config Name	Type	Default	Description
ldap.connection_pool_size	number	2	Number of connections in each LDAP server's connection pool.
ldap.failure_detector_interval	number	1	Interval (in seconds) between each failure detector check.
ldap.timeout_connection	number	10	Number of seconds to wait when connecting to an LDAP server before timing out.
ldap.timeout_ldap_operation	number	10	Number of seconds to wait for an LDAP operation before timing out.

Valkey LDAP Next Steps

- Improve LDAP server connection healing
- Support LDAP user groups
- Remove the limitation of requiring users to exist in Valkey

How to get Valkey LDAP

- **Source**: https://github.com/valkey-io/valkey-ldap
- **RPMs**: https://copr.fedorainfracloud.org/coprs/rjd15372/valkey-ldap/
- **Container**: https://hub.docker.com/r/valkey/valkey-bundle/
- **Documentation**: https://valkey.io/topics/ldap/

Audit

Auditing

- verify and prove compliance with data protection policies
- requires collection of
  - security events
  - user activity
  - system changes
- collected centrally

Valkey core

- valkey log file
  - errors and warnings
  - startup and shutdown
  - replication and persistence events
- `ACL LOG` for unauthorized access requests
- `MONITOR` command

ValkeyAudit

https://github.com/martinrvisser/valkey-audit

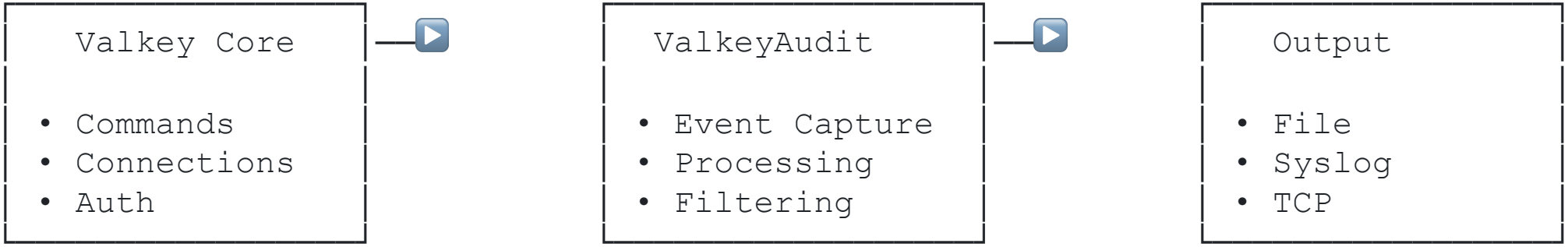
- **ValkeyAudit**(BSD-3-Clause) is a security auditing module for Valkey
- provides comprehensive logging of valkey activities for compliance and security monitoring
- tracks user actions, connections, authentication attempts, and command execution

- written in C (rust also an option for modules)

ValkeyAudit - requirements

- configurability
  - log output format and destination
  - events capture and filtering
- performance
  - lowest possible impact on normal operation
- security
  - removal of sensitive payloads
  - removal of authentication information
  - encryption of audit information

ValkeyAudit - architecture



- makes use of the standard Valkey module API
- uses an in-memory buffer for command auditing
- uses a separate thread for writing to destination

ValkeyAudit - events

- **Event Classification** categorizes events (AUTH, KEYS, CONFIG, etc.)
- **Source Tracking** which user and IP address
- **Exclusion Engine** applies user/IP exclusion rules
- **Payload Processing** handles command arguments and data

ValkeyAudit - exclusion

- Event category filter
  - KEY_OP, CONFIG, AUTH etc.
- Source filter
  - exclude specific users e.g. high-throughput application user
  - exclude specific IP addresses e.g. application server IP address
  - combination of both

ValkeyAudit - output

- **Buffered I/O**: asynchronous writing with circular buffer
- **Multiple Formats**: Text, JSON, CSV
- **Multiple Destinations**: File, Syslog, TCP endpoints
- **Retry Logic**: TCP reconnection and error handling

ValkeyAudit - auth handling

- AUTH command is intercepted pre-execution
- check for AUTH success/failure is currently timer based
- change in Valkey server :
  - https://github.com/valkey-io/valkey/pull/2237

Current timer workaround allows for AUTH to succeed or fail before checking the result in ACL LOG :

```
CONFIG SET audit.auth_result_check_delay_ms 10
```

ValkeyAudit - config basics

Enable : `audit.enabled ( default yes )`

Format : `audit.format ( default TEXT)`

```
audit.format TEXT
audit.format CSV
audit.format JSON
```

Protocol e.g.:

```
audit.protocol "syslog local0"
audit.protocol "tcp logserver.company.com:514"
audit.protocol "file /var/log/audit/valkey_audit_xyz.log"
```

ValkeyAudit - config events

Events that apply for all “not-excluded” users

```
    audit.events all                    # everything (default)
    audit.events none                   # nothing
    audit.events keys                   # std Valkey key commands
included
    audit.events config             # config commands included
    audit.events auth                   # AUTH commands included
    audit.events connections        # connection events included
    audit.events other              # anything not part of above categories
```

ValkeyAudit - config payload

Payload control for key operations

```
audit.payload_disable yes/no
    # whether to log the payload of a key command
audit.payload_maxsize 1024
    # size in bytes of the max amount to log for the
      payload for a single key command
```

ValkeyAudit - configure exclusion

Purpose of audit logs

- log abnormal operation
- high volume standard app operations have no audit value

Exclusion rules `audit.excluderules <comma separated list>`

- `username` `# do not audit this user`
- `@IPaddress` `# do not audit anything coming from this IP`
- `username@IPaddress` `# do not audit this user, IP combination`

Specific exception which overrides everything to log config changes :

```
audit.always_audit_config yes/no
```

ValkeyAudit - specific TCP settings

TCP connection settings

```
audit.tcp_host "logserver.company.com"
audit.tcp_port 514
```

or

```
audit.protocol "logserver.company.com:514"
```

Connection timeout

```
audit.tcp_timeout_ms 5000
```

ValkeyAudit - TCP retry config

Reconnect on connection failure

```
audit.tcp_reconnect_on_failure yes
```

Maximum retries on connection failure

```
audit.tcp_max_retries 3
```

Time between retries on connection failure

```
audit.tcp_retry_interval_ms 1000
```

To keep the audit entries in the buffer when TCP is disconnected. When full and disconnected, entries will be discarded.

```
audit.tcp_buffer_on_disconnect yes
```

ValkeyAudit - TCP retry config

When either condition is met, buffer is flushed to destination :

- Buffer flush based on size

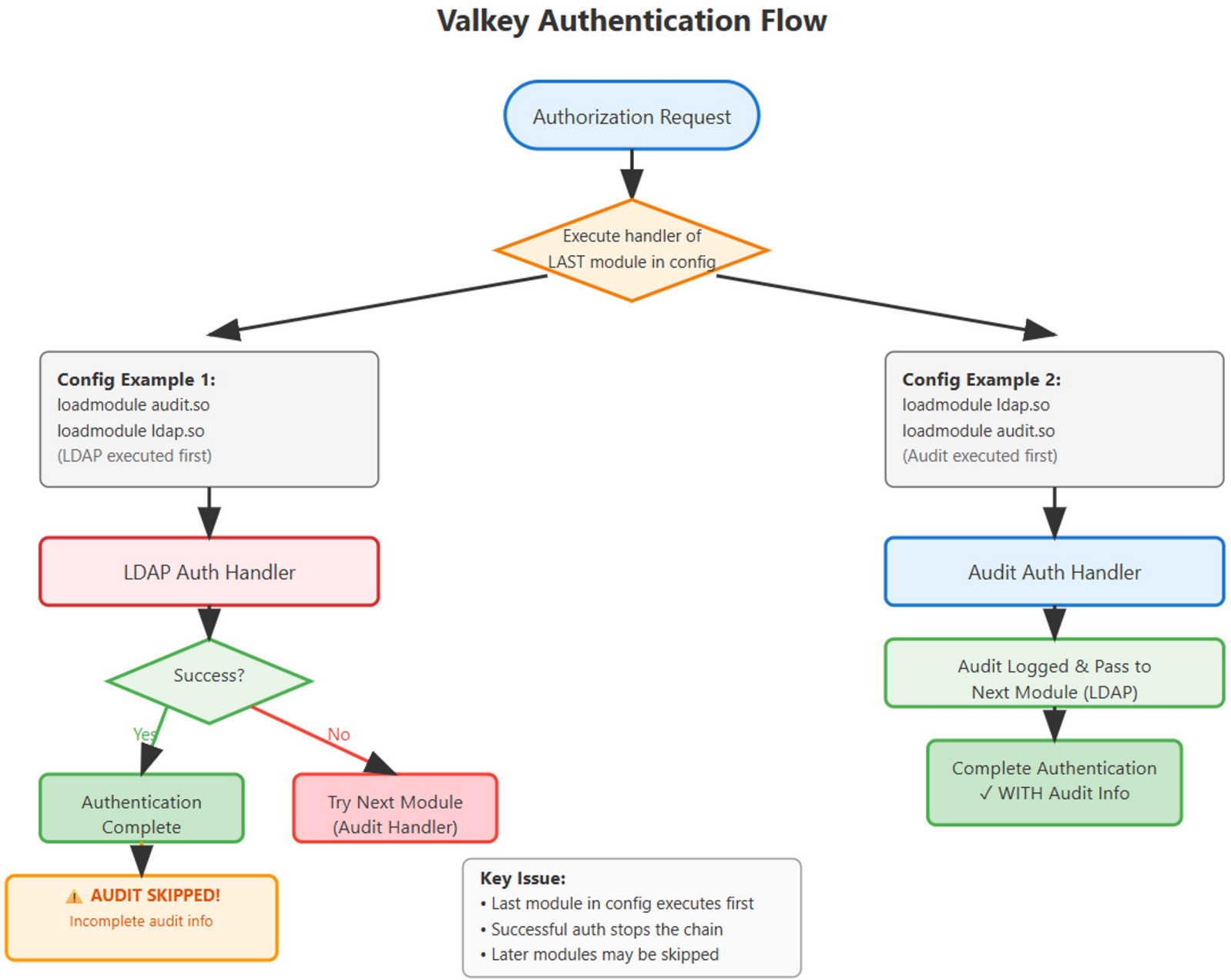
  ```
  audit.flush_threshold_bytes 8192
  ```

- Buffer flush based on time

  ```
  audit.flush_interval_seconds 1
  ```

Combining with ValkeyLDAP

Audit module must always be loaded last



Performance impact - wip

- performance is impacted
- early exclusion best for performance
- exclusion order
  - audit.enabled
  - excluded users (with exception of always audit_config_commands)
  - events filter

Performance impact

Test Configuration	RPS SET	RPS GET	AvgLat SET	AvgLat GET
1. Module not loaded	100.0%	100.0%	100.0%	100.0%
2. Loaded but enabled=no	96.9%	99.7%	103.5%	100.6%
3. Excluded user, always_audit_config=no	98.7%	102.7%	101.3%	97.7%
4. Excluded user, always_audit_config=no	98.6%	100.6%	101.2%	99.9%
5. No excluded user, events=all,	80.1%	82.7%	127.3%	123.2%

Further work

1. addition of metrics
2. performance optimizations
3. AUTH handling based on Valkey server PR
4. operation success/failure functionality - dependency on Valkey server change
5. further event categorization
6. exclusion/inclusion based on prefixes

Adoption by Valkey community?

Other requirements?

https://github.com/martinrvisser/valkey-audit : get involved

PERCONA

Questions?

https://percona.com/training