

volcengine / OpenViking Public

<> Code Issues 95 Pull requests 92 Discussions Actions Projects

⚠ This commit does not belong to any branch on this repository, and may belong to a fork outside of the repository.

Commit 0251c70



gin-ctx and claude committed on Feb 26 · ✓ 5 / 6



fix(server): 未配置 root_api_key 时仅允许 localhost 绑定

当 root_api_key 未配置时 resolve_identity() 将所有请求解析为 ROOT，结合默认绑定 0.0.0.0 会导致任何网络请求均可执行管理员操作。

- 将默认 host 从 0.0.0.0 改为 127.0.0.1
- 添加 validate_server_config() 启动校验：无 key + 非 localhost 时拒绝启动
- 将 dev mode 日志从 info 升级为 warning
- 更新中英文认证文档的开发模式段落

Closes [#302](#)

Co-Authored-By: Claude Opus 4.6 <noreply@anthropic.com>


1 parent [1a40839](#) commit 0251c70

5 files changed +100 -9 lines changed

↑ Top

🔍 Filter files...

- docs
 - en/guides
 - 04-authentication.md
 - zh/guides
 - 04-authentication.md
- openviking/server
 - app.py

 config.py


 tests/server

 test_auth.py

 **5 files changed** +100 -9 lines changed



docs/en/guides/04-authentication.md


```

@@ -101,17 +101,19 @@ client = ov.SyncHTTPClient(
101 101
102 102  ## Development Mode
103 103
104 104 - When no `root_api_key` is configured, authentication is disabled. All requests
    are accepted as ROOT with the default account.
104 104 + When no `root_api_key` is configured, authentication is disabled. All requests
    are accepted as ROOT with the default account. **This is only allowed when the
    server binds to localhost** (`127.0.0.1`, `localhost`, or `:::1`). If `host` is
    set to a non-loopback address (e.g. `0.0.0.0`) without a `root_api_key`, the
    server will refuse to start.
105 105
106 106  ```json
107 107  {
108 108  "server": {
109 109 -   "host": "0.0.0.0",
109 109 +   "host": "127.0.0.1",
110 110   "port": 1933
111 111  }
112 112  }
113 113  ```
114 114
115 115 + > **Security note:** The default `host` is `127.0.0.1`. If you need to expose
    the server on the network, you **must** configure `root_api_key`.
116 116 +
115 117  ## Unauthenticated Endpoints
116 118
117 119  The `/health` endpoint never requires authentication. This allows load
    balancers and monitoring tools to check server health.
  
```



```

docs/zh/guides/04-authentication.md
@@ -101,17 +101,19 @@ client = ov.SyncHTTPClient(
101 101
102 102  ## 开发模式
103 103
104 - 不配置 `root_api_key` 时，认证禁用。所有请求以 ROOT 身份访问 default account。
104 + 不配置 `root_api_key` 时，认证禁用，所有请求以 ROOT 身份访问 default account。**此模式仅允许在服务器绑定 localhost 时使用**（`127.0.0.1`、`localhost` 或 `::1`）。如果 `host` 设置为非回环地址（如 `0.0.0.0`）且未配置 `root_api_key`，服务器将拒绝启动。
105 105
106 106  ````json
107 107  {
108 108  "server": {
109 - "host": "0.0.0.0",
109 + "host": "127.0.0.1",
110 110  "port": 1933
111 111  }
112 112  }
113 113  ````
114 114
115 + > **安全提示：** 默认 `host` 为 `127.0.0.1`。如需将服务暴露到网络，**必须**配置 `root_api_key`。
116 +
115 117  ## 无需认证的端点
116 118
117 119  `/health` 端点始终不需要认证，用于负载均衡器和监控工具检查服务健康状态。

```

```

openviking/server/app.py
@@ -11,7 +11,7 @@
11 11  from fastapi.responses import JSONResponse
12 12
13 13  from openviking.server.api_keys import APIKeyManager
14 - from openviking.server.config import ServerConfig, load_server_config
14 + from openviking.server.config import ServerConfig, load_server_config,
    validate_server_config
15 15  from openviking.server.dependencies import set_service

```

```

16 16     from openviking.server.models import ERROR_CODE_TO_HTTP_STATUS, ErrorInfo,
      Response
17 17     from openviking.server.routers import (
      @@ -50,6 +50,8 @@ def create_app(
      ↓
      ↑
50 50         if config is None:
51 51             config = load_server_config()
52 52
53 53 +         validate_server_config(config)
54 54 +
53 55         @asynccontextmanager
54 56         async def lifespan(app: FastAPI):
55 57             """Application lifespan handler."""
      ⚙
      @@ -72,7 +74,13 @@ async def lifespan(app: FastAPI):
72 74             logger.info("APIKeyManager initialized")
73 75             else:
74 76                 app.state.api_key_manager = None
75 75 -                 logger.info("Dev mode: no root_api_key configured, authentication
      disabled")
77 77 +                 logger.warning(
78 78 +                     "Dev mode: no root_api_key configured, authentication disabled.
      "
79 79 +                     "This is allowed because the server is bound to localhost (%s).
      "
80 80 +                     "Do NOT expose this server to the network without configuring "
81 81 +                     "server.root_api_key in ov.conf.",
82 82 +                     config.host,
83 83 +                 )
76 84
77 85         yield
78 86
      ↓

```

openviking/server/config.py

...

```

      ↑
      @@ -2,22 +2,26 @@
2 2     # SPDX-License-Identifier: Apache-2.0
3 3     """Server configuration for OpenViking HTTP Server."""
4 4
5 5 + import sys
5 6     from dataclasses import dataclass, field

```

```

6   7   from typing import List, Optional
7   8
9   + from openviking_cli.utils import get_logger
8   10  from openviking_cli.utils.config.config_loader import (
9   11      DEFAULT_OV_CONF,
10  12      OPENVIKING_CONFIG_ENV,
11  13      load_json_config,
12  14      resolve_config_path,
13  15  )
14  16
17  + logger = get_logger(__name__)
18  +
15  19
16  20  @dataclass
17  21  class ServerConfig:
18  22      """Server configuration (from the ``server`` section of ov.conf)."""
19  23
20  -   host: str = "0.0.0.0"
24  +   host: str = "127.0.0.1"
21  25      port: int = 1933
22  26      root_api_key: Optional[str] = None
23  27      cors_origins: List[str] = field(default_factory=lambda: ["*"])
@@ -59,10 +63,47 @@ def load_server_config(config_path: Optional[str] =
None) -> ServerConfig:
59  63      server_data = data.get("server", {})
60  64
61  65      config = ServerConfig(
62  -   host=server_data.get("host", "0.0.0.0"),
66  +   host=server_data.get("host", "127.0.0.1"),
63  67      port=server_data.get("port", 1933),
64  68      root_api_key=server_data.get("root_api_key"),
65  69      cors_origins=server_data.get("cors_origins", ["*"]),
66  70  )
67  71
68  72      return config
73  +
74  +
75  + _LOCALHOST_HOSTS = {"127.0.0.1", "localhost", ":::1"}
76  +
77  +

```

```

78 + def _is_localhost(host: str) -> bool:
79 +     """Return True if *host* resolves to a loopback address."""
80 +     return host in _LOCALHOST_HOSTS
81 +
82 +
83 + def validate_server_config(config: ServerConfig) -> None:
84 +     """Validate server config for safe startup.
85 +
86 +     When ``root_api_key`` is not set, authentication is disabled (dev mode).
87 +     This is only acceptable when the server binds to localhost. Binding to a
88 +     non-loopback address without authentication exposes an unauthenticated ROOT
89 +     endpoint to the network.
90 +
91 +     Raises:
92 +         SystemExit: If the configuration is unsafe.
93 +     """
94 +     if config.root_api_key:
95 +         return
96 +
97 +     if not _is_localhost(config.host):
98 +         logger.error(
99 +             "SECURITY: server.root_api_key is not configured and server.host "
100 +             "is '%s' (non-localhost). This would expose an unauthenticated "
101 +             "ROOT endpoint to the network.",
102 +             config.host,
103 +         )
104 +         logger.error(
105 +             "To fix, either:\n"
106 +             " 1. Set server.root_api_key in ov.conf, or\n"
107 +             " 2. Bind to localhost (server.host = \"127.0.0.1\")'
108 +         )
109 +         sys.exit(1)

```

tests/server/test_auth.py

↑

@@ -4,10 +4,11 @@

4 4 """Tests for multi-tenant authentication (openviking/server/auth.py)."""

5 5

6 6 import httpx

7 + import pytest

7 8 import pytest_asyncio

```

8     9
9    10     from openviking.server.app import create_app
10   11     - from openviking.server.config import ServerConfig
11   12     + from openviking.server.config import ServerConfig, _is_localhost,
12   13     validate_server_config
13   14     from openviking.server.dependencies import set_service
14   15     from openviking.service.core import OpenVikingService
15   16     from openviking_cli.session.user_id import UserIdentifier
16   17
17   18     @@ -199,3 +200,40 @@ async def
18   19     test_cross_tenant_session_get_returns_not_found(auth_client: httpx.Asy
19   20     )
20   21     assert cross_get.status_code == 404
21   22     assert cross_get.json()["error"]["code"] == "NOT_FOUND"
22   23
23   24     +
24   25     + # ---- _is_localhost tests ----
25   26     +
26   27     +
27   28     + @pytest.mark.parametrize("host", ["127.0.0.1", "localhost", ":::1"])
28   29     + def test_is_localhost_true(host: str):
29   30     +     assert _is_localhost(host) is True
30   31     +
31   32     +
32   33     + @pytest.mark.parametrize("host", ["0.0.0.0", ":::", "192.168.1.1", "10.0.0.1"])
33   34     + def test_is_localhost_false(host: str):
34   35     +     assert _is_localhost(host) is False
35   36     +
36   37     +
37   38     + # ---- validate_server_config tests ----
38   39     +
39   40     +
40   41     + def test_validate_no_key_localhost_passes():
41   42     +     """No root_api_key + localhost should pass validation."""
42   43     +     for host in ("127.0.0.1", "localhost", ":::1"):
43   44     +         config = ServerConfig(host=host, root_api_key=None)
44   45     +         validate_server_config(config) # should not raise
45   46     +
46   47     +
47   48     + def test_validate_no_key_non_localhost_raises():
48   49     +     """No root_api_key + non-localhost should raise SystemExit."""

```

```
230 + config = ServerConfig(host="0.0.0.0", root_api_key=None)
231 + with pytest.raises(SystemExit):
232 +     validate_server_config(config)
233 +
234 +
235 + def test_validate_with_key_any_host_passes():
236 +     """With root_api_key set, any host should pass validation."""
237 +     for host in ("0.0.0.0", ":::", "192.168.1.1", "127.0.0.1"):
238 +         config = ServerConfig(host=host, root_api_key="some-secret-key")
239 +         validate_server_config(config) # should not raise
```

Comments 0



Please [sign in](#) to comment.