TACHYON tkctl 사용자 메뉴얼

Version 0.6.2

Date: 2026년 02월 26일

Author: INCA Incternet Co., Ltd.

📋 목차

소개

TKCTL은 TACHYON 시스템을 효율적으로 관리하기 위한 통합 명령줄 도구입니다. 서비스 상태 점검, 데이터베이스 용량 확인, 시스템 정보 조회 등 다양한 관리 기능을 제공합니다.

1. 설치 및 설정

1.1 설치 및 설정

`tkctl`은 단일 바이너리 형태로 배포됩니다. 시스템 환경에 맞는 바이너리를 다운로드하여 서버의 적절한 경로(예: `/usr/local/bin`)에 배치합니다.

실행 권한 부여

바이너리를 내려받은 후 반드시 실행 권한을 부여해야 합니다.

```
# 실행 파일에 권한 부여 (예: /usr/local/bin 에 위치한 경우)
chmod +x /usr/local/bin/tkctl
```

원클릭 환경 설정 (추천)

권한 부여가 완료되면 다음 명령어를 실행하여 현재 사용 중인 셸(Bash, Zsh 등)에 `tkctl` 통합 설정을 자동으로 반영합니다.

```
tkctl -i
```

이 명령어는 다음 작업을 수행합니다:

- **자동 완성 등록**: 명령어 입력 시 `Tab` 키를 통한 자동 완성 기능 활성화
- **셸 프로필 업데이트**: `~/.bashrc` 또는 `~/.zshrc`에 필요한 환경 정보 추가

자동 완성 자동 설치

별도의 초기 설정을 진행하지 않고 `tkctl`를 처음 실행하면 자동 설치를 묻지만, 다음과 같이 명시적으로 설정 제거 및 재설치를 수행할 수도 있습니다.

```
[root@localhost ~]# tkctl --uninstall
🗑️  tkctl 설정을 제거합니다...

📝 Bash 자동 완성 스크립트를 제거합니다...
  ✓ /etc/bash_completion.d/tkctl 제거됨
```

```
   ✓ 자동 완성 확인 플래그 제거됨
   ✓ 자동 완성 스크립트가 성공적으로 제거되었습니다

2026-01-17 19:31:26[SUCC] /root/.bashrc에서 레거시 설정이 제거되었습니다.
[root@localhost ~]#
[root@localhost ~]# tkctl --install -y
2026-01-17 19:31:33[SUCC] 설정 완료! /root/.bashrc에 적용되었습니다. 'source
/root/.bashrc'를 실행하거나 쉘을 재시작하십시오.
[root@localhost ~]#
```

1.2 구성 관리

`tkctl`은 실행 파일과 동일한 위치에 생성되는 `tkctl.ini` 파일을 통해 주요 설정을 관리합니다.

설정 키	설명	기본값
basedir	TACHYON 설치 루트 경로	`/usr/local/TACHYON/TTS40`
admin_api_url	tkadmin 연동 서버 주소	`http://127.0.0.1:13700`
log_level	로그 기록 수준 (DEBUG, INFO, WARN, ERROR)	`DEBUG`

1.2.1 tkadmin 연동 상세 로직

`tkctl`이 관리자 서버(`tkadmin`)로 작업 리포트를 보낼 때 사용하는 URL은 자동으로 구성됩니다.

1. **연동 URL 결정 우선순위**:
    - **1순위**: `tkctl.ini` 내의 `admin_api_url` 설정값.
    - **2순위 (자동 감지)**: `tkctl` 실행 폴더에 `tkadmin.yml`이 존재할 경우, 해당 파일의 `port` 설정을 읽어 `http://127.0.0.1:{port}` 로 자동 구성.
    - **3순위**: 기본값 `http://127.0.0.1:13700` 사용.
2. **통신 엔드포인트**:
    - 실제 보고는 설정된 주소 뒤에 `/tkadmin/api/monitor/report` 경로가 자동으로 붙어 수행됩니다.
    - 예: `admin_api_url`이 `http://192.168.0.100:13700`인 경우, 실제 통신 대상은 `http://192.168.0.100:13700/tkadmin/api/monitor/report`가 됩니다.
3. **적용 시점**:
    - 설정 변경 후 즉시 적용되며, 별도의 서비스 재시작은 필요하지 않습니다 (CLI 성격상 매 수행 시 설정을 로드함).

2. 기본 사용법

2.1 명령어 형식

```
tkctl [command] [subcommand] [arguments] [flags]
```

> **Tip**
>
> 대화형 쉘 모드로 진입하려면 `--shell` (또는 `-s`) 플래그를 사용하십시오. 자세한 내용은 4. 대화형 쉘 모드를 참고하세요.

2.2 도움말 확인

```
# 전체 도움말
tkctl --help

# 특정 명령어 도움말
tkctl service --help
tkctl size --help
```

2.3 버전 확인

```
# 간단한 버전 정보
tkctl --version

#상세 버전 정보 (미들웨어 버전 포함)
tkctl version
```

2.4 환경 설정 및 진단 (env)

SELinux, 시스템 리미트(`ulimit`) 등 시스템 환경 설정을 확인하고 최적화합니다.

```
# SELinux 상태 확인
tkctl env selinux

# 시스템 및 서비스 리미트 상태 진단
tkctl env ulimit

# 권장 설정 자동 적용
tkctl env ulimit --set
```

3. 주요 명령어

`tkctl`은 TACHYON 시스템의 설치, 운영, 유지보수를 위한 다양한 명령어를 제공합니다. 이 섹션에서는 모든 명령어의 기능과 주요 플래그를 요약하여 제공합니다. 상세한 사용법은 사이드 메뉴의 각 항목을 참고하십시오.

📋 명령어 요약 인덱스

1. 기본 시스템 관리 (`tkctl ...`) 상세 보기: 3.1 기본 시스템 관리

명령어	기능	주요 플래그 (옵션)
`version`	시스템 및 컴포넌트 버전 상세 조회	없음
`info`	시스템 리소스 및 디스크 사용량 조회	없음
`admin`	관리자 비밀번호 초기화	`[user_id]` (기본: admin)
`completion`	쉘 자동 완성 스크립트 생성	`[bash/zsh/fish/powershell]`
`update`	서비스 업데이트 및 패키지 관리	`[service] -f <file>` (업데이트) `package [service]` (패키지 생성)

2. 서비스 관리 (`tkctl service ...`) 상세 보기: 3.2 서비스 관리

명령어	기능	주요 플래그 (옵션)
`check`	전체 서비스 상태 점검	없음
`start`	서비스 시작	`[service]` (생략 시 all)
`stop`	서비스 중지	`[service]` (생략 시 all)
`restart`	서비스 재시작	`[service]` (생략 시 all)
`jvm`	JVM 메모리 최적화	`--set [auto/S/M/L/size]` (설정 적용) `--dry-run` (미리보기)
`loglevel`	로그 레벨 조회/설정	`[service] --set [LEVEL]`
`logrotate`	로그 로테이션 관리	`--set` (권장 설정 자동 적용)

3. 모니터링 및 분석 (`tkctl size` , `tkctl analyze ...`) 상세 보기: 3.3 모니터링 및 분석

명령어	서브커맨드	기능	주요 플래그 (옵션)
`size`	`tables`	MariaDB 테이블 용량	`[count]` , `--by-month` (월별 집계)
`size`	`opensearch`	OpenSearch 인덱스 용량	`--by-month` (월별 집계)
`analyze`	`recommend`	분석 추천 리포트	`--month` (대상 월)
`analyze`	`media-top`	매체제어 빈도 분석	`--month` , `--limit` , `--output` (CSV 저장)
`analyze`	`agent-anomaly`	에이전트 이상 로깅 탐지	`--month` , `--threshold` (임계치)
`analyze`	`process-top`	프로세스별 로그 집계	`--month` , `--limit`
`analyze`	`db-trend`	DB 데이터 증가 추세	`--output`

4. 백업 및 복원 (`tkctl backup` , `tkctl restore ...`) 상세 보기: 3.4 백업 및 복원

명령어	서브커맨드	기능	주요 플래그 (옵션)
`backup`	`mariadb`	MariaDB 백업	`full`, `incr`, `log`, `config` (유형 지정)
`backup`	`mariadb partition`	파티션 관리	`list`, `drop`, `backup`
`backup`	`opensearch`	스냅샷 생성	`--init-repo` (저장소 초기화)
`backup`	`opensearch index`	인덱스 관리	`list`, `delete`, `backup`
`backup`	`opensearch ilm`	ILM 정책 관리	`status`, `apply`
`restore`	`mariadb`	MariaDB 복원	`--target-date`, `--target-dir`
`restore`	`mariadb partition`	파티션 복원	`--table`, `--partition`, `--file`
`restore`	`opensearch`	스냅샷 복원	`[snapshot]`, `--indices`

5. 환경 구성 (`tkctl env ...`) 상세 보기: 3.5 환경 구성

명령어	기능	주요 플래그 (옵션)
`(기본)`	전체 환경 설정 상태 요약	`--set` (전체 권장 값 적용)
`selinux`	SELinux 설정	`[mode]`, `--set`
`ulimit`	시스템 리미트(Open Files) 설정	`[value]`, `--set`
`firewall`	방화벽 포트 개방	`[port]`, `--set`
`ssh`	SSH 포트 변경	`[port]`, `--set`
`web`	웹(HTTPS) 포트 변경	`[port]`, `--set`
`db`	DB(MariaDB) 포트 변경	`[port]`, `--set` (관련 서비스 일괄 적용)

6. 보안 진단 및 조치 (`tkctl analyze security`, `tkctl fix security`) 상세 보기: 보안 진단 및 조치

명령어	기능	주요 플래그 (옵션)
`analyze security`	보안 취약점 진단	`--target` (대상), `--format` (HTML 리포트), `--quiet`
`fix security`	취약점 조치	`--id` (취약점 ID), `--dry-run` (시뮬레이션), `--backup`

3.1 기본 시스템 관리

이 섹션에서는 시스템 전반의 상태를 확인하고 유지보수하는 기본 명령어(`version` , `info` , `admin` , `completion` , `update`)를 다룹니다.

3.1.1 버전 정보 (version)

명령어	기능	주요 플래그 (옵션)
`(기본)`	시스템 및 컴포넌트 버전 상세 조회	없음

TKCTL 및 관련 미들웨어의 버전 정보를 확인합니다. 모든 TACHYON 컴포넌트와 미들웨어의 상세 버전을 테이블 형식으로 제공하며, `--help` 를 통해 상세 설명을 확인할 수 있습니다.

사용법

```
tkctl version [flags]
```

출력 예시

```
2025-12-25 21:02:15 [INFO] ### TACHYON 서비스 버전 확인 ###
tkctl 버전: 0.3.30

--- Tachyon 엔진 ---
------------------------------------------------------------------------
  컴포넌트                      버전
------------------------------------------------------------------------
  MariaDB                       10.11.2
  NGINX                         1.29.4
  Redis                         8.4.0
```

```
  Kafka                    3.4.0
  Zookeeper                3.6.3
  OpenSearch               2.19.4
  OpenSearch Dashboards    2.8.0
  Logstash                 8.6.1
------------------------------------------------------------------

리비전 = 4.0.0.314
------------------------------------------------------------------------------
  컴포넌트              패치 버전            JAR 버전
------------------------------------------------------------------------------
  Front                 1.1.59.1             V4.0.0.1-20251104
  API                   1.1.50               V4.0.0.1-20251027
  Auth                  1.1.3                V4.0.0.1-20250915
  ...
------------------------------------------------------------------------------
2025-12-24 21:02:15 [SUCC] 버전 확인 완료.
```

3.1.2 시스템 정보 (info)

명령어	기능	주요 플래그 (옵션)
(기본)	시스템 리소스 및 디스크 사용량 조회	없음

라이선스 정보, 시스템 리소스, 디스크 사용량, 서비스 상태를 한눈에 확인합니다. 상세 정보는 `/usr/local/TACHYON` 경로 및 관련 미들웨어 데이터를 자동으로 수집하여 표시합니다.

사용법

```
tkctl info [flags]
```

출력 예시

```
2025-12-24 21:02:27 [INFO] ### TACHYON 시스템 정보 ###
--- 라이선스 및 기업 정보 ---
------------------------------------------------------
  자산 항목              값
------------------------------------------------------
  설치 기관명           TSGroup -오프라인
  기관 코드              00133FKFKG
```

```
  에이전트 수            100
------------------------------------------------------

--- 시스템 리소스 ---
----------------------------------------
  리소스                    사용량
----------------------------------------
  전체 물리 메모리       15 GB
  여유 메모리             0 GB
----------------------------------------

--- 디스크 사용량 ---
------------------------------------------------------------------------------
--------------------
  분류                                  크기         경로
------------------------------------------------------------------------------
--------------------
  설치 디렉토리                      7.7G        /usr/local/TACHYON
  데이터베이스                       2.1G
/usr/local/TACHYON/TTS40/mariadb/data
  OpenSearch                          15G
/usr/local/TACHYON/TTS40/opensearch/opensearch/data
------------------------------------------------------------------------------
--------------------

2025-12-24 21:02:28 [SUCC] 서비스 상태 확인 완료.
```

> **참고: OS 메모리 과점유 현황 안내** `tkctl info` 에서 리포팅되는 메모리 사용량은 운영체제 레벨의 물리적 점유 현황을 포함합니다. 특히 Linux 환경에서는 성능 최적화를 위해 유휴 메모리를 **Page Cache/Buffer**로 활용하므로, 실제 프로세스가 사용하는 양보다 높게 표시되는 '과점유' 현상이 발생할 수 있습니다. 이는 시스템 장애가 아닌 정상적인 운영체제 성능 관리의 결과이며, 상세 기술 설명은 **[부록 A - 8. OS 메모리 관리 명세]**를 참조하십시오.

3.1.3 관리자 비밀번호 재설정 (admin)

명령어	기능	주요 플래그 (옵션)
`(기본)`	관리자 비밀번호 초기화	`[user_id]` : 대상 사용자 ID (기본: admin)

관리자 계정의 비밀번호를 초기화합니다.

사용법

```
tkctl admin <USER_ID>
```

초기화 비밀번호

`!TumsAdmin2025`

예제

```
# admin 계정 비밀번호 재설정
tkctl admin admin

# 특정 사용자 비밀번호 재설정
tkctl admin user123
```

출력 예시

```
2025-12-07 16:15:00[INFO] 사용자 ID 'admin'의 비밀번호를 업데이트합니다...
2025-12-07 16:15:01[SUCC] 사용자 'admin'의 비밀번호 업데이트를 완료했습니다.
2025-12-07 16:15:01[INFO] 새로운 비밀번호와 키가 설정되었습니다.
```

3.1.4 자동 완성 (completion)

명령어	기능	주요 플래그 (옵션)
`(기본)`	쉘 자동 완성 스크립트 생성	`[bash/zsh/fish/powershell]`

다양한 셸에 대한 자동 완성 스크립트를 생성합니다.

지원 셸

- Bash
- Zsh
- Fish

사용법

```
tkctl completion [bash|zsh|fish|powershell]
```

설치 방법

Bash (Linux):

```
# 시스템 전역
tkctl completion bash | sudo tee /etc/bash_completion.d/tkctl

# 사용자별
tkctl completion bash > ~/.local/share/bash-completion/completions/tkctl
```

Zsh:

```
# Completion 활성화 (처음 한 번만)
echo "autoload -U compinit; compinit" >> ~/.zshrc

# Completion 스크립트 설치
tkctl completion zsh > "${fpath[1]}/_tkctl"
```

Fish:

```
tkctl completion fish > ~/.config/fish/completions/tkctl.fish
```

v0.4.20+ Tab Completion 기능 강화

v0.4.20부터 모든 파일 경로 완성이 비활성화되고, 지능형 자동 완성 기능이 대폭 강화되었습니다.

정적 완성 (Static Completion):

적용된 플래그에 대해 사전 정의된 값을 즉시 제안합니다.

플래그	적용 명령어	완성 값
`--format`	`analyze`, `backup`	`json`, `csv`, `table`
`--level`	전역	`debug`, `info`, `warn`, `error`
`--target`	`service`, `fix`	`all`, `tachyon`, `middleware`
`--shell`	`completion`	`bash`, `zsh`, `fish`
`--status`	`analyze security`	`vulnerable`, `good`, `manual`
`--source`	`analyze`	`opensearch`, `mariadb`, `filesystem`

동적 완성 (Dynamic Completion):

OpenSearch 연결을 통해 실시간으로 인덱스/필드 목록을 조회하여 제안합니다.

플래그	적용 명령어	동작
`--index`	`backup opensearch`, `analyze`	실시간 인덱스 목록 조회
`--field`	`analyze media-top` 등	선택된 인덱스의 필드 목록 조회
`--pattern`	`backup opensearch ilm`	인덱스 패턴 제안
`--policy`	`backup opensearch ilm`	ILM 정책 목록 조회
`--alias`	`backup opensearch`	별칭 목록 조회

사용 예시:

```
# 정적 완성
tkctl analyze --format <Tab>
# json  csv  table

# 동적 완성 (OpenSearch 연결 필요)
tkctl backup opensearch index --index <Tab>
# tachyon-agent-2026.01  tachyon-media-2026.01  ...
```

3.1.5 서비스 업데이트 (update)

명령어	기능	주요 플래그 (옵션)
`nginx`	Nginx 업데이트	`-f` : 패키지 파일 경로 (필수) `--force` : 버전 체크 무시 강제 진행
`redis`	Redis 업데이트	`-f` : 패키지 파일 경로 (필수)
`opensearch`	OpenSearch 업데이트	`-f` : 패키지 파일 경로 (필수) `--force` : 버전 체크 무시 강제 진행

TACHYON 시스템의 주요 미들웨어 서비스(Nginx, Redis, OpenSearch)를 업데이트합니다. 단순한 파일 교체가 아닌, **버전 검증 시스템**이 내장되어 있어 잘못된 버전 설치로 인한 시스템 장애를 방지합니다.

사용법

```
tkctl update [service] [flags]
```

옵션

- `--file` , `-f` : 업데이트할 패키지 파일(`.tar.gz`)의 경로를 지정합니다.
- `--force` : 버전 안전성 체크를 무시하고 강제로 업데이트를 진행합니다. (Downgrade 환경에서 유용)

안전 장치 (Safety Guard)

1. **다운그레이드 차단**: 설치하려는 패키지의 버전이 현재 시스템에 설치된 버전보다 낮을 경우 업데이트를 자동으로 중단합니다.
2. **메이저 버전 보호**: 버전의 첫 번째 숫자(Major)가 다를 경우(예: v2 → v3) 경고를 출력하여 호환성 문제를 사전에 인지시킵니다.
3. **무결성 검사**: 압축 해제 후 바이너리를 직접 실행하여 버전 정보를 추출하고 검증합니다.

예제 (상세 로그)

```
$ tkctl update nginx -f nginx-1.29.4.tar.gz

2024-12-21 15:30:10 [INFO] nginx 업데이트 시작...
2024-12-21 15:30:10 [INFO] 버전 검증: 현재 1.29.3 -> 대상 1.29.4 (안전)
...
2024-12-21 15:30:15 [SUCC] nginx 업데이트가 완료되었습니다.
```

3.1.6 업데이트 패키지 생성 (update package) [DEPRECATED]

운영 환경(폐쇄망 등)에 배포할 미들웨어 패키지를 생성합니다. 인터넷이 연결된 환경에서 최신 소스를 다운로드하고, 운영 서버 사양에 맞춰 바이너리를 컴파일 및 재포장합니다.

사용법

```
tkctl update package [service]
```

지원 서비스

- `nginx` : 최신 안정 버전 소스 다운로드 및 OpenSSL/PCRE2/Zlib 정적 컴파일 포함.
- `redis` : 최신 소스 다운로드 및 컴파일.
- `opensearch` : 공식 바이너리 기반 재포장.
- `mariadb` : MariaDB REST API 기반 바이너리 tarball 다운로드 (LTS 10.11 기본). REST API 실패 시 `archive.mariadb.org` 폴백 지원.

빌드 의존성 자동 확인 (Build Dependency Guard)

패키징 작업 전, 시스템에 필수 빌드 도구가 있는지 자동으로 검사합니다.

1. **검사항목**: `gcc` , `make` , `perl` (FindBin 모듈 포함), `tar` , `curl`
2. **자동 설치**: root 권한이 있는 경우, 누락된 도구를 `dnf` 또는 `apt` 를 통해 자동으로 설치합니다.
3. **수동 설치 안내**: 권한이 없거나 자동 설치가 불가능한 경우, 해당 시스템에 최적화된 설치 명령어를 화면에 안내합니다.

```
$ tkctl update package nginx

2025-12-23 11:00:05 [INFO] 빌드 의존성 확인 중...
2025-12-23 11:00:05 [WARN] 누락된 의존성 발견: gcc, perl
2025-12-23 11:00:05 [INFO] 필수 패키지를 자동으로 설치합니다...
...
2025-12-23 11:02:40 [SUCC] 빌드 의존성 설치 완료. 패키징을 계속합니다.
```

정상적인 업그레이드 상황

```
$ tkctl update opensearch --file opensearch-2.19.4.tar.gz
```

출력 결과:

```
2025-12-20 12:16:34[INFO] 대상 서비스: opensearch, 파일:
/root/packages/opensearch-2.19.4.tar.gz
2025-12-20 12:16:34[INFO] OpenSearch 업데이트를 시작합니다 (원본:
/root/packages/opensearch-2.19.4.tar.gz)
2025-12-20 12:16:34[INFO] opensearch 서비스를 중지하는 중...
2025-12-20 12:16:34[INFO] 압축 해제 중...
2025-12-20 12:16:49[INFO] 버전 확인: [현재: 2.18.0] -> [신규: 2.19.4]
2025-12-20 12:16:49[SUCC] 버전 업그레이드 확인: 2.18.0 -> 2.19.4
2025-12-20 12:16:49[INFO] 파일 동기화 중(rsync)...
2025-12-20 12:16:53[INFO] opensearch 서비스를 시작하는 중...
2025-12-20 12:16:54[SUCC] OpenSearch 업데이트를 성공적으로 마쳤습니다.
```

```
# 강제 복구(Downgrade)가 필요한 상황
tkctl update opensearch --file opensearch-2.19.4.tar.gz --force

# 출력 결과:
2025-12-20 12:24:49[INFO] 버전 확인: [현재: 3.4.0] -> [신규: 2.19.4]
2025-12-20 12:24:49[WARN] 치명적: 메이저 버전 불일치! 호환성 문제가 발생할 수 있습니다.
2025-12-20 12:24:49[WARN] --force 플래그에 의해 안전 검사를 건너뜁니다. 이전 버전으로 진
행합니다 (2.19.4 < 3.4.0).
2025-12-20 12:24:49[INFO] 파일 동기화 중(rsync)...
2025-12-20 12:24:54[SUCC] OpenSearch 업데이트를 성공적으로 마쳤습니다.
```

3.1.7 배포용 패키지 생성 (update package)

명령어	기능	주요 플래그 (옵션)
`nginx`	Nginx 패키지 생성	`--version`, `--clean-cache`
`redis`	Redis 패키지 생성	`--version`, `--clean-cache`
`opensearch`	OpenSearch 패키지 다운로드	`--version`, `--clean-cache`
`mariadb`	MariaDB 패키지 다운로드	`--version`, `--clean-cache`

플래그

플래그	단축	설명
`--version`	`-V`	특정 버전 지정 (예: `--version 10.11.11`)
`--clean-cache`	-	캐시 디렉토리 정리 후 빌드

전문 엔지니어가 폐쇄망 환경 배포를 위해 최신 버전의 미들웨어를 다운로드, 컴파일 및 패키징하는 전문 기능을 제공합니다. 이 기능은 운영 환경의 **호환성을 유지**하기 위해 현재 설치된 메이저 버전을 추적합니다.

특징

- **메이저 버전 보호**: 현재 v1.x를 사용 중이라면 v2.x가 출시되어도 v1.x 계열의 최신 패치 버전을 자동으로 찾아 패키징합니다.
- **의존성 자동 해결**: Nginx 빌드 시 필요한 PCRE2, OpenSSL, Zlib 등의 라이브러리를 소스와 함께 자동으로 다운로드하여 정적 빌드합니다.
- **표준화된 파일명**: `nginx-1.29.4.tar.gz`와 같이 버전이 명시된 파일명으로 패키지를 생성합니다.

예제

```
# Nginx 최신 패치 패키지 생성
tkctl update package nginx

# 출력 결과:
2025-12-20 12:12:40[INFO] 인터넷 연결 확인 중...
2025-12-20 12:12:41[SUCC] 인터넷 연결이 확인되었습니다.
2025-12-20 12:12:41[INFO] 현재 설치된 버전: 1.29.3 (메이저: 1)
2025-12-20 12:12:41[INFO] nginx의 최신 안정 버전을 감지하는 중 (메이저 1 필터링)...
2025-12-20 12:12:42[INFO] 대상 버전(1.29.4)이 현재(1.29.3)보다 최신입니다. 패키지 업데
이트를 권장합니다.
2025-12-20 12:12:43[INFO]   - 의존성 다운로드 중: pcre2-10.45
2025-12-20 12:12:48[INFO]   - 설정 중 (configure): --with-pcre=../pcre2-10.45
...
2025-12-20 12:12:53[INFO]   - 컴파일 중...
2025-12-20 12:14:25[SUCC] 패키지가 성공적으로 생성되었습니다:
/root/dist/packages/nginx-1.29.4.tar.gz
```

```
# MariaDB 최신 LTS 패키지 생성 (버전 미지정 시 자동 감지)
tkctl update package mariadb

# 출력 결과:
2026-02-09 16:59:51[INFO] 인터넷 연결 확인 중...
2026-02-09 16:59:52[SUCC] 인터넷 연결이 확인되었습니다.
2026-02-09 16:59:52[INFO] 현재 설치된 버전: 10.11.2 (메이저: 10.11)
```

```
2026-02-09 16:59:52[INFO] mariadb의 최신 안정 버전을 감지하는 중 (메이저 10.11 필터
링)...
2026-02-09 16:59:52[INFO] MariaDB 10.11 시리즈 최신 버전 탐지 중...
2026-02-09 16:59:54[INFO] 대상 버전(10.11.16)이 현재(10.11.2)보다 최신입니다. 패키지
업데이트를 권장합니다.
2026-02-09 16:59:54[INFO] mariadb 10.11.16 소스/아티팩트 다운로드 중...
2026-02-09 17:01:12[INFO] mariadb 10.11.16 빌드 및 패키지 생성 중...
2026-02-09 17:01:20[SUCC] 패키지가 성공적으로 생성되었습니다:
/root/dist/packages/mariadb-10.11.16.tar.gz
```

> **버전 자동 감지 (v0.5.7)**
>
> 버전을 지정하지 않으면 현재 설치된 MariaDB의 메이저 시리즈(예: 10.11)를 자동 추출하고, REST API(`downloads.mariadb.org`) 또는 아카이브(`archive.mariadb.org`) 폴백을 통해 최신 패치 버전을 감지합니다.

```
# 특정 버전 지정 (--version 플래그)
tkctl update package mariadb --version 10.11.10
tkctl update package nginx -V 1.24.0

# 전체 서비스 패키지 생성
tkctl update package all
```

참고 사항

- 이 명령어는 **인터넷 연결이 활성화**된 환경(예: 개발 서버, 인터넷 가능 운영 서버)에서 실행해야 합니다.
- 생성된 패키지는 폐쇄망 서버로 복사한 후 `tkctl update` 명령어를 통해 설치할 수 있습니다.

빌드 환경 요구사항

Nginx와 Redis 패키지 생성에는 **소스 코드 컴파일**이 필요하므로 다음 빌드 도구가 시스템에 설치되어 있어야 합니다:

구성 요소	설명	RHEL/Rocky/Alma 패키지	Ubuntu/Debian 패키지
GCC	C/C++ 컴파일러	`gcc` , `gcc-c++`	`build-essential`
Make	빌드 자동화 도구	`make`	`make`
Perl	OpenSSL 빌드에 필요	`perl-core` , `perl-IPC-Cmd`	`perl`
Tar	아카이브 처리	`tar`	`tar`

> **참고**
>
> OpenSearch와 MariaDB는 사전 빌드된 바이너리를 다운로드하므로 컴파일 도구가 필요하지 않습니다.

의존성 자동 설치

`tkctl` 은 패키징 시작 전에 빌드 의존성을 자동으로 확인하고, 누락된 패키지가 있을 경우 **자동으로 설치를 시도**합니다.

동작 방식:

1. OS 배포판(RHEL, Rocky, Ubuntu 등)을 자동 감지합니다.
2. 패키지 관리자(dnf, yum, apt)를 확인합니다.
3. root 권한으로 실행 중일 경우, 누락된 패키지를 자동 설치합니다.
4. 자동 설치가 불가능할 경우, 수동 설치 명령어를 안내합니다.

출력 예시 (의존성 자동 설치):

```
2025-12-23 17:00:00[INFO] 빌드 의존성 확인 중...
2025-12-23 17:00:00[WARN] 누락된 빌드 의존성: gcc, perl
2025-12-23 17:00:00[INFO] 누락된 패키지를 자동으로 설치합니다...
2025-12-23 17:00:00[INFO] dnf 명령어로 패키지를 설치합니다: gcc, gcc-c++, perl-
core, perl-IPC-Cmd
...
2025-12-23 17:00:15[SUCC] 빌드 의존성 설치가 완료되었습니다.
```

수동 설치 (자동 설치 실패 시):

```
# RHEL 9 / Rocky Linux 9 / AlmaLinux 9
sudo dnf install -y gcc gcc-c++ make perl-core perl-IPC-Cmd
```

```
# Ubuntu / Debian
sudo apt-get install -y build-essential make perl
```

3.2 서비스 관리 (service)

명령어	기능	주요 플래그 (옵션)
`check`	전체 서비스 상태 점검	없음
`start`	서비스 시작	`[service]` : 서비스명 (생략 시 all)
`stop`	서비스 중지	`[service]` : 서비스명 (생략 시 all)
`restart`	서비스 재시작	`[service]` : 서비스명 (생략 시 all)
`jvm`	JVM 메모리 최적화	`--set` : 설정값 (auto, S, M, L, 4G 등) `--dry-run` : 시뮬레이션 `--no-restart` : 재시작 건너뛰기
`loglevel`	로그 레벨 조회/설정	`[service]` : 대상 서비스 `--set` : 레벨 설정 (INFO, DEBUG 등)
`logrotate`	로그 로테이션 관리	`--set` : 권장 설정 자동 적용

TACHYON 및 미들웨어 서비스의 상태를 확인하고 제어합니다. 각 서브커맨드(`check` , `start` , `stop` , `restart` , `loglevel` , `jvm` , `logrotate`)는 독립적인 도움말(`-h`)을 지원합니다.

사용법

```
tkctl service [action] [service_name] [flags]
```

사용 가능한 액션

- `check` : 모든 서비스의 상태 확인
- `start <service>` : 서비스 시작
- `stop <service>` : 서비스 중지
- `restart <service>` : 서비스 재시작
- `jvm` : JVM 힙 메모리 설정 관리 및 자동 최적화
- `loglevel` : 로그 레벨 조회/설정
- `logrotate` : 로그 로테이션 관리

서비스 목록

미들웨어 서비스:

- `mariadb` - MariaDB 데이터베이스
- `redis` - Redis 캐시
- `nginx` - NGINX 웹 서버
- `zookeeper` - Zookeeper
- `kafka` - Kafka 메시지 브로커
- `opensearch` - OpenSearch 검색 엔진
- `opensearch-dashboards` - OpenSearch 대시보드
- `logstash-kafka-os` - Logstash

TACHYON 서비스:

- `TACHYON-Api1` - API 서버
- `TACHYON-Auth1` - 인증 서버
- `TACHYON-Manager1` - 관리 서버
- `TACHYON-Stat1` - 통계 서버
- `TACHYON-Report1` - 리포트 서버
- `TACHYON-Batch1` - 배치 서버
- `TACHYON-Watchdog1` - 감시 서버

특수 옵션:

- `all` - 모든 서비스 (순차적 시작/중지)

예제

```
# 모든 서비스 상태 확인
tkctl service check

# 특정 서비스 시작
tkctl service start mariadb
tkctl service start TACHYON-Api1

# 모든 서비스 시작 (순차적 실행 워크플로우)
tkctl service start all

/* 실행 순서 상세 */

1. Level 1: 기본 인프라 (Infra)
   - zookeeper
   - redis
   - mariadb
   - opensearch
   - opensearch-dashboards
   (3초 대기)
```

```
2. Level 2: 메시징 (Kafka)
   - kafka
   (5초 대기 & 연결 상태 검증)

3. Level 3: 로그 수집 및 웹 서버
   - logstash-kafka-os
   - nginx

4. Level 4: TACHYON API
   - TACHYON-Api* (발견된 모든 인스턴스: Api1, Api2...)
   (60초 대기 - API 초기화)

5. Level 5: TACHYON 서비스 로직
   - TACHYON-Auth*
   - TACHYON-Manager*
   - TACHYON-Stat*
   - TACHYON-Report*
   - TACHYON-Batch*
   (10초 대기)

6. Level 6: 감시 (Watchdog)
   - TACHYON-Watchdog*


# 특정 서비스 중지
tkctl service stop nginx

# 모든 서비스 중지 (역순 워크플로우)
tkctl service stop all
(Watchdog -> Service Logic -> API -> Logs/Web -> Kafka -> Infra 순서로 안전하게
종료)
```

출력 예시 (service check)

```
2025-12-24 21:02:46 [INFO] ### TACHYON 서비스 상태 점검 ###

--- 서비스 리소스 사용량 ---
------------------------------------------------------------------------
  서비스명                          상태         CPU(%)       메모리
------------------------------------------------------------------------
  mariadb                          Active       0.1          324.78 MB
  redis                            Active       0.1          512.0 KB
  zookeeper                        Active       0.1          123.01 MB
  kafka                            Active       1.0          1.36 GB
```

```
  opensearch                     Active      0.0         4.62 GB
  TACHYON-Api1                   Active      0.0         771.18 MB
  TACHYON-Batch1                 Active      0.2         1.08 GB
  TACHYON-Manager1               Active      0.0         1.21 GB
  ...
--------------------------------------------------------------------------

--- JVM 힙 메모리 설정 ---
---------------------------------------------
  서비스           초기(-Xms)     최대(-Xmx)
---------------------------------------------
  opensearch       4g             4g
  kafka            2G             2G
  ...
---------------------------------------------

  ✓ 모든 JVM 서비스의 메모리 설정이 확인되었습니다.

--- 로그 로테이션 상태 요약 ---
  ✓ 모든 서비스의 로그 로테이션 설정이 올바릅니다.

2025-12-24 21:02:46 [SUCC] 서비스 상태 확인 완료.
```

3.2.1 서비스 로깅 레벨 확인 및 설정 (loglevel)

운영 중인 각 서비스의 로깅 레벨(DEBUG, INFO, WARN, ERROR 등)을 확인하거나 변경합니다. Java 기반의 TACHYON 서비스와 주요 미들웨어(Nginx, Redis 등)의 설정 파일을 분석하여 현재 적용된 레벨을 표시하며, `--set` 옵션을 통해 즉시 변경이 가능합니다.

사용법:

```
# 모든 서비스의 로깅 레벨 확인
tkctl service loglevel

# 특정 서비스의 로깅 레벨 확인
tkctl service loglevel api

# 특정 서비스의 로깅 레벨 변경
tkctl service loglevel api --set DEBUG
tkctl service loglevel auth --set INFO
tkctl service loglevel batch --set WARN
```

지원되는 레벨:

- `DEBUG`, `INFO`, `WARN`, `ERROR`, `TRACE`, `OFF`

진단 대상 파일 및 세부 항목:

- **TACHYON 서비스**: `*.yml_dev` (평문 원본), `logback-spring.xml`, `log4j2.xml`
  - `logging.level.*` 설정을 기준으로 시스템 로깅 레벨을 표시합니다.
- **NGINX**: `nginx/conf/nginx.conf`
  - **Error Log**: `error_log` 지시어에 설정된 레벨(DEBUG, INFO, ERROR 등)을 추출합니다.
  - **Access Log**: NGINX 특성상 레벨 없이 기록 여부만 제어하므로, `access_log off;` 설정 여부를 파싱하여 **ON/OFF**로 표시합니다.
- **Redis**: `redis.conf` (`loglevel` 지시어)
  - Redis 표준 레벨(DEBUG, VERBOSE, NOTICE, WARNING)을 추출합니다.

로깅 레벨 변경 로직 (Java 서비스 - v0.4.22 개선):

단계	동작	설명
1	`*.yml_dev` 파일 수정	`logging.level.*` 아래 모든 키 일괄 변경 (root, com.tachyon 등)
2	`*.yml` 파일 복사	Spring Boot가 사용하는 실제 설정 파일에 반영
3	`*_ORIGIN` 파일 동기화	기본 설정 파일도 함께 업데이트 (재설치 후에도 설정 유지)
4	서비스 재시작	`systemctl restart TACHYON-[Svc]1.service`

> **참고**
>
> `logging.level.root` 키가 없는 서비스(batch, watchdog 등)도 자동으로 키가 생성됩니다.

3.2.2 서비스 로그 로테이션 상태 확인 및 설정 (logrotate)

TACHYON 및 미들웨어 서비스의 로그 로테이션 설정 상태를 확인하고, 누락된 설정을 자동으로 구성합니다.

사용법:

```
# 전체 서비스 로그 로테이션 상태 확인
tkctl service logrotate

# 특정 서비스 로그 로테이션 상태 확인
tkctl service logrotate nginx
```

```
# 누락된 서비스에 대해 로그 로테이션 자동 설정
tkctl service logrotate --set

# 특정 서비스 로그 로테이션 설정
tkctl service logrotate redis --set
```

지원 서비스:

- **미들웨어**: `nginx`, `redis`, `opensearch` (메인), `opensearch-gc` (GC 로그), `kafka`, `logstash`, `mariadb`
- **TACHYON**: `api`, `auth`, `manager`, `stat`, `batch`, `report`, `watchdog` (Java/Log4j2 기반)
- **관리 도구**: `tkadmin`, `tkctl`

로테이션 유형:

- `logrotate`: 시스템 logrotate 데몬 사용 (`/etc/logrotate.d/`)
- `log4j2` : Java Log4j2 내부 롤링 정책 사용
- `internal` : 서비스 자체 내장 로테이션 메커니즘 사용

3.2.3 서비스 JVM 메모리 최적화 및 관리 (jvm)

TACHYON 및 관련 미들웨어의 JVM 힙 메모리(-Xms, -Xmx) 설정을 조회, 변경하거나 시스템 사양에 맞춰 자동 최적화합니다.

사용법:

```
# 전체 JVM 서비스 메모리 설정 및 시스템 대비 할당량 점검
tkctl service jvm

# 특정 서비스의 메모리 설정 변경
tkctl service jvm api --set 2G

# 공식 권장 프리셋 적용 (XS, S, M, L)
tkctl service jvm --set S

# 시스템 사양 및 에이전트 수 기반 자동 최적화 (권장)
tkctl service jvm --set auto

# 에이전트 수를 명시하여 자동 최적화 수행
tkctl service jvm --set auto --agents 5000

# 실제 반영 없이 변경될 결과만 미리보기 (Dry-run)
tkctl service jvm --set auto --dry-run
```

```
# 설정 변경 후 서비스 자동 재시작 건너뛰기
tkctl service jvm --set auto --no-restart
```

자동 최적화 알고리즘 (auto):

- 서버의 **물리 메모리(RAM)** 크기와 관리 중인 **에이전트 수(Scale)**를 분석하여 서비스별 최적 비중을 계산합니다.
- OpenSearch(20%), Kafka(10%), API(5%) 등 각 컴포넌트의 중요도와 실제 부하 분담률을 반영합니다.
- 서비스별 최소 Heap(256MB~1GB) 보장 및 최대 상한(16GB) 정책이 적용되어 시스템 안정성을 확보합니다.

프리셋 유형:

- `XS` (Extra Small): 8GB 이하 저사양 테스트 환경용 (초경량 설정)
- `S` (Small): 16GB 권장 사양용
- `M` (Medium): 32GB 권장 사양용
- `L` (Large): 64GB 이상 고사양용

안전 장치:

- 모든 설정 변경 전 원본 파일에 대한 **백업(`.bak`)**을 자동 생성합니다.
- **리소스 경고**: 총 JVM 할당량이 물리 메모리의 **80%**를 초과할 경우, 운영 안정성을 위해 경고 메시지를 출력하고 수동 검토를 권장합니다.
- 서비스 재시작 시 `tkadmin`에 사전 보고하여 Watchdog 오탐을 방지합니다.

3.3 모니터링 및 분석 (monitoring)

3.3.1 용량 확인 (size)

명령어	기능	주요 플래그 (옵션)
`(기본)`	MariaDB 상위 N개 테이블 용량 조회	`[count]` : 출력 개수 (기본 20)
`tables`	MariaDB 테이블 용량 상세 조회	`[count]` : 출력 개수 `--by-month` : 월별 데이터 집계 보기
`opensearch`	OpenSearch 인덱스 용량 조회	`--by-month` : 월별 데이터 집계 보기

MariaDB 테이블 및 OpenSearch 인덱스의 용량을 확인합니다. `size tables`와 `size opensearch` 두 가지 하위 커맨드를 제공하며, 각각 상세한 정렬 및 집계 옵션을 지원합니다.

사용법

```
# 기본 사용 (MariaDB 상위 테이블 n개 확인)
tkctl size [n]

# MariaDB 테이블 상세 확인
tkctl size tables [n] [flags]

# OpenSearch 인덱스 상세 확인
tkctl size opensearch [flags]
```

MariaDB 테이블 크기

```
# 전체 테이블
tkctl size tables

# 상위 20개 테이블
tkctl size tables 20

# 하위 호환성
tkctl size 20

# LOG 테이블 월별 데이터 집계
tkctl size tables --by-month
tkctl size db -m
```

출력 예시:

```
2025-12-24 21:03:53 [INFO] ### MariaDB 테이블 용량 확인 (00133FKFKG) ###
2025-12-24 21:03:53 [INFO] 🔎 상위 20 개 테이블을 표시합니다.

------------------------------------------------------------------------
 테이블명                                    크기          행 수
------------------------------------------------------------------------
 TPU_ISTAT_MEDIA_EVENT_LOG                   19.3 MB       14,005
 TPU_INFO_DEPLOYING_DETAIL                   960.0 KB      17
 TPU_INFO_ENDPOINT_ISSUE_HISTORY             960.0 KB      15
 TPU_ISTAT_SYS_LOG                           800.0 KB      213
 TPU_INFO_SCHEDULE                           720.0 KB      15
 TPU_ISTAT_UMS_HISTORY_SUM                   720.0 KB      15
 TPU_ISTAT_MEDIA_BLOCK_SUM                   720.0 KB      15
 TPU_ISTAT_EVENT_LOG                         704.0 KB      306
 TPU_ISTAT_SYSTEM_SELF_TEST_LOG              592.0 KB      441
 TPU_ISTAT_AGENT_INTEGRITY_LOG               528.0 KB      95
```

```
------------------------------------------------------------------------
2025-12-24 21:03:53 [SUCC] 테이블 용량 조회 완료.
```

출력 예시 (월별 집계):

```
2026-01-13 23:21:10 [INFO] *_LOG 테이블 월별 데이터 집계 중...
2026-01-13 23:21:10 [INFO] 발견된 LOG 테이블: 44개

### TPU_ISTAT_MEDIA_EVENT_LOG (날짜 컬럼: client_dt)
---------------------------------
 MONTH            ROWS
---------------------------------
 2026-01          53,922
 2025-12          30,653
 2025-11          11,881
---------------------------------

### TPU_ISTAT_EVENT_LOG (날짜 컬럼: client_dt)
---------------------------------
 MONTH            ROWS
---------------------------------
 2026-01          423
 2025-12          595
 2025-11          164
---------------------------------
```

주요 기능:

- MySQL 스타일 테이블 형식 출력
- 자동 단위 변환 (B/KB/MB/GB)
- 천 단위 콤마 표시 (14,005)
- 크기 기준 내림차순 정렬
- 동적 컬럼 너비 조정
- **월별 집계 옵션** (`-m` , `--by-month`): `*_LOG` 테이블의 날짜 컬럼을 자동 감지하여 월별 데이터 집계

OpenSearch 인덱스 크기

```
# 전체 인덱스 목록
tkctl size opensearch

# 월별 집계 (날짜 패턴 인덱스)
tkctl size opensearch -m
tkctl size opensearch --by-month
```

출력 예시 (일반):

```
2025-12-08 15:17:47[INFO] OpenSearch Disk Usage Check...

------------------------------------------------------------------------
 인덱스 (INDEX)                                  크기          문서 수
------------------------------------------------------------------------
 00133fkfkg_2025_11                               7.2 MB        12,486
 00133fkfkg_2025_12                               3.4 MB        4,713
 top_queries-2025.12.07-15367                     507.1 KB      199
 top_queries-2025.12.05-15365                     481.5 KB      293
 ...
------------------------------------------------------------------------
2025-12-08 15:17:47[SUCC] Check Complete.
```

출력 예시 (월별 집계):

```
------------------------------------------------------------------------
 월 (MONTH)                                      크기          문서 수
------------------------------------------------------------------------
 00133fkfkg_2025-11                               7.2 MB        12,486
 00133fkfkg_2025-12                               3.4 MB        4,713
 2025-12                                          2.1 MB        1,795
------------------------------------------------------------------------
```

주요 기능:

- MariaDB와 동일한 MySQL 스타일 테이블 형식
- 자동 단위 변환 및 천 단위 콤마
- 크기 기준 내림차순 정렬
- 월별 집계 옵션 (`-m` , `--by-month`)

3.3.2 로그 분석 (analyze)

명령어	기능	주요 플래그 (옵션)
`recommend`	데이터 기반 분석 추천	`--month` : 분석 대상 월 (YYYY-MM)
`media-top`	매체제어 프로세스/경로 Top N	`--month` : 대상 월 `--limit` : 출력 개수 (기본 20) `--output` : CSV 파일로 저장
`agent-anomaly`	과도한 로깅 에이전트 탐지	`--month` : 대상 월 `--threshold` : 평균 대비 임계 배수 (기본 5.0)
`process-top`	프로세스별 로그 집계	`--month` : 대상 월 `--limit` : 출력 개수
`db-trend`	DB 데이터 증가 추세 분석	`--output` : CSV 파일로 저장

MariaDB 및 OpenSearch의 로그 데이터를 분석하여 원인 진단 및 이상 탐지를 수행합니다. 운영 환경에서 로그 급증의 원인을 파악하고, 분석이 필요한 항목을 자동으로 추천받을 수 있습니다.

사용법

```
tkctl analyze [subcommand] [flags]
```

서브커맨드

서브커맨드	설명	데이터 소스
`media-top`	매체제어 프로세스+경로별 Top N 분석	OpenSearch
`agent-anomaly`	과도한 로깅 에이전트 탐지	OpenSearch
`process-top`	프로세스별 로그 집계 Top N	OpenSearch
`recommend`	데이터 기반 분석 추천	MariaDB + OpenSearch
`db-trend`	DB 데이터 증가 추세 분석	MariaDB

공통 플래그

플래그	축약	설명	기본값
`--month`	`-m`	분석 대상 월 (YYYY-MM)	현재 월
`--limit`	`-n`	결과 개수 제한	20
`--timeout`		쿼리 타임아웃 (초)	30
`--output`	`-o`	결과 저장 파일 경로 (.csv, .txt)	(화면 출력)

분석 추천 (recommend)

데이터 상태를 자동 진단하여 어떤 분석을 수행해야 할지 권장 명령을 추천합니다. 운영자가 "무엇을 분석해야 하나요?"라는 질문에 자동으로 답변을 제공합니다.

사용법:

```
# 기본 실행 (현재 월 기준)
tkctl analyze recommend

# 특정 월 기준 분석
tkctl analyze recommend --month 2026-01
```

출력 예시:

```
📊 데이터 분석 권장 리포트
═══════════════════════════════════════════════════════════════
  기준 월: 2026-01

═══════════════════════════════════════════════════════════════
📈 [1] 매체제어 로그
──────────────────────────────────────────────────────────────
   테이블: TPU_ISTAT_MEDIA_EVENT_LOG
   현재 월: 55941건  |  전월: 28970건  |  증가율: +93.1%

   ⚠️ 전월 대비 93% 증가 – 원인 분석 권장

   → 권장 명령:
```

```
      tkctl analyze media-top --month 2026-01
═══════════════════════════════════════════════════════════════════════
═══
📈 [2] 이벤트 로그
───────────────────────────────────────────────────────────────────────
───
   테이블: TPU_ISTAT_EVENT_LOG
   현재 월: 503건   |   전월: 566건   |   증가율: -11.1%

   ✅ 정상 범위 내 변동
═══════════════════════════════════════════════════════════════════════
═══

🔍 요약: 1개 항목 분석 권장, 2개 정상
```

판정 기준:

- 🔴 **폭증 (Priority 1)**: 전월 대비 100% 이상 증가 → 즉시 분석 필요
- ⚠️ **급증 (Priority 2)**: 전월 대비 50~100% 증가 → 원인 분석 권장
- ✅ **정상 (Priority 3)**: 전월 대비 50% 미만 변동

매체제어 로그 분석 (media-top)

매체제어 이벤트 로그에서 프로세스+경로 조합별 발생 빈도 Top N을 분석합니다. 어떤 프로세스가 어떤 경로에 가장 많이 접근했는지 파악할 수 있습니다.

사용법:

```
# 현재 월 상위 10개
tkctl analyze media-top --limit 10

# 특정 월 분석
tkctl analyze media-top --month 2026-01

# 결과를 CSV 파일로 저장
tkctl analyze media-top --month 2026-01 --output media_report.csv
```

출력 예시:

```
2026-01-15 17:51:59 [INFO] 매체제어 로그 분석 (media-top)
2026-01-15 17:51:59 [INFO]   기간: 2026-01-01 ~ 2026-02-01
2026-01-15 17:51:59 [INFO]   소스: opensearch
```

```
2026-01-15 17:51:59 [INFO]   제한: 5건

------------------------------------------------------------------------------
------
 RANK    PROCESS                                          PATH
COUNT
------------------------------------------------------------------------------
------
 1       C:\Windows\Explorer.EXE
42,349
 2       C:\Windows\system32\svchost.exe
5,966
 3       C:\Program Files\...\MsMpEng.exe                 E:\
2,611
 4
1,019
 5       C:\Windows\System32\powershell.exe               \RaiDrive\MyDS\
811
------------------------------------------------------------------------------
------
  총 5건 출력
```

에이전트 이상 탐지 (agent-anomaly)

평균 대비 과도하게 로깅하는 에이전트를 탐지합니다. 특정 PC에서 비정상적인 이벤트가 다량 발생하는 경우를 식별할 수 있습니다.

사용법:

```
# 기본 실행 (평균 대비 5배 이상 탐지)
tkctl analyze agent-anomaly

# 임계치 조정 (평균 대비 10배 이상만 탐지)
tkctl analyze agent-anomaly --threshold 10

# 특정 월 분석
tkctl analyze agent-anomaly --month 2026-01
```

출력 예시:

```
2026-01-15 17:52:01 [INFO] 에이전트 이상 로깅 탐지 (agent-anomaly)
2026-01-15 17:52:01 [INFO]   기간: 2026-01-01 ~ 2026-02-01
```

```
2026-01-15 17:52:01 [INFO]   소스: opensearch
2026-01-15 17:52:01 [INFO]   임계치: 평균 x5.0 이상

------------------------------------------
  📊 에이전트 로깅 통계
------------------------------------------
  총 에이전트 수: 150
  평균 로깅 건수: 352.1
  표준 편차: 1,245.3
  이상 에이전트: 3개 (임계치: 평균 x5.0 이상)
------------------------------------------

------------------------------------------------------------------------
 RANK    AGENT_ID(PUID)                    COUNT         RATIO
------------------------------------------------------------------------
 1       a1b2c3d4-e5f6-7890-...            15,234        43.3x 🔴
 2       b2c3d4e5-f6a7-8901-...            8,456         24.0x 🔴
 3       c3d4e5f6-a7b8-9012-...            2,108         6.0x ⚠️
------------------------------------------------------------------------
```

판정 기준:

- 🔴 **심각**: 평균 대비 10배 이상 로깅
- ⚠️ **주의**: 평균 대비 5~10배 로깅

프로세스 집계 분석 (process-top)

프로세스별 로그 발생 건수를 집계하여 상위 N개를 표시합니다. 어떤 프로세스가 가장 많은 이벤트를 발생시키는지 파악할 수 있습니다.

사용법:

```
# 상위 20개 프로세스
tkctl analyze process-top

# 상위 5개만 표시
tkctl analyze process-top --limit 5

# 특정 월 분석 및 파일 저장
tkctl analyze process-top --month 2026-01 --output process.csv
```

출력 예시:

```
2026-01-15 17:52:03 [INFO] 프로세스별 로그 집계 (process-top)
2026-01-15 17:52:03 [INFO]   기간: 2026-01-01 ~ 2026-02-01
2026-01-15 17:52:03 [INFO]   소스: opensearch
2026-01-15 17:52:03 [INFO]   제한: 5건

------------------------------------------------------------------------------
-----------------------------------
 RANK    PROCESS
COUNT         RATIO
------------------------------------------------------------------------------
-----------------------------------
 1       C:\Windows\Explorer.EXE
42,349         80.3%
 2       C:\Windows\system32\svchost.exe
5,966          11.3%
 3       C:\ProgramData\Microsoft\Windows Defender\...\MsMpEng.exe
2,611          4.9%
 4
1,019          1.9%
 5       C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
811            1.5%
------------------------------------------------------------------------------
-----------------------------------
  총 5건 출력 (전체 집계 합계: 52,756건)
```

DB 증가 추세 분석 (db-trend)

DB 및 로그 데이터의 증가 추세를 분석하여 스토리지 용량 계획(Capacity Planning)을 지원합니다. **Fast Mode(기본)**와 **Precise Mode(정밀)** 두 가지 모드를 제공합니다.

사용법:

```
# 기본 실행 (Fast Mode: information_schema 기반 즉시 분석)
tkctl analyze db-trend

# 정밀 분석 (Precise Mode: COUNT(*) 쿼리 기반 상세 분석, 시간 소요됨)
tkctl analyze db-trend --precise
tkctl analyze db-trend -P

# 결과를 CSV 파일로 저장
tkctl analyze db-trend --output trend.csv
```

---

1. Fast Mode (기본)

`information_schema` 메타데이터를 사용하여 **즉시(O(1))** 결과를 반환합니다. 운영 중인 DB에 부하를 주지 않으며, 전체 시스템(MariaDB + OpenSearch + Disk)의 용량 현황을 통합 리포트로 제공합니다.

출력 예시:

```
2026-01-19 22:36:07 [INFO] DB 데이터 증가 추세 분석 (db-trend)
2026-01-19 22:36:07 [INFO] ⚡ 빠른 분석 모드 (information_schema 기반)

═══════════════════════════════════════════════════════════════════════
═══
📊 시스템 용량 현황 리포트 (Fast Mode)
───────────────────────────────────────────────────────────────────────
──
 📁 MariaDB 크기  : 138.29 MB (디스크 점유율 0.1%)
 📁 OpenSearch    : 57.80 MB
 📁 설치 디렉토리 : 71.07 GB (/usr/local/TACHYON/TTS40)
───────────────────────────────────────────────────────────────────────
──
 💾 디스크 가용량  : 27.86 GB (전체 98.93 GB 중 사용 71.8%)
───────────────────────────────────────────────────────────────────────
──
 📈 월간 예상 증가 : 16.30 MB / 월 (MariaDB: 10.52 MB + OpenSearch: 5.78 MB)
 📈 년간 예상 증가 : 195.59 MB / 년
 📅 디스크 포화 예측: 🟢 약 1750.6개월 후 포화 예상 (2171-11-19)
 📋 분석 테이블 수 : 50개
═══════════════════════════════════════════════════════════════════════
═══

📋 테이블 현황 (전체 50개 중 상위 10개)
-----------------------------------------------------------------------
---------------
 #     TABLE                                ROWS (추정)         SIZE
AVG_ROW
-----------------------------------------------------------------------
---------------
 1     TPU_ISTAT_MEDIA_EVENT_LOG            76,989               91.19 MB
1.21 KB
 2     TPU_ISTAT_EVENT_LOG                  1,079                1.02 MB
986 B
 3     TPU_ISTAT_AGENT_INTEGRITY_LOG        528                  592.00 KB
1.12 KB
 ...
 10    TPU_ISTAT_SUSB_INTEGRITY_LOG         16                   256.00 KB
16.00 KB
```

```
 ...    외 40개 테이블                          2,182                9.81 MB
-
------------------------------------------------------------------------------
---------------
```

주요 기능:

1. **통합 용량 리포트**: MariaDB, OpenSearch, 설치 디렉토리, 디스크 가용량을 한눈에 파악
2. **Capacity Planning**: 현재 데이터 크기를 기반으로 월간/년간 증가량 추정 및 디스크 포화 시점 예측
3. **대용량 테이블 Top 50**: 용량 순 상위 10개 테이블 상세 표시 및 나머지 요약

---

2. Precise Mode (정밀 분석)

`--precise` 또는 `-P` 옵션 사용 시 동작합니다. 실제 `COUNT(*)` 쿼리를 수행하여 정확한 행 수 및 기간별 (1주, 1개월, 1분기, 1년) 증가 추세를 분석합니다. 대용량 DB의 경우 시간이 소요될 수 있습니다.

출력 예시:

```
2026-01-17 20:21:17 [INFO] 🔍 정밀 분석 모드 (COUNT 쿼리 사용, 시간 소요)
...
📊 DB 용량 계획 리포트 (Capacity Planning)
...
------------------------------------------------------------------------------
---------------------------------------
 TABLE                  PERIOD           START(ACC)    END(ACC)      INCREASE
RATE        TREND
------------------------------------------------------------------------------
---------------------------------------
 STAT_MEDIA_EVENT_LOG   최근 1주          71,499        96,792        25,293
+35.4%       🟠 급증 (주의)
 STAT_MEDIA_EVENT_LOG   최근 1개월        16,268        96,792        80,524
+495.0%      🔴 폭증 (위험)
 ...
```

주요 기능:

1. **정확한 데이터**: InnoDB 추정치가 아닌 실제 Row Count 기반
2. **기간별 추세**: 1주/1개월/1분기/1년 전 데이터와 비교하여 정확한 증가율(%) 산출
3. **위험도 진단**: 증가율에 따른 트렌드 진단 (🔴 폭증, 🟠 급증 등)

3.4 백업 및 복원 (backup/restore)

MariaDB 및 OpenSearch의 데이터를 백업하고 복원합니다. Cobra 네이티브 서브커맨드 구조로 구현되어 자동완성 및 도움말이 자연스럽게 지원됩니다.

사용법

```
# 백업
tkctl backup [mariadb|opensearch] [flags]

# 복원
tkctl restore [mariadb|opensearch] [args]
```

3.4.1 MariaDB 백업 (backup mariadb)

mariabackup을 사용하여 MariaDB 데이터베이스를 백업합니다.

사용법:

```
# 전체 백업 (기본)
tkctl backup mariadb

# 백업 유형 지정
tkctl backup mariadb --type full    # 전체 물리적 백업
tkctl backup mariadb --type incr    # 증분 백업
tkctl backup mariadb --type config  # 설정 데이터만 (로그 제외)
tkctl backup mariadb --type log     # 로그 포함 전체 백업

# 옵션 조합
tkctl backup mariadb --type full --retention 30 --compress
```

플래그:

플래그	축약	설명	기본값
`--type`	`-t`	백업 유형 (config, log, full, incr)	full
`--retention`	`-r`	보관 일수	30
`--compress`	`-c`	qpress 압축 사용	true
`--incremental-basedir`		증분 백업 기준 디렉토리	(자동 탐지)

출력 예시:

```
📦 MariaDB 백업
[████████████▒▒▒▒▒▒▒▒] 60% | 백업실행    | 1m30s

✅ MariaDB Backup (full) completed successfully.
  Path: /opt/backup/mariadb/full_20260122_103000
  Size: 256.5 MB
  Duration: 2m30s
```

> 💡 **프로그레스바**: v0.4.23부터 TTY 환경에서 시각적 프로그레스바가 표시됩니다. 파이프(`|`) 또는 리다이렉트(`>`)로 출력하면 기존 텍스트 형식으로 표시됩니다.

3.4.2 OpenSearch 백업 (backup opensearch)

OpenSearch 스냅샷 저장소에 인덱스 스냅샷을 생성합니다.

사용법:

```
# 스냅샷 저장소 초기화 (최초 1회)
tkctl backup opensearch --init-repo

# 스냅샷 생성 (자동 이름)
tkctl backup opensearch

# 스냅샷 이름 지정
tkctl backup opensearch snap_20260116
```

플래그:

플래그	설명
`--init-repo`	스냅샷 저장소 초기화 (fs 타입)

출력 예시:

```
2026-01-16 10:35:00 [INFO] OpenSearch 스냅샷을 생성합니다...
2026-01-16 10:35:00 [INFO] 스냅샷 이름: snap_20260116_103500
2026-01-16 10:35:10 [SUCC] 스냅샷이 성공적으로 생성되었습니다.
```

3.4.3 MariaDB 복원 (restore mariadb)

mariabackup 백업으로부터 MariaDB 데이터베이스를 복원합니다.

사용법:

```
# 전체 백업에서 복원
tkctl restore mariadb /backup/full_20260116

# 전체 + 증분 백업에서 복원
tkctl restore mariadb /backup/full_20260116 /backup/incr_20260117
/backup/incr_20260118
```

출력 예시:

```
2026-01-16 11:00:00 [INFO] MariaDB 복원을 시작합니다...
2026-01-16 11:00:00 [INFO] 전체 백업: /backup/full_20260116
2026-01-16 11:00:00 [WARN] MariaDB 서비스가 실행 중입니다. 복원 전 중지하세요.
2026-01-16 11:00:00 [INFO] 명령어: systemctl stop mariadb
```

> ⚠️ **주의**: MariaDB 복원 전 반드시 서비스를 중지해야 합니다.

3.4.4 OpenSearch 복원 (restore opensearch)

저장된 스냅샷으로부터 OpenSearch 인덱스를 복원합니다.

사용법:

```
tkctl restore opensearch snap_20260116
```

출력 예시:

```
2026-01-16 11:10:00 [INFO] OpenSearch 스냅샷 복원을 시작합니다...
2026-01-16 11:10:00 [INFO] 스냅샷: snap_20260116
2026-01-16 11:10:15 [SUCC] 스냅샷 복원이 완료되었습니다.
```

3.5 환경 구성 (env)

명령어	기능	주요 플래그 (옵션)
`(기본)`	전체 환경 설정 상태 조회	`--set` : 각 항목의 권장값 자동 적용
`selinux`	SELinux 설정	`[mode]` : enforcing/permissive/disabled `--set` : 설정 적용
`ulimit`	시스템 리미트 설정	`[value]` : Open Files 한도값 (예: 65535) `--set` : 설정 적용
`firewall`	방화벽 포트 설정	`[port]` : 개방할 포트 `--set` : 설정 적용 (SSH, 443 등 기본값 포함)
`ssh`	SSH 포트 설정	`[port]` : 변경할 포트 번호 `--set` : 서비스 재시작 및 방화벽 연동
`web`	웹 포트 설정	`[port]` : 변경할 포트 번호 `--set` : 서비스 재시작 및 방화벽 연동
`db`	DB 포트 설정	`[port]` : 변경할 포트 번호 `--set` : 설정 적용 `--yes` : 서비스 재시작 자동 승인
`locale`	시스템 Locale 설정	`[lang]` : 언어코드 (ko/en) `--set` : 설정 적용

Tachyon 솔루션 구동에 최적화된 OS 환경(SELinux, Ulimit, Firewall, SSH)을 진단하고 설정합니다. `tkctl env [subcommand]` 형식을 사용하여 각 항목별로 상세 제어가 가능하며, 인자 없이 `tkctl env` 실행 시 모든 항목에 대한 요약 점검을 수행합니다.

`--set` (또는 `-s`) 옵션은 설정을 실제로 적용할 때 사용하며, 인자 생략 시 각 환경에 맞는 **TACHYON 권장값**을 자동으로 선택합니다.

3.5.1 SELinux 설정 (selinux)

SELinux의 현재 실행 모드(런타임)와 영구 설정을 확인하고 최적화합니다.

사용법:

```
# 상태 확인
tkctl env selinux

# 권장값(disabled)으로 자동 설정
tkctl env selinux --set

# 특정 모드로 설정
tkctl env selinux disabled --set
tkctl env selinux permissive --set
```

특징:

- 이미 대상 모드로 설정되어 있는 경우 중복 작업을 생략합니다.
- `disabled` 모드 설정 시 영구 설정을 변경하며, 재부팅 전까지 즉시 효과를 위해 런타임에서 `permissive` 를 적용합니다.

출력 예시

```
2025-12-24 21:03:14 [INFO] ### SELinux 구성 ###
-------------------------------------------
 항목                    상태
-------------------------------------------
 런타임                 Disabled
 영구 설정              disabled
-------------------------------------------
2025-12-24 21:03:14 [SUCC] 환경 확인 완료.
```

3.5.2 리미트 설정 (ulimit)

시스템 전체(`/etc/security/limits.conf`) 및 각 서비스별 `LimitNOFILE` 설정 상태를 진단합니다.

사용법:

```
# 리미트 상태 점검
tkctl env ulimit
```

```
# 권장값(65535)으로 시스템 리미트 자동 설정
tkctl env ulimit --set

# 특정 값으로 시스템 리미트 설정
tkctl env ulimit 99999 --set
```

주의

시스템 리미트 설정 적용을 위해 **로그아웃 후 다시 로그인**하거나 시스템 재시작이 필요합니다.

출력 예시

```
2025-12-24 21:03:22 [INFO] ### 리미트(ulimit) 구성 및 서비스 진단 ###

 [ 시스템 전체 설정 (/etc/security/limits.conf) ]
------------------------------------------------------------
 대상 (Domain)          유형         항목              값
------------------------------------------------------------
 *                      soft         nofile            65535
 *                      hard         nofile            65535
 root                   soft         nofile            65535
 root                   hard         nofile            65535
------------------------------------------------------------

 [ 서비스별 개별 설정 (systemd LimitNOFILE) ]
------------------------------------------------------------------------
 서비스명                          리미트 (NOFILE)       상태
------------------------------------------------------------------------
 mariadb                           32768                 OK
 redis                             524288                OK
 ...
------------------------------------------------------------------------
2025-12-24 21:03:22 [SUCC] 환경 확인 완료.
```

3.5.3 방화벽 설정 (firewall)

시스템 방화벽(`firewalld`) 상태를 확인하고 TACHYON 구동에 필요한 포트를 개방합니다.

사용법:

```
# 방화벽 상태 및 오픈 포트 진단
tkctl env firewall
```

```
# TACHYON 기본 포트(SSH, 443) 자동 개방
tkctl env firewall --set

# 특정 포트 추가 개방
tkctl env firewall 8080 --set
```

설정 로직:

- **인자 생략**: 현재 시스템의 SSH 포트를 자동 감지하여 443 포트와 함께 개방합니다.
- **포트 지정**: 입력한 포트를 `tcp` 프로토콜로 개방합니다.

출력 예시

```
2025-12-24 21:03:35 [INFO] ### 방화벽(Firewall) 구성 진단 ###

 [ 방화벽 상태 ]
-------------------------------------------
 서비스/존              상태
-------------------------------------------
 firewalld              비활성 (Inactive)
 iptables               비활성 (Inactive)
-------------------------------------------
2025-12-24 21:03:35 [SUCC] 환경 확인 완료.
```

3.5.4 SSH 포트 설정 (ssh)

OpenSSH 서비스의 포트 번호를 점검하고 변경합니다.

사용법:

```
# 현재 SSH 포트 확인
tkctl env ssh

# SSH 포트를 2022로 변경
tkctl env ssh 2022 --set

# SSH 포트를 기본값(22)으로 원복
tkctl env ssh --set
```

⚠️ **자동 구성 안내:**

- **방화벽 연동**: `--set` 을 통해 포트 변경 시, 변경된 포트가 **방화벽에서 자동으로 허용**됩니다. (접속 차단 방지)
- **서비스 자동 재시작**: 설정 변경 후 `sshd` 서비스가 자동으로 재시작되어 즉시 적용됩니다.

출력 예시

```
2025-12-24 21:03:46 [INFO] ### SSH (OpenSSH) 구성 진단 ###

 [ ### SSH (OpenSSH) 구성 진단 ### ]
-------------------------------------------
 항목                    SSH 포트
-------------------------------------------
 OpenSSH                  22
-------------------------------------------

 '--set <포트>'를 사용하여 SSH 포트를 변경하세요
2025-12-24 21:03:46 [SUCC] 환경 확인 완료.
```

3.5.5 웹 포트 설정 (web)

TACHYON 웹 서비스(Nginx HTTPS)의 리스닝 포트를 점검하고 변경합니다.

사용법:

```
# 현재 웹 포트 확인
tkctl env web

# HTTPS 포트를 8443으로 변경
tkctl env web 8443 --set

# 기본값(443)으로 원복
tkctl env web --set
```

v0.4.20+ 확장 옵션:

옵션	설명
`--mode unified/separated`	통합/분리 모드 전환
`--console <port>`	분리 모드 시 관리콘솔 포트 (기본: 8443)
`--backend <port>`	분리 모드 시 백엔드 API 포트 (기본: 443)
`--dry-run`	변경 사항 미리보기 (실제 적용 안 함)
`--restore`	백업에서 nginx.conf 복원

확장 사용 예시:

```
# 분리 모드로 전환
tkctl env web --mode separated --set

# 분리 모드 + 포트 지정
tkctl env web --mode separated --console 9443 --backend 10443 --set

# 미리보기 (변경 없이 확인만)
tkctl env web --mode separated --dry-run

# 백업에서 복원
tkctl env web --restore
```

동작 상세:

- **Nginx 설정 변경**: `/etc/nginx/nginx.conf` 또는 `conf.d/*.conf` 내의 `listen ... ssl` 지시어를 찾아 포트를 변경합니다.
- **방화벽 연동**: 변경된 포트를 방화벽에서 즉시 허용합니다.
- **서비스 제어**: 변경 사항 적용을 위해 Nginx 서비스를 자동으로 재시작합니다.
- **자동 백업**: 설정 변경 전 nginx.conf를 자동으로 백업합니다 (.bak).
- **복원 기능**: `--restore` 옵션으로 마지막 백업에서 설정을 복원할 수 있습니다.

3.5.6 DB 포트 설정 (db)

MariaDB 데이터베이스의 접속 포트를 변경하고, 이를 참조하는 모든 TACHYON 서비스 및 관리 도구의 설정을 일괄 동기화합니다.

사용법:

```
# 현재 DB 포트 확인
tkctl env db

# MariaDB 포트를 13306으로 변경 (자동 승인)
tkctl env db 13306 --set --yes

# 기본값(3306)으로 원복
tkctl env db 3306 --set
```

⚠️ **자동화 범위 및 영향도:** 이 명령어는 단순한 포트 변경 이상의 광범위한 시스템 동기화 작업을 수행합니다.

1. **시스템 포트 변경**: `/etc/my.cnf.d/server.cnf`의 MariaDB 포트를 변경합니다.
2. **서비스 참조 일괄 갱신**:
   - `TACHYON_HOME` 내의 **모든 서비스 설정 파일**(`*.yml_dev`)을 검색하여 `jdbc:mariadb://` 연결 문자열의 포트를 치환합니다.
   - **전역 설정(`app_info.properties`)**의 `db_port` 항목을 함께 갱신하여 시스템 전체의 일관성을 유지합니다.
3. **방화벽 및 재시작**:
   - 변경된 포트를 방화벽에 등록합니다.
   - MariaDB, API, Manager, Watchdog, **tkadmin** 등 영향을 받는 **모든 서비스를 순차적으로 재시작**합니다.
   - 운영 중 서비스 중단(Downtime)이 발생하므로, `--yes` 옵션이 없으면 사용자 승인을 요청합니다.

⚠️ 주의사항 (Nginx Stream)

만약 Nginx가 DB 포트(3306 등)를 프록시하기 위해 `stream` 블록(`nginx.conf` 하단)을 사용 중이라면, 이 부분의 포트는 기본적으로 **자동으로 변경되지 않습니다**.

v0.4.20+ Nginx Stream 자동 동기화:

옵션	설명
`--sync-stream`	nginx stream upstream 포트 자동 업데이트
`--dry-run`	변경 사항 미리보기 (실제 적용 안 함)

```
# DB 포트 변경 + nginx stream 자동 동기화
tkctl env db 3307 --set --sync-stream

# 미리보기 (변경 없이 확인만)
tkctl env db 3307 --dry-run
```

`--sync-stream` 옵션을 사용하면 nginx.conf의 `upstream tachyon_mariadb` 블록 포트가 함께 변경됩니다.

출력 예시:

```
2026-01-08 16:30:00 [INFO] Setting Database port to 13306...
2026-01-08 16:30:00 [INFO] Updating SERVICE configurations (TACHYON_HOME)...
2026-01-08 16:30:01 [INFO]   Updated: api.yml_dev
2026-01-08 16:30:01 [INFO]   Updated: manager.yml_dev
...
2026-01-08 16:30:02 [INFO] Updating nginx stream upstream 'tachyon_mariadb'
to port 13306...
2026-01-08 16:30:02 [SUCC] Updated nginx stream upstream for MariaDB
2026-01-08 16:30:05 [INFO] Restarting TACHYON services...
2026-01-08 16:30:10 [SUCC] Database port updated to 13306
```

3.5.7 DB 포트 감지 및 연결 문제 해결

`tkctl`은 MariaDB 연결 시 다음 순서로 포트를 자동으로 감지합니다:

1. **MariaDB 설정 파일**: `my.cnf` 또는 `server.cnf`의 `[mysqld]` 섹션에 명시된 `port` 값.
2. **Nginx Stream 설정**: `tkctl env db --sync-stream`으로 동기화된 Nginx `upstream tachyon_mariadb` 포트.
3. **기본값**: `3306`.

문제 상황: DB 포트를 변경했는데 `tkctl`가 접속하지 못하는 경우

1. **증상**: `dial tcp 127.0.0.1:3306: connect: connection refused` 에러 발생.
2. **원인**: MariaDB 설정 파일에 포트가 명시되어 있지 않고, Nginx 설정과도 동기화되지 않아 기본값(3306)을 시도했기 때문입니다.
3. **해결 방법**:
   - **방법 A (권장)**: `tkctl env db <PORT> --sync-stream` 명령을 실행하여 Nginx 설정을 업데이트하면, `tkctl`가 이를 감지하여 해당 포트로 접속합니다.
   - **방법 B**: MariaDB 설정 파일(`server.cnf` 등)의 `[mysqld]` 섹션에 `port=<PORT>`를 명시합니다.

> **Tip**
>
> `tkctl env db` 명령은 포트 변경과 동시에 Nginx Stream 설정을 동기화하므로, 안전하게 포트를 변경할 수 있는 가장 좋은 방법입니다.

환경 구성 권장 값 요약

항목	권장 값	명령어 예시
SELinux	`disabled`	`tkctl env selinux --set`
Ulimit	`65535`	`tkctl env ulimit --set`
Firewall	`ssh, 443, 3306`	`tkctl env firewall --set`
SSH Port	`22`	`tkctl env ssh --set`
Web Port	`443`	`tkctl env web --set`
DB Port	`3306`	`tkctl env db --set`
Locale	`ko_KR.UTF-8`	`tkctl env locale ko --set`

3.5.8 Locale 설정 (locale)

시스템 Locale(언어 및 문자셋) 설정을 확인하고 변경합니다.

배경: RockyLinux 9 Non-GUI(Minimal) 모드로 설치 시 `LANG=C` 또는 `LANG=en_US`로 기본 설정되어 제품 콘솔에서 한글이 깨질 수 있습니다. `ko_KR.UTF-8` 또는 `en_US.UTF-8`로 변경하여 해결합니다.

사용법:

```
# 현재 locale 상태 확인
tkctl env locale

# 한글 locale로 설정
tkctl env locale ko --set

# 영어 locale로 설정
tkctl env locale en --set
```

특징:

- 현재 설정이 이미 `*.UTF-8`이면 변경하지 않고 Skip 합니다.
- 언어팩(`glibc-langpack-ko`)이 미설치된 경우 수동 설치 안내를 출력합니다.
- 폐쇄망 환경 지원: 자동 패키지 설치 없이 경고만 출력합니다.

출력 예시:

```
2026-01-21 22:52:34 [INFO] ### Locale (언어/문자셋) 구성 ###
------------------------------------------------
 항목                    상태
------------------------------------------------
 시스템 Locale          LANG=ko_KR.UTF-8
 VC Keymap               kr
 한글 언어팩            ✓ 설치됨
------------------------------------------------
2026-01-21 22:52:34 [SUCC] 환경 확인 완료.
```

언어팩 미설치 시:

```
2026-01-21 22:00:00 [WARN] 언어팩(glibc-langpack-ko)이 설치되어 있지 않습니다.
  ℹ ISO 파일이나 로컬 레포지토리를 통해 설치하세요:
    rpm -ivh glibc-langpack-ko-*.rpm
```

3.5.9 Nginx ACL 설정 (web acl)

Nginx 웹 서버의 접근 제어 목록(ACL)을 관리하여 특정 IP만 관리콘솔에 접근하도록 제한합니다. IP 주소(IPv4) 또는 CIDR 대역(예: `192.168.1.0/24`)을 지원합니다.

> ⚠️ **영향 범위**: ACL은 **관리콘솔(프론트엔드)** 접근만 제어합니다. 백엔드 API 서비스(443 포트)에는 영향을 주지 않으므로, API 호출이나 서비스 연동에는 지장이 없습니다.

사용법:

```
# 1. ACL 정책 확인
tkctl env web acl list

# 2. 관리자 IP 추가 (필수)
tkctl env web acl add 203.0.113.10
tkctl env web acl add 192.168.100.0/24

# 3. ACL 기능 활성화 (모든 접속 차단, 허용된 IP만 접속 가능)
# 주의: 반드시 관리자 IP를 먼저 추가(add)한 후 활성화(enable)해야 합니다.
tkctl env web acl enable
```

```
# [기타]
# 특정 IP 허용 삭제
tkctl env web acl remove 203.0.113.10

# ACL 기능 비활성화 (모든 IP 접속 허용)
tkctl env web acl disable

# 설정 테스트 (문법 검사 및 적용 상태 확인)
tkctl env web acl test
```

주의사항:

- ACL 기능을 활성화(`enable`)하면 기본적으로 **모든 접속이 차단(deny all)**되며, `add` 명령으로 추가한 IP만 접속이 허용됩니다.
- 로컬호스트(`127.0.0.1`)는 내부 통신을 위해 자동으로 허용 규칙에 포함되지 않으므로 필요 시 명시적으로 추가하거나 `location` 구조에 따라 예외 처리됩니다(현재 구현에서는 명시적 추가 권장).
- 설정 변경(`add` , `remove` , `enable` , `disable`) 후에는 반드시 Nginx를 재시작해야 적용됩니다.
  - `tkctl service restart nginx`

설정 파일 동작:

- 기본적으로 `/app/nginx/conf/nginx.conf` (또는 설치 경로) 파일의 `http` 또는 `location` 블록 내에 `allow` / `deny` 지시어를 주입합니다.
- `--config` 옵션으로 대상 설정 파일 경로를 직접 지정할 수 있습니다.

3.6 보안 진단 및 조치 (Security Analysis & Remediation)

`tkctl` 은 OS, Web(Nginx 등), DB(MariaDB) 전 영역에 대한 통합 보안 취약점 진단(`analyze`) 기능과, 발견된 취약점에 대한 자동화된 조치(`fix`) 기능을 제공합니다.

3.6.1 보안 진단 (Analyze Security)

시스템의 현재 설정 상태를 KISA 주요정보통신기반시설 취약점 분석 기준에 따라 점검하고 리포트를 생성합니다.

사용법

```
tkctl analyze security [flags]
```

주요 플래그

플래그	설명	기본값
`--target`, `-t`	점검 대상 지정 (`os`, `web`, `db`, `all`)	`all`
`--format`, `-f`	리포트 포맷 지정 (`json`, `html`)	`json`
`--output`, `-o`	결과 파일 저장 경로	자동 생성
`--quiet`, `-q`	콘솔 출력 억제 (자동화 스크립트용)	`false`

사용 예시

1. 전체 시스템 정밀 진단 후 HTML 리포트 생성

```
tkctl analyze security --format html
# 실행 결과: tkctl-security-report-20260118-143000.html 생성됨
```

생성된 HTML 파일을 브라우저로 열면, 색상 코드로 구분된 직관적인 점검 결과를 확인할 수 있습니다.

2. OS 설정만 빠르게 점검

```
tkctl analyze security -t os
```

3. CI/CD 파이프라인 연동 (JSON 출력)

```
tkctl analyze security --format json --quiet
# 실행 결과: tkctl-security-report-YYYYMMDD-HHMMSS.json 생성됨 (로그 없음)
```

3.6.2 보안 조치 (Fix Security)

진단 결과 발견된 '취약(Vulnerable)' 항목에 대해 자동으로 설정을 수정하여 보안성을 강화합니다.

> ⚠️ **주의**
>
> 보안 조치는 시스템 설정을 직접 변경하므로, 반드시 백업을 수행하고(`--backup=true`), `--dry-run` 모드로 변경 내용을 미리 확인하는 것을 권장합니다.

사용법

```
tkctl fix security [flags]
```

주요 플래그

플래그	설명	기본값
`--id`	조치할 특정 취약점 ID (예: `U-01`)	**필수** (단, `--report`, `--all` 사용 시 제외)
`--report`	분석 리포트(JSON) 기반 자동 조치	`""`
`--all`	지원하는 모든 항목 일괄 조치 (주의 필요)	`false`
`--dry-run`	실제 변경 없이 조치 내용 시뮬레이션	`false`
`--backup`	조치 전 원본 설정 파일 백업 수행	`true`
`--interactive`, `-i`	각 조치 항목별 확인을 요청하는 대화형 모드	`false`

사용 예시

1. 특정 취약점(U-01) 조치 전 시뮬레이션 (Dry-run)

```
tkctl fix security --id U-01 --dry-run
# 실행 결과:
# [INFO] 보안 조치(Remediation) 프로세스 시작
# [INFO] [U-01] 조치 시작: SSH 설정 파일(sshd_config)에서 PermitRootLogin을 'no'로 설정하고 서비스를 재시작합니다.
# [INFO]   [DryRun] 실제 변경은 수행되지 않습니다.
# [SUCC] 모든 조치 프로세스가 완료되었습니다.
```

2. 특정 취약점(U-01) 실제 조치

```
tkctl fix security --id U-01
# 실행 결과:
# [INFO] [U-01] 조치 시작: SSH 설정 파일(sshd_config)에서 PermitRootLogin을 'no'로
```

```
설정하고 서비스를 재시작합니다.
# [SUCC]    백업 완료: /etc/ssh/sshd_config.bak_20260118_194500
# [SUCC]    조치 완료
```

3. 리포트 기반 일괄 조치 (`--report`)

진단 결과(`tkctl analyze --format json`)를 기반으로, 취약한(`VULNERABLE`) 항목만 선별하여 조치합니다.

```
tkctl fix security --report tkctl-security-report-20260118.json
# 실행 결과:
# [INFO] 리포트 분석 중: tkctl-security-report-20260118.json
# 발견된 취약점: 15건 | 자동 조치 가능: 3건
# ... (이후 조치 프로세스 진행)
```

4. 대화형 모드로 조치 (`--interactive`)

각 항목을 조치하기 전 사용자 확인을 요청합니다. 옵션: `y` (Yes), `n` (No/Skip), `a` (All/이후 전부 Yes), `q` (Quit/중단)

```
tkctl fix security --all --interactive
# 실행 결과:
# [INFO] 총 85건의 조치 항목이 선택되었습니다.
# ------------------------------------------------------------
# [INFO] [U-01] 조치 대상: Root 계정 원격 접속 제한 (SSH PermitRootLogin 설정)
# [U-01] 항목을 조치하시겠습니까? [y(Yes)/n(No)/a(All)/q(Quit)]: y
# [INFO]    -> 조치를 시작합니다...
# [SUCC]    백업 완료: /etc/ssh/sshd_config.bak_20260119_143000
# [SUCC]    조치 완료
# ...
```

3.6.3 지원하는 점검/조치 항목

`tkctl` 은 KISA 주요정보통신기반시설 기술적 취약점 분석 평가 상세 가이드를 준수하여 아래 항목들을 진단합니다.

3.6.3.1 Linux OS

> ✅ **전체 73개 항목 조치 지원 (v0.4.17)**

- **자동 조치 (Type A)**: U-01~U-04, U-07~U-12, U-15, U-18~U-23, U-28~U-30, U-32, U-37, U-39
- **수동 가이드 (Type C)**: 나머지 항목 - 상세 명령어 및 가이드 출력

ID	분류	항목명	상세 설명	진단	조치
U-01	계정 관리	Root 원격 접속 제한	SSH/Telnet Root 직접 접속 허용 여부 (`PermitRootLogin no`)	✅	✅
U-02	계정 관리	패스워드 복잡성 설정	영문/숫자/특수문자 조합 및 길이 강제 설정	✅	✅
U-03	계정 관리	계정 잠금 임계값 설정	로그인 실패 시 계정 잠금 (`deny=5`)	✅	✅
U-04	계정 관리	패스워드 파일 보호	패스워드 암호화 및 Shadow 파일 사용 여부	✅	✅
U-05	계정 관리	Root 이외의 UID 0 금지	UID 0을 가진 비인가 계정 존재 여부	✅	✅
U-06	파일 권한	root 홈/패스 권한	`su` 명령어 그룹 제한 및 권한 점검	✅	✅
U-07	계정 관리	패스워드 최소 길이	패스워드 최소 길이(8자 이상) 설정	✅	✅
U-08	계정 관리	패스워드 최대 사용기간	패스워드 최대 사용 기간(90일) 설정	✅	✅
U-09	계정 관리	패스워드 최소 사용기간	패스워드 최소 사용 기간(1일) 설정	✅	✅
U-10	계정 관리	불필요한 계정 제거	불필요한 시스템 계정 존재 여부	✅	✅
U-11	계정 관리	관리자 그룹 사용자 관리	관리자 그룹(root)에 불필요한 계정 등록 여부	✅	✅
U-12	계정 관리	계정이 없는 그룹 관리	소유자 없는 그룹(GID) 존재 여부	✅	✅
U-13	계정 관리	동일한 UID 금지	동일한 UID를 사용하는 중복 계정 존재 여부	✅	✅
U-14	계정 관리	사용자 Shell 점검	로그인이 불필요한 계정에 Shell 부여 여부	✅	✅
U-15	계정 관리	Session Timeout 설정	유휴 세션 타임아웃 설정 (600초)	✅	✅

ID	분류	항목명	상세 설명	진단	조치
U-16	파일 권한	Root 홈/패스 설정	PATH 환경변수에 '.' 포함 여부	✅	✅
U-17	파일 권한	파일/디렉터리 소유자	소유자(User/Group)가 없는 파일 존재 여부	✅	✅
U-18	파일 권한	/etc/passwd 파일	소유자(root) 및 권한(644) 점검	✅	✅
U-19	파일 권한	/etc/shadow 파일	소유자(root) 및 권한(400) 점검	✅	✅
U-20	파일 권한	/etc/hosts 파일	소유자(root) 및 권한(600) 점검	✅	✅
U-21	파일 권한	/etc/(x)inetd.conf	소유자(root) 및 권한(600) 점검	✅	✅
U-22	파일 권한	/etc/syslog.conf	소유자(root) 및 권한(644) 점검	✅	✅
U-23	파일 권한	/etc/services 파일	소유자(root) 및 권한(644) 점검	✅	✅
U-24	파일 권한	SUID/SGID 설정	주요 파일의 SUID/SGID 설정 여부	✅	✅
U-25	파일 권한	사용자 환경 파일	사용자 홈 환경파일 소유자/권한 점검	✅	✅
U-26	파일 권한	World Writable 파일	누구나 쓸 수 있는 파일 존재 여부	✅	✅
U-27	파일 권한	/dev 디바이스 파일	/dev 내 존재하지 않는 디바이스 파일 점검	✅	✅
U-28	서비스 관리	.rhosts/hosts.equiv	r-command 인증 파일 사용 여부 및 권한	✅	✅
U-29	서비스 관리	접속 IP/Port 제한	TCP Wrapper(hosts.allow/deny) 접근 제어	✅	✅

ID	분류	항목명	상세 설명	진단	조치
U-30	서비스 관리	hosts.lpd 파일	프린터 서비스 접근 제어 파일 소유자/권한	✅	✅
U-31	서비스 관리	NIS 서비스 비활성화	NIS 서비스 활성화 여부	✅	✅
U-32	서비스 관리	UMASK 설정	시스템 기본 UMASK(022) 설정 여부	✅	✅
U-33	서비스 관리	홈 디렉터리 권한	홈 디렉터리 소유자 및 권한(750) 설정	✅	✅
U-34	서비스 관리	홈 디렉터리 존재	계정별 홈 디렉터리 존재 여부	✅	✅
U-35	서비스 관리	숨겨진 파일 탐색	디렉터리 내 불필요한 숨겨진 파일 점검	✅	✅
U-36	서비스 관리	Finger 서비스	Finger 서비스 활성화 여부	✅	✅
U-37	서비스 관리	Anonymous FTP	익명 FTP 접속 허용 여부	✅	✅
U-38	서비스 관리	r 계열 서비스	rlogin, rsh, rexec 등 취약 서비스 활성화	✅	✅
U-39	서비스 관리	Cron 파일 권한	cron 접근 제어 파일(allow/deny) 권한	✅	✅
U-40	서비스 관리	DoS 공격 취약 서비스	echo, discard 등 DoS 취약 서비스 활성화	✅	✅

ID	분류	항목명	상세 설명	진단	조치
U-41	서비스 관리	NFS 서비스 비활성화	불필요한 NFS 서비스 활성화 여부	✅	✅
U-42	서비스 관리	NFS 접근 통제	NFS 공유 디렉터리 접근 제어(exports)	✅	✅
U-43	서비스 관리	automountd 제거	automountd 서비스 활성화 여부	✅	✅
U-44	서비스 관리	RPC 서비스 확인	불필요한 RPC 서비스 활성화 여부	✅	✅
U-45	서비스 관리	NIS, NIS+ 점검	NIS/NIS+ 서비스 활성화 여부	✅	✅
U-46	서비스 관리	tftp, talk 서비스	tftp, talk 등 불필요한 서비스 활성화	✅	✅
U-47	서비스 관리	Sendmail 버전	Sendmail 최신 버전 패치 여부	✅	✅
U-48	서비스 관리	스팸 메일 릴레이	SMTP 릴레이 제한 설정 여부	✅	✅
U-49	서비스 관리	일반사용자 Sendmail	일반 사용자의 Sendmail 실행 방지 설정	✅	✅
U-50	서비스 관리	DNS 보안 버전	BIND 최신 버전 사용 여부	✅	✅
U-51	서비스 관리	DNS Zone Transfer	DNS Zone Transfer 제한 설정 여부	✅	✅

ID	분류	항목명	상세 설명	진단	조치
U-52	서비스 관리	Apache 디렉터리 리스팅	디렉터리 인덱싱(Indexes) 제거 여부	✅	✅
U-53	서비스 관리	Apache 프로세스 권한	웹 서비스 데몬의 root 권한 구동 여부	✅	✅
U-54	서비스 관리	Apache 상위 디렉터리	상위 디렉터리 접근 제한(AllowOverride)	✅	✅
U-55	서비스 관리	Apache 불필요 파일	매뉴얼 파일 등 불필요한 파일 제거 여부	✅	✅
U-56	서비스 관리	Apache 링크 사용금지	심볼릭 링크 사용 제한(FollowSymLinks)	✅	✅
U-57	서비스 관리	Apache 파일 업로드	파일 업로드 용량 제한(LimitRequestBody)	✅	✅
U-58	서비스 관리	Apache 영역 분리	DocumentRoot 별도 디렉터리 지정 여부	✅	✅
U-59	서비스 관리	SSH 원격 접속	SSH 서비스 활성화 및 포트 점검	✅	✅
U-60	서비스 관리	FTP 서비스 확인	FTP 서비스 활성화 여부	✅	✅
U-61	서비스 관리	FTP 계정 Shell 제한	FTP 계정에 쉘 부여 여부	✅	✅
U-62	서비스 관리	Ftpusers 파일 권한	ftpusers 파일 소유자/권한	✅	✅

ID	분류	항목명	상세 설명	진단	조치
U-63	서비스 관리	Ftpusers 파일 설정	root 계정 등 FTP 접속 제한 설정	✅	✅
U-64	서비스 관리	at 파일 권한	at 접근 제어 파일(allow/deny) 권한	✅	✅
U-65	서비스 관리	SNMP 서비스	SNMP 서비스 활성화 여부	✅	✅
U-66	서비스 관리	SNMP Community	Community String 복잡성(public 금지)	✅	✅
U-67	서비스 관리	로그온 배너	서버/Telnet/FTP/SMTP/DNS 로그인 배너	✅	✅
U-68	서비스 관리	NFS 설정파일 권한	/etc/exports 파일 소유자/권한	✅	✅
U-69	서비스 관리	expn, vrfy 제한	SMTP expn, vrfy 명령어 제한 설정	✅	✅
U-70	서비스 관리	Apache 정보 숨김	웹 서버 버전/OS 정보 노출 제한	✅	✅
U-71	패치 관리	최신 보안 패치	OS 및 주요 패키지 보안 패치 적용 여부	✅	✅
U-72	로그 관리	로그 정기 검토	각종 로그의 정기적 검토 및 보고 체계	✅	✅
U-73	로그 관리	시스템 로깅 설정	syslog/rsyslog 로깅 정책 설정 여부	✅	✅

3.6.3.2 Web Server (Nginx)

✅ **전체 6개 항목 조치 지원 (v0.4.17)**

- 모든 항목 수동 가이드(Type C) - 상세 설정 변경 명령어 및 가이드 출력

ID	분류	항목명	상세 설명	진단	조치
N-01	정보 노출	버전 정보	HTTP 헤더/에러 페이지 내 버전 노출 여부	✅	✅
N-02	접근 제어	HTTP 메서드	불필요한 HTTP 메서드(PUT, DELETE 등) 허용	✅	✅
N-03	접근 제어	디렉터리 리스팅	디렉터리 내 파일 목록 노출(Autoindex)	✅	✅
N-04	설정 관리	파일 업로드	업로드 파일 크기 제한 설정 여부	✅	✅
N-05	접근 제어	파일 접근	숨겨진 파일/디렉터리 접근 허용 여부	✅	✅
N-06	로그 관리	접근 로그	Access Log 및 Error Log 활성화 여부	✅	✅

3.6.3.3 Database (MariaDB)

✅ **전체 6개 항목 조치 지원 (v0.4.17)**

- 모든 항목 수동 가이드(Type C) - 상세 설정 변경 명령어 및 가이드 출력

ID	분류	항목명	상세 설명	진단	조치
D-01	계정 관리	기본 계정	불필요한 기본 계정(Anonymous) 및 테스트 DB	✅	✅
D-02	계정 관리	Root 접근	Root 계정의 원격 접속 허용 여부	✅	✅
D-03	설정 관리	파일 권한	설정 파일 및 데이터 디렉터리 권한	✅	✅
D-04	패스워드	복잡성	패스워드 복잡성 플러그인 활성화 여부	✅	✅
D-05	네트워크	바인딩 주소	0.0.0.0 리스닝 (원격 접속 전체 허용) 여부	✅	✅
D-06	로그 관리	감사 로그	에러 로그 및 감사 로그 활성화 여부	✅	✅

3.6.4 문제 해결

Q. 조치 후 문제가 발생하여 원복하고 싶습니다. A. `fix` 명령어는 실행 시 항상 원본 파일의 백업(`*.bak`)을 생성합니다. 해당 파일을 원본으로 덮어쓰거나 수동으로 내용을 복구하세요. 예: `cp /etc/ssh/sshd_config.bak_20260118 /etc/ssh/sshd_config`

3.7 보안 취약점 수동 점검 매뉴얼

> ✅ **자동 진단 안내 (v0.4.24+)**
>
> **v0.4.24** 버전부터 KISA 2026 주요정보통신기반시설 기술적 취약점 **전체 항목(85개)에 대한 자동 진단 및 조치**가 구현되었습니다. 따라서 본 매뉴얼의 대부분 항목은 `tkctl analyze security` 명령어로 자동 처리 가능하며, `tkctl fix security` 를 통해 원클릭 조치가 가능합니다.
>
> 자세한 구현 명세는 KISA 2026 보안 취약점 구현 명세서를 참조하십시오. 본 문서는 레거시 환경이나 자동 진단이 불가능한 특수 상황을 위한 참고용으로 유지됩니다.

`tkctl analyze security` 실행 결과, 드물게 '수동 점검 필요(MANUAL)' 상태로 표시되거나 교차 검증이 필요한 경우 본 가이드를 참조하십시오.

> 📌 **참고**
>
> 이 문서는 KISA 주요정보통신기반시설 기술적 취약점 분석 평가 상세 가이드를 기반으로 작성되었습니다.

3.7.1 OS 보안 점검 (U-01 ~ U-73)

U-01. Root 계정 원격 접속 제한

항목	내용
분류	계정 관리
위험도	상
점검 대상	SSH, Telnet 등 원격 접속 서비스

점검 방법

```
# SSH 설정 확인
grep -i "PermitRootLogin" /etc/ssh/sshd_config

# Telnet 설정 확인 (사용 시)
cat /etc/securetty
```

양호 기준

- `PermitRootLogin no` 설정

조치 방법

```
# SSH 설정 변경
sudo sed -i 's/^#*PermitRootLogin.*/PermitRootLogin no/'
/etc/ssh/sshd_config
sudo systemctl restart sshd
```

U-02. 패스워드 복잡성 설정

항목	내용
분류	계정 관리
위험도	상
점검 대상	PAM 설정

점검 방법

```
# PAM 복잡성 설정 확인
cat /etc/pam.d/system-auth | grep pam_pwquality
cat /etc/security/pwquality.conf
```

양호 기준

- 영문, 숫자, 특수문자 조합
- 최소 8자 이상

조치 방법

```
# /etc/security/pwquality.conf 설정
minlen = 8
dcredit = -1
ucredit = -1
lcredit = -1
ocredit = -1
```

U-03. 계정 잠금 임계값 설정

항목	내용
분류	계정 관리
위험도	상
점검 대상	PAM faillock 설정

점검 방법

```
grep -E "deny=|faillock" /etc/pam.d/system-auth
grep -E "deny=|faillock" /etc/pam.d/password-auth
```

양호 기준

- 5회 이하 로그인 실패 시 계정 잠금

조치 방법

```
# faillock 설정 예시
auth required pam_faillock.so preauth deny=5 unlock_time=600
auth [default=die] pam_faillock.so authfail deny=5 unlock_time=600
```

U-04. 패스워드 파일 보호

항목	내용
분류	계정 관리
위험도	상
점검 대상	/etc/passwd, /etc/shadow

점검 방법

```
# Shadow 파일 사용 여부 확인
awk -F: '$2 != "x" && $2 != "!" && $2 != "*" {print $1}' /etc/passwd
```

양호 기준

- 모든 계정의 패스워드가 /etc/shadow에 암호화되어 저장

U-05. Root 이외의 UID 0 금지

항목	내용
분류	계정 관리
위험도	상
점검 대상	/etc/passwd

점검 방법

```
awk -F: '$3 == 0 {print $1}' /etc/passwd
```

양호 기준

- root 외 UID 0을 가진 계정이 없음

U-06. root 홈/패스 권한

항목	내용
분류	파일 권한
위험도	상
점검 대상	su 명령어 권한

점검 방법

```
ls -l /usr/bin/su
cat /etc/pam.d/su | grep pam_wheel
```

양호 기준

- su 명령어가 wheel 그룹으로 제한됨

U-07 ~ U-09. 패스워드 정책 점검

항목	점검 대상	권장 설정
U-07	최소 길이	8자 이상 (`PASS_MIN_LEN 8`)
U-08	최대 사용기간	90일 이하 (`PASS_MAX_DAYS 90`)
U-09	최소 사용기간	1일 이상 (`PASS_MIN_DAYS 1`)

점검 방법

```
cat /etc/login.defs | grep -E "PASS_MIN_LEN|PASS_MAX_DAYS|PASS_MIN_DAYS"
```

U-10 ~ U-14. 계정 관리 점검

ID	점검 항목	점검 명령어
U-10	불필요한 계정	`cat /etc/passwd | grep -E "lp:|uucp:|nuucp:"`
U-11	관리자 그룹	`grep "^root" /etc/group`
U-12	소유자 없는 그룹	`find / -nogroup -print 2>/dev/null`
U-13	동일 UID	`awk -F: '{print $3}' /etc/passwd | sort | uniq -d`
U-14	사용자 Shell	`grep -E "^nobody|^bin|^daemon" /etc/passwd`

U-15. Session Timeout 설정

점검 방법

```
echo $TMOUT
cat /etc/profile | grep TMOUT
```

양호 기준

- TMOUT=600 이하 설정

조치 방법

```
echo "export TMOUT=600" >> /etc/profile
source /etc/profile
```

U-16 ~ U-27. 파일/디렉터리 권한 점검

ID	점검 대상	권장 권한	점검 명령어
U-16	PATH 환경변수	'.' 미포함	`echo $PATH`
U-17	소유자 없는 파일	없음	`find / -nouser -print 2>/dev/null`
U-18	/etc/passwd	644 (root)	`ls -l /etc/passwd`
U-19	/etc/shadow	400 (root)	`ls -l /etc/shadow`
U-20	/etc/hosts	600 (root)	`ls -l /etc/hosts`
U-21	/etc/xinetd.conf	600 (root)	`ls -l /etc/xinetd.conf`
U-22	/etc/rsyslog.conf	644 (root)	`ls -l /etc/rsyslog.conf`
U-23	/etc/services	644 (root)	`ls -l /etc/services`
U-24	SUID/SGID	불필요 파일 제거	`find / -perm -4000 -print 2>/dev/null`
U-25	사용자 환경파일	적정 권한	`ls -la ~/.bashrc ~/.profile`
U-26	World Writable	없음	`find / -perm -2 -type f 2>/dev/null`
U-27	/dev 디바이스	정상 파일만	`find /dev -type f 2>/dev/null`

U-28 ~ U-35. 서비스 관리 점검 (1)

ID	점검 항목	점검 명령어
U-28	.rhosts 파일	`find / -name ".rhosts" 2>/dev/null`
U-29	TCP Wrapper	`cat /etc/hosts.allow /etc/hosts.deny`
U-30	hosts.lpd	`ls -l /etc/hosts.lpd`
U-31	NIS 서비스	`systemctl status ypserv ypbind`
U-32	UMASK 설정	`grep UMASK /etc/login.defs`
U-33	홈 디렉터리 권한	`ls -ld /home/*`
U-34	홈 디렉터리 존재	`awk -F: '{print $6}' /etc/passwd | xargs ls -d 2>&1`
U-35	숨겨진 파일	`find / -name ".*" -type f 2>/dev/null`

U-36 ~ U-51. 서비스 관리 점검 (2)

ID	점검 항목	양호 기준	점검 명령어
U-36	Finger 서비스	비활성화	`systemctl status finger`
U-37	Anonymous FTP	비활성화	`grep anonymous_enable /etc/vsftpd/vsftpd.conf`
U-38	r 계열 서비스	비활성화	`systemctl status rsh rlogin rexec`
U-39	Cron 파일 권한	600 (root)	`ls -l /etc/cron.allow /etc/cron.deny`
U-40	DoS 취약 서비스	비활성화	`grep -E "echo|discard|daytime" /etc/xinetd.d/*`
U-41	NFS 서비스	필요시만 활성화	`systemctl status nfs`
U-42	NFS 접근 통제	적정 설정	`cat /etc/exports`
U-43	automountd	비활성화	`systemctl status autofs`
U-44	RPC 서비스	비활성화	`rpcinfo -p localhost`
U-45	NIS, NIS+	비활성화	`systemctl status ypserv`
U-46	tftp, talk	비활성화	`systemctl status tftp talk`
U-47	Sendmail 버전	최신 버전	`sendmail -d0.1 -bv root 2>&1 | head -1`
U-48	스팸 릴레이	제한 설정	`postconf | grep smtpd_relay_restrictions`
U-49	Sendmail 실행권한	제한	`ls -l /usr/sbin/sendmail`
U-50	DNS 버전	최신 버전	`named -v`

ID	점검 항목	양호 기준	점검 명령어
U-51	Zone Transfer	제한	`grep allow-transfer /etc/named.conf`

U-52 ~ U-70. 서비스 관리 점검 (3)

ID	점검 항목	양호 기준	점검 명령어
U-52	Apache 디렉터리 리스팅	Indexes 제거	`grep -r "Indexes" /etc/httpd/`
U-53	Apache 프로세스 권한	비root 구동	`ps -ef | grep httpd`
U-54	Apache 상위 디렉터리	AllowOverride None	`grep AllowOverride /etc/httpd/conf/httpd.conf`
U-55	Apache 불필요 파일	매뉴얼 제거	`ls /var/www/html/manual/`
U-56	Apache 링크 사용금지	FollowSymLinks 제거	`grep FollowSymLinks /etc/httpd/conf/httpd.conf`
U-57	Apache 파일 업로드	LimitRequestBody 설정	`grep LimitRequestBody /etc/httpd/conf/httpd.conf`
U-58	Apache 영역 분리	별도 디렉터리	`grep DocumentRoot /etc/httpd/conf/httpd.conf`
U-59	SSH 원격 접속	활성화	`systemctl status sshd`
U-60	FTP 서비스	필요시만	`systemctl status vsftpd`
U-61	FTP 계정 Shell	제한	`grep -E "ftp:|ftpuser:" /etc/passwd`
U-62	Ftpusers 파일 권한	640 (root)	`ls -l /etc/vsftpd/ftpusers`
U-63	Ftpusers 설정	root 포함	`cat /etc/vsftpd/ftpusers`
U-64	at 파일 권한	640 (root)	`ls -l /etc/at.allow /etc/at.deny`
U-65	SNMP 서비스	필요시만	`systemctl status snmpd`
U-66	SNMP Community	public 금지	`grep community /etc/snmp/snmpd.conf`

ID	점검 항목	양호 기준	점검 명령어
U-67	로그온 배너	설정	`cat /etc/issue /etc/issue.net`
U-68	NFS 설정파일 권한	644 (root)	`ls -l /etc/exports`
U-69	expn, vrfy 제한	비활성화	`postconf | grep disable_vrfy_command`
U-70	Apache 정보 숨김	ServerTokens Prod	`grep ServerTokens /etc/httpd/conf/httpd.conf`

U-71 ~ U-73. 패치 및 로그 관리

ID	점검 항목	점검 방법
U-71	최신 보안 패치	`yum check-update --security` 또는 `apt list --upgradable`
U-72	로그 정기 검토	로그 검토 정책 및 이력 확인 (문서/프로세스)
U-73	시스템 로깅 설정	`cat /etc/rsyslog.conf` 정책 확인

3.7.2 WEB 보안 점검 (N-01 ~ N-06)

Nginx 웹 서버 보안 설정 점검 항목입니다.

Nginx 설정 파일 위치

```
# 공통 경로
/etc/nginx/nginx.conf
/etc/nginx/conf.d/*.conf

# Tachyon TTS 환경
/usr/local/TACHYON/TTS40/nginx/conf/nginx.conf
```

N-01. 버전 정보 노출 제한

항목	내용
분류	정보 노출
위험도	중

점검 방법

```
grep -i "server_tokens" /etc/nginx/nginx.conf
curl -I http://localhost 2>&1 | grep Server
```

양호 기준

- `server_tokens off;` 설정

조치 방법

```
# nginx.conf의 http 블록 내
http {
    server_tokens off;
}
```

N-02. HTTP 메서드 제한

항목	내용
분류	접근 제어
위험도	중

점검 방법

```
grep -i "limit_except" /etc/nginx/nginx.conf
curl -X OPTIONS http://localhost -I
```

양호 기준

- 불필요한 메서드(PUT, DELETE, TRACE 등) 차단

조치 방법

```
location / {
    limit_except GET POST {
        deny all;
    }
}
```

N-03. 디렉터리 리스팅 제한

항목	내용
분류	접근 제어
위험도	상

점검 방법

```
grep -i "autoindex" /etc/nginx/nginx.conf
```

양호 기준

- `autoindex off;` 또는 설정 없음 (기본값 off)

조치 방법

```
# autoindex on; 설정 제거 또는 off로 변경
autoindex off;
```

N-04. 파일 업로드 제한

항목	내용
분류	설정 관리
위험도	중

점검 방법

```
grep -i "client_max_body_size" /etc/nginx/nginx.conf
```

양호 기준

- 적정 크기로 제한 (예: 10M)

조치 방법

```
http {
    client_max_body_size 10M;
}
```

N-05. 숨겨진 파일 접근 제한

항목	내용
분류	접근 제어
위험도	중

점검 방법

```
grep -E "location.*\\\\." /etc/nginx/nginx.conf
curl http://localhost/.htaccess
```

양호 기준

- `.`으로 시작하는 파일 접근 차단

조치 방법

```
location ~ /\. {
    deny all;
}
```

N-06. 로그 설정 확인

항목	내용
분류	로그 관리
위험도	중

점검 방법

```
grep -E "access_log|error_log" /etc/nginx/nginx.conf
ls -l /var/log/nginx/
```

양호 기준

- access_log 및 error_log 모두 활성화

조치 방법

```
http {
    access_log /var/log/nginx/access.log;
    error_log /var/log/nginx/error.log warn;
}
```

3.7.3 DB 보안 점검 (D-01 ~ D-06)

MariaDB/MySQL 데이터베이스 보안 설정 점검 항목입니다.

MariaDB 설정 파일 위치

```
# 공통 경로
/etc/my.cnf
/etc/mysql/my.cnf
/etc/my.cnf.d/*.cnf

# Tachyon TTS 환경
/usr/local/TACHYON/TTS40/mariadb/my.cnf
```

D-01. 기본 계정 관리

항목	내용
분류	계정 관리
위험도	상

점검 방법

```
-- 익명 계정 확인
SELECT User, Host FROM mysql.user WHERE User='';

-- 테스트 데이터베이스 확인
SHOW DATABASES LIKE 'test%';
```

양호 기준

- 익명 계정 및 테스트 데이터베이스 없음

조치 방법

```
-- 익명 계정 삭제
DROP USER ''@'localhost';
DROP USER ''@'%';

-- 테스트 데이터베이스 삭제
DROP DATABASE IF EXISTS test;

-- 또는 mysql_secure_installation 실행
```

D-02. Root 원격 접속 제한

항목	내용
분류	계정 관리
위험도	상

점검 방법

```
SELECT User, Host FROM mysql.user WHERE User='root';
```

양호 기준

- root 계정의 Host가 'localhost' 또는 '127.0.0.1'만 허용

조치 방법

```
-- 원격 root 접속 제거
DELETE FROM mysql.user WHERE User='root' AND Host NOT IN ('localhost',
'127.0.0.1', '::1');
FLUSH PRIVILEGES;
```

D-03. 설정 파일 권한

항목	내용
분류	설정 관리
위험도	중

점검 방법

```
ls -l /etc/my.cnf
ls -l /var/lib/mysql/
```

양호 기준

- 설정 파일: 640 이하 (root/mysql 소유)
- 데이터 디렉터리: 750 (mysql 소유)

조치 방법

```
chmod 640 /etc/my.cnf
chown root:mysql /etc/my.cnf
chmod 750 /var/lib/mysql
chown mysql:mysql /var/lib/mysql
```

D-04. 패스워드 복잡성

항목	내용
분류	패스워드
위험도	중

점검 방법

```
SHOW VARIABLES LIKE 'validate_password%';
-- 또는
SHOW VARIABLES LIKE 'simple_password_check%';
```

양호 기준

- 패스워드 검증 플러그인 활성화

조치 방법

```
-- MariaDB
INSTALL PLUGIN simple_password_check SONAME 'simple_password_check';

-- MySQL
INSTALL COMPONENT 'file://component_validate_password';
SET GLOBAL validate_password.policy = MEDIUM;
```

D-05. 네트워크 바인딩 설정

항목	내용
분류	네트워크
위험도	상

점검 방법

```
grep bind-address /etc/my.cnf /etc/my.cnf.d/*
netstat -tlnp | grep 3306
```

양호 기준

- `bind-address = 127.0.0.1` 또는 특정 IP로 제한

조치 방법

```
# my.cnf [mysqld] 섹션
[mysqld]
bind-address = 127.0.0.1
```

D-06. 감사 로그 설정

항목	내용
분류	로그 관리
위험도	중

점검 방법

```
grep -E "log_error|server_audit" /etc/my.cnf
ls -l /var/log/mysql/
```

양호 기준

- 에러 로그 활성화
- 감사 플러그인 사용 권장

조치 방법

```
# my.cnf [mysqld] 섹션
[mysqld]
log_error = /var/log/mysql/error.log

# 감사 플러그인 (선택)
plugin-load = server_audit=server_audit.so
server_audit_logging = ON
server_audit_file_path = /var/log/mysql/audit.log
```

3.7.4 참고 자료

- KISA 주요정보통신기반시설 기술적 취약점 분석·평가 상세 가이드
- CIS Benchmarks
- Nginx Security Guide
- MariaDB Security Best Practices

4. 대화형 쉘 모드 (Interactive Shell)

인자 `--shell` (또는 `-s`) 플래그와 함께 `tkctl`를 실행하면 go-prompt 기반의 지능형 대화형 쉘 모드로 진입합니다.

4.1 실행

```
$ tkctl --shell
# 또는
$ tkctl -s
tkctl>
```

> **주의**
>
> 대화형 쉘 모드는 TTY(터미널) 환경을 필수적으로 요구합니다. 파이프(pipe)나 리다이렉션 상황에서는 패닉 방지를 위해 실행이 제한됩니다.

4.2 주요 기능

지능형 자동 완성 (Intelligent Completion)

명령어, 서브커맨드뿐만 아니라 **플래그(옵션)**와 **동적 인자** 자동 완성을 지원합니다.

- **명령어 완성**: 단어 입력 중 `TAB`을 눌러 자동 완성 및 하단 가이드 확인.
- **플래그 완성**: `-` 또는 `--` 입력 시 현재 명령어에서 사용 가능한 모든 옵션 목록 표시.
- **동적 추천**: `service start` 등 실행 시 현재 시스템의 실제 서비스 목록을 실시간으로 가져와 제안.

프리미엄 테마 및 입력 보호

- **세련된 UI**: Cyan 색상의 프롬프트와 가독성 높은 제안 목록 테마를 적용하였습니다.
- **Backspace 보호**: 실수로 프롬프트 영역(`tkctl>`)이 지워지지 않도록 입력 영역을 엄격히 보호합니다.
- **최적화된 팝업**: 최대 15개의 제안 목록을 한눈에 확인할 수 있습니다.

Signal Handling & History

- **스마트 Ctrl+C**: 명령어 실행 중에는 해당 작업만 중단하고, 입력 대기 중에는 입력 버퍼만 비워 즉각적인 재입력이 가능합니다.
- **History**: 위/아래 방향키로 이전 명령 히스토리를 탐색합니다.

종료

다음 방법으로 쉘을 종료할 수 있습니다:

- `exit` 입력
- `quit` 입력
- `Ctrl+D` (EOF)

4.3 사용 예시

```
$ tkctl -s
tkctl> version
tkctl 버전: 0.4.1
...

tkctl> service start --<TAB>
--help   --force   --no-wait
...

tkctl> exit
Bye!
```

5. 전역 옵션

모든 명령어에서 사용 가능한 전역 옵션입니다.

5.1 기본 디렉토리 설정 (`--basedir` , `-b`)

기본 설치 경로가 아닌 다른 경로를 사용할 경우 설정합니다.

설정값은 `tkctl.ini` 파일에 영구적으로 저장됩니다.

```
# 경로 변경 및 저장
tkctl -b /custom/path info

# 또는
tkctl --basedir=/custom/path info
```

5.2 디버그 모드 (`--debug` , `-d`)

디버깅을 위한 상세 출력을 활성화합니다.

```
tkctl -d service check
tkctl --debug info
```

5.3 설정 제거 (`--uninstall` , `-u`)

tkctl가 설치한 자동 완성 스크립트와 설정 파일을 제거합니다.

```
tkctl --uninstall
# 또는
tkctl -u
```

제거 대상:

- Bash completion 스크립트 (`/etc/bash_completion.d/tkctl` 등)
- Completion 체크 플래그 파일 (`~/.tkctl_completion_checked`)
- 레거시 `.bashrc` 설정 블록

5.4 버전 정보 (`--version` , `-v`)

간단한 버전 정보를 출력합니다.

```
tkctl -v
# 또는
tkctl --version
# 출력: tkctl version 0.3.19
```

5.5 쉘 모드 진입 (`--shell` , `-s`)

인프라 관리 및 테스트를 위해 인터랙티브 쉘 모드로 명시적으로 진입합니다.

```
tkctl -s
# 또는
tkctl --shell
```

6. 다국어 지원 (i18n)

tkctl은 시스템의 언어 설정을 자동으로 감지하여 다국어를 지원합니다.

6.1 자동 언어 감지

시스템의 `LANG` 또는 `LC_ALL` 환경 변수를 감지합니다.

```
# 영문 환경
LANG=en_US.UTF-8 tkctl info
# → 영어 출력

# 한글 환경
LANG=ko_KR.UTF-8 tkctl info
# → 한국어 출력
```

6.2 수동 언어 설정 및 강제 전환

SSH 클라이언트 설정이나 터미널 폰트 문제로 한글이 깨져 보일 경우, 다음 방법 중 하나를 선택하여 강제로 영문 언어로 전환할 수 있습니다.

방법 1: 실행 시 플래그 사용 (일회성 및 환경 자동 반영)

모든 명령어에 `-l` (소문자 L) 또는 `--lang` 플래그를 사용하여 언어를 지정할 수 있습니다. 한번 이 플래그를 사용하여 실행하면, 해당 설정이 자동으로 `tkctl.ini`에 저장되어 이후 실행부터는 플래그 없이도 유지됩니다. 또한 `auto` 값을 입력하면 저장된 설정을 삭제하고 다시 시스템 자동 감지 모드로 복원할 수 있습니다.

```
# 강제로 영문으로 전환하고 설정 저장
tkctl --lang en version

# 다시 자동 감지 모드로 복원 (설정 삭제)
tkctl --lang auto version

# 강제로 한글로 전환하고 설정 저장
tkctl -l ko version
```

방법 2: 설정 파일에서 영구 수정 (`tkctl.ini`)

`tkctl` 이 설치된 경로의 `tkctl.ini` 파일을 직접 수정하여 언어를 고정할 수 있습니다.

```
basedir=/usr/local/TACHYON/TTS40
log_level=DEBUG
language=en
```

6.3 지능형 인코딩 감지 (자동 전환)

`tkctl` 은 터미널 환경이 한글 출력을 정상적으로 지원하지 못할 것으로 판단되면 본인(시스템)의 로케일이 한글(`ko`)이더라도 자동으로 영문(`en`)으로 전환하여 출력 깨짐을 방지합니다.

- **감지 로직**: `LANG` 환경 변수가 `ko` 계열이지만 `UTF-8` 문자열을 포함하지 않는 경우 (예: `ko_KR.EUC-KR` 또는 `ko_KR`), 안전을 위해 자동으로 영문 언어팩을 활성화합니다.
- **복구 방법**: 만약 터미널이 한글 지원이 가능함에도 영문으로 나온다면, `tkctl --lang ko` 명령을 실행하여 수동으로 설정을 강제화하십시오.

6.4 내장 언어팩

- **영어 (en)**: 기본 언어
- **한국어 (ko)**: 완전 번역

6.5 다국어 데이터 정합성 유지 (v0.3.30+)

tkctl은 각 언어별 JSON 파일 간의 키 일치성을 자동화된 테스트를 통해 보장합니다.

데이터 검증 테스트

개발 단계에서 `go test ./internal/i18n/...` 명령을 통해 다음 사항을 검증합니다:

- **키 일치성 (`TestI18nKeysConsistency`)**: `ko.json` 과 `en.json` 에 정의된 모든 메시지 키가 1:1로 정확히 일치하는지 확인합니다.
- **도움말 필수 키 (`TestHelpKeysExist`)**: 모든 필수 명령어의 `Long` 설명과 `Example` 예시 키가 누락 없이 존재하는지 확인합니다.

명령어 도움말 키 명칭 규칙

동적 언어 전환 및 서브커맨드 지원을 위해 다음과 같은 명칭 패턴을 권장하며, 시스템은 이를 자동으로 인지하여 적용합니다:

- **명령어 설명**: `[command]_short` , `[command]_long`

- **예제**: `[command]_example`
- **서브커맨드**: `[parent]_[child]_short`, `[parent]_[child]_long`
- **플래그**: `[command]_[flag_name]_flag`

6.6 사용자 정의 언어팩

tkctl 실행 경로에 `lang` 디렉토리를 만들고 JSON 파일을 추가하면 새로운 언어를 지원할 수 있습니다.

언어팩 생성 절차

1. **디렉토리 생성**

```
mkdir lang
```

2. **언어팩 파일 생성**

```
vim lang/ja.json
```

3. **JSON 작성**

```
{
  "meta": {
    "version": "1.0",
    "compatibility": "0.1.0"
  },
  "messages": {
    "root_short": "TACHYON CLI - システム管理ツール",
    "info_title": "### TACHYON システム情報 ###",
    "service_check_title": "### TACHYON サービス状態チェック ###"
  }
}
```

4. **실행**

```
LANG=ja_JP.UTF-8 ./tkctl info
```

표준 언어팩 파일명

언어	파일명	환경 변수 예시
영어	`en.json`	`en_US.UTF-8`
한국어	`ko.json`	`ko_KR.UTF-8`
일본어	`ja.json`	`ja_JP.UTF-8`
중국어 (간체)	`zh-cn.json`	`zh_CN.UTF-8`
중국어 (번체)	`zh-tw.json`	`zh_TW.UTF-8`
프랑스어	`fr.json`	`fr_FR.UTF-8`
독일어	`de.json`	`de_DE.UTF-8`
스페인어	`es.json`	`es_ES.UTF-8`
러시아어	`ru.json`	`ru_RU.UTF-8`

6.4 폴백 메커니즘

- 요청한 언어의 키가 없으면 자동으로 영어로 폴백됩니다.
- 언어팩 파일이 없으면 영어가 사용됩니다.

6.7 시스템 로케일 설정 (System Locale)

`tkctl env locale` 명령어를 사용하여 OS 시스템 전체의 로케일 설정을 관리할 수 있습니다. 이는 터미널에서의 한글 깨짐 현상을 근본적으로 해결하는 데 유용합니다.

```
# 시스템 로케일 확인
tkctl env locale

# 한글 로케일(ko_KR.UTF-8)로 설정
tkctl env locale ko --set
```

> 상세한 사용법은 5. 환경 구성 > 3.5.8 Locale 설정을 참조하십시오.

7. 로깅 시스템

tkctl은 자체 로그를 파일에 기록하며, TACHYON 운영 서비스의 로그 로테이션 상태를 확인하고 관리하는 기능을 제공합니다.

7.1 tkctl 자체 로깅

로그 파일 위치 (우선순위)

1. `/usr/local/TACHYON/TTS40/logs/tkctl_YYYYMMDD.log` (기본)
2. `~/.tkctl/logs/tkctl_YYYYMMDD.log` (권한 없을 시)
3. `/tmp/tkctl_logs/tkctl_YYYYMMDD.log` (최종 대체)

로그 레벨 설정

`tkctl.ini` 파일에서 로그 레벨을 설정할 수 있습니다.

설정 파일 위치:

- tkctl 실행 경로의 `tkctl.ini`

설정 예시:

```
basedir=/usr/local/TACHYON/TTS40
log_level=INFO
verbose=false
```

로그 레벨:

- `DEBUG` : 모든 로그 기록 (개발/디버깅용)
- `INFO` : 일반 정보 및 상태 변경 (기본값)
- `WARN` : 경고 메시지만
- `ERROR` : 오류만 기록

로그 확인

```
# 오늘 로그 확인
tail -f /usr/local/TACHYON/TTS40/logs/tkctl_$(date +%Y%m%d).log
```

```
# 최근 100줄
tail -100 /usr/local/TACHYON/TTS40/logs/tkctl_$(date +%Y%m%d).log

# 특정 날짜 로그
cat /usr/local/TACHYON/TTS40/logs/tkctl_20251207.log
```

로그 보관 정책

tkctl 자체 로그는 **7일간 보관** 후 자동으로 삭제됩니다. 이 정책은 내부 `LogMaxAge` 상수로 관리됩니다.

7.2 서비스 로그 로테이션 관리 (logrotate 명령어)

TACHYON 운영 환경에서는 다양한 미들웨어와 서비스가 로그를 생성합니다. `tkctl logrotate` 명령어를 통해 모든 서비스의 로그 로테이션 상태를 한눈에 확인하고, 필요시 자동으로 설정할 수 있습니다.

빠른 사용법

```
# 전체 서비스 로그 로테이션 상태 확인
tkctl logrotate

# 특정 서비스만 확인
tkctl logrotate nginx

# 누락된 설정 자동 적용
tkctl logrotate --set
```

지원 서비스

서비스 유형	서비스 목록	로테이션 방식
미들웨어	nginx, redis	시스템 logrotate
미들웨어	opensearch, kafka, logstash, mariadb	내부 관리
TACHYON	Api, Auth, Manager, Stat, Batch, Report, Watchdog	Log4j2
관리 도구	tkadmin	tkadmin.yml 설정
관리 도구	tkctl	자체 로테이션

상세 사용법

자세한 사용법과 출력 예시는 서비스 관리 - 3.2.2 서비스 로그 로테이션 섹션을 참조하세요.

7.3 서비스별 로그 위치

서비스	로그 경로
tkctl	`$BASEDIR/logs/tkctl_YYYYMMDD.log`
tkadmin	`$BASEDIR/tkadmin/logs/`
NGINX	`$BASEDIR/nginx/logs/`
Redis	`$BASEDIR/redis/logs/`
OpenSearch	`$BASEDIR/opensearch/opensearch/logs/`
Kafka	`$BASEDIR/kafka/logs/`
Java 서비스	`$BASEDIR/logs/[ServiceName]/`

> **참고**
>
> `$BASEDIR` 은 기본적으로 `/usr/local/TACHYON/TTS40` 입니다.

8. 문제 해결

8.1 일반적인 문제

명령어를 찾을 수 없음

```
# PATH 확인
echo $PATH

# tkctl 위치 확인
which tkctl

# PATH에 추가 (임시)
export PATH=$PATH:/usr/local/bin

# PATH에 추가 (영구)
echo 'export PATH=$PATH:/usr/local/bin' >> ~/.bashrc
source ~/.bashrc
```

권한 오류

```
# 실행 권한 부여
chmod +x tkctl

# 소유권 변경
sudo chown $USER:$USER tkctl
```

데이터베이스 연결 실패

```
# MariaDB 상태 확인
tkctl service check | grep mariadb

# MariaDB 시작
tkctl service start mariadb
```

```
# 연결 테스트
mysql -u root -p
```

패키지 생성 시 버전 감지 실패

```
# MariaDB 버전 미지정 시 "최신 버전을 찾을 수 없습니다" 오류가 발생하는 경우

# 1. 인터넷 연결 확인
curl -s -o /dev/null -w "%{http_code}" https://downloads.mariadb.org
curl -s -o /dev/null -w "%{http_code}" https://archive.mariadb.org

# 2. 두 사이트 모두 차단된 경우 → 버전을 직접 지정
tkctl update package mariadb --version 10.11.16

# 3. v0.5.7 이상으로 업데이트 시 archive.mariadb.org 폴백 자동 지원
tkctl version
```

8.2 자동 완성이 작동하지 않음

```
# Completion 재설치
tkctl completion bash | sudo tee /etc/bash_completion.d/tkctl

# .bashrc 확인
grep tkctl ~/.bashrc
```

8.3 시스템 환경 문제

SELinux 로 인한 서비스 오류

TACHYON 서비스가 정상적으로 시작되지 않거나 파일 쓰기 권한 오류가 발생하는 경우 SELinux 설정을 확인하십시오.

```
# SELinux 상태 확인
tkctl env selinux

# SELinux 를 Permissive 모드로 전환 (로그만 활성, 차단 해제)
tkctl env selinux --set permissive
```

부록 A: 서비스 로깅 레벨 진단 기술 명세

본 부록은 `tkctl service loglevel` 명령어가 각 서비스의 로깅 레벨을 진단하기 위해 참조하는 설정 파일의 위치와 파싱 로직을 상세히 기술합니다. 시스템 엔지니어는 본 가이드를 참조하여 `tkctl`가 리포팅하는 정보의 근거를 파악하고, 필요시 수동으로 설정을 검증할 수 있습니다.

---

1. 개요

`tkctl`는 각 서비스의 설정 파일을 텍스트 기반으로 직접 파싱하거나 YAML 파서를 사용하여 로깅 관련 지시어를 추출합니다. 각 서비스는 설치 기반 디렉토리(`${BaseDir}`)를 기준으로 상대 경로를 탐색합니다.

2. 미들웨어 서비스 (Middleware)

서비스	참조 파일 경로	진단 방식 (Parsing Logic)
NGINX	`nginx/conf/nginx.conf`	**1. Error Log**: `error_log` 지시어의 마지막 인자를 추출하여 레벨(DEBUG, INFO 등)을 표시합니다. **2. Access Log**: NGINX는 액세스 로그에 레벨 개념을 사용하지 않으므로, `access_log off;` 지시어 존재 여부만 확인하여 **ON/OFF** 상태로 리포팅합니다. 표시 형식: `레벨 (Access: ON/OFF)`
Redis	`redis/conf/redis.conf` 또는 `redis/redis.conf`	`loglevel` 지시어의 우측 값을 추출합니다. 표시 예: `loglevel notice` → **NOTICE**
MariaDB	`mariadb/my.cnf`	파일 존재 여부만 확인하며, 기본값으로 **INFO (Default)**를 표시합니다. *(MariaDB는 단일 레벨 설정보다 상세 로깅 설정을 주로 사용함)*

3. TACHYON Java 서비스

TACHYON의 모든 Java 서비스(`api` , `auth` , `manager` , `stat` , `batch` , `report` , `watchdog`)는 다음의 우선순위에 따라 파일을 탐색합니다.

3.1 설정 파일 탐색 우선순위

1. `[svc]/conf/[svc].yml_dev` (YAML) - **최우선 순위 (Plaintext 원본)**
2. `[svc]/conf/[svc].yml` (YAML)
3. `[svc]/conf/logback-spring.xml` (XML)
4. `[svc]/conf/log4j2.xml` (XML)
5. `[svc]/conf/application.yml` (YAML)

3.2 파싱 기술 명세

YAML 파싱 (`.yml` , `.yml_dev`)

`gopkg.in/yaml.v3` 라이브러리를 사용하여 다음 계층의 값을 추출합니다:

- **경로**: `logging` > `level` > `root`
- **구조 예시**:

```
logging:
  level:
    root: INFO  # <-- 추출 대상
```

XML 파싱 (`.xml`)

문자열 패턴 매칭을 통해 `<root>` 태그의 `level` 속성을 추출합니다:

- **정규식/패턴**: `<root level="([^"]+)">`
- **구조 예시**:

```
<root level="DEBUG">  <!-- <-- level 속성값 추출 -->
    <appender-ref ref="CONSOLE" />
</root>
```

4. 관리 도구 (Management Tools)

서비스	참조 파일 경로	진단 방식 (Parsing Logic)
tkadmin	`${BaseDir}/tkadmin.yml`	YAML 파싱을 통해 `logging.level.root` 값을 추출합니다.
tkctl	`${BaseDir}/tkctl.ini`	설정 파일 내 `log_level` 키값을 확인합니다. (미설정 시 `INFO`)

5. 진단 상태(Status) 설명

상태	의미	대응 방법
OK	설정 파일을 정상적으로 찾고 파싱함	-
CONFIG NOT FOUND	지정된 경로에 설정 파일이 존재하지 않음	해당 서비스의 설치 경로를 확인하십시오.
INVALID	파일은 발견했으나 로깅 레벨 지시어 부재	설정 파일 내에 `root` 레벨 설정이 누락되었는지 확인하십시오.
-	진단 불가능	특정 서비스를 지정하여 상세 진단을 수행하십시오.

> **참고**
>
> `tkctl` 는 Java 서비스의 경우 `yml_dev` 파일을 수정하고 이를 `yml` 로 복사하여 반영하는 관리 워크플로우를 권장합니다.

---

6. 시스템 환경 설정 진단 및 구성 명세 (env)

`tkctl env` 명령어는 TACHYON 운영에 필수적인 커널 및 시스템 환경 변수를 진단하고 최적화합니다.

6.1 SELinux

구분	진단/참조 로직
현재 실행 모드	`getenforce` 명령어를 우선 실행하며, 부재 시 `sestatus` 결과에서 `Current mode` 를 추출합니다.
영구 설정 모드	`/etc/selinux/config` 파일의 `SELINUX=` 지시어 값을 직접 파싱합니다.
변경 프로세스	`--set` 실행 시 `setenforce` 를 통한 즉시 반영과 `config` 수정을 통한 영구 반영을 동시에 수행합니다. (Disabled 설정 시 재부팅 안내)

6.2 시스템 자원 제한 (Ulimit)

구분	진단/참조 로직
커널 전체 제한	`/etc/security/limits.conf` 파일을 파싱하여 도메인별 `nofile` 설정을 수집합니다.
서비스별 제한	`systemctl show -p LimitNOFILE --value [SVC]` 명령을 통해 systemd 유닛에 구동 시 할당된 실제 값을 조회합니다.
TACHYON 블록	`--set` 기능을 통한 설정 시, 기존 파일 훼손을 방지하기 위해 `# --- TACHYON LIMITS START ---` 마커를 사용한 전용 블록을 관리합니다.

6.3 방화벽 (Firewall)

- **활성 상태 감지**: `systemctl is-active`를 통해 `firewalld`와 `iptables` 서비스의 동작 여부를 동시에 체크합니다.
- **포트 목록 추출**: `firewalld`가 활성 상태일 때 `firewall-cmd --list-ports` 및 `--list-services` 결과를 파싱하여 제공합니다.
- **자동 허용 로직**: `--set` 인자 생략 시 TACHYON 관리 포트(443)와 현재 운영 중인 SSH 포트를 자동 탐지하여 개방합니다.

6.4 SSH 설정 및 연동

- **포트 감지**: `/etc/ssh/sshd_config` 파일에서 주석처리 되지 않은 `Port` 지시어를 탐색합니다. (지시어 부재 시 22번 표준 포트로 간주)
- **변경 및 연동 워크플로우**:
  1. `/etc/ssh/sshd_config` 내 Port 번호 수정
  2. `firewall-cmd --add-port=[NEW_PORT]/tcp`를 통해 신규 포트 방화벽 영구 허용 (`--permanent`)
  3. `firewall-cmd --reload`를 실행하여 방화벽 설정 즉시 반영
  4. `systemctl restart sshd`를 실행하여 설정 완료

---

7. JVM 메모리 설정 및 자동 최적화 기술 명세 (service jvm)

`tkctl service jvm` 명령어는 JVM 기반 서비스의 힙 메모리를 진단하고 최적화하기 위해 다음 기술 명세를 따릅니다.

7.1 서비스별 설정 파일 및 파싱 로직

서비스	설정 파일 경로	추출/변경 로직
OpenSearch	`opensearch/opensearch/config/jvm.options`	`-Xms`, `-Xmx`로 시작하는 라인을 직접 파싱합니다.
Kafka	`kafka/bin/kafka-server-start.sh`	`KAFKA_HEAP_OPTS` 환경 변수 할당 라인을 정규식으로 추출합니다.
Logstash	`logstash/logstash-kafka-os/config/jvm.options`	`-Xms`, `-Xmx` 라인을 직접 파싱합니다.
TACHYON 서비스	`[svc]/bin/tachyon-[svc].sh`	실행 스크립트 내 `java` 명령 인자 중 `-Xms`, `-Xmx` 패턴을 추출합니다.
Watchdog	`bin/tachyon-watchdog.sh`	동일한 쉘 스크립트 정규식 파싱 방식을 사용합니다.

7.2 자동 최적화 (auto) 알고리즘 상세

`--set auto` 옵션 실행 시, 다음 가중치 모델을 기반으로 시스템 총 메모리를 배분합니다.

1. **최적화 공식**: `할당량 = (전체메모리 - 기본 OS 점유분) * 서비스별 가중치`
2. **서비스별 가중치(Weight)**:
    - `opensearch` : 20%
    - `kafka` : 10%
    - `api` : 5%
    - `manager`, `auth`, `stat`, `batch`, `report` : 각 3~5%
    - `watchdog` : 2% (최소 128MB 보장)
3. **제한 정책(Guardrails)**:
    - **최대 상한**: 단일 서비스 힙 크기는 **16GB**를 초과하지 않도록 캡핑(Capping)합니다.
    - **최소 하한**: 서비스 가시성 확보를 위해 최소 **256MB~1GB** (서비스별 상이) 하한선을 적용합니다.
    - **에이전트 수량 가중치**: `--agents` 값이 클수록 API 및 Manager 서비스의 비중을 동적으로 상향 조정합니다.

7.3 안전 관리 프로세스

- **백업 정책**: 설정 변경 전 `${ConfigFile}.bak` 파일을 생성하며, 이미 백업이 존재하는 경우 덮어쓰지 않고 타임스탬프를 활용하여 이력을 보호합니다.
- **리소스 경고**: 총 JVM 할당량이 물리 메모리의 **80%**를 초과할 경우, 운영 안정성을 위해 경고 메시지를 출력하고 수동 검토를 권장합니다.

---

8. OS 메모리 관리 및 점유 방식 기술 명세 (Memory Management)

`tkctl info` 명령어가 리포팅하는 메모리 사용량이 사용자가 체감하는 수치보다 높게 확인되는 현상에 대한 기술적 근거를 설명합니다.

8.1 Linux vs Windows 메모리 관리 기조 차이

항목	Linux (운영 환경)	Windows (비교 환경)
핵심 철학	"사용되지 않는 메모리는 낭비다 (Free RAM is wasted RAM)"	사용자 가시성 중심의 리소스 유연 배분
캐시 활용	디스크 I/O 속도 향상을 위해 남은 모든 메모리를 **Page Cache**로 전용	Superfetch 등을 통한 선독처리를 수행하나 리포팅 시에는 '여유'로 간주하는 경향
사용량 계산	`Used = Total - Free` (Free는 순수 물리 빈 공간)	`Used = Total - Available` (즉시 사용 가능한 캐시 포함 여유분)

8.2 페이지 캐시(Page Cache)와 과점유 현황

1. **과점유 현상 (Memory Occupancy)**:
    - Linux 커널은 파일을 읽거나 쓸 때 해당 데이터를 물리 메모리에 보관(Page Cache)합니다. 이는 이후 동일 파일 접근 시 느린 디스크 대신 빠른 메모리에서 즉시 응답하기 위함입니다.
    - TACHYON 서비스(OpenSearch, MariaDB 등)는 대량의 데이터를 지속적으로 처리하므로 시간이 지날수록 캐시 점유율이 높아져 `Free` 메모리가 0에 가깝게 표시될 수 있습니다.
2. **사용자 가시성**:
    - `tkctl info`는 운영체제 커널의 물리적 점유 상태를 직접 리포팅하므로, 이러한 캐시 영역을 모두 포함하여 '사용 중'으로 표시합니다.
3. **실질 가용성 (Available Memory)**:
    - 이 캐시 영역은 애플리케이션이 실제 메모리 할당을 요청하면 OS가 즉시 비워줄 수 있는 **삭제 가능한(Reclaimable)** 공간입니다. 따라서 점유율 수치가 높더라도 시스템의 실질적인 작업 수행 능력에는 지장을 주지 않습니다.

8.3 결론 및 판정 기준

`tkctl info`에서 확인되는 높은 메모리 점유율은 대부분 **I/O 성능 향상을 위한 Linux 커널의 정상적인 최적화 결과**입니다. 시스템 리소스가 실제로 부족한지 여부는 점유율 수치보다는 다음 지표를 기준으로 판단하십시오.

- **스왑(Swap) 발생 여부**: 물리 메모리가 부족하여 디스크 압축/교환이 빈번하게 일어나는지 확인.
- **OOM Killer 로그**: 커널 로그(`dmesg`)에 메모리 고갈로 인한 프로세스 강제 종료 이력이 있는지 확인.

> **TIP**
>
> Linux 환경에서 실질적으로 '사용 가능한' 양을 알고 싶다면 유휴 공간(Free)이 아닌 가용 공간(Available) 지표를 참조해야 합니다. (tkctl는 향후 버전에서 가용 지표를 포함하도록 개선 예정입니다.)

---

9. 로그 분석 시스템 기술 명세 (analyze)

본 부록은 `tkctl analyze` 명령어가 MariaDB 및 OpenSearch에서 로그 데이터를 분석하기 위해 사용하는 내부 동작 원리와 쿼리 구조를 상세히 기술합니다. 시스템 엔지니어는 본 가이드를 참조하여 `tkctl` 의 분석 결과를 검증하거나, CLI 도구 없이 직접 수동 분석을 수행할 수 있습니다.

9.1 개요 및 데이터 소스

`tkctl analyze` 명령어는 TACHYON 시스템의 두 가지 데이터 소스를 쿼리합니다:

데이터 소스	용도	연결 정보 소스
OpenSearch	실시간 로그 집계, 이상 탐지	`app_info.properties` → `opensearch_ip`, `opensearch_port`, `opensearch_id`, `opensearch_pw`
MariaDB	로그 테이블 증감 추세 분석	`app_info.properties` → `db_ip`, `db_port`, `db_user`, `db_password`

9.2 OpenSearch 인덱스 명명 규칙

TACHYON 시스템은 매체제어 이벤트 로그를 **월별 인덱스**로 분리하여 OpenSearch에 저장합니다.

인덱스 이름 형식:

```
{SPCODE}_{YEAR}_{MONTH}
```

예시:

- 기관 코드(`SPCODE`): `00133fkfkg`
- 2026년 1월: `00133fkfkg_2026_01`
- 2025년 12월: `00133fkfkg_2025_12`

SPCODE 확인 방법:

```
# app_info.properties 파일에서 확인
grep "spcode" /usr/local/TACHYON/TTS40/conf/app_info.properties_dev

# 또는 데이터베이스 이름 확인 (동일한 값)
tkctl info
```

인덱스 목록 조회 (수동):

```
# OpenSearch API로 직접 조회
curl -s "http://localhost:9200/_cat/indices?v" | grep -E "00133fkfkg_"
```

9.3 OpenSearch 필드 매핑 (Field Mapping)

매체제어 로그 인덱스의 주요 필드 구조입니다:

필드명	타입	설명	예시 값
`server_dt`	`keyword`	서버 수신 일자 (YYYY-MM-DD)	`2026-01-15`
`client_dt`	`keyword`	클라이언트 이벤트 발생 일자	`2026-01-15`
`puid`	`keyword`	에이전트 고유 식별자 (UUID)	`a1b2c3d4-e5f6-7890-abcd-ef1234567890`
`process_nm`	`keyword` / `text`	실행 프로세스 경로	`C:\Windows\Explorer.EXE`
`file_nm`	`keyword` / `text`	접근 대상 파일/경로	`E:\Documents\report.xlsx`
`event_type`	`keyword`	이벤트 유형 코드	`READ` , `WRITE` , `COPY`

> **참고**

> `.keyword` 서픽스가 붙은 필드는 정확 일치 검색 및 집계에 사용됩니다. 집계 쿼리에서는 반드시 `process_nm.keyword` 형태로 사용하십시오.

9.4 media-top: 프로세스+경로별 Top N 분석

목적

어떤 프로세스가 어떤 경로에 가장 많이 접근했는지 파악

9.4.1 OpenSearch DSL 쿼리

`tkctl analyze media-top --month 2026-01 --limit 20` 실행 시 내부적으로 다음 DSL을 사용합니다:

```
{
  "size": 0,
  "query": {
    "range": {
      "server_dt": {
        "gte": "2026-01-01",
        "lt": "2026-02-01"
      }
    }
  },
  "aggs": {
    "by_process_path": {
      "composite": {
        "size": 20,
        "sources": [
          { "process": { "terms": { "field": "process_nm.keyword" } } },
          { "path": { "terms": { "field": "file_nm.keyword" } } }
        ]
      }
    }
  }
}
```

9.4.2 수동 실행 방법

```
# OpenSearch에서 직접 쿼리 실행
curl -X POST "http://localhost:9200/00133fkfkg_2026_01/_search" \
```

```
  -H "Content-Type: application/json" \
  -u "admin:admin" \
  -d '{
    "size": 0,
    "query": {
      "range": {
        "server_dt": { "gte": "2026-01-01", "lt": "2026-02-01" }
      }
    },
    "aggs": {
      "by_process_path": {
        "composite": {
          "size": 10,
          "sources": [
            { "process": { "terms": { "field": "process_nm.keyword" } } },
            { "path": { "terms": { "field": "file_nm.keyword" } } }
          ]
        }
      }
    }
  }' | jq '.aggregations.by_process_path.buckets'
```

9.4.3 응답 구조 및 해석

```
{
  "buckets": [
    {
      "key": {
        "process": "C:\\Windows\\Explorer.EXE",
        "path": ""
      },
      "doc_count": 42349
    },
    {
      "key": {
        "process": "C:\\Windows\\system32\\svchost.exe",
        "path": ""
      },
      "doc_count": 5966
    }
  ]
}
```

- `key.process` : 실행 프로세스 경로
- `key.path` : 접근 대상 파일/경로 (빈 문자열은 경로 미지정 이벤트)

- `doc_count` : 해당 조합의 발생 건수

9.5 agent-anomaly: 이상 에이전트 탐지

목적

평균 대비 과도하게 로깅하는 에이전트 식별

9.5.1 OpenSearch DSL 쿼리

```
{
  "size": 0,
  "query": {
    "range": {
      "server_dt": {
        "gte": "2026-01-01",
        "lt": "2026-02-01"
      }
    }
  },
  "aggs": {
    "by_agent": {
      "terms": {
        "field": "puid.keyword",
        "size": 10000
      }
    },
    "agent_stats": {
      "extended_stats_bucket": {
        "buckets_path": "by_agent._count"
      }
    }
  }
}
```

9.5.2 핵심 집계 설명

집계 유형	필드	설명
`terms`	`by_agent`	에이전트(PUID)별 로그 건수 집계
`extended_stats_bucket`	`agent_stats`	버킷 통계 계산 (평균, 표준편차 등)

9.5.3 이상 탐지 알고리즘

`tkctl`는 다음 로직으로 이상 에이전트를 판정합니다:

```
이상 여부 = (에이전트 로그 건수 / 전체 평균) >= 임계치(기본 5.0)
```

판정 기준:

- 🔴 **심각**: 평균 대비 10배 이상 (Multiplier ≥ 10.0)
- ⚠️ **주의**: 평균 대비 5~10배 (Multiplier ≥ 5.0)

9.5.4 수동 분석 절차

```
# 1. 에이전트별 통계 조회
curl -X POST "http://localhost:9200/00133fkfkg_2026_01/_search" \
  -H "Content-Type: application/json" \
  -u "admin:admin" \
  -d '{
    "size": 0,
    "aggs": {
      "by_agent": {
        "terms": { "field": "puid.keyword", "size": 10000 }
      },
      "agent_stats": {
        "extended_stats_bucket": { "buckets_path": "by_agent._count" }
      }
    }
  }' > /tmp/agent_stats.json

# 2. 평균값 확인
jq '.aggregations.agent_stats.avg' /tmp/agent_stats.json

# 3. 표준편차 확인
jq '.aggregations.agent_stats.std_deviation' /tmp/agent_stats.json

# 4. 상위 이상 에이전트 식별 (평균의 5배 이상 필터링)
AVG=$(jq '.aggregations.agent_stats.avg' /tmp/agent_stats.json)
THRESHOLD=$(echo "$AVG * 5" | bc)
jq --argjson th $THRESHOLD \
  '.aggregations.by_agent.buckets | map(select(.doc_count >= $th)) |
sort_by(-.doc_count) | .[0:10]' \
  /tmp/agent_stats.json
```

9.6 process-top: 프로세스별 로그 집계

목적

가장 많은 로그를 발생시키는 프로세스 순위 파악

9.6.1 OpenSearch DSL 쿼리

```
{
  "size": 0,
  "query": {
    "range": {
      "server_dt": {
        "gte": "2026-01-01",
        "lt": "2026-02-01"
      }
    }
  },
  "aggs": {
    "by_process": {
      "terms": {
        "field": "process_nm.keyword",
        "size": 20,
        "order": { "_count": "desc" }
      }
    }
  }
}
```

9.6.2 수동 실행 (OpenSearch Dashboards)

OpenSearch Dashboards에서 다음 시각화를 생성할 수 있습니다:

1. **Visualize** → **Lens** 또는 **Vertical Bar Chart**
2. **Index Pattern**: `00133fkfkg_*`
3. **Metrics**: Count
4. **Buckets**: Terms, `process_nm.keyword`, Top 20

9.7 recommend: 데이터 기반 분석 추천

목적

MariaDB 로그 테이블의 증가율을 분석하여 심층 분석이 필요한 항목 자동 추천

9.7.1 MariaDB 쿼리 구조

현재 월 및 전월 건수 조회:

```
-- 2026년 1월 건수
SELECT COUNT(*)
FROM TPU_ISTAT_MEDIA_EVENT_LOG
WHERE server_dt >= '2026-01-01' AND server_dt < '2026-02-01';

-- 2025년 12월 건수 (전월)
SELECT COUNT(*)
FROM TPU_ISTAT_MEDIA_EVENT_LOG
WHERE server_dt >= '2025-12-01' AND server_dt < '2026-01-01';
```

9.7.2 분석 대상 테이블

테이블명	분석 유형	권장 명령어
`TPU_ISTAT_MEDIA_EVENT_LOG`	매체제어 로그	`tkctl analyze media-top`
`TPU_ISTAT_EVENT_LOG`	이벤트 로그	`tkctl analyze process-top`
`TPU_ISTAT_SYS_LOG`	시스템 로그	`tkctl analyze process-top`

9.7.3 증가율 계산 공식

```
증가율(%) = (현재 월 건수 − 전월 건수) / 전월 건수 × 100
```

9.7.4 우선순위 판정 기준

우선순위	조건	표시 아이콘	권장 조치
1 (폭증)	증가율 ≥ 100%	🔴	즉시 분석 필요
2 (급증)	50% ≤ 증가율 < 100%	⚠️	원인 분석 권장
3 (정상)	증가율 < 50%	✅	정상 범위

9.7.5 수동 분석 스크립트

```
#!/bin/bash
# recommend 분석 수동 수행
```

```
DBNAME="00133FKFKG"
MONTH="2026-01"
PREV_MONTH="2025-12"

# 매체제어 로그 증가율 분석
CURRENT=$(mysql -u tachyon -p'password' $DBNAME -N -e \
  "SELECT COUNT(*) FROM TPU_ISTAT_MEDIA_EVENT_LOG
   WHERE server_dt >= '${MONTH}-01' AND server_dt < DATE_ADD('${MONTH}-01',
INTERVAL 1 MONTH);")

PREVIOUS=$(mysql -u tachyon -p'password' $DBNAME -N -e \
  "SELECT COUNT(*) FROM TPU_ISTAT_MEDIA_EVENT_LOG
   WHERE server_dt >= '${PREV_MONTH}-01' AND server_dt < '${MONTH}-01';")

if [ "$PREVIOUS" -gt 0 ]; then
  RATE=$(echo "scale=1; ($CURRENT - $PREVIOUS) * 100 / $PREVIOUS" | bc)
  echo "매체제어 로그: 현재=${CURRENT}, 전월=${PREVIOUS}, 증가율=${RATE}%"

  if (( $(echo "$RATE >= 100" | bc -l) )); then
    echo "🔴 폭증 - 즉시 분석 필요: tkctl analyze media-top --month $MONTH"
  elif (( $(echo "$RATE >= 50" | bc -l) )); then
    echo "⚠️ 급증 - 원인 분석 권장: tkctl analyze media-top --month $MONTH"
  else
    echo "✅ 정상 범위"
  fi
fi
```

9.8 db-trend: DB 데이터 증가 추세 분석

목적

LOG 테이블의 기간별(1주, 1개월, 1분기, 1년) 누적 증가 추세 파악

9.8.1 MariaDB 쿼리 구조

기간별 누적 건수 조회:

```
-- 1주 전 시점까지의 누적 건수 (Start Total)
SELECT COUNT(*) FROM TPU_ISTAT_MEDIA_EVENT_LOG
WHERE server_dt < DATE_SUB(CURDATE(), INTERVAL 7 DAY);

-- 현재까지의 누적 건수 (End Total)
SELECT COUNT(*) FROM TPU_ISTAT_MEDIA_EVENT_LOG
WHERE server_dt < DATE_ADD(CURDATE(), INTERVAL 1 DAY);
```

9.8.2 증가율 계산

```
기간 증가 건수 = End Total − Start Total
증가율(%) = (End Total − Start Total) / Start Total × 100
```

9.8.3 추세 판정 기준

추세	증가율 조건	표시
폭증 (위험)	≥ 50%	🔴
급증 (주의)	20% ~ 50%	🟠
완만 (정상)	5% ~ 20%	↗
유지 (안정)	0% ~ 5%	→
감소	< 0%	↘

9.8.4 수동 분석 쿼리 예시

```
-- 지난 1주간 증가 추세
SET @start_total = (SELECT COUNT(*) FROM TPU_ISTAT_MEDIA_EVENT_LOG
                    WHERE server_dt < DATE_SUB(CURDATE(), INTERVAL 7 DAY));
SET @end_total = (SELECT COUNT(*) FROM TPU_ISTAT_MEDIA_EVENT_LOG);
SET @increase = @end_total - @start_total;
SET @rate = IF(@start_total > 0, (@increase / @start_total) * 100, 100);

SELECT '최근 1주' AS period,
       @start_total AS start_cnt,
       @end_total AS end_cnt,
       @increase AS increase,
       CONCAT(ROUND(@rate, 1), '%') AS growth_rate,
       CASE
         WHEN @rate >= 50 THEN '🔴 폭증 (위험)'
         WHEN @rate >= 20 THEN '🟠 급증 (주의)'
         WHEN @rate >= 5 THEN '↗ 완만 (정상)'
         WHEN @rate > 0 THEN '→ 유지 (안정)'
         ELSE '↘ 감소'
       END AS trend;
```

9.9 OpenSearch 연결 설정 확인

`tkctl analyze`가 사용하는 OpenSearch 연결 정보를 확인하는 방법입니다:

9.9.1 설정 파일 위치

```
# 기본 설정 파일 (암호화된 상태일 수 있음)
cat /usr/local/TACHYON/TTS40/conf/app_info.properties

# 평문 원본 (운영 환경에 따라 다름)
cat /usr/local/TACHYON/TTS40/conf/app_info.properties_dev*
```

9.9.2 주요 설정 항목

```
opensearch_ip=127.0.0.1
opensearch_port=9200
opensearch_id=admin
opensearch_pw=admin
```

9.9.3 tkctl.ini 개별 설정 (우선순위 높음)

`tkctl.ini` 파일에서 OpenSearch URL을 별도로 지정할 수 있습니다:

```
[opensearch]
url=http://localhost:9200
```

9.10 분석 결과 파일 저장 형식

`--output` 옵션 사용 시 다음 형식으로 저장됩니다:

9.10.1 CSV 형식 (`.csv`)

```
RANK,PROCESS,PATH,COUNT
1,C:\Windows\Explorer.EXE,,42349
2,C:\Windows\system32\svchost.exe,,5966
3,C:\Program Files\Windows Defender\MsMpEng.exe,E:\,2611
```

- **인코딩**: UTF-8 with BOM (Excel 호환)
- **구분자**: 콤마(`,`)

- **줄바꿈**: CRLF

9.10.2 TXT 형식 (`.txt`)

테이블 형태의 텍스트 파일로 저장됩니다.

9.11 트러블슈팅

9.11.1 OpenSearch 연결 실패

```
OpenSearch 호출 실패: dial tcp 127.0.0.1:9200: connect: connection refused
```

원인 및 조치:

1. OpenSearch 서비스 확인: `systemctl status opensearch`
2. 포트 리스닝 확인: `ss -tlnp | grep 9200`
3. 방화벽 확인: `firewall-cmd --list-ports`

9.11.2 인덱스 없음 오류

```
해당 기간에 데이터가 없습니다.
```

원인 및 조치:

1. 인덱스 존재 확인: `curl "http://localhost:9200/_cat/indices?v" | grep {spcode}`
2. 날짜 범위 확인: 지정한 월에 데이터가 있는지 검증
3. SPCODE 확인: 인덱스명의 기관 코드가 올바른지 확인

9.11.3 MariaDB 연결 실패

```
DB 연결 실패: dial tcp 127.0.0.1:3306: connect: connection refused
```

원인 및 조치:

1. MariaDB 서비스 확인: `systemctl status mariadb`
2. 접속 정보 확인: `app_info.properties` 의 `db_ip` , `db_port` , `db_user` , `db_password`

9.12 고급 분석 쿼리 레퍼런스

9.12.1 특정 프로세스 상세 분석

```
# Explorer.EXE의 접근 경로 상세 분석
curl -X POST "http://localhost:9200/00133fkfkg_2026_01/_search" \
  -H "Content-Type: application/json" \
  -d '{
    "size": 0,
    "query": {
      "bool": {
        "must": [
          { "range": { "server_dt": { "gte": "2026-01-01", "lt": "2026-02-
01" } } },
          { "match": { "process_nm": "Explorer.EXE" } }
        ]
      }
    },
    "aggs": {
      "by_path": {
        "terms": { "field": "file_nm.keyword", "size": 50 }
      }
    }
  }' | jq '.aggregations.by_path.buckets'
```

9.12.2 특정 에이전트 이벤트 상세 조회

```
# 이상 에이전트의 최근 이벤트 100건 조회
curl -X POST "http://localhost:9200/00133fkfkg_2026_01/_search" \
  -H "Content-Type: application/json" \
  -d '{
    "size": 100,
    "query": {
      "bool": {
        "must": [
          { "term": { "puid.keyword": "a1b2c3d4-e5f6-7890-abcd-ef1234567890"
} }
        ]
      }
    },
    "sort": [{ "server_dt": "desc" }],
    "_source": ["server_dt", "process_nm", "file_nm", "event_type"]
  }' | jq '.hits.hits[]._source'
```

9.12.3 일별 로그 발생량 추이

```
# 일별 로그 건수 집계
curl -X POST "http://localhost:9200/00133fkfkg_2026_01/_search" \
  -H "Content-Type: application/json" \
  -d '{
    "size": 0,
    "aggs": {
      "by_day": {
        "terms": {
          "field": "server_dt",
          "size": 31,
          "order": { "_key": "asc" }
        }
      }
    }
  }' | jq '.aggregations.by_day.buckets'
```

참고

본 문서에 기재된 쿼리는 TACHYON 4.0 시스템의 기본 스키마를 기준으로 작성되었습니다. 실제 운영 환경에서 필드명이나 인덱스 구조가 다를 수 있으므로, 스키마 확인 후 사용하십시오.

KISA 취약점 분석·평가 구현 명세서 (2026)

이 문서는 **주요정보통신기반시설 기술적 취약점 분석·평가 방법 상세가이드**를 기반으로 `tkctl`가 수행할 보안 진단 및 조치 로직을 **상세 수준(Production-Grade)**으로 정의합니다.

1. 아키텍처 및 데이터 모델

보안 모듈은 확장성을 위해 인터페이스 기반으로 설계됩니다. 모든 진단 결과는 JSON으로 직렬화 가능해야 합니다.

1.1 데이터 구조 (Go Structs)

```
package security

// CheckStatus 진단 상태 상수
type CheckStatus string

const (
    StatusGood  CheckStatus = "GOOD"  // 양호
    StatusBad   CheckStatus = "BAD"   // 취약
    StatusNA    CheckStatus = "NA"    // 해당없음 (서비스 미사용 등)
    StatusError CheckStatus = "ERROR" // 점검 실패 (권한 부족 등)
)

// SecurityItem 진단 항목 메타데이터
type SecurityItem struct {
    Code        string      // 예: "U-01"
    Category    string      // "Linux", "DBMS", "Web"
    Level       string      // "상", "중", "하"
    Name        string      // 항목명 (예: "root 계정 원격 접속 제한")
    Description string      // 상세 설명
    Criteria    string      // 판단 기준
    Guide       string      // 조치 가이드 (텍스트)
}

// CheckResult 진단 결과
type CheckResult struct {
    Item       SecurityItem
    Status     CheckStatus
    CurrentVal string      // 현황 (예: "PermitRootLogin yes")
```

```
    Message    string      // 상세 메시지
    Fixed      bool        // 조치 수행 여부
    FixLog     string      // 조치 결과 로그
}

// Scanner 인터페이스
type Scanner interface {
    Name() string
    Detect(ctx context.Context) ([]CheckResult, error)
    Fix(ctx context.Context, itemCode string) (bool, error)
}
```

---

2. Unix/Linux 서버 점검 (U-01 ~ U-30)

대상: RHEL/CentOS/Rocky, Ubuntu/Debian 계열

코드	항목명	중요도	점검 내용	조치(Fix) 로직
U-01	**root 계정 원격 접속 제한**	상	SSH root 로그인 허용 여부	`sed -i 's/^PermitRootLogin yes/PermitRootLogin no/' /etc/ssh/sshd_config` 후 리로드
U-02	**패스워드 복잡성 설정**	상	영문/숫자/특수문자 조합 및 길이	`/etc/security/pwquality.conf` 의 `minlen=8` , `lcredit=-1` 등 설정 주입
U-03	**계정 잠금 임계값 설정**	상	로그인 실패 시 잠금(5회)	`pam_faillock.so` 또는 `pam_tally2.so` 설정 확인 및 추가
U-04	**패스워드 파일 보호**	상	`/etc/passwd` 권한 확인	`chmod 644 /etc/passwd && chown root:root /etc/passwd`
U-05	**root 홈/패스 디렉터리 권한**	상	PATH에 `.` 포함 여부	사용자 `.bash_profile` 등의 PATH 변수에서 `.` 제거 가이드 (자동수정 위험)
U-06	**파일/디렉터리 소유자 설정**	상	소유자 없는 파일(nouser) 존재	`find / -nouser -o -nogroup` 검색 후 리포팅 (삭제는 수동 권고)
U-07	**passwd 파일 소유자/권한**	상	`/etc/passwd` 권한	U-04와 동일 (중복 항목, 별도 체크)
U-08	**shadow 파일 소유자/권한**	상	`/etc/shadow` 권한	`chmod 400 /etc/shadow && chown root:root /etc/shadow`
U-09	**hosts 파일 소유자/권한**	상	`/etc/hosts` 권한	`chmod 600 /etc/hosts && chown root:root /etc/hosts`
U-10	**xinetd.conf 파일 권한**	상	`/etc/xinetd.conf` 권한	`chmod 600 /etc/xinetd.conf` (파일 존재 시)
U-11	**rsyslog.conf 파일 권한**	상	`/etc/rsyslog.conf` 권한	`chmod 640 /etc/rsyslog.conf`
U-12	**services 파일 권한**	상	`/etc/services` 권한	`chmod 644 /etc/services`
U-13	**SUID, SGID 설정 파일**	상	불필요한 SetUID 파일	주요 파일 외 SUID 제거 (`chmod -s`) 가이드 제공

코드	항목명	중요도	점검 내용	조치(Fix) 로직
U-14	**사용자 시작 파일 권한**	상	`.bashrc` 등 권한	홈 디렉터리 내 설정 파일 권한 640 이하 설정 가이드
U-15	**World Writable 파일**	상	누구나 쓰기 가능 파일	`find / -type f -perm -2` 검색 및 리포팅
U-16	**장치 파일 없는 디렉터리**	상	`/dev` 내 불필요 파일	`/dev` 디렉터리 내 일반 파일 존재 여부 스캔
U-17	**$HOME/.rhosts 권한**	상	r-command 인증 파일	파일 삭제 권고 (`rm .rhosts`)
U-18	**접속 IP 및 포트 제한**	상	`hosts.allow/deny` 설정	`/etc/hosts.deny` 에 `ALL:ALL` 설정 여부 확인
U-19	**Finger 서비스 비활성화**	상	Finger 서비스 동작 여부	`systemctl disable --now finger`
U-20	**Anonymous FTP 비활성화**	상	익명 FTP 접속 허용	`vsftpd.conf` 내 `anonymous_enable=NO` 설정
U-21	**r 계열 서비스 비활성화**	상	rlogin, rsh, rexec	`systemctl disable --now rlogin` 등 관련 서비스 중지
U-22	**Cron 파일 소유자/권한**	상	`/etc/crontab` 등 권한	`chmod 640 /etc/crontab`
U-23	**DoS 공격 취약 서비스**	상	echo, discard 등	`xinetd` 설정에서 해당 서비스 `disable = yes` 설정
U-24	**NFS 서비스 비활성화**	상	불필요한 NFS 구동	미사용 시 `systemctl stop nfs`
U-25	**NFS 접근 통제**	상	`/etc/exports` 설정	`*` (모두 허용) 설정 존재 시 취약
U-26	**automountd 비활성화**	상	자동 마운트 서비스	`systemctl disable --now autofs`
U-27	**RPC 서비스 확인**	상	불필요 RPC 서비스	`rpcinfo -p` 결과 분석

코드	항목명	중요도	점검 내용	조치(Fix) 로직
U-28	**NIS, NIS+ 점검**	상	NIS 서비스 사용 여부	`systemctl disable --now ypserv`
U-29	**tftp, talk 서비스 비활성화**	상	취약 서비스 구동 여부	`systemctl disable --now tftp`
U-30	**Sendmail 버전 점검**	상	최신 버전 사용 여부	패키지 매니저 버전 확인

---

3. DBMS (MariaDB/MySQL) 점검 (D-01 ~ D-10)

연결 방식: `root` 계정으로 로컬/원격 DB 접속 후 SQL 쿼리 실행.

코드	항목명	중요도	점검 쿼리 / 로직	조치(Fix) 가이드
D-01	**기본 계정 패스워드 변경**	상	`root` 패스워드 `NULL` 또는 기본값 확인	`ALTER USER` 구문 제공
D-02	**불필요한 계정 제거**	중	`test`, `guest` 등 기본 계정 존재 확인	`DROP USER 'test'@'%'`
D-03	**원격 접속 제한**	상	`bind-address` 및 `Host='%'` 계정 확인	로컬 바인딩 및 Host 권한 축소
D-04	**패스워드 정책 설정**	상	`validate_password` 플러그인 설정값	`INSTALL PLUGIN validate_password` 가이드
D-05	**로그인 실패 잠금**	상	`connection_control` 플러그인 확인	플러그인 활성화 및 임계값 설정
D-06	**파일 시스템 접근 제한**	상	`local_infile` 변수 확인	`SET GLOBAL local_infile = 0`
D-07	**감사 로그 (Audit) 활성화**	상	`general_log` 또는 Audit 플러그인 확인	감사 로그 활성화 (`my.cnf`)
D-08	**관리자 권한 분리**	상	`SUPER` 권한을 가진 비관리자 계정 확인	불필요한 관리자 권한 회수 (`REVOKE`)
D-09	**세션 타임아웃 설정**	중	`interactive_timeout`, `wait_timeout` 값	타임아웃 값 600초(10분) 이하 설정 권고
D-10	**최신 보안 패치 적용**	상	`SELECT VERSION()` 결과 비교	최신 마이너 버전 업데이트 권고

---

4. Web Server (Nginx) 점검 (N-01 ~ N-10)

대상: 메인 설정(`nginx.conf`) 및 `sites-enabled/*.conf` 파싱.

코드	항목명	중요도	점검 로직	조치(Fix) 설정
N-01	**디렉터리 리스팅 제거**	상	`autoindex on;` 지시어 탐지	`autoindex off;` 변경 또는 삭제
N-02	**불필요한 메소드 제한**	상	`limit_except` 블록 부재 확인	`limit_except GET POST { deny all; }` 추가
N-03	**관리자 페이지 접근 제한**	상	`/admin`, `/manager` 등 경로 ACL 확인	`allow 127.0.0.1; deny all;` 설정
N-04	**숨겨진 파일 접근 제한**	상	`location ~ /\. { deny all; }` 존재 여부	해당 location 블록 추가
N-05	**불필요한 파일 제거**	상	웹 루트 내 `.bak`, `.old`, `.log` 파일 검색	백업 파일 웹 루트 외부로 이동 권고
N-06	**웹 서비스 정보 숨김**	상	`server_tokens on;` 여부 확인	`server_tokens off;` 설정
N-07	**에러 페이지 설정**	중	기본 에러 페이지 노출 여부	`error_page` 지시어로 커스텀 페이지 연결
N-08	**파일 업로드 제한**	상	`client_max_body_size` 값 점검	적절한 용량 제한 설정 (예: 10M)
N-09	**웹 프로세스 권한**	상	`user` 지시어가 `root` 인지 확인	`www-data` 또는 `nginx` 계정으로 실행
N-10	**HTTPS 암호화 통신**	상	SSL/TLS 설정 여부 (`listen 443 ssl`)	인증서 적용 및 80 포트 리다이렉트 권고

---

5. 상세 구현 지침

5.1 스캔 실행 흐름

1. **OS/서비스 감지**: 현재 시스템이 Linux인지, 어떤 서비스(MariaDB, Nginx)가 실행 중인지 탐지.
2. **병렬 스캔**: 가능한 경우 카테고리별 고루틴(Goroutine)을 사용하여 병렬 진단 수행.
3. **결과 집계**: 모든 결과를 `CheckResult` 구조체 리스트로 취합.
4. **리포트 생성**: JSON 파일로 결과 저장 (`scan_report_YYYYMMDD.json`).

5.2 백업 및 롤백 (Safety First)

- **백업**: 설정 파일 수정 전 `cp source dest.bak` 필수 수행.
- **롤백**: 조치 후 서비스 재시작 실패 시, 백업 파일로 자동 원복 기능 구현 (`Rollback()` 메서드).

5.3 CLI 명령어 설계

```
tkctl analyze security              # 전체 진단
tkctl analyze security --target os  # OS(Linux)만 진단
tkctl analyze security --target db  # DB만 진단
tkctl fix security --code U-01      # 특정 항목 조치
tkctl fix security --auto           # 자동 조치 가능한 모든 항목 조치 (주의)
```

이 명세서는 **실 운영 환경(Production)** 수준의 견고함과 **KISA 인증 기준**을 충족하도록 설계되었습니다.

주요정보통신기반시설 기술적 취약점 분석·평가 가이드 (2026)

이 문서는 KISA(한국인터넷진흥원)에서 배포한 **[2026 주요정보통신기반시설 기술적 취약점 분석·평가 방법 상세가이드]**를 기반으로, `tkctl` 프로젝트의 Linux/Unix 서버 보안 진단 기능을 구현하기 위해 작성된 분석 가이드입니다.

1. 개요 및 목적

주요정보통신기반시설의 안정적인 운영을 위협하는 기술적 취약점을 분석하고 평가하기 위한 상세 기준을 정의합니다. `tkctl`는 이 기준을 바탕으로 리눅스 서버의 보안 설정을 자동으로 진단하고, 취약한 항목에 대한 조치 가이드를 제공하는 것을 목표로 합니다.

2. 진단 대상 및 분류 체계 (Unix/Linux 서버)

리눅스 서버의 보안 진단 항목은 크게 5가지 영역으로 분류되며, 각 항목은 `U-XX` (Unix의 약자) 형식의 코드로 식별됩니다.

대분류	항목 코드 범위	주요 점검 내용
1. 계정 관리	U-01 ~ U-04	root 원격 접속 제한, 패스워드 복잡성, 계정 잠금, 파일 접근 권한 등
2. 파일 및 디렉터리 관리	U-05 ~ U-18	중요 설정 파일(/etc/passwd, /etc/shadow 등)의 소유자 및 권한 관리
3. 서비스 관리	U-19 ~ U-34	불필요한 서비스(Finger, R-commands 등) 제거 및 데몬 설정 관리
4. 패치 관리	U-35 ~ U-36	최신 보안 패치 적용 여부 확인
5. 로그 관리	U-37 ~ U-XX	정기적인 로그 검토 및 로깅 설정 확인

3. 주요 점검 항목 상세 (U-01 ~ U-10)

`tkctl` 개발 시 최우선으로 구현되어야 할 상위 10개 핵심 항목에 대한 분석 내용입니다.

U-01. root 계정 원격 접속 제한

- **중요도**: 상
- **점검 내용**: 원격 터미널 서비스(SSH, Telnet 등)를 통한 root 계정의 직접 접속 허용 여부 점검
- **판단 기준**:
  - **양호**: 원격 터미널에서 root 계정 접속이 차단된 경우
  - **취약**: 원격 터미널에서 root 계정 접속이 허용된 경우
- **구현 가이드**:
  - `/etc/ssh/sshd_config` 파일 파싱
  - `PermitRootLogin` 값이 `no`로 설정되어 있는지 확인 (주석 처리된 경우 기본값 확인 필요)

U-02. 패스워드 복잡성 설정

- **중요도**: 상
- **점검 내용**: 시스템 패스워드 정책이 영문, 숫자, 특수문자 등을 조합하여 설정되어 있는지 점검
- **판단 기준**:
  - **양호**: 영문/숫자/특수문자 3종류 조합 시 8자리 이상, 2종류 조합 시 10자리 이상 설정된 경우
  - **취약**: 패스워드 복잡성 설정이 미흡한 경우
- **구현 가이드**:
  - `/etc/security/pwquality.conf` 또는 `/etc/pam.d/system-auth` (RHEL 계열), `/etc/pam.d/common-password` (Debian 계열) 확인
  - `minlen`, `lcredit`, `ucredit`, `dcredit`, `ocredit` 옵션 검사

U-03. 계정 잠금 임계값 설정

- **중요도**: 상
- **점검 내용**: 일정 횟수 이상 로그인 실패 시 계정을 잠그는 정책 설정 여부 점검
- **판단 기준**:
  - **양호**: 로그인 실패 5회 이하 설정 시 계정 잠금 설정이 적용된 경우
  - **취약**: 계정 잠금 정책이 설정되어 있지 않거나 기준을 초과한 경우
- **구현 가이드**:
  - `/etc/pam.d/system-auth` 또는 `/etc/pam.d/password-auth` 확인
  - `pam_faillock.so` 또는 `pam_tally2.so` 모듈의 `deny` 옵션 값 검사

U-04. 패스워드 파일 보호

- **중요도**: 상
- **점검 내용**: `/etc/passwd` 파일의 소유자 및 권한 설정 점검
- **판단 기준**:
  - **양호**: 소유자가 root이고, 권한이 644 이하인 경우
  - **취약**: 소유자가 root가 아니거나, 권한이 644보다 높은 경우
- **구현 가이드**:
  - `stat` 시스템 콜 또는 명령어를 사용하여 `/etc/passwd`의 uid(0)와 permission 확인

U-05. 패스워드 정책 파일 보호 (/etc/shadow)

- **중요도**: 상
- **점검 내용**: 패스워드 암호화 파일(`/etc/shadow`)의 소유자 및 권한 설정 점검
- **판단 기준**:
  - **양호**: 소유자가 root이고, 권한이 400 이하인 경우
  - **취약**: 소유자가 root가 아니거나, 권한이 400보다 높은 경우
- **구현 가이드**:
  - `/etc/shadow` 파일의 uid(0)와 permission(400 또는 000) 확인

U-06. 호스트 등가성 설정 파일 권한

- **중요도**: 상
- **점검 내용**: `/etc/hosts.equiv` , `$HOME/.rhosts` 파일의 소유자 및 권한 점검
- **판단 기준**:
  - **양호**: 해당 파일이 없거나, 소유자가 root(또는 계정 소유자)이고 권한이 600 이하인 경우
  - **취약**: 해당 파일의 권한이 600보다 높거나 소유자가 잘못 설정된 경우
- **구현 가이드**:
  - 파일 존재 여부 확인 후 `stat` 검사. 없을 경우 '양호' 판정.

U-07. syslog 설정 파일 권한

- **중요도**: 상
- **점검 내용**: `/etc/syslog.conf` 또는 `/etc/rsyslog.conf` 파일의 소유자 및 권한 점검
- **판단 기준**:
  - **양호**: 소유자가 root이고, 권한이 644 이하인 경우
  - **취약**: 소유자가 root가 아니거나, 권한이 644보다 높은 경우
- **구현 가이드**:
  - OS 버전에 따라 `rsyslog.conf` 파일 확인

U-08. 서비스 설정 파일 권한 (/etc/services)

- **중요도**: 상
- **점검 내용**: `/etc/services` 파일의 소유자 및 권한 점검
- **판단 기준**:
  - **양호**: 소유자가 root이고, 권한이 644 이하인 경우
  - **취약**: 소유자가 root가 아니거나, 권한이 644보다 높은 경우

U-09. inetd 설정 파일 권한

- **중요도**: 상
- **점검 내용**: `/etc/inetd.conf` 또는 `/etc/xinetd.conf` 및 `/etc/xinetd.d/*` 파일 권한 점검
- **판단 기준**:
  - **양호**: 소유자가 root이고, 권한이 600 이하인 경우
  - **취약**: 권한이 취약하게 설정된 경우

U-10. 공통 시작스크립트 권한

- **중요도**: 상
- **점검 내용**: `/etc/rc.d/init.d/` 내의 스크립트 파일 권한 점검
- **판단 기준**:

- **양호**: 소유자가 root이고, 권한이 755 이하인 경우
- **취약**: 소유자가 root가 아니거나, 권한이 755보다 높은 경우

4. tkctl 구현 시 고려사항

1. **OS 호환성**:
   - RHEL(CentOS, Rocky), Debian(Ubuntu) 계열의 설정 파일 경로 차이를 고려해야 합니다. (예: `pam` 설정 경로)
   - `tkctl` 내부 로직에서 OS 종류를 먼저 식별(`ProcessOSProbe`)한 후 분기 처리해야 합니다.
2. **진단 로직**:
   - **단순 권한 체크**: `os.Stat` 을 이용한 파일 모드 및 소유자(Uid) 검사.
   - **내용 파싱**: `bufio.Scanner` 등을 이용하여 설정 파일의 키-값 쌍 파싱. 주석(`#`) 처리된 라인은 무시해야 합니다.
3. **결과 리포팅**:
   - 진단 결과는 `Code` , `Name` , `Status` (양호/취약/N/A), `Detail` (현황 값) 필드를 포함해야 합니다.
   - JSON 포맷으로 출력하여 시각화 도구와 연동 가능하도록 설계해야 합니다.