Notice: This document contains information on products in the design phase of development. The information here is subject to change without notice. Do not finalize a design with this information.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software, or service activation. Learn more at intel.com, or from the OEM or retailer.

The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest Intel product specifications and roadmaps.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. No component or product can be absolutely secure. Check with your system manufacturer or retailer or learn more at http://intel.com.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

Warning: Altering PC clock or memory frequency and/or voltage may (i) reduce system stability and use life of the system, memory and processor; (ii) cause the processor and other system components to fail; (iii) cause reductions in system performance; (iv) cause additional heat or other damage; and (v) affect system data integrity. Intel assumes no responsibility that the memory, included if used with altered clock frequencies and/or voltages, will be fit for any particular purpose. Check with memory manufacturer for warranty and additional details.

Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. Consult other sources of information to evaluate performance as you consider your purchase. For more complete information about performance and benchmark results, visit http://www.intel.com/performance.

Cost reduction scenarios described are intended as examples of how a given Intel-based product, in the specified circumstances and configurations, may affect future costs and provide cost savings. Circumstances will vary. Intel does not guarantee any costs or cost reduction.

Results have been estimated or simulated using internal Intel analysis or architecture simulation or modeling, and provided to you for informational purposes. Any differences in your system hardware, software or configuration may affect your actual performance.

Intel does not control or audit third-party benchmark data or the web sites referenced in this document. You should visit the referenced web site and confirm whether referenced data are accurate.

Intel is a sponsor and member of the Benchmark XPRT Development Community, and was the major developer of the XPRT family of benchmarks. Principled Technologies is the publisher of the XPRT family of benchmarks. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases.

Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or by visiting www.intel.com/design/literature.htm.

Intel, the Intel logo, Intel TXT, and Xeon are trademarks of Intel Corporation in the U.S. and/or other countries.

*Other names and brands may be claimed as the property of others.

Copyright © 2018, Intel Corporation. All Rights Reserved.
Table of Contents

Introduction .................................................................................................................................................. 6

Intel® SecL-DC Threat Detection Technology Overview ................................................................. 6

Heuristics Models.................................................................................................................................. 6

Components and Deployment Architecture.......................................................................................... 6

Threat Detection Service....................................................................................................................... 6

Threat Detection Agent.......................................................................................................................... 6

Installation .................................................................................................................................................... 7

Installing the Threat Detection Service ............................................................................................... 7

Supported Operating Systems .............................................................................................................. 7

Recommended Hardware ....................................................................................................................... 7

Installation ............................................................................................................................................ 7

Installing the Threat Detection Agent.................................................................................................... 8

Supported Operating Systems .............................................................................................................. 8

Recommended Hardware ....................................................................................................................... 8

Prerequisites ......................................................................................................................................... 8

Installation ............................................................................................................................................ 8

Heuristics Plugins .................................................................................................................................. 8

Threat Detection Service Management .................................................................................................. 9

List Registered Agents........................................................................................................................... 9

Threat Reports ...................................................................................................................................... 9

Agent Registration ............................................................................................................................... 10

Delete Registered Agent ....................................................................................................................... 11

Threat Detection Configuration Settings .............................................................................................. 11

Logging .................................................................................................................................................. 11

Threat Detection Service....................................................................................................................... 11

Folder Layout ...................................................................................................................................... 11

Installation Options ............................................................................................................................. 11

Configuration Options ........................................................................................................................ 12

Threat Detection Agent ........................................................................................................................ 14

Folder Layout ...................................................................................................................................... 14

Installation Options ............................................................................................................................. 14

Configuration Options ........................................................................................................................ 15
Introduction

Intel® SecL-DC Threat Detection Technology Overview

Intel® SecL-DC TDT (“TDT”) provides run-time detection of active attacks based on CPU and platform telemetry. TDT uses pre-generated heuristics models to identify platform usage indicative of an in-progress attack, and provides a centralized management service to manage all servers enabled with Intel SecL-DC TDT.

Each enabled server in a datacenter will have an installed Threat Detection Agent (“Agent”), which contains the actual TDT runtime and heuristics plugins for each supported threat category. The Agent manages the configuration settings for the Threat Detection runtime, and allows communication to the Threat Detection Service (“Service”).

The Service provides a centralized management plane for all Agent instances. The Service maintains a registry of all managed Agents, and maintains a heartbeat with registered Agents to ensure connectivity. When an Agent detects an active threat, it will push the detected threat information to the Service for centralized collection and dissemination.

Heuristics Models

Intel® SecL-DC TDT uses heuristics models to analyze runtime platform telemetry for supported threat identification. Detection of a new threat type requires a new heuristic model to be installed. Intel demonstrated runtime threat detection utilizing the TDT framework at the RSA Conference 2019 with AI-ML based heuristics that use CPU telemetry as features to profile the execution of specific exploits. We are encouraging the developer community to take advantage of the Intel® SecL-DC TDT framework to develop their own threat detection heuristics. Please consult the developer guide distributed with Intel® SecL-DC TDT source for more information.

Components and Deployment Architecture

Intel® SecL-DC TDT uses a client/server model, with a centralized management service (the Service) that manages all registered clients and allows centralized reporting of detected threats, and a client (the Agent) installed on each enabled physical server. The Agent performs the actual runtime analysis of server telemetry for threats and reports any events to the Service.

Threat Detection Service

The Intel® SecL-DC Threat Detection Service is a management-plane service that acts as a single management point for all servers using Intel® SecL-DC TDT. All Agents are registered to this service for monitoring. All reports of detected threats will be pushed to this service.

Threat Detection Agent

The Intel® SecL-DC Threat Detection Agent is a lightweight application that runs on each managed server. This agent contains the actual Threat Detection engine, and will use heuristics plugins to add new threats to the detection profile. The agent pushes reports of any detected threats to the Threat Detection Service it is registered to.
Installation

Installing the Threat Detection Service

Supported Operating Systems
The Threat Detection Service supports RedHat Enterprise Linux 7.6.

Recommended Hardware
The Threat Detection Service is installed as a lightweight application that can be run on a baremetal server or a virtual machine.

- 1GB RAM
- 256MB Disk space
- Network access: The Threat Detection Service requires access to a network that can access all hosts that will be registered to the Service.

Installation

Prerequisites

- A Postgres database server is required. This database server may be installed on the same server as the Threat Detection Service, or may be located elsewhere on the network. An empty database schema must be created, and a user with full privileges to that schema must be provided to the Threat Detection Service installer via the `tdservice.env` installation answer file. If an existing Postgres database server will be used, md5 auth must be enabled in the `pg_hba.conf` configuration.

- A script is provided to automatically install a local postgres database on the same system that will run the Threat Detection Service. This script is named `install_pgdb.sh`, and will install relevant Postgres repositories and packages with the required settings, and will create the needed database and user.

Installation

Create the `tdservice.env` answer file. This file will contain values used during installation. A sample answer file is provided below.

```
TDS_DB_HOSTNAME=127.0.0.1
TDS_DB_PORT=5432
TDS_DB_USERNAME=<database username>
TDS_DB_PASSWORD=<database password>
TDS_DB_NAME=pgdb
TDS_ADMIN_USERNAME=<username>
TDS_ADMIN_PASSWORD=<password>
TDS_REG_HOST_USERNAME=<username>
TDS_REG_HOST_PASSWORD=<password>
```

Install the database

This step is not required if a database is already available.

Execute `install_pgdb.sh` script.

```
./install_pgdb.sh
```
Install the Threat Detection Service
Execute the Threat Detection Service self-extracting installer.

./tdservice-v1.0.bin

Installing the Threat Detection Agent
Supported Operating Systems
The Threat Detection Agent supports RedHat Enterprise Linux 7.6.

Recommended Hardware
The Threat Detection Agent must be installed on a physical server and should not be installed in a virtual machine.

- 64MB RAM
- 64MB Disk space
- Network access: the Threat Detection Agent requires access to a network that can also reach the Threat Detection Service to which it will register and send Reports.

Prerequisites
- The Threat Detection Service must be installed and running

Installation
1. Create the tdagent.env installation answer file:
   
   export TDS_SERVICE_BASEURL=https://tdservice.server.com:8443/tds/
   export TDS_SERVICE_USERNAME=<TDT Service registration username>
   export TDS_SERVICE_PASSWORD=<TDT Service registration password>

2. Install TDA (This will auto register TD Agent with the TD Service)
   
   #./tdagent-v1.0.bin

3. Check agent status
   
   #tdagent status

Assuming all variables were provided in the tdagent.env answer file, the Agent will automatically register itself with the Threat Detection Service at the end of the installation.

Heuristics Plugins
The Intel® Secl-DC TDT software is provided as Open Source software. However, individual heuristics models and profiles may be licensed separately.

Where to Get New Heuristics Profiles
The Intel Secl-TDT heuristics models are added as separate plugins with separate licensing from the rest of Intel® Secl-DC TDT. These plugins can be added either at compilation time (so that they are included with the Agent installer files), or after installation. The plugins need to be downloaded from Intel® separately from the actual software build/installation processes. Generally this should be done before compiling the Intel® Secl-DC TDT Agent from source, so that the models can be included in the Agent installer (see the Building from Source section for instructions on where to place the heuristics plugins).
Each heuristic model will be comprised of several .so binaries along with a .profile configuration file.

To obtain available heuristics plugins, contact tdt.ae@intel.com.

Adding a New Heuristic Profile to the TDT Agent

Place the .so files in /var/lib/tdagent. .profile file handles configuration.

The command “tdagent setup tdtprofile” can be used to set the profile setting. Alternatively, edit the profile in config.yml to set the new profile. This value should be the name of the profile file, without the “.profile” portion.

Restart agent service for the profile change to take effect.

Threat Detection Service Management

List Registered Agents

List all registered TD agents:

```
GET https://tdservice.server.com:8443/tds/hosts
Status: 200 OK
Response: 
[
  
  {
    "id": "ad6af677-0d9b-4979-8292-563c0e7df050",
    "hostname": "10.105.167.145",
    "version": "v0.2.0",
    "build": "68aba41",
    "os": "linux",
    "status": "online"
  }
]
```

The “status” response refers to a feature that will be implemented in a future release.

Available Query parameters:

hostname=(hostname)
version=(version)
build=(build)
os=(os)
status=(status) (not currently applicable, future feature)

Query parameters can be conjoined in any combination, so for example: GET /tds/hosts?os=linux&status=offline

Threat Reports

Search for Reports:

```
GET https://tdservice.server.com:8443/tds/reports
Status: 200 OK
Example Response : 
[
  
  {
    "id": "337861de-c542-4fe3-b3c7-cead2b6bd4e1",
    "host_id": "ad6af677-0d9b-4979-8292-563c0e7df050",
    "detection": {
      "pid": 38329,
      "tid": 38329,
```
"process_name": "<processname>",
"process_path": "/root/<process>",
"message": "attack suspected!",
"timestamp": 1552104354,
"severity": 0.989374,
"profile_description": "Detects <attack type>",
"profile_name": "<TDT heuristic lib name>",
"profile_author": "Intel (C) Corp.",
"profile_date": "2019-02-16",
"plugin_origin": "<plugin>",
"last_n_detections": 100,
"avg_severity_of_last_n_detections": 0.954999,
"cve_ids": [
  {
    "id": "CVE-...",
    "description": "<description>"
  },
  {
    "id": "CVE-...",
    "description": "<description>"
  }
]

Available Query parameters (if no parameters are specified, all reports will be returned):

hostname=(hostname)
hostid=(host uuid)
from=(from_date)
to=(to_date)

Note that Reports will only exist if a threat has been detected by one or more Agents. If no Reports are returned, this indicates that no active threats have been detected.

Agent Registration

The Agent is automatically registered to the Service during installation by default. This requires the tdagent.env file to be populated with the TDA_SERVICE_BASEURL, TDA_SERVICE_USERNAME, and TDA_SERVICE_PASSWORD variables before installation. Registration during installation is the recommended method for Agent registration.

If registration needs to be performed after installation, use the CLI command from the Agent (again, make sure the tdagent.env file is present with the required variables defined):

tdagent setup register

Alternatively, the Agent can be registered via REST API. This method is not recommended as it requires an additional step to modify the Agent configuration file. To register via REST API:

POST https://tdservice.server.com/tds/hosts

{
  "hostname": "hostname",
  "version": "v0.2.0",
  "build": "68aba41",
  "os": "linux"
}

The response will contain a host ID, username, and a token. These values must be added to the Agent config.yml file (this step is not needed if using the CLI command, or if registering during installation):

tds:
  baseurl: https://tds.server.com:8443/tds/
username: d4690700-d7bc-4652-8ce4-bddcfbde8266

token: u4Vj79GDM0J9aSPyTG/mgDIdE=

hostid: caf9c9fa-20f3-49e2-84fc-c93bd3922045

After updating the file, stop and start the Service:

```
tdservice stop
tdservice start
```

**Delete Registered Agent**

Delete a specified TD agent:

```
DELETE https://tdservice.server.com:8443/tds/hosts/{hostID}
```

Status: 204 no content

---

**Threat Detection Configuration Settings**

**Logging**

The Threat Detection Service and Agent use systemd logging. To access the logs, use journalctl. For example:

```
journalctl -u tdservice
```

or

```
journalctl -u tdagent
```

The logging level can be changed in the `config.yml` configuration file, by changing the “loglevel” value to a different logging level (such as “debug”).

**Threat Detection Service**

**Folder Layout**

```
/etc/tdservice
    cert.pem (Signing Key Cert)
    config.yml(System_configuration)
    key.pem (Signing Key)
/var/log/tdservice
    http.log (http log)
    tdservice.log (main log file)
/opt/tdservice/bin
    tdservice (executable binary)
```

**Installation Options**

The Service can be configured at install time using an installation answer file named “tdservice.env”.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Example Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TDS_DB_HOSTNAME</td>
<td>postgres.server.com</td>
<td>Hostname of the database server</td>
</tr>
<tr>
<td>TDS_DB_PORT</td>
<td>5500</td>
<td>Port number of the database server</td>
</tr>
<tr>
<td>TDS_DB_USERNAME</td>
<td>dbuser</td>
<td>Username for access to the database</td>
</tr>
<tr>
<td>TDS_DB_PASSWORD</td>
<td>password</td>
<td>Password for access to the database</td>
</tr>
</tbody>
</table>
TDS_DB_NAME | tddb | Database schema name
--- | --- | ---
TDS_ADMIN_USERNAME | tdsadmin | API user with Administrator privileges. This is the admin user used for REST API calls to the TD Service
TDS_ADMIN_PASSWORD | password | Password for the API user with Administrator privileges. This is the admin user used for REST API calls to the TD Service
TDS_REG_HOST_USERNAME | registraruser | User with permission only to register new Agents to the Service. Intel® recommends using this user for Agent installation/registration rather than the administrative user.
TDS_REG_HOST_PASSWORD | Password | Password for the registration user.

Configuration Options

System Configuration

The Service configuration contains database connection settings and the logging level. These settings are stored in /etc/tdservice/config.yml.

Sample config.yml file:

```yaml
port: 8443
postgres:
  dbname: tddb
  username: runner
  password: test
  hostname: tds-pg-db
  port: 5432
  sslmode: false
loglevel: error
```

<table>
<thead>
<tr>
<th>Variable</th>
<th>Example Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>port</td>
<td>8443</td>
<td>Threat Detection Service webservice port</td>
</tr>
<tr>
<td>postgres: dbname</td>
<td>tddb</td>
<td>Database schema name</td>
</tr>
<tr>
<td>postgres: username</td>
<td>dbuser</td>
<td>Database user</td>
</tr>
<tr>
<td>postgres: password</td>
<td>password</td>
<td>Database password</td>
</tr>
<tr>
<td>postgres: hostname</td>
<td>database.server.com</td>
<td>Database hostname or IP</td>
</tr>
<tr>
<td>postgres: port</td>
<td>5432</td>
<td>Database port</td>
</tr>
<tr>
<td>postgres: sslmode</td>
<td>false</td>
<td>Currently sslmode is not supported; this will be added in a future release. For now this must be set to false.</td>
</tr>
<tr>
<td>loglevel</td>
<td>error</td>
<td>Defines the log level for the tdservice.log file</td>
</tr>
</tbody>
</table>
Command Line Operations

Setup
> tdservice setup
   Available setup tasks:
   - database
   - admin
   - server
   - tls
   ---------------------
   - [all]

Setup - Database
> tdservice setup database [-force] --db-host=postgres.com --db-port=5432 --db-username=admin --db-password=password --db-name=tds_db

Environment variables TDS_DB_HOSTNAME, TDS_DB_PORT, TDS_DB_USERNAME, TDS_DB_PASSWORD, TDS_DB_NAME can be used instead of command line flags

Setup - HTTP Server
> tdservice setup server --port=8443

Environment variable TDS_PORT can be used instead of command line flags

Setup - TLS
> tdservice setup tls [-force] [--host-names=intel.com,10.1.168.2]

Creates a Self Signed TLS Keypair in /etc/tdservice/ for quality of life. It is expected that consumers of this product will provide their own key and certificate in /etc/threat-detection before or after running setup, to make TDA use those instead.

Environment variable TDS_TLS_HOST_NAMES can be used instead of command line flags

--force overwrites any existing files, and will always generate a self signed pair.

Setup - Admin
> tdservice setup admin --admin-user=admin --admin-pass=password

Environment variable TDS_ADMIN_USERNAME and TDS_ADMIN_PASSWORD can be used instead

This task can be used to create multiple admin-users, but if a duplicate username is specified it will error out.

Status
> tdservice status
Forwarding to "systemctl status tdservice"

Start/Stop
> tdservice start
   Threat Detection Service started
> tdservice stop
   Threat Detection Service stopped
Uninstall
> tdservice uninstall [--keep-config]
  Threat Detection Service uninstalled

Uninstalls Threat Detection Service, with optional flag to keep configuration

Help
> tdservice {help|-h|-help}
  Usage: tdservice <command> <flags>
    Commands:
    - setup
    - help
    - start
    - stop
    - status
    - uninstall
    - version

Version
> tdservice version
  Threat Detection Service v1.0.0 build 9cf83e2

Threat Detection Agent
Folder Layout

The daemon will create and use the following files on the OS:

/etc/tdagent
  config.yml (System Configuration)
/var/lib/tdagent
  Heuristic profile binaries (.so files)
  Heuristic profile configuration files (.profile files)
/var/log/tdagent
  http.log (http log file)
  tdagent.log (main log file)
/usr/bin/tdagent (executable binary)

Installation Options
The Agent uses an installation answer file (tdagent.env) to configure options at installation time.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Example Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TDS_SERVICE_BASEURL</td>
<td><a href="https://tdservice.server.com:8443/tds/">https://tdservice.server.com:8443/tds/</a></td>
<td>URL for the TD Service. This defines the TD Service the Agent will try to register with.</td>
</tr>
<tr>
<td>TDS_SERVICE_USERNAME</td>
<td>tdsregistration</td>
<td>Registration credentials for the TD Service</td>
</tr>
<tr>
<td>TDS_SERVICE_PASSWORD</td>
<td>password</td>
<td>Registration credentials for the TD Service</td>
</tr>
<tr>
<td>TDT_PROFILE</td>
<td>telemetry_collect_tdtlib</td>
<td>Selects the heuristic model profile to use. This can be changed after installation in the config.yml configuration file.</td>
</tr>
</tbody>
</table>
Configuration Options

System configuration

The Agent configuration contains database connection settings and the logging level. These settings are stored in `/etc/tdagent/config.yml`.

<table>
<thead>
<tr>
<th>Port:</th>
<th>Port:</th>
</tr>
</thead>
<tbody>
<tr>
<td>tds: baseurl</td>
<td>tds: baseurl</td>
</tr>
<tr>
<td>tds: username</td>
<td>tds: username</td>
</tr>
<tr>
<td>tds: password</td>
<td>tds: password</td>
</tr>
<tr>
<td>admin: username</td>
<td>admin: username</td>
</tr>
<tr>
<td>admin:token</td>
<td>Base64 encoded form of the token byte array returned during host registration</td>
</tr>
<tr>
<td>hostid</td>
<td>3812936d-db1e-4705-a7f4-5345849c0a6f</td>
</tr>
<tr>
<td>loglevel</td>
<td>loglevel</td>
</tr>
<tr>
<td>insecureSkipVerify</td>
<td>false</td>
</tr>
<tr>
<td>TDTProfile</td>
<td>telemetry_collect_tdtlib</td>
</tr>
<tr>
<td>TDTProfile</td>
<td>Determines the heuristic profile that will be enabled for this Agent. Only one profile may be enabled at a time. The profile names is the same as the “.profile” configuration file for that heuristic model in <code>/var/lib/tdagent/</code>, without the .profile extension.</td>
</tr>
</tbody>
</table>

Profile Configuration

The TDT Agent configuration contains a selection of which heuristic profile to use. The Agent may be configured to use only one profile at a time. The currently supported profiles are the `telemetry_collect_tdtlib`, which gathers platform telemetry but contains no actual heuristic for detection or reporting (included by default). The profile selection is made either by inclusion of the `TDT_PROFILE` environment variable in the `tdagent.env` installation answer file at install time, or in the `TDTProfile` variable in `config.yml` after the agent is installed.

The command “tdagent setup tdtprofile” can be used to set the profile setting.

Threat Detection Heuristic Configuration

The actual Threat Detection heuristics used inside of the Agent are configured separately, in a `.profile` file. For example, the telemetry heuristic is configured in `/var/lib/tdagent/telemetry_collect_tdtlib.profile`. This allows configuration of telemetry sampling frequency, as well as confidence thresholds for triggering a detection report. Each heuristic profile contains comments indicating what each setting does so that they can be configured as desired.

Command Line Operations

Help

> tdagent help

Usage:

```
tdagent <command> [arguments]
```

Available Commands:

- `help`-h|--help: Show this help message
- `setup` [task]: Run setup task
- `start`: Start tdagent
- `stop`: Stop tdagent
uninstall        Uninstall tdagent

Available Tasks for setup:

`tdagent setup tls [--force] [--hosts=<host_names>]`
  - Use the key and certificate provided in /etc/threat-detection if files exist
  - Otherwise create its own self-signed TLS keypair in /etc/tdagent for quality of life
  - Option [--force] overwrites any existing files, and always generate self-signed keypair
  - Argument <host_names> is a list of host names used by this device, separated by comma
  - Environment variable TDA_TLS_HOSTS=<host_names> can be set alternatively

`tdagent setup signingkey [--force]`
  - Creates a Self Signed Signing Key in /etc/tdagent for quality of life
  - Option [--force] overwrites any existing files, and always generate self-signed keypair
  - The signing key is expected to be provided in /etc/threat-detection by the user

`tdagent setup register --tds-url=<api_gateway> --tds-user=<username> --tds-pass=<password>`
  - Setup the connection to the TDS as well as registers this TDA with the TDS
  - <api_gateway> should be in format: https://<hostname>/tds/
  - Environment variable TDS_SERVICE_BASEURL=<api_gateway> can be set alternatively
  - Environment variable TDS_SERVICE_USERNAME=<username> can be set alternatively
  - Environment variable TDS_SERVICE_PASSWORD=<password> can be set alternatively

`tdagent setup tdtprofile --profile=<tdt_profile_name>`
  - Set up the profile to be used eg: telemetry
  - <tdt_profile_name> should be the name of the file without .profile extension
  - Environment variable TDT_PROFILE=<tdt_profile_name> can be set alternatively

`tdagent setup all`
  - Setup all previous options in listed order

Setup

> tdagent setup
  Available setup tasks:
  - server
  - tls
  - signingkey
  - register

Setup - HTTP Server

> tdagent setup server --port=8443]

Setup - tdtprofile

> tdagent setup tdtprofile --profile=<tdt_profile_name>

Updates the TDT Agent Profile setting to use a different heuristic profile. Environment variable TDT_PROFILE=<tdt_profile_name> can be set alternatively

Setup - Register

> tdagent setup register --tds-url=https://api.gateway/tds/ --tds-user=admin --tds-pass=password

Sets up the connection to the TDS, as well as registers this TDA with the TDS by making a POST request to /tds/hosts, sending it's host information as well as TDT capabilities from `agent::discover()`

The setup task can alternatively read variables from the environment, such as:

export TDA_SERVICE_BASEURL=https://api.gateway/tds/
export TDA_SERVICE_USERNAME=admin
export TDA_SERVICE_PASSWORD=password
See the Installation Options section on the `tdagent.env` answer file for all possible variables.

Running `tdagent setup all` will run all setup tasks in order, picking arguments from environment variables, then command line.

**Start/Stop**

> `tdagent start`
> Forwarding to "`systemctl start tdagent`"
> `tdagent stop`
> Forwarding to "`systemctl stop tdagent`"

**Status**

`tdagent status`
Forwarding to "`systemctl status tdagent`"

**Uninstall**

> `tdagent uninstall`
Threat Detection Agent uninstalled