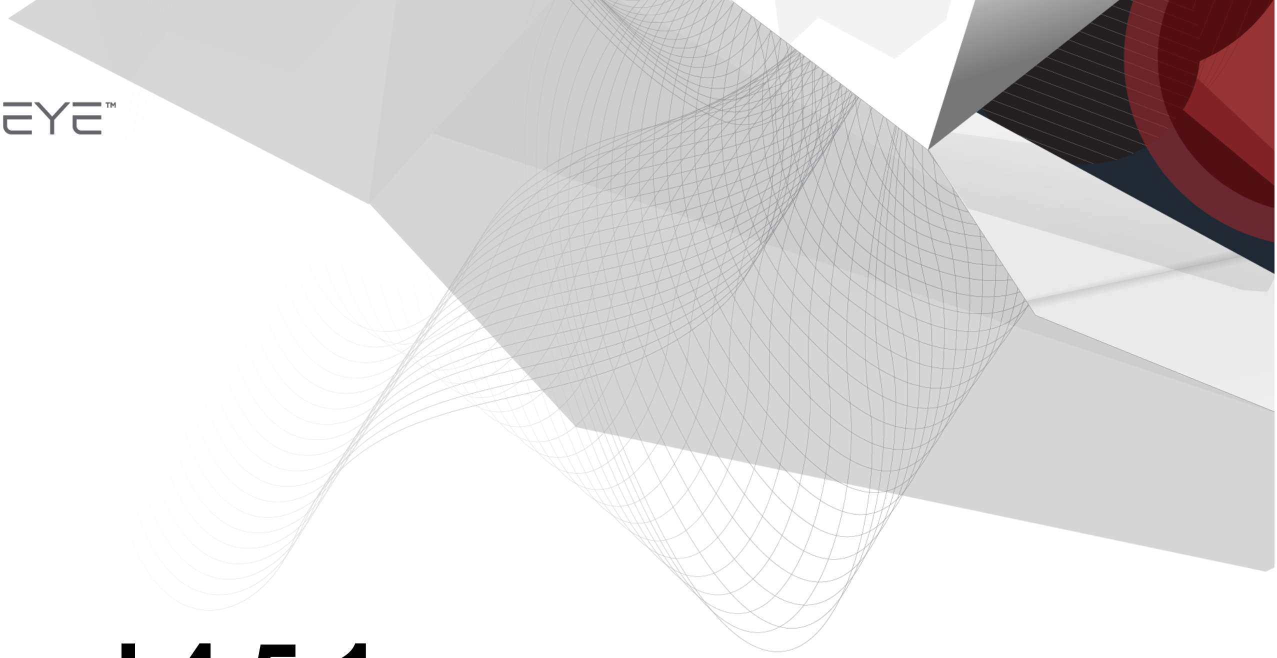

HXTool 4.5.1

Release notes

FireEye, Inc. | 601 McCarthy Blvd. Milpitas, CA 95035 | 408.321.6300 | 877.FIREEYE (347.3393) | info@fireeye.com | www.fireeye.com

HXTOOL 4.5.1 RELEASE NOTES

NEW FEATURES

- X15 streaming support for bulk acquisitions
- Linux as a platform under indicator creation
- The name of the user that requested the acquisition in the acquisition page
- Recurring bulk acquisition jobs will now use the refresh API command instead of creating a new job
- HX Policy API functions to hx_lib
- Additional checks in hx_lib to ensure that the Accept headers match the Content-Type headers

CHANGES

- Bulk acquisition download now utilizes a monitor task module which looks for completed hosts and downloads (and post-processes if so configured) bulk acquisition packages from the controller. This resolves several issues:
    - Tasks are only created for completed hosts, as for ***all*** hosts in the host set at job submission.
    - Scheduler threads are no longer monopolized by download jobs for hosts that aren't responding, which could have resulted in a deadlock.
    - New hosts that have been added to the host set will be downloaded and processed as they complete.
- The HXTool database(hxtool.db), certificates and config file(conf.json) have been moved to their own folder called data. This has been done to improve upgrades both on Docker and on standalone instances.
- Disabled the write cache for TinyDB, database corruption issues were reported with it enabled.
- Logging has been moved to its own class, hxtool_logging.py.
- hx_lib will now invalidate the API token when the controller responds with a 401.

BUGFIXES

- Memory leak in the scheduler that ultimately resulted in HXTool crashing over time
- The database wasn't explicitly closed when HXTool exited, resulting in corruption under some conditions.
- Interval parsing for HXTool scheduler jobs now works as expected.
- The Helix task module now works.
- Child tasks are now notified before calculating the next run, so that they are queued to run.
- The generic restGetUrl() function in hx_lib now supports offset and limit, resolving an issue where IOC conditions were being truncated to the default limit of 50
- The inactive host sets API call was checking the wrong HX API call return.
- The indicator page displayed the incorrect operator for conditions when editing an indicator.
- The indicator clone functionality was fixed to work in the intended way

KNOWN LIMITATIONS

- Some features greatly depend on the number of alerts/acquisitions or other type of data contained in your FireEye endpoint controller. We have limited means of testing with very large configurations so certain panels or tables might take a while to load. The reason behind this is that we need to poll certain data from the endpoint API which depends on resources, hardware specification where you run HXTool and network performance.
- Data stacking does currently not have any limitations of the number of rows that can be returned. Very large data stacking jobs can potentially return too much data causing long load times and high memory use in your web browser

With the addition of task processors HXTool can now potentially use much more system resources than earlier versions due to the fact that we are ingesting and processing each acquisition result. If this feature is heavily used, we recommend running HXTool on a dedicated server