FIREEYE TECHNICAL DOCUMENTATION

HXTOOL 3.2

RELEASE NOTES

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

HXTOOL 3.2 – RELEASE NOTES

NEW FEATURES

- New web framework allowing more dynamic pages
- Background processor have been retired in favor of a new task scheduler
- Dashboard have been recoded with new features
- Anti-virus dashboard
- New alerts functionality allowing custom time ranges and filtering
- You can now save OpenIOCs in HXTool for use with Enterprise Search
- New drill-down function for Enterprise Search allowing easier analysis
- Copy to clipboard, export to csv and xlsx added in multiple places
- Auto-refresh added in multiple places (for tables and graphs)
- Access to normal acquisitions (file, triage, data)
- Acquisition scripts can now be stored in HXTool for use with bulk acquisition feature
- Enterprise search, bulk acquisitions, data stacking and multi-file acquisition now all uses the
- new task scheduler
- Updated UI design

BUGFIXES

- Lots. A large portion of the code of HXTool has been rewritten for architecture reasons and with that many previously identified bugs have been dealt with.
- In total around 200 code commits have been done between v3.1 and v3.2

KNOWN LIMITATIONS

- Alert investigation panel can take a long time to load on an endpoint with many alerts. This is due to the time is takes to retrieve the alerts via the API and depends on network performance between HXTool and the HX Controller.
- 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.