CARBON BLACK

What's New in Carbon Black Response 5.2?

Dated November 2016

Table of Contents

Cb Response 5.2 Feature Benefits

This section outlines the overall benefits provided in Carbon Black (Cb) Response 5.2. With this release, we have achieved these high-level objectives:

- Reduce infrastructure requirements for customers
- Improve scale and data retention by 25%
- Provide Mac/Linux support
- Improve overall product quality and ease of use

The following is an overview of some of the features provided in 5.2 that will benefit customers:

- Suppression/optimized storage/focus on high-value data:
    - Common core Windows OS DLLs are optionally no longer collected.
    - Processes that have no activity will be included as child process of their parent to reduce clicks and noise during an investigation.
- User interface refresh:
    - Updates to improve the experience of the **Process Analysis** page.
    - Updated Cb Response branding.
- OS X and Linux enhancements:
    - Improved the OS X and Linux experience.
    - OS X and Linux sensor upgrades can now be managed in the Cb Response console.
    - Improved data accuracy through modifications in tracking Fork and Exec() calls on OS X and Linux processes.

Eventless (Uninteresting) Process Suppression

Starting with version 5.2, a process is classified as eventless (uninteresting) and therefore suppressed depending on the following definitions:

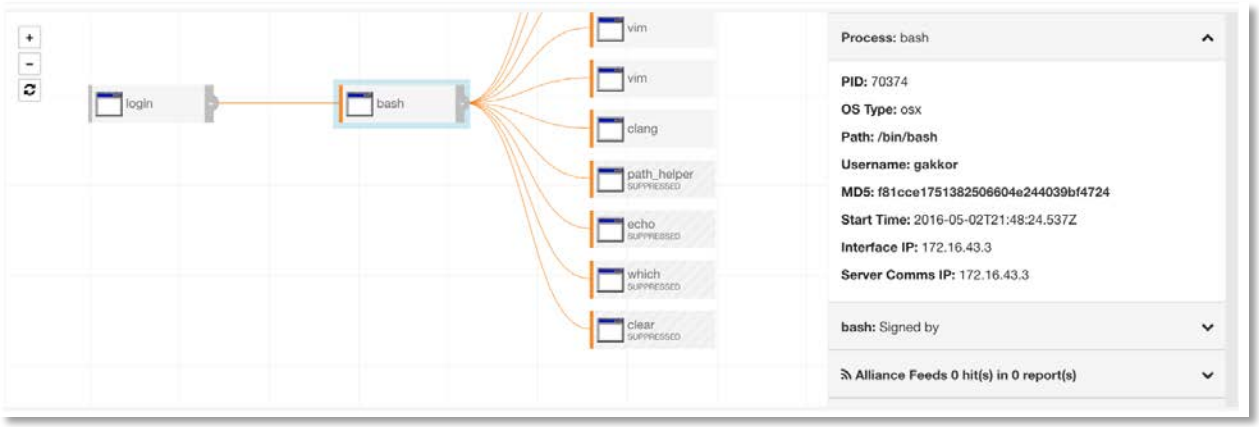
- **No suppression** – All processes are interesting. A process document is created for each process execution instance regardless of its activities and product works as before.
- **Medium suppression** – A process that has no network connections, file modifications, registry modifications, cross-process events, or child process events classified as uninteresting. The only event in an uninteresting process would be module loads.
- **High suppression** – A process that has no network connections, file modifications, registry modifications or child process events classified as uninteresting. The only events in an uninteresting process would be module loads and cross-process events.

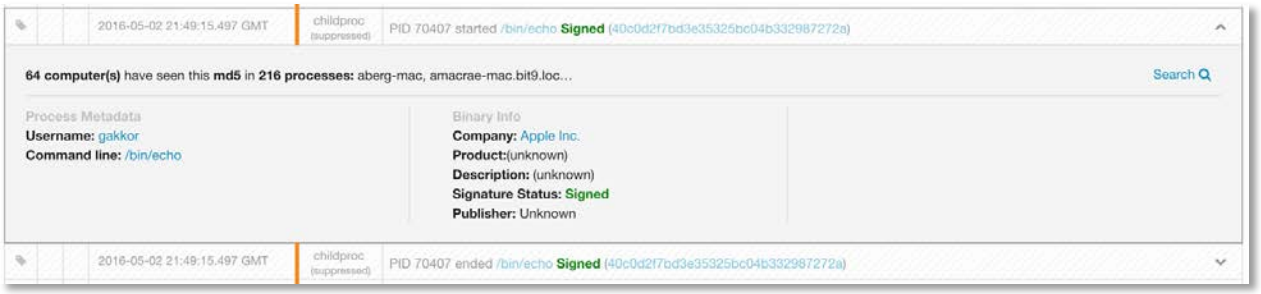
What happens to the suppressed processes?

Suppressed processes are not stored/indexed by the server as stand-alone process documents. From the UI workflow perspective, this means that such processes will not have their own **Process Analyze** page; they cannot be queried by the process_name field. However, there is tracking of the execution of suppressed processes under the parent process.

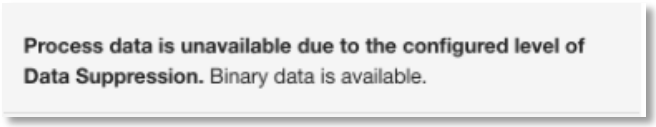
Version 5.2 expands the metadata details for the childproc event type under the parent to include, in addition to existing process and binary information metadata, command line and username information for suppressed processes. Such processes can still be searched by the childproc_name, childproc_md5, cmdline, and username fields from the **Search Processes** pages.

The following figures shows what the **Process Analyze** page looks like for a parent process that has suppressed child processes. In this example, bash has the following suppressed child processes: echo, which, path_helper, and clear.





Note that there is no **Analyze** link within the event dropdown (since there is no process document). If the process node is selected on the process tree, the metadata panel warns you that this process is suppressed:



Eventless process suppression will have a pronounced impact on the number of process documents created by OS X and Linux sensors. For example, many executions of clear, cat, which, and ls type commands on an OS X or Linux host will have reduced data processing impact on the deployment.

Other Important Details

- Suppression levels are configurable a per-sensor-group basis from the UI.
- Suppression is supported on all endpoint platforms (Windows, Linux, and OS X).
- Taking advantage of suppression features requires upgrading endpoints and the server. Legacy sensors still report all events, even if they connect to a 5.2 server with suppression enabled.
- On new installs and upgrades, ALL existing sensor groups will have their suppression level set to MEDIUM by default. The server upgrade process will notify customers of this fact during an upgrade. This can be later changed from the UI.

Improved POSIX Process Tracking

In previous versions of the Cb Response sensor, process tracking attempted to map each process fork and each process execution into unique process instances. This resulted in creation of a high number of process documents, since forks that occur in POSIX environments do not always correlate with a new logical process. Also, the tracking of fork() system calls was not always accurate. In some situations, this resulted in missed or incorrect process information.

In version 5.2, OS X and Linux process tracking becomes more nuanced. POSIX process execution is now handled differently. First, any time a process performs a fork() system call, all activity for that process will continue to be associated with the parent. A new "fork" event type will be displayed on the **Process Analyze** page of the parent, indicating that the parent process performed a fork. The PID of the forked process and the timestamp of when the fork has occurred will be recorded. The first time a process (with a given PID) performs an exec() system call, a new process document will be created and the product will track the execution as a new logical process (current child process behavior). The create time for that new execution will be reported and will correlate to the timestamp when the process was created; that is, when the fork occurred.

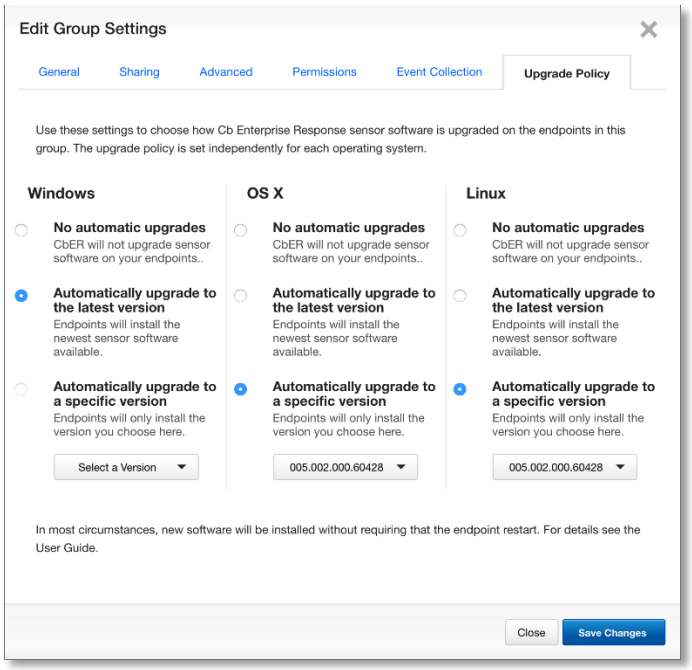
If at any point a process performs a second (or subsequent) exec() system call, a new process document will not be created. This activity will be reported as a new "posix_exec" event type within the process, and the process meta-data will be updated to reflect the new image and command line associated with the exec() system call.

This new process tracking will reduce process document counts generated from OS X and Linux sensors considerably and provide better visibility to different execution/instantiation paths. Fork and posix_exec type events apply only to OS X and Linux sensors. Windows sensors still report child process execution as before.

Support for OS X and Linux Sensor Upgrades from UI

In version 5.2, OS X and Linux sensor upgrades become fully configurable and controllable through the UI. In previous versions of Cb Response, only the Windows sensor upgrade policy was configurable through the UI on a per-sensor-group basis. The OS X and Linux sensor upgrade policy applied globally to all sensor groups at once and had to be done by editing the cb.conf file.

With this version, the upgrade policy for all platforms can be configured from the UI and differ on a per-sensor-group basis. The new **Upgrade Policy** tab on the **Edit Group Settings** page is shown below:



Configuration options that existed for Windows sensor are now extended to OS X and Linux sensors. When upgrading from a previous version of Cb Response server, the following rules apply:

- Configuration options previously set in cb.conf for upgrading OS X and Linux sensors will be ignored.
- For all sensor groups, the OS X and Linux upgrade policy will be set to manual.
- The Windows sensor upgrade policy will remain the same as what was previously set for each sensor group.

Other Features and Improvements

Suppression of Known Windows DLLs

In version 5.2, sensor group settings has a new option to enable suppression of known Windows DLLs. This is a Windows platform-only feature. A known DLL is a Microsoft Windows term for basic DLLs that are loaded into RAM instead of being read from disk with every single process load. When this feature is enabled, trusted DLLs are simply not sent from sensor to the server on a per sensor group setting. More information on the definition of known DLLs can be found here:

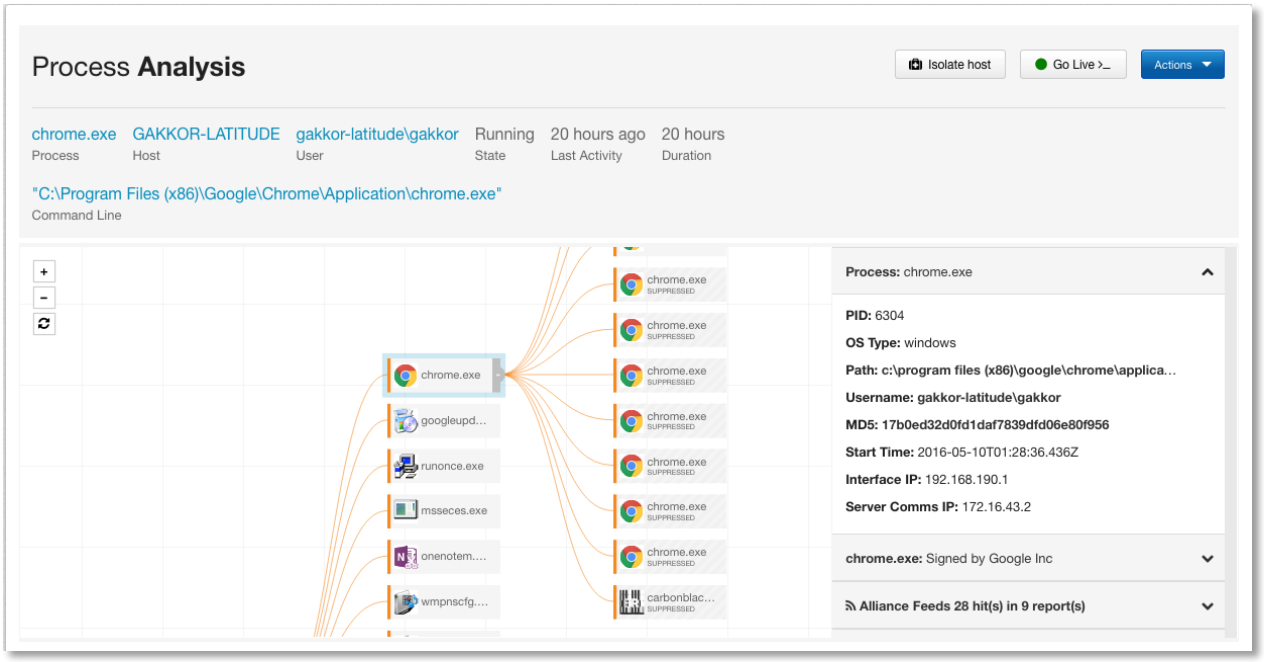
https://technet.microsoft.com/en-us/magazine/2007.09.windowsconfidential.aspx

Improved Triage Alerts Page Performance and Workflow

In version 5.2, the **Triage Alerts** page is re-designed to load faster (with support for viewing more rows at one time) and provide a cleaner workflow for triaging alerts.

Redesigned Process Analyze Page

In version 5.2, the **Process Analysis** page process information header and process tree view have been reworked to provide a cleaner look and richer content.



Corrective Content

The following section outlines all corrective content changes made for the 5.2 release.

Console and Server

1. Corrected an issue where CSV export of hosts that observed a binary in **Binary Details** page failed to work if **Search Processes** page facets were disabled from cb.conf. (CB-4073)
2. Corrected an issue where count of hosts displayed on **Dashboard** page did not correlate with the value displayed on the **Host/Sensor Detail** page. (CB-4042)
3. Corrected an issue where bulk resolve of more than 1000 alerts did not resolve all alerts on the **Triage Alerts** page. (CB-4031)
4. Fixed an issue where search links in **Watchlist** page failed if the search term for the watchlist contained forward slashes. (CB-7275)
5. Corrected an issue where Cb Live Response registry query command failed to return results for registry hives with spaces in them. (CB-3730)
6. Corrected an issue where nightly cron job for tagging documents that match newly added feed reports failed with a KeyError. (CB-7472)
7. Corrected an issue where the searches for time based process document fields showed incorrect syntax under “Showing Results for…” link on the UI. (CB.7724)
8. Fixed an issue where startup script for setting SELinux security context on a NFS share causing startup failures. (CB-3765)
9. Corrected an issue where feed tags associated with a process event erroneously deleted when process document was split into multiple files in the SOLR database. (CB-8346)
10. Fixed an issue where failure to download a file from Cb Alliance Server using cbget when requested file did not exist erroneously reported connectivity to Alliance Server status on the UI as disconnected. (CB-8423)
11. Threat Report create time based searches from the UI now correctly works. (CB-8613)
12. CSV export of events from all search pages are now generated on the server side for robustness. (CB-2826)
13. **Triage Alerts** page is redesigned for cleaner workflow and faster load times. (CB-7548)
14. **Sensors** page is redesigned for faster load times and ability to page list of sensors within a sensor group for cleaner workflow. (CB-8788)
15. Searches for command lines now correctly works for search terms that contain single quotes. (CB-2807)
16. **Process Analysis** page preview now correctly renders if process is missing process name or path. (CB-4956)
17. Notes are now retained correctly if a hash is unbanned. (CB-5113)
18. Resolved inconsistency in the **Action** button functionality on sensor detail and sensor list pages. (CB-5130)
19. Improved Watchlist Name edit functionality. (CB-4913)
20. Added a visual cue for facets selected when no search results are return to improve workflow. (CB-5189)
21. Directory (path) facet on **Process Analysis** page now correctly displays terms for Linux sensors. (CB-4642)
22. Now sharing settings can be configured while creating a new sensor group. (CB-5471)
23. Corrected an issue where default values for various settings on the sensor group dialog were not reflected correctly. (CB-7529)
24. **Watchlists** page sidebar now correctly persists sort order after item selection. (CB-7277)
25. “E-mail Me on Hit” option from **Watchlists** page now works correctly. (CB-7388)
26. **Search Threat Reports** page no longer erroneously display deleted reports for manually added feeds. (CB-7416)

27. **Watchlists** page tooltips now correctly disappear when cursor is moved away from the selection. (CB-7532)
28. Corrected an issue where sensor page did not load correctly when a user with access rights to a customer group did not have permissions to default sensor group. (CB-8320)
29. **Edit Group Settings** dialog now correctly handles team names with longer than 23 characters. (CB-7629)
30. Tooltips that contain quotes are now handled correctly in **Triage Alerts** page tooltips. (CB-7679)
31. Confirmation dialog for network isolation now more accurately inform users on the actions/limitations of this feature. (CB-8228)
32. **Search Binaries** page UI now correctly allows wildcard searches in filename field. (CB-7735)
33. SMTP server names that contain hyphen now can be correctly entered in e-mail settings. (CB-8330)
34. Server UI client application now is prevented from running inside another frame. (CB-8432)
35. **Process Analysis** page now correctly removes spinner when page is rendered. (CB-8886)
36. **Watchlists** page correctly displays the "last hit" time when there are positive hits to the query. (CB-8957)

Windows Sensor

1. Fixed an issue that caused system crash if the sensor was running on a VM that was going through live migration. (CB-7158)
2. GPO installer now have correct product version. (CB-6953)
3. Sensor core driver can cause system crash if installation fails for any reason. (CB-6929)
4. Sensor can associate wrong parent information to processes which it did not see start (sensor was installed on a running system.) (CB-6911)
5. Sensor can associate wrong start up context to processes which it did not see start (sensor was installed on a running system.) (CB-6873)
6. Sensor service can leak memory on system that are under heavy load (seeing high volume of process execution and termination events.) (CB-7065)
7. Sensor cbstream driver can cause softlock on boot or shutdown on Google Cloud Platform. (CB-6977)
8. Corrected an issue where MSI installer failed re-installation. (CB-7372)
9. Sensor stealth mode installation fails if sensor process name provided does not have .exe extension. (CB-7609)
10. Sensor service may cause network shares to disconnect or otherwise fail when accessing files (CB-7764)
11. Sensor uninstall from web UI fails if sensor name is changed under stealth mode. (CB-8291)
12. Corrected an issue where sensor cbtdiflt driver cause system crash when accessing buffers in chained receive handlers. (CB-8245)
13. Fixed an issue where DNS cache in sensor service was not being populated correctly. (CB-8407)
14. Fixed an issue where cbtdiflt driver was causing system crash due to access to pointers without checking their validity (CB-7718)
15. [New Feature] Sensor now implements suppression of eventless (uninteresting) processes. (CB-7266)
16. [New Feature] Sensor now suppresses known DLLs in Windows process executions when enabled per sensor group. (CB-7294)

Linux Sensor

1. Fixed an issue where network connection events were associated with incorrect parent process under load. (CB-8214, CB-8485)
2. Fixed an issue where network connection events did not have process path in the raw protobuf events similar to Windows platform, impacting monitoring of raw events from the enterprise event bus. (CB-9045)
3. Cb Live Response on Linux sensor now correctly accesses directories with apostrophes in their name. (CB-9024)
4. Corrected an issue that caused system crash when sensor is put in isolation mode. (CB-8236)
5. Corrected an issue where some process events were missing process PID information. (CB-8748)
6. Corrected an issue where cbdaemon initialization script referred to a directory that no longer exists. (CB-8467)
7. Sensor no longer reports username after user context event collection option is disabled. (CB-8419)
8. Sensor now correctly updates sensorsettings.ini file values received from the server. (CB-8424)
9. Corrected an issue where sensor driver sporadically crashed sending health alert level 75 (driver failure) to the server. (CB-6700)'
10. [New Feature] Sensor now implements suppression of eventless (uninteresting) processes. (CB-7266)
11. [New Feature] Sensor now differentiate between process forks and other executions. (CB-6756)

OS X Sensor

1. OS X sensor now correctly updates sensorsettings.ini file values received from the server. (CB-6463)
2. OS X sensor sensordiag.sh diagnostic script now does not collect log and diagnostic directories that are not pertaining to its operation when packaging diagnostic information. (CB-7785)
3. Corrected an issue in PSC_fork call in OS X sensor causing a kernel panic in process tracking. (CB-6476)
4. Corrected an issue in parsing of DNS packets that caused high CPU usage. (CB-8394)
5. Corrected an issue that caused up to 6 seconds delay in starting applications on a sensor that did not yet check-in with the server. (CB-8885)
6. Sensor no longer reports its own events from CbOsxSensorService. C(B-8840)
7. Corrected an issue where exceptions in protobuf library causing sensor daemon to crash randomly. (CB-6486)
8. [New Feature] Sensor now implements suppression of eventless (uninteresting) processes. (CB-7266)
9. [New Feature] Sensor now differentiate between process forks and other executions. (CB-6564)