Bit9 + Cb CARBON BLACK	Carbon Black Custom Feed Generation Guide 10-Feb-2014 Cb-Support@Bit9.com

Introduction

The purpose of this document is to setup CarbonBlack Custom feeds for alerts in the form of IPv4, DNS and MD5 Hash. The process details how to produce a Carbon Black JSON feed file. Utilizing the CarbonBlack UI a feed will be generated to retrieve this feed data.

Section

Feed Generation Setup

1. Pull feeds code for http://github.com/carbonblack public Git site.
2. Copy *gen_feed_from_raw_iocs.py* and all items in the *cbfeeds* directory to the platform you intend to distribute the feed from. Three options for feed distribution are:
    a. Feed from a file on the Cb server.
    b. Feed from a listener on the Cb server.
    c. Feed from an external web listener. HTTPS and authentication supported in the feed build. This is beyond the scope this paper.

> **Caution:** Diligence should be used in the creation of a feed file. Error checking of the feed file needs to occur in the feed generation process as the only thing the *cbfeeds* modules are going to provide is error checking on feed format and not a validity of the IOC. It will ensure you have an IP for example; however, it will not prevent you from entering a value that could potentially mark every document in the SOLR database causing performance issues.
>
> If you inadvertently tag documents with a bad IOCs you can un-tag all the documents in the SOLR document store with the following command:
>
> *python /usr/share/cb/cbfeed_scrubber --untag <feedname>*

3. CLI option for manual feed creation tool:

```
# python gen_feed_from_raw_ioc.py -h
Usage: gen_feed_from_raw_ioc.py [options]

Convert a flat file of IOCs to a Carbon Black feed

Options:
```

```
  -h, --help            show this help message and exit
  -n NAME, --name=NAME  Feed Name
  -d DISPLAY_NAME, --displayname=DISPLAY_NAME
                        Feed Display Name
  -u URL, --url=URL     Feed Provider URL
  -s SUMMARY, --summary=SUMMARY
                        Feed Summary
  -t TECHDATA, --techdata=TECHDATA
                        Feed Technical Description
  -i ICON, --icon=ICON  Icon File (PNG format)
  -I IOC_FILENAME, --iocs=IOC_FILENAME
                        IOC filename
  -r REPORT, --report=REPORT
            Report Name
```

4. Ensure that the appropriate .png image file is in the image directory.
5. Run the *gen_feed_from_raw_iocs.py* script to generate the feed JSON structure.

```
python gen_feed_from_raw_iocs.py -n CbFeed -d "Carbon Black Custom Feed" -u
"http://www.carbonblack.com" -s "This is a feed used to demostrate Cb custom feeds." -t
"Not much to say on tech data." -i image/Cb.png -I ip -r "Cb Report"
```

Produces the JSON output below that will be used to in feed creation:


```
{
  "feedinfo": {
    "provider_url": "http://www.carbonblack.com",
    "display_name": "Carbon Black Custom Feed",
    "name": "CbFeed",
    "tech_data": "Not much to say on tech data.",
    "summary": "This is a feed used to demostrate Cb custom feeds.",
    "version": 1,
    "icon":
"iVBORw0KGgoAAAANSUhEUgAABIAAAAKICAIAAACHSRZaAAAAAXNSR0IArs
4c6QAAAARnQU1BAACxjwv8YQUAAAAJcEhZcwAAEnQAABJ0Ad5mH3gAACN
eSURBVHhe7d173Nfz/ …..
///15552XZ6sZMGDAiBEjiotp06YV3ZYmTz/99Dlz5hTt9+CDDx522GFpMslPAwAA
UMZywCQ77rjjm2++eeCBB+ZxlXbt2hWxtO222xbNM3369KKdSr7zne/Mnj27YcO
Gbdu2nTRpUj6w2tChQ3/ ="
  },
  "reports": [
    {
      "title": "Cb Report",
      "timestamp": 1391911703,
      "iocs": {
        "ipv4": [
```

```
                "10.10.10.1",
                "10.11.11.1",
                "10.12.12.1"
              ]
            },
            "score": 100,
            "link": "http://www.carbonblack.com",
            "id": "Cb Report"
          }
        ]
}
```


The above command can be placed in a cron job and ran periodically to update the feed. You can use the '>' to redirect to a file or modify the python script to print to a file. In our case we redirected the output to a file name *CbFeed*.

Feed Local File Setup

This procedure is utilized when the feed data will be maintained locally on the Cb server and served up with the file://path syntax.

6. Place the output of the *gen_feed_from_raw_iocs.py* into a file which is located in a path accessible to the *cb* user.

```
# mkdir /var/cb/feeds
# chown cb /var/cb/feeds
# chgrp cb /var/cb/feeds
# python gen_feed_from_raw_iocs.py -n CbFeed -d "Carbon Black Custom Feed" -u
"http://www.carbonblack.com" -s "This is a feed used to demostrate Cb custom feeds." -t
"Not much to say on tech data." -i image/Cb.png -I ip -r "Cb Report" >
/var/cb/feeds/CbFeed
```

7. Utilize the UI to generate the feed from the file above.

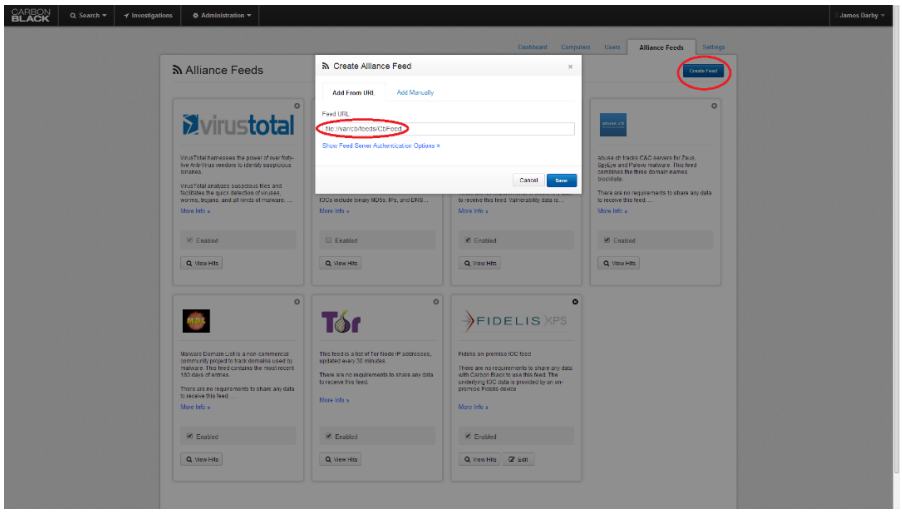

Figure 1 Feed from File

8. The feed will start disabled by default and must be enabled.

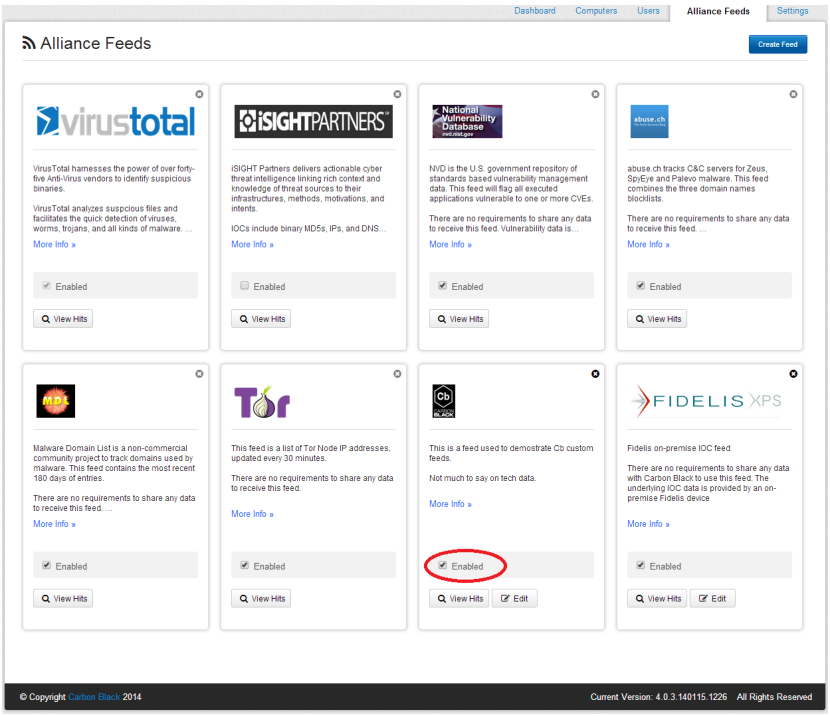

Figure 2 Enable Feed

9. If you would like documents tagged each time the feed is updated run the following command or add it to the a script designed to update the feed and execute the script via cron. By default there is a job in */etc/cron.d/cb* designed to run every 24 hours to tag documents matching IOCs in all defined feeds.

```
# Check newly added documents against active alliance feeds to see if anything new has shown up - every 24 hours
0  2  *  *  * root /usr/bin/python -m cb.maintenance.job_runner --master -s feed_search --
```

```
tag --terms >> /var/log/cb/job-runner/startup.out 2>&1
```

You can run this command to force a single feed to tag documents once the feed source is updated.

```
/usr/bin/python -m cb.maintenance.job_runner --master -s feed_search --tag –terms –feed=CbFeed

**Caution if numerous documents are tagged by the feed this will cause a performance impact of the server as it will have to traverse the entire SOLR data store to tag documents.**
```

Feed from a Listener on the Cb Server

This procedure is utilized when the feed data will be maintained locally on the Cb server and served up with the http://localhost:port syntax.

10. Create a file called */etc/nginx/conf.d/Feed.conf* with the following configuration:

```
server {
  listen        127.0.0.1:81;
  server_name   127.0.0.1;
  error_log     /var/log/cb/nginx/feed.log;
  error_page    404    /404.html;


  location / {
    autoindex on;
    root  /var/cb/feeds/;
  }

  location = /404.html {
    root /var/www/cb/templates;
  }
}
```

You will need to comment out all the entries in */etc/nginx/conf.d/default.conf* or delete the file.

11. Start the *nginx* service and check that you have a listener on 127.0.0.1:81.

```
# service nginx start
Starting nginx:        [  OK  ]

# netstat -lntp | grep nginx
tcp        0      0 127.0.0.1:81              0.0.0.0:*               LISTEN      4519/nginx
```

12. Add the feed the feed on the Alliance tab in the Carbon Black UI.

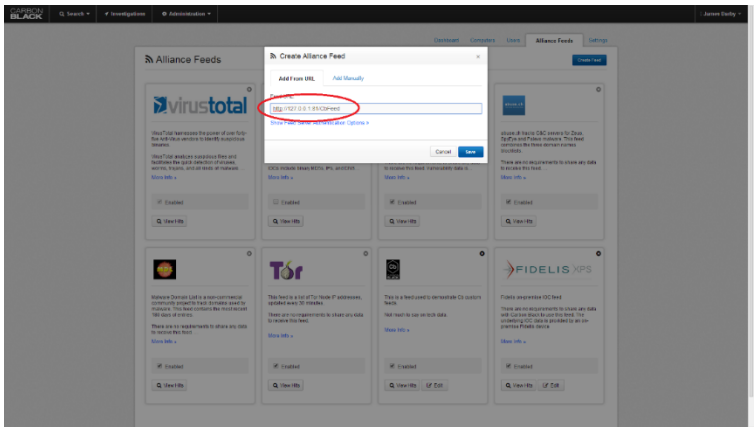

Figure 3 Feed from Localhost

13. Enable the feed.

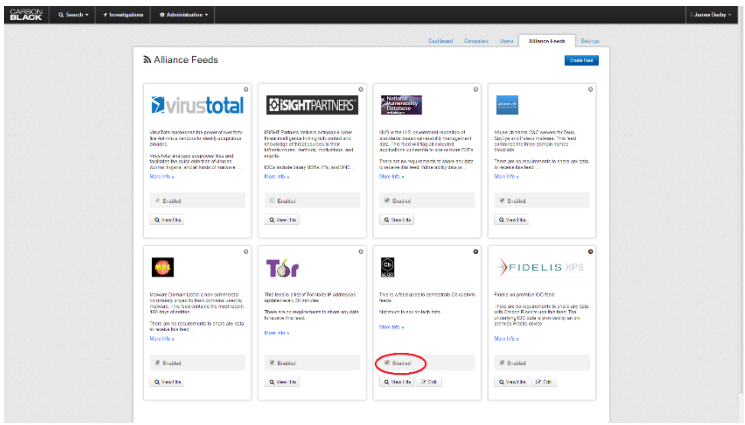

Figure 4 Enable the Feed

14. Utilize the Step 9 to tag the documents associated with this feed. You will need to substitute your feed name for *CbFeed* in the example.

Feed from an External Web Listener

This example utilizes NGINX on a remote server to deliver feed data to multiple Cb servers over HTTP as depicted by Figure 5.

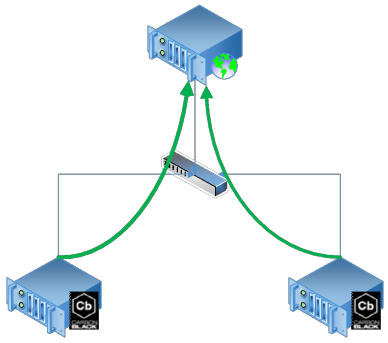

Figure 5 Feed from Remote Server

15. Follow steps 1-6 to produce the JSON feed file on the WebServer.
16. On the WebServer edit */etc/nginx/conf.d/default.conf* to open port 80 and point to the directory containing the feed file.

```
server {
   listen       80;
   server_name  localhost;

   access_log  /var/log/nginx/feed/host.access.log  main;

   location / {
      root   /var/cb/feeds;
      index  index.html index.htm;
   }
   # redirect server error pages to the static page /50x.html
   #
   error_page   500 502 503 504  /50x.html;
   location = /50x.html {
      root   /usr/share/nginx/html;
   }
}
```

17. Start NGINX and configure it to start on the next boot.

```
# service nginx start
# chkconfig –level 345 nginx on
```

18. Configure *iptables* on the WebServer to limit access to only port 80 at a minimum and to the IP addresses of the Cb servers preferably.

```
# vi /etc/sysconfig/iptables

Insert the following vales from the example in figure 5.


-A INPUT -s 192.168.245.135/32 -p tcp -m state --state NEW -m tcp --dport 80 -j ACCEPT
-A INPUT -s 192.168.245.137/32 -p tcp -m state --state NEW -m tcp --dport 80 -j ACCEPT

# service iptables restart
```

19. Configure CbServer01 and CbServer02 configure the feeds to point to http://192.168.245.245/CbFeed.

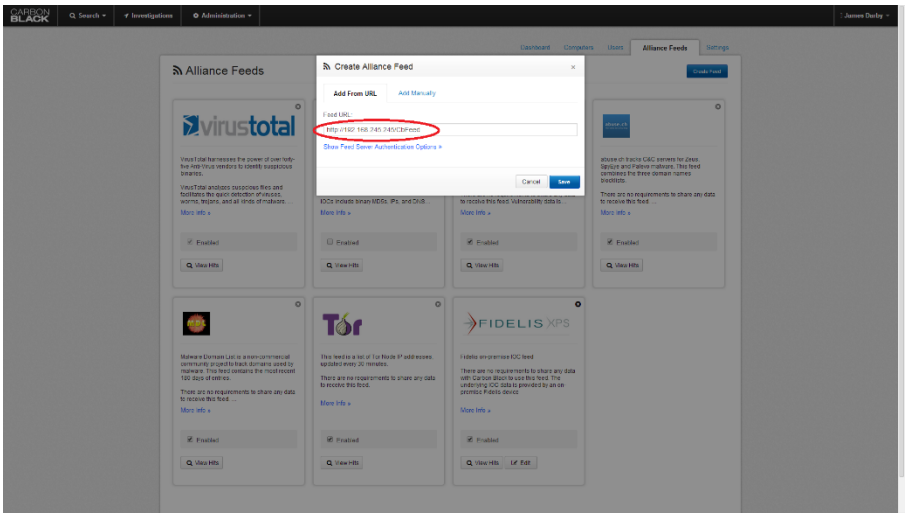

Figure 6 Configure Feed

20. Enable the feed on the Cb Servers.

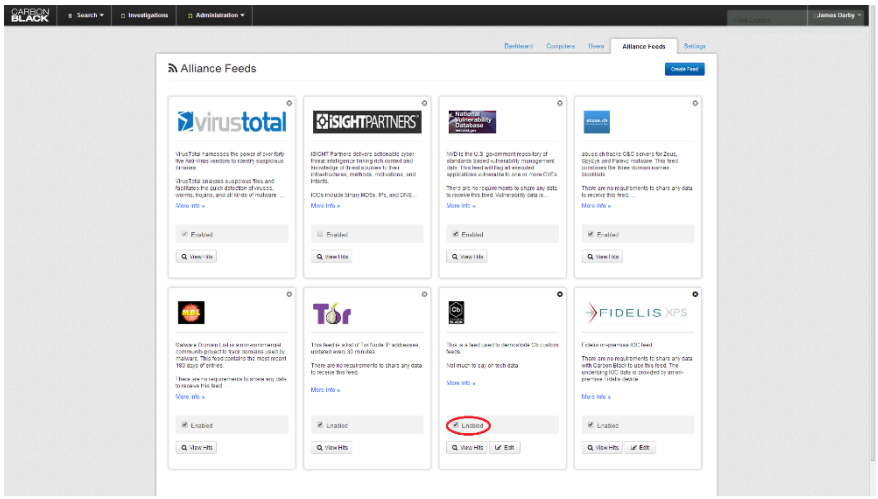

Figure 7 Enable Cb Feed

21. Utilize step 9 on the Cb Servers to read the feed in.