

01
SEP

Bit9 + CARBON BLACK

ARM YOUR ENDPOINTS.



Threat Hunting with Carbon Black

September 1, 2015 / [Ryan Nolette](#) / [Advanced Threat Protection, Detection and Response, Endpoint and Server Security, Prevention, Response, Tech Toolbox](#)

While working the SOC at DEF CON this year, some of the other analysts wanted to learn a bit more about manual threat hunting. Their request was tool agnostic so I did my best to adapt to those who use different tools. The main issue I ran into was that most products have a restrictive and proprietary searching language that makes converting techniques quite difficult. I don't know about you, but I really hate hitting a product limitation after I find something I want to inspect further.

Over the past few years I have tailored my hunting scripts and techniques to use the tools I had available to me at the time. To be honest, at this point, most of my stuff is tailored to Carbon Black and Bit9. But that isn't a bad thing. Especially since both products have very robust open APIs.

Here, I'd like to share a few methods I use to hunt for malware and other malicious activity on systems when I work on an incident response for a customer. You will be able to find more details about these methods in the [Bit9 User eXchange](#) if you are interested.

In the spirit of [OS X-related talks](#) at Black Hat and DEF CON, I am going to show you how to hunt for [WireLurker](#) in your environment. You can get a larger list of scripts as well as the actual scripts used in this blog on the [Bit9 User eXchange](#). I also have a previously posted blog on [WireLurker](#) that has related information and the IOCs that I am going to use.

To find what is anomalous, you must first know what is normal. This issue with this mantra is that every company's environment is different. They all use different applications. They all configure their systems differently. They all have different policies. So how do you create a baseline to define what is normal in order to find what is anomalous?

For OS X, the locations where malware gets persistence can vary significantly because of how the system interprets files. The persistence file could be in ~/Downloads and function the same as if the file were in /Library/StartupItems. The reason why ~/Downloads is not commonly

used by malware authors is because it is a commonly viewed directory where the user will notice the existence of new files.

A few of the common locations for persistence are:

- /Library/LaunchDaemons
- /Library/LaunchAgents
- /Library/StartupItems
- ~/Library/Preference/loginitems.plist
- ~/Library/LaunchAgents
- ~/Library/Frameworks
- /Library/Extensions (kernel extensions)

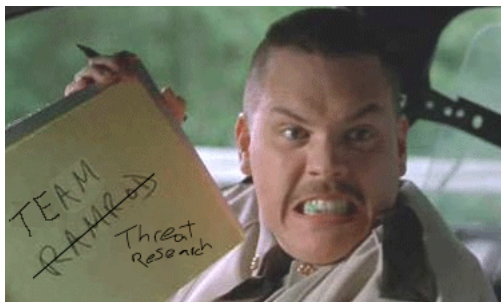
This is not a comprehensive list, but a good starting point for you to practice with. Use these locations as a baseline for comparing a fresh install to a suspect system.

Next up, I want to automate looking in these locations for things that do not belong.

Basically, one of these things is not like the others.



I am going to use the Carbon Black API via a script written by Bitg's Threat Research Team.



This script takes in an input file filled with queries such as "filemod:Users/Shared/run.sh". This query looks for a few of [Wirelurker's](#) artifacts. Below is the usage of the script:

usage: feedstats.py [-h] -c SERVER_URL -a TOKEN [-v] [-f] [-q FEEDFILE]

It requires input of the CB server's url, your CB API key, and the file where all your queries live.

EXAMPLE:

```
master:threatHuntin$ python feedstats.py -a 0g8sdf80g8fake0g8key907ad870870g7asdf0g -c https://192.168.1.2/ -q queries_wirelurker.txt -v
```

Note: No, that isn't my real API key

For further details about this script and other helpful scripts, visit the [Bitg User eXchange](#).

Let's break down my example command:

- python
 - the script is written in python. this is my interpreter. I could also make the file executable but I'm lazy so I just call python.
- py
 - the script I am going to run to interact between my queries and the CB server
- -a
 - my API key
- -c
 - my server url (IP address in this case)
- -q
 - a text file filled with the queries I want to run. one query per line
 - see below for contents

Content of queries_wirelurker.txt (gathered from blog)

```
filemod:Users/Shared/run.sh OR filemod:Library/LaunchDaemons/com.apple.machook_damon.plist OR
filemod:Library/LaunchDaemons/com.apple.globalupdate.plist OR filemod:usr/bin/globalupdate/usr/local/machook/ OR
filemod:usr/bin/WatchProc OR filemod:usr/bin/itunesupdate OR filemod:Library/LaunchDaemons/com.apple.watchproc.plist OR
filemod:Library/LaunchDaemons/com.apple.itunesupdate.plist OR
filemod:System/Library/LaunchDaemons/com.apple.appstore.pluginhelper.plist OR
filemod:System/Library/LaunchDaemons/com.apple.MailServiceAgentHelper.plist OR
filemod:System/Library/LaunchDaemons/com.apple.systemkeychain-helper.plist OR
filemod:System/Library/LaunchDaemons/com.apple.periodic-dd-mm-yy.plist OR filemod:usr/bin/com.apple.MailServiceAgentHelper OR
filemod:usr/bin/com.apple.appstore.PluginHelper OR filemod:usr/bin/periodicdate OR filemod:usr/bin/systemkeychain-helper OR
filemod:usr/bin/stty5.11.pl OR filemod:etc/manpath.d/ OR filemod:usr/local/ipcc/
```

Now that you know the structure of what I am doing for this method of querying the server, let's talk about output of the script.

First to create a baseline for my own knowledge, I run the script against my server.

Script output on my test server before running in production

```
master:threatHuntin$ python feedstats.py -a 0g8sdf80g8fake0g8key907ad870870g7asdf0g -c https://testServer.local/ -q queries_wirelurker.txt
-v
```

```
0.042605 seconds | filemod:Users/Shared/run.sh OR filemod:Library/LaunchDaemons/com.apple.machook_damon.plist OR
filemod:Library/LaunchDaemons/com.apple.globalupdate.plist OR filemod:usr/bin/globalupdate/usr/local/machook/ OR
filemod:usr/bin/WatchProc OR filemod:usr/bin/itunesupdate OR filemod:Library/LaunchDaemons/com.apple.watchproc.plist OR
filemod:Library/LaunchDaemons/com.apple.itunesupdate.plist OR
filemod:System/Library/LaunchDaemons/com.apple.appstore.pluginhelper.plist OR
filemod:System/Library/LaunchDaemons/com.apple.MailServiceAgentHelper.plist OR
filemod:System/Library/LaunchDaemons/com.apple.systemkeychain-helper.plist OR
filemod:System/Library/LaunchDaemons/com.apple.periodic-dd-mm-yy.plist OR filemod:usr/bin/com.apple.MailServiceAgentHelper OR
```

filemod:usr/bin/com.apple.appstore.PluginHelper OR filemod:usr/bin/periodicdate OR filemod:usr/bin/systemkeychain-helper OR
filemod:usr/bin/stty5.11.pl OR filemod:etc/manpath.d/ OR filemod:usr/local/ipcc/ | o

This step is a quick sanity check for me. You can skip this step if you like but I do it for two reasons. First, I want to make sure my query works before production testing. Second, I want to make sure I am not getting false positives in my results.

After my testing step, I run the same script again, but this time against my production server.

Script output on my production server after suspected WireLurker infections:

```
master:threatHuntin$ python feedstats.py -a 0g8sdf80g8fake0g8key907ad87087097asdf0g -c https://192.168.1.2/ -q queries_wirelurker.txt -v
```

```
0.042838 seconds | filemod:Users/Shared/run.sh OR filemod:Library/LaunchDaemons/com.apple.machook_damon.plist OR  
filemod:Library/LaunchDaemons/com.apple.globalupdate.plist OR filemod:usr/bin/globalupdate/usr/local/machook/ OR  
filemod:usr/bin/WatchProc OR filemod:usr/bin/itunesupdate OR filemod:Library/LaunchDaemons/com.apple.watchproc.plist OR  
filemod:Library/LaunchDaemons/com.apple.itunesupdate.plist OR  
filemod:System/Library/LaunchDaemons/com.apple.appstore.pluginhelper.plist OR  
filemod:System/Library/LaunchDaemons/com.apple.MailServiceAgentHelper.plist OR  
filemod:System/Library/LaunchDaemons/com.apple.systemkeychain-helper.plist OR  
filemod:System/Library/LaunchDaemons/com.apple.periodic-dd-mm-yy.plist OR filemod:usr/bin/com.apple.MailServiceAgentHelper OR  
filemod:usr/bin/com.apple.appstore.PluginHelper OR filemod:usr/bin/periodicdate OR filemod:usr/bin/systemkeychain-helper OR  
filemod:usr/bin/stty5.11.pl OR filemod:etc/manpath.d/ OR filemod:usr/local/ipcc/ | 4
```

Found items

You can find the full list of items found in its JSON formatted output below in Appendix A. Since that output takes up a lot of space, I left it at the bottom for those who want to see it.

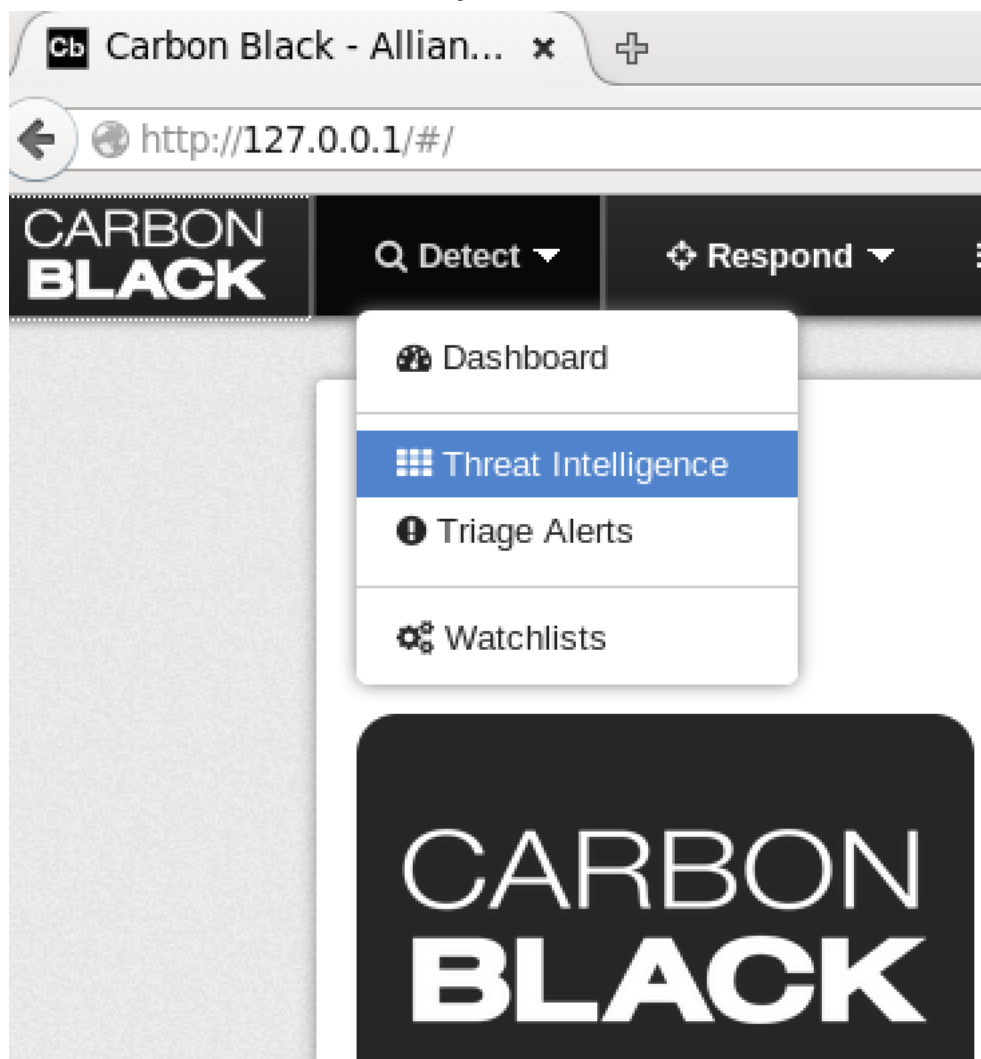
In summary, I used this script to find four artifacts of WireLurker on a system in my production environment.

Protecting the enterprise with my findings

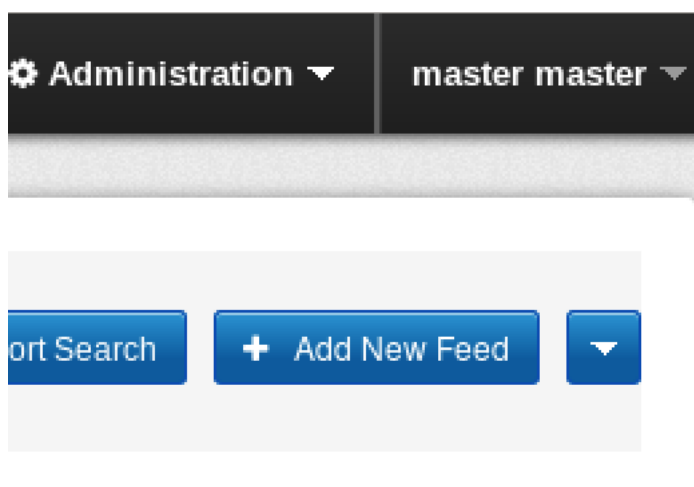
Now that I know my script works and fires on things I care about, I want to automate detection moving forward. Below, in Appendix C, you can see an example of a working feed with the queries used above.

To start the automation process, I create a file on my CB server called "feed_wirelurker" and place my feed text in it. I threw this into /tmp, but you can put it in any directory on the system that Carbon Black has permissions to read. Next I need to upload the feed to my server.



First, login to your production server. Next, go to **Detect -> Threat Intelligence** to get to a list of all your feeds.



Then click on **Add New Feed**



After clicking on **Add New Feed** you will be prompted with a window to input information about the feed. I highly recommend you make the description robust so you can look back on the feed you create six months from now and actually have an idea of what it does.

 **Edit Alliance Feed** 

[Add From URL](#) **Add Manually**

Name:

Feed URL:

Provider URL:

Summary

☐ Use Proxy

☐ Validate Server Cert

[Show Feed Server Authentication Options »](#)

Cancel

Save

Last but not least, we need to enable the feed and set it to generate alerts for hits. I choose to send alerts to console, syslog, and email me upon hits. You can choose any or all of these options, as you prefer. Usually I only have it log events to syslog so that I can work events from my SIEM, but not everyone has a SIEM or has a preexisting workflow for events. Use your preferences on what you should do here.

This feed is a list of wirelurker IOCs

There are no requirements to share any data to receive this feed.

ioc_type_query

[More Info »](#)

★★★★☆?

☒ Enabled

☐ Email Me On Hit

▼ Notifications

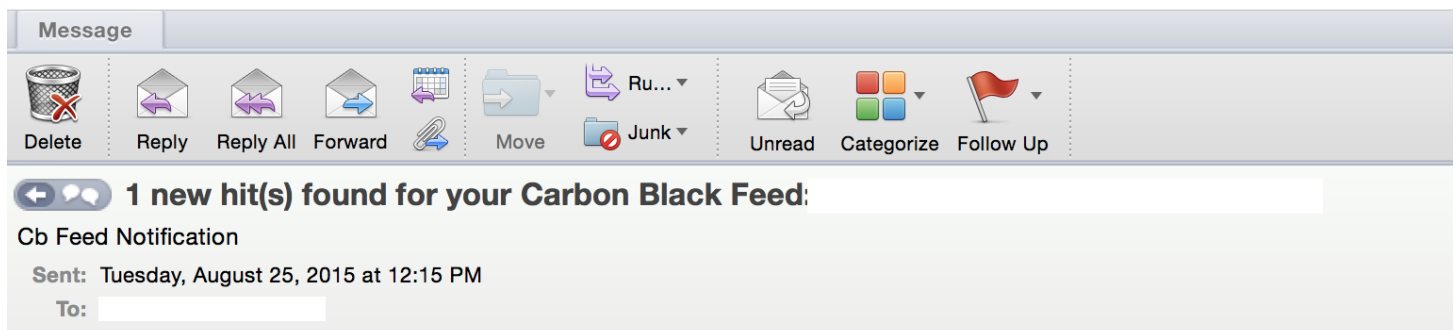
On Hit

☒ Create Alert

☐ Log to Syslog

ns ▼

After the feed has been enabled, if you have selected the CB server to email you, you will receive messages like this:



1 new hit(s) found for Carbon Black Feed: Bit9 + CB Advanced Threats Feed

Don't want to receive notifications for this feed? Unselect the checkbox next to the feed entry's name.

Feed:

Server: (

IOC information:

Type: query

{"index_type": "events", "search_query": "cb.urlver=1&q=

Value: (filemod%3AUsers%2FShared%2Frun.sh%20OR%20filemod%3ALibrary%2FLaunchDaemons%2Fcom.apple.machook_dar
helper.plist%20OR%20filemod%3ASystem%2FLibrary%2FLaunchDaemons%2Fcom.apple.periodic-dd-mm-yy.plist%20OR

Sensor Group: OSX Group - 1 host(s)

Hostname: - 1 new hit(s)

Process information:

Process: bash

Path: bash

This email tells me that a system in my environment now has WireLurker installed on it and needs immediate attention.

This leaves me with a few options for remediation, but personally, I say nuke it.



Until next time, remember my motto. Flag it, Tag it, and Bag it.

Appendix A

```
{  
  
"terms":[  
  
"filemod:Users/Shared/run.sh",  
  
"filemod:Library/LaunchDaemons/com.apple.machook_damon.plist",  
  
"filemod:Library/LaunchDaemons/com.apple.globalupdate.plist",  
  
"filemod:usr/bin/globalupdate/usr/local/machook/",  
  
"filemod:usr/bin/WatchProc",  
  
"filemod:usr/bin/itunesupdate",  
  
"filemod:Library/LaunchDaemons/com.apple.watchproc.plist",  
  
"filemod:Library/LaunchDaemons/com.apple.itunesupdate.plist",  
  
"filemod:System/Library/LaunchDaemons/com.apple.appstore.plughelper.plist",  
  
"filemod:System/Library/LaunchDaemons/com.apple.MailServiceAgentHelper.plist",  
  
"filemod:System/Library/LaunchDaemons/com.apple.systemkeychain-helper.plist",  
  
"filemod:System/Library/LaunchDaemons/com.apple.periodic-dd-mm-yy.plist",
```

```
"filemod:usr/bin/com.apple.MailServiceAgentHelper",

"filemod:usr/bin/com.apple.appstore.PluginHelper",

"filemod:usr/bin/periodicdate",

"filemod:usr/bin/systemkeychain-helper",

"filemod:usr/bin/stty5.11.pl",

"filemod:etc/manpath.d/",

"filemod:usr/local/ipcc/"

},

"total_results":4,

"facets":{

},

"results":[

{

"process_md5":"00000000000000000000000000000000",

"sensor_id":3632,

"modload_count":0,

"parent_unique_id": "",

"cmdline":"(unknown)",

"last_update":"2015-08-25T15:42:22.623Z",

"id":"00000e30-0000-003c-01d0-d9fc5a97f178",

"parent_name":"(unknown)",

"parent_md5":"00000000000000000000000000000000",

"group":"OSX Rollout Group",

"hostname":"master",

"filemod_count":10000,

"start":-1,

"comms_ip": ""
```

```
"netconn_count":0,

"interface_ip": "",

"process_pid":-1,

"username":"(unknown)",

"process_name":"(unknown)",

"path":"(unknown)",

"regmod_count":0,

"parent_pid":-1,

"crossproc_count":0,

"segment_id":11,

"watchlists":[

{

"wid":"178",

"value":"2015-08-25T16:10:05.056Z"

},

{

"wid":"145",

"value":"2015-08-25T16:10:05.056Z"

}

],

"host_type":"workstation",

"os_type":"osx",

"childproc_count":0,

"unique_id":"00000e30-0000-003c-01d0-d9fc5a97f178-00000011"

},

{

"process_md5":"00000000000000000000000000000000",

"sensor_id":3632,
```

```
"modload_count":0,

"parent_unique_id":"","

"cmdline":"bash",

"last_update":"2015-08-25T15:35:21.13Z",

"id":"00000e30-0000-ec54-01d0-db6c99618380",

"parent_name":"(unknown)",

"parent_md5":"00000000000000000000000000000000",

"group":"OSX Rollout Group",

"hostname":"master",

"filemod_count":2,

"start":"2015-08-20T17:21:07Z",

"comms_ip":"","

"netconn_count":0,

"interface_ip":"","

"process_pid":60500,

"username":"root",

"process_name":"bash",

"path":"bash",

"regmod_count":0,

"parent_pid":60383,

"crossproc_count":0,

"segment_id":1,

"watchlists":[

{

"wid":"145",

"value":"2015-08-25T16:10:05.056Z"

},

{
```

```
"wid":178",

"value":"2015-08-25T16:10:05.056Z"

}

],

"host_type":"workstation",

"os_type":"osx",

"childproc_count":0,

"unique_id":"00000e30-0000-ec54-01d0-db6c99618380-000000001"

},

{

"process_md5":"00000000000000000000000000000000",

"sensor_id":3632,

"modload_count":0,

"parent_unique_id":"",

"cmdline":"vi /Users/Shared/run.sh",

"last_update":"2015-08-25T15:33:57.505Z",

"id":"00000e30-0000-ec33-01d0-db6c60c1aa00",

"parent_name":"(unknown)",

"parent_md5":"00000000000000000000000000000000",

"group":"OSX Rollout Group",

"hostname":"master",

"filemod_count":20,

"start":"2015-08-20T17:19:32Z",

"comms_ip":"",

"netconn_count":0,

"interface_ip":"",

"process_pid":60467,

"username":"root",
```

```
"process_name":"bash",

"path":"bash",

"regmod_count":0,

"parent_pid":60383,

"crossproc_count":0,

"segment_id":1,

"watchlists":[

{

"wid":"145",

"value":"2015-08-25T16:10:05.056Z"

},

{

"wid":"178",

"value":"2015-08-25T16:10:05.056Z"

}

],

"host_type":"workstation",

"os_type":"osx",

"childproc_count":0,

"unique_id":"00000e30-0000-ec33-01d0-db6c60c1aa00-000000001"

},

{

"process_md5":"00000000000000000000000000000000",

"sensor_id":3632,

"modload_count":0,

"parent_unique_id":"",

"cmdline":"bash",

"last_update":"2015-08-25T15:32:46.75Z",
```

```
"id":"00000e30-0000-ec23-01d0-db6c3d96fa80",
```

```
"parent_name":"(unknown)",
```

```
"parent_md5":"00000000000000000000000000000000",
```

```
"group":"OSX Rollout Group",
```

```
"hostname":"master",
```

```
"filemod_count":1,
```

```
"start":"2015-08-20T17:18:33Z",
```

```
"comms_ip":"",
```

```
"netconn_count":0,
```

```
"interface_ip":"",
```

```
"process_pid":60451,
```

```
"username":"root",
```

```
"process_name":"bash",
```

```
"path":"bash",
```

```
"regmod_count":0,
```

```
"parent_pid":60383,
```

```
"crossproc_count":0,
```

```
"segment_id":1,
```

```
"watchlists":[
```

```
{
```

```
"wid":"145",
```

```
"value":"2015-08-25T16:10:05.056Z"
```

```
},
```

```
{
```

```
"wid":"178",
```

```
"value":"2015-08-25T16:10:05.056Z"
```

```
}
```

```
],
```

```
"host_type":"workstation",

"os_type":"osx",

"childproc_count":0,

"unique_id":"00000e30-0000-ec23-01d0-db6c3d96fa80-00000001"

}

],

"tagged_pids":[

"0-6":[

{

"id":1,

"name":"Default Investigation"

}

],

"2-1":[

{

"id":8,

"name":"master-user"

}

],

"1352201130871525179-1":[

{

"id":1,

"name":"Default Investigation"

}

],

"0-1":[

{

"id":1,
```



```
"name": "Default Investigation"
```

```
}
```

```
],
```

```
"0-3": [
```

```
{
```

```
"id": 8,
```

```
"name": "master-user"
```

```
}
```

```
],
```

```
"154-1": [
```

```
{
```

```
"id": 8,
```

```
"name": "master-user"
```

```
}
```

```
],
```

```
"None-1": [
```

```
{
```

```
"id": 1,
```

```
"name": "Default Investigation"
```

```
}
```

```
],
```

```
"3933845987791798920-1": [
```

```
{
```

```
"id": 1,
```

```
"name": "Default Investigation"
```

```
}
```

```
],
```

```
"13-1": [
```

```
{
  "id":8,
  "name":"master-user"
},
"6596804764947687655-1":{
  "id":7,
  "name":"20140709-CoreyAdware"
},
"5394681256263443455-1":{
  "id":7,
  "name":"20140709-CoreyAdware"
},
"10-1":{
  "id":1,
  "name":"Default Investigation"
},
"138-1":{
  "id":1,
  "name":"Default Investigation"
}
```

```
],  
  
"155-1":{  
  
  {  
  
    "id":8,  
  
    "name":"master-user"  
  
  }  
  
],  
  
"1-3":{  
  
  {  
  
    "id":8,  
  
    "name":"master-user"  
  
  }  
  
}  
  
},  
  
"elapsed":0.26839303970336914,  
  
"start":0,  
  
"filtered":{  
  
  },  
  
"events":{  
  
  {  
  
    "name":"PREPREPRE/Users/Shared/run.sPOSTPOSTPOSTh",  
  
    "ids":[  
  
      "00000e30-0000-ec54-01d0-db6c99618380-00000001",  
  
      "00000e30-0000-ec54-01d0-db6c99618380-00000001",  
  
      "00000e30-0000-ec33-01d0-db6c60c1aa00-00000001",  
  
      "00000e30-0000-ec33-01d0-db6c60c1aa00-00000001",  
  
      "00000e30-0000-ec23-01d0-db6c3d96fa80-00000001"
```

```
]
}
}
}
}
```

Appendix B

Bitg + Carbon Black

1 new hit(s) found for Carbon Black Feed: Bitg + CB Advanced Threats Feed

Don't want to receive notifications for this feed? Unselect the checkbox next to the feed entry's name.

Feed: Bitg + CB Advanced Threats
Feed

Server: lab1.bitg.com

IOC information:

Type: query

Value: {"index_type": "events", "search_query": "cb.urlver=1&q=(filemod%3AUsers%2FShared%2Frun.sh%20OR%20filemod%3ALibrary%2FLaunchDaemons%2Fcom.apple.machook_damon.plist%20OR%20helper.plist%20OR%20filemod%3ASystem%2FLibrary%2FLaunchDaemons%2Fcom.apple.periodic-dd-mm-yy.plist%20OR%20filemod%3Ausr

Sensor Group: OSX Rollout Group – 1 host(s)

Hostname: master-mac – 1 new hit(s)

Process information:

Process: bash

Path: bash

MD5:

OS: osx

Type:

Username: root

Start 2015-08-
Time: 20T17:18:33.000Z

Last 2015-08-
Updated: 25T15:32:46.750Z

Process 00000e30-0000-ec23-01d0-
GUID: db6c3d96fa80

Segment 1
ID:

Appendix C-- Log in to your Carbon Black server to view the new hit(s) in more detail.

Example code of feed

```
{  
  
  "feedinfo":  
  
  {  
  
    "provider_url": "https://www.carbonblack.com/",  
  
    "display_name": "Bitg + CB Wirelurker",  
  
    "name": "BitgWirelurker",  
  
    "tech_data": "There are no requirements to share any data to receive this feed.",  
  
    "summary": "This feed is a list of wirelurker IOCs",  
  
    "requires": ["ioc_type_query"],  
  
    "version": 1,  
  
    "icon": "",  
  
    "icon_small": "",  
  
    "category": "Bitg + Carbon Black OSX"  
  
  },  
  
  "reports":  
  
  [  
  
    {
```

```
"title": "query everything wirelurker",

"timestamp": 1388570000,

"iocs":

{

"query": [

{

"index_type": "events",

"search_query": "cb.urlver=1&q=
(filemod%3AUsers%2FShared%2Frun.sh%20OR%20filemod%3ALibrary%2FLaunchDaemons%2Fcom.apple.machhook_damon.plist%20OR%20filemod%3ALibrary%2FLaunchDaemons%2Fcom.apple.globalupdate.plist%20OR%20filemod%3Ausr%2Fbin%2Fglobalupdate%2Fusr%2Flocal%2Fmachhook%2F%20OR%20filemod%3Ausr%2Fbin%2FWatchProc%20OR%20filemod%3Ausr%2Fbin%2Fitunesupdate%20OR%20filemod%3ALibrary%2FLaunchDaemons%2Fcom.apple.watchproc.plist%20OR%20filemod%3ALibrary%2FLaunchDaemons%2Fcom.apple.itunesupdate.plist%20OR%20filemod%3ASystem%2FLibrary%2FLaunchDaemons%2Fcom.apple.appstore.pluginhelper.plist%20OR%20filemod%3ASystem%2FLibrary%2FLaunchDaemons%2Fcom.apple.MailServiceAgentHelper.plist%20OR%20filemod%3ASystem%2FLibrary%2FLaunchDaemons%2Fcom.apple.systemkeychain-helper.plist%20OR%20filemod%3ASystem%2FLibrary%2FLaunchDaemons%2Fcom.apple.periodic-dd-mm-yy.plist%20OR%20filemod%3Ausr%2Fbin%2Fcom.apple.MailServiceAgentHelper%20OR%20filemod%3Ausr%2Fbin%2Fcom.apple.appstore.PluginHelper%20OR%20filemod%3Ausr%2Fbin%2Fperiodicdate%20OR%20filemod%3Ausr%2Fbin%2Fsystemkeychain-helper%20OR%20filemod%3Ausr%2Fbin%2Fsty5.11.pl%20OR%20filemod%3Aetc%2Fmanpath.d%2F%20OR%20filemod%3Ausr%2Flocal%2Fipcc%2F
)"

}

]

},

"score": 50,

"link": "http://www.myfeedserver/feed/report/wirelurker_proc",

"id": "wirelurker_everything"

}

]

}
```

