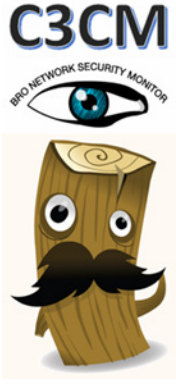
# C3CM: Part 2 – BroIDS with Logstash and Kibana

By Russ McRee – ISSA Senior Member, Puget Sound (Seattle), USA Chapter

## Prerequisites

Linux OS – Ubuntu Desktop 12.04 LTS discussed herein




In Part 1 of our C3CM discussion we established that, when applied to the practice of combating bots and APTs, C3CM can be utilized to *identify, interrupt, and counter the command, control, and communications capabilities of our digital assailants.*

Where, in part one of this three-part series, we utilized Nfsight with Nfdump, Nfsen, and fprobe to conduct our identification phase, we'll use BroIDS (Bro), Logstash, and Kibana as part of our interrupt phase. Keep in mind that while we're building our own Ubuntu system to conduct our C3CM activities, you can perform much of this work from Doug Burk's outstanding Security Onion (SO). You'll have to add some packages such as those we did for Part 1, but Bro as described this month is all ready to go on SO. Candidly, I'd be using SO for this entire series if I hadn't already covered it in *toolsmith*, but I'm also a firm believer in keeping the readership's Linux foo strong as part of tool installation and configuration. The best way to learn is to do, right?

That said, I can certainly bring to your attention my latest must-read recommendation for *toolsmith* aficionados: Richard Bejtlich's *The Practice of Network Security Monitoring.* This gem from No Starch Press covers the life cycle of network security monitoring (NSM) in great detail and leans on SO as its backbone. I recommend an immediate download of the latest version of SO and a swift purchase of Richard's book.[1]

Bro has been covered at length by Doug, Richard in his latest book, and others, so I won't spend a lot of time on Bro configuration and usage. I'll take you through a quick setup for our C3CM VM, but the best kickoff point for your exploration of Bro, if you haven't already been down the path to enlightenment, is Kevin Liston's Internet Storm Center Diary post, "Why I Think You Should Try Bro." [2] You'll note as you read the post and comments that SO includes ELSA as an excellent "front end" for Bro and that you can be up and running with both when using SO. True (and ELSA does rock[3]), but our mission here is to bring alternatives to light and heighten awareness for additional tools. As Logstash may be less extensively on infosec's radar than Bro, I will spend a bit of time on its configuration and capabilities as a lens and amplifier for Bro logs. Logstash comes to you courtesy of Jordan Sissel. As I was writing this, Elasticsearch announced that Jordan will be joining them to develop Logstash with the Elasticsearch team.[4] This is a match made in heaven and means nothing but good news for us from the end-user perspective. Add Kibana (also part of the Elasticsearch family) and we have Bro log-analysis power of untold magnitude. To spell it all out for you, per the Elasticsearch site, you now have at your disposal a "fully open-source product stack for logging and events management: Logstash for log processing, Elasticsearch as the real-time analytics and search engine, and Kibana as the visual front end." Sweet!

## Bro

First, a little Bro configuration work as this is the underpinning of our whole concept. I drew from Kevin Wilcox's Open-Source Toolbox[5] for a quick, clean Bro install. If you plan to cluster or undertake a production environment-worthy installation, you'll want to read the formal documentation[6] and definitely do more research.

You'll likely have a few of these dependencies already met, but play it safe and run:

```
sudo apt-get install cmake make gcc g++ flex
bison libpcap-dev libssl-dev python-dev swig
zlib1g-dev libmagic-dev libgoogle-perftools-dev
libgeoip-dev
```

Grab Bro: `wget http://www.bro-ids.org/downloads/release/bro-2.1.tar.gz`

Unpack it: `tar zxf bro-2.1.tar.gz`

CD to the bro-2.1 directory and run `./configure` then `make` and finally `sudo make install`.

Run `sudo visudo` and add `:/usr/local/bro/bin` (inside the quotation marks) to the `secure_path` parameter to the end of the line the save the file and exit. This ensures that `broctl`, the Bro control program is available in the path.

1 http://nostarch.com/nsm.

2 https://isc.sans.edu/diary/Why+I+think+you+should+try+Bro/15259.

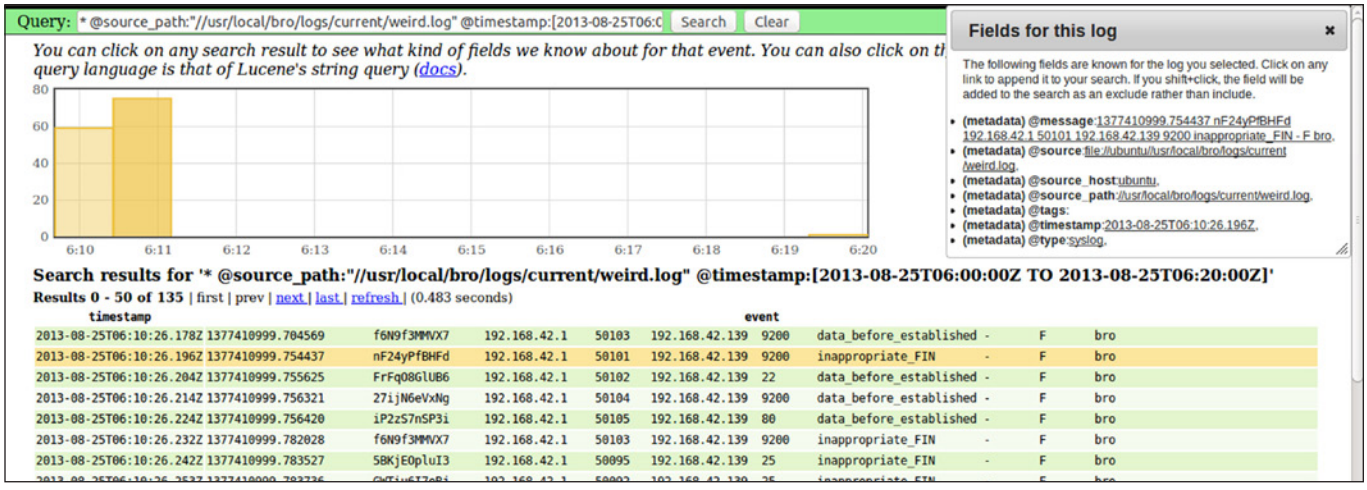
3 http://taosecurity.blogspot.com/2013/01/security-onion-elsa-or-snorby-capme.html.

4 http://www.elasticsearch.com/blog/welcome-jordan-logstash/.

5 http://opensecgeek.blogspot.com/2013/01/nsm-with-bro-ids-part-2-install.html

6 http://www.bro.org/sphinx/index.html.

**Figure 1 – Logstash query power**




Run `sudo broctl` and *Welcome to BroControl 1.1* should pop up, then exit.

You'll likely want to add `broctl start` to `/etc/rc.local` so Bro starts with the system, as well as add `broctl cron` to `/etc/crontab`.

There are Bro config files that warrant your attention as well in `/usr/local/bro/etc`. You'll probably want have Bro listen via a promiscuous interface to a SPAN port or tapped traffic (NSA pickup line: "I'd tap that." Not mine, but you can use it ☺). In `node.cfg` define the appropriate interface. This is also where you'd define standalone or clustered mode. Again keep in mind that in high traffic environments you'll definitely want to cluster. Set your local networks in `networks.cfg` to help Bro understand ingress versus egress traffic. In `broctl.cfg`, tune the mail parameters if you'd like to use email alerts.

Run `sudo broctl` and then execute `install`, followed by `start`, then `status` to confirm you're running. The most important part of this whole effort is where the logs end up, given that that's where we'll tell Logstash to look shortly. Logs are stored in `/usr/local/bro/logs` by default, and are written to event directories named by date stamp. The most important directory, however, is `/usr/local/bro/logs/current`; this is where we'll have Logstash keep watch. The following logs are written here, all with the .log suffix: communication, conn, dns, http, known_hosts, software, ssl, stderr, stdout, and weird.

## Logstash

Logstash requires a JRE. You can ensure Java availability on our Ubuntu instance by installing OpenJDK via `sudo apt-get install default-jre`. If you prefer, install Oracle's version,[7] then define your preference as to which version to use with `sudo update-alternatives --config java`. Once you've defined your selection `java –version` will confirm.

Logstash runs from a single JAR file; you can follow Jordan's simple getting started guide[8] and be running in minutes. Carefully read and play with each step in the guide, including saving to Elasticsearch, but use my `logstash-c3cm.conf` config file[9] that I've posted to my site for you as part of the running configuration you'll use. You'll invoke it as follows (assumes the Logstash JAR and the conf file are in the same directory):

```
java -jar logstash-1.1.13-flatjar.jar agent
-f logstash-c3cm.conf -- web --backend
elasticsearch://localhost/
```

The result, when you browse to `http://localhost:9292/search` is a user interface that may remind you a bit of Splunk. There is a lot of query horsepower available here. If you'd like to search all entries in the `weird.log` as mentioned above, execute this query:

```
* @source_path:"//usr/local/bro/logs/current/weird.log"
```

Modify the log type to your preference (dns, ssl, etc) and you're off to a great start. `Weird.log` includes "unusual or exceptional activity that can indicate malformed connections, traffic that doesn't conform to a particular protocol, malfunctioning/misconfigured hardware, or even an attacker attempting to avoid/confuse a sensor," and `notice.log` will typically include "potentially interesting, odd, or bad" activity. Click any entry in the Logstash UI and you'll see a pop-up window for "Fields for this log." You can drill into each field for more granular queries, and you can also drill in the graph to zoom into time periods as well. Figure 1 represents a query of `weird.log` in a specific time window.

There is an opportunity to create a Bro plugin[10] for Logstash; it's definitely on my list.

Direct queries are excellent, but you'll likely want to create dashboard views to your Bro data, and that's where Kibana comes in.

## Kibana

Here's how easy this is. Download Kibana,[11] unpack `kibana-master.zip`, rename the resulting directory to `kibana`, copy

7 http://askubuntu.com/questions/56104/how-can-i-install-sun-oracles-proprietary-java-6-7-jre-or-jdk.

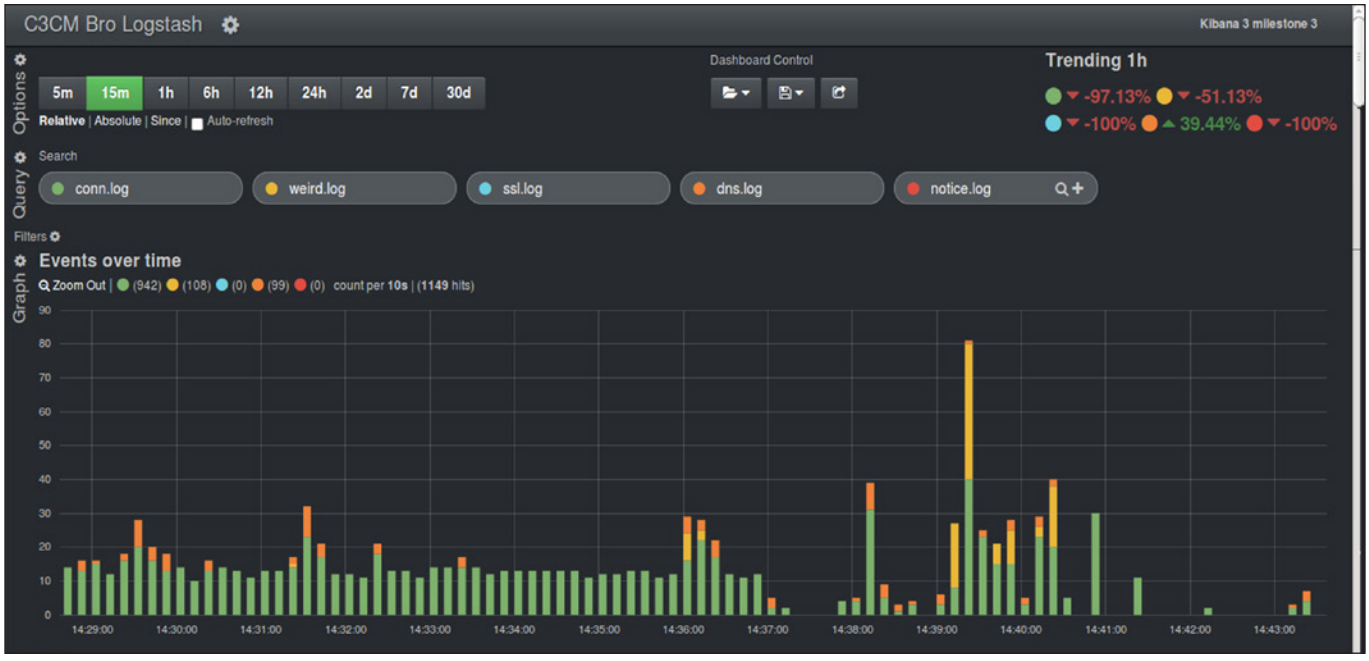
8 http://logstash.net/docs/1.1.13/tutorials/getting-started-simple.

9 http://holisticinfosec.org/toolsmith/files/logstash/logstash-c3cm.conf.

10 http://logstash.net/docs/1.1.13/.

11 https://github.com/elasticsearch/kibana.

**Figure 2 – Kibana dashboard with Nmap spike**




or move it to `/var/www`, `edit config.js` such that instead of `localhost:9200` for the `elasticsearch` parameter, it's set to the FQDN or IP address for the server, even if all elements are running on the same server as we are doing here. Point your browser to `http://localhost/kibana/index.html#/dashboard/file/logstash.json` and voila, you should see data. However, I've exported my dashboard file for you.[12] Simply save it to `/var/www/kibana/dashboards`, then click the open-folder icon in `Dashboard Control` and select `C3C-MBroLogstash.json`. I've included one-hour trending and search queries for each of the interesting Bro logs. You can tune these to your heart's content. You'll note the timepicker panel in the upper left-hand corner. Set auto-refresh on this and navigate over time as you begin to collect data as seen in figure 2 where you'll note a traffic spike specific to an Nmap scan.

Dashboards are excellent, and Kibana represents a ton of flexibility in this regard, but you're probably asking yourself, How does this connect with the Interrupt phase of C3CM? Bro does not serve as a true IPS per se, but actions can be established to clearly "interrupt control and communications capabilities of our digital assailants." Note that one can use Bro scripts to raise notices[13] and create custom notice actions per Notice Policy.[14] Per a 2010 write-up on the Security Monks blog,[15] consider Detection Followed By Action. "Bro policy scripts execute programs, which can, in turn, send e-mail messages, page the on-call staff, automatically terminate existing connections, or, with appropriate additional software, insert access control blocks into a router's access control list. With Bro's ability to execute programs at the operating system level, the actions that Bro can initiate are only limited by the computer and network capabilities that support Bro." This is an opportunity for even more exploration and discovery; should you extend this toolset to create viable interrupts (I'm working on it but ran out of time for this month's deadline), please let us know via comments or email.

## In conclusion

Recall from the beginning of this discussion that I've defined C3CM as methods by which *to identify, interrupt, and counter the command, control, and communications capabilities of our digital assailants.*

With Bro, Logstash, and Kibana, as part of our C3CM concept, the second phase (interrupt) becomes much more viable: better detection leads to better action. Next month we'll discuss the counter phase of C3CM using ADHD (Active Defense Harbinger Distribution) scripts.

Ping me via email if you have questions (russ at holisticinfosec dot org).

Cheers…until next month.

## About the Author

*Russ McRee manages the Security Analytics team (security incident management, penetration testing, monitoring) for Microsoft's Online Services Security & Compliance organization. In addition to toolsmith, he's written for numerous other publications, speaks regularly at events such as DEFCON, Black Hat, and RSA, and is a SANS Internet Storm Center handler. As an advocate for a holistic approach to the practice of information assurance Russ maintains holisticinfosec.org. He serves in the Washington State Guard as the Cybersecurity Advisor to the Washington Military Department. Reach him at russ at holisticinfosec dot org or @holisticinfosec.*

12 http://holisticinfosec.org/toolsmith/files/logstash/C3CMBroLogstash.json.

13 http://www.bro.org/sphinx/notice.html#raising-notices.

14 http://www.bro.org/sphinx/notice.html#id11.

15 http://blog.securitymonks.com/2010/08/26/three-little-idsips-engines-build-their-open-source-solutions/.