C3CM: Part 1 – Nfsight with Nfdump and Nfsen

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

C3CM Nfsight

Prerequisites

Linux OS –Ubuntu Desktop 12.04 LTS discussed herein

I've been spending a fair bit of time reading, studying, writing, and presenting as part of officer candidate training in the Washington State Guard. When I'm pinned I may be one of the oldest 2nd Lieutenants you've ever imagined (most of my contemporaries are Lieutenant Colonels and Colonels), but I will have learned beyond measure. As much of our last drill weekend was spent immersed in Army operations, I've become quite familiar with *Army Field Manuals 5-0 The Operations Process* and *1-02 Operational Terms and Graphics*. Chapter 2 of FM 1-02, Section 1 includes acronyms and abbreviations, and it was there I spotted it, the acronym for command, control, and communications countermeasures: C3CM. This gem is just ripe for use in the cybersecurity realm, and I intend to be the first to do so at length. C2 analysis may be good enough for most, but I say let's go next level. ;-) Initially, C3CM was most often intended to wreck the command and control of enemy air defense networks, a very specific Air Force mission. Apply that mind-set in the context of combating bots and APTs and you're onboard. Our version of C3CM, therefore, is to identify, interrupt, and counter the command, control, and communications capabilities of our digital assailants.

Part one of our three-part series on C3CM will utilize Nfsight with Nfdump, Nfsen, and fprobe to conduct our identification phase. These NetFlow tools make much sense when attempting to identify the behavior of your opponent on high-volume networks that don't favor full-packet capture or inspection.

A few definitions and descriptions to clarify our intent:

1. NetFlow is Cisco's protocol for collecting IP traffic information and is an industry standard for traffic monitoring
2. Fprobe[1] is a libpcap-based tool that collects network traffic data and emits it as NetFlow flows towards the specified collector and is very useful for collecting NetFlow from Linux interfaces
3. Nfdump[2] tools collect and process NetFlow data on the command line and are part of the Nfsen project
4. Nfsen[3] is the graphical web-based frontend for the Nfdump NetFlow tools
5. Nfsight,[4] our primary focus as detailed on its Sourceforge page, is a NetFlow processing and visualization application designed to offer a comprehensive network awareness. Developed as a Nfsen plugin to construct bidirectional flows out of the unidirectional NetFlow flows, Nfsight leverages these bidirectional flows to provide client/server identification and intrusion detection capabilities.

Nfdump and Nfsen are developed by Peter Haag while Nfsight is developed by Robin Berthier. Robin provided extensive details regarding his project. He indicated that Nfsight was born from the need to easily retrieve a list of all the active servers in a given network. Network operators and security administrators are always looking for this information in order to maintain up-to-date documentation of their assets and to rapidly detect rogue hosts. As mentioned above, it made sense to extract this information from NetFlow data for practicality and scalability. Robin pointed out that NetFlow is already deployed in most networks and offers a passive and automated way to explore active hosts even in extremely large networks (such as the spectacularly massive Microsoft datacenter environment I work in). The primary challenge in designing and implementing Nfsight lay in accurately identifying clients and servers from omnidirectional NetFlow records given that NetFlow doesn't keep track of client/server sessions; a given interaction between two hosts will lead to two separate NetFlow records. Nfsight is designed to pair the right records and to identify which host initiated the connection and does so through a set of heuristics that are combined with a Bayesian inference algorithm. Robin pointed out that timing (which host started the connection) and port numbers (which host has a higher port number) are two examples of heuristics used to differentiate client from server in bidirectional flows. He also stated that the advantage of Bayesian inference is to converge towards a more accurate identification as evidence is collected over time from the different heuristics. As a result, Nfsight gains a comprehensive understanding of active servers in a network after only few hours.

Another important Nfsight feature is the visual interface that allows operators to query and immediately display the results

1 http://fprobe.sourceforge.net/.
2 http://Nfdump.sourceforge.net/.
3 http://NfSen.sourceforge.net/.
4 http://sourceforge.net/projects/nfsight/?source=directory.

through any Web browser. One can, as an example, query for all the SSH servers.

"The tool will show a matrix where each row is a server (IP address and port/service) and each column is a time slot. The granularity of the time slot can be configured to represent a few minutes, an hour, or a day. Each cell in the matrix shows the activity of the server for the specific time period. Operators instantly assess the nature and volume of client/server activity through the color and the brightness of the colored cell. Those cells can even show the ratio of successful-to-unsuccessful network sessions through the red color. This enables operators to identify scanning behavior or misconfiguration right away. This feature was particularly useful during an attack against SSH servers recorded in a large academic network. As shown in figure 1, the green cells represent normal SSH server activity and suddenly red/blue SSH client activity starts, indicating a coordinated scan." – *Robin Berthier*

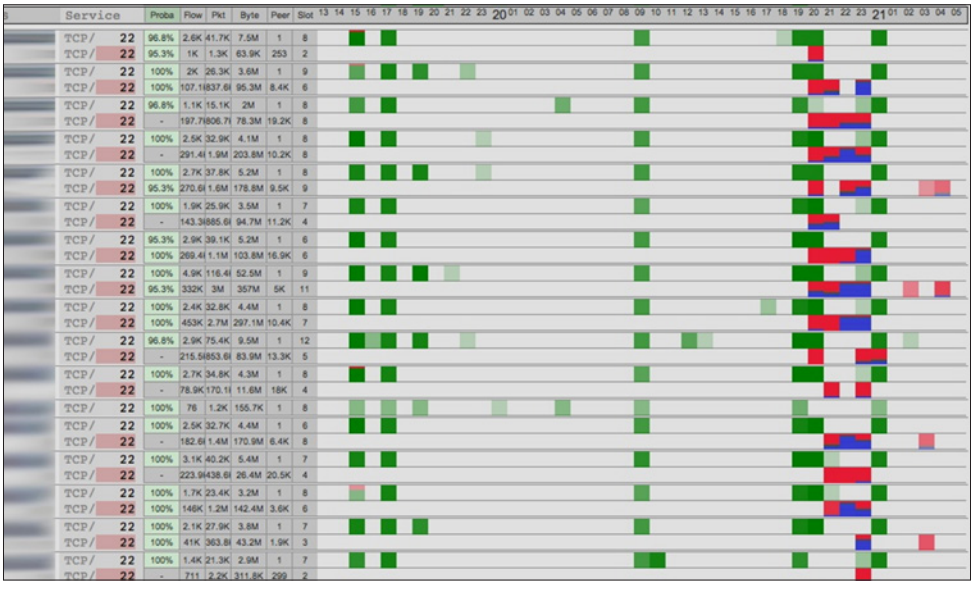

Figure 1 – Nfsight encapsulates attack against SSH servers

Robin described the investigation of the operating systems on those SSH servers where the sysadmins found that they were using a shared password database that an attacker was able to compromise. The attacker then installed a bot in each of the servers, and launched a scanning campaign from each compromised server. Without the visual representation provided by Nfsight, it would have taken much longer to achieve situational awareness, or worse, the attack could have gone undetected for days.

I am here to tell you, dear reader, with absolute experiential certainty, that this methodology works at scale for identifying malicious or problematic traffic, particularly when compared against threat feeds such as those provided by Collective Intelligence Framework. Think about it from the perspective of detecting evil for cloud services operators and how to do so effectively at scale. Tools such as Nfdump, Nfsen, and Nfsight start to really make sense.

Preparing your system for Nfsight

Now that you're all excited, I will spend a good bit of time on installation as I drew from a number of sources to achieve an effective working base for part one of our three-part series of C3CM. This is laborious and detailed so pay close attention. I started working from an Ubuntu Desktop 12.04 LTS virtual machine I keep in my collection, already configure with Apache and MySQL. One important distinction here. I opted to not spin up my old Cisco Catalyst 3500XL in my lab as it does not support NetFlow and instead opted to use fprobe to generate flows right on my Ubuntu instance being configured as an Nfsen/Nfsight collector. This is acceptable in a low volume lab like mine but won't be effective in any production environment. You'll be sending flows from supported devices to your Nfsen/Nfsight collector(s) and defining them explicitly in your Nfsen configuration as we'll discuss shortly. Keep in mind that preconfigured distributions such as Network Security Toolkit[5] come with the like of Nfdump and Nfsen already available, but I wanted to start from scratch with a clean OS so we can build our own C3CM host during this series.

From your pristine Ubuntu instance, begin with a system update to ensure all packages are current: `sudo apt-get update && sudo apt-get upgrade`.

You can configure the LAMP server during VM creation from the ISO or do so after the fact with `sudo apt-get install tasksel` then `sudo tasksel` and select LAMP server.

Install the dependencies necessary for Nfsen and Nfsight: `sudo apt-get install rrdtool mrtg librrds-perl librrdp-perl librrd-dev Nfdump libmailtools-perl php5 bison flex librrds-perl libpcap-dev libdbi-perl picviz fprobe`. You'll be asked two questions during this stage of the install. The fprobe install will ask which interface to capture from; typically the default is eth0. For collector address, respond with localhost:9001. You can opt for a different port, but we'll use 9001 later when configuring the listening component of Nfsen. During the mrtg install, when prompted to answer "Make /etc/mrtg.cfg owned by and readable only by root?" answer *Yes*.

The Network Startup Resource Center (NSRC) conducts annual workshops; in 2012 during their Network Monitoring and Managements event Nfsen installation was discussed at length.[6] Following their guidance:

5 http://networksecuritytoolkit.org/nst/index.html.

6 https://nsrc.org/workshops/2012/drukren-nsrc/raw-attachment/wiki/Agenda/install-nfsen.pdf.

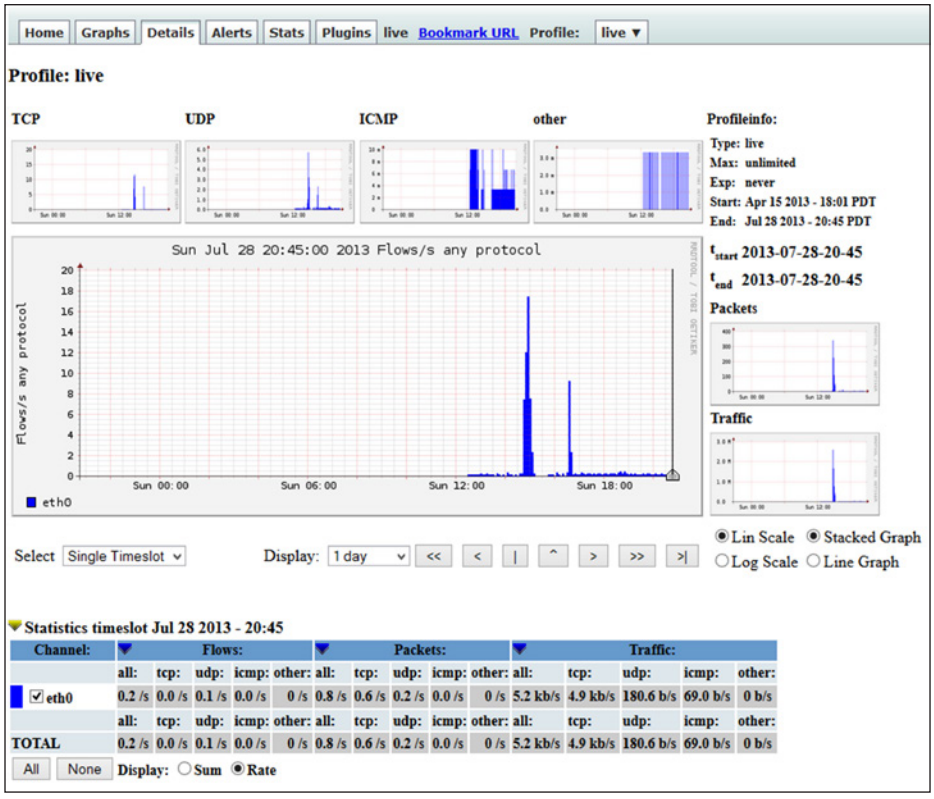

Figure 2 – Nfsen beginning to render data

Install and configure Nfsen

Go to http://holisticinfosec.org/index.php/how-to-main-menu-27/15-how-to/191-install-nfsen.

You're halfway there now. Check your Nfsen installation via your browser. The URL is http://192.168.42.131/nfsen/nfsen.php?tab=0 on my server. Note: if you see a backend version mismatch message, incorporate the changes into nfsen.php as noted in this diff file.[7]

As data starts coming in (you can force this with a ping –t (Windows) of your Nfsen collector IP and/or an extensive Nmap scan) you should see results similar to those seen from the *Details* tab in figure 2 (allow it time to populate).

Install Nfsight

Install Nfsight,[8] as modified from Steronius' Computing Bits[9] (follow me explicitly here): http://holisticinfosec.org/index.php/how-to-mainmenu-27/15-how-to/192-install-nfsight.

Congratulations, you should now be able to login to Nfsight! The credentials to login to Nfsight are those you defined when running the Nfsight installer script (installer.php). On my server, I do so at http://192.168.42.131/nfsen/nfsight/index.php.

Nfsight in flight

After all that, you're probably ready to flame me with a "WTF did you just make me do, Russ!" email. I have to live up to being the tool in *toolsmith*, right? I'm with you, but it will have been worth it, I promise. As flows begin to populate data you'll have the ability to drill into specific servers, clients, and services. I generated some noisy traffic against some Microsoft IP ranges that I was already interested in validating, which in turn gave the impression of a host on my network scanning for DNS servers. Figure 3 show an initial view where my rogue DNS scanner shows up under Top 20 active internal servers.

You can imagine how, on a busy network, these Top 20 views could be immediately helpful in identifying evil egress traffic. If you click on a particular IP in a Top 20 view, you'll be treated to service activity in a given period (adjustable in three-hour increments). You can then drill in further by five-

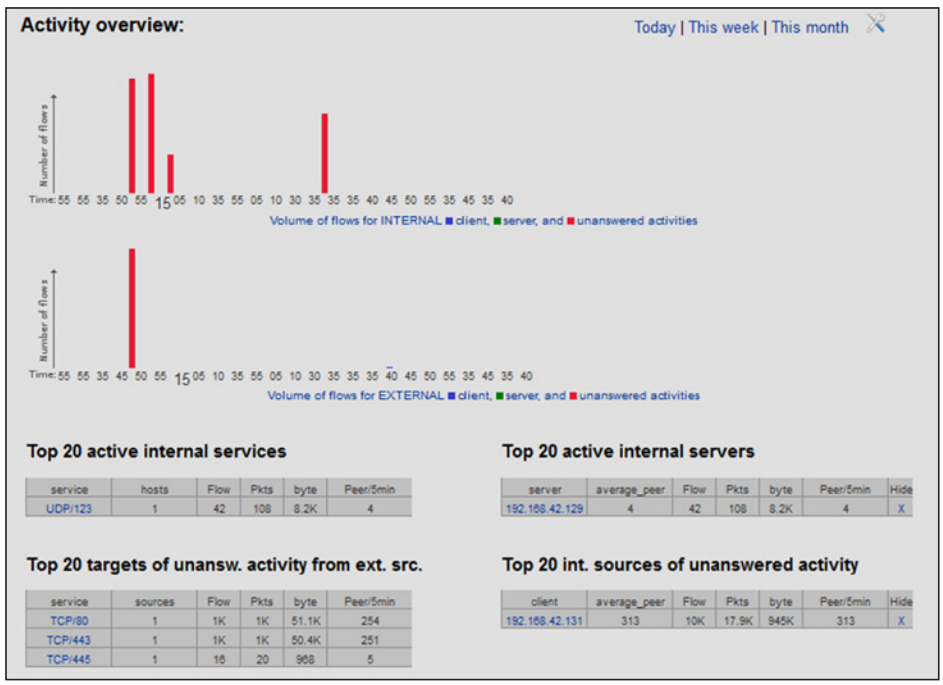

Figure 3 – Nfsight's Top 20

7 http://sourceforge.net/p/NfSen/bugs/43/.

8 http://sourceforge.net/p/nfsight/wiki/Installation/

9 http://steronius.blogspot.com/2013/05/install-nfsight-plugin-for-nfsen-on.html.

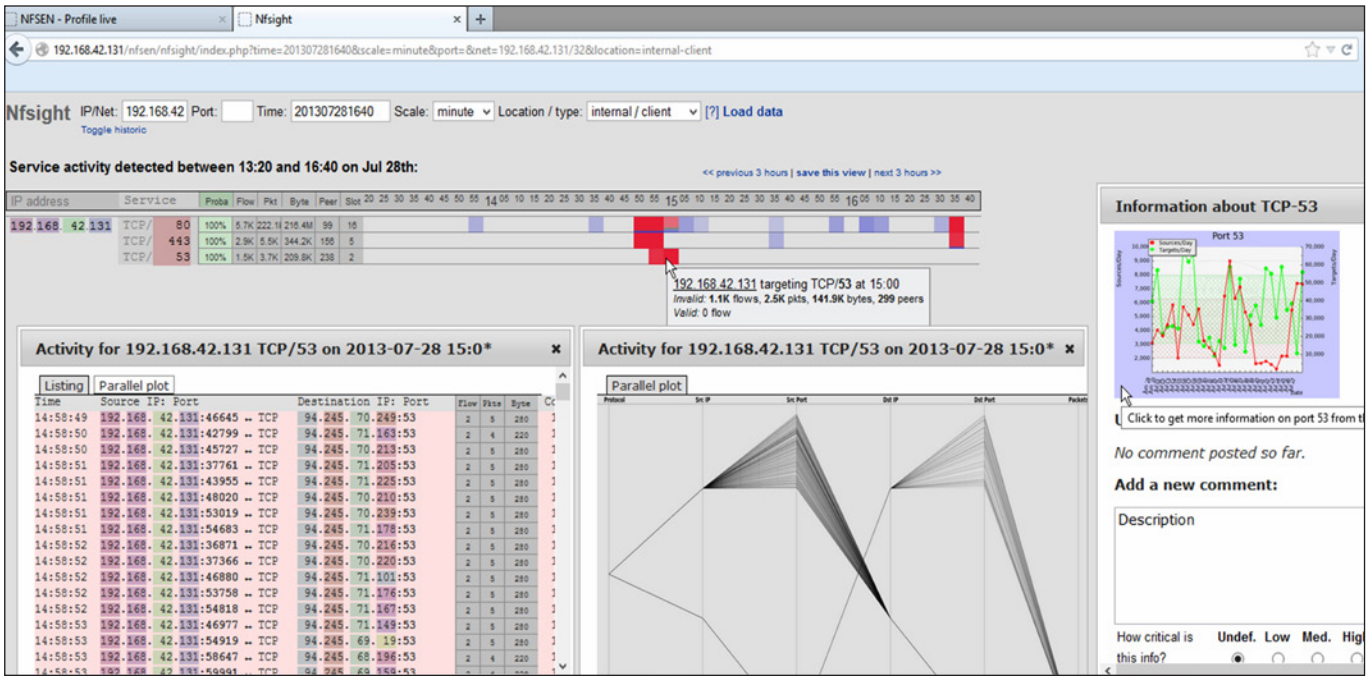

Figure 4 – Nfsight Activity Overview

minute increments as seen in figure 4, where you'll note all the IPs my internal hosts was scanning on port 53. You can also render a parallel plot (courtesy of PicViz installed earlier). Every IPv4 octet, port number, and service are hyperlinks to more flow data, so just keep on clicking. When you click a service port number and it offers you information about a given port, thanks to the SANS Internet Storm Center as you are directed to the ISC Port Report for that particular service when you click the resulting graph.

See? I told you it would be worth it.

All functionality references are available on the wiki; most importantly recognize that the color codes are red for unanswered scanner activity, blue for client activity, and green for server activity.

You can select *save this view* and create what will then be available as an event in the Nfsight database. I saved one from what you see in figure 4 and called it Evil DNS Egress. These can then be reloaded by clicking *Events* from the upper right-hand corner of the Nfsight UI.

Nfsight also includes a flow-based intrusion detection system called Nfids, still considered a work in progress. Nfids will generate alerts that are stored in a database and aggregated over time, and alerts that are recorded more than a given number of time are reported to the frontend. These alerts are generated based on five heuristic categories including: malformed, one-to-many IP, one-to-many port, many-to-one IP, and many-to-one port.

You can also manage your Nfsight settings from this region of the application including Status, Accounts, Preferences, Configuration, and Logs. You can always get back to the home page by simply clicking Nfsight in the upper-left corner of the UI.

As the feedback on the Nfsight SourceForge site says, "small and efficient and gets the job done."

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. Nfsight, as part of our C3CM concept, represents the first step (and does a heck of a good job doing it) of my C3CM process: identify. Next month we'll discuss the interrupt phase of C3CM using BroIDS and Logstash.

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

Cheers…until next month.

Acknowledgements

—Robin Berthier, Nfsight developer

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.