Shedding Light on Security Incidents Using Network Flows

GIAC (GCIH) Gold Certification

Author: Kevin Gennuso, kgennuso@gmail.com
Advisor: Rodney Caudle

Accepted:

Abstract

Knowing which hosts are involved in a security incident is vital information to any incident handler. The faster the attackers and their targets can be identified, the quicker the incident can be contained. Collecting this information from disparate logging systems can be difficult and time consuming, and capturing and storing every packet sent across the network for long periods of time isn't technically feasible in most cases. Fortunately, most modern networking hardware has the ability to track and export network flows. A well-tuned network flow collection gives the incident response team a clear view into all past and present conversations between hosts, allowing for fast identification of attacking or infected machines. Network flows can also demonstrate what "normal" and "abnormal" network conditions look like and can help identify outliers or potential data leaks. This capability can be a valuable tool used throughout the incident handling process to help bring clarity and visibility during the "fog of war".

1. Introduction

Incident handlers, and information security teams in general, face significant challenges when dealing with incidents in modern networks. Recent trends toward the consumerization of IT coupled with disappearing network borders and a dynamic threat landscape make identifying and responding to incidents an increasingly complex task. Without visibility into what is happening on their networks, an incident handler's goal of quickly containing and purging attacks against their systems can be very difficult. Flow data can be used as another tool in an incident handler's arsenal to help bring needed visibility to what is often an unclear security incident situation.

1.1. Network Flows – A Record of Who Called Who

A network flow is, in essence, a record of a given conversation between two hosts on a network. As one leading security researcher put it: "this information is much like a phone bill: you can't tell what was said during the conversation, but you can use it to prove who talked to who," (Anstee, 2011). Each flow contains the IP addresses of the source and destination hosts, the TCP/UDP ports they used to communicate, which router or switch interface the flow was seen on, and some other important information (Cisco, 2007). Cisco's NetFlow protocol is the most common and is also supported by Juniper; however, other vendors have adopted variants such as jFlow, cFlow, sFlow, and IPFIX that provide similar functionality. Most modern networking hardware supports some type of network flow collection (Hay, 2010).

Any network device configured to send network flows must have a place to send them. Typically, this is a server running flow collection and analysis software. There are a number of free and commercial flow collectors available on the market for various platforms. Some appliances such as SIEM devices or network analyzers such as Cisco's Network Analysis Module can also process network flows (Cisco, 2010). Selecting a suitable flow collector really depends on the needs of the incident handlers and of the organization as a whole. Flow collectors can be standalone systems that have the sole function of collecting and analyzing flow data, or they can be part of a larger system (such as an SIEM) that is used to correlate network and/or security events.

Kevin Gennuso, kgennuso@gmail.com

1.2. Advantages of Network Flows

There are all sorts of tools that can be used to monitor a network, from logging and bandwidth graphing to packet captures and deep inspection using an IDS or IPS. These technologies all have their place and are invaluable in their own ways; however, none of them offer the unique characteristics that network flows do. Network flow data should not be seen as a replacement for any of these monitoring or security technologies. Flows are an accompaniment to any existing set of monitoring tools. Some of the advantages of network flows are outlined below.

- Flows reveal detailed information about a particular network conversation without the technical challenges of sniffing and storing every packet. Similarly, someone looking at a network flow report will be able to glean useful data from it without investing the time it takes to analyze large packet captures.
- Flows are still useful even if the connection is encrypted. When looking at flow data, an incident handler can learn when and where the data went, how much and at what time, even without knowing the content.
- Flows are historical, allowing an organization to develop a good baseline to use for detecting anomalous (and possibly dangerous) traffic.
- Flow data are not easily scrubbed like a log file on a compromised server might be, and are typically generated by an uncompromised source that isn't a target in the attack (a network device such as a router or switch).
- Flow data can be taken from multiple points on the network, allowing for a distributed view. This can aid in correlating how an attack or infection began or how successful containment has been.
- Flow exports offer another layer of visibility and detection that is very easy to implement on both local and remote networks.

Kevin Gennuso, kgennuso@gmail.com

2. Network Flows in the Incident Handling Process

The SANS incident handling process consists of six (6) phases: Preparation, Identification, Containment, Eradication, Recovery, and Lessons Learned. Integrating network flow tools into this process is an easy task since the framework works well for any new tool in an incident handling team's toolbox. As stated previously, there are many different types of network flow protocols, as well as various collection and analysis software. This paper will generalize wherever possible, but examples of specific configurations will be given in terms of Cisco's NetFlow protocol and the free/open-source tools available for use with that protocol.

2.1. Preparing to utilize network flows

The first phase in the incident handling process involves preparing to deal with an incident. With regards to network flows, this is the phase where flows should be enabled and verified to be capturing the information needed to provide whatever visibility an incident handler requires. This is also the phase where a collection and analysis tool should be selected, installed, configured, and tuned for the environment in which it will be deployed. Once the entire system is in place and functional, it can be used to identify incidents or used in conjunction with other identification processes to aid the incident scenario at hand.

2.1.1. Selecting a flow collector

The first step in deploying a network flow system that can aide incident handlers is to select the collector and analysis system itself. There are many, many options available that can meet the requirements of most any environment. The main issues to take into consideration when selecting a flow collector include:

- Flexibility: What types/versions of network flows are supported? What operating systems does it run on?
- Scalability: How many flows can it handle per minute? Is it easy to add more capacity?
- Performance: How quickly does it generate reports? How much historical data can be saved?

- Supportability: Does the vendor provide help if required? In the case of an open source tool, is there an active community using the tool?
- Cost: Commercial or free? Cost tends to go up in line with how much flow data is being collected. Don't forget to factor in ease of deployment.

There is no "right" answer to any of these questions; the answers depend on the organization and its requirements and budget. It may make sense to deploy one of the many free tools as a pilot and then use the results produced by the system as leverage for purchasing a higher-end tool.

Ntop, SiLK, and flow-tools with FlowViewer are three very good free flow collection and analysis suites that can be used in conjunction with Cisco's NetFlow. Ntop works on many different operating systems and provides a quick way to stand up a flow collector, but it doesn't have the extensive historical capabilities out of the box that some other tools do. SiLK, part of the CERT NetSA Security Suite, is an advanced set of flow collection and storage tools maintained by Carnegie Mellon University's Software Engineering Institute. It is extremely powerful and is geared toward large-scale networks, but it is also the most complex of the three to deploy. FlowViewer, a NetFlow analyzer written by Joe Loiacono at NASA, is a front-end for Mark Fullmer's flow-tools. It requires a light-to-moderate amount of configuration, but in return it provides excellent capabilities for reporting on historical data. Due to their ease of deployment and historical capabilities, flow-tools combined with FlowViewer work great for an initial implementation. The examples below will focus on these two tools running on Linux.

Kevin Gennuso, kgennuso@gmail.com

2.1.2. Installation and configuration

Installing and configuring a flow collector will vary depending on the software. However, there are a few general steps that can be followed for any new install:

- Installation of the required packages
- Configuring the collector(s) to listen for network flows
- Ensuring that the system is listening for flows

In the case of FlowViewer, the installation requires a Web server with CGI capabilities, Perl and some supporting modules, and a few other packages (Loiacono, 2012). FlowViewer is a front-end to query the flow data, while flow-tools collects and stores the actual flow information. All dependencies for both packages will need to be installed for the system to function properly.

The documentation for FlowViewer and flow-tools is quite complete, so there is no need for a step-by-step install guide here. However, there are a few not-so-obvious details worth mentioning. First, when configuring the `/etc/init.d/flowcap` startup script, one flow-capture daemon is required for each device exporting flows. Consider this example:

```
    /usr/local/flow-tools/bin/flow-capture -p \
/var/flows/pids/flowtool.pid -w /var/flows/router1 -E5G -S3 0/0/3600
    /usr/local/flow-tools/bin/flow-capture -p \
/var/flows/pids/flowtool.pid -w /var/flows/router2 -E10G -S3 0/0/3601
```

Here, there are two instances of flow-capture configured. The one for *router1* listens on UDP 3600 and is set to keep 5GB of data, while the instance for *router2* listens on UDP 3601 and saves 10GB of historical information. Be sure to select an unused UDP port for flow-capture's use, and make sure to allow that port through any firewalls that might be in the way, including the one that might be running on the server itself.

The other item worth mentioning is the configuration file for FlowViewer (aptly named `FlowViewer_Configuration.pm`). Aside from setting the paths to the binaries and flow files, the single most important line is the `@devices` array. This is where the list of devices from `/etc/init.d/flowcap` come into play:

```
@devices    =     ("router_1","router_2");
```

This line is what allows the user to select the device they'd like to run a report against from the drop-down box in FlowViewer.

Once FlowViewer and its dependencies have been installed and their configuration complete, it is wise to ensure that the collector is listening for flows before moving forward. A simple `netstat -an` will tell the tale:

```
udp        0      0 0.0.0.0:3200                0.0.0.0:*
udp   109952      0 0.0.0.0:514                 0.0.0.0:*
udp        0      0 0.0.0.0:3600                0.0.0.0:*
udp        0      0 0.0.0.0:3601                0.0.0.0:*
udp        0      0 0.0.0.0:3800                0.0.0.0:*
```

Netstat shows the server listening on UDP 3600 and 3601, so it should be ready to receive flows from devices pointing toward it.

2.1.3. Generating flows on a router

Now that the flow collector is ready to receive flow data, it is time to configure the devices that will be sending the flows. With NetFlow on Cisco devices, the first step is to know which router interfaces are interesting from a flow collection perspective. For performance and data storage reasons, it is always best to only enable flow collection on the interfaces that pass the network traffic that will be interrogated by the flow reporting tools. A decision must also be made with regards to how much traffic NetFlow should watch on those interfaces. Ideally, information should be collected on every network conversation, but there may be cases where samples taken every so often (called sampled

flows) are sufficient. It is also important to enable flow collection on both the inbound and outbound interfaces to ensure the collector sees both sides of the conversation. Lastly, it is important to export flow data using a version of NetFlow that is supported by the collector. Version 5 is supported by nearly all of the free and commercial collectors available, while Version 9 supports newer technologies such as IPv6 and MPLS.

In this example, the Cisco device is a router with two interesting interfaces. Since the FlowViewer tool only supports Version 5 at this time, so a configuration snippet for a router with two interesting interfaces would look like this:

```
! Enter configuration mode

router1#conf t

! Set the interface IP address from which flow data will come from
! This does not have to be a loopback – any interface will do

router1(config)#ip flow-export source Loopback0

! Set the NetFlow version to 5

router1(config)#ip flow-export version 5

! Set the IP of the flow collector and the corresponding UDP port

router1(config)#ip flow-export destination 10.10.10.100 3600

! Enable flow collection on two GigabitEthernet interfaces

router1(config)#interface GigabitEthernet0/1
router1(config-if)#ip flow ingress
router1(config)#interface GigabitEthernet0/2
router1(config-if)#ip flow ingress
```

If the device were a Layer 3 switch, there are a few additional commands that must be entered to ensure that complete network flows are being captured.

```
! Enable the creation of flows on a per-VLAN basis

L3switch(config)#mls netflow interface

! Use the most specific flow mask. All flow entries will contain:
! Source and destination IP, port, protocol, and SNMP ifIndex of the
! source interface

L3switch(config)#mls flow ip interface-full
```

Kevin Gennuso, kgennuso@gmail.com

```
! Also create flow entries for flows that are Layer 2 only (bridged,
not routed)

L3switch(config)#ip flow ingress layer2-switched vlan 100-110
```

Once all network devices have been configured to export flows, it is a simple task to see if they are exporting flows as expected. The command `show ip flow export` will show all of the statistics regarding the export of flow data. Run the command a few times and check to see if the "flows exported" counter is increasing. If it is, then it can be assumed that the device is configured properly.

2.1.4. Verifying flow collection

The easiest way to ensure that flows are being received from a particular network device is to simply run a report after 15-30 minutes of flow data has been captured. A query for all traffic to or from a known chatty host will quickly show if flow collection is functioning properly for a given network device.

With FlowViewer, there is another way to verify this. Each collector writes to a set of files that correspond to a given network device. The location for these files is defined in `Flow.pm`. Each device has its own subdirectory that is further broken down by date. If data are being written to the flow files and they are expanding in size, then everything is working properly.

2.1.5. Establish a baseline

Now that the flow collector is functional and gathering flow data from core network devices, it is important for the incident handlers to understand what "normal" is when it comes to network traffic. As stated earlier, one of the big advantages of network flows is the capability to see into the past. While it is possible to run reports to see what the network normally looks like at any time, that really shouldn't be the focus during the heat of managing an incident. A bit of additional work during the Preparation phase will allow the handlers to direct their attention to dealing with the security breach in other phases.

To this end, occasional reports should be run during times where no security incidents are taking place, and these should be saved for future reference. Since network

traffic is dynamic and will change over time, so will the definition of "normal". Running these baseline reports should be an ongoing, routine process so that there is always a fresh view of what the network typically looks like. While it still may be necessary to run historical comparison reports during the heat of an incident, having a compilation of pre-run reports will save time when it is needed most. This practice also allows the incident handler to become familiar with how the system works, which will ultimately make it easier to use when it really counts.

2.2. Identifying incidents with network flows

As discussed in the SANS 504 training, the second phase of the incident handling process involves identifying incidents. This is where network flows really show their value. With flow data being collected from all critical points throughout the network infrastructure, an incident handler can use this information to initially identify or add further clarity to incidents that have already been declared. Network flows can help identify systems that are infected with malware or participating in a botnet. They can also reveal an attacker's targets, systems that are already compromised, and even the attackers themselves. Knowing as much information as possible about the systems involved in a security incident allows for a more focused approach to managing the incident. This also provides insight into how best to proceed with containment, eradication, and recovery efforts.

2.2.1. Hunting for bots

One of the biggest threats in modern information security is the rise of botnets. Botnets are flexible, extensible, remotely controlled, and put the power of thousands of computers into the hands of an attacker. They can be commanded to steal passwords or credit card numbers, launch Distributed Denial of Service attacks, or send spam on behalf of whoever pays the bot herder (F-Secure, 2012). They typically have a propagation component that provides a worm-like ability to spread using different infection vectors, and they always have some sort of command-and-control channel used by the attacker to issue instructions to his minions.

While bot-infected systems are extremely dangerous, they cannot hide from a network with properly deployed flow collection. Much like fingerprints at a crime scene,

all bots leave some telltale signs on the networks that host them. Incident handlers can use this information to identify bots on their network and quickly limit their effectiveness and ability to do damage.

One signature of a bot infestation is a sudden and substantial uptick in DNS requests (Villamarín-Salomón, Ricardo & Brustoloni, José Carlos (2008)). Most of the botnets today use DNS to find their command and control (C&C) servers. Many of those employ a technique called DNS fluxing, which allows the attacker to create dynamically generated domains for pointing bots to their C&C server (Yadav et al., 2012). This helps the attacker stay in control since the host and domain name constantly changes, making it difficult for defenders to blacklist or otherwise shut it down. Since the domain and host names are created randomly using an algorithm, a bot looking for its C&C server must run a large number of DNS queries as it guesses the server name and domain. The authors of newer bots have even gone so far as to hide the actual communication between bot and master within packets disguised as the DNS protocol itself (Lau, 2009). Fortunately, NetFlow sees and records all DNS requests, legitimate or otherwise, allowing a flow-savvy security team to use them as an indicator of a possible bot infection.

Requesting this information is pretty straightforward. A simple query for all hosts communicating with a destination UDP port of 53 will show every DNS request made on the network. In FlowViewer and most other flow reporting tools, a protocol must be specified using its IANA-assigned protocol number (in the case of UDP, protocol 17 is used). Given the amount of information that must be processed, a report like this can take a while, so be prepared to wait a bit.

Kevin Gennuso, kgennuso@gmail.com

FlowGrapher | Save Report
FlowViewer
POWERED BY FLOW-TOOLS
FlowTracker | Save Filter

```
     Report: Source/Destination IP                          Sort Field: 4
  Start Time: February 1, 2012 8:00:00 EDT                    End Time: February 1, 2012 9:00:00 EDT
     Device: router_1                                         Exporter:
     Source:                                               Destination:
 Source Port:                                       Destination Port: 53
  Source I/F:                                        Destination I/F:
   Source AS:                                         Destination AS:
   TOS Field:                                               TCP Flag:
  Include if: Any part of flow in Time Period            Protocols: 17
Lines Cutoff: 100                                       Octets Cutoff:
```

Source	Destination	Flows	Octets	Packets	Avg Rate(bps)
10.100.100.24	172.24.100.13	7821	3.82 MB	20841	2,304
10.100.100.24	172.24.100.11	6269	2.55 MB	19068	1,982
10.100.100.24	172.24.100.10	6120	2.21 MB	17322	1,718
10.100.100.72	172.24.100.10	4345	318.39 KB	4512	241
10.100.100.109	172.24.100.13	2232	293.12 KB	3158	205
10.100.100.33	172.24.100.13	2054	262.51 KB	2058	199
10.100.100.232	172.24.100.10	1458	187.04 KB	1470	141
10.100.100.215	172.24.100.10	359	183.71 KB	1203	139
10.100.100.142	172.24.100.13	1993	125.88 KB	2108	95
10.100.100.109	172.24.100.10	1495	91.11 KB	1522	69
10.100.100.72	172.24.100.11	1371	87.27 KB	1465	66
10.100.100.138	172.24.100.11	1289	79.60 KB	1334	60
10.100.100.53	172.24.100.13	1220	75.84 KB	1222	57
10.100.100.184	172.24.100.13	592	43.71 KB	592	33
10.100.100.88	172.24.100.10	594	41.59 KB	594	31
10.100.100.184	172.24.100.10	580	40.89 KB	583	31
10.100.100.33	172.24.100.11	566	35.13 KB	590	26
10.100.100.232	172.24.100.11	511	34.00 KB	512	25
10.100.100.53	172.24.100.10	532	33.50 KB	532	25
10.100.100.215	172.24.100.13	532	33.33 KB	532	25
10.100.100.142	172.24.100.10	244	31.02 KB	244	23
10.100.100.138	172.24.100.10	422	30.71 KB	422	23

FlowViewer report showing large number of DNS requests

As shown in the screen shot, there is one host in particular (10.100.100.24) that is generating a huge number of DNS requests. This should not be taken as an automatic indication of an infection; DNS activity is often heavier for systems that do a lot of name resolution (like root servers, forwarders, and monitoring systems). However, since it is very unusual for an end-user workstation to generate this many DNS requests in an hour, it is suspicious enough to the alert incident handler that the workstation at the top of the list should be investigated for other signs of malware.

Another sign that may indicate the presence of a bot on the network is any unauthorized connection attempt to Internet Relay Chat (IRC) servers. IRC, used primarily for party-line style chatting, is also a great vehicle for attackers to pass commands to the bots they control. The attacker needs only to choose an IRC server and create a channel for the bots to join. Once joined, the bots can receive commands on attacks to be performed or updates to the bots themselves (Jacob, 2009).

IRC isn't normally associated with normal business use and is typically disallowed by most organizations. That won't stop zombie PCs from attempting to connect to a botnet's C&C server. The default TCP port for IRC is 6667, but IRC servers

usually use ports 6660-6669. If the baseline reports created in the Preparation phase do not indicate the regular use of IRC on this network, any connection attempts to destination ports in this range should set off the alarm bells.

FlowGrapher
FlowViewer
POWERED BY FLOW-TOOLS
FlowTracker
Save Report
Save Filter

```
      Report: Source/Destination IP                          Sort Field: 4
  Start Time: February 1, 2012 8:00:00 EDT                     End Time: February 1, 2012 9:00:00 EDT
      Device: router_1                                          Exporter:
      Source:                                               Destination:
 Source Port:                                       Destination Port: 6667, 6668, 6669
  Source I/F:                                        Destination I/F:
   Source AS:                                         Destination AS:
   TOS Field:                                               TCP Flag:
  Include if: Any part of flow in Time Period             Protocols: 6
Lines Cutoff: 100                                       Octets Cutoff:
Source                          Destination               Flows              Octets             Packets            Avg Rate(bps)

10.100.100.24                   124.40.192.143            8318               9.81 MB            42139              3,504
10.100.100.24                   124.40.192.143            7241               8.15 MB            36124              2,942
10.100.100.24                   124.40.192.144            6120               4.34 MB            23561              2,021
```

FlowViewer report showing IRC channel joins

As seen in the report above, that same workstation from the previous report tried to talk to two systems in China on TCP ports 6667-6669, and it did so with gusto. Since that same workstation appears at the top of the DNS report, it's a pretty good indication that something nefarious is happening with that host.

Another excellent way of utilizing network flow data to identify bots is to mine for known C&C servers. There are some excellent resources available, such as those from Emerging Threats and the Shadowserver team, that publish updated lists of IP addresses that act or have acted as C&C servers for bots (Emerging Threats, 2012). These lists can then be queried against a flow collection to see if any hosts on the network have communicated with the malicious addresses. With a bit of scripting know-how, the process can be automated by pulling down fresh lists of malicious servers and using a command-line interface like flow-tools to run the reports. Using this method, an incident handler could be greeted every morning with an email listing potentially bot-infected systems on their network. A populated list is better than coffee, and provides insight into possible incidents that may not have been otherwise available. See *Appendix A* for a sample script.

The bot masters are aware of the network defenders' efforts to detect and destroy their creations, and they are actively working on a more decentralized model. The latest bot evolution includes the ability to receive commands via a peer-to-peer network. This

Kevin Gennuso, kgennuso@gmail.com

eliminates the inherent weaknesses associated with a centralized server. Luckily, researchers are actively working on botnet detection frameworks that utilize network flows and traffic profiles to identify this new, more stealthy type of bot. (Zeidanloo & Manaf, 2010).

Bots can also behave similarly to other types of malware, such as mass-mailers or worms that are attempting to spread to other hosts. These behaviors can also be detected by using network flows, which will be described in the next section in more detail.

2.2.2. Spotting other malware infestations

Bots attempting to phone home are not the only nasty critters that can be identified using flow data. PCs infected with a wide range of malware can also be found through the use of flow analysis.

Mass-mailing worms, such as the Melissa and ILOVEYOU viruses, became prevalent in the late 90s, but their functionality is still present in modern-day malware. They use SMTP to send large amounts of email, typically in an attempt to deliver spam or infect other systems. Reviewing network flow data can easily spot this activity. A query for all systems communicating with a destination TCP port of 25 will show any machine sending SMTP. Large amounts of SMTP traffic sourced from a system that is not a mail server or known sender of email should be considered suspicious. This kind of traffic can also indicate the presence of a bot that is sending spam on behalf of its bot herder (Jacob, 2009).

There are other worms that can be found using NetFlow, too. Some worms (like the modern Conflicker) spread via SMB, which communicates on TCP 139 and 445. Others, like the older Welchia, spread via RPC vulnerabilities using TCP 135 (Perriot, 2003). Still others spread using vulnerabilities in services that listen on less well-known ports (such as SQL Slammer on UDP 1434). The common link here is that all three examples use very distinct port numbers when propagating, leaving traces behind that NetFlow will record. Most worms, Trojans, and other malware leave some sort of telltale network pattern behind that can be detected even if no IDS or antivirus signature exists.

In the scenario below, users in one particular location have reported that their domain accounts are being locked out and that their PCs are running slow. This sound suspicious, but no IDS or antivirus alerts have been triggered. The incident handler decides to check SMB traffic coming into the user VLAN by running two reports:

```
      Report: Source/Destination IP                          Sort Field: 4
  Start Time: March 15, 2012 19:00:00                          End Time: March 15, 2012 19:15:00
      Device: core_L3_switch                                   Exporter:
      Source:                                             Destination: 10.200.100.0/24
 Source Port:                                        Destination Port: 445
  Source I/F:                                         Destination I/F:
   Source AS:                                          Destination AS:
   TOS Field:                                                TCP Flag:
  Include if: Any part of flow in Time Period               Protocols:
Lines Cutoff: 100                                       Octets Cutoff:
```

Source	Destination	Flows	Octets	Packets	Avg Rate(bps)
10.100.100.101	10.200.100.97	116	30.42 KB	522	276
10.100.100.101	10.200.100.95	116	30.42 KB	522	276
10.100.100.101	10.200.100.68	116	30.42 KB	522	276
10.100.100.101	10.200.100.65	116	30.42 KB	522	276
10.100.100.101	10.200.100.73	116	30.42 KB	522	276
10.100.100.101	10.200.100.90	116	30.42 KB	522	276
10.100.100.101	10.200.100.82	116	30.42 KB	522	276
10.100.100.101	10.200.100.88	116	30.42 KB	522	276
10.100.100.101	10.200.100.77	116	30.42 KB	522	276
10.100.100.101	10.200.100.99	116	30.42 KB	522	276
10.100.100.101	10.200.100.75	116	30.42 KB	522	276
10.100.100.101	10.200.100.71	116	30.42 KB	522	276
10.100.100.101	10.200.100.103	116	30.42 KB	522	276
10.100.100.101	10.200.100.114	116	30.42 KB	522	276
10.100.100.101	10.200.100.113	116	30.42 KB	522	276
10.100.100.101	10.200.100.96	116	30.42 KB	522	276
10.100.100.101	10.200.100.89	116	30.42 KB	522	276
10.100.100.101	10.200.100.104	116	30.42 KB	522	276
10.100.100.101	10.200.100.78	116	30.42 KB	522	276
10.100.100.101	10.200.100.72	116	30.42 KB	522	276
10.100.100.101	10.200.100.70	116	30.42 KB	522	276
10.100.100.101	10.200.100.64	116	30.42 KB	522	276
10.100.100.101	10.200.100.102	116	27.47 KB	464	250
10.100.100.101	10.200.100.93	104	26.86 KB	461	244
10.100.100.101	10.200.100.66	102	26.75 KB	459	243
10.100.100.101	10.200.100.79	104	26.05 KB	447	237
10.100.100.101	10.200.100.86	98	25.29 KB	434	230
10.100.100.101	10.200.100.76	98	24.88 KB	427	226
10.100.100.101	10.200.100.131	96	24.76 KB	425	225
10.100.100.101	10.200.100.92	96	24.76 KB	425	225
10.100.100.101	10.200.100.107	92	24.12 KB	414	219
10.100.100.101	10.200.100.108	94	23.83 KB	409	216
10.100.100.101	10.200.100.123	86	22.14 KB	380	201

All TCP 445 traffic inbound to VLAN 10.200.100.0/24

Kevin Gennuso, kgennuso@gmail.com

```
      Report: Source/Destination IP                          Sort Field: 4
  Start Time: March 15, 2012 19:00:00                          End Time: March 15, 2012 19:15:00
      Device: core_L3_switch                                   Exporter:
      Source: 10.200.100.0/24                             Destination:
 Source Port: 445                                    Destination Port:
  Source I/F:                                         Destination I/F:
   Source AS:                                          Destination AS:
   TOS Field:                                                TCP Flag:
  Include if: Any part of flow in Time Period               Protocols:
Lines Cutoff: 100                                        Octets Cutoff:
```

Source	Destination	Flows	Octets	Packets	Avg Rate(bps)
10.200.100.102	10.100.100.101	116	27.13 KB	290	246
10.200.100.104	10.100.100.101	116	24.07 KB	232	219
10.200.100.99	10.100.100.101	116	24.07 KB	232	219
10.200.100.95	10.100.100.101	116	24.07 KB	232	219
10.200.100.78	10.100.100.101	116	24.07 KB	232	219
10.200.100.113	10.100.100.101	116	24.07 KB	232	219
10.200.100.88	10.100.100.101	116	24.07 KB	232	219
10.200.100.89	10.100.100.101	116	24.07 KB	232	219
10.200.100.77	10.100.100.101	116	24.07 KB	232	219
10.200.100.70	10.100.100.101	116	24.07 KB	232	219
10.200.100.71	10.100.100.101	116	24.07 KB	232	219
10.200.100.65	10.100.100.101	116	24.07 KB	232	219
10.200.100.114	10.100.100.101	116	24.07 KB	232	219
10.200.100.96	10.100.100.101	116	24.07 KB	232	219
10.200.100.97	10.100.100.101	116	24.02 KB	232	218
10.200.100.72	10.100.100.101	116	24.02 KB	232	218
10.200.100.90	10.100.100.101	116	24.02 KB	232	218
10.200.100.64	10.100.100.101	116	24.02 KB	232	218
10.200.100.73	10.100.100.101	116	23.96 KB	232	218
10.200.100.68	10.100.100.101	116	23.85 KB	232	217
10.200.100.82	10.100.100.101	116	23.73 KB	232	216
10.200.100.103	10.100.100.101	116	23.62 KB	232	214
10.200.100.75	10.100.100.101	116	23.56 KB	232	214
10.200.100.93	10.100.100.101	102	21.17 KB	204	192
10.200.100.66	10.100.100.101	102	21.17 KB	204	192
10.200.100.79	10.100.100.101	98	20.34 KB	196	185
10.200.100.86	10.100.100.101	96	19.92 KB	192	181
10.200.100.76	10.100.100.101	94	19.51 KB	188	177
10.200.100.92	10.100.100.101	94	19.51 KB	188	177
10.200.100.131	10.100.100.101	94	19.46 KB	188	177
10.200.100.107	10.100.100.101	92	19.09 KB	184	173
10.200.100.108	10.100.100.101	90	18.68 KB	180	170
10.200.100.123	10.100.100.101	84	17.43 KB	168	158

All TCP 445 traffic outbound from VLAN 10.200.100.0/24

In a typical corporate network with lots of Windows machines, traffic on TCP 445 between many workstations and some known number of file servers is typical and expected. However, in this example, there is one PC attempting to reach that port on large numbers of other workstations. Since users in this scenario do not have the ability to share files from their local workstations, there would be no real reason to see so much activity on TCP 445. The small number of packets and how they are dispersed across the entire subnet should also raise a flag. This type of pattern could be seen if a worm or botnet tried to spread, or if an attacker was attempting brute force logins on multiple targets. A report like this is not absolute proof of malicious activity, but it is strange enough to warrant further investigation.

Kevin Gennuso, kgennuso@gmail.com

2.2.3. Finding attackers and their targets

Potential security events can be escalated to an incident handling team by a myriad of different sources. While flow data can be very useful in helping initially identify potential security threats, they can also help provide a clear picture during an already-declared incident identified through one of these other sources. Properly-deployed flow collection can help an incident handling team understand the full scope of the attack and allow them to move into the containment phase with the information they need to be the most effective.

In the next scenario, an incident handler has already identified an attack that is already underway due to alerts from the NIDS in front of an Internet-facing firewall. The NIDS has alerted on attacks against one target, but the incident handler is concerned that there may be others. Since the handler already has the source IP address of the attacker from the NIDS (203.x.x.127), it makes running a NetFlow report pretty simple.

```
       Report: Source/Destination IP                          Sort Field: 4
   Start Time: March 15, 2012 08:00:00                          End Time: March 15, 2012 08:15:00
       Device: border_router                                     Exporter:
       Source: 203.xxx.xxx.127                                Destination:
  Source Port:                                           Destination Port:
   Source I/F:                                            Destination I/F:
    Source AS:                                             Destination AS:
    TOS Field:                                                   TCP Flag:
   Include if: Any part of flow in Time Period                  Protocols:
 Lines Cutoff: 100                                         Octets Cutoff:
Source           Destination      Flows              Octets              Packets             Avg Rate(bps)

203.xxx.xxx.127  12.xxx.xxx.145   29                 86.26 KB            1602                 0
203.xxx.xxx.127  12.xxx.xxx.121   14                 21.26 KB            431                  0
203.xxx.xxx.127  12.xxx.xxx.118   14                 21.26 KB            430                  0
203.xxx.xxx.127  12.xxx.xxx.119   14                 21.20 KB            407                  0
203.xxx.xxx.127  12.xxx.xxx.124   14                 21.07 KB            389                  0
203.xxx.xxx.127  12.xxx.xxx.144   14                 21.07 KB            389                  0
203.xxx.xxx.127  12.xxx.xxx.147   14                 21.02 KB            371                  0
203.xxx.xxx.127  12.xxx.xxx.142   14                 20.81 KB            354                  0
203.xxx.xxx.127  12.xxx.xxx.146   14                 20.64 KB            334                  0
```

One attacker, multiple targets

As shown above, the attacker was indeed communicating with other hosts on that same network segment at around the same time as the known target. Perhaps the attacker was looking for vulnerable services, or perhaps they had already compromised the other systems using attacks not known to the NIDS. The fact that a known bad guy was attempting to communicate with more than just one system expands the scope of the Identification phase, if even just to ensure that the other systems were not compromised.

Kevin Gennuso, kgennuso@gmail.com

This same methodology can be applied in the other direction: finding all attackers in a coordinated attack against a target. One of the more common attacks in this category is known as a Distributed Denial of Service attack, where multiple systems send garbage traffic (such as SYN packets) in an attempt to overwhelm and exhaust the resources of a given target. Finding the attackers can be a more daunting, and sometimes, less fruitful task, especially against public-facing servers that already see lots of traffic from large numbers of remote systems. Still, there are still a few NetFlow tricks that can help a savvy incident handler.

```
       Report: Source/Destination IP                          Sort Field: 4
   Start Time: March 15, 2012 18:00:00                          End Time: March 15, 2012 18:15:00
       Device: router_2                                          Exporter:
       Source:                                               Destination: 12.xxx.xxx.33
  Source Port:                                           Destination Port: 80
   Source I/F:                                            Destination I/F:
    Source AS:                                             Destination AS:
    TOS Field:                                                   TCP Flag: 2
   Include if: Any part of flow in Time Period                  Protocols:
 Lines Cutoff: 100                                         Octets Cutoff:
Source           Destination      Flows              Octets              Packets             Avg Rate(bps)

24.xxx.xxx.12    12.xxx.xxx.33    409460             17.25 MB            409460              160,140
58.xxx.xxx.63    12.xxx.xxx.33    409455             17.22 MB            409455              160,138
111.xxx.xxx.177  12.xxx.xxx.33    341216             14.35 MB            341216              133,448
112.xxx.xxx.42   12.xxx.xxx.33    341201             14.33 MB            341202              133,446
36.xxx.xxx.239   12.xxx.xxx.33    272973             11.51 MB            272973              106,760
12.xxx.xxx.31    12.xxx.xxx.33    5                  600                 10                  5
208.xxx.xxx.90   12.xxx.xxx.33    3                  360                 6                   3
194.xxx.xxx.185  12.xxx.xxx.33    2                  240                 4                   2
```

Multiple attackers performing a SYN flood

Any decent flow analysis tool will allow the handler to hone in on network conversations with certain TCP flags set. In the case of a DDoS attack, it would be interesting to see which systems are sending the most SYN packets. FlowViewer and flow-tools require the combined decimal value of the TCP flags requested, so the value of 2 is passed for this type of query (Pierky, 2010). The report above shows clearly that there are five hosts generating huge bursts of SYN packets headed toward a web server. The matching flows and packets number is a great indicator of a SYN flood; normal traffic will have many more SYN packets in a single flow, not just one. While further investigation should be done to confirm that this traffic is indeed malicious, it is a good first step in identifying potential attackers and their attack methods.

The same technique of using TCP flags can be used to find SYN scans – even ones that occur over long periods of time. A stealthy attacker will quietly map a network rather than noisily throw piles of SYN packets at it. While an IDS or firewall may not

trigger on just a few SYN packets per minute, a flow report will catch them. Using this method, worms that spread using scanning techniques can also be found and traced to their source (Gong, 2010).

2.2.4. Pinpointing potential data leaks

As stated earlier, a collection of network flows is like a record book of all conversations between hosts on a network. As with any historical data, this information can be used to spot trends and help paint a picture of what "normal" network activity looks like. It can also help a keen incident handler identify abnormalities that may indicate a breach of security.

In the next scenario, the information security team has been alerted to some terminations that are being planned for two employees who are suspected of stealing company information. Human Resources has provided the names of these employees, and a quick review of their access reveals that they do indeed have elevated privileges. They have asked that extra scrutiny be applied to these individuals to ensure that any potential data leak be detected, but they do not want to alert the individuals to this increased surveillance.

Nothing suspicious has been detected while the employees are in the office, so a report is run against the subnet where remote access VPN tunnels terminate (users are assigned an IP in the 10.222.100.0/24 range):

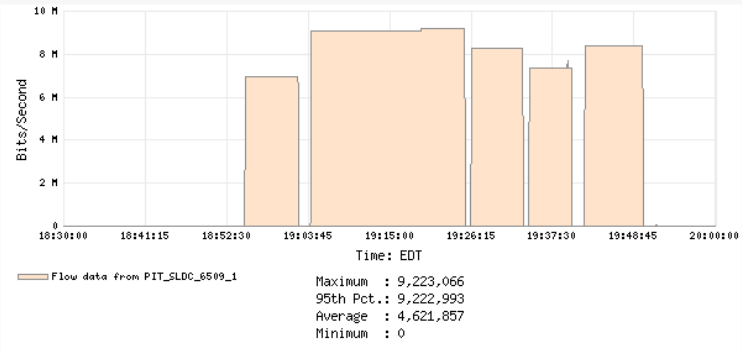
```
       Report: Source/Destination IP                          Sort Field: 4
   Start Time: March 15, 2012 18:30:00                          End Time: March 15, 2012 20:00:00
       Device: core_L3_switch                                   Exporter:
       Source:                                             Destination: 10.222.100.0/24
  Source Port:                                        Destination Port:
   Source I/F:                                         Destination I/F:
    Source AS:                                          Destination AS:
    TOS Field:                                                TCP Flag:
   Include if: Any part of flow in Time Period               Protocols:
 Lines Cutoff: 100                                        Octets Cutoff:
Source          Destination      Flows              Octets             Packets            Avg Rate(bps)

10.99.1.30      10.222.100.8     305                2.91 GB            2489989            4,622,198
10.50.50.98     10.222.100.37    6269               2.55 MB            19068              1,982
10.50.50.124    10.222.100.14    4345               318.39 KB          4512               241
10.50.50.22     10.222.100.37    116                27.13 KB           290                246
10.50.50.22     10.222.100.14    92                 19.09 KB           184                173
```

Report of VPN user activity for a 90-minute timeframe

Interestingly, there is one standout in the crowd, and it shows large amounts of data (almost 3GB) being transferred from an IP on the protected file share subnet (10.99.1.0/24) to a VPN user. This is especially bizarre since only admins with reserved

addresses can reach through the firewall directly to these sensitive servers. Another report is run using FlowGrapher, which is part of FlowViewer and can better visualize flows and when they occur:

```
      Report: Flow Graph Bits/Second                    Sample Time: 5
  Start Time: March 15, 2012 18:30:00                      End Time: March 15, 2012 20:00:00
      Device: core_L3_switch                                Exporter:
      Source: 10.99.1.30                               Destination: 10.222.100.8
 Source Port:                                     Destination Port:
  Source I/F:                                      Destination I/F:
   Source AS:                                       Destination AS:
   TOS Field:                                             TCP Flag:
  Include if: Any part of flow in Time Period            Protocols:
Detail Lines: 2000                                 Graph Multiplier: 1
```



Start	End	Len	Source Host	Port	Destination Host	Port	Total Bytes	Mbps
18:51:40	19:06:42	902.4	10.99.1.30	3389	10.222.100.8	61650	2,519,975	0.022
18:55:01	18:55:01	1.0	10.99.1.30	137	10.222.100.8	63689	203	0.002
18:55:01	18:55:01	1.0	10.99.1.30	139	10.222.100.8	61654	112	0.001
18:55:01	18:55:01	1.0	10.99.1.30	445	10.222.100.8	61653	60	0.000
18:55:01	19:02:24	443.3	10.99.1.30	139	10.222.100.8	61655	384,115,734	6.931
19:03:16	19:03:16	1.0	10.99.1.30	139	10.222.100.8	61655	190	0.002
19:04:09	19:19:16	906.4	10.99.1.30	139	10.222.100.8	61655	1,025,737,422	9.053
19:06:23	19:06:23	1.0	10.99.1.30	8	10.222.100.8	0	46	0.000
19:06:43	19:18:15	691.6	10.99.1.30	3389	10.222.100.8	61650	1,424,270	0.016
19:19:16	19:25:26	370.1	10.99.1.30	139	10.222.100.8	61655	426,692,585	9.223
19:26:16	19:33:26	429.7	10.99.1.30	139	10.222.100.8	61655	445,503,061	8.294
19:28:16	19:28:16	1.0	10.99.1.30	3389	10.222.100.8	61650	1,140	0.009
19:33:07	19:33:37	30.0	10.99.1.30	8	10.222.100.8	0	34,914	0.009
19:33:50	19:44:04	614.0	10.99.1.30	3389	10.222.100.8	62046	376,439	0.005
19:33:53	19:33:53	1.0	10.99.1.30	3389	10.222.100.8	61650	217	0.002
19:34:18	19:34:18	1.0	10.99.1.30	139	10.222.100.8	62135	60	0.000
19:34:18	19:34:18	1.0	10.99.1.30	445	10.222.100.8	62134	158	0.001
19:34:18	19:40:11	353.1	10.99.1.30	445	10.222.100.8	62136	324,642,942	7.356
19:39:38	19:39:38	1.0	10.99.1.30	80	10.222.100.8	62352	60	0.000
19:39:38	19:39:39	1.3	10.99.1.30	80	10.222.100.8	62351	195,706	1.223
19:41:52	19:41:52	1.0	10.99.1.30	8	10.222.100.8	0	46	0.000
19:41:53	19:41:53	1.0	10.99.1.30	80	10.222.100.8	62351	46	0.000
19:41:58	19:50:02	484.1	10.99.1.30	445	10.222.100.8	62136	508,091,730	8.397

FlowGrapher report for same 90-minute timeframe

There are the large file transfers, along with what appears to be an RDP session. A review of the VPN logs shows that one of the suspected employees is using IP address 10.222.100.8. It appears that the data leaker has been caught in the act.

Again, this is not decisive proof that data was stolen. Network flows only track the fact that a conversation occurred, not the details of what was discussed. Still, this large data transfer outbound to a non-corporate host on the Internet should set off enough

alarm bells to warrant further investigation. Being able to see that this transfer occurred would not have been possible had there not been a well-tuned flow collection system in place. Commercial software vendors have taken notice of this distinct advantage that NetFlow has when identifying data leaks and have begun to incorporate this functionality into their own collector technology (Lancope, 2010).

2.3. Containment efforts with network flows

Having visibility into the traffic flows throughout the network is just as important during the containment phase as it is in initially identifying an incident. The reasoning here is that flow data can show an incident handler if containment efforts are having an impact or if the incident is expanding in scope. Flows can also reveal information essential for containment: which addresses or ports should be firewalled, which DNS names should be blackholed, and which network segments require the incident response team's attention.

In a scenario where malware such as a botnet or worm is spreading, it can be difficult to determine exactly how far the infestation has spread or if it was contained before infecting other systems. As it turns out, the same flow analysis techniques used in the identification phase can be used here to determine containment progress. As the incident response team report back to the incident handers with updates on containment efforts, the incident handlers can run flow reports to see if the expected containment result has been achieved. Are the hosts that were originally infected still attempting to propagate? Are any of the hosts that the infected machine talked to now exhibiting behavior similar to what was seen from the infected system? These are the questions that the NetFlow reports run in this phase should try to answer.

Flow data might also reveal that TCP 6667 should be blocked to prevent a bot from phoning home. Or perhaps every infected machine is grabbing new data from an unusual external host, suggesting we should firewall that IP as a part of containment.

```
      Report: Source/Destination IP                          Sort Field: 4
  Start Time: March 15, 2012 21:00:00                          End Time: March 15, 2012 21:15:00
      Device: border_router                                     Exporter:
      Source:                                               Destination: 10.111.48.0/22
 Source Port:                                           Destination Port:
  Source I/F:                                            Destination I/F:
   Source AS:                                             Destination AS:
   TOS Field:                                                  TCP Flag:
  Include if: Any part of flow in Time Period                 Protocols:
Lines Cutoff: 100                                          Octets Cutoff:
Source          Destination      Flows              Octets              Packets             Avg Rate(bps)

60.255.65.17    10.111.49.102    4                  3.26 MB             2648                30,355
60.255.65.17    10.111.48.217    4                  3.26 MB             2648                30,355
60.255.65.24    10.111.50.82     3                  3.21 MB             2622                30,301
60.255.65.19    10.111.49.114    3                  3.21 MB             2620                30,300
```

FlowViewer report showing downloads from China

In this report, there are four machines on the workstation subnet that were originally infected but reported as cleaned by the incident responders. It seems that they are now very busy downloading large amounts of data from three hosts on the same subnet in China. A quick look at the baseline reports that were generated in the Preparation phase might reveal that workstations rarely communicate with networks in Asia. In that case, this activity would be considered suspicious, warranting additional containment actions such as a firewall rule or DNS blackhole.

Likewise, in a situation where an attacker has compromised a target system, network flows can also help with containment efforts. A few quick queries can reveal if the attacker is performing ongoing attacks or attempting to own other systems. Perhaps the attacker has moved to a proxy, using a different source address to perform the attacks. Queries for network flows involving the targeted system would reveal this. A proper report will also show if any internal systems suddenly called back to the attacker, a sign that could mean a more widespread compromise.

2.4. Network flows in the Eradication phase

In the Eradication phase, the incident handling team takes on a janitorial role and begins cleaning up the mess. NetFlow can be useful here as well. As incident responders report back to the handler with updates on cleanup progress, flow reports can be generated to verify their work. Are there still systems beaconing to that firewalled IP in China? Has the volume of DNS traffic returned to normal? These questions can be answered using the same types of flow reports that were used during the previous phases.

Kevin Gennuso, kgennuso@gmail.com

It is also worthwhile to examine the flow reports for gaps. If there were any network segments that were invisible due to lack of flow data, the network engineering team should be engaged to enable flow exports on the corresponding Layer 3 devices. Improving visibility means improving defenses. Seeing as much of the threat landscape as possible is crucial at this phase of the incident.

2.5. Recovering systems with network flow visibility

As the incident moves into the Recovery phase, NetFlow continues to be extremely useful for insight into what is happening on the network. Hosts involved in the incident are brought back into service at this time, so the incident handling team must watch closely to ensure this goes smoothly. Special attention should be paid to the systems that were compromised. In the case of a malware infection, the goal is to ensure that the hardened systems (complete with updated antivirus definitions) are not re-infected and do not attempt to call home to their C&C servers. Likewise, in a situation where a box was actively owned, it is likely that the original attacker or his friends will attempt to regain control of the system that once belonged to them. Using network flow reports in conjunction with other tools like intrusion detection and centralized logging, the recovery team can be alerted if the recovery team if the attacker has returned.

Incident handlers should not limit this phase to a few hours and should take advantage of the historical capabilities that come with a comprehensive flow collection system. Depending on the severity and breadth of the compromise, reports should be run for the next few days, weeks, or even months to ensure that the attacker has not launched additional salvos or to make certain that no additional systems have been infected. How often and how many reports should be run is ultimately up to the incident handling team. It is wise to err on the side of paranoia versus assuming the network is completely hardened and impervious to future attacks.

2.6. Discussing flows in the Lessons Learned phase

With the incident in its sixth and final phase, it is time to evaluate how well the incident response process worked and what improvements can be made. There are a few questions that should immediately come to mind with regards to the NetFlow system

itself. Were there any reports that were particular useful? Were all gaps in coverage addressed? Did the system perform well enough to handle the size and scope of the reports needed during the incident? Any improvements that can be made to reporting and the NetFlow system itself should be identified here.

Flow information can also be useful is for evaluating other aspects of the incident response process. Perhaps there are some reports that should be run on a daily or even hourly basis that can help identify potential incidents more quickly. Consider the use network flows to determine how long it took for proper containment and to show if eradication was successful on the first pass. Discuss improvements that can be made to the process, using flow reports as a way to measure effectiveness. There should be a lot of ideas after the first time NetFlow reports are used during an incident, so make sure to capture and integrate them into the process.

Kevin Gennuso, kgennuso@gmail.com

3. Conclusion

Incident handlers, and information security teams in general, face significant challenges when dealing with incidents in modern networks. Recent trends toward the consumerization of IT coupled with disappearing network borders and a dynamic threat landscape make identifying and responding to incidents an increasingly complex task. Without visibility into what is happening on their networks, an incident handler's goal of quickly containing and purging attacks against their systems can be very difficult. Flow data can be used as another tool in an incident handler's arsenal to help bring needed visibility to what is often a foggy security incident situation.

Kevin Gennuso, kgennuso@gmail.com

4. Appendix A

Below is a script that will retrieve the latest list of botnet command-and-control servers (as identified by the Shadowserver team) from the Emerging Threats web site and run them through a collection of flows created by flow-capture. The script uses a few of the other binaries that come with the flow-tools suite.

Kevin Gennuso, kgennuso@gmail.com

```
# !/bin/sh
#
# Quick and dirty script for finding flows matching Emerging Threats
# C&C block list
#
#
# Set paths to important files here
#
# ftpath = /path/to/flow-tools/binaries
# $flows = /path/to/completed/flows
#
ftpath=/usr/local/flow-tools/bin
flows=/var/flows/completed/router_1/2012/
#
#
# Grab the latest block list from Emerging Threats and parse
# into a format flow-filter can use (which happens to be a Cisco
# standard-type (1-99) access-list
#
# This only creates a filter for the C&C servers.  If you also want the
# dropper networks, comment the next section and uncomment the
# following section.
#
# C&C only:
wget -q -O - http://rules.emergingthreats.net/fwrules/\
emerging-PIX-CC.rules | sed '/#/d' | \
sed 's/ET-cc deny ip /standard ET permit /' | \
sed '/access-list ET-drop/d' | sed 's/any//' | sed '/^$/d' | \
sed '/access-list ET-cc/d' | sed 's/^/ip /' > /tmp/ccfilter
#
# C&C and dropper networks:
#wget -q -O - http://rules.emergingthreats.net/fwrules/\
#emerging-PIX-CC.rules | sed '/#/d' | \
#sed 's/ET-cc deny ip /standard ET permit /' | \
#sed 's/ET-drop deny ip /standard ET permit /' | sed 's/any//' | \
#sed '/^$/d' | sed '/access-list ET-cc/d' | \
#sed 's/^/ip /' > /tmp/ccfilter
#
#
# Now run the flows through the filter and print any hits.  The
# argument -DET uses the list as a destination filter.  Change to -SET
# for source address.
#
$ftpath/flow-cat $flows | $ftpath/flow-filter -f/tmp/ccfilter -DET \
-i1,2 | $ftpath/flow-print
#
#
#
# that’s all folks
```

Kevin Gennuso, kgennuso@gmail.com

Here is some sample output from this script. It shows a number of hosts on the network communicating with known C&C servers. Most of the communication is occurring over HTTP or HTTPS ports (80 and 443), but there is also one IRC channel join identified at the bottom.

srcIP	dstIP	prot	srcPort	dstPort	octets	packets
10.200.222.12	208.87.35.105	6	45956	80	578	5
10.200.222.131	64.202.107.109	6	27184	443	624	6
10.200.222.43	208.91.197.54	6	54998	80	921	5
10.200.222.12	208.87.35.105	6	61500	80	682	5
10.200.222.38	199.59.241.230	6	35133	443	630	5
10.200.222.12	208.87.35.105	6	49013	80	772	5
10.200.222.87	208.87.35.105	6	34909	80	963	5
10.200.222.12	208.87.35.105	6	23868	80	945	6
10.200.222.12	208.87.35.105	6	24371	80	770	5
10.200.222.87	208.87.35.105	6	31416	80	750	7
10.200.222.217	208.91.197.133	6	49506	80	1540	6
10.200.222.217	208.91.197.133	6	17518	80	1729	5
10.200.222.132	208.91.197.133	6	40609	80	1729	5
10.200.222.104	108.61.240.240	6	3072	6667	40	1

Kevin Gennuso, kgennuso@gmail.com

5. References

Anstee, Darren (2011, June 15). *FIRST2011: Listening to the network: Leveraging Network Flow Telemetry for Security Applications*. Referenced from: http://www.cupfighter.net/index.php/2011/06/first2011-listening-to-the-network/

Cisco (2007). *Introduction to Cisco IOS NetFlow – A Technical Overview*. Retrieved from: http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6555/ps6601/prod_white_paper0900aecd80406232.html

Cisco (2010). *Cisco Network Analysis Module Deployment Guide*. Retrieved from: http://www.cisco.com/en/US/prod/collateral/modules/ps2706/white_paper_c07-505273.html

Emerging Threats (2012). *Emerging Threats Rules*. Retrieved from: http://rules.emergingthreats.net/fwrules/emerging-PIX-CC.rules

F-Secure (2012). *About Botnets*. Retrieved from: https://www.f-secure.com/en/web/labs_global/articles/about_botnets

Gong, Yiming (2010, November 2). *Detecting Worms and Abnormal Activity with NetFlow*. Retrieved from: http://www.symantec.com/connect/articles/detecting-worms-and-abnormal-activities-netflow-part-2

Hay, Andrew (2010, October 10). *Monitoring with Network Flow Technology*. Retrieved from: http://www.darkreading.com/blog/228200581/monitoring-with-network-flow-technology.html

Lancope (2010, March 23). *Lancope Introduces Data Loss Alarming in NetFlow-based StealthWatch System*. Retrieved from: http://www.lancope.com/blog/lancope-introduces-data-loss-alarming-in-netflow-based-stealthwatch-system/

Lau, Hon (2009, June 29). *DNS Botnet Phun*. Retrieved from: http://www.symantec.com/connect/blogs/dns-botnet-phun

Loiacono, Joe (2012). *FlowViewer Version 3.4 Release Notes*. Retrieved from: http://ensight.eos.nasa.gov/FlowViewer/

Kevin Gennuso, kgennuso@gmail.com

Jacob, Don Thomas (2009, September 4). *Stopping the Bots with NetFlow and NetFlow Analyzer*. Retrieved from: https://blogs.manageengine.com/netflowanalyzer/2009/09/04/stopping-the-bots-with-netflow-and-netflow-analyzer-part-2

Perriot, Frederic (2003). *W32.Welchia.Worm | Symantec*. Retrieved from: http://www.symantec.com/security_response/writeup.jsp?docid=2003-081815-2308-99

Villamarín-Salomón, Ricardo & Brustoloni, José Carlos (2008). *Identifying Botnets Using Anomaly Detection Techniques Applied to DNS Traffic*. Retrieved from: http://www.cs.pitt.edu/~jcb/papers/ccnc2008.pdf

Yadav, Sandeep, Reddy, Ashwath Kumar Krishna, Reddy, A.L. Narasimha, Ranjan, Supranamaya (2012, February). *Detecting Algorithmically Generated Domain-Flux Attacks with DNS Traffic Analysis*. Retrieved from: http://ee.tamu.edu/~reddy/papers/tnet12.pdf

Zeidanloo, Hossein Rouhani & Manaf, Azizah Bt Abdul (2010, March). *Botnet Detection by Monitoring Similar Communications Patterns*. Retrieved from: https://sites.google.com/site/ijcsis/volume-7-march-2010

Kevin Gennuso, kgennuso@gmail.com