Global Information Assurance Certification Paper

GIAC Certification

GSEC Practical Assignment

Version 1.4b Option 1

Logging Cisco Switches On Your LAN: Another Layer of Security

by

Joshua Miller

Abstract:

Smart network administrators know that logging of servers is important. Sending these logs to a syslog server is even more important, because it will be evidence of an intrusion, as well as backing up the logs that are running on your servers. But logging the log files on your Cisco switches offers another insight into what means the intruder used to compromise your system.

In this paper, we'll look at sending your Cisco switch logs to a centralized syslog server running Red Hat Linux, and why logging your network infrastructure is just another layer of security defense. By doing so, you can review your logs quickly and easily to find nefarious activities on the network you administer. While Cisco LAN switches are not common targets by intruders, they can be a key witness to what the intruder may have done.

Defense In Depth

If you've attended any of Eric Cole's classes at a SANS conference, you'll know exactly what I'm talking about. In fact, the above statement is still ringing in my ears. When someone talks about "Defense in Depth", they mean that security infrastructure should consist of many layers. Each layer has it's own purpose, and not all layers are meant to stop an intruder, because network attackers are always finding new ways to penetrate a network. Trying to find every hole would be an impossible task.

Logging is just such a layer. While it does not prevent an attacker from entering your system, it does record what malicious activity may be occurring. Logging allows the administrator to see what system or systems have been affected, what the attacker may have been looking for, and what tools the attacker may have used. In some ways, logging may be more important that prevention.

That last statement may seem a bit bizarre, but imagine this: you approach your management after a suspected network penetration and say "I think our network has been hacked." Not the sort of statement any security administrator likes to make to their superiors. The first question is usually "How did they get in" followed by "so what did they do". You can already see that it's somewhat of a moot point to discuss how the attacker got in. Certainly closing that vulnerability that allowed the attacker access in the first place is important, but it's somewhat analogous to closing the barn door after the horses were stolen. The large task at hand is to begin investigating what the attacker may have done while inside your network. If your reply to management is to shrug your shoulders and say, "I don't know what systems/data has been affected", you may want to start brushing up on that resume. If you don't know what systems or data have been affected, or have absolutely no way of finding out, your data integrity just dropped to zero. Even if the attacker was some curious person who just decided to get into your network and have a look around and not touch anything.

So why should we log network infrastructure devices that aren't considered key devices? What are "key" security devices on your network? Key devices would include your firewalls and edge routers. Remember, we're talking about layers here. Granted, Cisco switches do not hold vital data the way a server does. They don't generally filter network traffic the way that a router or firewall does. Switches aren't usually the primary target serious hackers look for when they're trying to gain a foothold inside of a network to plant a root kit, backdoor or Trojan. Some hackers may target a switch to cause a denial of service attack, but experienced hackers who are involved in corporate espionage or grudges against a company don't want to be detected. They may return to the same network over and over again. So why gather the logs from these devices?

Exactly the same reason mentioned above – because it's another layer. In the initial scanning phase, unless hackers already have some knowledge of your network, they will scan your IP addresses scope looking for active devices on the network. Once these active devices are spotted, then the attacker would begin systematically scanning these devices with an operating system (OS) fingerprinting tool. This allows the hacker to see what exact devices are running on the network and which types of vulnerabilities may be present with particular systems. After developing the list of what operating systems your devices are running, the hacker would target what they're looking for, such as company payroll or a proprietary design document. Do switches have such data? No. But your switches are an innocent bystander to this process the whole time, watching the initial IP address scan, the OS fingerprint scan, and possibly a login attempt to one of the switches.

What if I was to say that even with the OS scan and the port scan, it was not detectable by your firewall or edge router? Some of you reading this may know what I'm leading to. The question you're asking is, how? How could they avoid your firewall and edge routers and still port scan and OS scan my network? **Because they're on the inside.** This is where logging of Cisco switches may really come into it's own. Port scanning tools are not hard to find and are very easy to use. If a curious employee decides they want to have a look around your network, the firewall and routers are not even an issue.

So maybe I've convinced you that logging of your Cisco switches is an important layer in your security model. What does it take? It doesn't take special tools or equipment to go about this; in fact Cisco switches already have the capability to send logs in their IOS. It just needs to be configured. For the log server, there are different utilities out there, but in this discussion I'll be using Red Hat Linux 7.2. Red Hat Linux distributions are available free on the Internet so the only equipment you'll need to supply is a server or workstation.

The syslog server doesn't need to be the most powerful server you can find. Syslog messages are very small, don't use a lot of bandwidth, and only occur

when an event has occurred on or to the switch. Obviously the faster the processor the better, but a more important consideration is disk space. Depending upon how many switches you'll be logging, what parameters you'll be logging, and how often you will rotate, delete or save the logs to different media will all factor into how much disk space you'll need. Remember to go larger than you may need because in the event that someone attacks your network, the havoc they bring may fill up your logfile many times faster than just during average daily network operations.

When configuring your syslog server, it is very important to configure Network Time Protocol (NTP). I can't stress enough how important NTP is to keep your switches, log servers, and log files synchronized. Imagine for a moment that you're alerted that you've had an intruder on your network. They've done some port scanning and attempted to login into your devices. You then begin to review your logs and discover that each one of your devices is running with their time not synchronized. Cisco switches set their system clock time in the past upon startup and unless they are configured with an NTP server, will use this timestamp until they are restarted again. If all your switches are out of synch by seconds, much less a different day, month or year, you've just increased your work exponentially. Now you must first synchronize each of the switch logs before you begin your investigative work.

Next, you would have to disable any unnecessary services on your log server. Experienced hackers know that a network with good security practices will have logging set up for servers and network devices. If an attacker has entered a network and would like to return at a later date, they will attempt to cover their tracks by either erasing your log files or creating large amounts of log entries to overwrite your logs. While the latter point is not controllable, securing your log server is. You must deactivate any unnecessary services. Use a port scanning tool such as Nmap against your log server during setup will help make sure you don't have any ports open that absolutely do not need to be open.

Another very important point: your log server should be your log server. It shouldn't be running for any other reason. If you decide that because your mail server has more disk space than you'll ever need, you're asking for trouble by using it as a log server. Vulnerabilities that may be present with the mail system you're running may allow an attacker to also have access to your logging server, and the same is true in reverse.

Finally, limit the clients to your logging server. This device is meant solely for logging network devices and there shouldn't be anyone who needs access to this device for any other reason. Other users who have access to the syslog server may install software or services that may open ports or vulnerabilities that you may be unaware of. In the event of an intrusion, you may have to spend additional time deciphering whether the user or the intruder caused the change.

I mentioned earlier that Cisco switches already have the built in utility to allow them to log to a syslog server. Later in the paper the command and instructions will be covered with how to do this. Make sure that you know what type of Cisco switch you are running. Cisco switches run with two types of operating systems. They either use a CatIOS, which is what is referred to as a "set" command OS, or a Cisco IOS, which is very similar to the look and feel of a Cisco router. You'll see what I mean in the next section.

Log Server Setup

Load your operating system onto your log server. For the purposes of this paper, Red Hat Linux 7.2 will be our operating system. Most of the commands, syntax, file names, and locations should be similar to other Linux/Unix distributions. Using a server with an OS already installed can be done, but it's not recommended. By loading the OS fresh, we can disable all the unnecessary services and packages that would just make our system more vulnerable.

Note: All commands and files are configured as root.

First, let's set NTP on the system. To direct your syslog server to a device for NTP, edit the /etc/ntp.conf file and change the following:

server 127.127.1.0

to

server (IP address of device that provides time to your devices)

Next, edit the /etc/services file. Port 514/udp is the port that the syslog server will receive the log messages from the Cisco switches. Ensure that the following is entered into the file:

syslog 514/udp

If this line does not exist, add it to the services file and save. On my system, 514/udp is listed in the "UNIX specific services" section of the /etc/services file and was already present in the services file.

Now let's edit the /etc/syslog.conf file. Add the following line to the top of the syslog.conf file:

local6.* /var/cisco.log

This tells the syslog daemon that all messages from systems with the local6.* logging facility enabled will log to this file. In this example, I'm using local6.* in the syslog.conf file because local7*. is already being used as the boot logging

facility on my Linux system. Note: there must be a spacing of five tabs between local6.* and /var/cisco.log in the syslog.conf file.

Now it's time to edit the syslog daemon itself. Go to the /etc/rc.d/init/syslog file and edit the following line:

daemon syslogd –m 0

to

daemon syslogd –r –m 0

So what does the –r option do? The syslog daemon by default does not accept messages from remote hosts. By adding the –r, we've allowed the Cisco switches to log to the server.

In order for these changes to take effect, the syslog daemon needs to be stopped and restarted. This goes for any change that is made to either the syslog.conf file or to the syslog daemon. To do this while still in the /etc/rc.d/init/syslog directory, type the following command:

./syslog stop
./syslog start

The ./ in front of the syslog start command allows you to start and stop the daemon without the need to have the correct command path. So now your syslog server is ready to go; now it's time for the switches.

Configuring Logging on Cisco Switches

As I mentioned earlier, Cisco switches have two different operating systems and related commands. CatOS is what is known as a "set" command operating system. Commands are entered while the user is in "enable" mode. Once the enter key is pressed after a command with proper syntax, the change is made and is immediately saved to non-volatile memory. Cisco switch models in the 4000 series, 5000 series, and 6000 series have the CatOS or "set" switch commands.

In contrast, when making configuration changes on a switch with a Cisco IOS operating system, the user needs to perform the following actions. First, enter global configuration mode; make the change (no "set" is needed, just the correct command); exit global configuration mode; and save the changes by writing them to non-volatile memory. Cisco Catalyst switch models such as the 2900 series, 2950 series, and 3500 series have the Cisco IOS operating system. Now, let's look at how logging is enabled for both types of switches.

Cisco IOS configuration commands:

For switches with the Cisco IOS operating system, such as the Cisco Catalyst 3548, the commands to enable logging are as follows:

Enter global configuration mode:
Switch# configure terminal

Direct switch to the IP address of the syslog server:
Switch(config)# logging 10.10.10.1 (10.10.10.1 is the IP address of the syslog server)

Set the switch to the desired severity level:
Switch(config)# logging trap debugging

Set the switch to the proper logging facility on the syslog server:
Switch(config)# logging facility local6

Set NTP on the switch:
Switch(config)# ntp server *IP address* (insert IP address of device that is providing time to the devices on your network)

Exit global configuration mode:
Switch(config)# exit

Save the configuration changes to non-volatile memory:
Switch# write memory

This switch is now ready to begin logging to the syslog server.

CatOS configuration commands:

For CatOS switches the configuration commands slightly different. Remember, once the enter key is pressed after the command the configuration file will be changed!

Direct the switch to the IP address of the syslog server:
Switch> (enable) set logging server 10.10.10.1 (10.10.10.1 is the IP address of the syslog server)

Set the switch to the proper logging facility on the syslog server:
Switch>(enable) set logging server facility local6

Set the switch to the desired severity level:
Switch>(enable) set logging server severity 7

Turn the logging on:
Switch>(enable) set logging server enable

Set the switch's NTP:
Switch>(enable)

This CatOS switch is now able to log to the syslog server.

Severity Levels

So what are the various severity levels? Cisco switches can log messages from level 0 to 7, with 0 being the most important messages and 7 being least. The table shown below from Cisco systems gives a brief description of each level. [1]

Table 1-1: Message Logging Keywords and Levels

Level	Keyword	Description	Syslog Definition
0	**emergencies**	System is unusable.	LOG_EMERG
1	**alerts**	Immediate action is needed.	LOG_ALERT
2	**critical**	Critical conditions exist.	LOG_CRIT
3	**errors**	Error conditions exist.	LOG_ERR
4	**warnings**	Warning conditions exist.	LOG_WARNING
5	**Notification**	Normal, but significant, conditions exist.	LOG_NOTICE
6	**informational**	Informational messages.	LOG_INFO
7	**Debugging**	Debugging messages.	LOG_DEBUG

After developing some operating experience with logging, you may decide that logging the switches at level 7 may give you too much information. Every severity level up to and including the level you set on the Cisco switch will be logged into the log fie (in this case, levels 0-7). Obviously, the amount of syslog entries will depend upon how many devices you have logging to your syslog server.

If you have a large network with many switches that are logging to a single syslog server you may want to consider sending each severity level to a separate file. This will make finding the data much easier. To log to separate files, add the following lines to your syslog.conf file:[2]

local6.emerg	/var/cisco.emerg
local6.alert	/var/cisco.alert
local6.crit	/var/cisco.crit
local6.err	/var/cisco.err
local6.warning	/var/cisco.warning
local6.notice	/var/cisco.notice
local6.info	/var/cisco.info

```
local6.debug                    /var/cisco.debug
local6.*                        /var/cisco.log
```

The above configuration change is only needed on the syslog server. By configuring the Cisco switches with the "logging debugging" for Cisco IOS switches and "set logging server severity 7" for CatOS switches, each level message will go to its associated log file. All messages will still log cumulatively to the local6.* file.

Log Messages

So our syslog server is logging away and everything is working smoothly. So what type of information are we looking for in the logs? For starters, let's say you suspect a user inside your network is using a port scanner to do a little snooping around. They want to see what devices you have on your network and what ports those devices may have open. Unless you catch the user in the act of port scanning, you may have no idea what is going on if the user is smart enough to avoid scanning your firewall or IDS.

Here's what a port scan on a Cisco Catalyst 3500 series switch looks like when it is logged to your syslog server:

Jan 8 16:25:44 10.10.10.25: 44: 6w3d: %RCMD-4-RSHPORTATTEMPT: Attempted to connect to RSHELL from 10.10.10.164

This was just a simple Nmap portscan with no options to a Cisco switch. For this example, I used Nmap, but you would get the same log entry with a tool like ScanPort. The switch's IP is 10.10.10.25 and the device doing the scanning was 10.10.10.164.

Next is a log entry of an attacker who attempts to telnet to a switch without knowing the correct password. This login was attempted on a Cisco Catalyst 6000 series switch. Catalyst 6000 series switches run CatOS, which will give a user three chances to log in and then will log this error.

2002 Nov 27 08:50:03 CST –06:00 %MGMT-5-LOGIN_FAIL:User failed to log in from 10.10.10.164 via Telnet

The following entry is from a Catalyst 3500 series switch. It shows that the intruder is using a duplicate IP address that's the same as this switch. As you can see it also gives the intruder's MAC address.

00:12:06: %IP-4-DUPADDR: Duplicate address 10.10.10.25 on VLAN 1, sourced by 0005.4376.d0077

Managing Log Files

Once logging has begun, you'll need an easy way to get the information you need. There are many ways to manage your log files with Linux search commands, Perl scripts and cron jobs.

Grep is probably one of the most popular search commands for Linux. Grep can search on words, numbers, or patterns of just about anything you may want to find. Grep's manual pages offer all of the information you will need to use this powerful command. I have noticed that once log files become larger than 65536 bytes, grep had difficulty finding what I was looking for.

For those of you who dabble in programming, Perl scripts can be very useful for managing and searching log files. The following link describes a Perl script that is used to check the Cisco log file, copy entries to a history file, clears the log file, sends an e-mail to the designated recipient and exits the script. http://my.execpc.com/~keithp/bdlogcis.htm#loghost. This script can be very effective if you expect an attack on your network and need to be alerted the minute it happens. This specific Perl script is accompanied by a cron job that will run the script on a regular basis to check your logs for any activity.

Linux also contains a utility that rotates your logs so that they do not consume excessive disk space. By configuring the /etc/logrotate.conf file, you can manage each file individually and specify how often each file is rotated. Details on how to configure log rotation can be found on the following website: http://www.siliconvalleyccie.com/logging.htm.

Summary

In summary, logging Cisco switches can provide an extra layer of security to your network. Remember, not all attacks to your network come from the Internet. Logging your Cisco switches can provide a very simple yet powerful way to detect malicious activity inside your network. By keeping tabs on your logs, you can detect attacks that occur inside your network that may not be identified by your IDS, firewall, or servers.

As I've shown here, setting up logging doesn't need to be expensive or difficult since Red Hat Linux and Cisco switches already have system logging built in. All it takes is a workstation or server and some configuration. Once you have your syslog server up and your Cisco switches sending messages, managing those logs can be as simple as a couple of commands. This is a simple way to add another layer to your Defense in Depth.

References:

(1) Cisco Systems. "Using Debug Commands". Cisco IOS Releases 11.2. URL: http://www.cisco.com/en/US/products/sw/iosswrel/ps1824/products_command_reference_chapter09186a0080087d97.html.

(2) Vennard, Larry. "Logging to a Unix Syslog Server". Faq557-899. July 9, 2001. URL: http://www.tek-tips.com/gfaqs.cfm/lev2/8/lev3/58/pid/557/fid/899.

(3) Bitterfield, Colin. "Configuring Syslog for Datacenter Use" Revision 1.0. August 3, 2001. URL: http://colin.bitterfield.com/Syslog_for_the_datacenter.html

(4) Harrison, Peter. "Configuring Syslog". Linux Home Networking Tips. August 25, 2002. URL: http://www.siliconvalleyccie.com/logging.htm#_Toc32543569.

(5) Jordan, Joseph. "Error Logging with IOS". IOS Command Central. November 1999. URL: http://www.tcpmag.com/archives/article.asp?EditorialsID=17.

(6) Networking Unlimited. "Automated Analysis of Cisco Log Files". White Papers from the files of Networking Unlimited, Inc. URL: http://www.networkingunlimited.com/white007.html.

(7) Ranjbar, Amir. "Applying Cisco Troubleshooting Tools". November 16, 2001. URL: http://www.informit.com/isapi/product_id~%7BEC37821C-41A7-4A27-AB00-24F739A97754%7D/element_id~%7BCEA5568F-85A5-4AF5-A442-0F20EC2AA2AF%7D/st~%7B82A60075-2BDA-4E61-B46C-75C270CC4DE5%7D/content/articlex.asp .

(8) Cisco Systems. "Cisco Troublshooting Commands". January 17, 2003. URL: http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgcr/fun_r/frprt3/frtroubl.htm#1017942 .

(9) Parkansky, Keith. "Automate the Monitoring of Cisco Devices". 2001. URL: http://my.execpc.com/~keithp/bdlogcis.htm#loghost.

(10) McCarty, Bill. Learning Red Hat Linux. Sebastopol: O'Reilly and Associates, 2002.

Last Updated: May 25th, 2017

Upcoming Training

SANS Stockholm 2017	Stockholm, Sweden	May 29, 2017 - Jun 03, 2017	Live Event
SANS San Francisco Summer 2017	San Francisco, CA	Jun 05, 2017 - Jun 10, 2017	Live Event
Security Operations Center Summit & Training	Washington, DC	Jun 05, 2017 - Jun 12, 2017	Live Event
SANS Houston 2017	Houston, TX	Jun 05, 2017 - Jun 10, 2017	Live Event
Community SANS Ottawa SEC401	Ottawa, ON	Jun 05, 2017 - Jun 10, 2017	Community SANS
SANS Charlotte 2017	Charlotte, NC	Jun 12, 2017 - Jun 17, 2017	Live Event
SANS Rocky Mountain 2017 - SEC401: Security Essentials Bootcamp Style	Denver, CO	Jun 12, 2017 - Jun 17, 2017	vLive
Community SANS Portland SEC401	Portland, OR	Jun 12, 2017 - Jun 17, 2017	Community SANS
SANS Secure Europe 2017	Amsterdam, Netherlands	Jun 12, 2017 - Jun 20, 2017	Live Event
SANS Rocky Mountain 2017	Denver, CO	Jun 12, 2017 - Jun 17, 2017	Live Event
SANS Minneapolis 2017	Minneapolis, MN	Jun 19, 2017 - Jun 24, 2017	Live Event
SANS Columbia, MD 2017	Columbia, MD	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS Cyber Defence Canberra 2017	Canberra, Australia	Jun 26, 2017 - Jul 08, 2017	Live Event
SANS Paris 2017	Paris, France	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS London July 2017	London, United Kingdom	Jul 03, 2017 - Jul 08, 2017	Live Event
Cyber Defence Japan 2017	Tokyo, Japan	Jul 05, 2017 - Jul 15, 2017	Live Event
SANS Cyber Defence Singapore 2017	Singapore, Singapore	Jul 10, 2017 - Jul 15, 2017	Live Event
Community SANS Minneapolis SEC401	Minneapolis, MN	Jul 10, 2017 - Jul 15, 2017	Community SANS
SANS Los Angeles - Long Beach 2017	Long Beach, CA	Jul 10, 2017 - Jul 15, 2017	Live Event
SANS Munich Summer 2017	Munich, Germany	Jul 10, 2017 - Jul 15, 2017	Live Event
Community SANS Phoenix SEC401	Phoenix, AZ	Jul 10, 2017 - Jul 15, 2017	Community SANS
Mentor Session - SEC401	Ventura, CA	Jul 12, 2017 - Sep 13, 2017	Mentor
Mentor Session - SEC401	Macon, GA	Jul 12, 2017 - Aug 23, 2017	Mentor
Community SANS Colorado Springs SEC401	Colorado Springs, CO	Jul 17, 2017 - Jul 22, 2017	Community SANS
Community SANS Atlanta SEC401	Atlanta, GA	Jul 17, 2017 - Jul 22, 2017	Community SANS
SANSFIRE 2017	Washington, DC	Jul 22, 2017 - Jul 29, 2017	Live Event
SANSFIRE 2017 - SEC401: Security Essentials Bootcamp Style	Washington, DC	Jul 24, 2017 - Jul 29, 2017	vLive
Community SANS Charleston SEC401	Charleston, SC	Jul 24, 2017 - Jul 29, 2017	Community SANS
Community SANS Fort Lauderdale SEC401	Fort Lauderdale, FL	Jul 31, 2017 - Aug 05, 2017	Community SANS
SANS San Antonio 2017	San Antonio, TX	Aug 06, 2017 - Aug 11, 2017	Live Event
SANS Prague 2017	Prague, Czech Republic	Aug 07, 2017 - Aug 12, 2017	Live Event