Snort Management System:
Managing Multiple Snort Instances on Many Systems

Andy Bair

October 11, 2006
Version 1.0

Abstract

The snort management system enables engineers to efficiently manage and deploy small and/or large production snort environments. This system employs the open-source WebJob framework and several other open-source technologies including: oinkmaster, rsync, snort, and ssh. The system is designed to minimize the workload involved in managing the snort rules and related snort configuration files, while maintaining a high degree of security and robustness. A secondary goal of this system is to advance the open-source methodology for managing a large number of snort instances in an enterprise environment. This article describes how the system functions, and it further discusses the advantages and disadvantages of the solution.

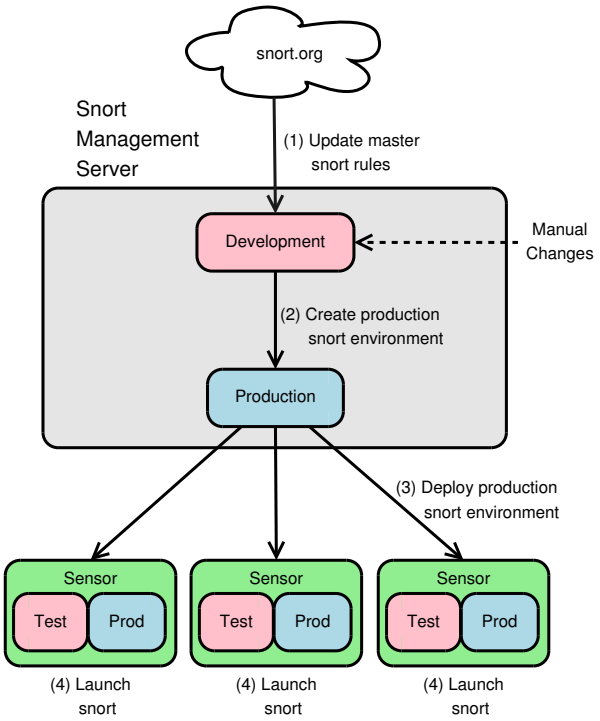

Figure 1: Snort Management System Architecture

Description

The snort management system uses four stages to manage snort environments. Two stages are performed on the snort management server and two stages are performed on the snort sensors. Please refer to Figure 1 when reading the detailed explanation of each stage below.

1. In stage 1, the snort rules are downloaded from `snort.org` using the Perl script `oinkmaster.pl`[1] and placed in a development location on the snort management system. This development location is where snort engineers will spend most of their time modifying snort configuration information.

2. In stage 2, the snort configuration environment for each snort instance is created and placed in a production location on the server. The snort environment includes all of the snort configuration and rules files. The rules files are generated using a combination of the master rule sets and oinkmaster configuration files. This production location is the handoff between stages 2 and 3.

3. In stage 3, snort sensors deploy the production snort environment from

the snort management server. First, the new environment is placed in a test location on the sensors, then it is tested using snort in test mode ('-T' switch). If the tests are successful, the new environments are moved into a production location on the sensors. This production location is the handoff between stages 3 and 4.

4. Lastly, in stage 4, when there is a change to the snort environment and the environment tests are successful, snort is restarted with the new production environment. All snort instances are also restarted if there is a mis-match between the number of defined snort instances and the number of running snort instances.

Advantages

Centralized – All snort configuration information is managed on the snort management server making it easier for engineers to modify configurations or add additional sensors.

Efficient – This solution employs a large amount of automation to eliminate human error and minimize the engineer time involved in managing snort configuration information. Engineers can spend minutes modifying snort configuration files on the snort management server, then sit back while the changes are uniformly deployed over the whole system.

Extensible – This solution facilitates running multiple snort instances on many network interfaces and on many sensors. It is trivial to add a new sensor to manage or manage a second snort instance on a single network interface because each snort instance has a separate configuration space, known as a snort environment. These configuration environments are independent from one another.

Fast – The snort management system is quick to implement, requiring approximately 4 hours to setup the server and approximately 10 minutes to setup a single sensor. So, it would take about 8 hours to setup the server and manage 24 sensors. After initial setup, an engineer would require approximately one hour per day to manage the snort rules on 24 sensors. Once an engineer modifies the snort configuration files, the snort management system will automatically deploy those changes to the appropriate sensors and snort instances.

Responsive – The snort management system informs a list of users via email when a change occurs in the configuration environment or when there are errors in the system. The error reporting is particularly useful to diagnose configuration problems.

Robust – One of the highest goals of the snort management system is to keep snort running. If a new snort environment is deployed to a sensor, it

is always tested before snort is restarted. In other words, snort is only started or restarted using a valid configuration environment. Also, if a snort daemon dies for some reason, all snort daemons are restarted even if there are no changes in the snort configuration environment.

Scalable – Once in place, this solution scales up to manage many snort sensors. It is possible to manage 100 or more snort instances on an hourly basis from a single snort management server, provided the server is a relatively modern 1U server.

Secure – The snort management system components are protected on the server, on the sensors, and in transit to the sensors. On the server, all snort management system configuration files are protected via restrictive permissions and only users within the snort group will have the ability to modify sensor configuration files. These restrictive file permissions are maintained when the files are deployed to the sensors. Also, since the configuration files are periodically deployed from a central server, rogue modifications to the configuration on the sensors is minimized. Finally, the snort management system components are protected while in transit to the sensors via a secure shell tunnel.

Standardized – Since snort configuration files are centrally managed and deployed to sensors, this standardizes the deployed snort environments. This helps to reduce error and minimizes system drift caused by one-off tweaking.

Variable Cost – Once this system is in place, the engineer time cost to to add additional snort instances and modify snort configurations is very small. For example, it can take an engineer less than 10 minutes of labor to configure the server to managing an additional snort instance and only a minute of labor to disable a snort rule across all sensors. The majority of the process has been automated in order to minimize variable cost involved in managing snort configurations.

Disadvantages

Fixed Cost – This solution requires more up-front work in order to create the snort management infrastructure than just manually deploying a snort configuration. While it may seem like less work to just manually push out snort configuration, it is often fraught with human error and maintenance can be cumbersome. Additionally, once this solution is in place, it is relatively trivial to add additional snort instances and modify snort configurations.

Mistake Magnification – Since the system centrally manages all snort configuration is it possible that a configuration error will be deployed to all snort

instances and it may go unnoticed. For this reason, only experienced snort users should manage the snort configuration environments.

Cost Comparison

Figures 2 and 3 show a cost comparison analysis between manually updating snort environments and updating snort environments using the snort management system. The cost comparison is the work factor, human time, to delete a single snort rule, generate a new snort environment, deploy those environments, test those environments, and finally restart snort with the new environments. The work factor for each solution was computed then multiplied by four to account for error. So, for the automated solution it took about 15 seconds to modify the oinkmaster configuration file, but a work factor of 1 minute is used in the graphs. The cost comparison assumes snort is installed and functioning properly on clients, and snort rule sets have been downloaded and are available.

The following formulas were used to compute the manual and fixed costs where the variables below are defined as follows.

cc	change count	=variable
fca	fixed cost automatic	=24 hours
fcm	fixed cost manual	=0 hours
ic	snort instance count	=variable
vca	variable cost manual	=0.01 hours per change
vcm	variable cost manual	=0.16 hours per change

$$\begin{aligned} manual\ cost\ in\ hours &= fcm + (cc * ic * vcm) \\ automatic\ cost\ in\ hours &= fca + (cc * ic * vca) \end{aligned}$$

Figure 2 shows the cost comparison between manually updating snort environments and using the snort management system. Three variables are tracked: total labor cost, snort instance count, and the number of deployed snort environments that contain changes. On the far left of the graph you can see the the initial labor cost for manually updating snort environment is assumed to be zero and the initial labor cost snort management is assumed to be 24 hours. 24 hours is a conservative estimate that includes installing a WebJob server, configuring WebJob clients, and finally implementing the snort management system. The break-even point is where the two graphs intersect. So, the snort management system will start paying off after 5 snort environment changes on 30 snort instances. This intersection is located in the front right corner of the graph. On the back right corner, you can see the cost difference for 30 changes to 30 snort instances. The snort management system costs is about 35 hours while manually making these changes costs about 140 hours. That's about a 400 percent cost difference.

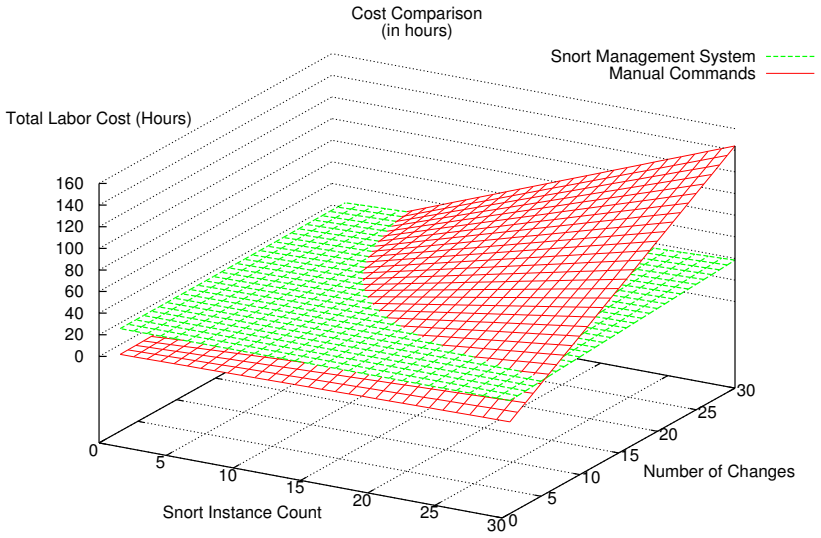

Figure 2: Cost Comparison

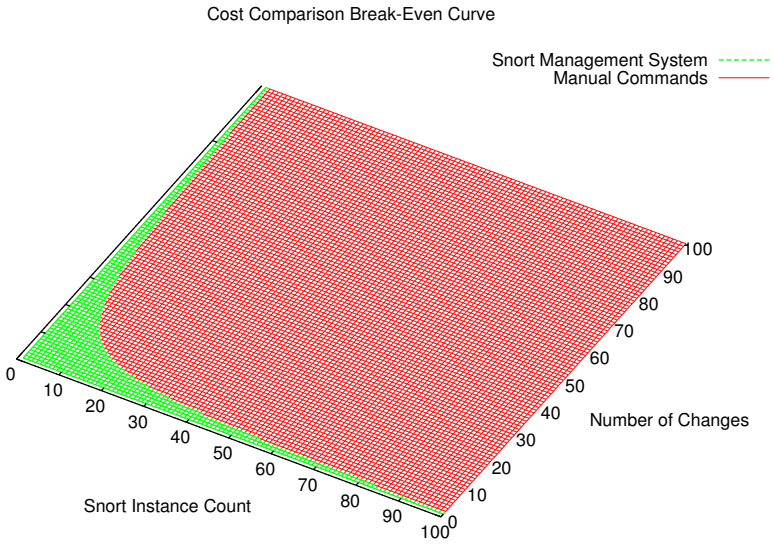

Figure 3: Break-even Cost Comparison

Figure 3 shows the break-even curve for the two solutions. The snort management system does not start paying off until you move from the blue area into the red. So, if there are a small number of snort instances that do not change very much, the snort management system may not pay off. On the other hand, the snort management system can pay off within a few changes if you have a large number of snort instances. Similarly, the snort management system can pay if you have a small number of snort instances but make many changes. The sweet-spot seems to be around 10. So, if you have 10 or more snort instances and you make 10 or more changes to the snort environments, the snort management system will pay for itself.

Conclusions

While the snort management system minimizes the amount of labor involved with changing, deploying, and testing snort environments, it does involve some up-front cost to setup and configure. Therefore, the snort management system is best suited for environments that are actively managing 10 or more snort instances.

References

[1] `oinkmaster.pl` can be downloaded from here: `http://www.snort.org/dl/contrib/rule_management/oinkmaster/`

[2] WebJob can be downloaded from here: `http://webjob.sourceforge.net/WebJob/index.shtml`

[3] More information on snort can be found here: `http://www.snort.org`