SANDIA REPORT
SAND2007-4475
Unclassified Unlimited Release
Printed August 2007

Network Configuration Management: Paving the Way to Network Agility

Joseph H. Maestas

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of Energy's
National Nuclear Security Administration under Contract DE-AC04-94AL85000.

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd.
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2007-4475
Unclassified Unlimited Release
Printed August 2007

Network Configuration Management: Paving the Way to Network Agility

Joseph H. Maestas
Advanced Network Systems and Technology Development
Sandia National Laboratories
P.O. Box 5800
Albuquerque, New Mexico 87185-0806

ABSTRACT

Sandia networks consist of nearly nine hundred routers and switches and nearly one million lines of command code, and each line ideally contributes to the capabilities of the network to convey information from one location to another. Sandia's Cyber Infrastructure Development and Deployment organizations recognize that it is therefore essential to standardize network configurations and enforce conformance to industry best business practices and documented internal configuration standards to provide a network that is agile, adaptable, and highly available. This is especially important in times of constrained budgets as members of the workforce are called upon to improve efficiency, effectiveness, and customer focus. Best business practices recommend using the standardized configurations in the enforcement process so that when root cause analysis results in recommended configuration changes, subsequent configuration auditing will improve compliance to the standard. Ultimately, this minimizes mean time to repair, maintains the network security posture, improves network availability, and enables efficient transition to new technologies. Network standardization brings improved network agility, which in turn enables enterprise agility, because the network touches all facets of corporate business. Improved network agility improves the business enterprise as a whole.

ACKNOWLEDGMENTS

The author wishes to express his gratitude to the following individuals without whose assistance network configuration auditing and policy enforcement would not be practiced at Sandia, despite its widespread use in industry to achieve a high degree of network accuracy and confidence.

Patrick Manke, 09338, Manager, Systems Analysis and Trouble Resolution. Pat took the initiative to assign staff to develop a plan for auditing configurations. I would also like to thank Pat for his commitment to move this project beyond the political and cultural issues, which for a long time impeded progress in the area of network configuration management.

Merle Benson, 09334, Retired Manager, Network Systems Design and Implementation. I would like to thank Merle for fighting many battles regarding the integration of network modeling and simulation with network design activities. He believed that quantitative analysis could augment anecdotal design practices and would give the organization a technical advantage.

Mike Sjulin, 02120, Senior Manager and former 09334 Manager. I would like to thank Mike for his vision in leveraging tools and information systems to help people analyze and track complex network systems. His vision inspired network modeling and simulation methods to take root in the network technology development organizations.

Leonard Stans, 09336, Manager, Advanced Network and Technology Development. I would like to thank Len for allowing me to continue to develop the methodologies for network configuration management and for demonstrating his commitment to this important task. I would also like to him for his vision of using modeling and simulation and data analysis techniques as a way to help solve complex networking issues.

Phillip Ayala, 09338, Phillip helps me perform the actual audits periodically. I would like to thank him for taking the time to learn the nuances of the configuration audit tools and for his desire to learn to write complex regular expressions. I believe soon he will be writing rules that move the state of configuration auditing technology beyond where it is today.

Ron Moody and Ed Klaus, 09338. These men drafted the plan for configuration auditing, derived the original configuration audit list, and are diligent in making the changes in the deployed configurations as required. They additionally provide excellent and timely feedback with regard to rule accuracy.

CONTENTS

FIGURES

TABLES

NOMENCLATURE

09330	Cyber Infrastructure Development and Deployment Department
09334	Network Systems Design and Implementation Organization
09335	Corporate Computing Infrastructure and Support Operations Organization
09336	Advanced Network Systems and Technology Development Organization
09338	Systems Analysis and Trouble Resolution Organization
API	Application Programming Interface
ARP	Address Resolution Protocol
CCMP	Configuration and Change Management Process
CEM	Cyber Enterprise Management
CMMI	Capability Maturity Model Integrated
CMP	Change Management Process
DRIfT	Dynamically Rendered Infrastructure Topology (Sandia Government Use Software)
FES	FastIron Edge Switch
HPC	High-Performance Computing
ICMC	Infrastructure Change Management Council
ISO	Internal Organization for Standardization
IT	information technology
ITIL	Information Technology Infrastructure Library
MAC	Move Add Change
MIB	Management Information Base
MTTR	Mean Time to Repair
NCST	Network Configuration Standardization Team
NDR	NetDoctor
NNM	Network Node Manager
OAM&P	Operation, Administration, Maintenance, and Provisioning
OUO	Official Use Only
Regex	Regular Expression
RR&R	Rapid Response and Recovery
SCN	Sandia Classified Network
SHN	Sandia Hotel Network
SILC	Software and Information Life Cycle
SIPRNET	Secret Network Protocol Router Network
SNL	Sandia National Laboratories
SNL/CA	Sandia National Laboratories/California Site
SNL/NM	Sandia National Laboratories/New Mexico Site
SNMP	Simple Network Management Protocol
SON	Sandia Open Network
SRN	Sandia Restricted Network
TFS	Template File Specification
TRP	Trouble Resolution Process
URL	Uniform Resource Locator
VNES	Virtual Network Environment Server
WFS	Web File Share

EXECUTIVE SUMMARY

Sandia networks consist of nearly nine hundred routers and switches and nearly one million lines of command code, and each line ideally contributes to the capabilities of the network to convey information from one location to another. Sandia's Cyber Infrastructure Development and Deployment organizations in Department 09330 have the responsibility to manage over $100 million in infrastructure assets as effectively as possible from multiple perspectives, including inventory of installed base, technology development and integration, planning and design, rapid response and recovery, and ensuring system reliability and integrity. These organizations have leveraged network management technologies when possible, have formalized many operations activities, and have engaged in the ISO 9000 quality system in order to streamline this colossal activity. The department realizes that commitment to its established body of standards provides an effective means to extend its operations and asset management system predictably and reliably. In the process of improving operations, operational inefficiencies are targeted; root causes are identified and often traced to the lack of sufficient or effective standardization in some areas. Ineffective network standardization is one of the areas identified that directly impacts network agility because it adversely affects all facets of network operations, administration, management, and provisioning.

Because it is therefore essential to standardize network configurations and enforce conformance to industry best business practices and documented internal configuration standards to provide a network that is agile, adaptable, and highly available, the 09330 organizations have taken strides to correct the problem. Section 2 describes some of the steps taken. Organization 09334 developed and documented standardized configurations for all three network environments at Sandia National Laboratories/New Mexico (SNL/NM). Organization 09336 developed a system of customized rules to enforce the documented configuration standards on all of the deployed configurations of routers and switches. Organization 09338 reviews the audit reports and makes changes as required to the deployed configurations and to the documented standards. Additionally, Organization 09338 provides feedback to the rule developer concerning accuracy.

Best business practices recommend using the standardized configurations in the enforcement process so that when root cause analysis results in recommended configuration changes, subsequent configuration auditing will improve compliance to the standard [1]. The configuration auditing process is slowly progressing across all environments, though the Sandia Restricted Network (SRN) is where this effort is primarily focused. Significant progress has been made this year in all three work environments to align network configurations with the documented standards. Approximately 50 percent (over 250 devices) of the SRN has been analyzed and corrected. Seventeen percent of devices have been analyzed in the Sandia Open

Network (SON) and 47 percent of the Sandia Classified Network (SCN) devices have been analyzed and corrected.

Recent configuration audits of SRN devices revealed that out-dated configurations, configurations that were previously corrected on all of the scanned devices, are showing up in the configurations of new devices. The bottom line is that the wrong configuration in a switch or router can wreck havoc on Sandia's network security, customer satisfaction and network agility. To prevent this from reoccurring in the immediate future, network staff must create and maintain configuration templates that comply with the documented standards. These templates must be audited against the standard to maintain them over time. They can then be used to configure new devices. A better long-term solution is to minimize the use of the command line interface and instead, use configuration management tools to deploy all configurations. The benefits gained by using these tools include: all configuration changes are audited prior to deployment meaning that the configuration standards are continuously enforced, configuration changes are tracked at a granular level and made available for fault isolation and service restoration.

As the auditing process continues, Organizations 09334 and 09338 will need to monitor and trend the outcome of these audits. A flat or increasing trend in the number of misconfigured items indicates there is a problem in the process of device deployment and configuration. Not fixing the problem here will cause us to "chase our tails," so to speak, and never achieve the desired level of accuracy. In the process of developing configuration audits, an omission in the current Change Management Process (CMP) was discovered. The process does not contain a provision that requires a given network configuration change to meet any level of accuracy with respect to the documented configuration standards. Correcting the CMP accordingly changes its functionality to include Configuration Management. A fitting name for the modified process might be Configuration and Change Management Process.

Ultimately, automated configuration auditing minimizes mean time to repair, maintains the network security posture, improves network availability, and enables efficient transition to new technologies. Network standardization brings improved network agility, which in turn enables enterprise agility, because the network touches all facets of corporate business. Improved network agility improves the business enterprise as a whole.

1. INTRODUCTION

Network agility is a somewhat ambiguous term because it means different things to different people. Agility can mean being in a position to move quickly and decisively to changing business conditions, and to adapt quickly in response to a growing need. To be in a position where agility is a natural outcome requires proactive planning and operational diligence across all areas at Sandia National Laboratories (SNL). Configuration management is more familiar to most information technology (IT) professionals. It is well defined and documented in several formal frameworks of best practice approaches and international standards intended to facilitate the delivery of quality IT products and services. Some of these frameworks and standards include the Information Technology Infrastructure Library (ITIL), International Organization for Standardization (ISO), Capability Maturity Model Integrated (CMMI) and Software and Information Life Cycle (SILC). Network agility and configuration management are intimately related. Through a well-designed process for configuration management, network agility is achieved. According to the ITIL, configuration management ensures that only authorized hardware and software are used in the IT infrastructure. Configuration management identifies the configuration items in the IT infrastructure as well as the logical and physical dependencies between the configuration items, and maintains pertinent records that sustain the end-to-end change and configuration management process. "Its goals include: 1) account for all IT assets and their configurations; 2) verify configuration records against the IT infrastructure and correct the exceptions; and 3) provide a sound basis for incident management, release management, and change management," [2].

Some of the lessons learned from the IT industry and from the IT technology areas at SNL show that by and large IT seems to grow in a generally uncontrolled fashion, despite having formalized processes designed to control the introduction of change into the IT infrastructure. Over time, the system becomes unnecessarily complex and costly to maintain.

Within the network infrastructure itself, i.e., routers and switches only, complexity is introduced during the normal turnover of equipment, operating system levels, and other variables within the operational environment (move add change (MAC) requirements, access control lists, network management systems, community strings, etc.). Each variation in the equipment model and in the operation system brings with it variations in the operation, administration, maintenance and provisioning (OAM&P) of the system. Without periodic correction, the natural variations that occur in the system cause configuration drift, which consequently impairs network agility.

A goal stated by the Infrastructure Change Management Council (ICMC) for the classified work environment is the idea of simplification. In this report and in the corresponding configuration management activities, we do not attempt a comprehensive treatment of configuration management as outlined by the ITIL. Such

a treatment would be time-consuming and expensive. Instead, we consider each of the ITIL goals and strategically apply them to the degree that is needed to accomplish our objectives to simplify network configurations, as described below. Nearly nine hundred network devices and over one million lines of command code comprise Sandia enterprise networks. For the network, simplification begins by identifying the critical elements within the set of variations listed above. Critical elements are limited to those that most contribute to an unstable system, those that contribute to a weakened security posture, and lastly, those that can degrade mean time to repair (MTTR). Additionally, because of the sheer size of Sandia's network infrastructure, we began this huge task by limiting the configuration analysis to the devices that were most critical to the commodity network. We began by first considering the core and distribution routers in the Sandia Restricted Network (SRN). As the process matured and our results improved, we refocused our attention to include other devices in all three security environments.

This report begins with a historical background when network configuration management via back-end configuration audits was introduced at Sandia. The political and cultural issues surrounding this technology severely impacted its effectiveness as a viable technology for the enterprise. The Network Configuration Standardization Team (NCST) was formed and slowly the documented configurations standards were developed. The documented standards opened the door wide and made network configuration auditing an easier concept to accept. Configuration auditing proved to be an effective method to enforce compliance to the standards as well as to identify errors in the documented standards. Additionally, by using out-of-the-box rules that assess network configurations against some of the well known best business practices (e.g., Cisco SAFE Blueprint for Enterprise Networks, NSA Router Security Configuration Guide, NSA Cisco IOS Switch Security Configuration Guide), Sandia's documented standards can be further enhanced as appropriate.

By trending the results of the audits, network administrators and managers can identify if there are problems within the process of deploying configuration changes. A preliminary assessment of Sandia networks showed the extent of configuration inconsistencies and traced the root cause to the Change Management Process (CMP).

To make the task of correcting the deployed configuration tractable, the NCST developed a plan for performing configuration audits and derived an initial scan list consisting of a command set that represents the highest risk to the network in the event they are misconfigured.

The report then moves on to the subject of rule development, beginning with an introduction to the NetDoctor paradigm. A high-level systematic procedure guides the analyst through the process of rule development, testing, configuration analysis, result validation, and finally report publishing. The software rules that are used to perform the audits are intentionally omitted from this document since they generally contain information that is official use only (OUO).

A brief description of the network data model gives the user perspective with regard to the activities behind the scenes that make back-end configuration management possible.

The report culminates with a brief discussion of several areas where improvements to this configuration management process can be made.

2. DEVELOPING AND ENFORCING CONFIGURATION STANDARDS

Periodic network configuration auditing had a rough beginning. Using tools to analyze network configurations were new at the time in the Networking organization. People viewed the technology as unnecessary; some even took it as a personal insult, as if somehow professional integrity to get the job done right was in question and challenged. (Professional integrity of the staff was never an issue.) With network availability metrics reaching 0.9998 levels (the goal is 0.998), "why is configuration auditing necessary?" was frequently asked and rightly so. The answer is very simple: Inconsistencies in the network affect many tasks associated with the OAM&P of a network. Tasks ranging from rapid response and recovery (RR&R), future technology migration, network security assurance, tool facilitated device configuration (a future goal) — all of the things that need to be optimized to achieve network agility, really — are impacted by inconsistencies in the configurations. Conversely, a consistent network configuration simplifies many of these tasks and brings network agility, within the realm of possibility. Configuration auditing is a way to scale the operations in a way that achieves a high degree of configuration accuracy. Industry leaders also incorporate configuration auditing for achieving network agility and high availability [1].

This section discusses periodic configuration auditing after device deployment. Aside from achieving network agility, configuration auditing is necessary because during the lifetime of a network device its configuration will inevitably change after its initial configuration. Device configurations must change, often daily, to reflect the MAC activity dictated by requirements from customers, program changes, and changes in ancillary network management systems. Seventy to ninety percent of the time, network staff manually implements these point-specific network changes via the command line interface of the device. (Tool-facilitated device configuration is reserved for replicating a given configuration change to a large number of devices of the same type.) Hence, many opportunities exist for accidental misconfigurations resulting in a natural, gradual digression from the documented standards unless tools are in place to catch the misconfigured items so that corrections can be made. This is what has been achieved for many of the deployed devices.

Figure 1 demonstrates the flow of intent for network configuration audits and analysis. The flow of intent begins by bringing industry best practices to bear on Sandia's operational issues and requirements. The combined outcome is then used to formulate a network configuration policy, which is then translated to a set of documented standards that are device-specific. After this, rules are developed to meet the requirements of the documented standard and they are subsequently used to audit the deployed configurations. Subsequent changes to the policy or to the configuration requirements results in respective changes to the rules. It is important that the flow of information remains intact and is readily accessible by all who need the information. The figure additionally demonstrates a flexible interval for performing the audits. Initially, the audits will be more frequent due to the high

incidence rate of non-compliance. However, as the configurations are corrected, the frequency for subsequent audits should decrease.

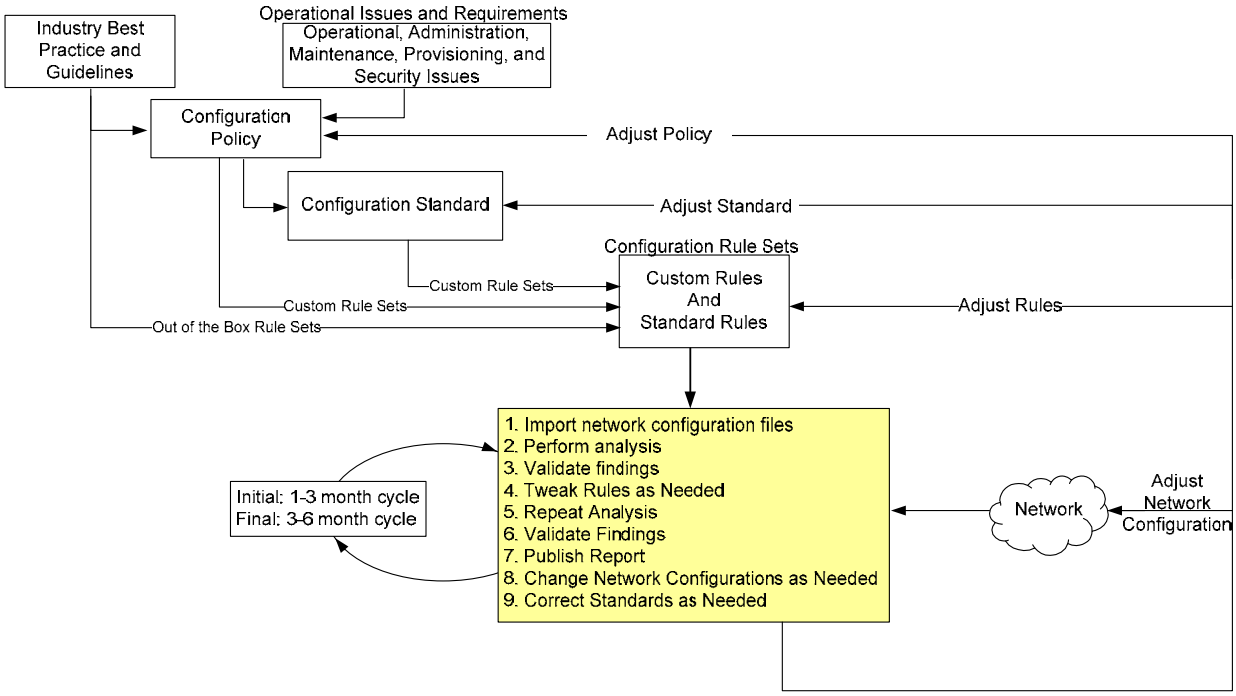


Figure 1: Network Configuration Analysis Flow of Intent

Organizations 09334 and 09338 will need to monitor and trend the outcome of the audits. A flat or increasing trend in the number of misconfigured items indicates there is a problem in the process of device deployment and configuration. Not fixing the problem here will cause us to "chase our tails," in effect, and never achieve the desired configuration accuracy.

2.1. Estimating the Extent of Configuration Inconsistency

To get an estimate as to how large a task it would be to simplify Sandia's network configurations, the first step was to identify a set of commands common to several different device types and to group them by device type. The next step was to use these results to write device-specific rules to detect the missing commands as well as the variations within a given command. After running the analysis, the results revealed that all devices analyzed — several hundred devices were analyzed in this initial assessment — had one or more statements that were either missing or had command parameters and keywords that were inconsistent within a given platform [3]. These results were to be expected since Sandia had never before documented nor enforced configuration standards. Even though these results were not surprising, it implored a deeper look into the formal ISO processes, the CMP and the Trouble

Resolution Process (TRP) used by the Networking organizations, for provisions that would help work out this issue.

2.2. The Root Cause of the Problem

At the heart of the Department 9330 ISO 9000 quality system are two processes that govern and manage change in the network: the CMP and the TRP. In the most fundamental terms, the output from these two processes usually involves a change to the network's configuration. For example, a network change is usually required to satisfy a specific customer request. A change in the network configuration is usually a part of the solution to a reported network problem. Albeit network change ultimately occurs as a result of either of these two processes, and within them are a means to ensure that a customer's request is satisfied, **there is currently nothing in either process that ensures that a given network configuration change meets any standard of accuracy with respect to configuration policies and configuration standards** [4, 5]. The configuration survey results discussed in the previous paragraph corroborate this finding. Since the CMP and TRP are core processes that govern the core business of the networking organizations, the lack of a provision for controlling configuration accuracy very likely contributes to inconsistent device configurations. It would be beneficial to enforce configuration standardization at the point of configuration deployment by amending the processes accordingly. Doing so expands the functionality of the CMP to include Configuration Management. The process name might be fittingly changed to Configuration and Change Management Process (CCMP).

Since the two processes, the CMP and the TRP, externally influence network configurations, it is clear that the documented standards must influence the CMP and TRP. Please consider Figure 2.

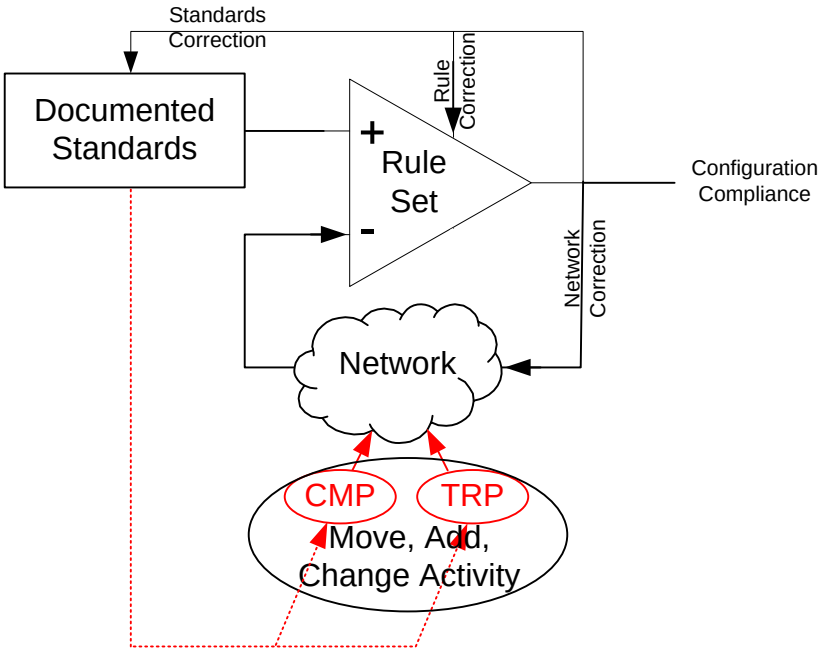


Figure 2: External Influences on Network Configuration Must Reflect the Documented Standards

2.3. Documented Configuration Standards and the Configuration Scan List

The Network Configuration Standardization Team, NCST, was formed after the initial configuration assessments. The NCST had several charters. One of them was to derive and document a configuration standard for each type of device deployed in the unclassified networks and to maintain the standards over time [6]. By establishing a standard, it becomes easier to detect the critical areas in the network that do not conform to the standard. Hence, the standards open the door for performing periodic network configuration audits to ensure that, indeed, the deployed configuration meets the intended configuration policy of the organization.

Another task of the NCST was to develop a "Configuration Monitoring" process. This process would lay the groundwork for deciding and defining the sets of commands that would be audited for each device type as well as deciding what would be audited, who would review the configuration audit reports, and who would implement the changes to the devices [7].

The effort to enforce standard network configurations requires continual collaboration between the participants, as Figure 2 also illustrates. Network correction, rule correction, and standards correction, though performed by members of different teams, must come together in chorus because each is an active component of this closed-loop system. In effect, we have taken a control systems

approach to solve the network standardization problem. In summary, the activities of each of the participants have been as follows: Organization 09334 develops and documents standardized configurations for all three network environments at SNL/NM. Organization 09336 develops the system of customized rules to enforce the documented configuration standards on all of the deployed configurations of switches and routers. Additionally, Organization 09336 collects and archives the configuration files of the network devices on a daily basis (an automated process). Organization 09338 performs the configuration audits, reviews the audit reports, and makes changes as required to the deployed configurations and to the documented standards. Additionally, Organization 09338 provides feedback to the rule developer concerning accuracy. A work agreement was fashioned between the managers of the NCST and the rule developer to solidify the commitment and the expectations of each participant. The work agreement additionally placed time restraints to ensure timely progress was made to this end. The interested reader may refer to the appendix for the work agreement.

A short list of high-priority commands was initially selected by the NCST. Items were later added to the initial scan list to accommodate operational needs of Organizations 09334 and 09338. These particular commands were selected because they have higher consequences to the network in the event they are misconfigured. A generalized scan list is necessarily shown in Table 1 to avoid the disclosure of sensitive network information. The four devices represented in the table comprise 50 percent of the SRN and 36 percent of the Sandia Open Network (SON). As the audit results improved over time and fewer configuration issues were identified with each subsequent audit, additional commands have been added to the scan list as well as additional devices. The NCST decides when commands are to be added, what commands are to be added, and where the rules are to be applied.

Table 1: Command List for Configuration Auditing

6500 IOS	6500 CatOS	3550 IOS	3560 IOS
banner motd	set banner motd	banner motd	banner motd
ntp server	set ntp server	ntp server	ntp server
	set ntp client		
service timestamps	set logging timestamp	service timestamps	service timestamps
mac-address aging	set cam agingtime	mac-address aging	mac-address aging
boot system flash	set boot system flash	OS version	OS version
	set boot config-register		
management access-lists 80, 81, 86	set ip permit	management access-lists 80, 81, 86	management access-lists 80, 81, 86
default route	default route	ip default gateway	ip default gateway

Snmp-server community	set snmp community	snmp-server community	snmp-server community
Snmp-server host	set snmp trap	snmp-server host	snmp-server host
logging	set logging server	logging	logging
tacacs-server host	set tacacs server	tacacs-server host	tacacs-server host
Name server	set ip dns server	ip name server	ip name server
	set timezone	clock timezone	clock timezone
	set summertime	clock summer-time	clock summer-time
	set vtp domain		
	set interface sc1 and sc0		

For the classified network, the approach to configuration auditing was very different. A relatively complete standardized configuration for the Foundry FastIron® Edge Switch (FES) served as the scan list.[1] This scan list is shown in Table 2. Once again, the list is generalized to avoid disclosing sensitive network information. The FES devices comprise approximately 72 percent of the classified network and 92 percent of the Foundry devices.

[1] FastIron is a registered trademark of Foundry Networks.

Table 2: Configuration Audit List for Foundry FESX devices

FESX	FESX Continued	FESX Continued	FESX Continued
aaa	slfow enable, destination	clock summer-time, timezone	default-gateway
enable password	ssh idle-time	sntp server	logging host
enable telnet	access-list 80, 81	no web-mngt	telnet access-group
hostname	fdp run	banner motd	
ip address	snmp-server community, host	username vty password	
dns	telnet timeout	tacacs-server host, key, retransmit, timeout	

2.4. Developing and Documenting the Rules

OPNET® NetDoctor (NDR) provides the framework for performing network configuration assessments.[2] It is a very flexible tool with the ability to customize it to meet unique organizational requirements and thus leverage Sandia's intellectual property. The purpose of this section provides the reader with a high-level view of the work that was performed to develop the custom organizational policies used in assessing Sandia's network configurations against the documented standards. However, a brief introduction to the NDR paradigm will help the user understand this section.

2.4.1 Introduction to the NetDoctor Paradigm

This overview is limited to the parts of NDR necessary to carry out customized configuration audits of device configuration files. For brevity, the overview necessarily omits the other aspects of NetDoctor that leverage the out-of-the box rules that give the ability to access the network against industry best practices such as the Cisco SAFE Blueprint from Enterprise Networks, or the NSA Router Security and many others or combinations thereof.

OPNET NetDoctor provides the framework for performing network configuration assessments. In general, NDR rules are written in the Python scripting language. A NetDoctor rule is a logical program that performs a set of operations against the network configuration variables. Values for the network configuration variables are stored in the attributes of device models resident in the modeling and simulation software core. Access to the attributes is provided through the NDR Application Programming Interface (API). Network attributes can be manually configured or they

[2] OPNET® is a registered trademark of OPNET Technologies, Inc.

can be imported. The OPNET Virtual Network Environment Server (VNES) or the eXpress Data Import modules provide the two primary methods to import the attribute values. Additionally, actual device configuration files can be optionally imported along with the attribute values. In order to do the assessments as discussed in this document, it is necessary to import the device configuration files.

A specific type of NDR rule exclusively uses Python regular expressions. Regular expressions are a powerful method for manipulating strings. The NDR "Organizational Policies" rule allows one to analyze device configurations using a system of user-supplied regular expressions (Regex). The Regex statements are written in one or more files called configuration templates. Each regular expression in NetDoctor is typically written to represent one and only one command line in the device configurations. However, depending on the need, complex regular expressions can also represent multiple command lines in a device configuration file or a family of command lines that meet (or do not meet) a certain format or that are grouped together. In all of these cases, the Regex itself cannot span multiple lines, which partly is responsible for some of the most hideous looking regular expressions. Consider the following expression for a Catalyst 5xxx device that ensures the system prompt matches the name of the device:

```
^set system name\s+(\w+\(.+\))[\s\S]+^set prompt\s+\1|^set prompt\s+(.+)>[\s\S]+^set system name\s+\2
```

Regular expressions can be difficult to read and write. However, realizing that every character in a multi-line string must be accounted for by either an explicit or an indirect reference in the Regex makes the job easier. When writing or interpreting regular expressions, there is no replacement for a few good references [8, 9]. There are also some very good GUI-based tools for testing regular expressions. One of these is Regex Buddy by Just Great Software.

Since the current NDR scheme does not support multiple sections within a configuration template, multiple files must be used to represent the configuration requirements for a given device type. Usually, one of the files represents the common configuration statements (expressed in regular expression vernacular) that must appear on all devices, and the additional files contain the exceptions or the commands that only exist in certain devices. A less efficient method to express the required configurations would duplicate the configuration template containing the common commands and supplement it with (or remove) the commands that should be in the exception list or excluded from it. The latter method is easier to set up initially, but more difficult to maintain in the long run. In the interest of longevity, the first method is implemented.

The configuration templates (the files containing the Regex statements) are either "match" templates or "noMatch" templates. The match templates will search the configuration files for the required configurations. When a match is not found, NetDoctor reports the "Statements not Found." On the other hand, the noMatch

templates will search the configuration files for statements that do not match the required configurations. When a noMatch statement is found, NetDoctor reports the "Statements Found." The example below will clarify this paradigm.

Example 1: This example demonstrates the match template
This is the device configuration file:
Statement 1 is required
Statement 2 is extraneous and should be eliminated

This is the match template file:
Statement 1 is required
Statement 3 is required

Without a noMatch template, the analysis of this configuration file would produce the following result:

"Statements not Found": Statement 3 is required

This means that the first statement was expected and found (since it is not in the report), but the third statement was expected and not found. Also, notice that the extraneous second statement in the configuration file was not detected. To correct this issue, consider the next example demonstrating the noMatch template.

Example 2: This example demonstrates the noMatch template
This is the configuration file:
Statement 1 is required
Statement 2 is extraneous and should be eliminated
Statement 3 is required

This is the same match template from Example 1:
Statement 1 is required
Statement 3 is required

This is the noMatch template:
Statement 2 is extraneous and should be eliminated

This time, the analysis of the configuration file will produce the correct result:

"Statements Found": Statement 2 is extraneous and should be eliminated

This means both statements 1 and 3 were expected and found. Notice that this time, the extraneous second statement was correctly detected.

In practice, it is actually a bit more complicated than these two examples lead one to believe because the noMatch template would be written so that it would detect any statement that matched: "Statement n is" (where n is a digit) and only report an issue if anything other than the keyword "required" occupied the fourth parameter.

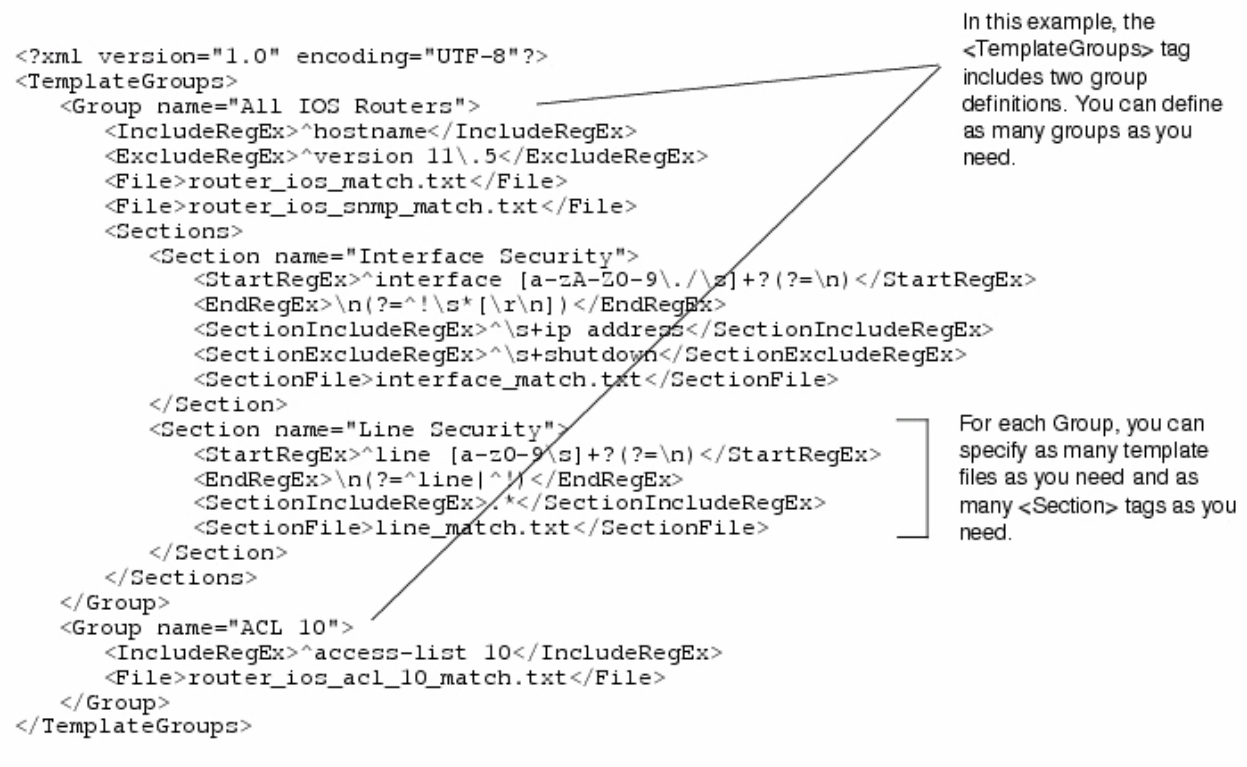
The final part of this overview deals with the method NetDoctor uses to match a configuration template with a particular device or group of devices. NetDoctor has a number of mechanisms either to identify devices that are of interest to the analyst or to exclude devices not of interest. Understanding how they work together will simplify the task of limiting configuration analysis to as small or large of group as desired. As discussed later, the first opportunity to group devices is at the VNES. The VNES is not a part of NetDoctor, but an integral component that builds and maintains a high-fidelity model of the operational network. A group created in the VNES is persistent; however, it is also static and requires updating, which is both good and bad; bad because it requires manual updating, good because it can be used to identify new devices added to the network between successive audits.

The second opportunity for filtering devices of interest for configuration monitoring is in the "Organizational Policy" rule suite in NetDoctor. Devices are selected for auditing by checking the box next to the rules of interest. The selected rules automatically enable filters that identify devices by vendor and by the particular operating system running on the device. The granularity of the filter at this level is coarse, but may in fact suffice if it uniquely identifies the devices of interest. If not, additional filtering is available.

The "Device Name" rule parameter (the rule parameters are displayed when a rule is selected as instructed above) provides the third opportunity for filtering. Usually, this parameter is not used in the filtering process because it does not persist for subsequent audits and because it is too specific. Its default value in the Regex venacular, ".*", which in this context essentially means "all device names," will suffice. It is better to identify devices with a general attribute, such as device type, than one that is dependent on an attribute that can change from one audit to the next.

If device specificity is not achieved with the above combination of filtering capabilities, the template file specification (TFS), "template_file_specification.xml," unequivocally provides unlimited flexibility. It is also the last opportunity for device filtering. Among other things, the TFS can be used to create any attribute that is useful to uniquely identify any particular group of devices. Such flexibility is afforded with the combination usage of the "IncludeRegEx," the "ExcludeRegEx," and any permutation of commands present in the device's configuration. The use of this file is optional. However, it is not optional when multiple configuration templates are used to describe the configuration requirements of a particular group of devices. In addition to grouping devices by the attributes in the IncludeRegEx and ExcludeRegEx tags, the TFS serves to direct NDR to a specific set of configuration templates for a specific device group. This is done with the "File" tags. It can be thought of as a band-pass filter where specific device groups are selected for analysis, while other devices are blocked from analysis. Figure 3 is an example of the TFS with its full complement of XML tags. TFS details are available in the online NetDoctor documentation. As you can see, it is fairly difficult to read. It allows for an enormous amount of flexibility and a rather complex system of configuration

templates that would be time-consuming to maintain. Instead, a more useful purpose for this figure is that it provides a hint as to how the TFS can be simplified and still meet SNL's auditing requirements. Aside from this figure, there is no documentation or guidance identifying the optional or mandatory XML tags.




```
<?xml version="1.0" encoding="UTF-8"?>
<TemplateGroups>
   <Group name="All IOS Routers">
      <IncludeRegEx>^hostname</IncludeRegEx>
      <ExcludeRegEx>^version 11\.5</ExcludeRegEx>
      <File>router_ios_match.txt</File>
      <File>router_ios_snmp_match.txt</File>
      <Sections>
         <Section name="Interface Security">
            <StartRegEx>^interface [a-zA-Z0-9\./\s]+?(?=\n)</StartRegEx>
            <EndRegEx>\n(?=^!\s*[\r\n])</EndRegEx>
            <SectionIncludeRegEx>^\s+ip address</SectionIncludeRegEx>
            <SectionExcludeRegEx>^\s+shutdown</SectionExcludeRegEx>
            <SectionFile>interface_match.txt</SectionFile>
         </Section>
         <Section name="Line Security">
            <StartRegEx>^line [a-z0-9\s]+?(?=\n)</StartRegEx>
            <EndRegEx>\n(?=^line|^!)</EndRegEx>
            <SectionIncludeRegEx>.*</SectionIncludeRegEx>
            <SectionFile>line_match.txt</SectionFile>
         </Section>
      </Sections>
   </Group>
   <Group name="ACL 10">
      <IncludeRegEx>^access-list 10</IncludeRegEx>
      <File>router_ios_acl_10_match.txt</File>
   </Group>
</TemplateGroups>
```

Figure 3: Sample Template Specification File.
Ref. Online OPNET NetDoctor Documentation

A closer inspection of the above figure indicates that the "Section" tags and everything in between are unneeded (one group has a section tag, the other does not). Removing these tags will help obtain a system of configuration templates that are much easier to manage and maintain.

This concludes the overview of the NetDoctor paradigm.

2.4.2 Step-by-Step Implementation of the Rules

The steps provided in this section assume the user has some experience navigating the NetDoctor tool. Detailed steps are available through the online NetDoctor and ITGuru or NetDoctor and Modeler documentation. The online documentation is accessed via the help menu item.

Step 1: Create an Appropriate Directory Structure
Since the documented standards differentiate between device types, so must the configuration templates. That is, each template set will apply to a specific type of device. Multiple template files within a given set contain the common rules that are applicable to all devices of this type as well as any exceptions that may apply to a subset of the devices of this type. Exceptions are maintained in separate and distinct template files to make it easy to identify all exceptions that are applicable to a given device or to a given group of devices. The directory structure differentiates between device types. It was selected to simplify the bookkeeping task for the initial development of the rules. Please refer to Figure 4.

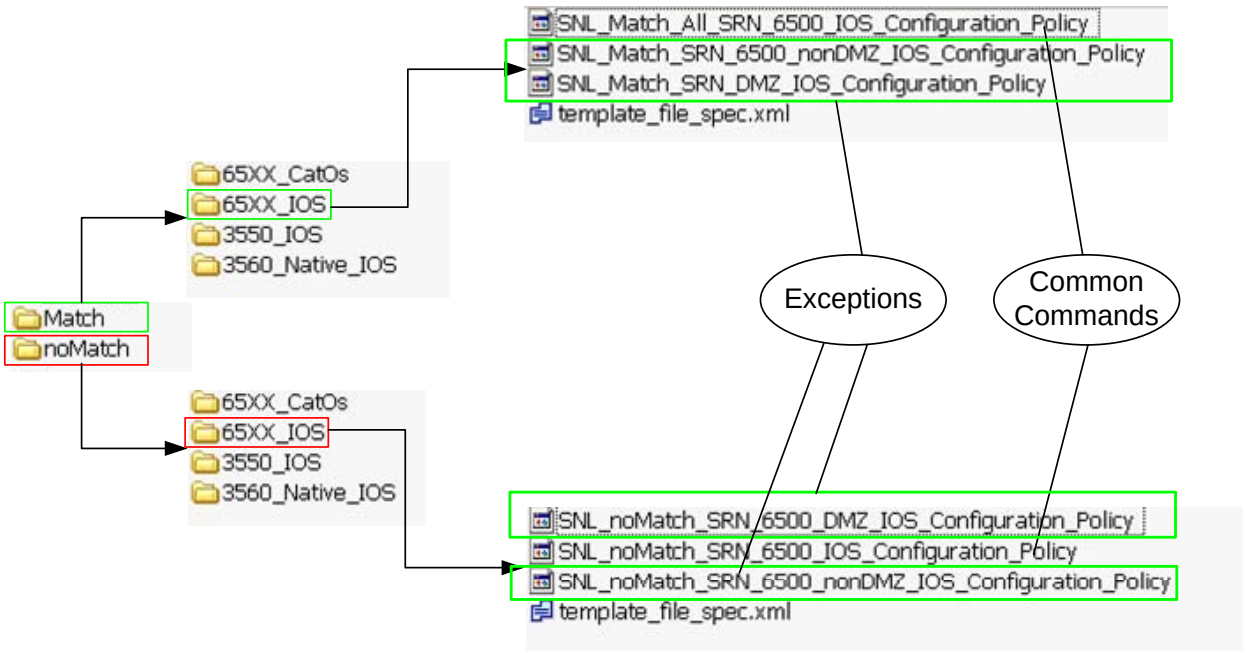


Figure 4: Directory and File Structure for Configuration Templates

However, such a hierarchical structure is unnecessary. Removing it will also simplify the steps the analyst must take to configure NetDoctor, which is a better long-term solution. However, to do so would require the consolidation of every TFS in the match and noMatch templates. At some point in the near future, the hierarchical structure will be removed (the subdirectories of Match and noMatch will be eliminated). This would result in one TFS for the Match templates and another for the noMatch templates.

Step 2: Select a Device Type and Create the Match and noMatch Configuration Templates
The match templates will contain the list of commands (in the Regex vernacular) that must appear in the configuration of the device. Separate the commands regarded as

exceptions. Collect the command list of exceptions in one or more configuration templates, each of which are specific to a device or a group of devices. Figure 4 illustrates this concept with the file names. Exceptions are usually identified during the process of implementing the configuration corrections into the device. Typically, but not always, the Match command list corresponds one-to-one with the scan list for the device. That is, for each command in the scan list there is a corresponding Regex command in the configuration template.

The noMatch templates are a little more difficult to conceptualize and implement. In general, the noMatch configuration templates will contain the command list (again in the Regex vernacular) that should not appear in the configuration of a device. As in the match case, the exceptions are collected into one or more configuration templates. The following Regex form is useful for the noMatch configuration template and for understanding how the noMatch template works.

```
^mac-address-table aging-time (?!14400\s*$|14400 routed-mac\s*$)
```

Note, it is the form that is important at this time and not the detail of every Regex meta character. This particular Regex will inspect every command in the configuration of the device looking explicitly for "mac-address-table aging-time". When a match is found, the Regex engine will ensure that only "14400" or "14400 routed-mac" appear as parameters. Any parameter that does not explicitly match either of these will be flagged as:

"Statements Found": `^mac-address-table aging-time (?!14400\s*$|14400 routed-mac\s*$)`

Obviously, this global command will only appear once in the configuration of a device, but other commands may have multiple instances, each with a slightly different parameter. Every acceptable variation in the command parameters must be listed in the Regex and separated with the "or" (i.e., |) meta character.

When creating the Match and noMatch Regex expressions, be sure to test each one thoroughly before moving on the next one.

Step 3: Testing the Regular Expressions

Import a small representative sample of device configurations from the VNES. To do so will require that you create a device group in VNES. At the time of this writing, only the VNES administrators can create this group. For this reason, group creation will not be discussed further. Ensure that the group consists of the device types of interest and a few other device types to test the case where the rules should not be applied and/or where variations in the command parameters necessarily exist, perhaps due to OS variations or other reasons.

Configure NetDoctor by expanding the "Organizational Policy" rule suite and selecting the specific types of devices you are interested in assessing. Recall that only course selections of devices are available in this particular filter. Here, the vendor and the operating system differentiate the devices. For example, "Cisco Native IOS Differs

from Template File" is one of several selectable filters. Be sure to configure the parameters at the bottom of the window. The "Device Name" parameter contains the ".*" Regex. Next, click into the "Match Template" values section and browse to the directory containing the Match configuration templates of interest. Repeat for the "NoMatch Templates" parameter, only this time browsing to the noMatch template directory. Also ensure that the "Administration > Device OS Version (Summary)" rule is selected. Refer to Figure 5.

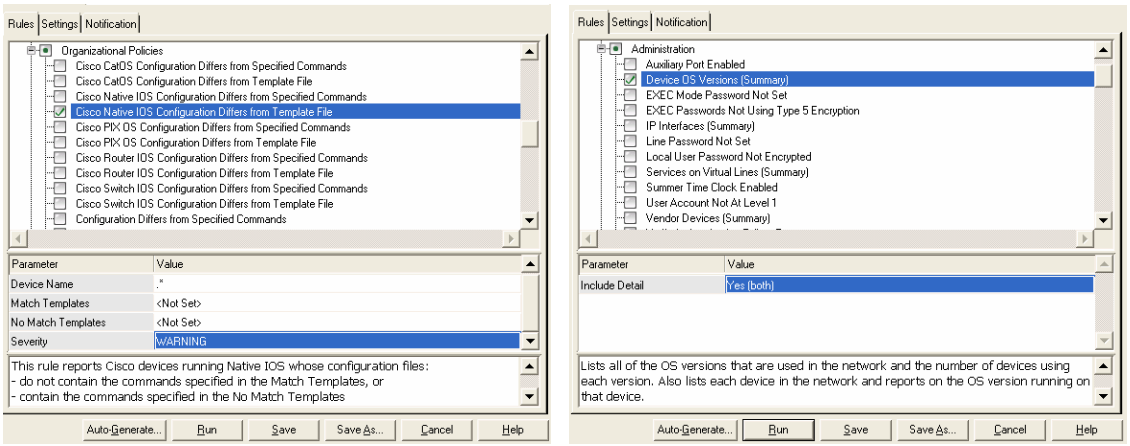


Figure 5: NetDoctor Configuration

Note that some devices — the Cisco 3560 is one of them — are a little tricky. These devices, and a few others, support layer-3 functionality as well as layer-2.[3] When these devices are imported in to the modeling and simulation domains, the modeling software identifies some of them as running "Native IOS," while others run "Switch IOS." Select both NetDoctor rules (filters) "Cisco Switch IOS Differs from Configuration Template File" and "Cisco Native IOS Configuration Differs from Template File" to correctly handle this situation. The Match and noMatch Template parameters for both filters must reference the same set of configuration templates. If not, the audit results will not be valid.

Similarly, when auditing other types of devices, be sure to match the rule with the "OS Type"selected by the modeling software. Identify all OS Types selected by the software for the devices of interest and select only those rules that match the OS Type. To view the OS Type, edit the device model attributes, then expand the attribute tree as follows: "System Information > OS Type." Use this value to select the appropriate rule (filter) when configuring NetDoctor.

Click on Save after NetDoctor has been configured. Next select the Settings tab. Refer to Figure 6. Enter a proper title for the report and select the Web Report format. Also ensure that the remaining check boxes are as shown in Figure 6. Click

[3] The terms layer-2 and layer-3 refer to the Operating Systems Interconnection Reference Model.

on Save again and then on Run. Within a few moments, a Web report will appear with the results of the analysis.

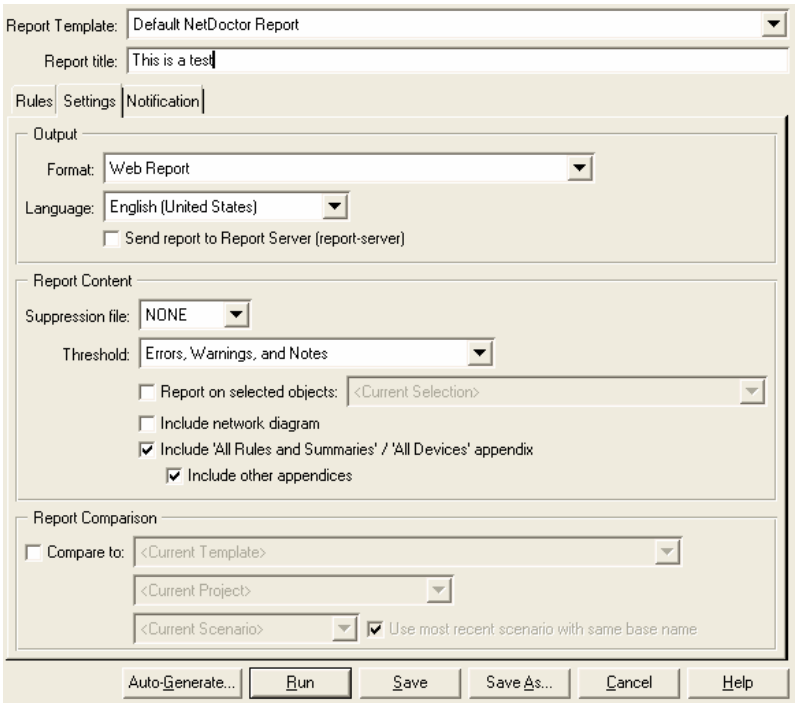


Figure 6: Selecting the Report Format

Be aware of the following idiosyncrasy in NetDoctor, version 12.0: the report will not indicate a NetDoctor malfunction when the configuration template files are not accessible by NetDoctor. The report will appear as if all went well and no errors were identified in the device configurations. To ensure that the configurations are set correctly and that NetDoctor is actually accessing the configuration templates, retrace each of the steps previously outlined. Then, modify one of the device configurations in the model and insert a few errors in one or more of the commands being audited. If the report still indicates zero findings, there is a problem with the configuration of the tool. Do not proceed to develop more rules until this problem is resolved. After the Match and noMatch configuration templates have been completed and tested, the next step is to configure the template file specification.

Step 4: Configure the Template File Specification

Figure 7 shows the TFS that was written for the SRN 6500 CatOS devices. First, note the value of the first "IncludeRegEx" tag. This Regex forms a group of devices that are of Model 65xx. The "ExcludeRegEx" tag excludes devices with names beginning with SACC3 or SACC9 (these names are indicative of SON devices). The "File" tag applies the named configuration template to this newly formed group of devices. In the 6500 CatOS case, note that one configuration template applies to all 6500 CatOS devices, and the second group applies to all non-6513 devices.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- This is the template specification file for the *match* templates. -->
<!-- It applies to 65xx devices. The first group applies only to all 65xx devices. The second group appl
<TemplateGroups>
    <Group name="All SRN 65XX CatOS Devices">
        <IncludeRegEx>show version[\s\S]+Model[\s]*:[\s]+[\S]+C65\d{2}[\s]+[\S\s]+$</IncludeRegEx> <!--
        <ExcludeRegEx>^SACC3|^SACC9</ExcludeRegEx> <!-- Don't check devices with names sacc3 or sacc9. -
        <File>SNL_Match_All_SRN_65XX_CatOS_Configuration_Policy</File> <!-- The 65xx CatOS switches must
    </Group>
    <Group name="All SRN non-6513 65XX CatOS Devices">
        <IncludeRegEx>show version[\s\S]+Model[\s]*:[\s]+[\S]+C65\d{2}[\s]+[\S\s]+$</IncludeRegEx> <!--
        <ExcludeRegEx>^SACC3.*$|^SACC9.*$|^Hardware Version:[\s]+\d\.\d[\s]+Model: WS-C6513[\s\S]+$</Exc
        <File>SNL_Match_SRN_65XX_not_6513_CatOS_Configuration_Policy</File> <!-- Configuration template
    </Group>
</TemplateGroups>
```

Figure 7: Match Template File Specification for the SRN 6500 CatOS Devices

Repeat the TFS configuration for the noMatch configuration templates. Very often, the two TFS files will be identical. As the user gains experience with the TFS, it becomes increasing easier to manipulate them to do what is desired. Copy the TFS files into their respective directories containing the appropriate configuration templates. Be sure to test them thoroughly.

Step 5: Run the Analysis and Validate the Results

Depending on how the TFS for the device of interest is written, either the entire network is imported into the modeling environment or a subset by device type must be imported. The TFSs for the various device types were written with both of these scenarios in mind. However, it is best to write the TFS so that regardless of the device types that are imported, the analysis will occur correctly on each type of device.

For example, consider the TFS in Figure 7. This TFS is correctly written because the "IncludeRegEx" tags will group devices that are of type 6500. However, in this case, if the imported network also contains other CatOS non-6500 devices, then when the "Cisco CatOS Configuration Differs from Template File" rule is selected and the Match and NoMatch template parameters are set to reference the TFS in Figure 7, only the 6500 CatOS devices will be analyzed; the other CatOS devices will be skipped. If the intent were to simultaneously analyze other CatOS devices, such as the 2948, the TFS would be modified first to create a third group for the 2948 devices and second to direct NetDoctor to the corresponding 2948 configuration templates. Ideally, this is how all TFS files should be written. When the TFS is written in this manner, the imported network can be either device-specific or can be a combination of many different types of devices. Memory allocation becomes a problem when the imported networks get too large. Sometimes, it is best to import only a specific type of device at a time. This keeps the audit report short and concise.

Now that the TFS is understood, the appropriate network can be imported and analyzed with NetDoctor. It is important to validate the results before publishing them. If the report contains errors (false positives or false negatives), correct the corresponding regular expressions and repeat the analysis. Publish the reports only after you are satisfied with their accuracy.

Step 6: Publish the Results
Reports are published in two ways. One method is integrated with NetDoctor. This method automatically sends the WEB report to the Report-Server. The problem with this method is that at the time of this writing, Report-Server Version 2 only holds the published report in its repository for 30 days, after which it is deleted. Audit reports need to be held for much longer periods to trend the results. Since the reports often contain sensitive OUO information, access to them is controlled via the Report-Server accounting system.

The second method to publish the report involves manual copying of the report files and pasting them into a WEB server directory. Even though this method takes longer, it is currently the preferred method because the reports persist until deleted manually. To copy the report files, browse to the directory indicated by the Uniform Resource Locator (URL) address in the browser. Copy all the files and paste them in the designated directory on the WEB server. Access to the reports is controlled through the corporate Kerberos system.

Step 7: Congratulations. You are on your way to attaining network agility.

2.5. What's Behind the Data Model

Network systems and network management solutions necessarily create islands of information that are difficult to manage and leverage holistically. Each disparate source is used to manage some aspect of the network. For example, one solution may provide a comprehensive view of the logical network, but only for a single vendor. Another solution provides topology information, but it lacks configuration and utilization detail. Yet another solution provides utilization data, but it lacks fault management. No one solution provides a comprehensive set of data that accurately describes the unified IT infrastructure. We recognized a need for this type of information and instrumented an inventory system for hardware, software, and operational data for heterogeneous networks for the three security environments at Sandia. The system is based on the OPNET Virtual Network Environment Server (VNES). It is designed to merge data from many disparate sources to create a unified view of the IT environment.

Via different types of adaptors (Simple Network Management Protocol (SNMP) Management Information Base (MIB), command line interface, standard database connections, etc.) the VNES collects physical and logical configuration details of

network devices from the devices themselves as well as links utilization metrics from the Statseeker tool. It additionally collects operational data like MAC address tables, route tables, address resolution protocol (ARP) tables, discovery protocol tables, forwarding tables and much more. We attempted to collect this information from other network management systems early on (HP Network Node Manager (NNM), CiscoWorks, Mapping network-object-server); however, we were either unable to obtain data reliably from them or the data that we needed was not provided by them. For example, NNM provided an excellent layer-3 logical view of the network, but its topology rendering capabilities were inadequate. The VNES determines integrated layer-2 and layer-3 topology using the configuration data collected from the network devices. Without question, the VNES provides us with the most reliable and comprehensive set of network information than any other solution currently in use at Sandia.

On a daily basis, network configurations are collected and archived. ARP and MAC address tables are collected more often. Currently, the data that is collected with the VNES is made available and used by the following configuration management tasks and activities:

1. Network configuration auditing;
2. Network configuration archiving;
3. ARP table collection in support of the corporate host scanning process (classified network at SNL/NM and SNL/CA);
4. Network modeling and simulation — "what if" analysis;
5. Sandia's Dynamically Rendered Infrastructure Topology (DRIfT) application: automated network diagramming and analysis, host mapping, business views, root event identification;
6. Network configuration summary reports; and
7. Visionael.

2.6. Configuration Management Deficiencies and Improvements

This section identifies some of the areas within SNL's existing network and configuration management system that are not working as effectively as need be. The author summarizes some of the methods that can be used to improve and correct these areas. Specific suggestions for improvement come either from industry best practices or from Sandia IT staff. In all cases, the ideas and suggestions from both sources are fundamentally the same.

2.6.1 Tracking Network Changes and Maintaining Standardized Configurations

First and foremost, Sandia does not have the means to track the changes that are made to all network devices at a sufficiently granular level that is needed to further improve MTTR. Sandia collects device configuration files periodically on a 24-hour basis using several different tools including CiscoWorks, IronView, the Virtual Network

Environment Server (VNES), and the Network Object Database, which is commonly knows as "Mapping." The first three of these are commercial tools, while Mapping is a homegrown solution.[4] The first two are proprietary in nature in the sense that they work only with network gear from a specific vendor; CiscoWorks is a network management tools for Cisco devices, while IronView is a network management tool for Foundry devices. The VNES is a more generalized network management tool capable of collecting configuration files from Cisco and Foundry, as well as from many other vendors. The homegrown Mapping tool, has historically collected from Cisco devices only. To collect from other devices using this tool requires a fluent Perl programmer. A description of the full breadth of capabilities for each of these tools is not within the scope of this document. However, the interested reader may learn more about them from the specific vendor web sites. In short, their capabilities vary widely with some common functionality.

Scheduled configuration collection, as is currently performed in all of the above solutions, does not produce the required granularity needed to track the changes to the network as they occur. Trap driven configuration collection would indeed improve the granularity of the data, however, it must be accompanied with integrated capabilities to search and identify the differences between the configuration file archives for a given device. Some of the commercial tools mentioned above support trap driven configuration collection (e.g. VNES) and some do not. Such tools must also have the ability to track command sets, which tracks a given set of changes across one or more devices as a set of commands that fulfills a common change requirement. For example, the configuration needed to stand up a new link between two devices would represent a single command set.[5]

While a trap driven configuration file collection scheme can be used to track the details of a given configuration change, which is irreplaceable for efficient fault isolation and resolution, it would not prevent the introduction of a "bad" change to the configuration of the network. For example, allowing auto-address summarization to persist when discontiguous networks are configured in a network will cause a routing failure.[6] These types of failure scenarios and others are preventable by using appropriate tools (with integrated policy enforcement) to deploy configuration changes instead of relying on the device-specific command line interface to manually deploy configuration change one device at a time. Tool assisted configuration deployment is an efficient means to deploy error-free configuration changes, to track configuration changes at a granular level, and to perform post-mortem failure analysis. The Cisco Configuration Assurance Solution (CCAS) by Cisco and Configuration Management solutions by Intelliden are examples of such tools for heterogeneous networks.

---

[4] The Mapping tool was developed by Jeffrey Diehl, ORION International Technologies, Inc.

[5] A change set is a common capability of many software versioning systems. A command set is a similar term in the configuration management arena. It's use as an effective capability for configuration management has been suggested by staff in Org. 9336.

[6] This example was derived from an actual service outage scenario on the Sandia Open Network.

2.6.2 Manual Service Configuration (i.e. using the command line interface)

The previous discussion attempted to highlight the importance of using configuration management tools to deploy and manage network configuration changes. This section deals with the case where the command line interface is predominantly used to deploy device configuration changes.

Currently, Sandia networks are predominantly configured using the command line interfaces supported by the network devices. This manual process is not only costly, it is also not risk adverse because it is prone to error and relies on the ability of the implementer to visually spot configuration errors in a configuration set that consists of thousands of lines of command code. Recent configuration audits of new devices revealed that some of the out-dated configurations (were previously corrected) were once again used on the new devices despite having documented configuration standards. Two important facts were revealed as a result: (1) Simply having documented standards is not a sufficient mechanism for ensuring that new device deployments will satisfy the standard; (2) network deployment staff are using uncorrected configurations to configure new and re-deployed devices. The bottom line is that the wrong configuration in a switch or router can wreck havoc on Sandia's network security, customer satisfaction and network agility. Indeed, the manual process is costly and introduces many risks that Sandia cannot afford to incur.

If manual service configuration is allowed to persist at Sandia, the recommended practice is that networking personnel create configuration device templates that are correct per the documented standard and that these templates are audited against the standard. (The auditing methods discussed in this report can be used to audit the templates.) These device templates can then be used to subsequently configure new devices using the popular copy and paste method.

2.6.3 Front-end Auditing versus Back-end Auditing or Both

Thus far, in this project configuration auditing has been restricted to the back end. That is, on a periodic basis network configurations are audited after they have been deployed. Back-end auditing, as this is called, is adequate provided the network implementation staff are diligent in applying the standards to the configurations before the deployment of a device. Recall that since neither the CMP nor the TRP processes require that configurations changes meet any level of accuracy with respect to the documented configuration standards, the chance that recently deployed devices will contain errors is high over the long term.

If configuration auditing were moved to the front end, so that changes would have to be validated before deployment, the chance that recently deployed configurations would contain errors would be very small. (Front-end auditing was introduced at Sandia several years ago as a way to validate device configurations prior to transitioning a new device to production status. However, political and cultural pressures severely impacted its effectiveness.) Many vendors in the configuration

management arena offer pre-deployment validation solutions. Examples include NetDoctor by OPNET, Cisco Configuration Assurance Solution by Cisco (currently uses OPNET's Netdoctor engine for rule-based policy enforcement), and Configuration Management Solutions by Intelliden.

Back-end auditing, however, does bring advantages that front-end auditing does not address. Via back-end auditing, Sandia's networks can be evaluated against best business practices such as the Cisco SAFE Blueprint for Enterprise Networks, the NSA Router Security Configuration Guide, the NSA Cisco IOS Switch Security Configuration Guide, and so on. These can be used to further enhance Sandia's network policies and configuration standards as appropriate. Hence, the total replacement of a back-end auditing solution for a front-end auditing solution is not recommended.

2.6.4 Reliance on the ISO 9000 Quality System to Control and Manage the Documented Network Configuration Standards

The documented configuration standards are published within the ISO 9000 Quality system in Organization 09334. During the course of developing these configuration management practices, we often discovered errors in the published standard and required timely correction of the standards. (If the networking staff are expected to use the standards, it would be prudent to maintain accuracy of the documents as often and timely as needed.) As is, the ISO 9000 Quality system prevented the timely correction of these documents. To overcome this deficiency, the documents have been duplicated and are separately maintained outside the confines of the ISO system. The upside is that now the standards are maintained as frequently as required. However, the shortcomings are many: (1) a change in the standard does not automatically notify the rule developers (no subscription services); (2) the updated standards are not readily available to all who have need for them; (3) the up-to-date standards are no longer the "official" documents used by the networking staff to control device configurations. For the benefit of the organization, the ISO 9000 steering committee should make the necessary changes that will enable it to be more responsive to the needs of the organization. After the necessary changes are made and the quality system is improved, the documented standards should be relocated under ISO.

2.6.5 Revealing Deficiencies in the Service Configuration System

Trending the results of the configuration audits will reveal whether or not the network deployment staff are using the configuration standards to configure and deploy new devices. Unfortunately, we do not have any means to automatically trend the results over time. Trending must be done manually by observing the number of findings over time during each pass of the configuration scan cycle. Software is commercially available to perform automated trending.

2.6.6 Enforcing Configuration Standards in other Network Environments

As stated earlier in this document, the NCST developed and documented configuration standards for Cisco devices on the SRN and SON. This effort was indeed valiant. However, it was narrowly focused because it only considered Cisco configurations for the unclassified environments. Due to changes in staff, the NCST was adversely affected. Should the NCST continue or cease to exist, its functionality should be encouraged to continue, with an even broader scope. The classified network, the High Performance Computing (HPC) environment, the Secret Internet Protocol Router Network (SIPRNET), the Sandia Hotel Network (SHN), the Sandia Wireless Networks, the intersite Virtual Private Networks, and all other Sandia network environments, including those at SNL/CA, would benefit immensely should each develop and document standardized configurations. Recalling that the network is the enabling medium through which enterprise agility is carried out and made possible, subsequent monitoring of these configurations would ensure that network agility is maintained over the long term. Not surprisingly, such an effort would serve an immense benefit to the 09334, 09336, and 09338 organizations and to their SNL/CA counterparts.

2.6.7 Software Version Control in the Access Layer Impacts the User Experience

Configuration management best practices suggest limiting the software version of a specific device type to a known version that is proven stable. Without version control, variations in the software versions adversely impact network stability because of potential interoperability issues and because of the potential introduction of defective versions in the network. Software version control makes the network less complex because it reduces the number of variations in the system.

Sandia maintains a maintenance contract with Cisco (Cisco SMARTnet) for the production Cisco gear. Among other things, the contract enables Sandia to replace the operating system shipped with a particular device type with a version that Sandia had proven to be stable. The SMARTnet contract enables Sandia to effectively apply software version control principles to maintain network stability and agility. To reduce the maintenance costs associated with the network, the scope of the contract was necessarily reduced to include primarily the large Cisco WS-C6500 devices comprising the core, the distribution layer, and a portion of the access layer in the SRN. About 60 percent of the SRN has been eliminated from the SMARTnet contract, contributing to the difficulty to standardize the image of the operating system on the affected devices, all of which are in the access layer; the layer which most affects the user's experience. Access layer switches impact the end customer more so than distribution or core devices because link redundancy is not usually carried out to the access layer. Now that version control is severed in the access layer, network instability will likely increase. Customers will be impacted when version interoperability issues cause unexpected behavior in the user interfaces. Unfortunately, Sandia network availability metrics will currently not measure the resulting instability because the metrics are currently collected from core and distribution devices. Availability metrics from the access layer should be collected to

track and determine the degree to which version variations affect network availability for the user. Additionally, with the variations in the versions of the software come variations in the command parameters and supported feature sets, leading to a gradual, patchy digression from the standard. This irregularity will also affect operational efficiencies.

3. CONCLUSIONS

Back-end configuration auditing has its advantages in assessing deployed network configurations against the documented standards as well as against the well-known and respected best business practices including: the Cisco SAFE Blueprint for Enterprise Networks, the NSA Router Security Configuration Guide, and the NSA Cisco IOS Switch Security Configuration Guide. Via back-end configuration auditing, we discovered that out-dated configurations (configuration that do not comply with the documented standards) are often used to configure and deploy new devices. **The bottom line is that the wrong configuration in a switch or router can wreck havoc on Sandia's network security, customer satisfaction and network agility.** Obviously, back-end configuration auditing it is not a sufficient mechanism in itself to continuously maintain standardized configurations in the network.

By minimizing the use of the command line interface to configure services one device at a time and by enforcing the use of the correct configurations each time a new device is deployed and each time changes are made in the network, we can be assured that our network will continuously comply with the standard. Some network configuration management tools have these capabilities and other capabilities that will enable Sandia to track and correlate configuration changes to service outages, hence improving MTTR. This is a direction that needs to be seriously considered for the long term. It is a more efficient way to operate, administer, manage, and provision the network.

Due to shrinking budgets at Sandia, the Cisco maintenance contract (Cisco SMARTnet) was necessarily reduced. This reduction will indeed affect network stability at the access layer. This is because the contract currently covers primarily the large 6500 devices commonly found in the core and distribution layers of the network and excludes the smaller devices located in the access layer. Software version control in the access layer limits the software versions that can be applied to the access layer switches. The expanded SMARTnet contract (the previous contract) enabled Sandia to replace the software of these devices with a version that it had proven to be stable. Because of the current contract limitations, Sandia is not at liberty to replace software and must retain the software that is shipped with a given device type. As a result, we expect customers will be impacted when version interoperability issues cause unexpected behavior in the user interfaces. However, because current network availability metrics only measure the core and distribution devices, availability measurements that most affect the customer are not collected. Moving forward, it is

recommended that availability metrics be expanded to include access layer switches so that we can track the degree to which customers are impacted.

The NCST did well to develop and document a set of standards for the Cisco equipment in the unclassified environments. This work should continue for the other network environments because development and enforcement of standards makes the network environment less complex. Some of the other environments include the core and distribution devices in the classified network, the High Performance Computing (HPC) environment, the Secret Internet Protocol Router Network (SIPRNET), the Sandia Hotel Network (SHN), the Sandia Wireless Networks, the intersite Virtual Private Networks, and all other Sandia network environments, including those at SNL/CA. The Foundry FES devices in the Sandia Classified Network (SCN) are close to meeting the documented standard developed for them. Network auditing would never have developed to the point it is today had it not been for the development of these standards. However, continued maintenance of the standard is important, as is the maintenance of the rules used to enforce the standard.

A better mechanism is needed to closely tie the documented standard to the rules that enforce that standard. As it exists today, a change in the standard does not automatically raise a flag to the rule developers. Web File Share (WFS), on the other hand, provides subscription services. However, if the standards are bound within the ISO document set, we discovered that they could not be effectively maintained at the required frequencies to satisfy the configuration audit cycle. Because of this, the standards have been duplicated in the Telecom directory of snl\collaborative. Because of the subscription services of WFS, it might be best to relocate the standards once again into WFS, but outside the confines of the ISO establishment, at least until the ISO system is improved.

The audit reports show that we have made significant progress in advancing our goal in all three environments. The Sandia Classified environment appears to be the furthest along. The SRN comes in second. The SON is only starting out. However, much work remains to be done in all three network environments. Sandia investment in developing a network data model is paying dividends as increasingly more uses for the data are being identified. Configuration auditing is but one of its uses.

4. REFERENCES

1 Cisco Systems, 2003. Copyright. How Cisco IT-LAN-SJ achieved high availability. http://www.cisco.com/web/about/ciscoitatwork/downloads/ciscoitatwork/pdf/cisco_it_high_availability.pdf (12 May 2007).

2. Office of Government Commerce, Best Practices for Service Support: ITIL The Key to Managing IT Services. Published by The Stationery Office, London, 2000.

3 . Index of /modeling/reports/network_audits, https://sadl23693.sandia.gov/modeling/reports/network_audits/

4 . Sandia National Laboratories, March 2007. Change management process. https://wfsprod01.sandia.gov/groups/srn-uscitizens/documents/other/wfs009546.pdf (12 July 2007).

5. Sandia National Laboratories, April 2007. Trouble resolution process. https://wfsprod01.sandia.gov/groups/srn-uscitizens/documents/other/wfs004128.pdf (27 June 2007).

6. Sandia National Laboratories, March 2007. Configuration guide for Cisco xxxx router/switch (27 June 2007). https://wfsprod01.sandia.gov/intradoc-cgi/idc_cgi_isapi.dll?IdcService=GET_SEARCH_RESULTS&QueryText=xCollectionID=5205&SearchProviders=WFS_Prod,&ftx=&AdvSearch=True&SortField=dDocTitle&SortOrder=Asc&ResultCount=25&ResultTemplate=

7. Moody, R., Klaus, E., 2006. Sandia National Laboratories. *Configuration Monitoring Process*. Internal document.

8 Friedl, J.E.F, Mastering Regular Expressions, Second Edition. O'Reilly Media Inc.

9 Kuchling, A. M. Regular expression how to. http://www.amk.ca/python/howto/regex (March 2007).

BIBLIOGRAPHY

Cisco Systems, 2007. Copyright. Configuration management: Best practices white paper. http://www.cisco.com/warp/public/126/configmgmt.html#topic12 (May 2007).

Werner, J. and Schouls, M. (2005). IT gets it. http://support.novell.com/techcenter/articles/nc2005_042.html (May 2007).

APPENDIX: WORK AGREEMENT – CONFIGURATION COMPLIANCE AUDITING CAPABILITY

J.H. Maestas, 04336, March 2007

Primary Goals

Several goals are to be accomplished during this project:

1) Develop the rules that are needed to analyze the configurations of deployed network devices against a list of commands specified by the Network Configuration Standardization Team (NCST) in Organizations 04334 and 04338.

2) Train an individual in 04338 to perform the audits as needed by the NCST. To train this individual to maintain the rules as needed by the NCST. A long term goal is to enable 04338 staff (employee or contractor) to assume responsibility for analyzing network configurations for the NCST.

3) Evaluate network configurations against Cisco best practices such as the Cisco SAFE Blueprint for Enterprise Networks and other protocol configuration requirements that can lead to increased mean time to repair (MTTR) and decreased availability. Provide these analyses to managers in 04334, 04336, and 04338.

Stakeholders

Network Design and Implementation, 04334
Advanced Networking and Integration, 04336
Systems Analysis and Trouble Resolution, 04338

Constraints

1. During rule development, work with the NCST to ensure the rules are accurate and produce the desired results.

2. Organization 04338 must assign an individual to perform the configuration audits and to maintain the configuration rules as required by the NCST. The selected individual(s) is (are) to work with 04336 during rules development for on the job training, OJT.

3. The rule development work performed by 04336 for this agreement should continue until 04338 staff are comfortable with maintaining the rules and performing the configuration audits. However, a sufficient effort and concentration time must be provided by all involved parties.

4. The work in this agreement is purposed to dovetail with the configuration monitoring process currently being developed by 04338.

5. Time constraints in the deliverables ensure that progress continues and additional device types are added to the configuration-managed device list.

4336 Deliverables

1. Develop the rules needed to analyze the configurations of network devices including Cisco and Foundry systems. Rules development for a given device type should be completed within 14 working days. If additional time is required, the NCST must be notified.

2. Monitor the commands in the baseline command list — refer to 04338 deliverables for the list — on all device types and operating system (OS) types as appropriate. Develop rules accordingly for each device type. The NCST may request that additional commands be monitored for a given type of device. Such commands are not to be added to the baseline list unless requested by the NCST.

3. Provide Organization 04338 with complete documentation. The scope of the document is limited to:
    a. Explaining how the rules work and how they can be expanded to audit other commands and device types. The purpose of the document is not to instruct the user to write regular expressions.
    b. Documenting the complete workflow, beginning with data import to rules selection to report publication.

4. Provide 04338 personnel with in-house training on how to use the configuration analysis technology and how to expand the rule set to audit additional device types.

4338 Deliverables

1. Provide 04336 with the device type to be monitored.

2. Provide 04336 with a baseline list of commands to be monitored as appropriate for the device type and OS type. The baseline list has been defined by the NCST: banner, ntp servers, time zone, service time stamps, mac-address aging, version, boot flash, access-lists 80, 81, and 86, default route, snmp community strings, snmp server hosts, logging trap errors and servers, tacacs-server, dns, vtp domain, management interfaces sc1 and sc0.

3. Provide 04336 with additional commands, as needed, to be monitored that are not currently a part of the baseline list.

4. Within 14 days of receiving the audit report, the NCST will provide feedback to 04336 for rule adjustment and correction. Within this same 14-day period, the NCST will correct the device configurations, as necessary. If additional time is required, 04336 must be notified.

Resources

DCs

- 1 license – Modeler or ITGuru. Annual license maintenance fees apply.
- 1 license – NetDoctor. Annual license maintenance fees apply.

FTEs

- Approximately 0.25 - 0.5 FTE

Project Members/Responsibilities

- Joseph Maestas, 04336: rules development
- Ron Moody, 04338: NCST member: provide command list, validate reports, re-configure devices as needed, provide feedback
- Ed Klaus, 04338: NCST member: provide command list, validate reports, re-configure devices as needed, provide feedback
- Phillip Ayala, 04338: Perform audits and develop/maintain rules

Approval: Original signed by Pat Date: 3/13/2007
Pat Manke, 04338

Approval: Original signed by Len Date: 3/14/2007
Leonard Stans, 04336

Comments by Stakeholders:

1. Provide hard delivery dates on deliverables: Each scan of a specific device type is a moving target and varies with the number of issues found within the device configurations, the documented standards, and the customized rules. Convergence in the three areas marks the end of one scan cycle and the beginning of the next of a new device type. Several iterations are often needed to achieve the desired results.

2. A technical report should be one of the deliverables: Item 3 listed in the 09336 deliverables is intended to be a Sandia Technical Report. The estimated date for completion is July 2007.

DISTRIBUTION

1	MS 0662	T. Klitsner	09334
1	MS 0672	B. P. VanLeeuwen	05614
1	MS 0788	P. A. Manke	09338
1	MS 0788	D. R. Garcia	09334
1	MS 0788	M. J. Hamill	09334
1	MS 0788	E. Klaus	09838
1	MS 0788	R. L. Moody	09338
1	MS 0788	D. R. Porter	09338
1	MS 0788	D. L. Van Houten	09335
1	MS 0788	V. K. Williams	09334
1	MS 0795	P. D. Warner	09317
1	MS 0799	B. A. Potts	09318
1	MS 0799	T. Bruner	09318
1	MS 0799	J. A. Chavez	09318
1	MS 0801	R. W. Leland	09300
1	MS 0801	D. S. Rarick	09330
1	MS 0809	P. D. Ayala	093385
1	MS 0801	R. W. Leland	09300
1	MS 0801	D. R. White	09340
1	MS 0806	L. Stans	09336
1	MS 0806	J. L. Akins	09336
1	MS 0806	J. P. Brenkosh	09336
1	MS 0806	J. M. Eldridge	09336
1	MS 0806	S. A. Gossage	09336
2	MS 0806	J. H. Maestas	09336
1	MS 0806	L. G. Martinez	09336
1	MS 0806	J. H. Naegle	09336
1	MS 0806	T. J. Pratt	09338
1	MS 0806	L. F. Tolendino	09334
1	MS 0806	J. S. Wertz	09336
1	MS 0813	R. M. Cahoon	09311
1	MS 0832	J. H. Dexter	09335
1	MS 0823	J. D. Zepper	09320
1	MS 9012	B. A. Maxwell	08949
1	MS 9012	C. T. Deccio	08949
1	MS 9012	R. D. Gay	08949
2	MS 9018	Central Technical Files, 08944	
2	MS 0899	Technical Library, 04536	