



realtimepublishers.comtm

Tips and Tricks *Guidetm To*

Network Configuration Management

2005 Edition

AlterPoint

Don Jones

Introduction to Realtimepublishers

by Sean Daily, Series Editor

The book you are about to enjoy represents an entirely new modality of publishing and a major first in the industry. The founding concept behind Realtimepublishers.com is the idea of providing readers with high-quality books about today's most critical technology topics—at no cost to the reader. Although this feat may sound difficult to achieve, it is made possible through the vision and generosity of a corporate sponsor who agrees to bear the book's production expenses and host the book on its Web site for the benefit of its Web site visitors.

It should be pointed out that the free nature of these publications does not in any way diminish their quality. Without reservation, I can tell you that the book that you're now reading is the equivalent of any similar printed book you might find at your local bookstore—with the notable exception that it won't cost you \$30 to \$80. The Realtimepublishers publishing model also provides other significant benefits. For example, the electronic nature of this book makes activities such as chapter updates and additions or the release of a new edition possible in a far shorter timeframe than is the case with conventional printed books. Because we publish our titles in “real-time”—that is, as chapters are written or revised by the author—you benefit from receiving the information immediately rather than having to wait months or years to receive a complete product.

Finally, I'd like to note that our books are by no means paid advertisements for the sponsor. Realtimepublishers is an independent publishing company and maintains, by written agreement with the sponsor, 100 percent editorial control over the content of our titles. It is my opinion that this system of content delivery not only is of immeasurable value to readers but also will hold a significant place in the future of publishing.

As the founder of Realtimepublishers, my *raison d'être* is to create “dream team” projects—that is, to locate and work only with the industry's leading authors and sponsors, and publish books that help readers do their everyday jobs. To that end, I encourage and welcome your feedback on this or any other book in the Realtimepublishers.com series. If you would like to submit a comment, question, or suggestion, please send an email to feedback@realtimepublishers.com, give feedback on our Web site at <http://www.realtimepublishers.com>, or call us at 800-509-0532 ext. 110.

Thanks for reading, and enjoy!

Sean Daily
Founder & Series Editor
Realtimepublishers.com, Inc.

Note to Reader: This book presents tips and tricks for seven network configuration management topics. For ease of use, the questions and their solutions are divided into topics, and each question is numbered based on the topic, including

- Topic 1: Configuration Management Best Practices
- Topic 2: Network Management Security
- Topic 3: Network Configuration Troubleshooting
- Topic 4: Change Configuration Management Technologies
- Topic 5: Selecting and Deploying a Network Configuration Management Solution
- Topic 6: Enterprise Network Configuration Management
- Topic 7: Compliance Management for the Network

Introduction to Realtimerepublishers	i
Topic 1: Configuration Management Best Practices	1
Q 1.1: How can we ensure that our configuration management continues to meet best practices? 1	
Q 1.2: What is configuration change management, and why should I care?	2
Q 1.3: What is the best way to “do” configuration change management with network devices? ...3	
Planning for Change	3
Identify Risks	4
Categorize Risks	5
Mitigate Risks	6
Prioritize Changes	8
Managing Changes.....	8
Want to Know More?.....	9
Q 1.4: How can I prevent overzealous administrators from making unauthorized changes to network devices?.....	9
Q 1.5: How can I ensure uniform device configuration throughout my organization?	11
Q 1.6: How can I reduce network device problems through configuration management?.....	13
Inventory	13
Configuration Management	14
Audit Trails	14
Network Topology	14
Topic 2: Network Configuration Management Security	15
Q 2.1: How does network management contribute to an overall information security plan?	15

Q 2.2: We manage network devices by using Simple Network Management Protocol. Are there security risks?.....16

Q 2.3: How can configuration management improve network security?18

Q 2.4: How will wireless devices change the way I secure network devices?.....20

Q 2.5: Network device security updates are issued every week. How can we ensure that all of our administrators heed them?22

Q 2.6: My company considers network configuration information to be confidential. How can I ensure that this information is secure?.....24

 Securing Information on Devices25

 Securing Information in Transit.....26

 Securing Information in Storage.....27

Q 2.7: Can I use TACACS+ for device authentication?.....29

 What Does TACACS+ Do?29

 Implementing the TACACS+ Server.....31

 Configuring Devices to Use TACACS+.....31

 Authorization33

 Accounting.....33

Topic 3: Network Configuration Management Troubleshooting34

Q 3.1: We are having difficulty determining who made changes to network devices when configuration troubles occur. What can we do?.....34

Q 3.2: What is the first step toward fixing a router that isn't working?36

Q 3.3: How can configuration management contribute to improved network performance?37

Q 3.4: What are some industry best practices for troubleshooting network devices?39

Q 3.5: How can I determine whether a new product or a consultant makes changes to our network devices?.....39

 Manually Detecting Changes40

 Proactive Change Notification.....42

 Automation on the Cheap44

 Automating the Configuration File Dump44

 Automating the File Comparison.....45

 Emailing the File Comparison Results45

Q 3.6: What is the best way to start troubleshooting router problems?.....46

Topic 4: Configuration Change Management Techniques47

Q 4.1: Which new technologies can help ease network configuration management?.....47

Q 4.2: How can I back up all of my network devices?.....49

Q 4.3: What is the easiest way to detect unauthorized changes in the configuration of routers and other network devices?50

Q 4.4: Short of buying a dedicated software application, how can I implement change management for network device configurations?54

Q 4.5: Our branch office routers are identical, yet users in one office say their router is slower than another office’s router. What’s the difference?55

Q 4.6: How can I ensure that all of the devices in my enterprise are consistently configured?57

 Creating Standards58

 Standardizing Versions58

 Standardizing Addressing59

 Standardizing Naming60

 Standardizing Configurations60

 Creating Configuration Templates.....61

 Ensuring Adherence to Standards62

Topic 5: Selecting and Deploying a Network Configuration Management Solution63

Q 5.1: How do network configuration management solutions support policy-based management?63

Q 5.2: All of our equipment is from one vendor. Why not use a vendor-supplied device management solution?66

Q 5.3: We’re preparing to roll out a device management solution. However, we have hundreds of devices. What’s the best way to proceed?67

Topic 6: Enterprise Network Configuration Management70

Q 6.1: How does policy-based management make it easier to manage a large numbers of devices?.....70

Q 6.2: We have hundreds of network devices, so manually retrieving configurations via Trivial File Transfer Protocol just isn’t an option. What are our alternatives?75

Topic 7: Compliance Management for the Network77

Q 7.1: How can policies help us better manage regulatory compliance for our network?.....77

Q 7.2: What does my network configuration have to do with compliance?.....79

Q 7.3: How can we make our network verifiably compliant?80

Q 7.4: How effective are audits at maintaining compliance in network devices?81

Q 7.5: Once my network is compliant, how can I ensure it stays that way?82

Copyright Statement

© 2005 Realtimedpublishers.com, Inc. All rights reserved. This site contains materials that have been created, developed, or commissioned by, and published with the permission of, Realtimedpublishers.com, Inc. (the "Materials") and this site and any such Materials are protected by international copyright and trademark laws.

THE MATERIALS ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. The Materials are subject to change without notice and do not represent a commitment on the part of Realtimedpublishers.com, Inc or its web site sponsors. In no event shall Realtimedpublishers.com, Inc. or its web site sponsors be held liable for technical or editorial errors or omissions contained in the Materials, including without limitation, for any direct, indirect, incidental, special, exemplary or consequential damages whatsoever resulting from the use of any information contained in the Materials.

The Materials (including but not limited to the text, images, audio, and/or video) may not be copied, reproduced, republished, uploaded, posted, transmitted, or distributed in any way, in whole or in part, except that one copy may be downloaded for your personal, non-commercial use on a single computer. In connection with such use, you may not modify or obscure any copyright or other proprietary notice.

The Materials may contain trademarks, services marks and logos that are the property of third parties. You are not permitted to use these trademarks, services marks or logos without prior written consent of such third parties.

Realtimedpublishers.com and the Realtimedpublishers logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners.

If you have any questions about these terms, or if you would like information about licensing materials from Realtimedpublishers.com, please contact us via e-mail at info@realtimedpublishers.com.

Topic 1: Configuration Management Best Practices

Q 1.1: How can we ensure that our configuration management continues to meet best practices?

A: The easiest way to be sure that your configuration management meets best practices is to have a tool or a set of tools that help enforce whatever business processes you have identified as being “best practices.” For example, many organizations implement workflow processes for configuration management. Figure 1.1 shows a simplified management process that could benefit—as the callouts highlight—from solutions to help enforce the workflow.

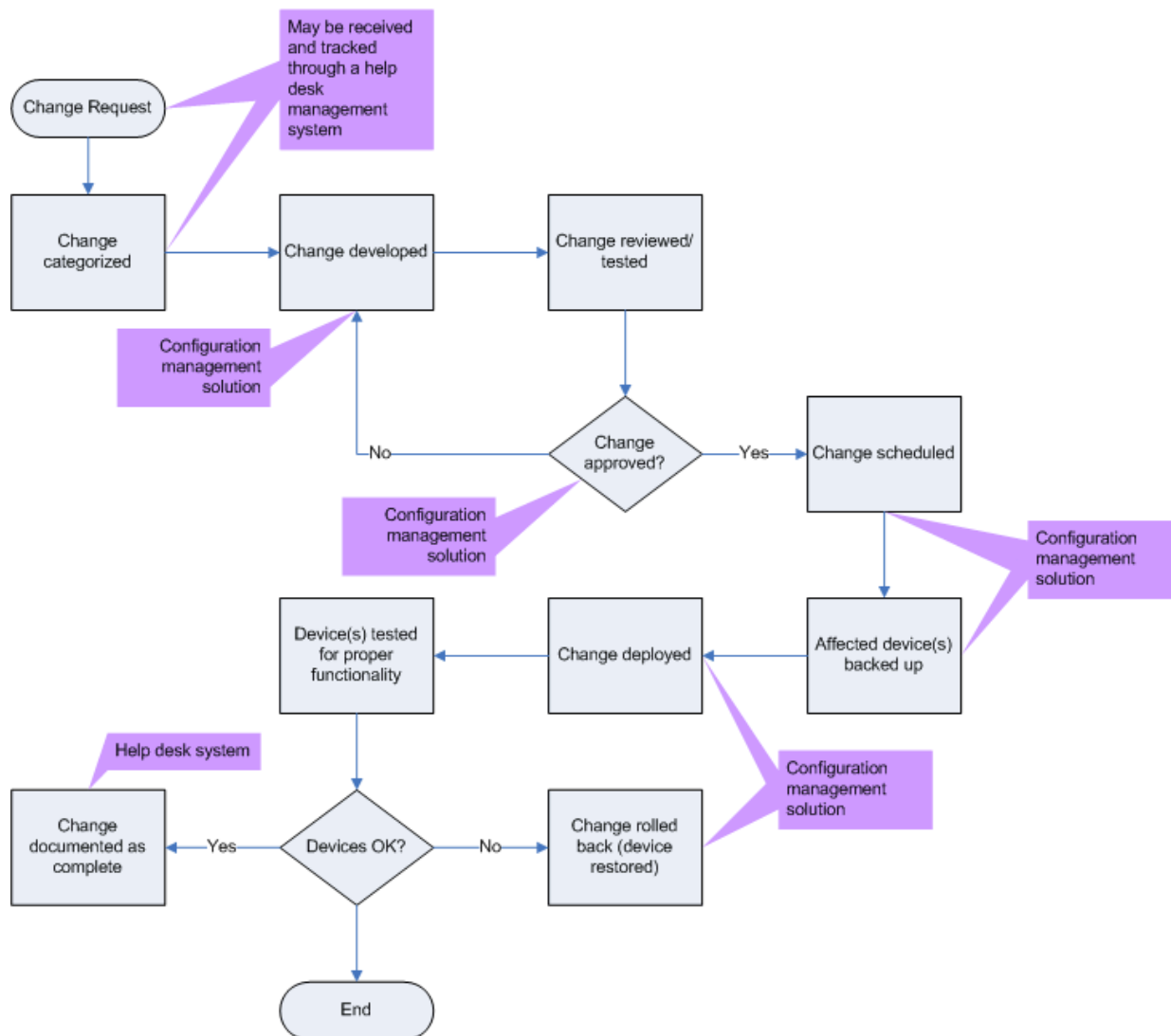


Figure 1.1: Simplified configuration management process.

This process includes workflow steps for receiving and categorizing changes (by risk, priority, and so on); a process that can generally be assisted within a Help desk management system. The change is then developed and reviewed—an important step that can be enforced by a network configuration management solution that includes customizable workflow capabilities.

Essentially, the solution becomes the only interface through which changes are introduced into the network, and the solution itself implements its own level of security to ensure that only approved individuals can enter changes into the system, and that only authorized individuals can review and approve those changes for a production deployment. The solution can also automatically schedule the change for deployment; that deployment can include an automated backup of targeted devices' existing configurations. Those backups can be used to roll back the change if the change does not function as expected or causes a problem.

It is always possible for an administrator to modify a device's configuration directly, bypassing the solution you have implemented and the workflow that it enforces. One way to help prevent such activity is to ensure that your network devices are configured to log activity to a syslog server or a Remote Authentication Dial-In User Service (RADIUS)/Terminal Access Controller Access Control System (TACACS) accounting server or to send SNMP traps. All of these forms of logging can be intercepted by a network configuration management solution, triggering the solution to query the device's configuration via Trivial File Transfer Protocol (TFTP) or by other means. The solution can then compare the device's configuration with the device's last approved configuration, and if any differences are discovered, roll back the device configuration or, at the very least, notify a manager of the discrepancy. This scenario doesn't *prevent* out-of-process changes from occurring, but it can help detect them quickly and remediate them, if necessary, returning devices to their known-good configuration state.

Q 1.2: What is configuration change management, and why should I care?

A: No matter how large or small your network environment, change is inevitable. Hiring new employees, adding new offices, supporting new network services, improving security, fixing bugs—all of these activities result in change, especially to your network infrastructure devices, such as routers, switches, hubs, firewalls, and so forth. Although change is almost always a good thing in the end, change can cause bad things to happen. For example, a careless typo in a firewall configuration file could have alarming security implications. So no matter how minor or beneficial a change may be, you should always approach change with a healthy dose of caution. *Configuration change management* is a set of policies and procedures that you adopt and follow to formalize that caution into a repeatable, consistent process.

At its simplest, configuration change management simply means keeping track of the changes you make and evaluating proposed changes for their effect before actually implementing them. In practice, change management involves some fairly well-defined tasks:

- Maintaining documentation that describes the current configuration of all network devices
- Maintaining documentation that describes the purpose and details of any changes
- Maintaining an archive of older configurations so that they can be used in an emergency
- Implementing policies that control the rate of change
- Implementing policies that control who may perform changes

Why should you bother with all of that? Primarily, to improve network uptime. Unauthorized or unplanned changes are the number one cause of network device failures and unplanned downtime for organizations. Failure to document current configurations makes it difficult, if not impossible, to recover gracefully from a failed change procedure. Failure to control the rate of change as well as who can make changes results in an inconsistent environment that is difficult to maintain long-term. Unauthorized changes can also result in the network no longer complying with regulations or legislation that might apply to your organization, such as the Health Insurance Portability and Accountability Act (HIPAA), the Sarbanes-Oxley Act, the Graham-Leach-Bliley Act, and so forth. Changes that take your network out of compliance can result in hefty fines, government penalties, loss of customers, and so forth.

Instead of thinking of change management as extra work, consider change management a measure that *saves* your organization work: By simply following some simple methodologies and processes, you can ensure that changes to network devices never become a nightmare. Or, at least, if they *do* become a nightmare, you can quickly recover without having to spend all night at the office!

Q 1.3: What is the best way to “do” configuration change management with network devices?

A: The actual mechanics of change management depend on which types of devices and tools you have on your network; the ways in which you should conduct a change management program, however, are universal. There are two main steps to a change management program: planning and management.

Planning for Change

Too many change management methodologies ignore the planning phase, which is perhaps the most important. Planning allows you to identify and reduce risk, provide a means to rollback in case of disaster, and so forth. Essentially, planning requires you to

- Identify everything that could possibly go wrong as a result of a change.
- Assign a level of likelihood and severity to each potential risk.
- Identify means of mitigating risks or, at least, provide a means of recovery should the risk actually become a reality.

A solid change management planning methodology will make it easier for you to prioritize changes according to their business impact. For example, if you find yourself making several high-risk, low-benefit changes, you can implement policies to reduce such activity, for example, by adopting a policy of only making low-benefit changes during a regular update cycle, such as at the end of each month.

How you actually conduct each step of the planning process depends on your environment and your personal preferences. The next four sections provide some examples to get you started.

Identify Risks

What might go wrong when you update the routing table on one of your routers? Many possibilities spring to mind:

- You could mistype something and corrupt the entire routing table, making the router functionally useless.
- You could enter incorrect information, preventing the change from working properly.
- You could enter incorrect information that makes existing routes stop working correctly.
- While uploading changes to a router, you could lose your network connection, resulting in a partial change to the router.
- You could upload changes to the wrong router, causing routing problems across the network.
- Any incorrect changes you make could replicate across your network through routing protocols, corrupting all of your network devices.

The objective with your risk list is to identify everything that could *possibly* go wrong, not just the things that are *likely* to go wrong. Keep in mind that changing the configuration of *any* network device, not just a router, creates a set of potential risks.

☞ Keep your risk lists handy! After you've developed a list of risks for a particular type of change, such as a router update or a firewall change, keep that list. You are likely to make the same type of change again in the future, so there is no reason to unnecessarily repeat the risk-identification process. You will be building your risk list into a checklist for *avoiding* risks, so the list can become part of your network's change management documentation and act as a list of procedures to be followed to help avoid unnecessary risk during network device management.

Categorize Risks

After you've got a list of everything that could go wrong, assign likelihood and a severity to each item. I prefer a simple scale of 1 to 3, where 1 represents highly unlikely risks, or risks that would be very minor if they did occur, and 3 represents risks that are likely to occur and would be very severe if they did. Working with the previously created list of potential risks, you might assign the following ratings:

- You could mistype something and corrupt the entire routing table, making the router functionally useless—likelihood is 2, severity is 3. The likelihood is high because you manually type all the router configuration information and, although you're always careful, there is no data-validation process in place.
- You could enter incorrect information, preventing the change from working properly—likelihood is 2, severity is 1. Severity is less than that of the first risk because you're simply failing to implement the change, not affecting anything else.
- You could enter incorrect information that makes existing routes stop working correctly—likelihood is 2, severity is 2. The severity is 2 for this risk because you're affecting an entire device.
- While uploading changes to the router, you could lose your network connection, resulting in a partial change to the router—likelihood is 1 because you have backup power supplies everywhere and a very reliable network; severity is 2 because if the risk did occur, it would take the entire device offline.
- You could upload changes to the wrong router, causing routing problems across the network—likelihood is 1 because you are careful; severity is 2 because if you did make this blunder, you would ruin an entire router.
- Any incorrect changes you make could replicate across your network through routing protocols, corrupting all your network devices—likelihood is 3 because if you do make an incorrect change, it will propagate fairly rapidly; severity of 3 because this mistake could potentially take your entire network offline.

The purpose of this list is to help identify the risks that are in most need of specific mitigation. The risk list for a switch reconfiguration might include similar items, but the risks listed would be unique to switches; the same can be said of firewalls, managed hubs, or any other network device. One simple way to rank your risks is to add your two ratings, giving you a prioritized list of things that could go wrong:

- Any incorrect changes you make could replicate across your network through routing protocols, corrupting all of your network devices—risk: 6.
- You could mistype something and corrupt the entire routing table, making the router functionally useless—risk: 5
- You could enter incorrect information that makes existing routes stop working correctly—risk: 4
- You could enter incorrect information, preventing the change from working properly—risk: 3

- While uploading changes to the router, you could lose your network connection, resulting in a partial change to the router—risk: 3
- You could upload changes to the wrong router, causing routing problems across the network—risk: 3

With this list in hand, you are ready to start planning ways to avoid these risks and, should the worst happen, recover as quickly as possible. Again, although I'm using a router in this example, you will want to prioritize the risks associated with changing any type of network device.

Mitigate Risks

Risk mitigation is a planning process in which you try to think of ways to prevent your identified risks from ever occurring; while at the same time coming up with a means of recovery should the risk become a reality in spite of your efforts. Add the mitigation and recovery ideas to your list to create a risk-avoidance and recovery checklist:

- Any incorrect changes you make could replicate across your network through routing protocols, corrupting all your network devices.
- Avoidance—Disable routing protocols on router until change is verified by a senior administrator.
- Recovery—Ensure that a backup of all router configurations is available before you make a change. In the event that incorrect data propagates, immediately restore device configurations from backup.
- You could mistype something and corrupt the entire routing table, making the router functionally useless.
- Avoidance—Use vendor-supplied tools to make changes rather than manually entering changes. Vendor tools provide some data validation to help prevent data-entry errors. Also, document all changes and have another administrator review and approve them for accuracy. Have the other administrator verify the accuracy of the changes after they are made.
- Recovery—Back up the device configuration before making a change. Immediately restore the device configuration if changes made do not comply with the change documentation.
- You could enter incorrect information that makes existing routes stop working correctly or prevents the change from working properly.

Avoidance—Use vendor-supplied tools to make changes rather than entering changes directly in router. Vendor tools provide some data validation to help prevent data entry errors. Also, document all changes on paper and have another administrator review and approve them for accuracy. Have the other administrator verify the accuracy of the changes after they are made.

Recovery—Back up the device configuration before making a change. Immediately restore device configuration if changes made do not comply with the change documentation. Always have a workstation available that can make a direct serial connection to the router in case an incorrect change disables the router's ability to communicate via network interface.

- While uploading changes to the router, you could lose your network connection, resulting in a partial change to the router.

Avoidance—Ensure that router, administrative workstation, and intermediate devices (hubs and switches) are on power backup. If possible, place an administrative workstation on same network segment as the router to be changed to eliminate the possibility of an intermediate router failure during upload.


Recovery—Back up the device configuration before making a change. Ensure that the router being changed is accessible to a local-segment workstation on which the back up resides, allowing easier restore. Always have a workstation available that can make a direct serial connection to the router in case an incorrect change disables the router's ability to communicate via network interface. As a last-ditch recovery method, many network devices offer a hardware reset switch that restores the device's factory configuration. Combined with a recent configuration backup, you can use this reset function to quickly get the device up and running again.

- You could upload changes to the wrong router, causing routing problems across the network.

Avoidance—Have another administrator confirm your changes and settings prior to upload.


Recovery—Back up all network devices before making a change. If data is uploaded to the wrong device, restore that device's configuration from backup. Always have a workstation available that can make a direct serial connection to the router in case an incorrect change disables the router's ability to communicate via network interface.

Some network devices, such as managed hubs and switches, might offer simpler recovery methods. Some managed hubs, for example, can create a backup of the last-known good configuration to a built-in flash RAM module, and let you recover that configuration with a hardware reset switch. Other network devices, such as firewalls, might require more extensive planning to ensure that a fast recovery is possible.

 After you have developed a complete risk list, including mitigations, for a particular type of change, save it! This list should become a checklist for all future changes of the same type. By following the checklist each time you make that type of change, you will automatically mitigate the potential risks as well as have prepared recovery options in case the worst happens. If your network administration is primarily accomplished by junior administrators, these mitigation lists can become a mandatory part of the procedures the administrators follow, helping ensure that you're sort of looking over their shoulder, even when you're not.

Prioritize Changes

Don't get into the habit of making every change that pops into your head. Prioritize changes based on their impact on business operations. You can use a simple 1-to-3 scale or something more complex. High-priority changes are worth more risk, of course, while lower-priority changes—especially those with a high-risk rating—should be put off until they can be made under tightly controlled circumstances. For example, some companies save all low-priority changes until the end of the month. Before implementing any changes, they carefully review them all. They also back up every single network device in case something goes horribly wrong, and they put the necessary support personnel on alert. This process requires a lot of effort and isn't something that these companies want to go through on a daily basis.

 Software tools can make this process easier, of course, by automatically backing up devices prior to deploying a change, and by automatically deploying changes for you on a schedule you set.

For emergency changes that need to be implemented immediately, the companies have a fast-track process that requires two senior administrators to approve and implement the change; the idea being that senior administrators have enough experience to pull off the change with less risk. How you prioritize and handle changes really becomes a matter of change management policy, which I'll discuss next.

Managing Changes

Changes can easily get out of control, and the only way to rein them in is to have in place a firm set of change management policies that all administrators are required to follow. For example, you might implement a change management policy as follows:

- All changes must be documented and approved by a senior administrator. Change documentation must include the current state of the device as well as the proposed change.
- Changes identified as high-priority require a senior administrator's approval. All other changes require the approval of two administrators, including at least one senior administrator.
- All changes must include a detailed description of the intent of the change (for example, To allow the Nevada office to communicate directly with the Seattle office rather than communicating through the New York hub office).
- All completed changes will be reviewed at a weekly meeting of administrators. This meeting will help make all staff aware of recent changes and allow an opportunity to review failed changes.
- Changes classified as emergency priority can be made only by two senior administrators working together. These changes can bypass the normal review process, but that process will be completed as soon as possible after the change is complete to ensure that a complete set of documentation for the change is created.

The actual policies your company might adopt may differ; however, the important point is to have some procedural guidelines in place.

☞ You can use software tools to help enforce a change management methodology. For example, some tools let changes be developed and deployed to test devices but require a second administrator or manager to review and approve changes before they can be deployed to your production devices.

Want to Know More?


No matter what you do, make sure that you have a system in place for change management. If you would like some ideas for how to physically implement such a system, check out the University of Kentucky's Change Control FAQ, located online at <http://www.uky.edu/~change/faq.html>. This site should give you some ideas of how a change management system works at a very high level, including change requests, tracking, and so forth. You should also check out Cisco System's excellent white paper about change management, available at <http://www.cisco.com/warp/public/126/chgmt.shtml>. This white paper provides a great overview of change management and gives detailed examples of process flows. The white paper also provides examples of change management documentation, which can help jump-start a new change management process in your organization. Another great resource is the Information Technology Infrastructure Library (ITIL) at <http://www.ogc.gov.uk/index.asp?id=2261>, which provides several best-practices frameworks for IT management, including change management.

Q 1.4: How can I prevent overzealous administrators from making unauthorized changes to network devices?


A: First, realize that many administrators feel that they're doing users and the company a favor by performing so-called "minor" changes without following their company's sometimes complex configuration management process. Some administrators are frustrated by the politics involved with making a change to a network device and dislike the fact that they can't simply reconfigure their routers when they need to do so. Your first step is to overcome that mindset and make sure that all administrators understand the purpose and benefits of the change-management process:

- In the end, configuration management reduces work—Changes are less likely to cause failures and recovery is easier in the event of a problem, resulting in fewer late nights spent at the office.
- Configuration management shares the responsibility for making changes—A good process includes several sign-offs, eliminating the need for a single person to bear the brunt of mistakes.
- A well-designed configuration management process can reduce stress by eliminating the "do it now!" demands often placed on network administrators—The process can help absorb that stress by regulating change requests into a manageable stream.
- Configuration management processes protect the organization's level of compliance with legal and other business requirements—In the rush to make a change on their own, administrators may not often realize the impact the change might have on compliance. Administrators may not even be fully trained in all of the organization's compliance needs, making the configuration management process the only means by which those needs are met.

In addition to changing administrators' perception of the configuration management process, you can take advantage of the fact that most network devices offer a physical means of ensuring changes occur only when authorized (that is, through passwords). I've worked in environments in which utilities were used to automatically change router passwords every day. Before they could make changes, administrators had to check out the day's password, which forced the administrators to follow procedure. With a robust configuration management solution (such as AlterPoint Device Authority or Voyence VoyenceControl), you might be able to automate password changes. Doing so can help to ensure that the configuration management solution—rather than individual administrators—know devices' passwords.

 As network device security and configuration management become more critical to organizations' compliance and continued operations, future configuration management solutions may simply offer an automated password management option that you enable with a check box. This option would regularly change devices' administrative passwords, keeping the current password on file in a protected database. Such an option would virtually eliminate the ability for individual administrators to bypass the configuration management solution, while still offering an emergency means of accessing devices directly.

Other utilities can retroactively catch unauthorized changes. Doing so makes it easier to correct or undo unauthorized changes before they cause problems and to educate the offending administrator about the proper configuration management process in your organization. Unfortunately, assuming your administrators have password access to your devices, there is almost no way to prevent them from making changes outside of your change-control process.

 The newest configuration management solutions allow administrators to build device configuration changes in a graphical user interface (GUI). The application then pushes the changes to the network devices by using the devices' configuration passwords on a defined schedule. Administrators don't actually need to know the configuration passwords; instead, the administrators will authenticate to the application separately, perhaps using their regular network-security credentials. The result will be a front-end application that provides business rules and processes to the change process, then pushes authorized changes to devices on the back end.

You can utilize software tools such as Tripwire (<http://www.tripwire.com>), which periodically logs onto your network devices and compares their configuration with a known-good *baseline* configuration. Changes to the configuration generate an email alert, giving a senior administrator the opportunity to analyze the change and either accept it—making it part of the baseline—or reject it, causing the original baseline configuration to be restored to the device. Other tools, such as AlterPoint's DeviceAuthority, offer detailed configuration management reports, which you can use to not only review the specific changes made to your network devices but also to get a better idea of the type and volume of changes made to your devices over specific periods of time. Using configuration management tools such as these will help centralize configuration management and prevent individual administrators from bypassing your configuration management process.

Q 1.5: How can I ensure uniform device configuration throughout my organization?

A: Companies with a large number of network devices often have difficulty maintaining consistent configurations across those devices. The benefits of consistency are fairly numerous:

- Consistent configurations make it easier to train new administrators and make it easier for administrators to take over each other's tasks based on workload.
- Consistency improves network reliability by using tried-and-true configurations on all devices.
- Consistency simplifies troubleshooting because the standard configuration has predictable, known behaviors. In addition, deviations from the standard can be easily detected by simple file comparisons.
- Consistency makes compliance auditing easier because there are fewer variations for an auditor—who may have minimal technical experience—to deal with.

Some of the high-end network device management solutions include the ability to enforce consistent device configurations within an enterprise. A senior administrator develops configuration policies, which are enforced by the software on new configuration changes. Most packages with this capability can also review existing configurations for compliance with policies, allowing you to retrofit the software into an existing environment and clean up inconsistent configurations.

You don't necessarily need fancy software to enforce consistency, though. You can create configuration templates quite easily, making it easier for other administrators to use the same configuration settings and syntax across your organization.

Start by configuring a single device to be a model of your new, standardized configuration. Get the device's configuration into a text file either through Trivial File Transfer Protocol (TFTP), FTP, HTTP, or whatever other means the device supports. Then modify the text file—adding comments where necessary—into a template. Listing 1.1 shows an example for a Cisco Catalyst switch.



Note that the exclamation marks in this sample file are comment lines and don't affect the actual configuration.

```
.
!  
!! For Cat switches / firmware 3 / template v4  
!  
interface FastEthernet0/1  
  description [officename]_local1  
  duplex half  
  speed 10  
!  
interface FastEthernet0/2  
  description [officename]_local2  
  duplex half  
  speed 10  
!  
interface FastEthernet0/3  
  description [officename]_local3  
  duplex half  
  speed 10  
!  
interface FastEthernet0/4  
  description [officename]_local4  
  duplex half  
  speed 10  
!  
interface FastEthernet0/9  
  description [officename]_backbone  
  duplex full  
  speed 100  
!  
interface FastEthernet0/10    !!! Omit speed and duplex  
  description [officename]_lab !!! for autoconfiguration  
!  
interface: FastEthernet0/11  
  description [officename]_admin  
  duplex half  
  speed 10  
!  
interface VLAN1  
  ip address [ipaddress] [subnetmask]  
  no ip directed-broadcast  
  no ip route-cache  
!  
.
end
```


Listing 1.1: An example for a Cisco Catalyst switch.

👉 Use version numbers! This sample configuration includes not only its own version number but also a comment to tell administrators which type of device and version of the device's firmware is required to use the template.

Notice in this example that several replacement variables, in [brackets], are included in the text. To use this template, simply use a text editor's search and replace feature to replace the variables. For example, replacing [officename] with newyork will result in the desired interface names, newyork_local, newyork_backbone, and so forth. Create a separate document that describes the variables in use. For example:

- [officename] = Replace with the name of the city in which the device is installed.
- [ipaddress] = Replace with the device's VLAN1 IP address (obtain the IP address from master tracking list).
- [subnetmask] = 255.255.255.0 in all field offices and 255.255.0.0 in all labs (see IP master tracking list for exceptions for certain subnets).

After you edit the file, you can load it into the new device. Most devices support a means of loading configuration files.

 Store your configuration templates in a version control system. Using a version control system allows you to retrieve older configuration templates in the event of a problem with a new template.

In addition to making it easier to ensure consistent device configuration, configuration templates make it easier to deploy new devices throughout your enterprise. Rather than having to follow a complex set of configuration instructions—or worse, configuring new devices from memory—you'll have an easy-to-use template that you can quickly complete, load into the device, and place into production.

Whenever you make changes to your network devices, be sure to consider the changes for inclusion in your templates. For example, you might decide to add several RIP configuration commands to your routers to improve RIP performance; be sure to make those same changes to your templates.

Q 1.6: How can I reduce network device problems through configuration management?

A: Throughout this book, I've been preaching the wonders of configuration management as a means of preventing problems—or at least of easily detecting them. Configuration management, however, is really just a form of documentation, and it's documentation that can really help avoid—or quickly determine the cause of—network device issues.

Inventory

One way to prevent network device problems is by maintaining an ongoing inventory of your network devices. This inventory should include items such as:

- Hardware inventory, software versions, module descriptions, and so forth
- Port assignments, connected media type and speed, and other logical configuration values
- Routing configuration, VLAN configuration, access lists, and other security concerns
- Any out-of-band management configuration
- Cable requirements


For example, suppose you receive from your configuration management system a notification email that informs you that a particular router has had a particular interface's media type changed. You examine the change and find that it was switched from a 10Mbps Ethernet connection to a 100Mbps connection. "No problem," you think "all the hardware is 100Mbps anyway."

Except that this particular router is connected with CAT3 cabling, which doesn't reliably support 100Mbps connections. The router starts to experience periodic failures that are difficult to pin down. What you should have done is examined that configuration change in light of the router's hardware inventory, which would specify that it was using CAT3 cables. You would know that you need to immediately switch to CAT5 or better cabling to prevent problems.

Your inventory should also contain a list of users or services that would be affected by the device's failure. When a network problem does occur, you can search through your documentation to match affected users or services, and quickly narrow the problem to the devices servicing those users or services.

Configuration Management

As I've mentioned before, configuration management of devices' running configurations is critical. You need to maintain—through whatever means you prefer—a running list of changes, a backup of the current and previous configuration files, and so forth. Most network device problems occur as a result of a configuration change; thus, being able to quickly spot the change and roll back to a known working configuration is your best weapon in the fight against network downtime.

 Interestingly, when I'm helping a new consulting client establish some management guidelines in their network, we often find that about half of their devices have differences between their startup and running configurations. That leads to lots of problems when devices are restarted!

Audit Trails

Configure your devices, if possible, to use Remote Authentication Dial-In User Service (RADIUS) or Terminal Access Controller Access Control System + (TACACS+) accounting. Configure them to synchronize their time with a Network Time Protocol (NTP) server so that accounting messages will include timestamps. Accounting makes it much easier to track the source of a problem in time. This ability is especially important if your device configurations have changed multiple times between configuration management pulls, because the accounting log might be your only indication of which configuration settings have changed, who changed them, and when the changes occurred.

Network Topology

Always, always, always have an up-to-date, accurate diagram of your entire network. Troubleshooting any kind of problem—especially with routers—is much easier when you have a diagram that shows how things *should* be working. Maintenance of your diagrams is especially important and should be part of a formal configuration management process. Although up-to-date diagrams can make troubleshooting vastly easier and more efficient, outdated diagrams can be a significant hindrance.

Topic 2: Network Configuration Management Security

Q 2.1: How does network management contribute to an overall information security plan?

A: Many companies have detailed information security plans, including physical security of information, electronic security, and more. They implement technologies such as IPSec to encrypt sensitive information and 802.1X to help restrict who can connect to the network, and they use detailed firewall configurations to help ensure that only authorized traffic crosses from the internal network to the Internet. These companies have almost no security whatsoever on their network devices—the same devices that ensure that IPSec functions, 802.1X is properly configured, the firewalls remain properly configured, and so forth. Some of these companies even have their devices' read-write SNMP community strings set to "private," making those configurations anything *but* private for anyone managing to get into the device's configuration.

Any security plan is, of course, only as good as its weakest link, and in the case of these organizations, the network devices are the weakest link. Because the devices are so poorly secured and implement core security technologies, the entire network becomes more susceptible to attack than it need be. In addition to security concerns, these companies can face major regulatory compliance requirement problems that result from the lack of security at the network device layer.

The problem most of these companies have is not a lack of understanding about security or networking technologies; they simply have poor policies regarding network management and don't understand the impact that unsecured network devices can have on their carefully thought out security plan. One reason behind their problem is that they have given high-level business policies to their network administrators. For example, a policy such as *Only authorized computers must be able to connect to the network and obtain a DHCP address* tells a network administrator that 802.1X might be called for; it doesn't, however, say anything about securing the network switches that actually implement 802.1X. These companies need to specify some additional policies regarding the security of network devices.

When working with network policies, start with the high-level business policies, which are generally driven by specific business requirements. From there, develop technological policies, which are implemented by specific technological rules. This method creates a one-to-one mapping that both business and technology professionals can understand more easily. Figure 2.1 illustrates the concept.

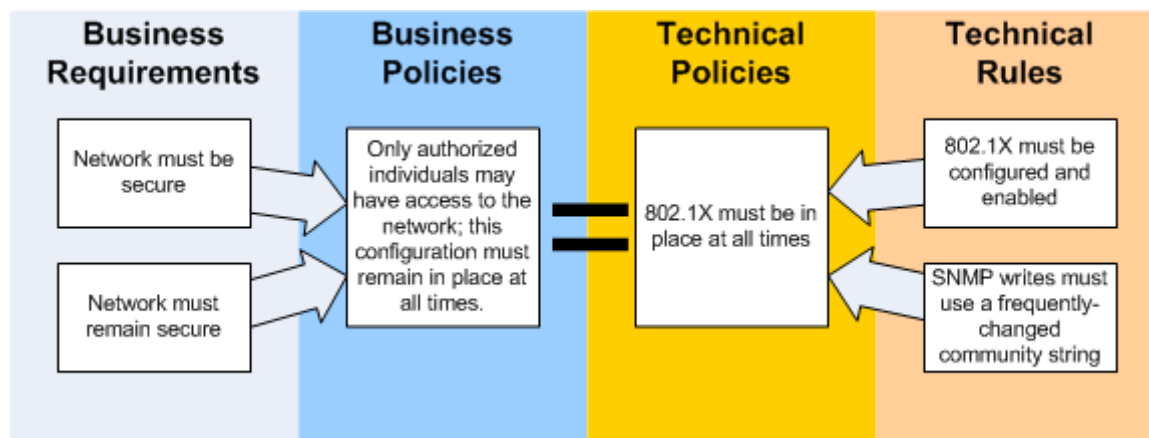


Figure 2.1: Mapping business requirements and policies to technical policies and rules.

Once you have defined policies and rules at this level, you can begin implementing them on network devices. Actually taking the time to write out your policies—and how they relate to one another—has two beneficial effects:

- You will be communicating policies and requirements more clearly and effectively
- You will be able to more easily spot any gaps in your policies and rewrite them or draft additional policies to fill the holes


Finally, a new generation of network configuration management solutions is able to manage your devices directly from these policies, rather than simply managing raw device configurations. This policy-based management is a key driver behind so-called agile business strategies such as Hewlett-Packard (HP) Adaptive Enterprise, IBM OnDemand, and Microsoft Dynamic Systems Initiative.

Q 2.2: We manage network devices by using Simple Network Management Protocol. Are there security risks?

A: You betcha. For starters, remember that Simple Network Management Protocol (SNMP) can be used to read and write (*set*) information on network devices. Reading might not seem like a security risk, but it is. Attackers can use the information gleaned from SNMP to learn more about your network's infrastructure, effectively building a complete blueprint of your network. Every movie that includes bank break-ins teaches us that a blueprint is the best way to plan your attack, so denying attackers your network blueprint is a wonderful first defense. Of course, SNMP's ability to change network devices' configuration settings can, of course, be lethal. The following list highlights tips for keeping SNMP from becoming a hazard:

- Disable SNMP entirely, if you can. Other, more secure, management protocols are available (most of them proprietary, such as Hewlett-Packard's Insight protocol). At the very least, configure your devices for read-only SNMP, and make configuration changes through other means, if possible.
- If you are not using SNMP, you definitely need to disable it in your devices. You might think that your firewall prevents Internet intruders from using SNMP to attack your devices, but don't forget about internal intruders or just plain mischievous users!

- Change your SNMP community strings often. Once a week isn't too often. Of course, changing strings on all your devices can be a tedious task, so check whether your network device vendors offer any tools to automate the process. Again, don't assume that your devices are safe from attack just because they are behind a firewall!

 *Never ever* leave your SNMP community strings set to “public,” which is often the default setting. Every attacker knows to try that first.

- The SNMP specification requires community strings to be case-sensitive, so use a mix of uppercase and lowercase letters as well as numbers. In addition, don't use cutesy community strings; use completely random ones, just as you would for an especially secure password. For example, e3N7Rft8eH8H would make an effective community string.
- Configure boundary devices, such as firewalls, to block SNMP traffic from entering or leaving your network.
- Ideally, build a separate network to carry SNMP traffic, and physically separate it from your production network. This technique will make it more difficult for hackers to get SNMP traffic to your devices. And never forget that hackers can come from within; simply blocking SNMP at the firewall isn't sufficient to protect your devices. A separate network will also help protect against SNMP Denial of Service (DoS) attacks (when an attacker fires invalid SNMP packets at devices in an attempt to bog them down and prevent them from responding to legitimate requests). A separate network is definitely an expensive proposition; however, it provides the ultimate in security for your network devices. Organizations that have especially stringent security requirements, such as banks and government-regulated entities, might find the investment worthwhile.
- Higher-end devices can often be configured to accept SNMP instructions only from a specific IP address or address range. Determine whether your devices support this capability, and if they do, use it. Set up your administrative workstations and management consoles with fixed IP addresses (either static IP addresses or Dynamic Host Configuration Protocol—DHCP—reservations) and instruct your devices to ignore SNMP instructions that come from any other IP address.
- Stay up to date on your device's operating system (OS) updates. Most network device manufacturers release regular patches and security bulletins to help make their devices as secure as possible. Many manufacturers provide electronic mailing lists to which you can subscribe; use those lists to notify you of the latest fixes. Patches should, of course, fall under a configuration management procedure to help ensure that they are properly deployed and that they create a minimal impact on production.
- Log SNMP traffic. Some devices provide an option to automatically log received SNMP requests. If they don't, you can use a network sniffer device to monitor SNMP traffic and capture any that it sees passing on your network. Even if you only run the monitoring software occasionally, it will help detect any unauthorized SNMP traffic on your network.

SNMP can be useful, if you're aware of the risks and take the necessary steps to make SNMP more secure.

A relatively new concept in change management is called *policy-based management*. This management concept generally requires sophisticated tools in order to implement effectively, but the concept is simple: Rather than configuring devices, you configure a set of rules and policies. For example, one rule might simply require devices to have a particular SNMP community string. The rule is then applied to the appropriate devices. The solution analyzes the devices' current configurations and compares them with the rule; devices that do not meet the rule are, at least, flagged for your attention. Better software solutions can automatically correct or *remediate* the problem by *applying* the rule and actually changing the device's configuration to comply. Want to change the SNMP community string on every device? Simple—change the rule. The solution will detect all devices as out-of-compliance and fix them for you (provided it has that capability and you have configured it to do so).

Q 2.3: How can configuration management improve network security?

A: “Hey, we just need these holes opened in the firewall as a test, then you can close them again.” How often have you had a similar request? So-called “temporary” changes are a common occurrence in most networks, even if they're only made to troubleshoot other problems. Unfortunately, these temporary changes often have a way of becoming permanent through neglect. Perhaps the administrator who made the change did so on Friday afternoon and forgot to undo the change the following Monday. Or maybe the change needed to be in place for a week, and everyone simply forgot to undo it. In some cases, those changes might not seem important, but they can add up: Administrator Joe makes one minor change, and Administrator Sally makes another; separately, neither change is a problem, but together they allow the world to access the company's private network.

Changes don't have to be drastic to cause a problem. For organizations dealing with legislative compliance issues (such as the Health Insurance Portability and Accountability Act—HIPAA, the Sarbanes-Oxley Act, and so forth), even a minor and technically harmless change can cause the organization to go out of compliance, resulting in major operational, legal, and financial problems.

Configuration management can help. First, a good configuration management process ensures that all changes are reviewed by some central party so that dangerous change interactions can be detected before they're made. By helping to automatically document changes, a good configuration management process can ensure that temporary changes are removed at the earliest opportunity. To provide these extra security precautions, your configuration management process must consider the following factors for *each proposed change* to a network device:

- What other pending or temporary changes might interact with the proposed change to create an insecure situation?
- When will the change be undone, if ever? Who will be responsible for undoing the change? Both the responsible party and another administrator should set reminders to both undo the change and review the affected devices' configuration files to verify the removal.
- Some changes, especially those made for testing purposes, might be very short-lived. In those cases, you might be able to use a configuration tool (either provided by your device manufacturer or a third party) to automatically restore devices' original configurations after the change is no longer needed, ensuring that an administrator doesn't forget to make the change.

A configuration management process should also include periodic device configuration audits. These audits can help spot potentially insecure configurations that weren't caught by the up-front configuration management process. With experience, you'll build a checklist of concerns that you can use during a configuration audit, such as commonly opened ports, router-configuration mistakes, and so forth. Bulletins from device vendors might call attention to potential security and configuration problems, and those bulletins can be incorporated into your device configuration audits to improve the overall security of your network.

Your configuration management process should embrace the fact that temporary changes will occur. Your process can accommodate these changes by including a change deployment schedule, listing the dates and times at which all changes will be deployed. When deploying a temporary change, immediately develop and schedule the accompanying "undo" of that change, placing the temporary change's removal right on schedule from the beginning. The idea is for your process to accept and accommodate the various changes that go on in your environment, rather than trying to stifle or prevent them.

Q 2.4: How will wireless devices change the way I secure network devices?

A: With the widespread adoption of wireless networking by many businesses, network security is more of a concern than ever. Of course, wireless networking (particularly the prevalent 802.11x standards in use by most companies) offers security features. The Wireless Encryption Protocol (WEP), for example, helps ensure that only authorized users attach to a network to begin with, regardless of their ability to authenticate to network devices (revisions to the original WEP have helped bolster the protocol's security). WEP is a *minimum requirement* if you manage your network devices over a network accessible to wireless users. Why? Without WEP, anyone can see the packets sent across a wireless network, and those packets might include the configuration passwords of your network devices—passwords that are often sent in clear text when you Telnet in to make configuration changes. WEP protects the network by allowing only authorized users to attach.

A newer protocol, Wi-Fi Protected Access (WPA), is available on most wireless devices. WPA helps to address a number of shortcomings in WEP, and is essentially as easy to deploy and use as WPA. Ideally, you should be using the newer WPA whenever possible; at a minimum, you should be using WEP. Working with no protection of wirelessly transmitted information—such as device administrative passwords—is simply asking for trouble.

However, the very idea that passwords might be transmitted in the open is unnerving and is all the more reason to establish a dedicated, *wired* network for network device management. This dedicated network can be configured with its own firewalls so that only authorized administrators can even access the network, and a hardwired network eliminates any potential for passwords being intercepted by wireless eavesdroppers. Even wireless devices, such as wireless access points (WAPs), should be managed via a wired network whenever possible. Figure 2.1 shows a model network configuration with network devices connected to a separated, dedicated, wired network (shown in green; the regular network is shown using black lines) designed for network management.

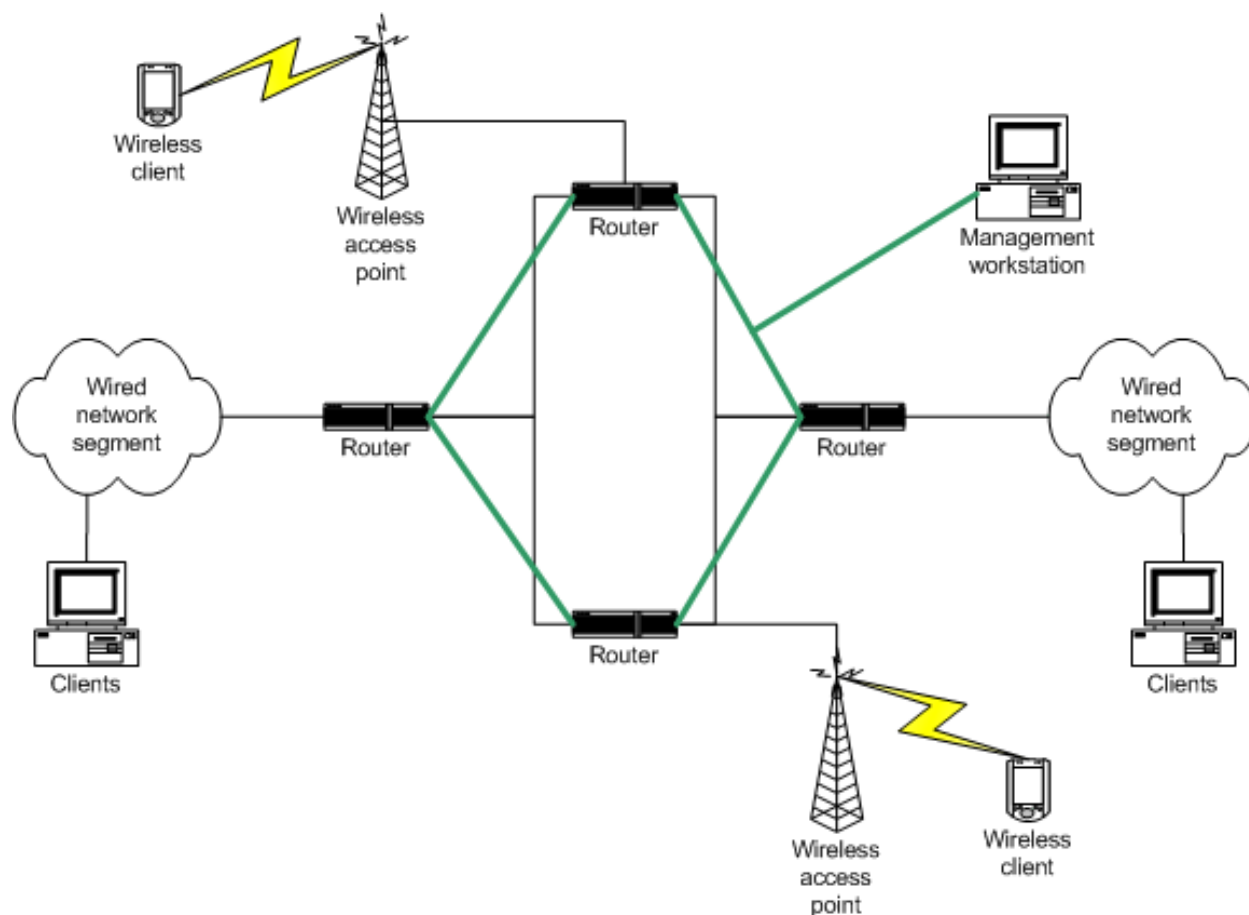


Figure 2.1: Dedicating a network to network device management.

The potential ability for someone physically outside of your company's buildings to access your network devices and make configuration changes is yet another argument for having a comprehensive, software-supported configuration management process in place. Configuration management software that can automatically archive device configuration backups will let you recover more quickly in the event that your wireless network is attacked and used to upload inaccurate configuration information to a network device.

Sure, with 128-bit (and now, 256-bit) encryption and other wireless security protocols (including 802.1X), well-chosen device configuration passwords, and other security measures, the odds of a wireless attacker ruining your routers are slim. But the most secure networks are run by rampant paranoid administrators who take every possible precaution; we would all do well to learn from their example and take no chances.

Now, for a quick reality check: Is anyone likely to deploy a dedicated wired network just for device management? Certainly—some government agencies and contractors, for example, operate under such extreme security precautions that they have already deployed their networks in this fashion. However, for the average corporation, a dedicated administrative network might be overkill, at least in terms of cost and overhead. That doesn't mean, however, that the *idea* of a dedicated wired network can't help drive some network management decisions. For example, ensuring that wireless access points are managed only over their wired connection isn't especially burdensome and helps keep sensitive information off the airwaves. Clusters of centralized equipment—Internet-connected routers, for example—can be more easily added to a dedicated administrative network thanks to their physical proximity. The idea is to simply help isolate management traffic to the greatest degree practicable in your organization.

Q 2.5: Network device security updates are issued every week. How can we ensure that all of our administrators heed them?

A: Administrators have enough to keep up with without having to memorize the periodic updates issued by device manufacturers and industry security resources. Wouldn't it be nice if some genius could just look over the shoulder of every administrator and give them a quick tap if the administrator was making a change that conflicted with a recent security bulletin?

Of course, a thorough configuration management process can help tremendously by giving each proposed change the benefit of a thorough review by more than one administrator, helping to ensure that the change is reviewed with the most recent security bulletins in mind. There is also an emerging class of software applications that can help. Ecora Software (<http://www.ecora.com>), for example, offers its Configuration Auditor software for several platforms (including Cisco routers, Windows servers, and Solaris devices). The software includes an updateable database of known security vulnerabilities, and it can analyze your devices' configurations and notify you of any vulnerability that it finds. Figure 2.2 shows an example in which the software has detected that a Cisco router's enable password is disabled, which is a security violation.

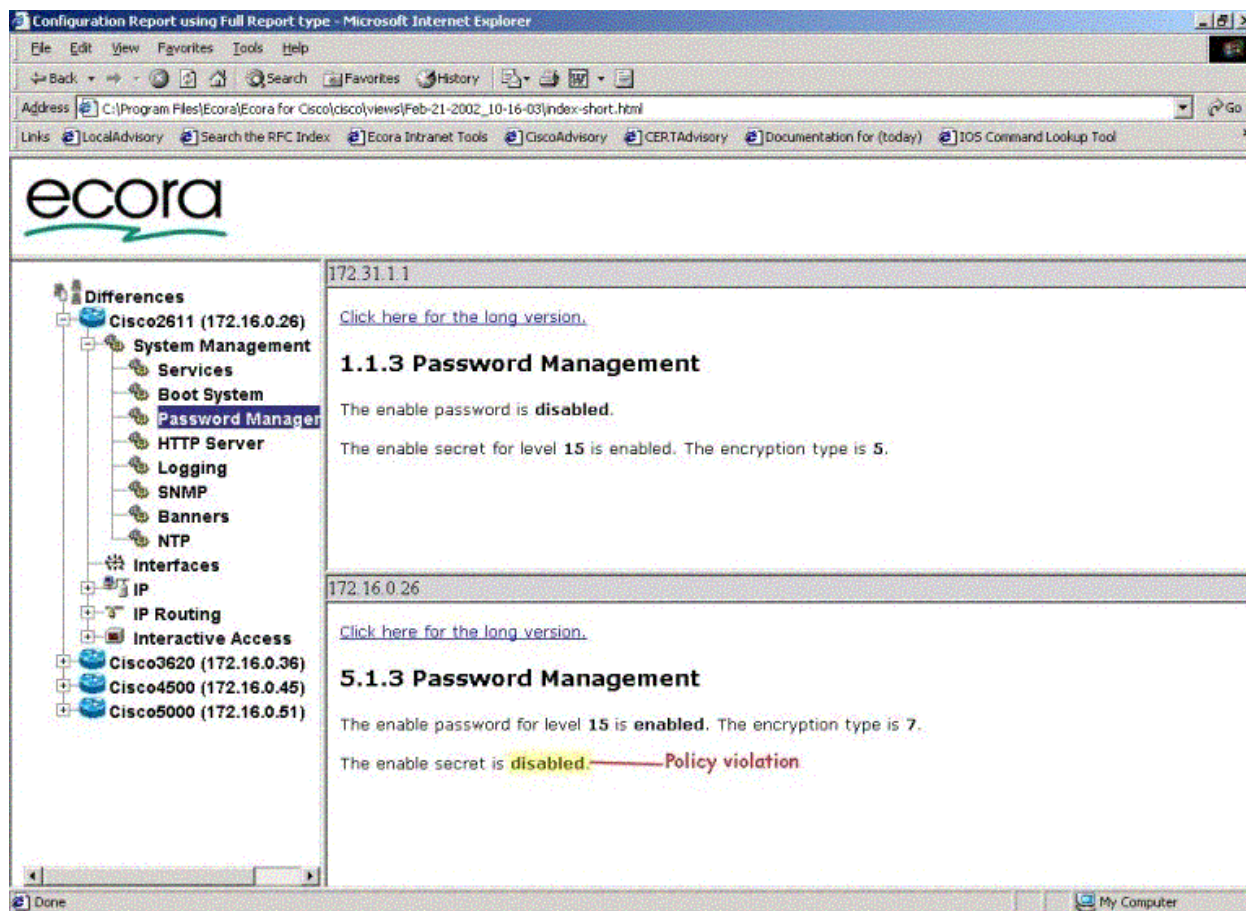


Figure 2.2: Using software to automatically audit network devices can turn up common security vulnerabilities.

The software also lets you define an authorized baseline configuration and alerts you to any changes that deviate from that baseline. The software serves as that all-knowing second set of eyes, ready to tap you on the shoulder if you make a poorly chosen change. Unfortunately, Ecora's only offering in the network device category is for Cisco devices. But other manufacturers are beginning to offer this type of configuration analysis capability for many different brands of devices.

Other configuration management solutions take a slightly different approach. AlterPoint DeviceAuthority, for example, allows you to define your own configuration policy templates, which can analyze a variety of configuration parameters within devices. By applying these templates to your devices, DeviceAuthority will compare the devices' configurations—on an ongoing basis—with your templates. Configurations that conflict with your template are either reported to you via an alert of some kind, or, in many cases, automatically corrected by DeviceAuthority with no intervention from you.

👉 Network device management software falls into two categories: agent-based software and agentless software. Ecora, along with many other software vendors, makes *agentless* software, which means the company's software runs entirely from your workstation and doesn't install any software components on the network devices that you're running. Agent-based software either installs additional software on your network devices or modifies your devices' configurations in some way to work with the management software.

Because agentless software doesn't require any changes to your network devices, it's definitely preferable. If you decide to purchase agent-based software, remember that the software installation will need to go through your configuration management process so that you can minimize the risks involved with modifying your network devices.

Finally, don't underestimate the ability of a thorough configuration management process to prevent insecure changes from making their way into your device configurations to begin with. A good review of proposed changes lets a smaller number of administrators focus on the difficult task of keeping up with bulletins and vulnerabilities. Those administrators can take the responsibility for enforcing the bulletins. If you're following best practices and making frequent backups of device configurations, you'll be able to quickly restore to a known-good configuration in the event that a change is made that impacts your security.

Q 2.6: My company considers network configuration information to be confidential. How can I ensure that this information is secure?

A: Many companies consider the configuration of the network devices to be confidential information. After all, that configuration information would give a potential attacker plenty of detailed information with which to conduct a very effective attack. Ensuring that your network device configuration files are safe from prying eyes is an excellent measure that should be mandatory in any organization.

Unfortunately, it can be tough to keep that information safe. There are several possible points at which an attacker could gain access to this information:


- While the information is on the network device. Of course, this location is where the information has to be, so you must find a way of securing your devices.
- While the information is in transit to or from the device. This point is the toughest for an attacker to use against you, as the attacker must capture the information from your internal network while the information is physically being transmitted. However unlikely, though, it is still a vulnerability worth considering.
- While the information is stored in a network device management solution. As I've discussed in previous tips, these solutions store device configuration files for backup purposes, change management purposes, and so forth. These solutions represent an excellent point of attack.

Many administrators might feel safe from attack because most of their network devices—often all but a router and a firewall or two—are behind a firewall. Firewalls can't protect your network devices from an inside job, though, and most corporate espionage and damage comes from disgruntled employees, physical intruders, and others who have access to your firewalled internal network.

Securing Information on Devices


Network devices are probably the safest place for their configuration data. Any managed network device can be configured to require username and password authentication before an administrator is allowed to access the device's configuration or command the device to download its configuration via Trivial File Transfer Protocol (TFTP) or other means. Be sure to enable this authentication requirement on your devices. And, as with any username and password combination, follow standard password best practices:

- Select a password that is at least 8 characters and is comprised of a mix of uppercase letters, lowercase letters, numbers, and symbols.
- Change passwords on a regular basis—at least every 90 days and more frequently if possible. Remember that these passwords are often transmitted in the clear, making it ridiculously simple for an attacker to pick them up off the network as they're used.
- Use different passwords for each network device. That way, one compromised password doesn't open the gate to every device on your network.

 Passwords getting tough to manage? The beauty of a device management solution is that the solution can remember your passwords for you. This functionality makes it easy to select long, complicated passwords—such as g&s8E4%g5kQe—and to change them on a regular basis. Of course, be sure to keep a written master list of passwords locked in a safe in case of emergency.

Many network devices support multiple modes of operation. So-called *privileged* modes provide the ability to change the device's configuration, and these modes are nearly always password-protected. For example, the following steps show you an example of how to set a password for the privileged mode in a switch via Telnet:

1. Use a Telnet client to connect to the device.
2. Enter the logon password for the device, if one is configured. If not, simply press Enter.
3. Type
`enable`
to enter privileged mode.
4. Enter the password for privileged mode, if one is configured. If not, simply press Enter.
5. Enter the command set
`enablepass`
to set or change the privileged mode password.
6. Enter the command set password to set the initial console logon password.

 Keep passwords different! If your network devices support multiple modes of logon, use different passwords for each. Doing so ensures that a single compromised password doesn't provide full access to the device.

Securing Information in Transit

Securing information in transit is the toughest vulnerability to protect in any environment. (However, before you lose hope, see the sidebar “Putting the Problem in Perspective.”)

Unfortunately, almost no network devices exist that transmit information in an encrypted format. Most network devices require clear-text Telnet or TFTP sessions for management purposes, and typically only firewalls and some very high-end routers offer the option to manage via virtual private network (VPN).

So how can you ensure that eavesdroppers on your network won't use your device management sessions as an opportunity to lift valuable configuration settings? About the only way is to set up a dedicated network that you use for management. Then you can control the workstations and other devices that connect to that network and ensure that only authorized management workstations and your devices are connected. Unfortunately, setting up such a dedicated network is expensive, complex, and time-consuming. So much so, in fact, that all but the most security-conscious organizations, such as certain government organizations, will feel that the time and expense are justified.

Putting the Problem in Perspective

How big a risk is it to transmit device configuration information across your internal network? The answer actually depends on how secure your network is in other areas. Although it is probably unlikely that an attacker will physically break into your office with a computer and plug into your network, there are a number of more likely scenarios that could provide attackers with access to network traffic.

For example, an attacker could send viruses to company employees via email. If only one employee activates the virus, the virus could run for days in the background of the employee's workstation, capturing information and transmitting it to the attacker across the Internet. Securing against this type of attack is no different than securing against any other type of virus—virus scanning software on firewalls, email servers, and workstations will mitigate the threat.

Also, you should consider how likely an attacker is to capture network device configuration information. After all, you probably don't manage your devices every single day. An attacker with access to your internal network is far more likely to capture other confidential information—confidential documents, internal emails, and so forth—rather than going for device configuration traffic.

As I mention in the sidebar, securing your network configuration traffic is really no different than securing the other traffic on your network. Doing so isn't easy to do directly—nobody wants to completely encrypt all internal network traffic—but it's fairly easy to do as part of an overall security plan. Here are some suggestions:

- Implement a physical security plan. Know who is in your buildings and connected to your network.
- Implement a virus protection plan. Use virus scanners at multiple levels (email servers, firewalls, workstations, servers, and so forth) to prevent viruses from becoming spies on your network.
- Implement software control. Disgruntled employees can't run network sniffers and other intrusion tools if you don't let them. Use features in operating systems (OSs) that restrict end users to authorized software packages.

Securing Information in Storage

Of all the places that network configuration information can be compromised, information that is in storage is the most likely. As I've described, network devices themselves can be made fairly secure so that they're not giving up their information to just anyone. Attackers have to work pretty hard to catch information in transit on your network, and if they've worked that hard, there is a lot of more interesting information they could catch instead. But your network device management solution represents a persistent storage location, and you might not think to secure it.


Of course, the best defense your device management solution can incorporate is an encrypted storage repository or database. Encryption makes a stolen database useless to attackers and safeguards your network device configuration information. The device management solution should also require some kind of username and password logon before providing access to the database through the solution's user interface. After all, an encrypted database is pretty pointless if anyone with a copy of the application can open it.

☞ Use file security. Of course, you should use your OS's file security features to prevent unauthorized access of any kind to your device management database files. Also, use the file system's security features to protect access to the device management application itself. If unauthorized users can't read the application file, they can't run the application and attempt to guess a legitimate logon password—providing an extra layer of protection.

Most of the major device management solutions on the market—such as AlterPoint's DeviceAuthority, ReadyRouter, and Tripwire—include database encryption features and require a logon. For example, Figure 2.3 shows DeviceAuthority's logon screen—this interface is Web-based, so DeviceAuthority protects your logon information by using a secure HTTPS connection.



Figure 2.3: DeviceAuthority uses encryption to protect your logon credentials.

 Watch out for temp files! Some lower-end device management solutions, such as freeware and shareware packages, use temporary files when obtaining device configurations. Typically, the solution will command the device to send its configuration via TFTP. That configuration is stored in a file, which the solution reads into its database. Even if the solution uses an encrypted database, it must delete that temporary TFTP file after reading the file into the database. Failure to delete the file will result in a clear-text, more easily accessible copy of the configuration information.

Be sure to evaluate device management solutions with this shortcoming in mind. Watch out for the creation of temp files and ensure that the solution you choose either doesn't use them or deletes them immediately.

If security is a concern in your environment, you should add database encryption and user logon requirements as an item on your check list as you evaluate device management solutions.


Q 2.7: Can I use TACACS+ for device authentication?

A: The latest iteration of the Terminal Access Controller Access Control System (TACACS), TACACS+, is supported on many high-end network devices, particularly those from Cisco. TACACS isn't as universal as RADIUS because TACACS was originally developed by Cisco to provide authentication, authorization, and accounting (AAA) services for dial-in users. RADIUS, however, is a much more open standard and was originally developed (despite its name) to support applications other than just dial-in AAA. TACACS+ is an excellent protocol for controlling router (or other device) access, especially in all-Cisco organizations.

TACACS is documented in Request for Comments (RFC) 1492, and the TACACS+ enhancements are available from Cisco's Web site. TACACS servers, like RADIUS servers, are available for UNIX platforms as well as for Windows. In fact, Cisco offers its Cisco Secure TACACS+ server software both for UNIX and Windows platforms.

What Does TACACS+ Do?

Like RADIUS, TACACS+ provides centralized authentication for users. Rather than using their own embedded access control lists (ACLs), network devices will ask the TACACS+ server to authenticate each user attempting to log on. TACACS+ can then provide an ACL to the device, informing the device which permissions (authorizations) the user has on that device. Finally, TACACS+ can serve as a central source for accounting, meaning it can maintain a log of actions performed by the logged on user.

 Centralized accounting can be a big help in compliance situations where you must provide audit trails for changes to devices. While TACACS+ (and RADIUS) doesn't always provide details of what was changed, you can use them in conjunction with a network configuration management solution. The solution monitors TACACS+ (or RADIUS) to see when someone has logged in or out of devices' configuration modes, using that information as a trigger to re-examine the device and detect any configuration changes.

Most references you'll see to TACACS+, such as those included in the Cisco Secure ACS documentation, assume that you're using TACACS+ to authenticate remote users, such as dial-up or virtual private network (VPN) users. Remote user authentication is the reason TACACS+ (and RADIUS) was invented, but don't think that dial-up is its only application. In this tip, I'll assume that you're locally connected to your network devices and that you want to use TACACS+ to provide centralized AAA for those devices. (For more information about AAA, see the sidebar "AAA.")

AAA

The three components provided by a TACACS+ server are authentication, authorization, and accounting. It's important that you understand the role each component plays in device management.

Authentication is the process of validating the identity of a security principal, such as a particular router administrator. Authentication often requires a username or password, although it can also involve smart cards, biometric components such as fingerprint scans, and so forth. Successful authentication doesn't imply any permissions on the device; it simply verifies that the person logging on is in fact who he or she claims to be.

Authorization adds permissions into the mix, detailing what a given user—once authenticated—is allowed to do. In Cisco devices, for example, users are assigned a privilege level and are only allowed to perform commands associated with their privilege level or lower.

Accounting provides an audit trail of log messages, detailing what an authenticated user used his or her authorizations to do. This can include a detailed list of configuration changes, device restarts, and other actions of interest. Accounting *could* provide a form of configuration management control, except that combing through a log file looking for changes, then comparing those line-by-line to an original configuration, is extremely time-consuming. However, accounting does provide a useful companion to a configuration management solution. Once the configuration management solution identifies a device configuration change, you can use the accounting log to track exactly who made the change, and when the change was made, if necessary.

Figure 2.7 shows the relationship between you, the TACACS+ server, and the network device for authentication.

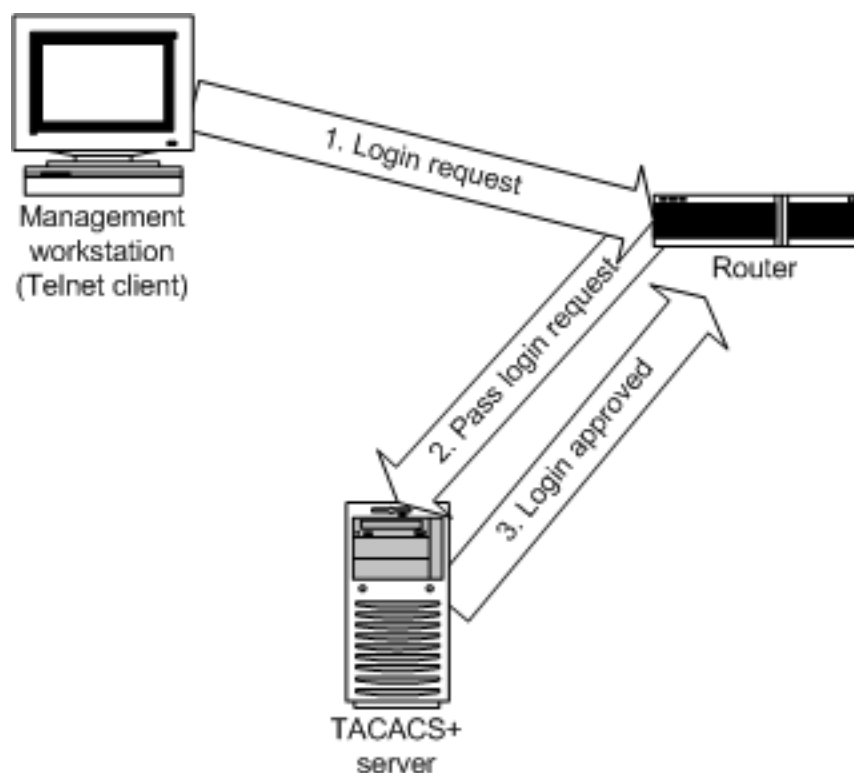



Figure 2.7: Communications between a Telnet client, Cisco router, and a TACACS+ server.

Implementing the TACACS+ Server

There are several third-party TACACS+ servers on the market, but if you're in a Cisco shop, you're likely to use Cisco Secure ACS. I'll cover the setup steps for a Windows server; UNIX server setup is very similar. First, make sure that your Windows server meets the basic requirements:

- Windows 2000 (Win2K) Server and Windows Server 2003 (Win2003) are supported with 256MB of RAM or more. You'll need about 350MB of disk space.
- You'll need either Netscape Communicator 4.76 or later, or Microsoft Internet Explorer (IE) 5.0 or 5.5. IE 6.0 is not specifically supported, although it should work fine.
- All users who will authenticate to your devices via TACACS+ must have the Windows Grant Dial-In Permission option (called Allow Access in Win2K) selected in their domain user profile.
- Cisco devices must be running IOS 11.2 or later.

 Cisco offers a freeware version of its TACACS+ server software for UNIX. You'll have to compile the software after obtaining it from <ftp://anonymous@ftp-eng.cisco.com/pub/tacacs>.

Cisco Secure installs easily and doesn't require a lot of up-front authentication (although the freeware TACACS+ server requires a fairly complex configuration file). Cisco Secure uses an HTML and Java administrative interface to set configuration options, such as users and groups, protocol options, and so forth.

Configuring Devices to Use TACACS+

Once you have the TACACS+ server up and running, you can configure your devices to use it. In the following configuration example (see Listing 2.1), *x.x.x.x* represents the IP address of the TACACS+ server. Other italicized information should be replaced with the appropriate configuration values.

```
!--- Enable mode
aaa new-model
enable password password

!--- Configure TACACS+ for various login methods
aaa authentication login linmethod tacacs+ enable
aaa authentication login vtymethod tacacs+ enable
aaa authentication login conmethod tacacs+ enable

!--- Set the TACACS+ server
tacacs-server host x.x.x.x
tacacs-server key secretkey
line con 0
  password password
  exec-timeout 0 0
  login authentication conmethod
line 1 8
  login authentication linmethod
  modem InOut
  transport input all
  rxspeed 38400
  txspeed 38400
  flowcontrol hardware
line vty 0 4
  password password
  exec-timeout 0 0
  login authentication vtymethod
```

Listing 2.1: Example TACACS+ device configuration file.

This configuration will

- Require TACACS+ authentication for Telnet (VTY), serial line, and console logons
- Set no timeout for Telnet and console logons; after testing, change this timeout to a reasonable value based on your configuration standards

Want to force users to use TACACS+ to get into enable mode? Add the following line to the previous configuration file:

```
aaa authentication enable default tacacs+ enable
```

Authorization

So far we've covered authentication but not authorization or accounting. Authorization is optional, but it makes sense to use it if you've gone through the trouble to set up TACACS+ for authentication. Keep in mind that Cisco routers have three command levels by default:

- 0—Allows basic logon and logout and provides access to higher levels
- 1—Normal level on a Telnet session
- 15—Max level (enable)

Each version of Cisco IOS assigns specific commands to each privilege level, and you can reassign commands to different levels. You can also create your own levels (2 through 14) to represent more fine-tuned access control.

 If you create your own privilege levels or reassign IOS commands to levels different than the defaults, be sure to document your custom configuration and replicate it consistently to all of your devices.

To force a device to use TACACS+ for authorization, decide which privilege levels you want to use TACACS+ with. For example

```
aaa authorization commands 15 default tacacs+ none
```


will require authorization for level 15 commands when the TACACS+ server can be reached. If the server is down, no additional authorization will be required. You can specify other options if, for example, you want the router to deny all access if the TACACS+ server is unavailable.

Accounting

Devices can be configured to send log information to TACACS+, providing an audit trail of each action performed on the router. To configure accounting in Cisco routers, use the following commands:

```
aaa accounting exec default start-stop tacacs+
aaa accounting connection default start-stop tacacs+
aaa accounting network default start-stop tacacs+
aaa accounting system default start-stop tacacs+
```

Doing so will enable accounting for commands, connections, network configuration, and system actions, both for the start and stop of each action. If your devices are set up to use Network Time Protocol (NTP) to synchronize time, then each log entry will be accompanied not only by the ID of the appropriate user, but also by the time the action was performed.

 Cisco provides a sample router and TACACS+ configuration file at http://www.cisco.com/en/US/tech/tk583/tk642/technologies_tech_note09186a00800946a3.shtml. This file can provide a useful tutorial for quickly getting TACACS+ up and running with your devices.

Topic 3: Network Configuration Management Troubleshooting

Q 3.1: We are having difficulty determining who made changes to network devices when configuration troubles occur. What can we do?

A: Auditing is the key to determining who made network device changes. If your organization is dealing with regulatory compliance requirements, auditing is absolutely necessary. However, most network devices aren't terribly well-designed when it comes to auditing. In a robust operating system (OS) such as Microsoft Windows, for example, nearly every type of user activity can be audited—for example, access to files (whether allowed or denied), access to directory objects, and so forth. Details about each activity (which file was accessed, for example) is readily available. Network devices, in contrast, provide much less granularity for their logging and offer much less detail in the logging they provide.

The first step to auditing is to implement a logging system. Most devices can work with Remote Authentication Dial-In User Service (RADIUS), Terminal Access Controller Access Control System (TACACS), Simple Network Management Protocol (SNMP), or syslog for logging purposes. As Figure 3.1 shows, you might implement RADIUS for device authentication and syslog for logging (although you could as easily use RADIUS for both).

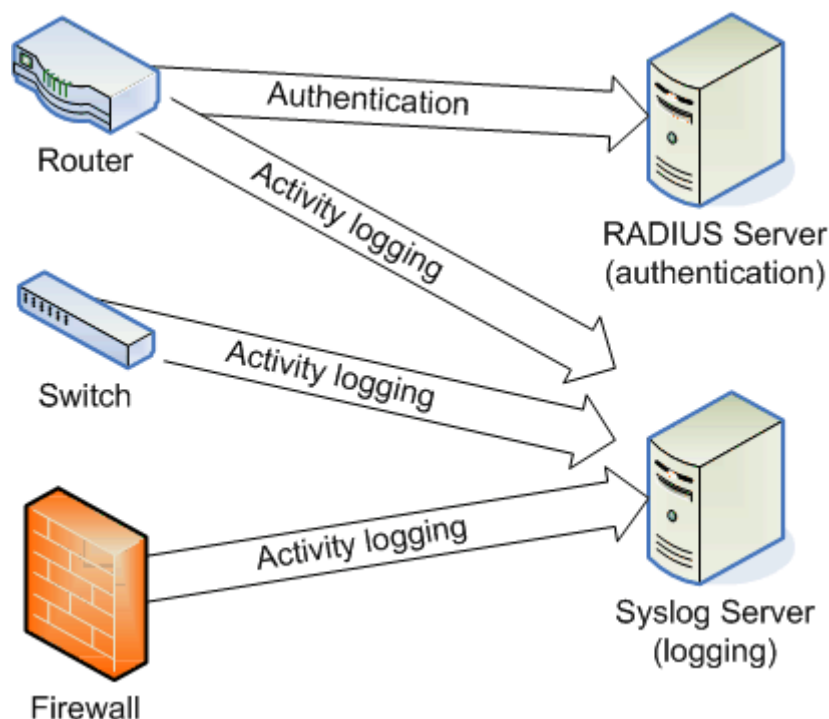


Figure 3.1: Using different technologies for logging and authentication.

Centralized authentication—as provided by RADIUS and TACACS—is crucial to obtaining more insight and control in your network devices. Because administrators are able to use a single, centralized set of credentials to manage all devices, your audit logs will be more consistent and any activity will be easier to relate to a single individual.

The problem with network device logging is that it is not detailed. For example, a network device can generate a log message whenever an administrator enters or exits the device's configuration mode. However, there is no guarantee that the administrator *did* anything in that mode, and there is no telling *what* the administrator did, if anything. In this area, network devices tend to fall short in the auditing department—they lack the detail you need to associate specific changes with a specific individual.

Third-party network configuration management tools can intercept RADIUS, TACACS, SNMP, and syslog traffic, looking for clues that someone might be configuring the devices. Some solutions even go one step further and query the device's configuration, then compare that configuration with an earlier configuration backup. This comparison allows the solution to determine that a change has occurred, what the change involved, and which administrator made the change. This information is useful for both management and troubleshooting purposes, and it fulfills the auditing requirements set down by many pieces of legislation (such as the Health Insurance Portability and Accountability Act—HIPAA, the Sarbanes-Oxley Act, and so forth). Figure 3.2 illustrates the process and shows how the device's normally insufficient logging can serve as a trigger for a more robust auditing plan.

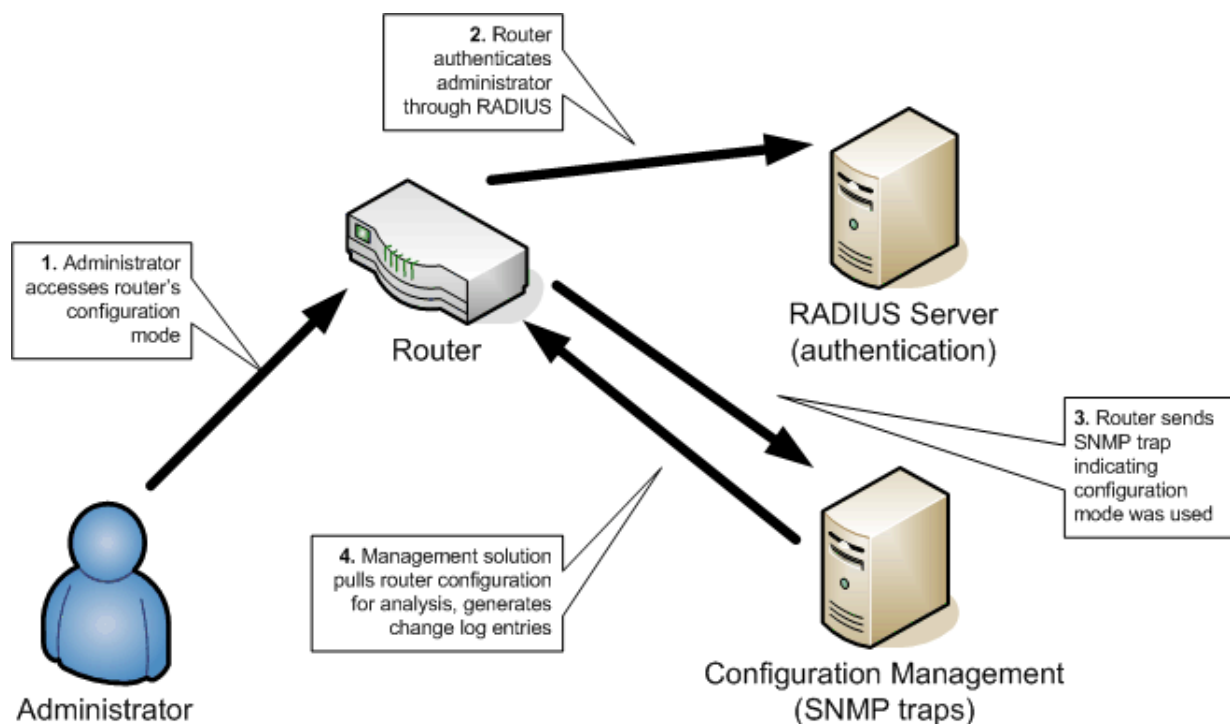


Figure 3.2: Creating more detailed logs by using a configuration management solution.

Network configuration management solutions with these capabilities often store their data in independent databases, allowing the auditing data to be secured and/or encrypted, and allowing more robust reporting. In fact, many solutions include several built-in reports to help highlight recent changes, track changes over time, provide a summary of changes to a particular device, and so forth. Thus, although devices themselves aren't capable of generating extremely detailed who-what-when-where auditing logs, when used in conjunction with a network configuration management solution, you can still get all the auditing information you need to more effectively manage and troubleshoot your network.

Q 3.2: What is the first step toward fixing a router that isn't working?

A: The first question you should ask is “What changed?” Very few network devices go belly up on their own; you'll find that it usually requires human involvement to really screw things up. Assuming that you have eliminated hardware failure as the cause of the problem, the culprit is most likely a recent change made to the device's configuration. Of course, if the hardware is at fault, you simply need to replace the hardware and restore your configuration from a backup.

Restoring from a backup—you do *have* a backup of the router's configuration, don't you?—is a good first step even if the hardware is fine. Ideally, the backup configuration will resolve the problem, and you can use a tool to compare the old and new configurations to determine the differences. This process is not exactly troubleshooting the problem, but unless you're working in a lab, your goal should be to restore the device to operation *first* and determine what caused the problem later.


☞ One change at a time, please! The idea of using a known-good backup to recover from a device failure only works if you tend to make a small number of changes at a time, let them settle to ensure that they're working properly, then immediately make a backup. If you are in the habit of making a raft of changes at once, you will have a much more difficult time tracking down the change that caused the problem.

If you plan to release changes in batches—doing so is a best practice from the ITIL framework—you can only do so if the individual changes *and* the batch have been thoroughly tested to ensure that they won't cause a problem. Obviously, testing is far preferable to the “deploy it to production and see what happens” network management technique.

If you don't actually have a recent backup, shame on you! Hopefully you have change management documentation that describes the changes that have been made to the router in recent memory. Start examining those changes to determine which ones might apply to the problem you are having. If necessary, manually undo each change, one at a time, until the problem is resolved.

Other changes might involve a device operating system (OS) upgrade or patch. In such cases, you should never make a change without understanding how you can roll back to the prior (working) version of the OS. If necessary, keep a spare router on hand in case the OS upgrade or patch kills your production unit. The goal, in any event, is to not worry so much about troubleshooting the current problem, and to simply fall back to the last configuration that worked.

Keep in mind that not all changes need to involve the router's configuration files or OS. For example, perhaps your company recently hired someone to straighten out that rat's nest of a wiring closet, and that person accidentally plugged the router into the wrong subnet when he or she put the closet back together. The wiring closet change should have been documented as a network change, and would tip you off that you need to check the router's interfaces to see what they are plugged into.

 There is no such thing as a minor change! Every single change to your network devices should go through your change management process. No change is too minor. We've all heard the story about the technician who blew dust out of a router's cooling fan. He blew hard enough to stop the fan (good lungs), causing the router to overheat and restart itself at seemingly random intervals. Had that simple maintenance action—cleaning out the router—been logged as a change, a senior administrator might have guessed that the problem was in the cooling fan, and checked that out first for a speedy resolution to the problem.

Of course, if you don't have a change management program in place or, at least, a backup of the router's configuration, you're out of easy options. You will need to start troubleshooting the problem the hard way, which might eventually involve completely reloading the router's factory configuration and rebuilding your configuration from scratch. Such drastic measures highlight the importance of both backups and a solid change management methodology.

Obviously, the easy way to always make sure you have a backup is to have a configuration management solution that does it for you. These solutions can detect—through technologies such as Syslog, Simple Network Management Protocol (SNMP), Remote Authentication Dial-In User Service (RADIUS), or Terminal Access Controller Access Control System (TACACS)—changes to your devices, and automatically download the latest configuration into a repository. You can analyze the differences between versions and roll back to any prior version any time you need to. In fact, these solutions often do a better job of answering the question “what has changed?” than a manual log would do, because they can tell you with a button click what has changed in a device's configuration. If the solution is enforcing a change management workflow for you, it can also alert you when out-of-workflow changes occur, letting you immediately focus on the potential problem.

Q 3.3: How can configuration management contribute to improved network performance?

A: Managing large networks is a complex, difficult task. Suppose you took a job at a large corporation with tens of thousands of users spread across dozens of offices. Your job, you're told, is to find out why network performance is slow. Where do you start?

You could whip out your network analysis tools and start analyzing bandwidth utilization, broadcast traffic, router load, switch bandwidth, firewall utilization, and so forth, but doing so would require tons of time and might never point to a real performance bottleneck. If you do find a bottleneck, all you could really do is start shooting in the dark, making device configuration changes in an attempt to fix the bottleneck. More often than not, that practice simply reveals additional bottlenecks, creating an unending process of network configuration changes that never really improve performance. If you're after actual results, your best starting place is gathering some basic performance trend information and analyzing the network's configuration management log.

If you can pin down a rough point in time when performance started to become less than optimal, you can start analyzing the changes that were made to the network's infrastructure devices around that time. You might discover, for example, a switch to a less-efficient routing protocol, or you might find that the routers connecting the various offices are providing packet filtering services. You might discover incorrectly configured multicast boundaries that are resulting in excess WAN traffic. Regardless, the configuration history can point to potential problems that contribute to the network's current condition. Discovering those problems empirically could take weeks or more, but finding them in the configuration history can be much, much easier.

The fact is that modern networks are becoming too large and too complex to manage as a single unit. Instead, you have to manage them in bits and pieces and in small chunks of time. For example, suppose your company is getting ready to make a whole series of network device reconfigurations designed to improve performance or simply designed to increase network addressing capacity. Before making the changes, you can take a complete set of performance measurements. By taking another set of measurements after the changes are complete, you can determine the performance impact of the change, and relate those changes to specific configuration changes from the configuration history. You're not attempting to manage the network's overall performance. Instead, you're simply trying to manage the performance *delta*, or difference between the two configurations. Some administrators refer to this process as *managing in increments*, and it's an effective way to keep on top of large, complex networks.

Of course, managing in increments is only possible if you have a solid configuration management process in place. The configuration management process provides some important capabilities:

- Configuration management provides a logical checkpoint, allowing you the opportunity to take performance measurements before and after a discreet set of changes.
- Configuration management provides a history, enabling you to compare before and after configurations and relate them to measured performance changes.
- Configuration management provides a rollback mechanism, making it easier to revert to a previous configuration if the performance of a new configuration isn't what you desired.
- Configuration management should provide an enforcement mechanism, which helps ensure that properly configured devices *remain* properly configured.

Ideally, you'll have access to software that can help gather and maintain device configuration information for historical and analytical purposes. That software might even allow you to store performance measurements so that you can save a performance baseline with each set of changes, defining a point in time at which that performance was measured and relating it to the device configuration that resulted in the performance.

Q 3.4: What are some industry best practices for troubleshooting network devices?

A: Network devices have been around a long time, and the technology industry has developed several best practices that make troubleshooting easier and often let you avoid the need to troubleshoot altogether. As author Scott M. Ballew stated in his book *Managing IP Networks with Cisco Routers* (O'Reilly and Associates) “The best way to handle network problems is to avoid them.”

Here are some additional tips I've picked up over the years:

- Create detailed documentation of your network's physical connections. One of the most common reasons for network downtime is swapped cables, and a detailed map of which wires go where can be a huge benefit during troubleshooting. Given the alternative—tugging on wires until you figure out where they go—making documentation is a great investment in time.
- As I've described in other tips, document every change you make to network devices' configurations and have backup configurations ready in case a change backfires.
- Your first troubleshooting step should often be to simply undo whatever it was you did last. Backup configuration files can make doing so very easy and will let you review the problem-causing changes at your leisure.
- Make as few changes as possible at a time; that way, if problems occur, you'll have fewer changes to sort through to find the cause. How long you wait between changes is a matter of personal taste; I like to wait at least 1 week so that my network can experience the full range of a week's workload before I certify the change as a success. Of course, in a busy network environment that uses the latest technologies, limiting your workload can be difficult or impossible, making third-party configuration management tools all the more valuable.

Experienced administrators have learned these tips through trial and error. You likely have a few other common practices you follow in your environment to keep things running smoothly.

Q 3.5: How can I determine whether a new product or a consultant makes changes to our network devices?

A: Large companies are likely to have any number of consultants and contractors running around on different projects at any given time. Some of them might have the authority to make changes to your network devices, probably with the understanding that they document any changes they make. However, there's always a change or two that gets made right before the weekend that doesn't make it into the documentation.

In addition, it's possible for new software applications to make changes to your network devices. Suppose you're evaluating a new network performance monitoring solution that needs to query information from your routers. Or perhaps you're installing an enterprise management solution that needs credentials to access your managed network devices. In these cases, the software might make minor configuration changes to your devices without your knowledge. That's not necessarily a bad thing; the changes made by these software packages are usually minor and simply make it easier for the software to do its job. But you still need to know about those changes in order to control your device change management process. So what can you do?

Unfortunately, very few network devices are designed to automatically notify an administrator when their configurations are changed. After all, only an administrator should have the credentials to make a change, so the devices quite reasonably assume that the administrator made any changes and doesn't need to be notified.

Manually Detecting Changes

Most network devices allow you to use Trivial File Transfer Protocol (TFTP) to transfer the devices' configuration files to a TFTP server (I explained how to set up a TFTP server in Tip 4.3). If you regularly dump your devices' configurations to TFTP and save the files, you have a baseline from which to check for changes to the devices' configuration. For example, suppose you downloaded a router's configuration into a file you named Router5Feb03.txt. A contractor recently finished installing a new enterprise management solution, and you want to see if any changes were made to Router5. Just follow these steps:

1. Enter

```
telnet routername
```

to Telnet to the router that you want to back up (for this example, I'll assume you're using a Cisco router; change the following commands as necessary if you're using a different device). Obviously, you could also use the router's IP address instead of a name.

2. Log on to the router.

3. Enter

```
enable
```

and provide the correct password. Doing so enters privileged mode and lets you access the router's configuration.

4. Enter

```
write network
```

then enter the IP address of your TFTP server.

5. Enter the name of the configuration file (I'll use Router5Mar03.txt for this example).

6. Press Enter to confirm the write. Ensure that the router responds with an [OK] prompt after writing the configuration.

7. Enter

```
exit
```

to log out of the router.

Now you've got two text files, one with the old configuration and one with the new configuration. You simply need to compare the two. Assuming you're running on a UNIX computer, enter the following

```
Diff -abls Router5Feb03.txt Router5Mar03.txt
```

If you're using Windows, you can use a graphical version of Diff, called CSDiff, which I mentioned first in Tip 4.3. It's available from Component Software (<http://www.componentsoftware.com>) and makes it much easier to spot changes between versions of a text file—and it's a free tool. Figure 3.3 shows how CSDiff highlights the differences between two text files.

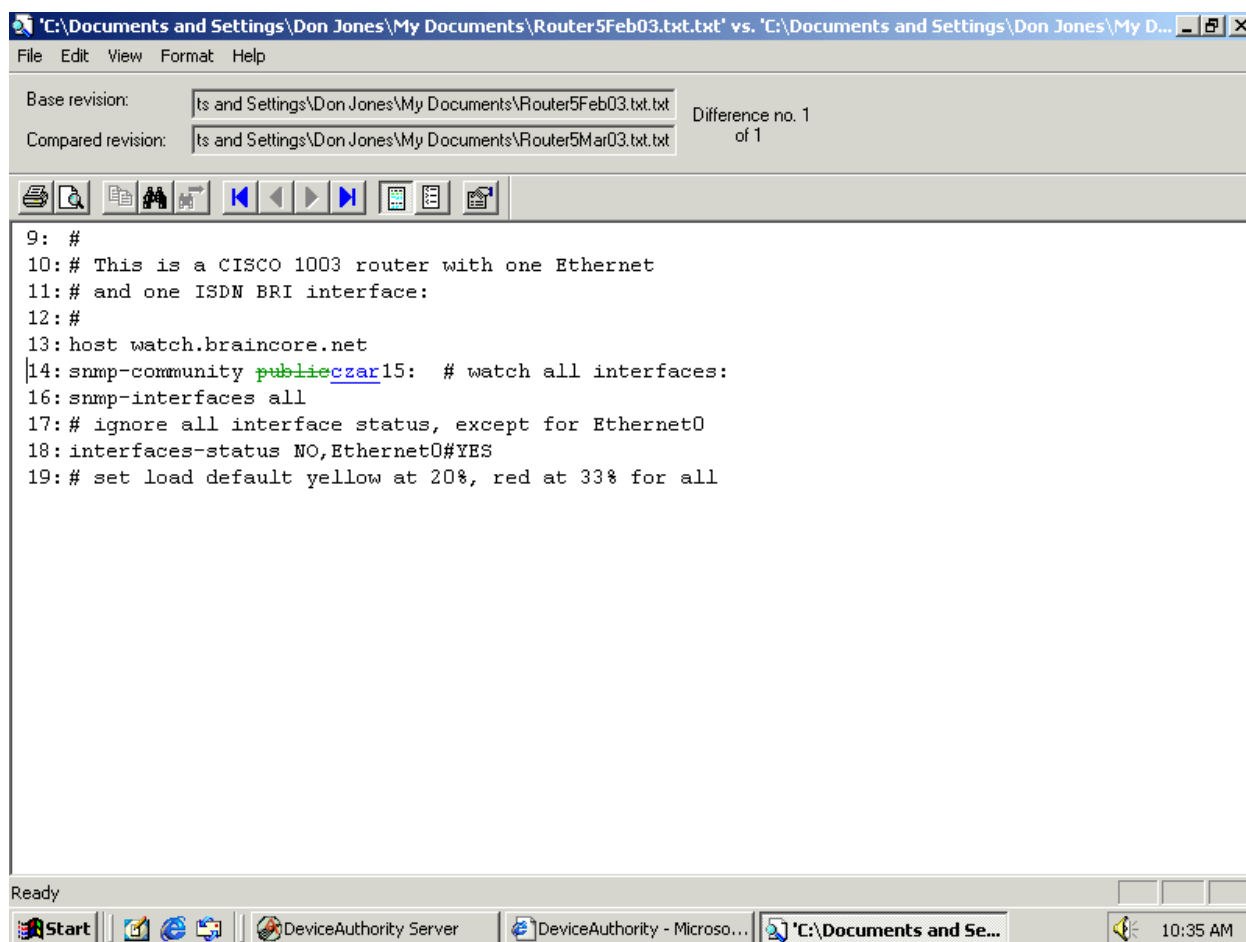



Figure 3.3: Using CSDiff to analyze the differences between two router configuration files.

Unfortunately, watching for changes manually is a lot of work. You must regularly monitor for changes on each and every network device or you could easily miss one. Because the whole point of this exercise is to pick up changes that you didn't know were being made, you need to have a change detection system that's a bit more automated.

Proactive Change Notification

Enter device change management software. Most of the big players in this field, including AlterPoint DeviceAuthority, Tripwire, and Cisco's CiscoWorks can immediately notify you via email when a network device's configuration changes. These solutions run on a server, and periodically (usually daily, although you can configure more frequent intervals) download your devices' configuration files. They then perform an internal comparison—not unlike the manual Diff I used earlier—to compare the most recent configuration with the last one downloaded. If they spot any changes, they generate an email to an administrator.

 Software management solutions often use a more sophisticated comparison than a simple Diff. Instead, they create a cryptographic checksum of each version of a configuration file. The checksum can only be the same if no changes were made to the file; if any changes occur, the checksum is different, and the software knows to investigate more closely to determine exactly which changes occurred.

Using a checksum—rather than a line-by-line comparison—allows these software packages to accurately and quickly compare configuration files that might include thousands of lines of text.

Some configuration management solutions can detect changes even more quickly by intercepting Simple Network Management Protocol (SNMP) traps, Syslog traffic, or TACACS/RADIUS accounting entries from devices. This traffic allows the solution to detect when activity occurs on the device that may be related to a change, such as an administrator placing the device into configuration mode. The solution can then download the device's configuration and perform a comparison to see what, if anything, has changed.

Ideally, your change management software should allow you to configure daily reports. That way, you'll be able to carefully review changes on a day-to-day basis rather than waiting a week or more and having to review dozens of potential changes. For example, as Figure 3.4 shows, DeviceAuthority provides a great deal of flexibility in scheduling reports. You can also configure reports to be emailed to multiple recipients. For example, I like to receive a copy of the report myself, and I have another copy sent to my Help desk manager for archival. Whenever we're conducting a process audit, a third copy is emailed to an auditor, who compares the report to our official change log to verify our compliance with our internal change management process.

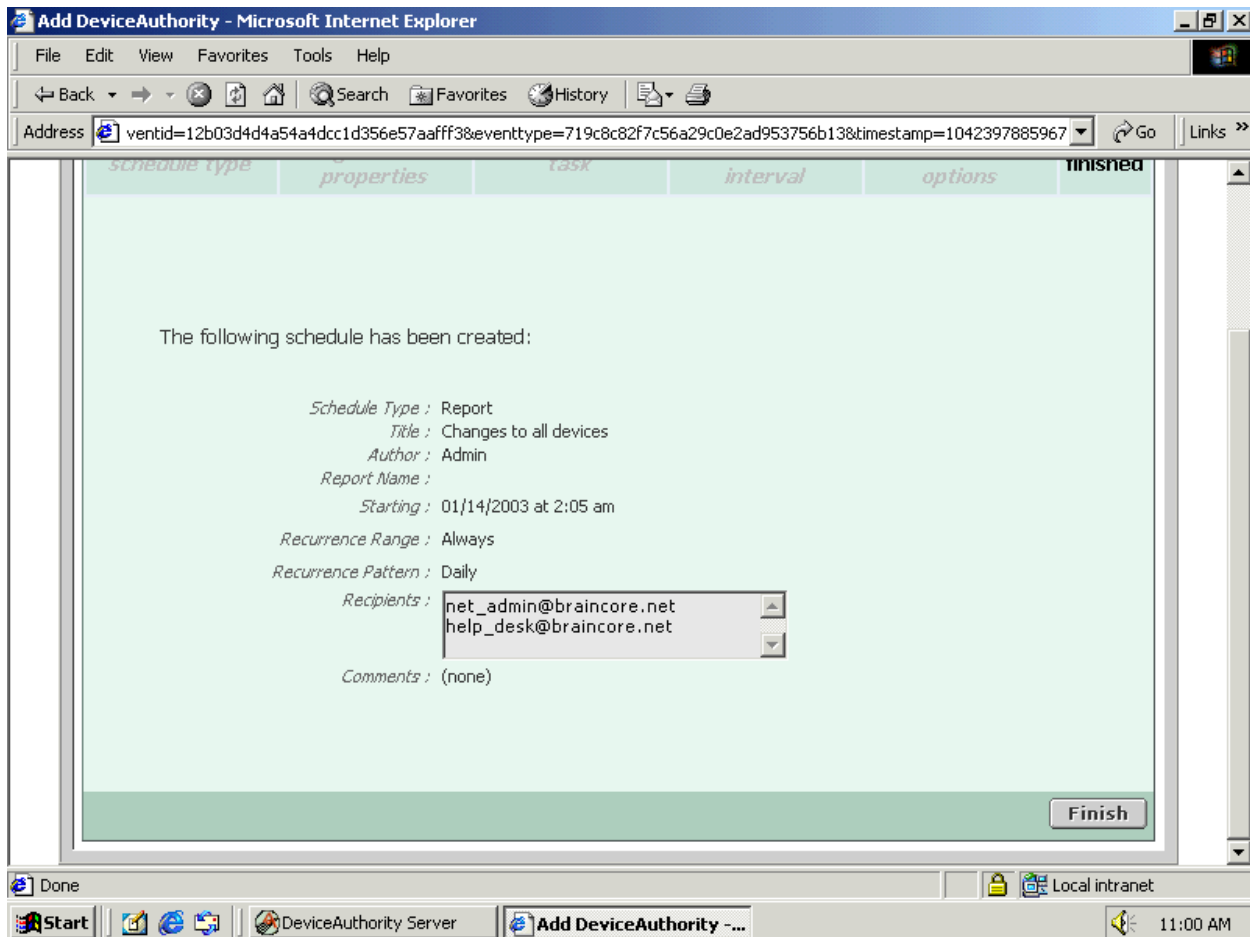


Figure 3.4: Creating a daily schedule keeps you on top of unexpected device changes and is a useful tool for auditing your change management process.

Although these change management software solutions involve additional expense and require effort to deploy, they provide a much better means of keeping tabs on your network devices than a manual process.

Automation on the Cheap

If you're unable to implement a change management software solution, you're not completely out of luck. You can still automate parts of the manual detection process and provide some basic functionality for keeping track of unexpected changes to network devices. Basically, you need to break down the process into its component steps, and come up with a means of automating each step:

- Commanding devices to dump their configuration files via TFTP. If you have any devices that don't support TFTP, you're going to have a difficult time automating a means of retrieving their configuration settings. Software solutions can pull configuration data from just about any kind of managed device, so if you have a lot of non-TFTP devices, you have one more argument for purchasing a software package.
- Comparing new and old configuration files.
- Emailing the results.


Each of these tasks can be performed on Windows- or UNIX-based computers, although the exact techniques obviously differ. Because Windows is the most common desktop OS, I'll focus on techniques for Windows. Where possible, I'll mention UNIX alternatives.

Automating the Configuration File Dump


You need to be able to script a Telnet session to automatically log onto your devices and command a TFTP dump. Unfortunately, Windows' built-in Telnet client doesn't support scripting. However, you can get a scriptable Telnet client, called Cybersource Scriptable Telnet, from <http://www.cyber.com.au/cyber/product/cybertel>. Another scriptable client, which I prefer, is the ZOC Terminal Emulator and Telnet/SSH Client available from <http://www.emtec.com>. ZOC understands a superset of the REXX scripting language, which make it a pretty powerful automation tool.

Use the scriptable Telnet client of your choice to create a batch file. For example, suppose you decide to use the ZOC client, and you create a script named GetRouter5.zrx. This REXX script logs onto a particular router and commands it to write its configuration to a TFTP server. You then create a batch file—I'll use Router5.bat as the filename—that contains the following text:

```
ZOC /RUN:SCRIPT\GetRouter5.zrx /U
```

 Note that the /U parameter places ZOC into unattended mode, forcing it to take the default settings for any prompts rather than hanging and waiting for a reply.

After the batch file is ready, use Windows' Task Scheduler to schedule the batch file to run once a day, say at around 1:00 AM. On UNIX systems, you can use CRON to set up a similar automation, using a scriptable Telnet client for UNIX. So every morning at 1:00 AM, this batch file will run and command the router to dump its configuration to your TFTP server.

 If you have multiple devices (and who doesn't?), simply create a Telnet script for each one. Include multiple lines in your batch file, with each line executing the Telnet client and one Telnet script. The batch file will then run through each device in turn, commanding them to dump their configuration to TFTP.

Automating the File Comparison

You don't want a fancy GUI to automate file comparison, so CSDiff isn't really appropriate. Instead, you want a basic command-line Diff (like the UNIX guys have) that will output differences to a file. You can get one from MKS at <http://www.mksoftware.com>. The syntax to use is:

```
diff -ir -c folder1 folder2
```

The cool part about this utility is that it can compare all of the files in a folder. So suppose you've stored your most recent configuration files in a folder named Old, and you've had your devices TFTP their current configurations to a folder named Current. You could execute the following command:

```
diff -ir -c Old Current > changed.txt
```

This command will compare each and every file in the two folders and write the results to a file named Changed.txt. The results will include each changed line, plus an additional three lines before and after the change to help you locate the change's context. If you're using this technique, it's important that your devices dump their configurations to the same filename each time. Simply create a new batch file—probably on your TFTP server, where the files are located—and schedule it to run by using Task Scheduler. If you set it to run at about 3:00 AM, that should give your first batch file time to complete.

Emailing the File Comparison Results

You're ready to email Changed.txt, the file that contains any changes found in your device configuration files. You'll need a command-line email utility, such as BySoft's Command Line E-mailer at <http://www.bysoft.se>. Create a third batch file with this command:

```
clemail -quiet -from changes@domain.com  
-to recipient@domain.com  
-subject "Report"  
-bodyfile changed.txt  
-smtpserver mail.domain.com  
-smtpport 25
```

Of course, you'll need to type all of that on a single line. Schedule the batch file to run at about 4:00 AM, after the second file finishes running, and you should have an email waiting in your mailbox when you get to work.

So there you have it, a no-cost (or low-cost, depending on how much you pay for the various utilities you'll need) solution for automatically detecting changes to network device configurations and emailing those changes to you in a daily report. It's a lot of work to set up, and you'll need to fine-tune it to work in your environment. After a while, I suspect you'll start looking at those change management solutions with a new appreciation for the work that they do!

Q 3.6: What is the best way to start troubleshooting router problems?

A: That's a tall order! Routers are complex, powerful computers in their own right, and can have several problems: routing tables can be wrong, CPU utilization can be high, network interfaces might be down, passwords can be lost, or the router might simply crash.

The best way to start, no matter what the problem, is with a step-by-step troubleshooting flowchart. Most routers' documentation includes basic troubleshooting flowcharts, which are designed to help narrow the problem as much as possible.

Most manufacturers, including Cisco, Nortel, and 3Com, offer flowcharts for their devices and provide them for download from their Web sites. For example, Cisco 7304 router troubleshooting is available at <http://www.cisco.com/cgi-bin/tsa7304/trouble.pl?tree=7304>. You start by selecting from a basic menu of problems (for example, high CPU utilization, interface issues, IOS upgrade, line card issues, password recovery, power, PXF feature support, router crash, and startup). Suppose you were to select interface issues from the main menu; the troubleshooter would walk you through a variety of questions to narrow the problem:

- Are you using an ATM interface?
- What is the output of show interfaces pos?
- What encapsulation method—such as frame relay or PPP—are you using?

At the end, the troubleshooter displays a recommended solution. This might include links to other portions of the troubleshooting tree to eliminate or confirm potential causes of the problem.

Cisco also offers these flowcharts in PDF format so that you don't need Internet access to use them. For the 7304 router, you can download PDF flowcharts by going to <http://www.cisco.com/cgi-bin/tsa7304/flows.pl?tree=7304>, then clicking Flow Charts in the left-hand menu.



Cisco offers flowcharts for most of its network devices, and you can access all of them from the support section of Cisco's Web site.

Topic 4: Configuration Change Management Techniques

Q 4.1: Which new technologies can help ease network configuration management?

A: Until very recently, network configuration management has worked pretty much the same way it has for more than a decade: Administrators deal with devices on a case-by-case basis. To be sure, new tools have made it possible to use configuration templates, for example, to help ensure that all routers are configured consistently. However, by and large, network configuration management has always been a per-device activity. One problem with this reality is that it doesn't work well in environments that have different types of devices. For example, when a Cisco expert is looking at a Cisco device's configuration, the expert knows to look for certain considerations to make sure the device is properly configured. However, when looking at a Nortel device's configuration, the expert might easily overlook settings simply because he or she is unfamiliar with the different format. Because most organizations aren't homogenous—they use at least a few different manufacturers' devices—it can be difficult to manage devices consistently.

One way new technology can help is by abstracting devices' configurations into a vendor-neutral format. Rather than requiring administrators to work directly with a device's configuration, administrators can work—to a certain degree, at least—with an abstracted, vendor-neutral configuration. Although this approach doesn't work for *every* element of a device's configuration, it is especially helpful for consistency in security and other common areas of configuration. For example, Figure 4.1 illustrates how a solution might collect information on the SNMP community string information—a key security setting—from multiple devices, and represent that information in the same fashion. This format makes it easier for an administrator—who might not be experienced with every manufacturer's configuration format—to easily check for the correct, consistent settings.

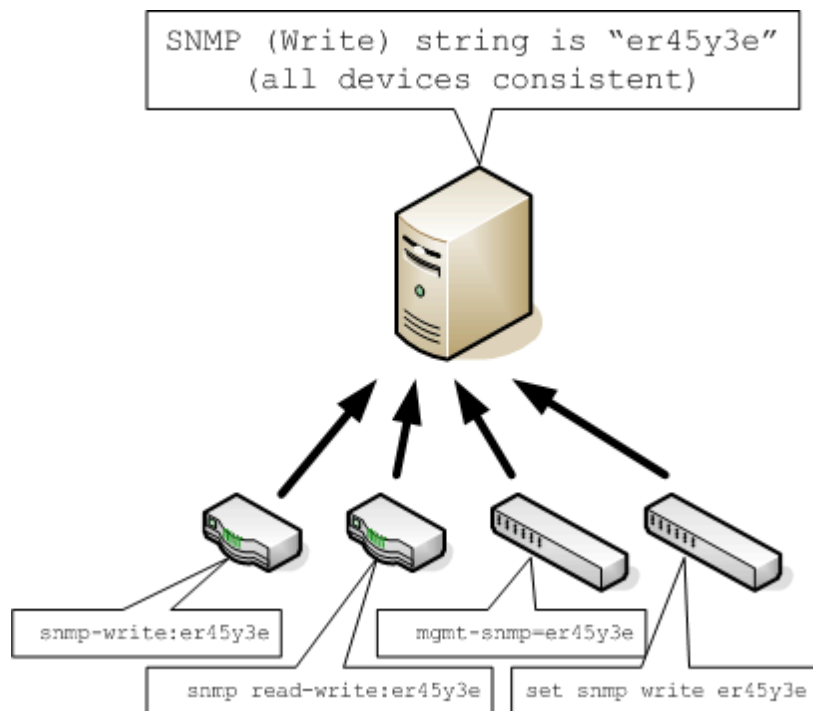


Figure 4.1: Displaying configuration information in a vendor-neutral format.

Many configuration management solutions can display details about changes made to a device; that capability is, in fact, a key part of most solutions. However, early solutions displayed the information simply as a difference between two configuration versions, as Figure 4.2 shows.

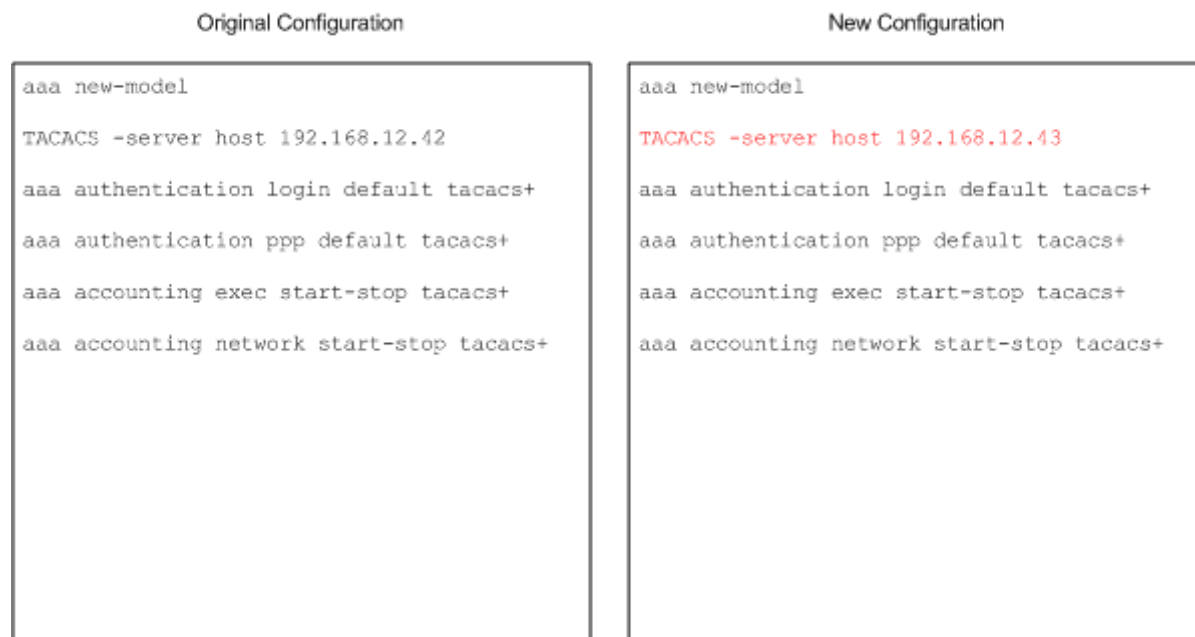



Figure 4.2: Reviewing changes made to a device's configuration.

There is no intelligence in this report; it is simply an analysis of the differences between two versions of a text file, such as you might generate with a diff command-line tool. Such reports are extremely useful, but they again require someone who is familiar with the device configuration format to determine the actual business impact of the change.

A configuration management solution capable of abstracting this information into a vendor-neutral format can generate a shorter, more business-level report of the change, such as a simple statement: *The TACACS server address was changed*. This statement has immediate meaning to any technical professional familiar with network devices, regardless of whether the professional understands this particular device's configuration format. Native-format change reports will *always* be useful, especially for troubleshooting; abstracted change reports merely provide additional uses and make the configuration management solution somewhat more flexible. Newer configuration management solutions are beginning to offer configuration abstraction and are beginning to leverage that capability in change reports and in other areas of functionality.

One key area of functionality made possible by configuration abstraction is policy-based management. Current configuration management solutions can allow you to specify higher-level configuration policies in a *vendor-neutral format*. In other words, you're managing in a slightly more business-level fashion than a purely technical fashion because you can specify configurations in an abstract, vendor-neutral format. The solution then translates those policies into the appropriate per-device configurations, and implements the configurations on your devices in their native formats.

 This technique is discussed in more detail in tip 5.1.

New technologies are also helping network configuration management systems better integrate with leading enterprise management frameworks, such as IBM Tivoli and HP OpenView. By integrating tightly with these frameworks, the new technologies are enabling enterprise administrators and managers to work with a broader array of tools from within a single interface, streamlining network administration, improving consistency, and improving several operational metrics (including the all-important metric of network uptime).

Q 4.2: How can I back up all of my network devices?

A: Sadly, not many networks are built around one vendor's solution. You could simply implement each vendor's solution, and deal with the different techniques each uses to accomplish tasks such as device configuration backup. A better alternative, however, is to implement a solution that can simplify network configuration management by handling *all* your network devices, regardless of their manufacturer. One such solution is available from AlterPoint (<http://www.alterpoint.com>). Still another solution is ReadyRouter (<http://www.readyrouter.com>), a product designed to save device configurations automatically, restore them when necessary, and track changes made to them. A third is made by Voyence (<http://www.voyence.com>), which, like the other three, can detect and save changes automatically.

If you are fortunate enough that all your network devices came from the same manufacturer, the manufacturer probably provides some kind of software to help automate device backups, which is a key part of change management. Cisco Systems (<http://www.ciscosystems.com>), for example, offers a useful piece of software called the CiscoWorks Resource Manager Essentials (RME), which provides a Web-based interface for inventory management, change auditing, device configuration, and much more. RME works with most Cisco devices, from routers to switches. RME can inventory and monitor your Cisco devices, and report any changes that occur to their configuration, and much more.

If you don't want to invest in a commercial solution, you can probably cobble together something on your own. For example, most network devices support Trivial File Transfer Protocol (TFTP) for retrieving their configuration files; you can easily write a command-line script that queries each of your devices for their configuration files and saves them to a file server. You could even schedule the script (using cron on UNIX systems and Task Scheduler on Windows systems) to run on a regular basis, ensuring that you get a weekly or even nightly backup of your device configurations.

Unfortunately, many devices don't support TFTP. For those that don't, you will need to log on to the device and manually query its configuration, perhaps writing down the results of the query or saving them in a text file for future reference. A benefit of AlterPoint's product and similar solutions is that they can automatically perform the tedious task of collecting configuration data from devices that don't support TFTP or some other bulk-transfer method. In fact, one key shopping point for a commercial configuration management solution is vendor-neutrality, meaning the solution can work with many vendors' products, and relative technology-neutrality, meaning the solution supports several methods for retrieving configuration information from devices.

Q 4.3: What is the easiest way to detect unauthorized changes in the configuration of routers and other network devices?


A: The *easiest* way is to purchase a software tool, such as those from AlterPoint and ReadyRouter. Most network device vendors, such as Cisco and Nortel, offer software that can perform the service for their devices. If you're in a mixed-vendor environment, though, and don't want to shell out for configuration management software, you can still detect unauthorized changes, although it's a manual process and anything but *easy*. First, you'll need a file server that supports the Trivial File Transfer Protocol (TFTP).



Many network devices are also capable of downloading firmware and operating system (OS) updates via TFTP. That's just another reason to add a TFTP server to your environment if your network devices support it.

The *trivial* in TFTP comes from this protocol's almost complete lack of security, so you'll need to carefully configure your TFTP server to avoid creating a security hole in your network. Most UNIX and Linux variants include a TFTP server, although most (such as Red Hat Linux) disable it by default to avoid the security issue. I recommend creating a dedicated directory for TFTP and keeping any sensitive files out of that directory. For example, create a `/tftp/` directory, ensure that it is owned by the root user, and modify your TFTP server to use that directory. Red Hat Linux, for example, requires the following line in its `/etc/inetd.conf` file:


```
Tftpd dgram udp wait root /usr/sbin/tcpd in.tftpd /tftp
```

 The `/tftp` at the end of the line specifies where the TFTP server is allowed to access files. If you leave that bit out, the TFTP server will be able to access files in any location on your server, which is definitely a huge security risk and an overall terrible idea.

On Red Hat Linux and most other OSs, you'll need to restart the Inetd daemon to reread the configuration file. Simply enter

```
killall -HUP inetd
```

to restart the service.

 Windows has TFTP, too. Although Windows 2000 (Win2K) doesn't ship with a TFTP server, you can get one fairly easily. Download.com (<http://www.download.com>) has links to several TFTP servers, including the free TFTP Server 3.0 from Ruksun Software (<http://www.ruksun.com>).

At this point, you're ready to download your device's configuration. You need to do so when you have a known-good configuration, and you need to retain that configuration for comparison purposes later. Start by creating a new file that will contain the router configuration and setting the file's configuration to permit writes. Assuming you've created a file called Routerbackup in a directory named `\tftp\`, you can use the following steps to set the permissions and download the backup:

8. Enter

```
cd /tftp
```

to change to the TFTP directory that you created.

9. Enter

```
chmod a+w Routerbackup
```

to set the correct permissions on the file.

10. Enter

```
telnet routename
```

to Telnet to the router that you want to back up. For this example, I'll assume you're using a Cisco device; change the following commands as necessary if you're using a different device.

11. Log on to the router. Enter

```
enable
```

and provide the correct password.

12. Enter

```
write network
```

and enter the IP address of the TFTP server.

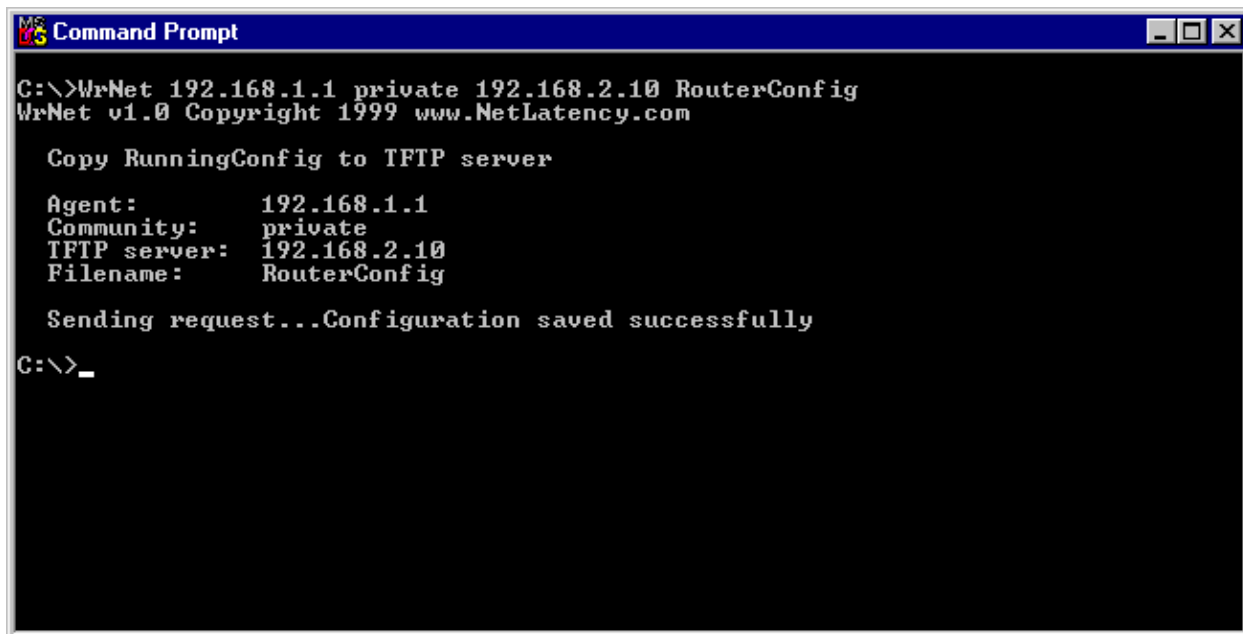
13. Enter the name of the configuration file (Routerbackup if you're following this example).**14.** Press Enter to confirm the write. Ensure that the router responds with an [OK] prompt after writing the configuration.**15.** Enter

```
exit
```

to log out of the router.

☞ If your TFTP server is running Windows, you'll either need to locate your TFTP folder on a FAT or FAT32 partition, which doesn't have any file security, or locate the folder on an NTFS partition and apply Read and Write permissions to the special Everyone group. Doing so will ensure that the TFTP service and anyone who uses it has the necessary permissions to write configuration files.

There are free tools out there, such as NetLatency's WrNet (<http://www.netlatency.com/>), that can help automate the configuration backup process. As Figure 4.3 shows, WrNet runs from the command line of a Windows server, logs onto your routers, and executes the instructions necessary to dump the router's configuration to a TFTP server. Because it's a command-line utility, WrNet can be scheduled to run automatically. Similar utilities exist for other types of network devices.



```
Command Prompt
C:\>WrNet 192.168.1.1 private 192.168.2.10 RouterConfig
WrNet v1.0 Copyright 1999 www.NetLatency.com

Copy RunningConfig to TFTP server

Agent:      192.168.1.1
Community:  private
TFTP server: 192.168.2.10
Filename:   RouterConfig


Sending request...Configuration saved successfully
C:\>_
```

Figure 4.3: Running WrNet from the command line.

So let's say you download a backup configuration file for each of your network devices (I used a router in this example, but managed switches, firewalls, and most other network devices work the same way). You'll need to periodically repeat this process if you want to detect unauthorized changes. Let's assume that your master configuration is in a file named Routerconfig and that you've downloaded the current configuration (which might include unauthorized changes) to Routercurrent. How can you easily compare the two to see what's different?


Most UNIX and Linux variants include a utility named Diff, which can be used to compare two files and display differences. For example, given the two files you've got, you could run the following command to compare them:

```
Diff -abls Routerconfig Routercurrent
```

 Diff, not Cmp! Most UNIX and Linux variants also include a utility named Cmp, which can be used to compare two files. Cmp, however, compares files character-by-character, which isn't terribly useful for text files. Diff runs line by line, showing you each line that has any changes from file to file.

Diff's options can help make the comparison easier. The options include:

- -a—Forces all files to be treated as text files.
- -b—Ignores changes to white space. Most network devices aren't sensitive to white space, so using this option will help eliminate trivial changes to the configuration that don't affect the device's operation.
- -l—Passes the output of Diff to the pr utility, which pauses after each screen full of information.
- -s—Forces Diff to report if the two files are identical rather than display no output.

 Different implementations of Diff might include other helpful parameters; be sure to check the manual for your version (generally, typing

```
man diff
```

will display the manual). If you're running Windows, Component Software offers a graphical implementation of Diff (<http://www.componentsoftware.com>) that can be a bit easier to use than a command-line version.

You're probably thinking that this whole process is a heck of a lot of work just to detect unauthorized device configuration changes. You're right; I never use this method because it's too time-consuming. However, if you're on a small (or nonexistent) budget, it might be your best shot. If you've got some money to spend, pick up a configuration comparison or configuration management tool. You can search the Internet by using "Compare router configurations" to find a fairly comprehensive list of configuration comparison utilities, including a few freeware tools that are specific to a particular vendor's devices. In an ideal world, you'll be able to acquire a complete configuration management solution from a company such as AlterPoint, Voyence, TripWire, Ecora, and so forth. These solutions can automatically download device configurations on a regular basis, compare them, and notify you of any changes.

☞ Newer configuration management solutions are integrating with devices' support of RADIUS, TACACS, Syslog, and SNMP technologies, allowing the solution to detect when a change may have been made to a device. This allows the solution to automatically pull the device's configuration right then to check for a change, giving you immediate notification of changes to device configurations.

Q 4.4: Short of buying a dedicated software application, how can I implement change management for network device configurations?

A: It's funny—server administrators have zero problems convincing the boss to buy a backup application for the company's file servers, and developers always have some kind of version-control tool to keep the company's software projects safe. Network infrastructure administrators, however, often have trouble buying even inexpensive configuration management tools, even though a router failure makes those file servers and version-control tools completely useless. Obviously, the best choice is a tool that's designed to handle network device change management. Several exist, including those from AlterPoint (<http://www.alterpoint.com>) and ReadyRouter (<http://www.readyrouter.com>). You can also use vendor-specific tools from vendors such as Cisco and Nortel Networks, and outsourced services from companies such as Greenwich Technology Partners (<http://www.greenwichtech.com>) and SilverBack Technologies (<http://www.silverbacktech.com>). The list goes on and on. However, if everything on the list is too much money for you to spend, there are some home-grown solutions you can use instead.

First, you can use a simple routine of backing up your device configurations to files by using a Trivial File Transfer Protocol (TFTP) server. If your devices don't support TFTP, you might need to use some other means of backing up their configurations. Save each configuration in a separate file, perhaps using a folder hierarchy to separate devices' files, and name files based on the date they were created. You can use a free utility such as Diff to compare configuration files, when necessary.

📖 I discussed using TFTP and Diff in Question 4.3.

A somewhat more elegant way to track versions of device configurations is to use a software developer's version-control tool. You can download Concurrent Versions System (CVS) for free from <http://www.cvshome.org> under the GNU General Public License. CVS is available for most platforms, including UNIX, Linux, and Windows, and lets you *check in* device configuration files and retrieve any prior version of a file. CVS' security ensures that only authorized administrators can add configuration files or retrieve past files. Microsoft offers a similar version-control system called Visual SourceSafe, which is bundled with Visual Studio.

☞ You've probably already got version control! If you've got software developers, chances are you have access to a version-control system, too. It's usually no hassle for version-control systems to maintain multiple projects, and one system can be dedicated to containing your device configuration files. If you have developers in your organization and *don't* yet have a version-control system, your developers can likely justify the purchase of one. CVS is free, and Visual SourceSafe is included with Microsoft's development tools (or is a small separate purchase).

A good version-control tool is a great first step in change management. Of course, you'll still need to implement policies to control the rate of change, such as restricting who can make changes on your network, when changes will be made, and so forth; having a version-control system gives you a repository for device configuration files, which can be used to restore failed devices, compare configuration versions, and so forth. Naturally, if you can afford a specific software application or configuration management service, your life will be easier. If you can't, version-control tools can provide a less painful way of managing device configuration files, a key part of change management.

Q 4.5: Our branch office routers are identical, yet users in one office say their router is slower than another office's router. What's the difference?

A: I'll assume that you've eliminated any physical or connectivity differences. After all, a lot of factors can contribute to one router being slower than another, such as:

- Number of users—A router dealing with traffic from 50 users is going to run slower than a router that handles only 10 users.
- Amount of traffic—Even the same number of users can generate unequal traffic. If one office is full of dedicated, hardworking individuals, while the other has a bunch of lazy Internet-surfing ne'er do wells, the routers in each office will see different traffic loads.
- Type of traffic—Simple HTTP traffic is a lot easier for a router to handle than streaming video, for example.
- Client computer configuration—Computers using different DNS servers, for example, might receive responses at different rates, which would affect performance.

The configuration of your network can make a huge difference too, of course. For example, Figure 4.4 shows a network in which the branch offices' connections to the Internet are far from equal. Because of the additional router hop one office has between its users and the Internet, users might be forgiven for thinking that their connection is a bit slower.

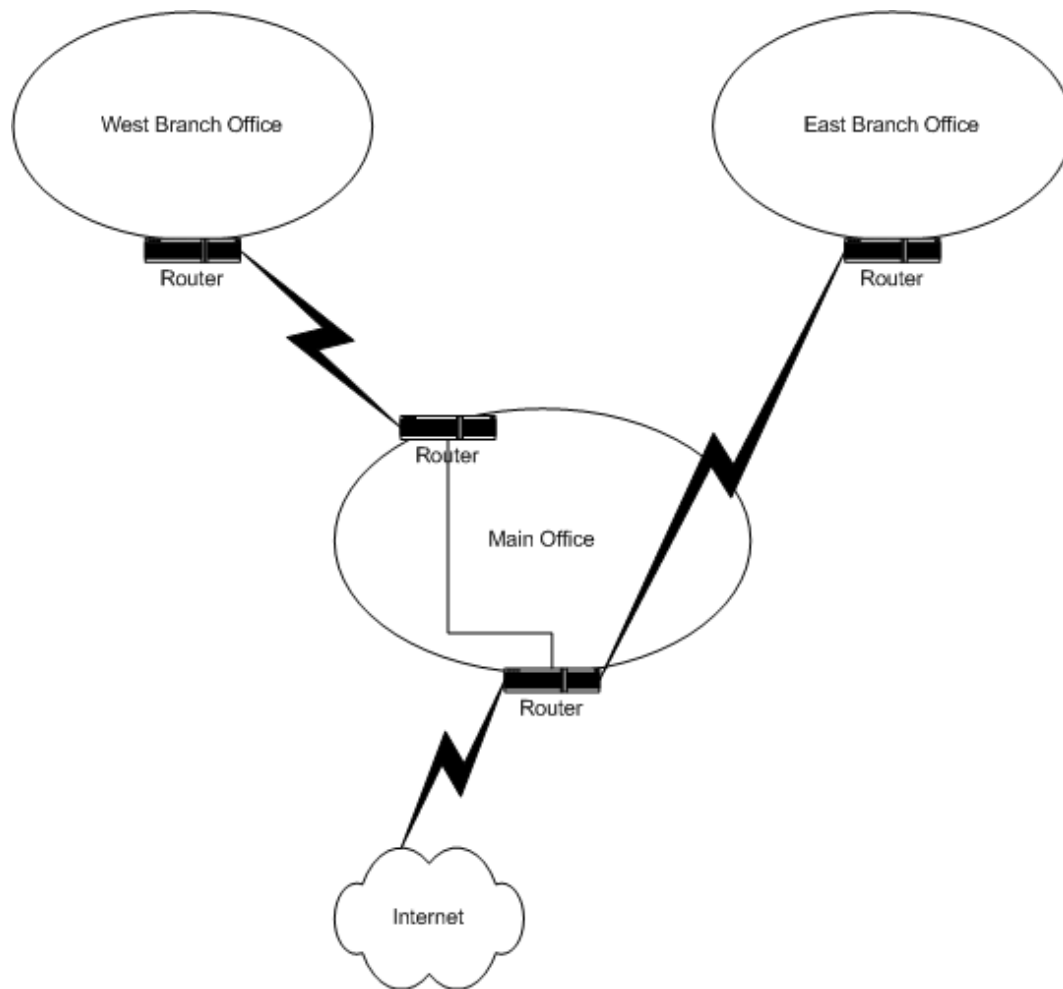


Figure 4.4: Differences in network connectivity can affect users' perception of performance.

Obviously, if the offices have different amounts of bandwidth in their WAN connections, performance would be different. However, for the sake of argument, let's assume that you've eliminated all of these possibilities. Your branch offices have the exact same number of users, the exact same network hardware, the exact same WAN bandwidth, and so forth. They still have different levels of performance, so the problem is obviously in the router's configuration.

The routers must, of course, have slightly different configurations. After all, their interfaces have different IP addresses and their routing tables are going to be a bit different to reflect the IP addresses in use in the branch offices. However, if routers are exhibiting significantly different performance, there is probably some other configuration difference that is responsible.

Unfortunately, most configuration management software packages don't make it easy to compare two devices' configurations. Thus, you will need to manually compare the two configuration files. Assuming that your routers support TFTP, you can start by having them write their current configuration to a file on a TFTP server (I provided a how-to for TFTP in Tip 4.3). Once you've got the two, use a graphical file comparison tool to compare the two files. Figure 4.5 shows sample comparison results.

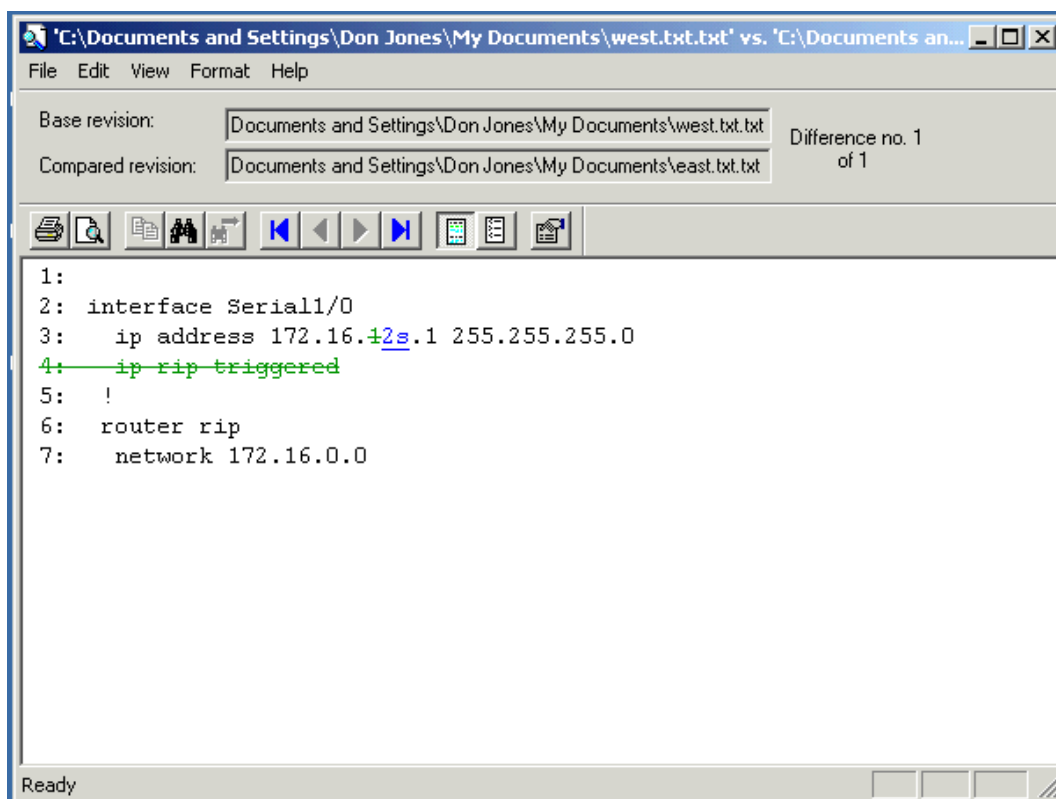


Figure 4.5: Sample comparison between two router configurations.

Notice that the comparison isn't listing the entire file; it's simply listing the lines that have been changed, and a few lines before and after to provide context. In this case, the routers' IP addresses are flagged, which is expected. However, one router is configured to have IP rip triggered, while the other router isn't. That particular setting is a Cisco IOS command that improves efficiency of the RIP protocol over WAN links. Depending upon the exact circumstances and network configuration, the router without this configuration option could certainly be operating less efficiently. You can proactively prevent this sort of thing from happening with high-end network configuration management software that lets you specify business rules for device configuration changes. Such functionality helps ensure that devices are configured uniformly across your enterprise.

Q 4.6: How can I ensure that all of the devices in my enterprise are consistently configured?


A: The first step to ensuring consistent configurations for the devices in your enterprise is to make sure that you have a consistent target configuration. In other words, you need to ensure that you've created and documented configuration standards that can be used to configure devices. Once standards are in place, you can worry about whether your devices conform to those standards.

Creating Standards

There are several areas in which you can define configuration standards for network devices, including:

- Version control
- IP addressing
- Naming
- Operational configuration

In the next few sections, I'll offer some suggestions for creating standards for each of these areas.

 If you're dealing with compliance issues, such as Sarbanes-Oxley, having a standard is the first step to becoming compliant. The first thing any auditor will want to see is your standard, telling them how devices of various types and in various roles should be configured.

Standardizing Versions

Network devices are basically single-function computers; as such, they require operating systems (OSs) to run. (Most Cisco devices, for example, run the Cisco IOS operating system.) Like any OS, network devices' software is available in different versions, and new versions are periodically released to correct bugs and add new features. Ideally, you would keep all of your network devices on the same version of their OS. Doing so reduces support costs because each device will function identically.

However, different types of devices often require different OS levels. I recommend grouping devices by basic function and model line. For example, group all your Cisco 2500 series routers in one group and your Foundry switches in another. Within each group, standardize on a specific version of the devices' OS. Document this decision, and when the time comes to upgrade to a new version, do so for the entire group of devices.

Standardizing Addressing

Decide where network devices fit into your network's IP addressing scheme. For example, many organizations will give routers the first few addresses available in the network's address space, such as 192.168.1.1 or 192.168.2.1. Set aside a block of addresses for other managed devices, as well. For example, you might decide that the last 50 or so addresses of each class C range will be reserved for managed devices, meaning they would start at something like 192.168.1.200. The following list provides an additional suggestion:

- Reserve the .1 and .2 addresses for routers, assuming that each subnet will have no more than two router interfaces.
- Reserve .3 for standby routers using the Hot Standby Router Protocol (HSRP). You can use .4 if you need an additional standby on a different address.
- Use .5 through .9 for switches. Most subnets won't need more than five switches total; if you need more than that, either migrate to switches that have more ports (thus allowing you to use fewer total switches) or consider creating new subnets.
- Use .10 through .15 for statically addressed devices such as print servers, file servers, and so forth. Alternatively, you can assign IP addresses to these devices by using Dynamic Host Configuration Protocol (DHCP) reservations, ensuring that they get the same "dynamic" address every time.
- Use .16 and higher for dynamic addressing.

Of course, if you're not using a basic class C address range for your subnets, then you'll need to adjust these suggested addresses accordingly.

Standardized addressing becomes especially important in IPv6. Under IPv6, every network interface—whether from a client computer or a network device such as a router—actually has multiple IPv6 addresses, including:

- A local link address—This address isn't routable and can only be used on the local subnet. IPv6 configures this address automatically in a fashion similar to IPv4 Automatic Private IP Addressing (APIPA).
- A site link address—This address is routable within an intranet but not across the Internet. Its closest IPv4 equivalent are Request for Comments (RFC) 1918 address ranges, such as 192.168.0.0/16, 172.16.0.0/12, and 10.0.0.0/8. You'll need to come up with an addressing scheme for site link addresses, perhaps using a specific range for routers, another for switches, and another for DHCPv6 assignment to client computers.
- A global address—This address is routable and unique across the entire Internet. Most internal devices won't use one of these, as internal devices will typically rely on the IPv6 version of network address translation (NAT) to access the Internet.

Other IPv6 addresses, such as multicast and broadcast addresses, are managed more or less dynamically and don't need much in the way of prior planning on your part.

☞ IPv6 is coming! Microsoft released its first production IPv6 stack in Windows Server 2003; Cisco has provided growing IPv6 support for years now. Most other network devices also support IPv6 in their latest OS versions. The bottom line is that IPv6 is coming, and it will need to be implemented first on network devices. Take the time now to come up with an IPv6 configuration strategy! When the time comes to use it, you'll have a preplanned, consistent scheme in place and ready to go.

Standardizing Naming

Create a standardized naming convention for your devices' network interfaces. For example, most routers will have one interface for each subnet they route to, and might have a management interface as well. So for a router named Router4 that is connected to the 192.168.12.0/24 subnet, is connected to the 192.168.13.0/24 subnet, and has a management interface on the 192.168.13.0/24 subnet, you might configure the following names:

- 192-168-12-0-router4.mycompany.pri
- 192-168-13-0-router4.mycompany.pri
- mgmt-router4.mycompany.pri

Names such as these will be more useful during a `tracert` session, helping identify exactly which interfaces are being used. You'll also benefit from the standardization of the management interface's name, allowing you to easily connect to `mgmt-routerx.mycompany.pri` in order to manage any given router.

Standardizing Configurations

Different classes of devices will obviously have different configuration requirements, but you need to take the time to define whichever ones are appropriate for each device class. Considerations should include:

- Media type
- Protocol configurations
- Routing protocols
- Access control
- RADIUS/TACACS configuration
- Simple Network Management Protocol (SNMP) configuration

Some higher-end network configuration management solutions include functionality for storing standardized configurations. These solutions can be used to deploy the template configuration to new devices, then to customize the configuration of each device as appropriate for its role on the network. Most device manufacturers also offer solutions to help manage standardized configurations.

Newer network configuration management solutions can even help enforce your standards. For example, configuration items such as RADIUS/TACACS, SNMP, and so forth are fairly static in most organizations. The right solution can monitor these settings for changes, and reconfigure devices as necessary to ensure that your standard remains in place. This real-time enforcement is better than mere auditing because it doesn't allow incorrect configurations to remain in place for any length of time.

Creating Configuration Templates

Once you have your standards documented, create configuration templates. For example, a router's configuration template might look something like Table 4.1.

Configuration	Standard
Interface 1 address	x.y.z.1
Interface 2 address	x.y.z.2
Interface 1 name	x-y-z-0-routera.mycompany.pri
Interface 2 name	x-y-z-0-routera.mycompany.pri
Interface 1 media	10/100 Ethernet
Interface 2 media	10/100 Ethernet
SNMP community string	C0m%paN4y
Routing protocol	RIPv2
Authentication	via TACACS; see config document
Memory	64MB
Slot 1	10/100 Ethernet
Slot 2	10/100 Ethernet
Slot 3	Empty
Slot 4	Extended management card
Slot 5	Empty
Out of band management	Via serial
Power	110VAC 60Hz Max 2A
Environmental	19" rack mountable requires 2-post rack mounting.


Table 4.1: Example router configuration template.

Notice that this template addresses not only software configuration for the device, but also its basic hardware configuration. To reduce troubleshooting complexity, all devices within a class (router, switch, and so on) should have hardware that is as identical to the others in that class as possible. These configuration templates can be used to create the basic configuration for every device added to your network, and will serve as the basis for compliance assurance.

Ensuring Adherence to Standards

Ensuring adherence to standards can be tricky. If you're not using any kind of configuration management software in your environment, it can be almost impossible (other than a time-consuming manual review of each device's configuration).

One of the only software solutions that currently supports compliance management is Ecora's Auditor. This tool can scan groups of network devices and compare their running configurations against a predefined template, then report on any differences it discovers. This functionality isn't actually configuration management, but rather compliance assurance.

 Compliance assurance may be the law! Certain industries, such as healthcare and public accounting, are required by law to meet standards for information processing. Healthcare is regulated by Health Insurance Portability and Accountability Act (HIPAA), and financial services are regulated by the Gramm-Leach-Bliley practice standards. In many instances, your network devices' configurations—particularly with regard to security—might be affected by these regulations. Developing a standardized template that complies with the regulations' requirements is easy enough, but using a software package to enforce your template is often a must.

Topic 5: Selecting and Deploying a Network Configuration Management Solution

Q 5.1: How do network configuration management solutions support policy-based management?

A: Policy-based management is a new concept in information technology management that has been gaining traction in the area of operating systems (OS) management. It is relatively new in the realm of network device management, but can actually be more effective there because network devices typically have more accessible and less complicated configuration repositories than the average network OS.

Traditional management has typically involved the creation of policies governing how the network will be configured; technical professionals then translate those policies into specific configuration settings, and implement them on their devices. One significant problem with this approach is that it is difficult to quickly reconfigure the enterprise because devices must be individually reconfigured to meet changing needs. Configuration templates are often used to help mitigate the pain of individual device reconfiguration and to improve consistency; however, the fact remains that the devices need to be individually configured. Another problem is that per-device management doesn't accommodate *configuration drift*.

Configuration drift in network devices occurs primarily when changes are made outside of your normal configuration process. As Figure 5.1 shows, you start with a base configuration that meets your business policy requirements (including any compliance requirements). Out-of-process changes are made over time, leaving you *thinking* you're configured one way, but in reality, are configured differently.

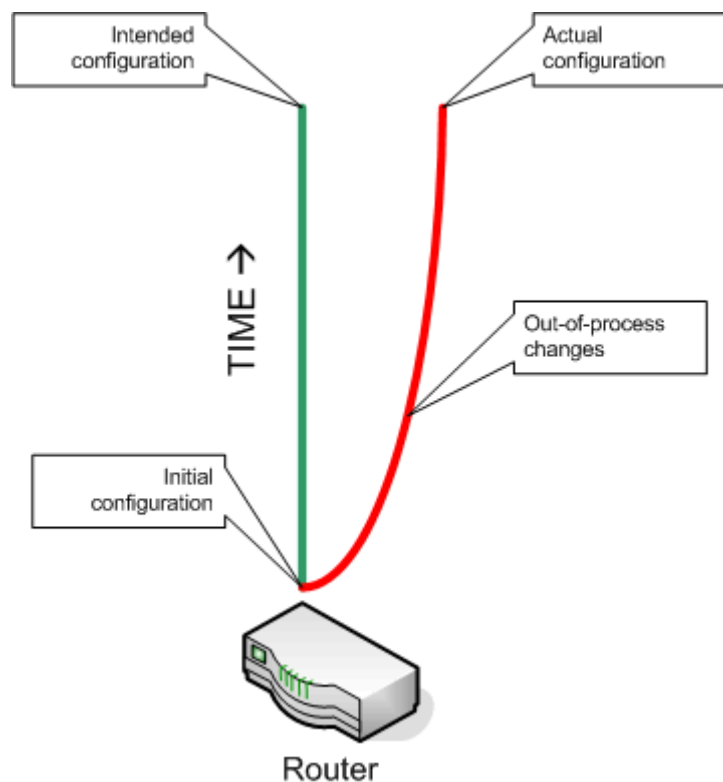


Figure 5.1: Configuration drift results from out-of-process changes.

The problem is simply that you're creating configurations to match policies, then trying to manually enforce the configurations. Policy-based management takes a somewhat different approach: You define technical-level policies that meet your business requirements, then a solution *enforces* those policies (or, at the very least, informs you when your devices aren't meeting your policies).

This difference in management style is subtle, but the ramifications are significant. For example, by establishing policies for what your device configurations should be like, you no longer have configuration drift. When configurations drift from the policy standard, you're notified immediately. In some instances, the configuration management solution might even be able to *remediate* the problem, reapplying your policy-compliant configuration in place of the non-compliant one.

Enabling an agile enterprise is also a key benefit of policy-based management. Want to change the Remote Authentication Dial-In User Service (RADIUS) server used by all your network devices? Simple: Change your policy. Your management solution will quickly determine that all of your devices are now non-compliant (with the new policy, that is), and can remediate them, reconfiguring them automatically to meet the policy. Your enterprise can thus be more adaptable to change, because you only need to redefine your policies. Figure 5.2 illustrates policy-based management.

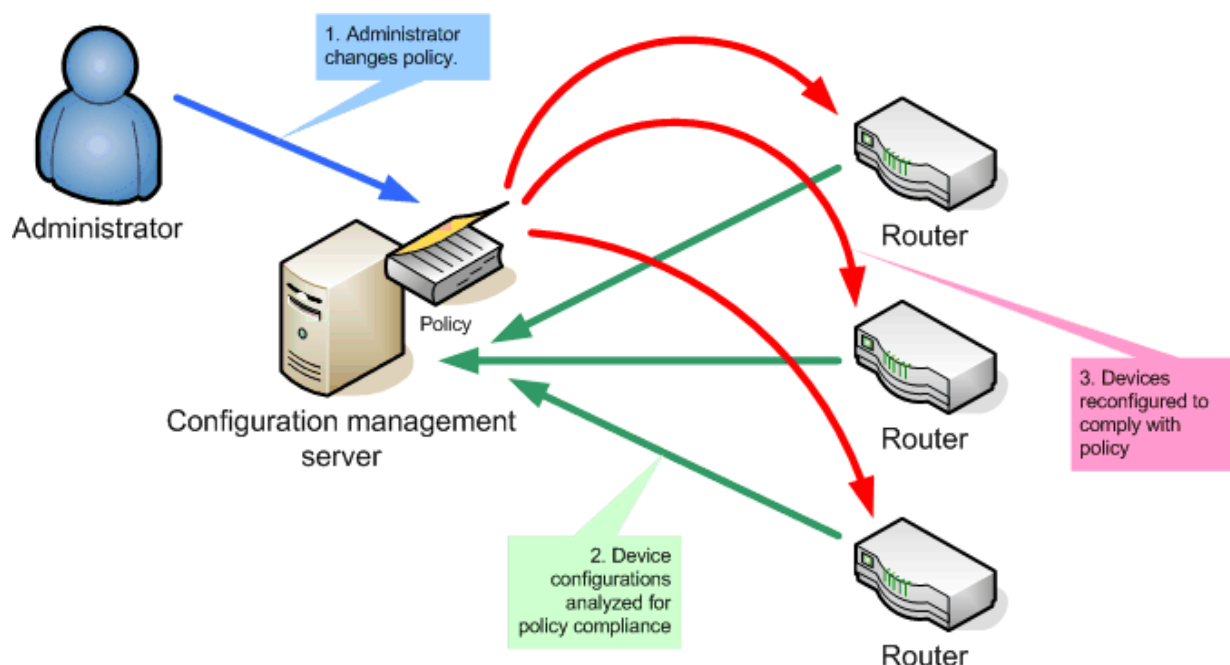



Figure 5.2: Managing devices through policies rather than direct configuration.

Policies are more complex and flexible than mere configuration templates. Generally, policy-based management combines with a layer of configuration abstraction, allowing policies themselves to be written in a vendor-neutral format; templates, in contrast, are typically vendor-specific.

In addition, multiple policies can be layered. For example, one policy might focus entirely on Simple Network Management Protocol (SNMP) community strings, while another might focus on RADIUS configuration. Policies can be applied to devices as-needed; routers might be subject to a policy regarding multicast boundaries, while switches might be subject to a different policy that deals with 802.1X configuration. Both might be subject to common policies such as SNMP configuration.

 If you're familiar with how Microsoft Active Directory (AD) employs Group Policy to configure desktop computers, then you're familiar with policy-based management. An entire set of policies can be applied to groups of managed units (such as to desktop computers or, in the case of network management, to network devices); each policy states the desired configuration, then an enforcement mechanism ensures that the policy is met.

Today, most network administrators are a bit leery of letting an automated solution make automatic changes to their network devices. After all, the devices *are* the network; a misconfiguration can be costly. For now, allowing lower-risk policies to be automatically remediated is a good middle ground. For example, enforcing SNMP configurations won't break the network and can provide a significant security advantage. At the very least, however, policy-based management provides better alerts and notifications when policies *aren't* being met. Even if the management solution isn't automatically fixing things for you, it's at least letting you know—quickly—that something's wrong, allowing you to fix it yourself.

Q 5.2: All of our equipment is from one vendor. Why not use a vendor-supplied device management solution?

A: Some network device vendors, most notably Cisco, provide great device management software. CiscoWorks has earned a reputation for being feature-laden, fairly easy to use, and inexpensive. It is, of course, limited to managing Cisco devices. But what if you're in an all-Cisco environment? Why would you select a third-party solution?

You might not. However, take a good, long look at your environment to see if you *really* are an all-Cisco (or whatever) shop. I've worked with a number of companies who've "standardized" on Cisco, or 3Com, or Nortel, and yet still have a surprising number of devices from other manufacturers. Here are some oft-overlooked devices:

- Firewalls are often overlooked. Although most device manufacturers offer firewall solutions, it's unusual for companies to select a firewall solution based on brand rather than features.
- Are your switches the same brand as your routers? Most enterprises agree that standardizing on Cisco routers is a great idea, for example, but also agree that Foundry switches are the best-of-breed.
- Have you considered print servers? Many of these devices are Simple Network Management Protocol (SNMP)-enabled and manageable from many third-party device management solutions. They're almost never included, however, in software provided by router vendors.


- Do you use load balancing solutions for Web sites or other scalable applications? If so, there's a good chance they're an independent brand because not every router manufacturer produces efficient load balancers.
- Do you have other "black box" devices, such as email servers, Web servers, or application servers that support SNMP or other network management? Basic support for these devices, including the ability to download their configuration files for archival and change management purposes, is improving in third-party device management packages.
- Network-Attached Storage (NAS) devices are often manufactured by vendors other than typical network device vendors, yet offer SNMP management and complex configurations that could benefit from third-party management software.

Third-party solutions such as AlterPoint DeviceAuthority, Tripwire, and R10 boast impressive lists of supported hardware vendors and devices. DeviceAuthority's supported device list contains devices from vendors such as 3Com, Avaya, Cisco, Dell, Enterasys, Extreme, Foundry Networks, Hewlett-Packard, Lucent, Motorola, NetScreen, Nortel, and more. Tripwire for Devices, for example, supports Cisco, Nokia, Hewlett-Packard, Foundry Networks, Extreme, NetScreen, Nortel, and others. Most of these solutions also provide an extensibility model, allowing their manufacturers to add device support in between product revisions.

Q 5.3: We're preparing to roll out a device management solution. However, we have hundreds of devices. What's the best way to proceed?

A: One step at a time. The following list provides basic tips for a successful rollout:

- Start small by placing a few devices under configuration management first to see how things go. Most solutions offer an auto-discovery module that can quickly add all of your other devices later, when you're ready to proceed.

 Start small, pay small. Ideally, your configuration management solution should be licensed on a per-device basis, with the ability to add any number of additional device licenses at any time. That way, you can pay for just a dozen or so devices to start with, then add more as you gradually bring your entire inventory under change control.

- Start by adding a representative device from each manufacturer or device class, and review any problems the configuration management solution has. For example, add a Cisco switch, Cisco router, Nortel switch, and 3Com hub to give the solution a broad range of devices to try out.
- For especially large environments, skip the solutions' auto-discovery feature. Particularly across WAN links, these features can generate a noticeable amount of Simple Network Management Protocol (SNMP) traffic, which might hinder other network operations. Instead, add devices to the management solution manually.

- Consider implementing solutions on a per-location basis or in some other modularized fashion. Most solutions are licensed by the number of devices you're managing, so managing each company office from a separate database won't usually cost any more than using a single database. However, if you have highly centralized management of your devices, it will make the most sense to use a single, centralized configuration management database.
- Don't turn on automatic notification features until you're certain the management solution is working properly and configured correctly. In one implementation, I used auto-discovery to add my network's devices. Unfortunately, it also added the devices from a lab network that was connected to the main network at the time. As the folks in the lab played with their device configurations, I got a ream of change notification emails.
- Decide how you're going to set up the solution to poll configurations from your devices. For example, you might have configurations pulled and analyzed once a day. I recommend having the configurations analyzed after each working shift so that you have automated documentation of the changes performed by each shift. The process of pulling configurations doesn't usually have high overhead, so having it run multiple times per day isn't a big deal.
- Dedicate a system to running the configuration management solution. Doing so ensures that the computer is always available to pull configurations from devices and always available for reporting or other tasks. I don't recommend installing the solution on your personal workstation, as too many factors can affect the solution's ability to perform in a large environment.

Most agentless network device configuration management solutions, such as those from AlterPoint and Ecora, have a fairly easy deployment methodology:

1. Install the software.
2. Either use auto-discovery to add devices or add them manually.
3. Configure options for pulling configurations and sending reports or automated change notifications.

In summary, take things slowly, add devices a few at a time, and make sure that the solution is meeting your expectations and your pilot test parameters (you *did* pilot the solution in a lab before implementing it in production, right?). Helpful configuration management solutions will make this process easy.

If you're looking for an even simpler deployment methodology, try this:

- Deploy the solution and allow it to discover your devices. Configure it to “see” any devices it doesn't discover.
- Within the solution, create your configuration standards. Allow the solution to create reports indicating how compliant your devices are with those standards (expect to see a very high noncompliance rate at this point; 100% noncompliance isn't at all rare).
- *Manually* reconfigure devices to bring them into compliance with your standard. This allows you to proceed slowly and ensure nothing impacts production services.
- Once everything is configured to standard, allow the solution to alert you when noncompliant changes are made, and allow it to automatically repair (or *remediate*) any noncompliant changes you feel comfortable with it managing.

Topic 6: Enterprise Network Configuration Management

Q 6.1: How does policy-based management make it easier to manage a large numbers of devices?

A: One of the biggest network management challenges in an enterprise environment is maintaining consistency. With hundreds—or even thousands—of network devices to manage, ensuring that they remain consistently and correctly configured can be difficult, if not downright impossible. This challenge applies especially to service providers managing customer networks. With multiple networks to worry about—each having its own unique requirements and configuration standards—ensuring consistency can be a nightmare.

Configuration templates have long been used to help ensure consistency. Listing 6.1, for example, shows a portion of a secure Cisco IOS template written by Rob Thomas (you can find the complete template at <http://www.cymru.com/Documents/secure-ios-template.html>). As you can see, this template specifies several “best practice” configuration settings to create a more secure IOS within Cisco routers. The problem with a template is that it is only good for the initial configuration of a router. For example, this template configures the router to use Terminal Access Controller Access Control System (TACACS) for accounting and authentication; if your TACACS configuration changes, you can’t easily reuse this template to change only that one setting. Instead, you would need to change that one section of settings on each router.

```
! Secure router configuration template.
! Version 3.1
! @(#)Secure IOS template v3.1 17 NOV 2003 Rob Thomas robt@cymru.com
! @(#)http://www.cymru.com/Documents/secure-ios-template.html
!
! This configuration assumes the following topology:
!
! Upstream/Internet
! 5.5.5.1/24
!     |
! 5.5.5.254/24 (Ethernet 2/0)
! THIS ROUTER
! 6.6.6.254/24 (Ethernet 2/1)
!     |
! 6.6.6.1/24
! Firewall
! 7.7.7.1/24
!     |
! 7.7.7.0/24
! Intranet
!
! In this case, 7.7.7.5 is the loghost, FTP server, etc.
! for the router. It could also be the firewall if
! circumstances dictate.
!
service nagle
service tcp-keepalives-in
service tcp-keepalives-out
! Show copious timestamps in our logs
service timestamps debug datetime msec show-timezone localtime
service timestamps log datetime msec show-timezone localtime
```

```

service password-encryption
no service dhcp
!
hostname secure-router01
!
boot system flash slot0:rsp-pv-mz.121-5a.bin
logging buffered 16384 debugging
no logging console
enable secret <PASSWORD>
no enable password
!
! Use TACACS+ for AAA. Ensure that the local account is
! case-sensitive, thus making brute-force attacks less
! effective.
aaa new-model
aaa authentication login default group tacacs+ local-case
aaa authentication enable default group tacacs+ enable
aaa authorization commands 15 default group tacacs+ local
aaa accounting exec default stop-only group tacacs+
aaa accounting commands 15 default stop-only group tacacs+
aaa accounting network default stop-only group tacacs+
tacacs-server host 7.7.7.5
tacacs-server key cheezit
!
! In the event that TACACS+ fails, use case-sensitive local
! authentication instead. Keeps the hackers guessing, and
! the router more secure.
username <USERNAME> password <PASSWORD>
!
! Don't run the HTTP server.
no ip http server
no ip http server-secure
!
! Allow us to use the low subnet and go classless
ip subnet-zero
ip classless
!
! Disable noxious services
no service pad
no ip source-route
no ip finger
no ip bootp server
no ip domain-lookup

```

Listing 6.1: A portion of a secure Cisco ISO template.

Network configuration management solutions can help deploy that change to hundreds of routers by allowing you to script the change—better solutions can even generate a script for you. But ease of deployment aside, you have now “violated” your original template, meaning you can no longer use it as a comparison point to determine whether your routers are properly configured. In short, templates are great for initial configurations, but they don’t help with the ongoing maintenance of the device.

Here is where policy-based management comes in. You might still use a template like the one that Listing 6.1 shows to help configure devices initially, but from there, a policy-based configuration management solution takes over. You define a set of *rules*, each of which relates to a single configuration setting. For example, you might create each of the following settings as a single rule:

- No service pad
- No IP source-route
- No IP finger
- No IP bootp server
- No IP domain-lookup

These rules are then combined into a *policy*, which in this example might be named “Disable unnecessary services.” The policy is then assigned, or applied, to one or more devices. Assigning the policy to a device causes the configuration management solution to alert you whenever that device falls out of compliance with the policy; some solutions might even provide for automated remediation, reconfiguring the device to meet the policy again without any intervention from you.

An effective network configuration management solution will allow policies to be assigned dynamically. In other words, policies would be assigned to all devices meeting certain criteria, such as a specific device OS (like Cisco IOS), a firmware version number, and so forth. The benefit of this dynamic assignment is that the policy can be made to apply to all devices that need it, without any intervention from you. If a new router is added to your network, the appropriate policies are *automatically* enforced, and you’re not required to manually apply them to the new device. Dynamic assignment therefore provides better consistency in your network’s configuration. By contrast, static assignment requires you to determine which devices require a certain policy and to constantly maintain your assignments as devices are added to and removed from the network.

Once in place, policies can provide a more effective means of managing large numbers of devices. For example, suppose you create a policy to enforce the correct TACACS configuration on Cisco devices. It might contain the same configuration settings as the template you use to initially configure devices:

```
aaa new-model
aaa authentication login default group tacacs+ local-case
aaa authentication enable default group tacacs+ enable
aaa authorization commands 15 default group tacacs+ local
aaa accounting exec default stop-only group tacacs+
aaa accounting commands 15 default stop-only group tacacs+
aaa accounting network default stop-only group tacacs+
tacacs-server host 7.7.7.5
tacacs-server key cheezit
```

If your TACACS server changes, you don't need to manually reconfigure each device. Instead, you simply change your policy:

```
aaa new-model
aaa authentication login default group tacacs+ local-case
aaa authentication enable default group tacacs+ enable
aaa authorization commands 15 default group tacacs+ local
aaa accounting exec default stop-only group tacacs+
aaa accounting commands 15 default stop-only group tacacs+
aaa accounting network default stop-only group tacacs+
tacacs-server host 7.7.7.10
tacacs-server key slurpee
```

All existing devices aren't configured this way, and therefore don't comply with the policy; the configuration management solution will detect this discrepancy and either alert you to non-compliant devices or, even better, remediate the devices' configurations and bring them into compliance—in other words, reconfigure them for you. Network configuration becomes a matter of simply maintaining your policies correctly, and allowing the policies to filter down to the devices in terms of the proper configuration.

Another major enterprise challenge is the array of devices found on the network, and the array of different manufacturers who provide those devices. Few networks are “all Cisco” or “all Nortel;” most use best-of-breed solutions and select routers, switches, firewalls, and other network infrastructure devices from a variety of manufacturers. This setup makes maintaining configuration policies more challenging because you might need to maintain multiple versions of each policy to represent each different device vendor. Network configuration management solutions can help by providing configuration *abstraction*, a process whereby vendor-specific configuration settings are displayed to you (in a “policy builder” of some kind) as vendor-neutral information. The solution then translates those policies upon application, as Figure 6.1 shows, into the vendor-specific configurations. Because the solution understands the different configuration settings, you don't need to; you can maintain a single set of abstract policies and let the solution worry about how to physically implement them.

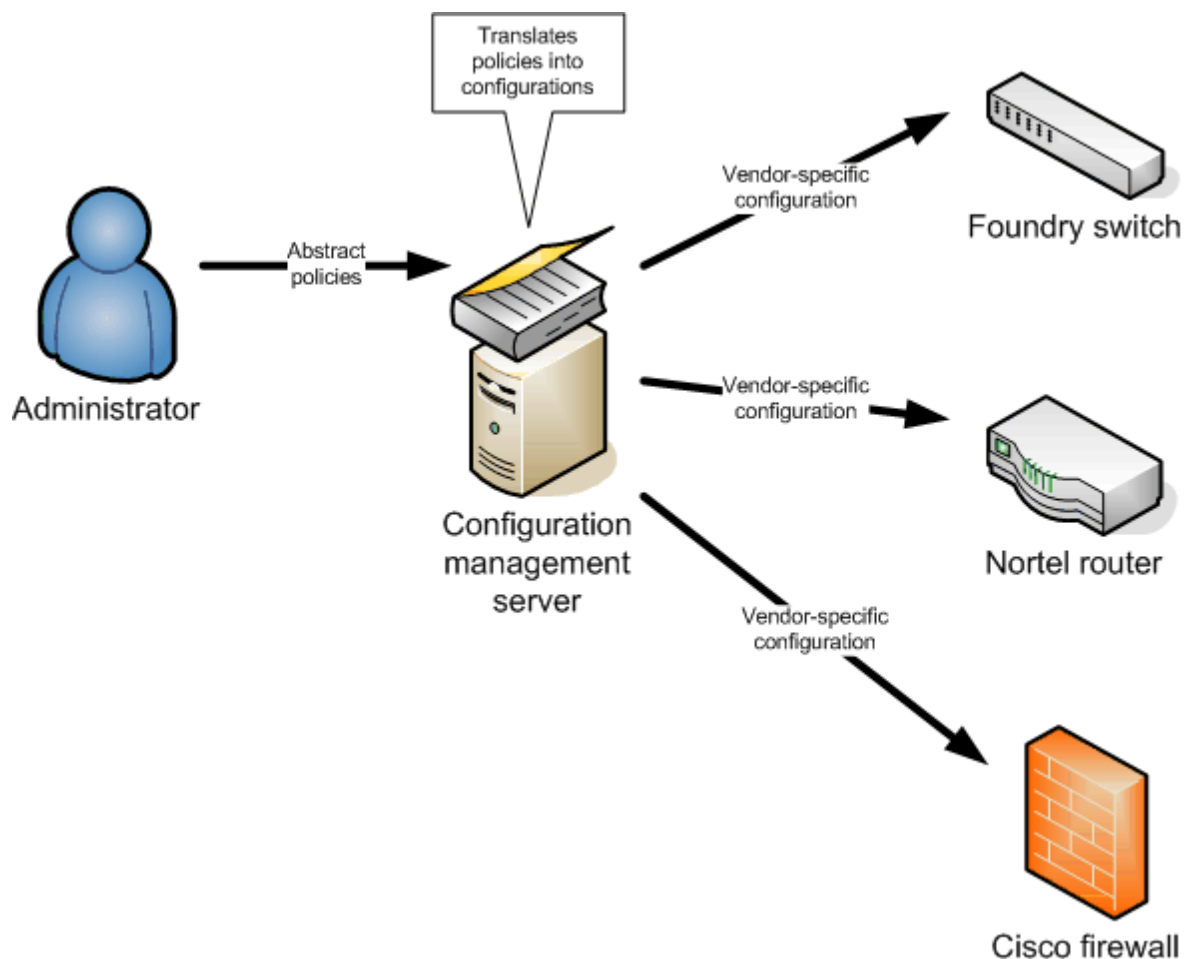


Figure 6.1: Translating abstract policies into configuration settings.

The ideal network configuration management solution will combine abstraction with policy-based management to provide easier, more consistent, single-seat administration for any number of devices. It may even be able to discover additional devices as they are added to the network (device discovery is a common feature) and apply the necessary policies, ensuring that your network always remains configured the way you want it to be.

Q 6.2: We have hundreds of network devices, so manually retrieving configurations via Trivial File Transfer Protocol just isn't an option. What are our alternatives?

A: Trivial File Transfer Protocol (TFTP) is great for small environments, but like so many manual solutions, it doesn't scale well. My preference in a large environment is to select a network device configuration management solution. Such solutions usually rely heavily on TFTP and other common protocols to retrieve device configurations, but automate the entire process so that dealing with hundreds of devices doesn't require a team of dedicated people working 'round the clock.

You can, of course, stick with TFTP if you find some way to automate it. Listing 6.1 shows an example Linux script that can use TFTP to pull configurations from either Cisco or Ascend routers (this example is adapted from a more complete script that is available at <http://www.securiteam.com/exploits/5RP0E000AA.html>).

```
#!/bin/sh
# grabrtrconf:
# by: Eric Monti 11/1997

TFTPLISTEN="true"
DIR=/tftpboot #might want to use something else
WAIT=6
INT=ppp0

test "$4" = "" && echo "Usage: `basename $0` target ↵
  write-community tftphost filename [type]" && exit 1
TYPE=$5
test "$5" = "" && TYPE="cisco"

IPADDR=$3
test "$IPADDR" = "." && IPADDR=`/sbin/ifconfig $INT | ↵
  grep inet | sed "s/\:\/\ /" | awk '{print $3}'`

echo $3

if [ -n $TFTPLISTEN ];then
echo "tftp dgram udp wait root /usr/sbin/in.tftpd ↵
  in.tftpd $DIR" > /tmp/ind.conf
/usr/sbin/inetd -d /tmp/ind.conf &
rm /tmp/ind.conf
rm -f $DIR/$4
touch $DIR/$4
chmod 666 $DIR/$4
fi

#CISCO get config
test "$TYPE" = "cisco" && \
snmpset -r 3 -t 3 $1 $2 .1.3.6.1.4.1.9.2.1.55.$IPADDR s $4

#ASCEND get config
if [ "$TYPE" = "ascend" ];then
  snmpset -r 3 -t 3 $1 $2 .1.3.6.1.4.1.529.9.5.3.0 a $IPADDR
  snmpset -r 3 -t 3 $1 $2 .1.3.6.1.4.1.529.9.5.4.0 s $4
```

```

snmpset -r 3 $1 $2 .1.3.6.1.4.1.529.9.5.1.0 i 3
snmpset -r 3 $1 $2 .1.3.6.1.4.1.529.9.5.3.0 a "0.0.0.0"
snmpset -r 3 $1 $2 .1.3.6.1.4.1.529.9.5.4.0 s ""
fi

sleep $WAIT

if (test `pidof in.tftpd`);then
  echo Receiving file:
  while (test "`pidof in.tftpd`");do
  echo -n .
  sleep 1
  done
  echo
  echo Transfer Complete

fi

if [ -n $TFTPLISTEN ];then
kill `cat /var/run/inetd.pid`

fi

```

Listing 6.1: A sample Linux script that can use TFTP to pull configurations.



Some of the lines of code in Listing 6.1 were too long to fit on one line in this format; the line continuation character (`\`) was used to break up lines of code that were too long. You should type these lines of code all on one line, without the line continuation character.

Now, if you're perceptive, you'll have noticed the URL to which I referenced this script: <http://www.securiteam.com>. This script is listed as a security exploit because, as the site states, "This allows a remote attacker fill knowledge of the router configuration, routes, etc." And the site is absolutely correct; if your devices are accessible from the Internet, you might want to seriously consider disabling TFTP (or Simple Network Management Protocol—SNMP—which is what this script uses to force the device to send its configuration via TFTP) on them to prevent this sort of thing from happening. Fortunately, most companies' internal routers use non-routable IP addresses (such as 192.168.0.1), meaning the routers can't be contacted by outside attackers.

Topic 7: Compliance Management for the Network

Q 7.1: How can policies help us better manage regulatory compliance for our network?

A: Most regulatory compliance requirements—such as those of the Health Insurance Portability and Accountability Act (HIPAA), the Sarbanes-Oxley Act, 21 Code of Federal Regulations (CFR), various European Commission rulings, and so forth—revolve around two central themes: security (often referred to as privacy) and accountability. The driving philosophy behind these regulations is to provide protection for sensitive data (such as healthcare or financial information) and to provide a layer of accountability so that all access to this sensitive data is recorded and can be tracked.

Network infrastructure devices don't usually need to worry about direct data access or accountability; however, because the network transmits all of that data, the network obviously plays a role in protecting the data's confidentiality. Thus, network devices need to be configured to provide the necessary privacy. They also need to be configured to provide the necessary auditing so that any changes to the devices' configurations that might compromise privacy can be tracked.

As I've discussed in other tips, most network devices have fairly primitive internal capabilities when it comes to logging device access; instead of building logging databases into devices, the industry has evolved around external technologies such as Terminal Access Controller Access Control System (TACACS) and Remote Authentication Dial-In User Service (RADIUS), which receive *accounting*, or logging, messages from devices and store them in server-based databases. None of these logging technologies typically provide a sufficient level of detail with regard to which changes are made in a device—they tend to focus primarily on administrative access to the device, not what the administrator does once he or she is in. Network configuration management solutions can use the logging messages as a sort of trigger. The messages inform the solution that an administrator has accessed the device (and might therefore have made changes), so the solution can pull the device's configuration and look for changes.

All of this activity, however, depends entirely on the device being configured to send those logging messages. A Cisco router can be configured to use TACACS, a popular logging technology, with a configuration similar to the following:

```
aaa new-model
aaa authentication login default group tacacs+ local-case
aaa authentication enable default group tacacs+ enable
aaa authorization commands 15 default group tacacs+ local
aaa accounting exec default stop-only group tacacs+
aaa accounting commands 15 default stop-only group tacacs+
aaa accounting network default stop-only group tacacs+
tacacs-server host 7.7.7.5
tacacs-server key cheezit
```

The same router could be configured to use syslog with the following configuration:

```
service timestamps log datetime localtime
no logging console
no logging monitor
logging 192.168.1.100
```

Cisco Catalyst switches would use a slightly different configuration for syslog:

```
set logging server enable
set logging server 192.168.1.100
set logging level all 5
set logging server severity 6
```

Cisco PIX firewalls have an even different syslog configuration:

```
logging on
logging standby
logging timestamp
logging trap notifications
logging facility 19
logging host inside 192.168.1.100
```

The point in listing three syslog configurations for devices from a single manufacturer is to highlight that all the configurations are different and that maintaining those configurations on a network with multiple *vendors* can be difficult to say the least. This arena is where policy-based management applies.

The bottom line is that it is critical that certain device configuration settings remain in place in order for your network to remain compliant. Ensuring that configurations remain in place can be a daunting manual task. You must become familiar with a *lot* of different configuration files and you need to look at them *constantly*; being out of compliance for even a single moment means that someone could reconfigure a router and lower its security without being noticed.

Policy-based management can automatically alert you to devices that aren't compliant, and can even automatically reconfigure those devices to be compliant again, removing the possibility for auditing to be turned off without someone at least being notified of the problem. If your network configuration management solution also supports configuration abstraction, you'll only need to configure the policy *once*, using a vendor- and device-neutral syntax; the configuration solution will translate that into device-specific configuration settings as necessary and implement the policies for you. In short, policy-based management is the single most effective tool you can have to ensure your network remains compliant.

Q 7.2: What does my network configuration have to do with compliance?

A: These days, the word *compliance* is used as a shortcut for *compliance with legal requirements*, and typically refers to some industry-specific regulations or legislation dealing with information management; examples in the United States include the Health Insurance Portability and Accountability Act (HIPAA), the Sarbanes-Oxley Act, 21 Consolidated Federal Rules (21 CFR), the Graham-Leach-Bliley Act, and so forth.

The connection between these rules and your network configuration can be subtle. Generally speaking, these laws concern themselves with the security, privacy, and accountability of specific types of information. HIPAA, for example, is all about patient information in the health care industry; Sarbanes-Oxley deals primarily with financial information and practices for auditors. Their broad goal is to ensure that all data is maintained as confidential, access to data is controlled and monitored, and access to data is *accountable*, meaning you can always look back and see who accessed what and what changes they made, if any, to data covered by the rules.

It's easy to see how network file servers become involved, because they store a lot of the data these laws are concerned with. But how does data go to and from file servers? The network. Thus, if someone could compromise your routers, that person could in theory gain access to every byte of data that passes through those routers. Hence, your routers need to be secured. In order to maintain that security, every change to a router's configuration needs to be examined to make sure it doesn't compromise the router's security. Every change needs to be *accountable* so that any changes that *do* compromise security can be traced to the guilty party. A Sarbanes-Oxley Report, for example, might detail every change made to every network device in your environment, along with information about when the change was made, who made it, and the details of the change. Such a report would help an auditor examining your Sarbanes-Oxley compliance to review changes and ensure that they were all made within the scope of a configuration management process—that is, changes were reviewed for their security implications, approved by management, and deployed as planned.

HIPAA carries roughly similar requirements, including the requirement that anyone in possession of health care records provide a report of all disclosures of those records. As it would be nearly impossible to record disclosures made accidentally over the network—such as through a network device such as a switch or router—you have to make sure such disclosure can't occur. Again, this means a secure configuration and complete accountability for all changes made to the configuration.

These are all difficult tasks without the help of a decent configuration management solution, such as those offered by companies such as AlterPoint (<http://www.alterpoint.com>) and Voyence (<http://www.voyence.com>). High-end configuration management solutions are designed to detect changes to device configurations, report on those changes, and even undo unauthorized or improper changes. They can also help enforce a configuration management workflow process, which ensures that changes are properly reviewed, approved, scheduled, deployed, and logged for auditing purposes.

Q 7.3: How can we make our network verifiably compliant?

A: Verification is a key factor of most compliance efforts. Outside the realm of technology, compliance is usually verified by testing the *end state*. For example, if you've moved all of your sensitive paper files into a locked storage room, you can verify your level of compliance by testing the door to see if it's locked and if it can be readily opened without using a key. You're not checking purchase orders to see if a lockable door was purchased and installed; you're testing the end condition of a locked door being in place. You may also keep a list of individuals to whom you've issued keys, and use that as a part of your compliance verification.

Technology is tougher. We often can't easily test for the end state. For example, if you use file permissions to secure a group of sensitive files on a file server, you can test to make sure a specific user who *shouldn't* have access *doesn't*, but that won't guarantee that other unauthorized users won't have access. You can't easily test the entire range of possibilities to verify that your files are compliant. Instead, verification occurs by testing the configuration itself: Examining the access control list (ACL) on the files to ensure they're configured properly, then trusting in the file server's operating system (OS) to properly enforce the configured ACLs.

☞ "Trusting the OS" is a key reason why patch management is so important for compliance management. The unfortunate fact is that most OSs—and I'm including network device OSs in this statement—*can't* be fully trusted, because they have bugs. When those bugs are fixed in a patch, the patch *must* be deployed in order to help ensure the OS's proper, intended operation.

Moving into the world of device configuration management, you'll find that creating a *verifiably compliant* network is also a matter of testing the configuration of your network devices. You create a configuration that addresses your compliance needs, deploy that to your network devices, then periodically audit your devices to ensure that the desired configuration is in place. This periodic audit is part of the verification process. Software solutions can make the auditing easier by centralizing device configurations into a configuration database; in fact, such databases can be a real benefit because you can often give auditors access to the database—allowing them to verify configurations—without having to give them access to devices, and without having to manually dump devices' configurations into a file or hardcopy for auditing purposes.

A high-end network configuration management solution will go a step further by notifying you of device changes—a *really* good solution can distinguish between changes that affect compliance and those that don't, based on templates you provide—and allowing you to respond to changes appropriately. This setup is better than auditing, actually, because the solution is on the job every hour of every day, whereas weeks or months typically separate manual audits. High-end network configuration management solutions can even provide reports that assist with a manual audit, including a report of all configuration changes made to a device over a specific time period. All of this helps to make your network more *verifiably* secure, allowing you to meet legislative and other compliance requirements.

Q 7.4: How effective are audits at maintaining compliance in network devices?

A: I would go so far as to say that audits are *totally useless* at maintaining compliance in network devices. I realize, however, that the compliance industry as a whole relies almost entirely upon point-in-time auditing, so I'll temper that assessment to: *You can do a lot better.*

The problem with audits is that they occur only on a periodic basis, sometimes weeks or months apart. The auditor's job is generally to examine a network device's configuration and look for certain key configuration items: SNMP community strings, administrative passwords, TACACS/RADIUS configuration, and so forth; all configuration items that in some way support the device's security or accountability (logging) functions. So if Johnny Auditor checks out a router on Monday and finds everything is okay, what does that tell you about the router's configuration on Tuesday? Or Friday a week from now? Or 2 months from then? Answer: *nothing*, which is why I feel that auditing is functionally useless in a fundamentally dynamic field like network configuration management. Administrators know perfectly well when the audit is coming and they know what the auditor is going to see that he or she won't like. So they fix those things and put them right back when the auditor is gone. I've done it myself, which is why I've never understood how auditing is supposed to help.

I did say, however, that you can do better, and you can. The goal is *continuous configuration management*, meaning your configurations are managed to a desired state at all times, every hour of every day, whether anyone's looking or not. The technologies now exist to make this entirely possible, eminently practical, and even affordable. You simply need to use an automated network configuration management solution. A number of them exist on the market from companies such as AlterPoint, TripWire, Ecora, Voyence, and more; many of them are now offering some form of continuous configuration management.

The lower level of these solutions offers automated change notification. Let's face it: Network device configurations don't change *that* often. When an authorized change does occur, a relatively small number of individuals within the company will usually know about it. Unauthorized changes are what you want to watch out for, and that's where these solutions help. By configuring your network devices to provide SNMP traps, Syslog messages, or RADIUS/TACACS logging, a network configuration management solution can be notified of key events that may indicate a change in progress—events such as an administrator logging on and placing the device into configuration mode. These key events trigger the solution to download the device's configuration and compare it with the last-approved configuration to see what has changed. Any changes found are forwarded to one or more administrators as an alert. Thus, administrators don't need to wait for an audit to tell them something was misconfigured; they'll immediately know a change was made, be able to review the change, and quickly change it back if it was unauthorized.

Higher-level configuration management solutions go a bit further by allowing you to define configuration templates that detail your authorized configuration. This may include settings such as an SNMP community string used in your organization, or a specific TACACS configuration used to ensure all device activity is logged. When changes are made to a device, the solution pulls the device's configuration and compares it with these templates. Any configuration setting that conflicts with the template can trigger an alert, meaning you'll immediately know a *bad* configuration change was made and take appropriate action. The solution may even be able to automatically *remediate* the change, reconfiguring the device automatically to reflect your templated configuration. AlterPoint DeviceAuthority is one solution that can take this step, automatically fixing certain configuration parameters if you want it to.

Such automatic remediation and notification is *considerably* better than point-in-time auditing. In fact, the auditor's job might be redirected to audit your configuration templates rather than device configurations. By ensuring your templates are correct, the auditor will be ensuring that devices managed by that template are correct. Suddenly, you're no longer dealing with device configurations! Instead, you're managing at a more abstract level, configuring templates to reflect the way your network should work, and allowing the configuration management solution to implement and enforce those templates on your devices.

Q 7.5: Once my network is compliant, how can I ensure it stays that way?

A: The best way to ensure compliancy is to use a network configuration management solution that can push changes out to devices, enforce standards (through templates or policies), and act as a one-stop-shop for device management. In other words, the solution should provide administrators with the ability to access devices via Telnet, *through* the solution and using the solution's built-in security. Most solutions offer this capability and can even capture all Telnet activity in a log. Of course, it's always possible for administrators to bypass the solution and Telnet directly to devices *if they know devices' configuration passwords*. Make sure that doesn't happen!

Most high-end solutions can be configured to automatically change, or *roll*, device configuration passwords on a regular basis (say, every 45 or 90 days). You can select new passwords for your devices using the following tips:

- Use a unique, difficult-to-remember password for each device, such as m%t6@YhgeReem
- Document each device's password and lock that document in a fire safe to which network administrators don't have access

Administrators will be able to Telnet into devices by asking your network configuration management solution to do so on their behalf; the solution knows the password, but administrators don't. This setup prevents administrators from bypassing the solution. Should some sort of emergency occur, you'll still have the ability to perform direct Telnet by unlocking the safe and retrieving the current passwords.

This technique simply reflects that fact that network devices were never design with robust management security and auditing features in place. By layering a more robust solution on top of the devices' native capabilities, and by preventing direct access to devices, you can allow that robust solution to provide better management capabilities, make compliance easier to maintain, and make security easier to enforce (as well as more granular).