realtimepublishers.com™

Tips and Tricks Guide™ To

Network Configuration Management

AlterPoint

Don Jones

Introduction

By Sean Daily, Series Editor

Welcome to *The Tips and Tricks Guide to Network Configuration Management*!

The book you are about to read represents an entirely new modality of book publishing and a major first in the publishing industry. The founding concept behind Realtimepublishers.com is the idea of providing readers with high-quality books about today's most critical IT topics—at no cost to the reader. Although this may sound like a somewhat impossible feat to achieve, it is made possible through the vision and generosity of corporate sponsors such as AlterPoint, who agree 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 books does not in any way diminish their quality. Without reservation, I can tell you that this book 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). In addition to the free nature of the books, this publishing model provides other significant benefits. For example, the electronic nature of this eBook makes events such as chapter updates and additions, or the release of a new edition of the book possible to achieve in a far shorter timeframe than is possible with 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 although it is true that the sponsor's Web site is the exclusive online location of the book, this book is by no means a paid advertisement. Realtimepublishers is an independent publishing company and maintains, by written agreement with the sponsor, 100% editorial control over the content of our titles. However, by hosting this information, AlterPoint has set itself apart from its competitors by providing real value to its customers and transforming its site into a true technical resource library—not just a place to learn about its company and products. It is my opinion that this system of content delivery is not only of immeasurable value to readers, but represents the future of book publishing.

As series editor, it is my raison d'être to locate and work only with the industry's leading authors and editors, and publish books that help IT personnel, IT managers, and users to 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 do so by sending an email to feedback@realtimepublishers.com, leaving feedback on our Web site at www.realtimepublishers.com, or calling us at (707) 539-5280.

Thanks for reading, and enjoy!

Sean Daily

Series Editor

Note to Reader: This book presents tips and tricks for six 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: Change Management Best Practices
- Topic 2: Network Management Security
- Topic 3: Network Management Troubleshooting
- Topic 4: Change Management Techniques
- Topic 5: Selecting and Deploying a Network Device Management Solution
- Topic 6: Enterprise Network Device Management

Copyright Statement

Topic 1: Change Management Best Practices

Q 1.1: What is 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. *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, 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. Instead of thinking of change management as extra work, I like to think of change management as *saving* me work: By simply following some simple methodologies and processes, I can ensure that changes to network devices never become a nightmare. Or, at least, if they *do* become a nightmare, I can quickly recover without having to spend all night at the office!

Q 1.2: What's the best way to "do" 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.

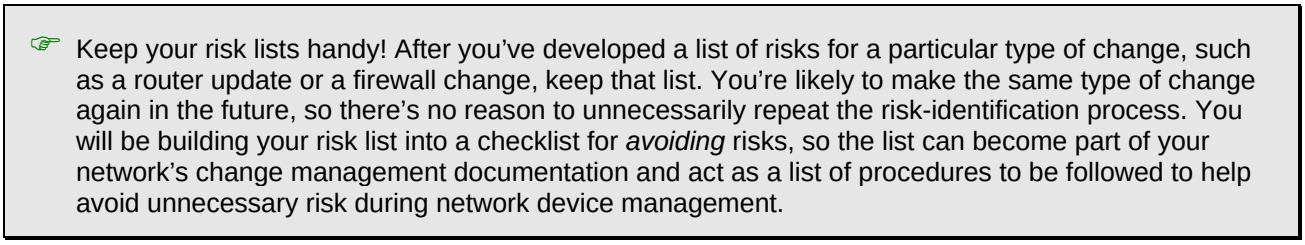
How you actually conduct each step of the planning process depends on your environment and your personal preferences. In the next four sections, I'll provide some examples to get you started.

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.



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's 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've got 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're 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'll 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 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've 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'll 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, I've worked with companies that 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. 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 a complete set of documentation for the change.

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

Want to Know More?

No matter what you do, make sure that you have a system in place for change management. If you'd 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. That 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/chmgmt.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 kick-start a new change management process in your organization.

Q 1.3: 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 change-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, change 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.
- Change management shares the responsibility for making changes. A good change-management process includes several sign-offs, eliminating the need for a single person to bear the brunt of mistakes.
- A well-designed change-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.

In addition to changing administrators' perception of the change-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.

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 on the proper change-management process in your organization. Unfortunately, assuming your administrators have password access to your devices, there's almost no way to prevent them from making changes without following your change-control process.

In a few years, you'll see a new class of network device-management application that builds upon the solutions already available. This new class of solutions will provide a complete front end to device management, letting administrators use a friendly graphical user interface (GUI) or even an intermediate command-line command to make changes. The application will push the changes to the network devices using the devices' configuration passwords. Administrators won't actually 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.

These types of applications already exist for network server management. Aelita (http://www.aelita.com), for example, makes a suite of applications that provide this type of management interface for Windows networks. Network device management hasn't caught up yet, due in part to the variety of devices in common use on large networks. But these solutions are on their way. Companies to watch for these applications include AlterPoint (http://www.alterpoint.com) and Network Mantra (http://www.networkmantra.com); both emerging leaders in network device management.

Today, 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, add detailed change-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.

Q 1.4: 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.

Some of the highest-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 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 not only includes its own version number, but also includes a comment to tell administrators which type of device and which 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 can 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 1.5: How can I ensure that all of the devices on my network are accounted for and under change management control?

A: Whenever I walk into a new client to talk about change control for network device management, one of the first things I ask for is a network diagram and an inventory of network devices. After all, a key in change management is ensuring that *every* device is under control, and the client's mix of devices often dictates which change management solutions I can recommend. A few clients have surprised me by having a complete inventory on hand; most have, at best, a few incomplete diagrams and an administrator or two who have a pretty good recollection of which devices are on the network. Needless to say, some research is required to generate a more accurate inventory.

Automatic Device Discovery

There are a number of products that can automatically search for managed network devices. For example, Microsoft Visio 2002 Enterprise Edition provides a fairly accurate network discovery feature. Microsoft Systems Management Server (SMS) 2.0 includes a similar feature. Some change management solutions, including AlterPoint DeviceAuthority and ReadyRouter also include device discovery capabilities.

One software package that I've had a lot of success with is Synexsys Inventory. It's actually a very full-featured network inventory package, including capabilities that extend to desktop software license management. Thus, many companies in need of asset control capabilities can benefit from it. It's also got a fantastic Simple Network Management Protocol (SNMP) discovery module, which seeks out managed network devices. Practically every device supports SNMP (unless you've disabled SNMP), so you can use Synexsys Inventory to quickly acquire an accurate report of your network device assets.

Any automated network discovery software will likely return devices that you don't care about. Network-attached printers, for example, will often respond to SNMP queries, and you might not have any desire to place them under your change management program. However, it's far better to have a complete, accurate inventory that contains too much information than to miss some critical device simply because everyone's forgotten about it!

Device Discovery and Security

Automatic device discovery can also be a great security aid. I've been to a number of clients at which, at one point or another, unauthorized network devices wreaked havoc on their networks. In one case, an unauthorized router started advertising itself and its routes. The router wasn't correctly configured for the network, so all routing operations became pretty unreliable in a short while.

If you've picked up a package such as Synexsys Inventory or a change management solution that includes an auto-discovery option such as DeviceAuthority, you can periodically repeat the auto-discovery process to see whether any new devices have popped up on your network. For example, Figure 1.1 shows DeviceAuthority's auto-discovery wizard, which queries for SNMP devices on your network.

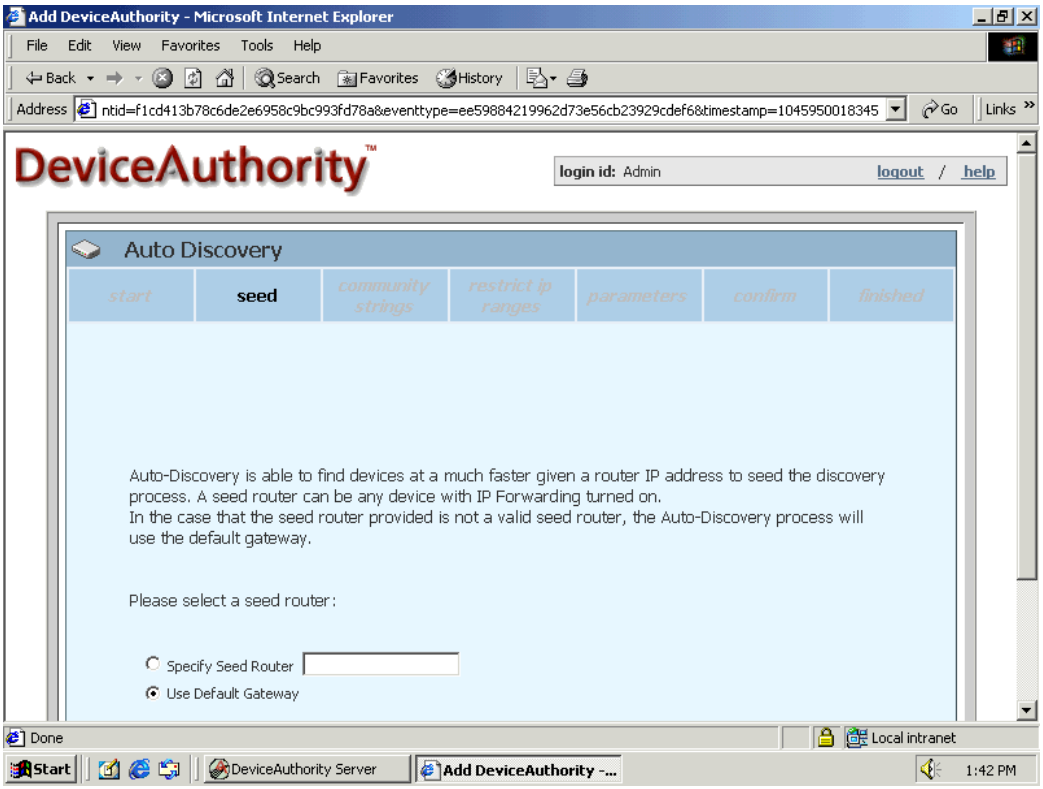


Figure 1.1: The auto-discovery function in DeviceAuthority can start with a seed router for faster operation

> Most network device discovery routines simply query for SNMP devices on your network. Typically, that will return all of the managed devices on your network. It's possible, of course, that you have non-SNMP devices. For example, most inexpensive hubs and switches won't respond to SNMP. However, those devices are inherently unmanageable. You don't need to worry about keeping them under change control, because there's nothing in them that can be changed. Capturing a list of your SNMP devices will tell you which devices can be configured, and those are the devices that you'll want under change control.

Device Discover and Documentation

One reason I'm a fan of Visio 2002 Enterprise Edition is its network discovery feature. In addition to locating the devices on your network, it automatically constructs a complete network diagram, showing how the devices are connected and even labeling router interfaces with the appropriate IP addresses and other information. The discovery process can take a very long time to complete, and the finished diagrams usually need some cleaning up and rearranging to be more useful, but it's a great feature that can be a real timesaver. The feature works best when it's running on a very powerful workstation and when your routers use a routing protocol (such as RIPv1, RIPv2, or Open Shortest Path First—OSPF) to exchange routing information.

SNMP, Discovery, and Security

For security reasons, some organizations choose to disable or in various ways restrict the use of SNMP on their network devices. Unfortunately, doing so will prevent most auto-discovery routines from locating managed devices, because those routines generally rely almost entirely on SNMP. Thus, you'll have to manually configure each device in the change management solution rather than having a tool seek them out automatically.

If your organization has SNMP enabled on your network devices, you'll have an easier time generating a network device inventory. However, don't underestimate the potential security risk that SNMP and auto-discovery can represent. One of the biggest battles an intruder fights is finding out the structure and architecture of your network, and with SNMP enabled, those intruders can utilize auto-discover routines just as easily as you can. Auto-discovery is also very, very difficult to detect when it's in progress. You can minimize the effectiveness of an intruder's auto-discovery efforts by changing your devices' SNMP community strings to something very difficult to guess, and by taking steps to secure your physical network so that unauthorized persons don't have access to it.

> ☞ There's been a recent upswing in so-called *war driving,* the practice of driving around with a wireless laptop looking for unsecured wireless networks. Most war drivers are simply looking for free Internet access through your wireless LAN; others might have more nefarious purposes. An unsecured wireless LAN can represent an easy point of entry for intruders who want to conduct an SNMP auto-discover on your network. Changing community strings, using wireless encryption to restrict wireless access, and configuring wireless access points not to bridge SNMP packets will help prevent your wireless LAN from becoming a way for intruders to learn more about your network infrastructure. Of course, if you disable SNMP in your wireless access points, you won't be able to use automatic device discovery features as effectively.

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

A: Throughout this book I've been preaching the wonders of change management as a means of preventing problems—or at least of easily detecting them. Change management, however, is really just a form of documentation, and it's documentation that can really help avoid—or quickly determine the case 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 concerts
- Any out-of-band management configuration
- Cable requirements

For example, suppose you receive from your change management system a notification email that informs you that a particular router 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 tough 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'd know to get right over and 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.

Change Management

As I've mentioned before, change management over 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.

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 change management process. Although up-to-date diagrams can make troubleshooting vastly easier and more efficient, outdated diagrams can be a significant hindrance.

Q 1.7: We're an ISO9001 shop. How can we incorporate network device change management into our processes?

A: ISO9001 can seem like a difficult framework in which to run a business because you can't seem to do anything without following a flowchart. However, this requirement is a good thing and designed to make sure that your business' results are predictable and repeatable. In fact, ISO9001 is an excellent framework for change management—it forces you to use a formalized, repeatable process.

The easiest way to incorporate network device change management into your processes is to create an official ISO process for doing so. Figure 1.2 shows a sample flowchart that you can use as a starting point.

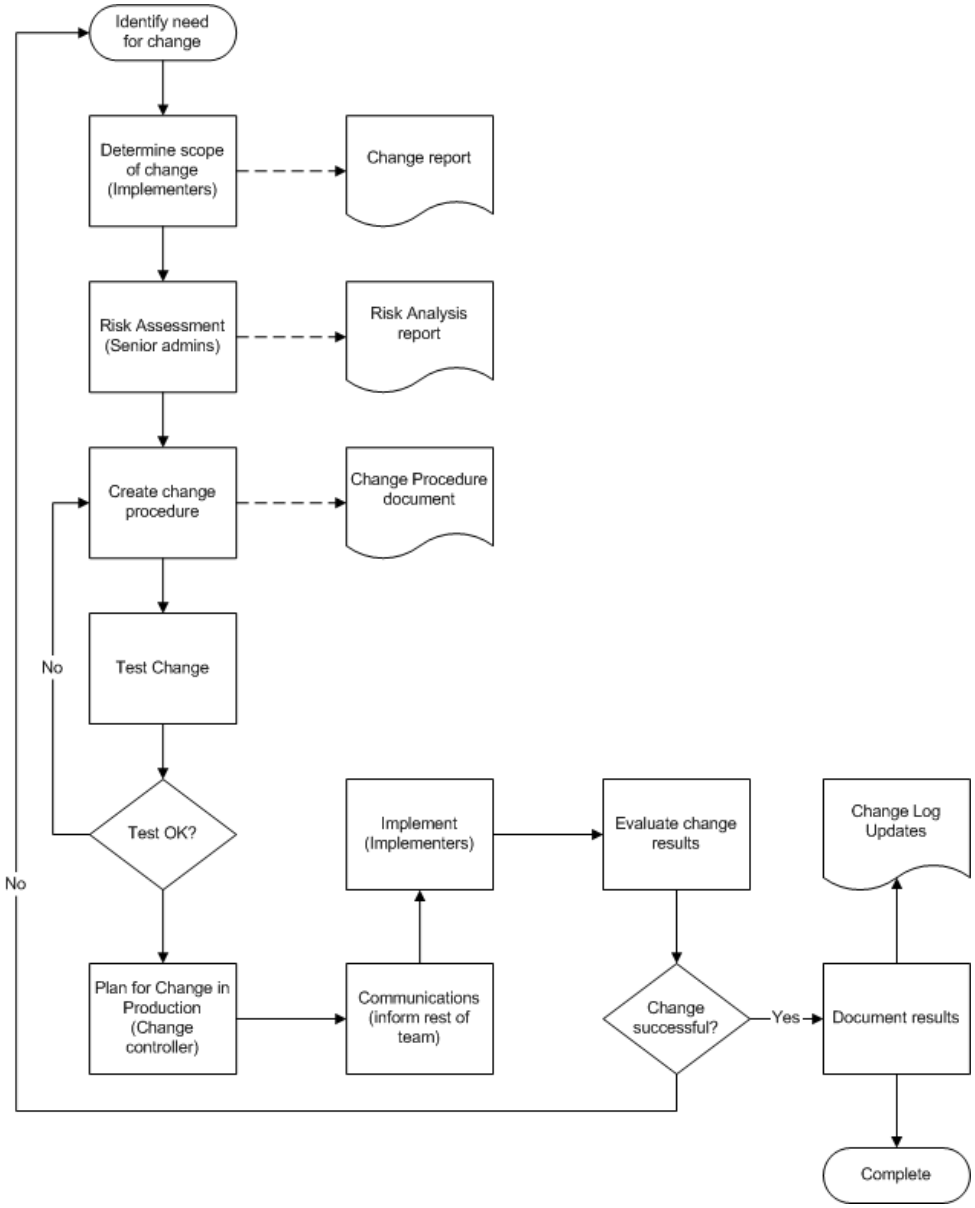


Figure 1.2: Sample change management process flowchart.

This flow incorporates many of the change management elements I've discussed in previous tips:

- Identify the specific change that needs to be made.
- A technically knowledgeable person then needs to figure out exactly what that change involves. For example, adding a new office may involve adding new static routes to a number of routers or simply involve a bit of additional routing protocol traffic with no actual configuration changes required. This scope—essentially a list of everything that will change—needs to become a formal document.
- Senior administrators should evaluate the risks of making the changes—as I've touched on in several earlier tips. Their analysis should be documented in a formal risk analysis, and the decision to proceed should be based upon that analysis.
- An official change procedure—the precise steps that will be taken to implement the requested change—must be documented and approved.
- The procedure should be tested in a lab environment if at all possible, particular if the change is deemed risky. If the test is unsuccessful, the procedure can be modified and the test repeated.
- If the test goes well, a change controller needs to plan for the change to actually be made in production. This process might involve the controller ensuring that all devices are currently backed up in case the change fails in production and a rollback is required.
- Communications involves informing anyone who might be affected by the change so that everyone involved can be alert for signs of failure and those affected don't misinterpret the effects of the change as a problem and initiate corrective action.
- The implementation should be evaluated for success. Ideally, the change plan should identify metrics for success so that the change can be definitively proclaimed a success or not.
- If the change was not successful, it needs to be rolled back and the entire process repeated to find out what went wrong. A change management solution can be especially effective at the rollback step, allowing devices' former configurations to be restored quickly.
- If the change was successful, it needs to be documented. The baseline configurations for affected devices should be updated to reflect the change, and the devices' configurations should be backed up. That way, if the device fails for other reasons, a restore will also include the recent changes.

Your company's change controller—usually a network device administrator—should own this change management process. You'll likely need to tweak this flowchart a bit to meet your specific needs, but it should give you a useful ISO-quality start.

Topic 2: Network Management Security

Q 2.1: 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. Hackers 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 include bank break-ins teaches us that a blueprint is the best way to plan your attack, so denying hackers your network blueprint is a wonderful first defense. Of course, SNMP's ability to change network devices' configuration settings can, of course, be lethal. Here are some 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're 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 pain, so see if 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 hacker 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 a great 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 harder 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, which is when a hacker 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. Find out 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 and use those lists to notify you of the latest fixes.
- Log SNMP traffic. Some devices provide an option to automatically log received SNMP requests. If they don't, you can use network sniffer devices 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.

Q 2.2: How can change 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.

Change management can help. First, a good change-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 documenting changes, a good change-management process can ensure that temporary changes are removed at the earliest opportunity. To provide these extra security precautions, your change-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 configuration after the change is no longer needed, ensuring that an administrator doesn't forget to make the change.

A change-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 change-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.

Q 2.3: 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. WEP is a *must* 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.

However, the very idea that passwords might be transmitted in the open gives me the willies 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 (WAP), 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 designed for network management.

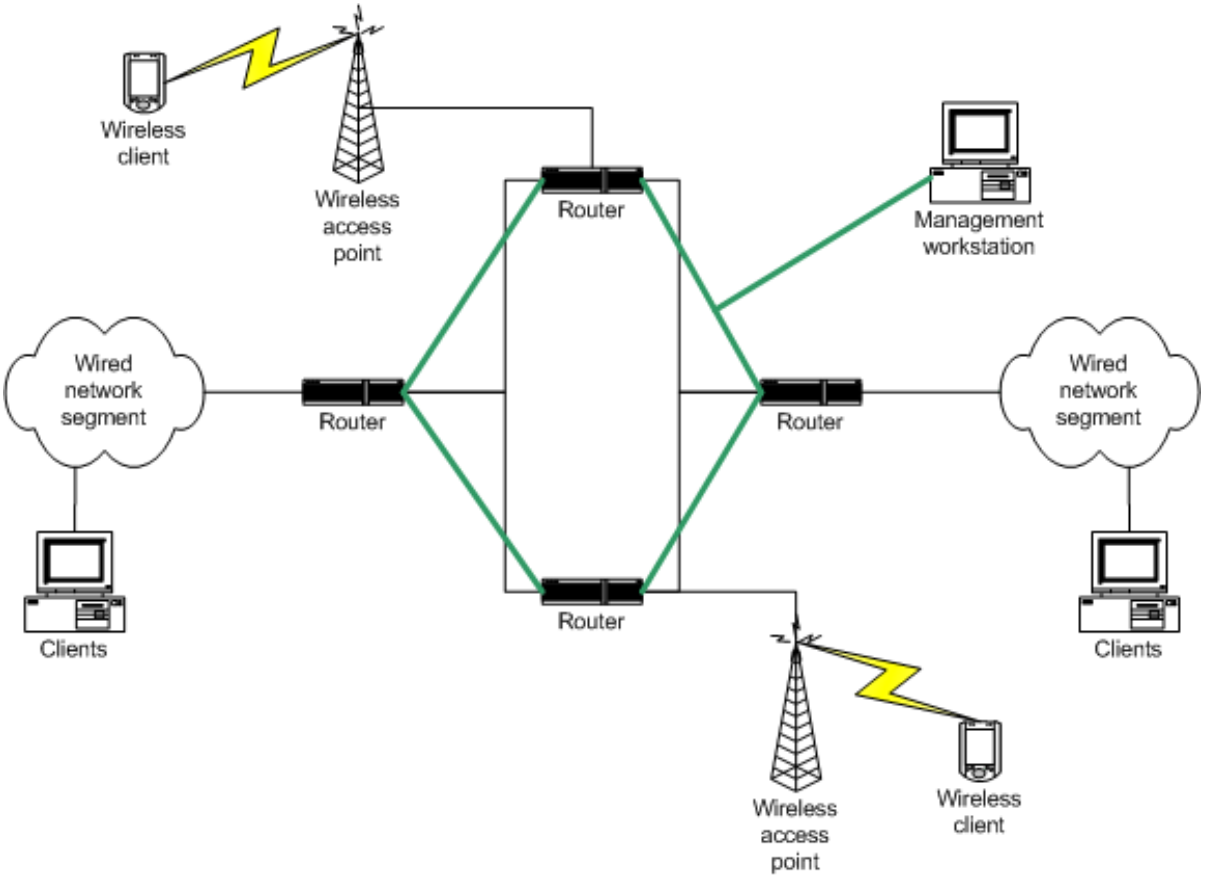


Figure 2.1: Dedicating a network to network device management.

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 change-management process in place. Change-management software that can automatically archive device configuration backups will let you recover more quickly in the event that your wireless network is hacked and used to upload inaccurate configuration information to a network device.

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

Q 2.4: 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 change-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's 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.

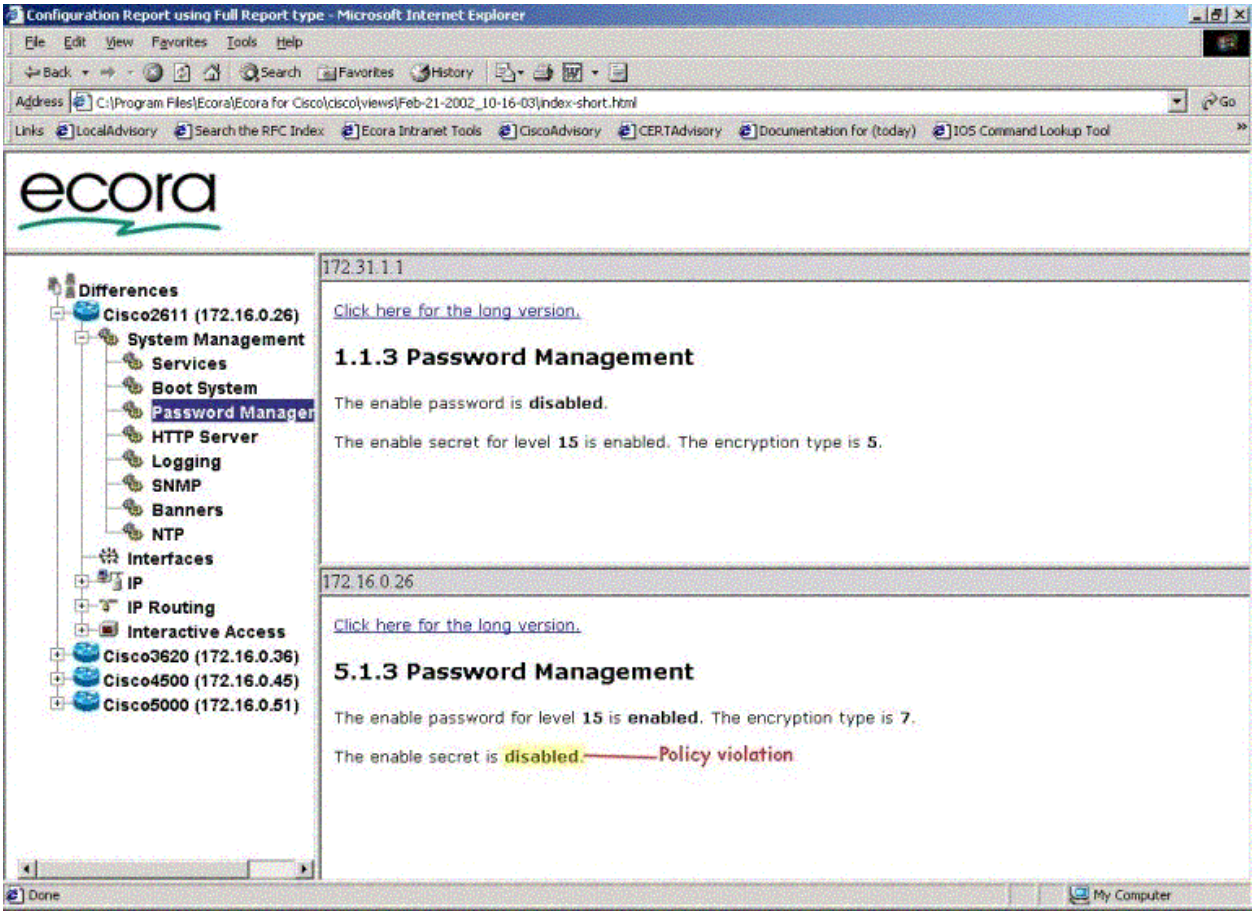


Figure 2.2: Using software to automatically audit network devices can turn up common security vulnerabilities.

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 you can expect to see other software manufacturers, including AlterPoint, offering more broad-based support of this kind in the near future.

> ☞ 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 change-management process so that you can minimize the risks involved with modifying your network devices.

Finally, don't underestimate the ability of a thorough change-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 2.5: 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 hacker 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 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 eight 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.
- Use different passwords for each network device. That way, one compromised password doesn't open the gates 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 then 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's 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) such as Microsoft Windows XP Professional to 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.

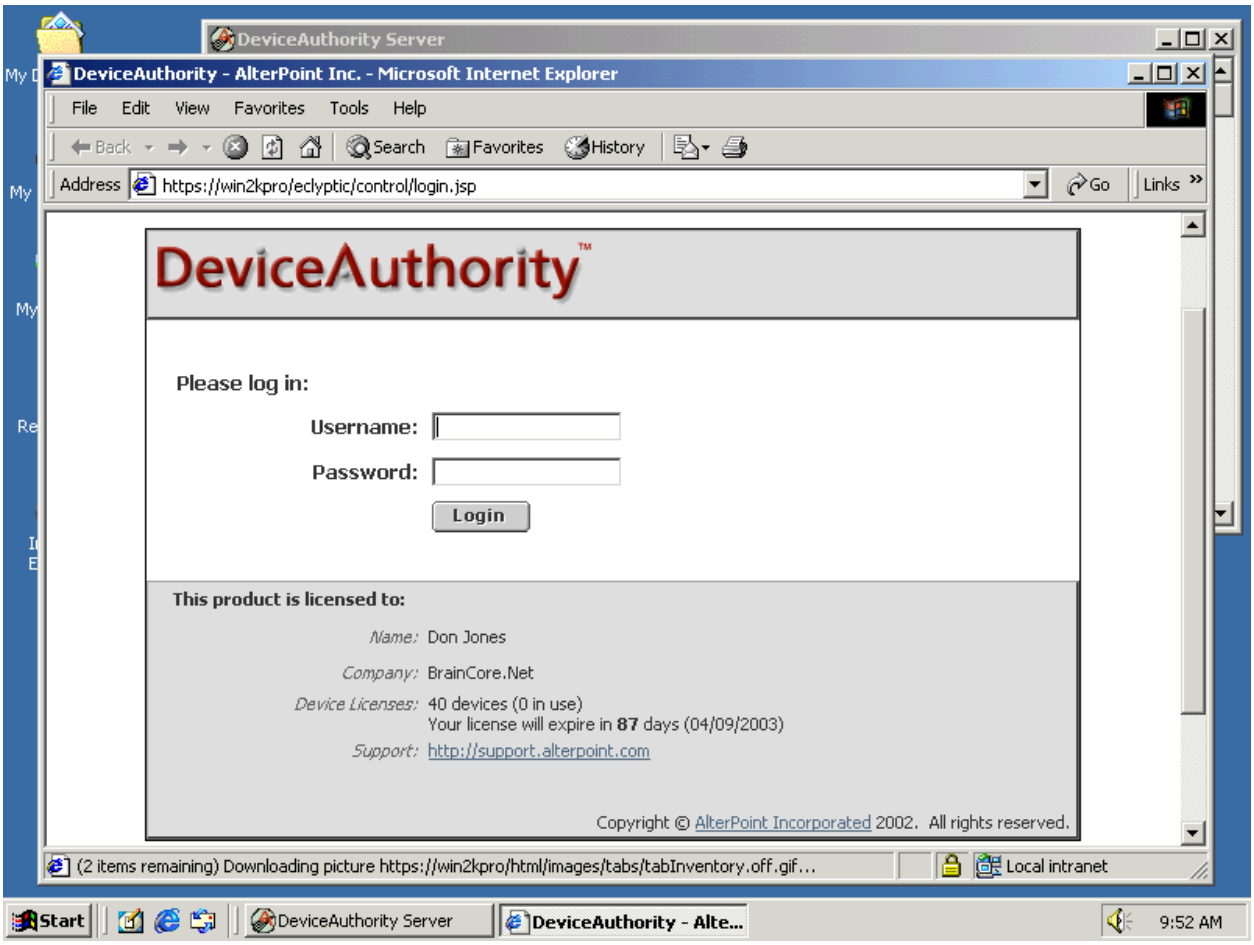


Figure 2.3: DeviceAuthority uses encryption to protect your logon credentials.

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 2.6: Per-device passwords don't seem to be very secure. What alternatives can I use?

A: In several tips, I've mentioned that the use of per-device passwords to secure routers, switches, and other managed devices doesn't provide the best security. Too often, too many individuals need to know the passwords, providing no accountability for individuals' actions, no easy way to routinely change passwords, and too many opportunities for critical passwords to become compromised. Fortunately, newer network devices often provide an alternative in the Remote Authentication Dial-In User Service (RADIUS).

> Many devices, notably Cisco routers, also provide support for Terminal Access Controller Access Control System+ (TACACS+), a somewhat more advanced authentication service. Because TACACS+ is loosely based upon RADIUS, at least conceptually, I'll focus on RADIUS in this tip and cover TACACS+ later.

How RADIUS Works

Request for Comments (RFC) 2865 specifies how RADIUS authentication works. Note that a companion RFC, RFC 2866, specifies RADIUS *accounting*. RADIUS accounting is designed to provide a standardized method of logging user activity. However, accounting isn't generally supported by a wide range of network devices, so I'll focus on RADIUS authentication.

RADIUS was originally designed as a means for standalone dial-up servers to authenticate dial-up users against a central database. Prior to RADIUS, these standalone servers had to maintain independent lists of users, passwords, and user dial-up permissions (who was allowed to dial in at what time and so forth). Large companies with dozens of dial-up routers spent a lot of time keeping their individual user lists in sync. The idea behind RADIUS was that the individual servers would simply pass authentication information to a central server. The server would then tell the device whether to allow the authenticating user to remain online. Figure 2.4 shows the basic structure for RADIUS.

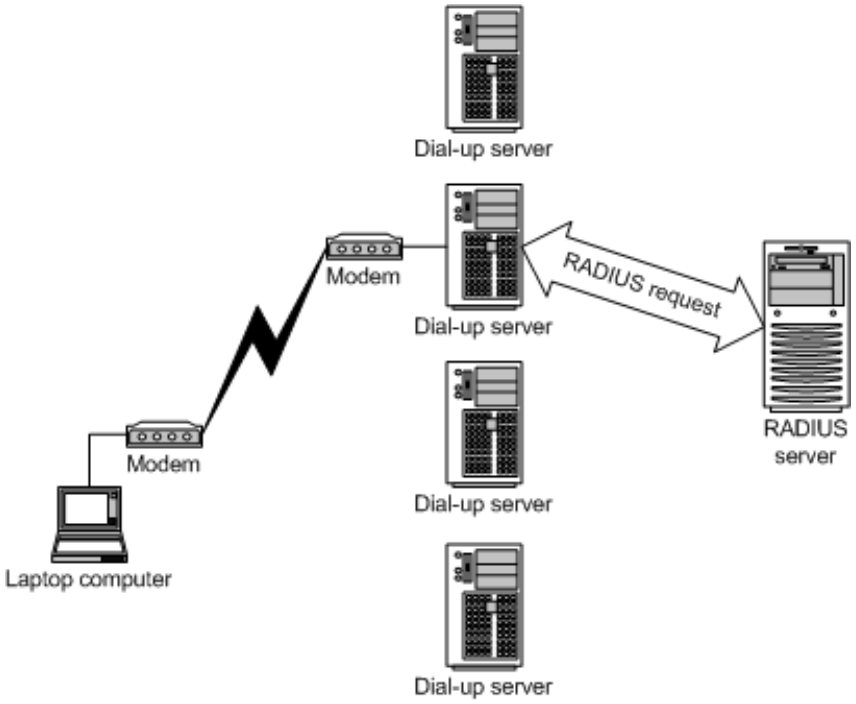


Figure 2.4: RADIUS provides a central authentication database for multiple independent dial-up servers.

In this example, the dial-up server is a RADIUS *client*, meaning it communicates directly with the RADIUS server. Because RADIUS servers are evaluating authentication information (such as usernames and passwords), they represent a potential security hole. After all, if a hacker can get a RADIUS server to give it a "thumbs up" on a username and password pair, the hacker knows he or she has a valid set of credentials. For this reason, RADIUS servers generally won't communicate with anyone that hasn't been specifically configured as a client. RADIUS clients share a password (often called a *shared secret* or *key*) with the client, allowing the client to log on and present authentication requests.

RADIUS also provides *authorization.* Authentication simply tells the RADIUS client that a given set of logon credentials are valid; authorization tells the client what the user is allowed to do. In a dial-up scenario, authorization might include a list of allowed protocols. In a network device situation, this authorization information might include a user's level of privilege within the device's command interface.

RADIUS in Network Devices

For starters, recall that most network devices have various levels of access. Specific configuration commands are assigned to different levels, depending on the sensitivity of the command. For example, Cisco routers support 15 levels of privilege, and, by default, commands are assigned to one of three privilege levels: Level 1 for basic commands, Level 15 for most configuration commands, and Level 0 for a few non-sensitive commands. Devices from other manufacturers use a similar privilege hierarchy.

Sticking with Cisco as an example, you can configure the router to pass authentication requests to a RADIUS server. You'll need something like the following configuration in the router config file:

```
aaa new-model
aaa authentication login default radius local
aaa authorization exec radius local
username backup privilege 7 password 0 backup
radius-server host 192.168.0.12
radius-server key cisco
```

This configuration sets the router to use RADIUS authentication, and establishes that the username "backup" will be assigned Level 7 privilege. Presumably, you would also reassign some router commands from Level 1 to Level 7, making this privilege level useful and a means of controlling user capabilities.

This configuration also specifies the RADIUS server address 192.168.0.12 as well as the shared secret that the router and RADIUS server will share: cisco, in this case. Of course, the RADIUS server will need to be configured to have a matching key, and you should definitely use a more complex key that is less easy to guess.

> Versions differ! The previous configuration example works with Cisco IOS 11.2; later versions, and configurations for other manufacturers, will be similar. Consult your devices' manuals for more information about configuring them to use RADIUS authentication.

Essentially, then, users attempting to access this router will provide a username and password. The router will make no attempt to validate these credentials, instead passing them along to the RADIUS server. The RADIUS server can return with an accept or reject response, depending on whether the credentials are valid. If accepted, the router can provide the user with the appropriate level of privilege within the command interface.

Configuring a RADIUS Server

There are several freeware UNIX-based RADIUS daemons, and Cisco offers a RADIUS server called CiscoSecure. However, for this example, I've chosen to use Microsoft Internet Authentication Service (IAS), which is included in Windows 2000 (Win2K) Server and Windows Server 2003. IAS makes it fairly easy to use Active Directory (AD) domain user accounts for router authentication, which provides a handy, single-account means for managing both network devices and AD or Windows server resources.

> The cool part about RADIUS is that it is truly cross-platform. Cisco routers are perfectly happy to pass their authentication requests to a Microsoft RADIUS server or, for that manner, any compliant RADIUS server you might have in your environment. This cross-platform capability allows you to select a RADIUS solution that best meets your authentication needs, such as one that works with your organization's directory services infrastructure.

IAS is an optional component of Windows, which you'll need to install using the Add/Remove Programs icon from the Control Panel. You'll also need your original Windows product CD-ROM or a network copy of the CD-ROM files, along with any recent service pack CD-ROMs. Installation of IAS is pretty basic: Just select the appropriate check box and let Windows copy files. You'll be able to immediately configure the service once it's installed. Simply launch the Internet Authentication Service console, located in the Administrative Tools folder. Figure 2.5 shows the basic console after a RADIUS client has been configured.

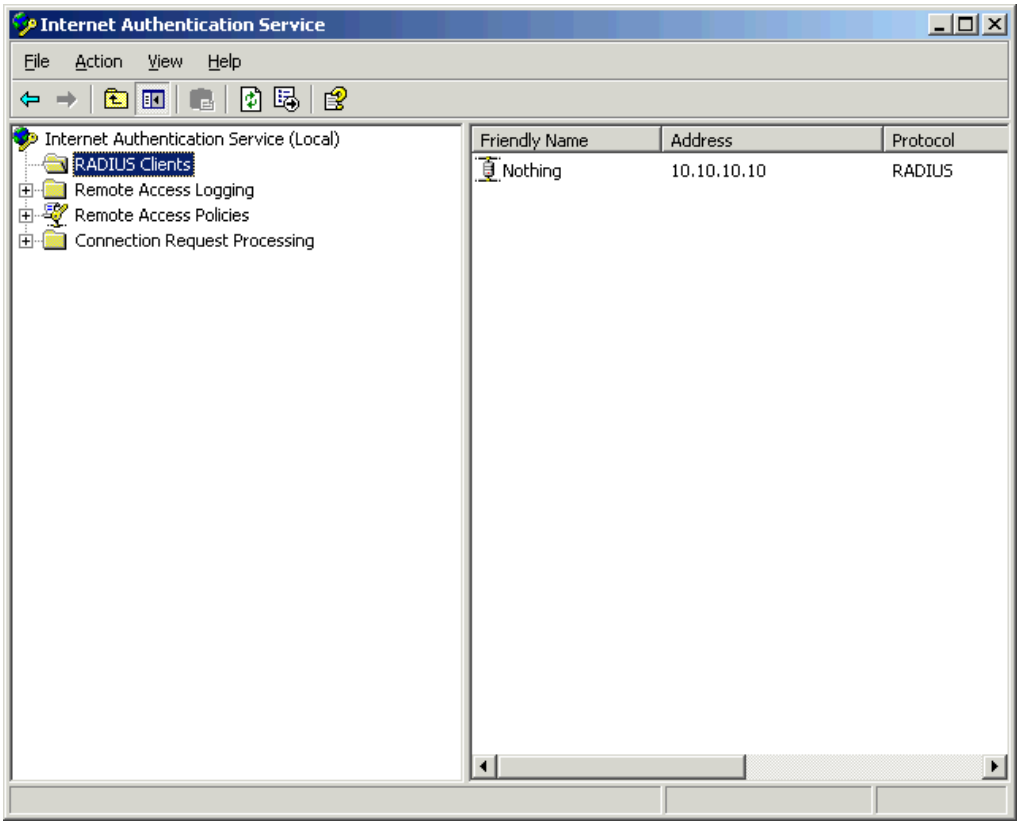


Figure 2.5: The Microsoft IAS console, showing the current allowed RADIUS clients.

To add a network device as a RADIUS client, right-click RADIUS Clients in the console, and select New RADIUS Client from the pop-up menu. You'll provide a name for the client along with its IP address. The IP address is the only important information; IAS doesn't do a reverse DNS lookup to ensure that the device has the same DNS name as the one you've provided.

As Figure 2.6 shows, you'll also specify the type of client and a key for it. Although you can specify Generic RADIUS Client and get perfect results, specifying the devices' vendor—if listed—provides a slightly higher degree of security. IAS will only accept packets from the client's IP address if the client has the right key *and* presents the correct vendor ID code in the RADIUS request packet.

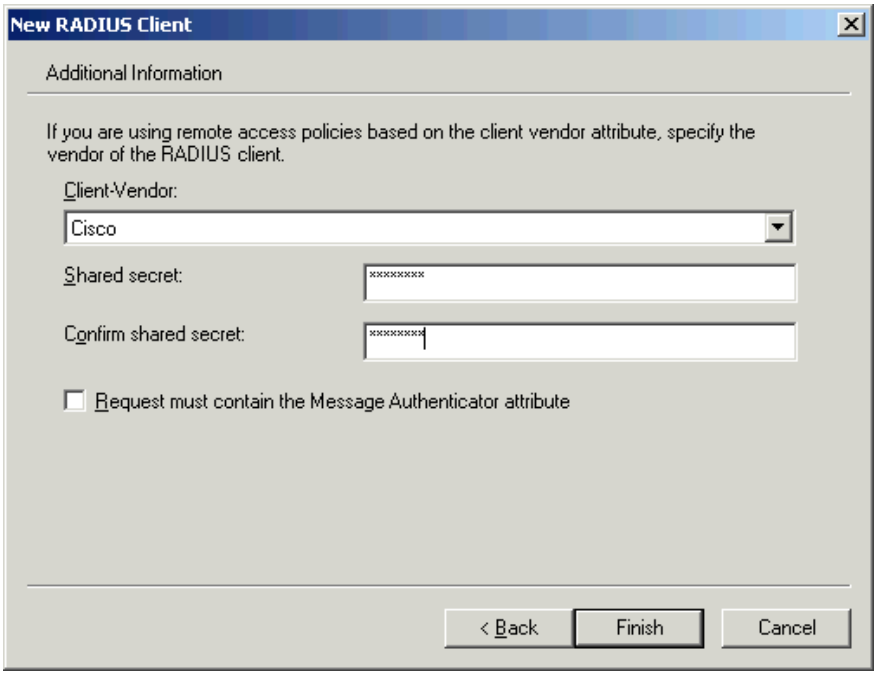


Figure 2.6: Specifying the client type and key.

Once you're finished setting up the client information in IAS, your network device can start passing RADIUS requests to IAS for authentication. The main benefit is that your network device users will authenticate using RADIUS, which in turn can be based upon a very secure directory services architecture, such as AD. You won't need to rely on less-secure per-device passwords; you'll be able to centrally control device authentication via RADIUS, and your devices will be more secure.

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.

📖 See Question 2.6 for more information about RADIUS.

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.

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 change 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 change management solution. Once the change 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.

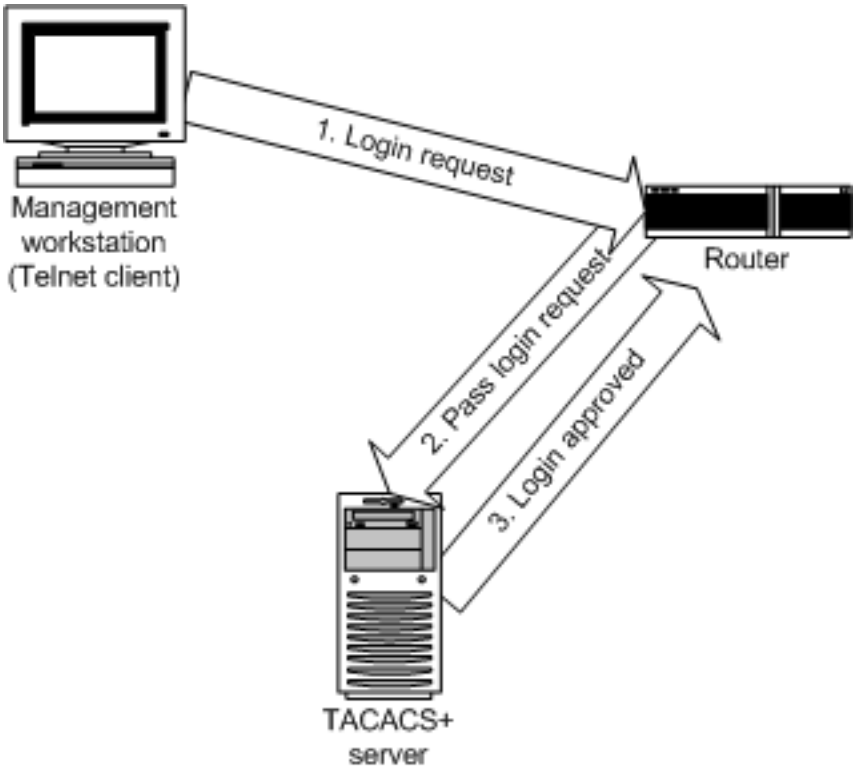


Figure 2.7: Communications between a Telnet client, Cisco router, and a TACACS+ server.

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 is 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.

Q 2.8: What is the best way to ensure that our network devices are secured against outside attack?

A: In the old Westerns, the bad guys wore black and the good guys wore white. The terminology persists in the world of network security, where *black hats* are individuals bent on cracking your network security for their own nefarious purposes, while the *white hats* try to break in so that you can see your vulnerabilities and fix them. More formally referred to as *penetration testing* or *vulnerability assessments*, this "good guy hacking" is the best way to make sure that your network devices—and other portions of your network—are secure.

Vulnerability assessments should be conducted by a trained, skilled professional who ideally has very little idea of what your network looks like. That way, the good guy hacker is starting on the same playing field as a bad guy hacker would be on. Consulting firms across the world offer vulnerability assessments for various fees and are your best resource. You might want to conduct yearly assessments or rely on less periodic assessments as a formal means of enhancing your internal security tests.

> ☞ Get credentials! The only difference between a black hat and a white hat is intent. Make sure that your hired gun can provide customer references and credentials that unquestionably state their intent. They're likely to be exposed to confidential information, passwords, and more, so make sure that you trust them not to use this information outside the assessment test. Also get them to sign the appropriate nondisclosure agreements, make sure they're carrying liability and intellectual property insurance, and have an attorney review their consulting contract.

What Will Be Tested?

Which parts of your network the white hat tests is pretty much up to you; however, you should definitely consider, at a minimum, testing the following:

- Your policies—These need to be reviewed by an experienced security professional (not necessarily the white hat who's doing intrusion testing) who can spot procedural flaws that might leave your network devices open to intrusion.
- Basic port scans—Most network devices are pretty locked down out of the box, but over time, ports can be opened for various reasons. Scanning for these ports can show you exactly where an attacker will start their efforts, particularly on Internet-connected devices. However, don't be tempted to ignore vulnerabilities on internal devices. Firewalls aren't invulnerable, and many corporate attacks come from *inside* the firewall. Internal devices should be as secure as possible.
- Restrictions—This area is where interpretation and experience comes in. Yes, you've got TFTP open on a server because you need to use it to dump your network devices' configurations on a regular basis. Fine, but have you configured the TFTP server to only accept connections from those network devices? This sort of configuration fine-tuning can make your environment a lot less susceptible to attack, and an experienced white hat should spot these situations and offer suggestions.

C'mon, Can't I Do This Myself?

Security testing isn't especially difficult. A number of readily available tools, including Network Associates' CyberCop ASaP and Internet Scanner, are used by most security consultants and can be just as easily used by your team. However, the trick isn't in running the tools, it's in figuring out what the tools are telling you. You might overlook the open HTTP port on your firewall because you know you have a Web server inside; a security consultant might realize that the port is a direct pass-through and use it to initiate an attack against the server. Their recommendation might be to use a reverse proxy rather than a pass-through to try to minimize the attack methodologies available to a hacker.

Consultants usually—or at least, should—show up with multiple tools. Very few tools catch everything, and running several different ones is the best way to find all the vulnerabilities you've got. You could certainly run these tools yourself, but there comes a point at which the expense of licensing all the software becomes more than simply hiring someone who's already got it all. There's also the know-how that a consultant should bring to the table, which often represents years of specialized experience that's very difficult to duplicate, no matter how many tools you buy.

You can generally specify a broad scope for security testing as a starting point. For example, most outside firms will perform a *remote penetration test,* which simply duplicates the efforts a true hacker would take, using no inside knowledge of your network. Many businesses prefer this type of test, and prefer to handle inside testing themselves, as it tends to expose less confidential information. You can also select additional categories for testing:

- Internal tests, which involve onsite testers using tools to check your internal resources. These tests might include a check for the latest device operating system (OS) patches and firmware updates, port scans, and so forth.
- Social testing, which interviews users about their attitude toward security. This type of test is helpful during a full security review, but usually isn't required if you're just testing the security of your infrastructure devices.
- Device-specific tests; for example, load-balancing and remote access devices have unique vulnerabilities, and an experienced firm can test your devices to determine whether they're secured against common forms of device-specific attacks. Remote access devices, for example, might be subjected to a common *wardialing* attack to ensure that only authorized users can gain access.

Figure 2.8 illustrates the different attack points and targets that you might consider.

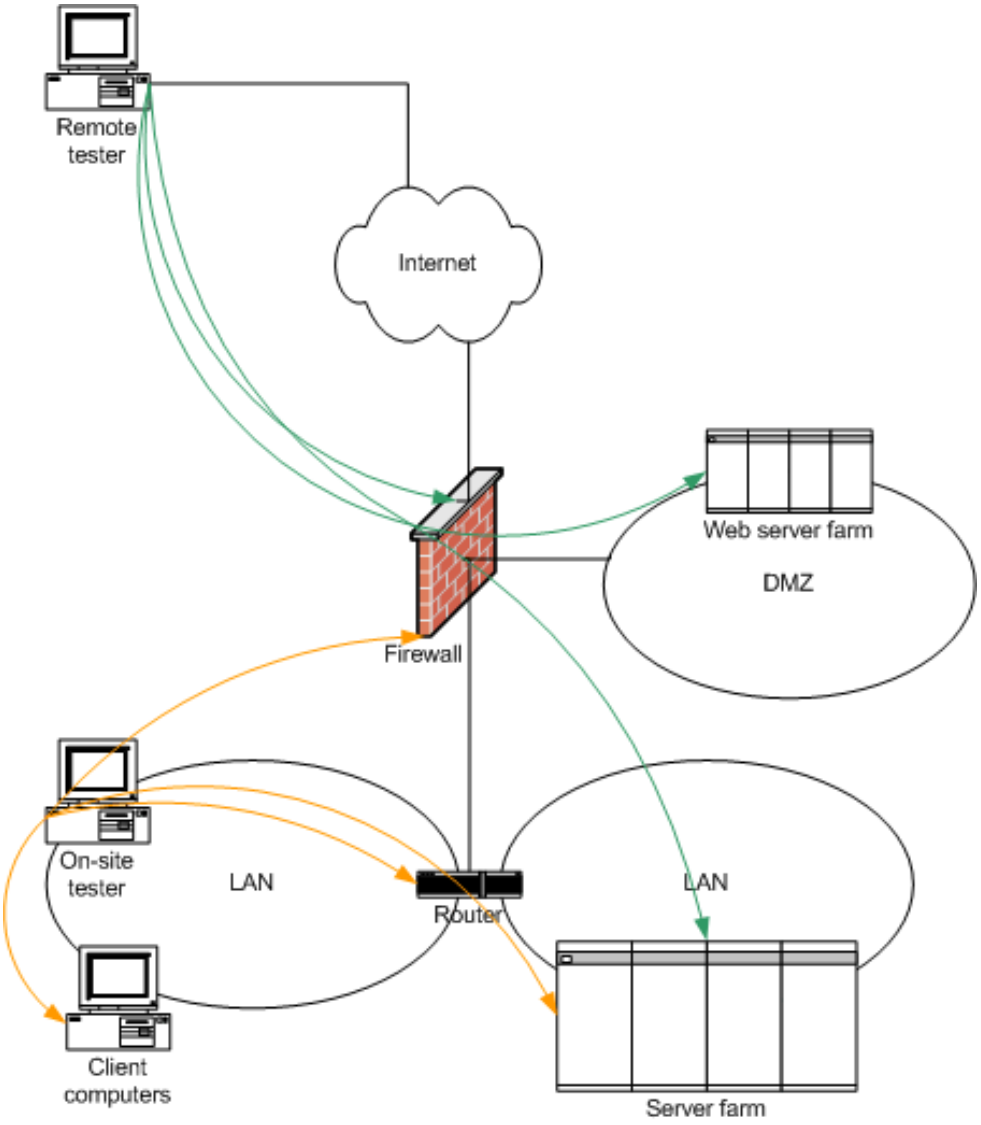


Figure 2.8: Targets for external and internal security testing.

What Are the Drawbacks?

Security testing can definitely be taken too far. You probably don't want your network tested to the failure point. However, if you do want to test to the breaking point, you're going to obviously have to schedule some off-hours time when that will be permissible.

You might also want to limit or ignore certain types of attacks in the test. For example, denial-of-service (DoS) attacks are a common way to bring a network to its knees (or at least specific servers and devices). Testing to see whether your network is vulnerable to this type of attack is pointless; trust me when I tell you that *all* networks can be affected by a DoS attack. You might simply want to take it as a given and move on to testing less brute-force attacks, especially when you consider that a simulated DoS attack can be just as devastating as a "real" attack. If DoS attacks are an existing or anticipated problem, you can do some testing to see whether the scale of your network can survive against a specific size of DoS attack (say, one launched by ten attacking computers), but no degree of this so-called *hardening* will make your network invulnerable to every possible DoS attack.

The biggest drawback, which I've already mentioned, is that the tester will almost certainly be exposed to confidential information or, at the very least, captured user passwords. Plan to have someone from your own team work closely with the testers at all times to monitor the information revealed and ensure that it is treated appropriately—particularly when an outside firm is conducting internally based assessments. Reputable firms, such as En Garde Systems and eSecurityOnline, can help you plan your testing before they begin and will provide you with detailed reports as well as recommendations for fixing any security problems they find.

Are There any Assessment Resources out There?

The Open-Source Security Testing Methodology Manual (OSSTMM) is a useful assessment resource that you can access online at http://www.isecom.org/projects/osstmm.htm. This manual provides a methodology for network testing that is primarily designed to be thorough, and it outlines the requirements that you should look for in a white hat consultant. The manual is hosted by the Institute for Security and Open Methodologies (http://www.isecom.com), which also offers security training, OSSTMM certification, and more.

The SANS Institute (http://www.sans.org) is another valuable resource. Specifically, check out http://www.giac.org/practical/gsec/Adrien_Beaupre_GSEC.pdf—an article about vulnerability assessments.

Q 2.9: How can we ensure a consistent security configuration on our devices?

A: In prior tips, I've pointed out how a consistent configuration on similar devices is critical for maintaining a high level of security. By developing a known-good configuration for your devices—one with the proper routes, passwords, security levels, and other information—you can more effectively secure those devices against attack. Developing a consistent security configuration is a simple task—the difficult aspect of this process is actually deploying the configuration to multiple devices.

To manually back up your devices' configurations by using a TFTP server, you simply log on to each device and instruct it to TFTP its configuration files onto the TFTP server. You can use a variation of this technique—simply performed in reverse—to upload a particular configuration to each device on your network. For example, to force a Cisco router to write its current configuration to a TFTP server:

```
ciscorouter> enable
ciscorouter#write net
Remote host [192.168.1.100]? 192.168.1.100
Name of configuration file to write [ciscorouter-config]?
ciscorouter-config
Write file ciscorouter-config on host 192.168.1.100? [confirm] y
ciscorouter# exit
```

The following commands are used to erase the current configuration and load a new configuration from a TFTP server:

```
ciscorouter> enable
Password: ********
ciscorouter# write erase
ciscorouter# copy tftp:file-name startup-config
ciscorouter# reload
```

A somewhat less-drastic version simply copies over a configuration file from the TFTP server without erasing the NVRAM first:

```
ciscorouter> (enable) copy tftp config
IP address or name of remote host []? 192.168.1.100
Name of file to copy to []? config1.txt
Configure using tftp:config1.txt, (y/n) [n]? y
Console> (enable)
```

This method of deployment is easy to perform—with the appropriate commands—on nearly any brand of router. However, an automated process would be even easier. An inexpensive means of doing so is the Solarwinds Config Uploader, available from http://www.solarwinds.net. This small piece of Windows-based software is capable of sending Simple Network Management Protocol (SNMP) commands to your Cisco routers, instructing them to retrieve a specific configuration file from a TFTP server on your network. Figure 2.9 shows the software's primary dialog box in which you specify the device to update, the TFTP server on which to locate the update, and the actual file that the device is to upload. You must also specify the SNMP community string used by the device so that the software can connect to it via SNMP and issue commands.

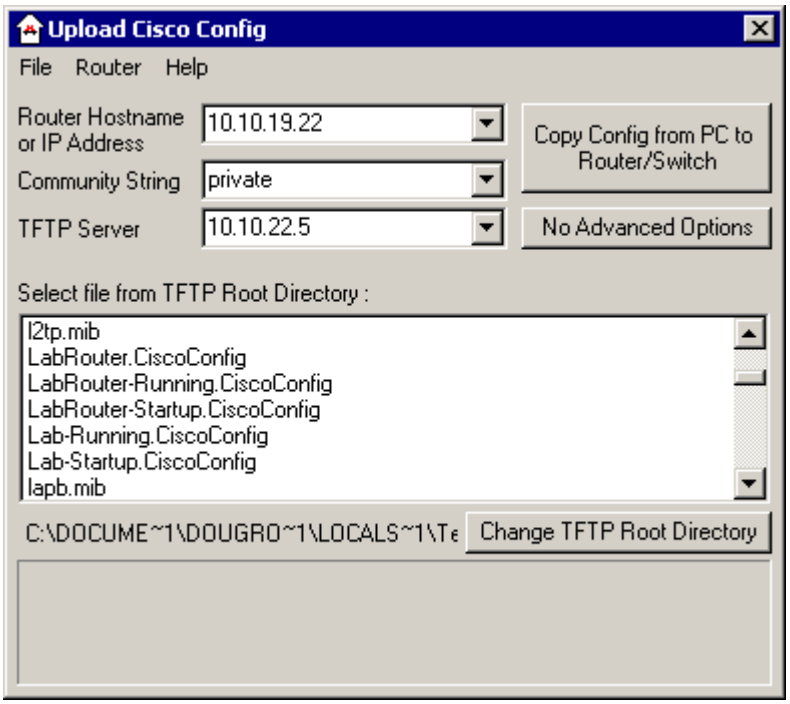


Figure 2.9: Selecting a configuration file to upload to a router.

Keep in mind that SNMP communication between the tool and the device is required; the device must also be able to access the TFTP server. Figure 2.10 illustrates the required communications.

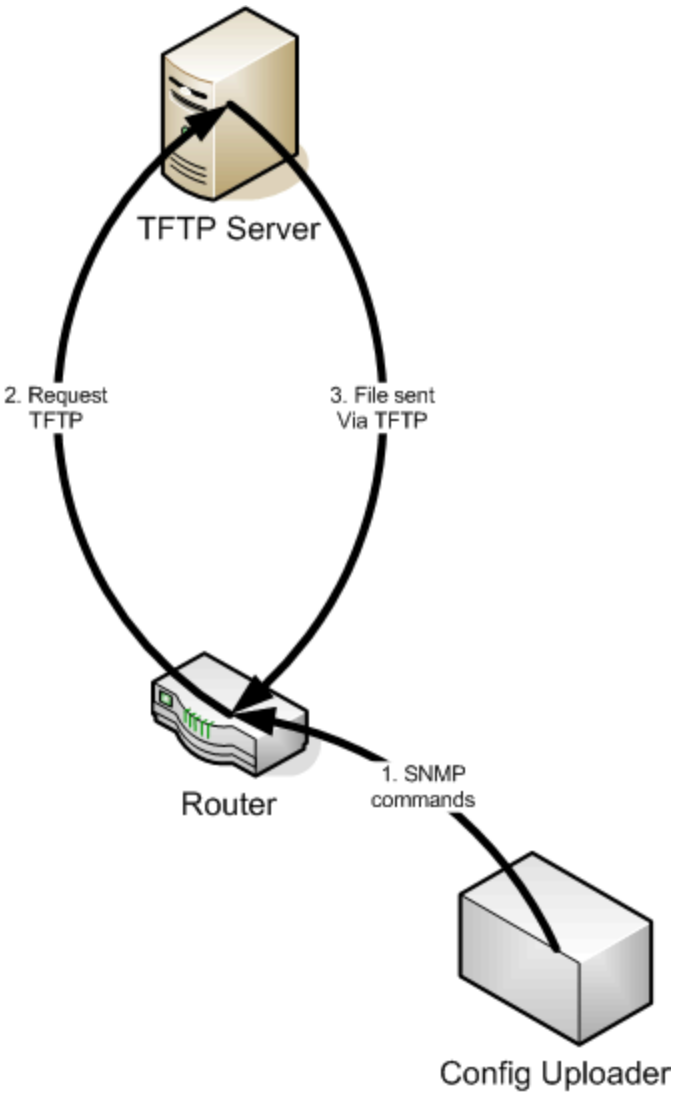


Figure 2.10: The Config Uploader uses SNMP to instruct the router to obtain a new configuration.

Unfortunately, this method has a few drawbacks:

- It only works with Cisco devices.
- It only works with one device at a time.
- It only works immediately—meaning you can't schedule an update to occur during off hours.

Alternative software addresses these problems. For example, AlterPoint DeviceAuthority uses a similar technique to upload configuration files to your devices, but allows you to push a configuration out to multiple devices at once. DeviceAuthority ensures that it can contact each device you've selected by using a ping command. It also backs up all of the devices, as Figure 2.11 shows, allowing you to more easily roll back to a known working version in case your update doesn't go as smoothly as you had hoped. You can use this product not only to roll out new configuration files but also entirely new software versions.

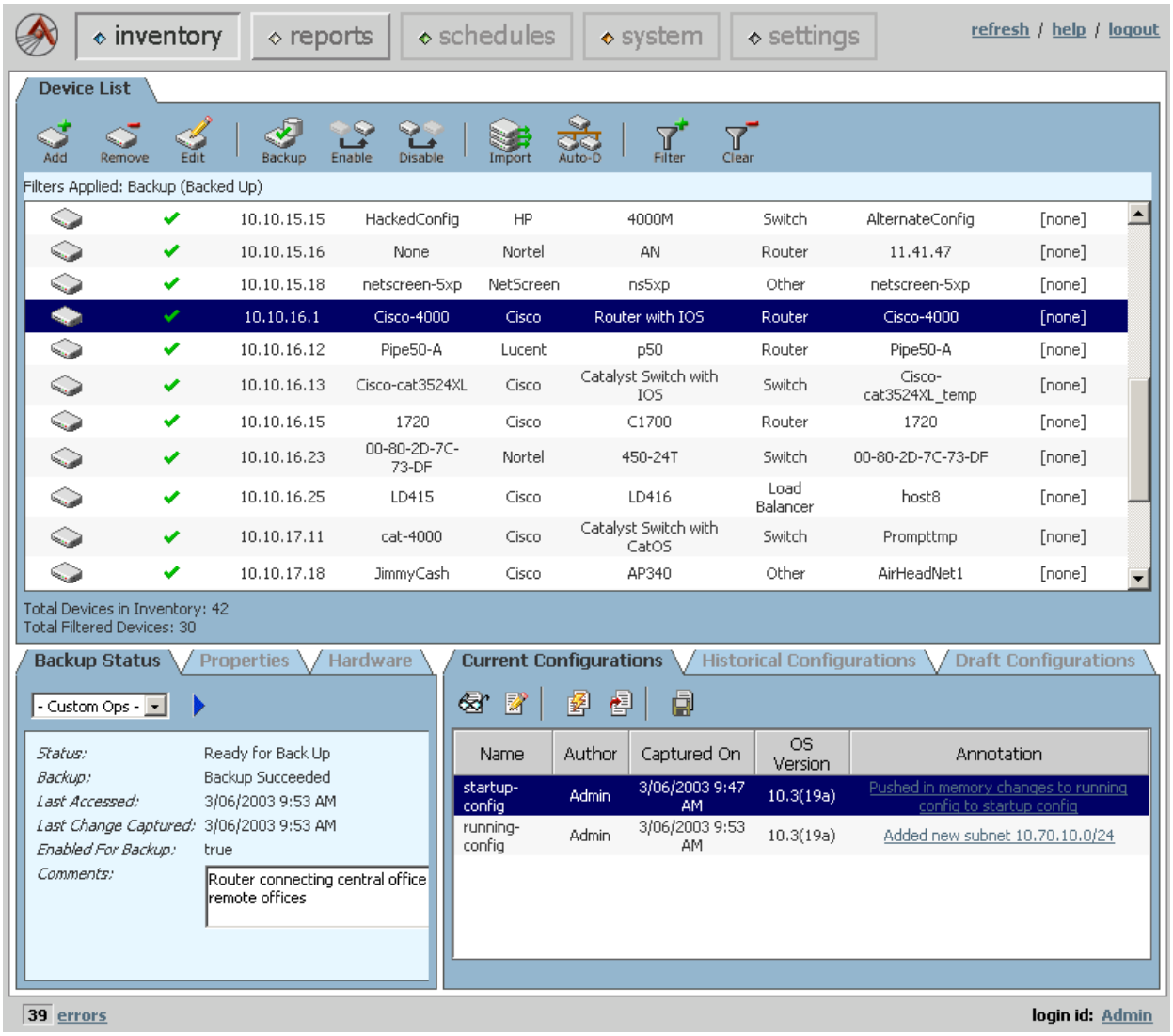


Figure 2.11: DeviceAuthority provides automated backup of all device configurations.

Q 2.10: What is the best method for quickly deploying a security patch to devices?

A: Security patches can be difficult to deploy to any number of devices—the complexity of the task depends upon the patch. Unfortunately, the difficulty in deploying patches means that many administrators simply don't do so. As networks become more hostile (with continual virus attacks, malicious users, and so on), the need to quickly deploy security patches is becoming even more critical.

The first step in a security patch deployment is to test the patch. You should do so as soon as you discover the release of a patch. Ideally, you have access to a spare device that you can use for patch testing. Realistically, the expense of maintaining spare devices is pretty high, so you might need to simply deploy the patch to one device on your production network as a test. Be sure to use a change-management process and tool so that you can quickly return the device to a functioning state in the event that the patch causes a problem.

> ☞ Stay informed! Sign up for vendor-provided patch-notification services. Most vendors—including Cisco, Bay, 3Com, and others—provide email subscription services that will notify you when device patches become available.

After you've tested the patch, you can begin the patch deployment. Vendors typically rely on fairly primitive deployment techniques for device patches, often using FTP or TFTP to load new software into the device, then requiring the device to restart or reload in order for the updated software to take effect. Such deployments methods are a far cry from the more sophisticated patch management provided with operating systems OSs such as Windows that offer built-in software to look for, download, and install patches from a centralized location on the corporate network.

To ease the pain of deploying software updates to multiple devices, you can write a deployment script. For example, you might write scripts that run on each device on your network, load the patch, restart the device, and perform any other necessary actions. Writing the script can be time-consuming, and you will still need to manually run the script on each device that needs to be updated.

Software tools provide a better solution. For example, AlterPoint DeviceAuthority and Voyence VoyenceControl both offer the ability to target multiple devices on your network and automatically deploy a software update to those devices. These tools present an efficient, logical way of managing security patches on any network that supports more than a handful of devices. The software keeps track of each device, its current software version, and other details. The software can then accurately target the devices that require an update, deploy the update, and monitor the devices to ensure that the update was properly deployed.

> ☞ Get an all-in-one solution. There are a few change-management solutions that don't offer software deployment and device-updating capabilities; there are also solutions that strictly focus on sending out updates to multiple devices and deploying software updates. Look for a change-management solution that incorporates update deployment, which will make the process more efficient.

Your change-management process should incorporate prioritization so that security patches—which can prevent security breaches and operational problems—take a higher priority than other software updates, which might simply fix a few bugs. Security patches from most device vendors should be deployed as quickly as possible, and a deployment solution or tool can help make this deployment fairly easy.

When the Cure Precedes the Problem

Why is it so critical that security patches (or any patch that addresses a security vulnerability) be deployed as quickly as possible? Often because the mere presence of the patch will *ensure* that the vulnerability is exploited by attackers.

Remember the Blaster worm that affected Microsoft Windows-based computers? Although the worm didn't attack devices such as routers or switches, it was an excellent example of the cure preceding the problem. In that case, Microsoft was notified of a vulnerability in the remote procedure call (RPC) component. The notification was quietly provided by a security consulting firm that found the bug; there was no existing exploit of the problem.

Microsoft then released a patch that corrected the problem. *Weeks* later, the Blaster worm began to circulate. The worm's authors downloaded the Microsoft patch, analyzed the differences in the patched files, and reverse-assembled the patched code. They used the patch to construct the worm, knowing that many users wouldn't apply the patch. As a result, the worm was tremendously successful.

Device OS updates and patches are just as easy to reverse-engineer (if not easier, in some cases), meaning the mere availability of a patch can serve as a set of guidelines to attackers creating exploits. In other words, when you hear about a security-related patch for your devices, apply it *immediately* (after testing it, of course, to see if it causes any operational problems). Even if you're not experiencing the problem that the patch fixes, you probably will once the patch is published and attackers use it as a roadmap to create exploits.

Topic 3: Network Management Troubleshooting

Q 3.1: 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've eliminated some kind of 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. That's 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 figure out 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're in the habit of making a raft of changes at once, you'll have a much more difficult time tracking down the change that caused the problem.

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 see which ones might apply to the problem you're having. If necessary, manually undo each change, one at a time, until the problem goes away.

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 rollback 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 out the router's interfaces to see what they're plugged into.

There's 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, 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 speedier 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'll 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.

Q 3.2: How can change 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 change-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 you have to manage them 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 change-management process in place. The change-management process provides some important capabilities:

- Change management provides a logical checkpoint, allowing you the opportunity to take performance measurements before and after a discreet set of changes.
- Change management provides a history, enabling you to compare before and after configurations and relate them to measured performance changes.
- Change 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.

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 3.3: 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 states 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 change-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 3.4: 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 higher-end 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 4.2). 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 4.2. It's available from Component Software (http://www.componentsoftware.com) and makes it much easier to spot changes between versions of a text file. Best of all, it's a free tool. Figure 3.1 shows how CSDiff highlights the differences between two text files.

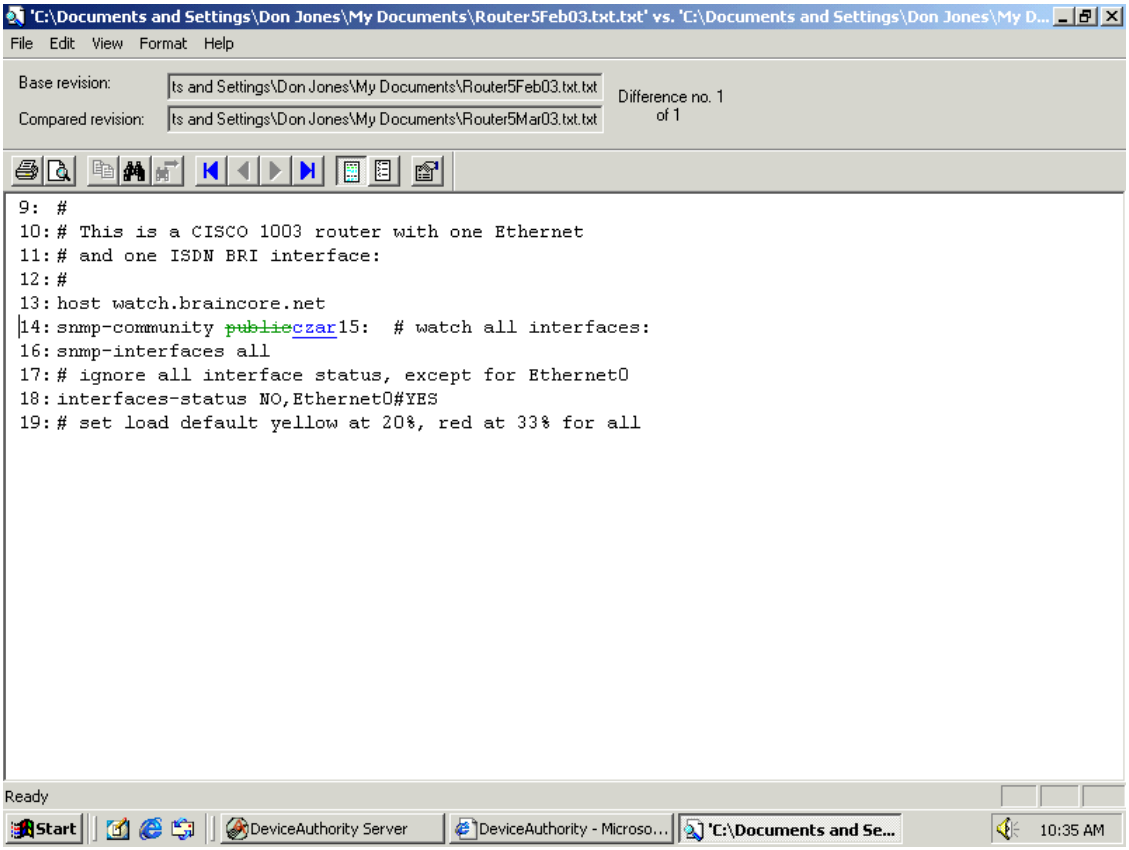


Figure 3.1: Using CSDiff to analyze the differences in a router configuration file

Unfortunately, watching for changes manually is a lot of work. You have to 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 they 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.

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 3.2 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.

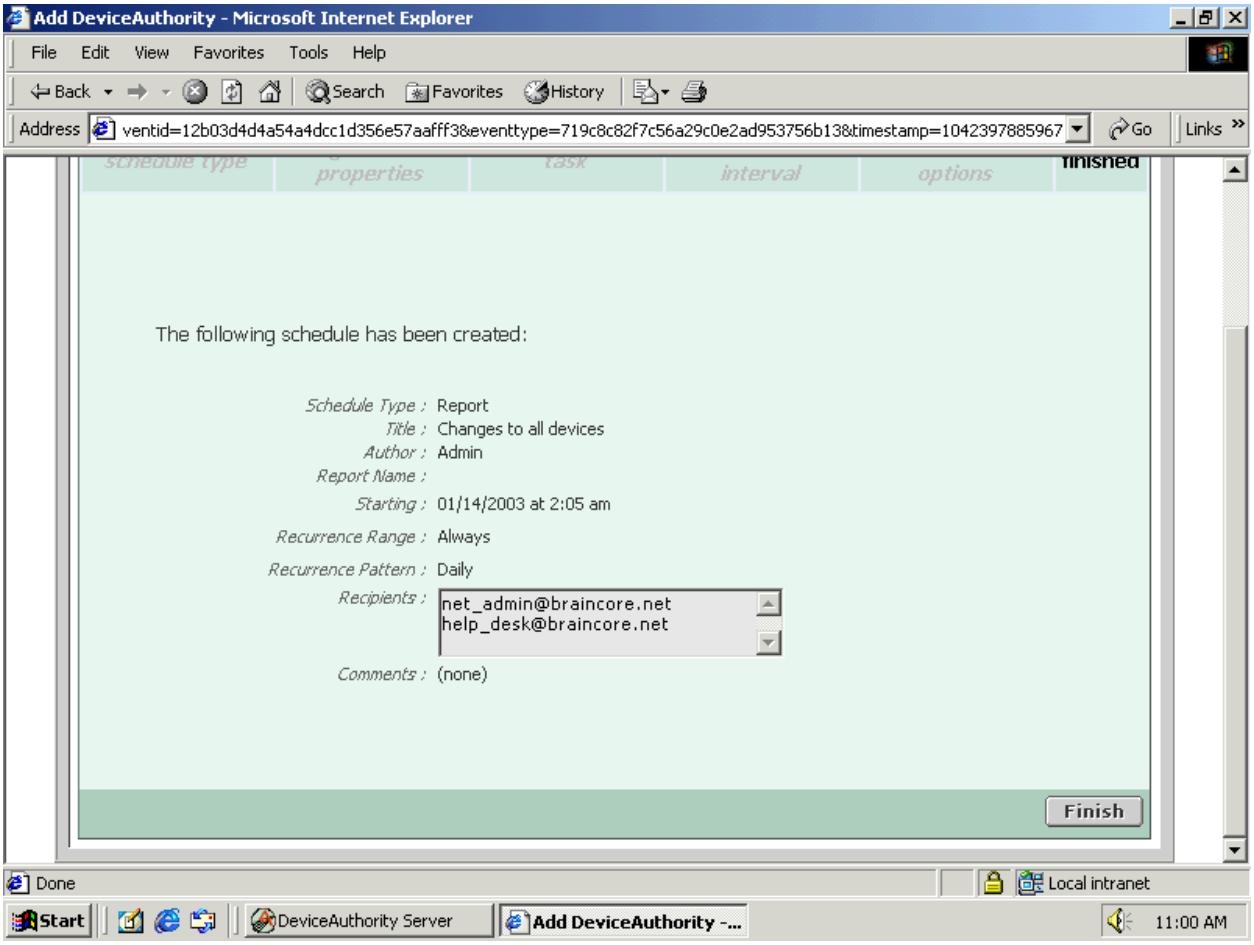


Figure 3.2: Creating a daily schedule keeps you on top of unexpected device changes and is a useful tool for auditing your change management process.

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 completely 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 hard 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'd 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.mkssoftware.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 3.5: Troubleshooting network devices is complicated. Is there a general framework that can make it easier?

A: There's no industry-standard framework to make network device troubleshooting easier, but there are several resources that can help you develop a framework that works in your environment:

- Cisco provides a detailed Internetwork Troubleshooting Guide at http://www.cisco.com/univercd/cc/td/doc/cisintwk/itg_v1/index.htm. This guide provides troubleshooting steps for just about every aspect of network troubleshooting.
- I often use the links at http://www.teklnk.com/links.htm to find troubleshooting resources. There's a wealth of tips, tools, and concepts for Cisco, Nortel, and a variety of other vendors.

As I've mentioned in previous tips, the best place to start troubleshooting network devices is to look at what has recently changed. You can usually trace most device problems to a recent configuration change that's not working out as well as you'd hoped; network change management software or even simple text file comparisons of device configurations can help highlight recent changes and let you quickly focus your troubleshooting efforts.

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/pcgi-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/pcgi-bin/tsa7304/flows.pl?tree=7304, then clicking Flow Charts in the left-hand menu.

Cisco offers flowcharts for most of its network devices; you can access them from the support section of the Web site.

Q 3.7: We have a number of junior administrators, so we need to make network device troubleshooting more of a science and less of an art. What can we do?

A: You can create a sound troubleshooting methodology. To do so, simply answer this question: "How do you find a wolf in Siberia?" Sounds frivolous, but it's a similar task to network device troubleshooting, which can often seem to an inexperienced administrator like looking for a needle in a haystack. The answer provides the solution: Build a wolf-proof fence down the middle of Siberia, and look for the wolf on one side. If he's not there, divide what's left in half again, and repeat. Technically, the technique is referred to as a *binary search*.

An Example Problem

Consider the network diagram that Figure 3.3 shows. Imagine that the client using the laptop computer isn't able to communicate with the desktop computer in Office 1.

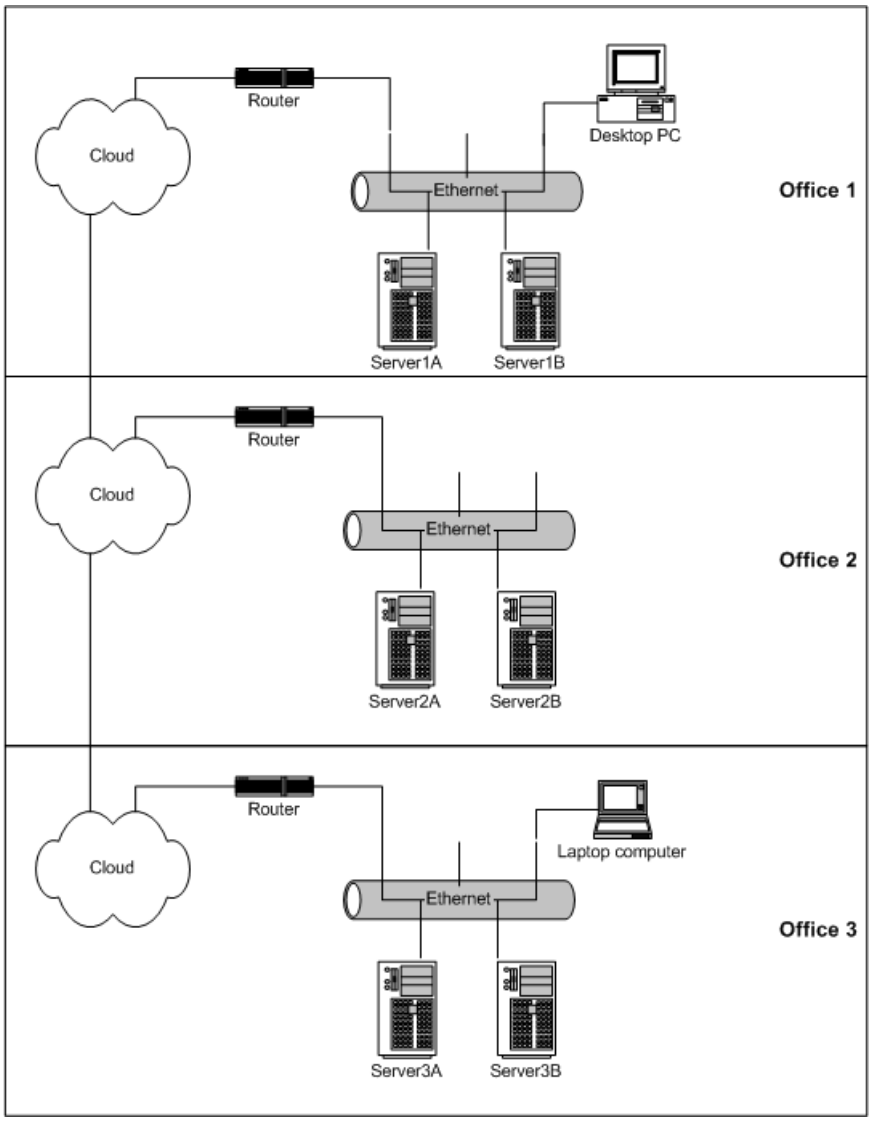


Figure 3.3: Sample troubleshooting problem.

This is a simplistic example, but it will serve to illustrate a troubleshooting methodology, which can be used for any problem, no matter how complex.

Identifying the Problem Domain

The first step is to simply make a list of everything that could be causing the problem. Experienced administrators do this in their head, but it's worth writing down the list if you're just getting the hang of troubleshooting. In this case, the list might include:

- Laptop unplugged
- Laptop network stack failure
- Desktop unplugged
- Desktop network stack failure
- Router in Office 3 failed
- Router in Office 1 failed
- WAN link failed
- DNS server not working
- Bad routes in Office 1 router
- Bad routes in Office 3 router

It's important to make this list because doing so will rule out elements that might seem to be problems—such as the router in Office 2—that obviously aren't. Of course, the ability to generate a list such as this example list requires a thorough understanding of how the network is built (having documentation such as the network diagram is invaluable) and a thorough knowledge of how the network operates. For example, if you don't know how computers resolve names to IP addresses, you might not suspect the DNS server.

Breaking the Testable Systems in Half

Next, develop some logical means of dividing the land in half. In this case, about half the potential problems seem to be router-related, and the other half are client-related; breaking the list along those lines creates a basically even set of possibilities.

Router Problems

- Bad routes in Office 1 router
- Bad routes in Office 3 router
- Office 1 router failed
- Office 2 router failed
- Bad WAN line

Client Problems

- Laptop unplugged
- Desktop unplugged
- Stack failure in laptop
- Stack failure in desktop
- DNS server failed

Figure 3.4 illustrates how this process effectively divides your suspect subsystems into a logical half.

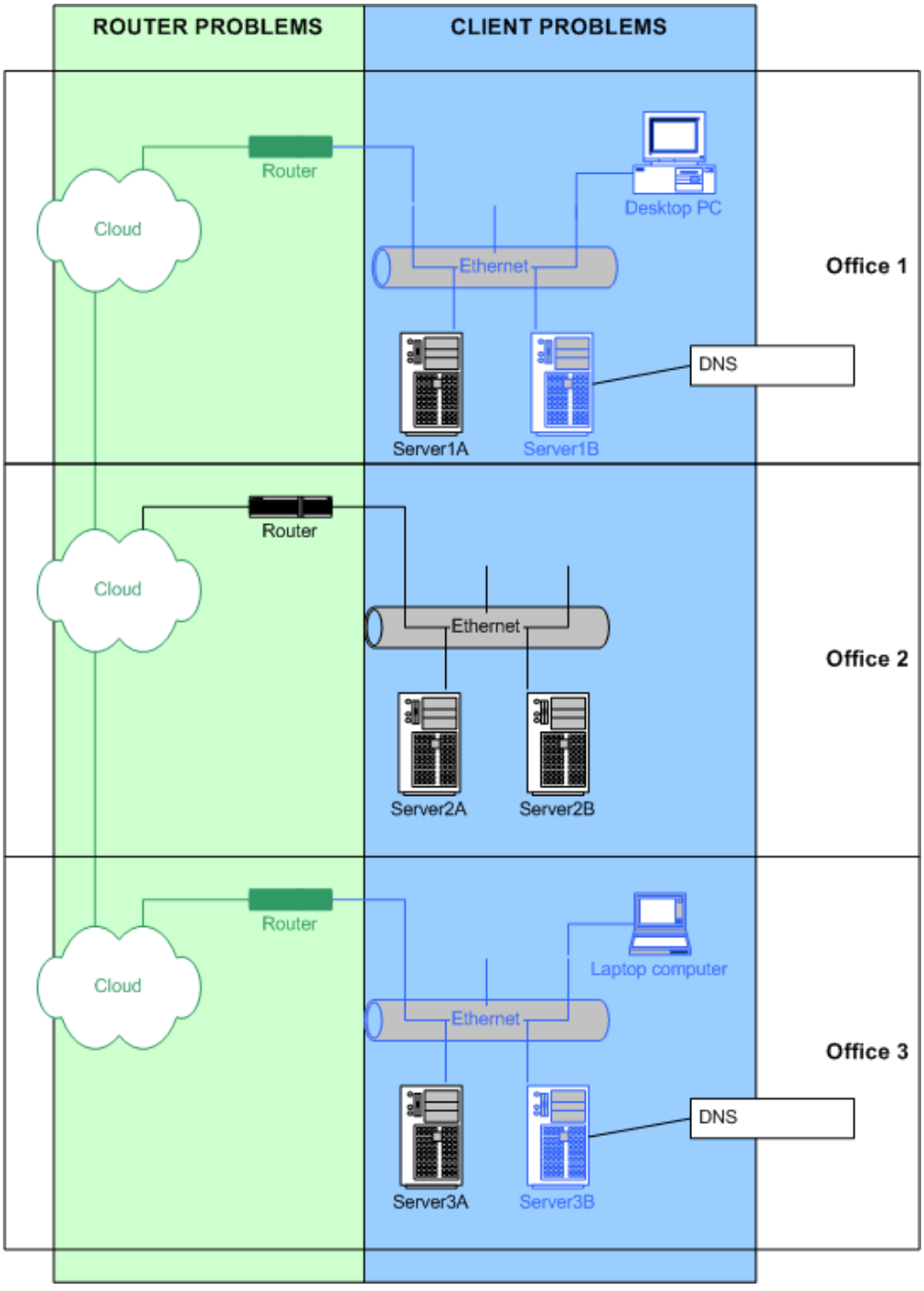


Figure 3.4: Dividing the suspect subsystems into half.

Now you need to build your wolf-proof fence down the middle by conducting a test.

Performing Tests

The only useful troubleshooting tests are those that allow you to definitively eliminate some potential problem. For example, suppose you determine that the laptop computer also can't connect to a server in Office 2. What have you proven? Well, nothing, really. You can't even say for sure that the Office 3 router is OK, although it's now less likely that it has failed or has a bad route. In other words, you haven't built a wolf-proof fence at all.

Suppose, however, that you are able to connect to computers on the Office 3 network from the laptop, and connect to computers on the Office 1 network from the desktop. That's a definitive test: you can eliminate half of your suspect systems from the list because you've proven that they work.

> ☞ Stuck for tests? Go one-by-one. If you can't readily think of a test that will result in your wolf-proof fence, you can just eliminate half of the list on a subsystem one at a time. For example, you can check the connections on both computers and ensure that they can ping their gateways to ensure that their stacks are functioning. You can use nslookup to test the DNS server(s) to eliminate them from the list. However, *efficient* troubleshooting requires you to be able to divide the list in such a way that one or two tests can eliminate half the list. That type of efficiency comes primarily with string knowledge of how the network works and with good old experience.

Divide, Conquer, Repeat

With half the list out of the way, you can start working on the other half. Figure 3.5 illustrates the systems you've eliminated, including DNS servers at each office (shown in the diagram as Server1B and Server3B), the client computers, and their network connections.

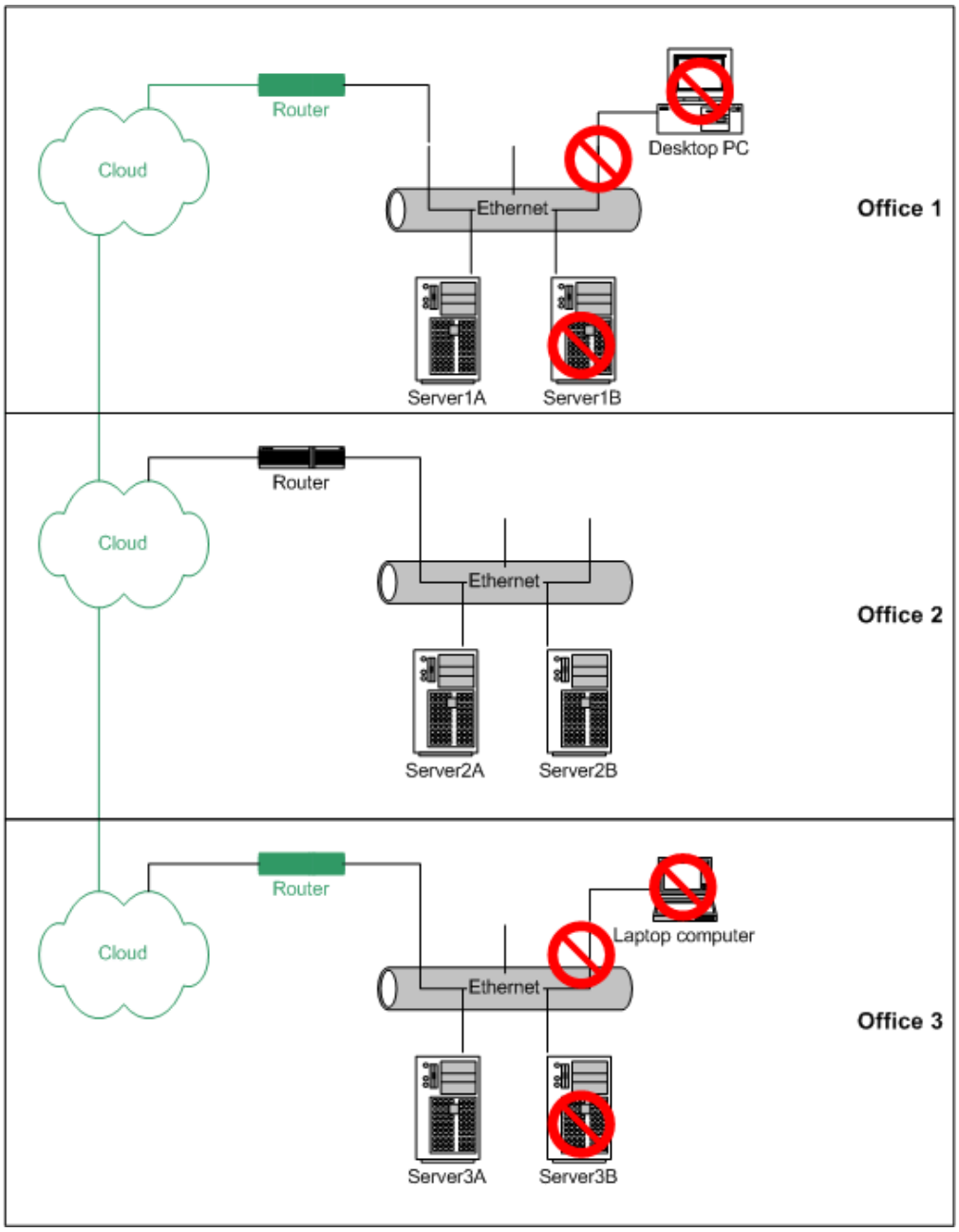


Figure 3.5: Half the suspect systems eliminated, with just the green-colored half to go.

Additional tests at this point could involve logging on to one of the two routers and attempting to ping the other one. That test, if it worked, would eliminate the WAN links as a potential suspect and let you know that at least the routers' external interfaces are up and running. You'd be down to a quarter of your original list, and the odds would start looking good for a bad route in one of the routers. Manually checking the routing tables would let you know whether that was the problem.

Shortcuts

In some cases, you might be able to go after the entire list of suspect systems with one good test. For example, running tracert from the laptop to the desktop will help you eliminate most, if not all, of the suspect systems. If DNS has failed, tracert will tell you so. If it's a local connectivity issue, you'll see that in the results. If a router has a bad route, you'll see that in the results, too. A WAN failure won't be distinguishable from a failed router interface, but you'll at least have narrowed the list to two possible candidates.

> ☞ Know your tools! Another trick to performing this methodology is having thorough knowledge of the troubleshooting tools at your disposal. Knowing what ping, pathping, and tracert can do, for example, will enable you to select the most effective test for eliminating a particular subsystem.

Selecting the right testing tools can make all the difference, particular with regard to efficiency. For example, if you were following the troubleshooting path I've been using, you might have spent an hour or so figuring out that a bad route was at fault. Tracert, however, could have brought you to this conclusion in 5 minutes or so. However, you would have found the problem either way, eventually, proving that the methodology is useful even to an administrator without years of experience.

Now It's a Science

Where do most new administrators get caught up? First, they might not completely understand how the network functions, so they ignore suspect subsystems and spend their time troubleshooting only part of the problem. Second, they often don't perform conclusive tests—they might incorrectly eliminate a suspect subsystem, and waste time looking for wolves in the wrong part of Siberia.

It's a simple methodology, one that experienced administrators follow almost without thinking about it—which makes it difficult to teach to newer personnel. To summarize:

- Identify the actual cause of the problem
- List suspect subsystems
- Break the list into halves so that one half can be eliminated by one or two conclusive tests
- Perform conclusive tests to focus on one half or the other; repeat the process by splitting what's left into half
- Ensure that all tests can conclusively eliminate something; essentially, all tests must prove that something is either working or not with no room for question

This tried-and-true methodology becomes instinctive through experience, but for less experienced technical professionals, it can make the daunting task of network troubleshooting more approachable, methodical, and efficient.

Q 3.8: How can we proactively ensure that our devices are properly configured at all times?

A: To ensure proper device configuration, always use templates to configure them. First, develop standards for creating templates. IP addresses usually need to be represented in some kind of code form. For example, use a.b.c.d and e.f.g.h to represent the IP addresses of management stations; a.b.c.d usually represents my network's Hewlett-Packard OpenView station or some other Simple Network Management Protocol (SNMP)-based software, so I use that address to configure the device to send SNMP traps to that address. The management software usually provides syslog functionality as well, so I program devices to send syslog traffic there.

As time is an important consideration, I also built NTP server addresses into my templates, using the addresses m.n.o.p and q.r.s.t. This technique is fairly common (although everyone uses different letters) for building configuration templates. Listing 3.1 shows a configuration template for Cisco routers and switches (and other devices) that runs the Cisco IOS.

For your convenience, I've boldfaced the items that you'll need to change in this template: IP addresses, your SNMP community strings, and some management user account names.

```
! Permit SNMP access from mgmt software
! Template provides for two:
access-list 60 permit a.b.c.d
access-list 60 permit e.f.g.h

! Set SNMP community string
snmp-server community <comm-string> RO 60
snmp-server community <comm-string> RW 60

! Add contact info:
snmp-server contact admin@company.com

! Source SNMP traps from the loopback address:
snmp-server trap-source loopback 0

! Allow reboot of device after upgrade:
snmp-server system-shutdown

!
! where to send SNMP traps to, which traps to send
! (this will send ALL traps)
snmp-server host a.b.c.d traps version 2c public
snmp-server enable traps

! Disable the some verbose or redundant traps.
no snmp-server enable traps snmp authentication
no snmp-server enable traps syslog
no snmp-server enable traps config

! disable link status traps on user IOS switch ports
! or dialer interfaces on routers:
interface ...
no snmp-server trap link-status
```

```
! set logging options
logging buffered 128000 debugging
no logging monitor
no logging console
logging trap informational

! Send syslog messages to mgmt station
logging a.b.c.d

! set logging source to loopback
logging source-interface loopback 0

!
line con 0
logging synch
exec-timeout 0 0
line vty 0 4
logging synch
exec-timeout 10 0

!
! Enable RCP for config and IOS transfers:
ip rcmd rcp-enable
ip rcmd remote-host <user> a.b.c.d <user> enable

!
! Enable Web management unless:
ip http server

!
service timestamps log datetime show-timezone
service timestamps debug datetime show-timezone

! Set EST time zone:
clock timezone EST –5

! point devices at the Internet sources
! for time.
ntp server m.n.o.p
ntp server q.r.s.t
! Might change this so that only two devices
! look at the Internet, and the other devices
! look at those two.

! For boxes with hardware clock/calendar
ntp update-calendar

! Globally enable SNMP ifindex persistence
snmp-server ifindex persist

!
! Limit SNMP-triggered TFTP to the mgmt workstations:
snmp-server tftp-server-list 60

! Force sub-interface traps
snmp-server trap link ietf
```

Listing 3.1: An example device configuration template.

By creating a template like this example, you can ensure that certain configuration options aren't overlooked. Obviously, you'll need to modify this template to meet your specific device configuration standards; this script sets up some specific SNMP traps as well as time synchronization and other options. Using the template is easy. Within a copy of the template, search and replace the placeholders—such as a.b.c.d—with the appropriate information, then use the template as a script to configure a device.

A best practice is to create a small library of such templates. That way, you can have one basic template that is used to configure devices, and others that are used, for example, to change the SNMP community strings on a regular basis. You'll need other templates, of course, for other device brands or versions, such as switches that use CatOS.

An even simpler solution is to employ a device management solution that has built-in support for templates. Some solutions store the template internally, then use an onscreen form to collect IP addresses and other device-specific information. The information from the form is plugged into the template's placeholders before deploying the configuration. This type of device management solution—particularly solutions that support change management and configuration version tracking—can be incredibly useful in helping junior administrators make consistent device changes.

Templates only provide part of the solution for making sure that your devices are properly configured at all times. Specifically, they help ensure that configuration changes are made using a consistent, approved group of settings. Devices, however, don't *require* anyone to use a template, so it is still possible for an administrator to make an inconsistent change without using a template. In such cases, the only solution is to *catch* and roll back the change after it occurs. For this type of real-time notification, you'll need a third-party solution that can monitor devices' SNMP traps and syslog entries to watch for people logging on and off of the devices. The logon or logoff event should trigger the third-party software to poll the device's configuration and compare it with earlier versions to determine whether the configuration had changed.

Topic 4: Change Management Techniques

Q 4.1: 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). AlterPoint's product can automatically back up device configurations, alert you to changes, show you exactly which changes were made, and even restore devices' configurations if a disaster occurs. 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.

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 great 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'll have 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.

Q 4.2: What's 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 change-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).

Now 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 files 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:

1. Enter

   ```
   cd /tftp
   ```

   to change to the TFTP directory that you created.

2. Enter

   ```
   chmod a+w Routerbackup
   ```

   to set the correct permissions on the file.

3. 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 device; change the following commands as necessary if you're using a different device.

4. Log on to the router. Enter

   ```
   enable
   ```

   and provide the correct password.

5. Enter

   ```
   write network
   ```

   and enter the IP address of the TFTP server.

6. Enter the name of the configuration file (Routerbackup if you're following this example).

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

8. 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 4.1 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.

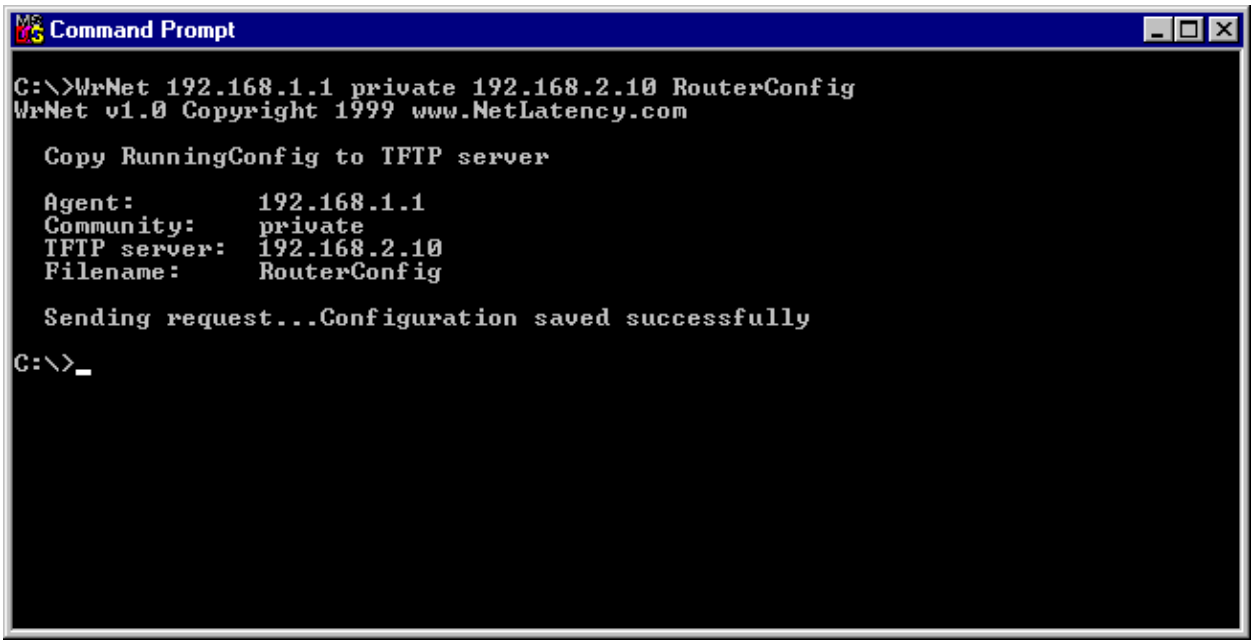


Figure 4.1: Running WrNet from the command line.

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 change-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.

Q 4.3: 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 change-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 (http://www.ciscosystems.com) and Nortel Networks (http://www.nortelnetworks.com/), 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 4.2.

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 change-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 4.4: 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 4.2 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.

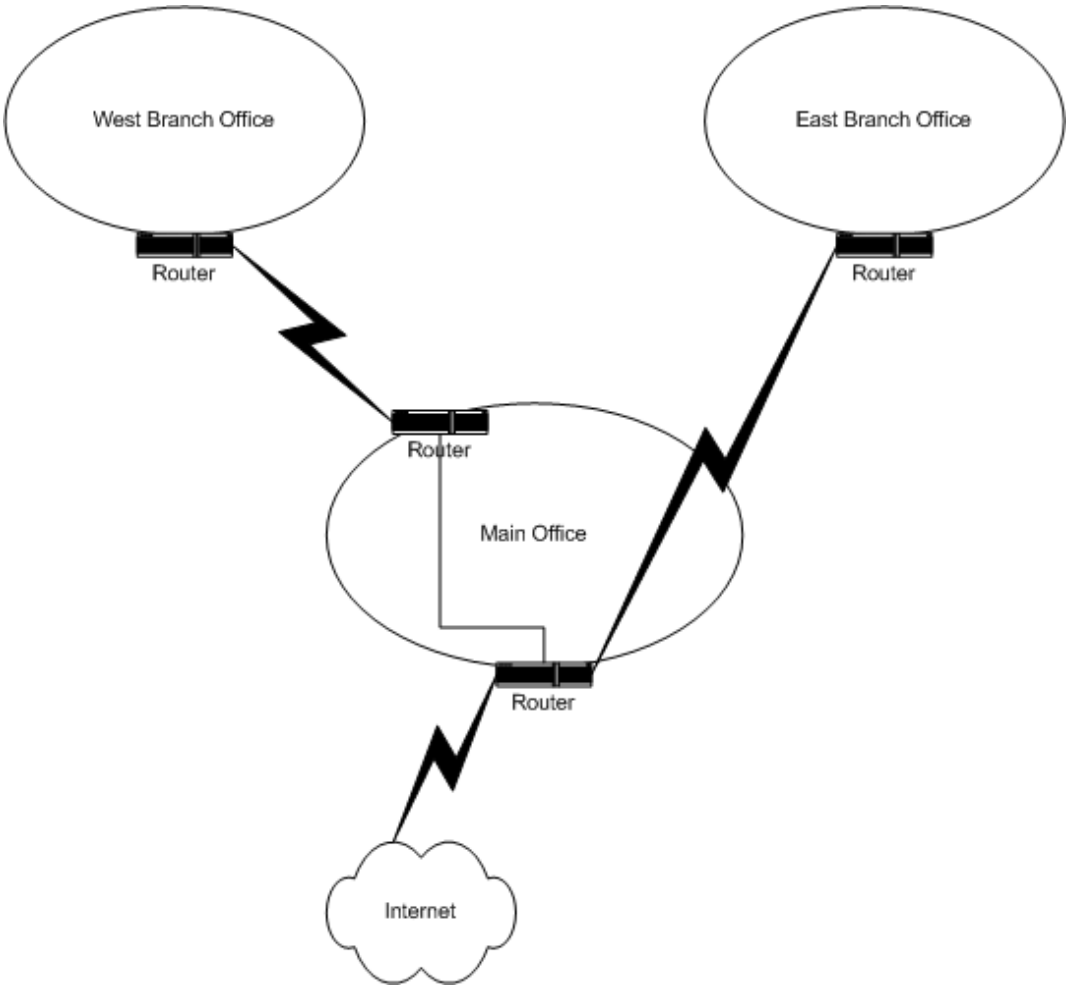


Figure 4.2: Differences in network connectivity can affect users' perception of performance.

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's probably some other configuration difference that's responsible.

Unfortunately, most change management software packages don't make it easy to compare two different devices' configurations to one another. So you'll 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 4.2). Once you've got the two, use a graphical file comparison tool to compare the two files. Figure 4.3 shows sample comparison results.

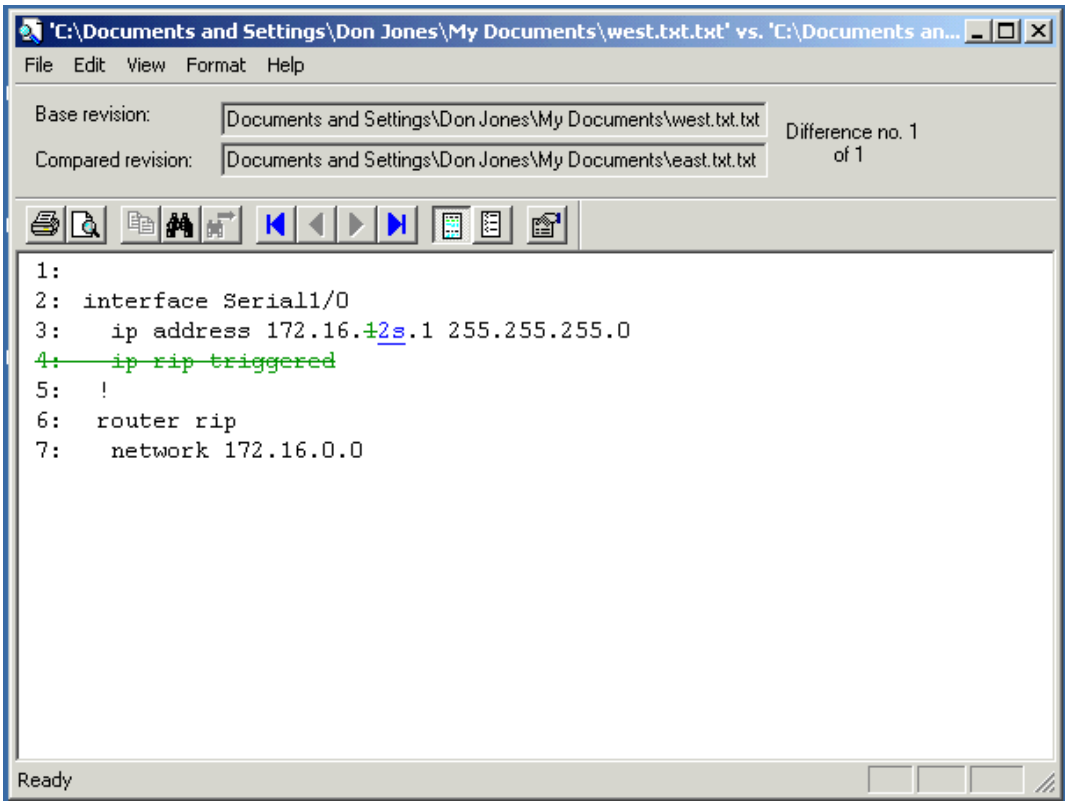


Figure 4.3: Sample comparison between two router configurations.

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 device 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 4.5: Aside from Trivial File Transfer Protocol, what are other ways to retrieve a device's configuration information?

A: If you're creating your own change management infrastructure, the ability to retrieve your devices' configuration settings is at the heart of the process. In previous tips, I've discussed how to use the Trivial File Transfer Protocol (TFTP) to retrieve this information. Although TFTP is supported by almost all high-end devices, those devices often also support alternative means, such as the Remote Copy Protocol (RCP).

RCP is a subset of the Rshell service, which is included on most UNIX-based systems. Rshell is designed to allow administrators to remotely execute UNIX shell commands and is a common remote administration tool for UNIX-based servers. Like Rshell, RCP uses TCP connections on port 514 (by default). Many network devices, such as some Cisco IOS-based devices, include Rshell (also called Rcmd, or Remote Command). RCP is supported on all Cisco IOS-based devices.

Enabling RCP

RCP is generally disabled by default for security reasons—no sense in having a protocol enabled that you might not use! So if you do want to use RCP, you'll need to turn it on. Listing 4.1 shows a sample router configuration file that includes RCP support.

```
Current configuration:
!
version 11.3 service timestamps debug uptime
service timestamps log uptime
service password-encryption
!
hostname calvi
!
boot system c2500-is-l.113-11a.T1.bin 255.255.255.255
enable password 7 1106170043130700
!
username rmuser password 7 000C1C0A05
ip rcmd rcp-enable
ip rcmd remote-host cwuser 172.17.246.221 cwuser enable
ip rcmd remote-username cwuser
!
!
process-max-time 200
!
interface Loopback0
 ip address 5.5.5.5 255.255.255.255
 no ip directed-broadcast
!
interface Ethernet0
 description Connection to Backbone
 ip address 172.17.246.4 255.255.255.0
 no ip mroute-cache
!
interface Serial0
 no ip address
 no ip mroute-cache shutdown
 no cdp enable
```

```
!
interface Serial1
 no ip address
 no ip mroute-cache shutdown
 no cdp enable
!
interface Async1
 no ip address
!
ip classless
ip route 0.0.0.0 0.0.0.0 172.17.246.1
!
logging monitor informational
snmp-server community private RW
snmp-server community public RO
snmp-server enable traps snmp
snmp-server host 172.17.246.117 traps public
!
line con 0
 exec-timeout 0 0
 password 7 0504080A754D4205
 login
line 1 8
 exec-timeout 0 0
 login
 transport input all
line aux 0
 password 7 06090124184F0515
 login
line vty 0 4
 exec-timeout 0 0
 password 7 06090124184F0515
 login
!
end
```

Listing 4.1: A sample router configuration file that includes RCP support.

The boldfaced lines in Listing 4.1 are the keys to RCP:

- **Username rmuser password 7 000C1C0A05**—Creates a user named rmuser on the router and allows you to choose a password for the user.
- **ip rcmd rcp-enable**—Enables this RCP service on the device.
- **ip remote-host rmuser 172.17.246.221 rmuser enable**—Allows a remote user, named rmuser, using IP address 172.17.246.221, to execute the copy command on the device using the local rmuser user account.
- **ip rcmd remote-username rmuser**—Configures the remote username that will be used when requesting a remote copy through RCP.

RCP works a bit like TFTP in that it will dump the devices' configuration information to an Rshell server. Thus, you'll also need to configure an Rshell server in your environment (UNIX systems, as I mentioned, come with Rshell by default). Windows systems don't come with Rshell by default, and Microsoft doesn't provide one. Thus, if you're using a Windows server to store device configurations, you'll need to obtain a third-party Rshell service.

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.

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 (HSPR). 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 s 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-router*x*.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 change 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.

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 change 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 change 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.

Q 4.7: How can we incorporate server change management with our network device change management?

A: That's a tall order. Network device change management products such as AlterPoint's DeviceAuthority and ReadyRouter use a fairly common set of techniques for retrieving configuration data, primarily Telnet, Simple Network Management Protocol (SNMP), and Trivial File Transfer Protocol (TFTP). Servers, however, typically use their own proprietary methods. Windows servers might use remote procedure calls (RPCs), Windows Management Instrumentation (WMI), or even product-specific management agents (such as Compaq Insight Manager) to retrieve information. Incorporating all of that functionality into one product is difficult.

One company that has taken a stab at multi-platform change management is Ecora, with its Enterprise Auditor product. This solution can collect information from network devices, Windows, UNIX, Linux, NetWare, SQL Server Exchange, Active Directory (AD), and many other systems. Another company that offers such a solution is Marimba, which also supports client operating systems (OSs) such as Mac OS X. Both solutions offer differing features, and you'll need to carefully evaluate them to see which best meets your needs.

There are other, platform-specific solutions that offer more functionality. Microsoft's Systems Management Server (SMS), for example, provides change management, software deployment, license metering, and other systems management features for Windows-based computers. Novell's ZENWorks provides similar capabilities for NetWare clients and servers.

All of these products have a fairly significant implementation requirement, both in terms of cost and administrative support. They're not like device-only solutions, which can often be installed, configured, and fully deployed in an afternoon. SMS, in particular, is almost a form of rocket science, with complex server and network deployment requirements, hierarchies of management servers, and so forth (see Figure 4.4). Deploying SMS is a significantly more complex deployment than that of most network device change management solutions, which often require nothing more than a single client computer for an entire enterprise. SMS also requires the distribution of software agents (which it will do for you through logon scripts). Some competing solutions, such as the product from Ecora, don't require software agents but also don't provide features such as software distribution. Marimba's solutions compete more closely with SMS, but don't provide change management for devices such as routers and switches.

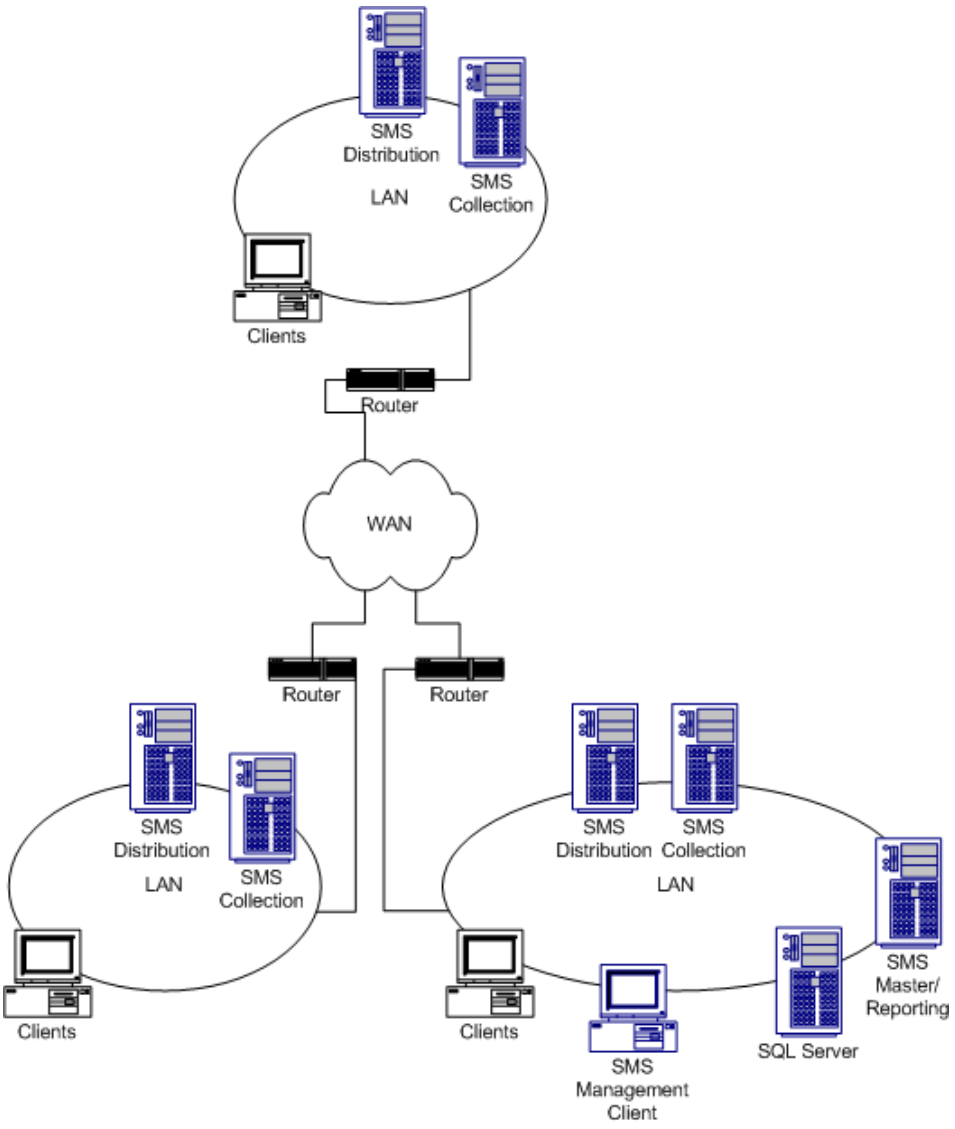


Figure 4.4: Sample Microsoft SMS deployment.

However, there is a serious issue you must consider in the quest for a "do it all" solution. Companies today are trying to reduce overhead and expenses, and finding one solution that does everything can definitely help meet that goal. At the same time, though, you need to recognize that a "one size fits all" solution won't necessarily fit anything *well*. Solutions designed specifically for network device change management have to fill a different type of role than solutions designed for server change management. After all, you certainly don't expect to review your servers' configuration files in a two-pane window that shows the differences between two versions of the configuration. Most servers have literally dozens of configuration files that might need to be reviewed in such a manner. However, that technique is extremely useful for working with network devices, which are inherently simpler machines.

My advice is to find solutions that are designed for a specific task, such as server management or network device management. You'll get a better feature mix for each task, have fewer deployment issues, and in the end, will probably spend less time wishing you'd bought something else.

Q 4.8: How can I reset all of my devices to a known-good baseline configuration?

A: Hopefully, you *have* a known-good baseline configuration. Ideally, you'll have such a baseline stored on a TFTP server because you're making backup copies of your device configurations. You can simply restore that configuration to your devices. The commands to do so for a Cisco device look something like the following example:

```
ciscorouter> (enable) copy tftp config
IP address or name of remote host []? 192.168.1.100
Name of file to copy to []? Backup-04-05-2004.txt
Configure using tftp:config1.txt, (y/n) [n]? y
Console> (enable)
```

Network configuration management software such as AlterPoint DeviceAuthority, Cisco CiscoWorks, and ReadyRouter can make the process easier by pushing a previously saved configuration into a device for you.

If you have multiple devices that require a known-good baseline configuration to be re-implemented—perhaps an overzealous junior administrator plugged the wrong IP address into the wrong place in a dozen routers—you will benefit from automated software that can roll back the devices' configurations for you. Thus, if you're considering a change-management solution for your network devices, consider one that offers the ability to roll back devices to a prior version of their configurations, making it easy to not only audit changes but to quickly recover from an erroneous change.

Q 4.9: How do we configure network devices to immediately alert IT staff when a configuration change is made?

A: Network devices—routers, hardware firewalls, switches, and the like—simply aren't designed with event-driven notification services built in. In fact, most devices don't create a Simple Network Management Protocol (SNMP) trap when a configuration parameter is changed. However, combined with the right third-party software, you can create real-time alerts for configuration changes.

First, define what *real-time* and *immediately* mean within your organization. For example, in other tips, I've explained how you can use third-party change-management software to periodically poll your devices and pull their configurations. Such software can easily compare the just-polled configuration with a prior configuration, highlight any changes, then send those changes via email to IT staff (or provide the changes in a report or another format, if desired). This method is a proactive, hands-off way of managing device configuration changes. However, unless you're planning to configure the change-management software to perform this process six times an hour—which wouldn't be practical for performance reasons—this technique isn't really immediate or real-time.

Constant checks aren't practical because devices can't report when a change occurred. Each time the change-management software checked devices' configurations, the software would need to completely download the devices' configurations and compare them with earlier versions to look for changes.

Although devices don't automatically notify you of a change, you can configure them to send an SNMP trap, or an accounting message, every time someone logs on or off a device. An accounting message, for example, might create an entry in a Remote Authentication Dial-In User Service (RADIUS) or TACACS+ log file, indicating that someone logged on or off of the device. An SNMP trap would provide similar information. As it is pretty much impossible to change a device without logging on in some fashion, a logon notice is a clue that something *might* have changed in the device configuration. You can use that notification as a trigger to check the device's configuration against an earlier version to determine whether anything has changed.

Third-party software packages provide this type of functionality. AlterPoint DeviceAuthority, for example, can read SNMP, TACACS+, and other logs to look for evidence that someone might have modified a device's configuration, then treat that evidence as a trigger to pull the device's configuration file. DeviceAuthority—like Cisco CiscoWorks, ReadyRouter, and other software—maintains a database of past configuration files for each device, so the software can compare the current version to the most recently stored version to determine whether the configuration has changed. It is possible that an administrator simply logged on and didn't make any changes; in such cases, the software will not detect any configuration changes and won't treat the event as an alert. In cases in which a change is made, the software can log that fact and take an appropriate action based on your preferences.

Figure 4.5 shows DeviceAuthority's events view. In this example, several configuration change events have been logged. For each event, the date, time, and device information are logged. You can view any of the events to see exactly what happened in the event.

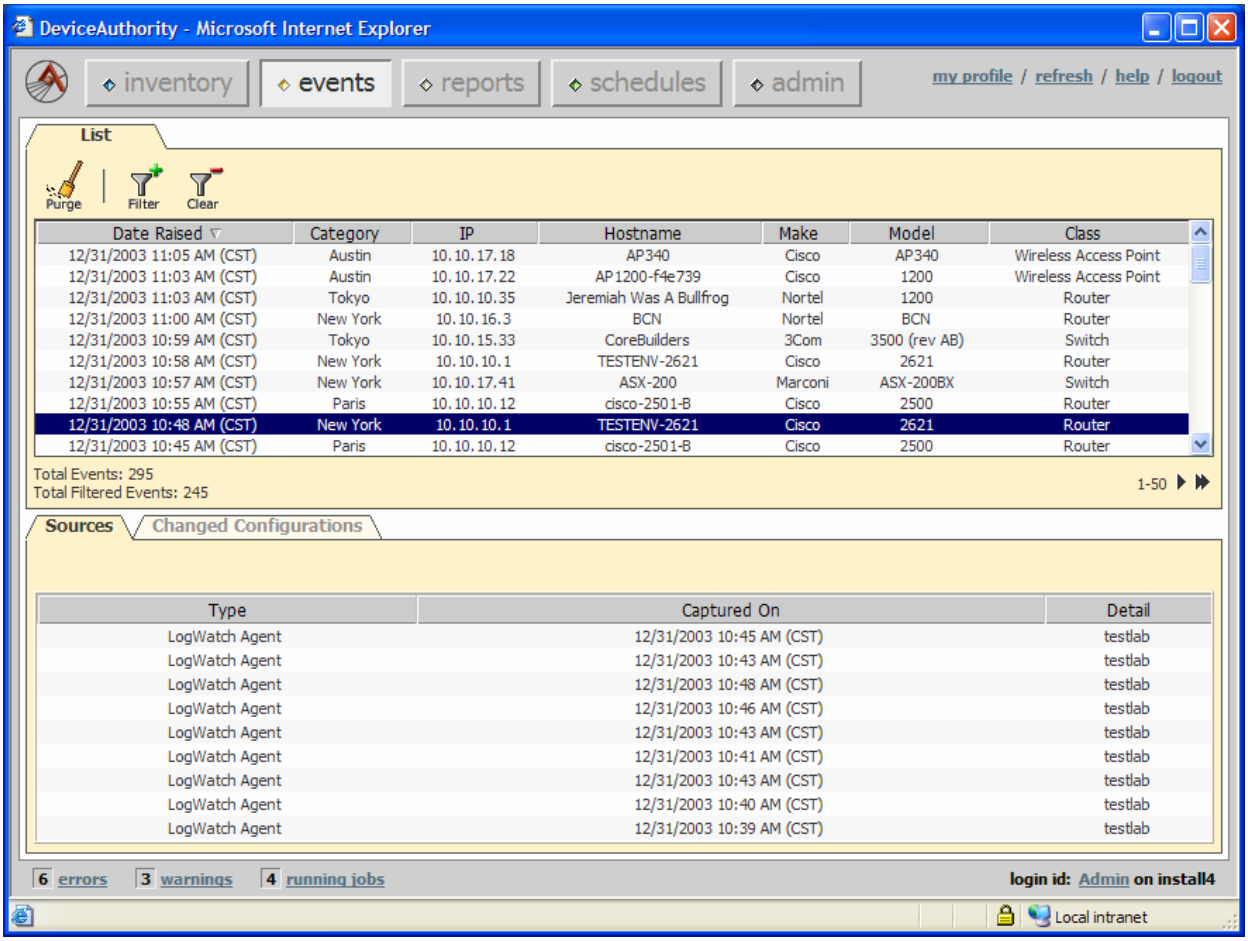


Figure 4.5: Viewing configuration change events in DeviceAuthority.

Essentially, the software is filling the gap between the functionality built-in to the network devices—SNMP traps and RADIUS or TACACS+ logging capability—and your business requirements for real-time configuration change alerts. The software looks for evidence that the configurations *might* have changed, checks the configuration for changes, and creates alerts and events as necessary to keep you informed.

> In many prior tips, I've showed you how to cobble together your own change-management solution using scripts and freely available tools. For example, I showed how to use TFTP and command-line utilities to store device configurations and scan them for changes. In this case, however, the business requirement for proactive, real-time (or near-real-time) notifications of changes goes beyond what you can readily create with scripts. You'll need to evaluate a third-party network device management and change-management solution to provide the needed functionality.

Topic 5: Selecting and Deploying a Network Device Management Solution

Q 5.1: All of our equipment is from that vendor. Why not use a vendor-supplied device management solution?

A: Some network device vendors, most notably Cisco, provide some 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 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 change 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 5.2: We want to evaluate change management products but don't want to wreck our production environment in doing so. What's the best way to proceed?

A: Obviously, evaluations of any kind should be conducted on a test network that's completely disconnected from your production network. That way, you can evaluate the solution's impact on your network in a completely safe and recoverable fashion. However, change management solutions can't generally be accurately evaluated on a small, limited scale. After all, the point of the solution is to manage hundreds of devices; if you only needed to manage one or two devices, you wouldn't need a solution! So at some point, you'll likely want to evaluate the solution on your production network to see how it will behave under real conditions.

Test Network First!

No matter how comfortable you are that a change management solution won't impact your production network, install it on a standalone test network first. Set it up to manage one or two different devices, and perform a careful analysis of its impact. Specifically, look for the following:

- What changes does the solution require you to make to your devices, and what changes does it make on its own?
- What security credentials will the solution need to access your devices? Will it be able to use existing credentials, or will you have to set up something entirely new?
- How much network bandwidth does the solution utilize during its initial setup and device acquisition? How much bandwidth does it use for normal day-to-day operations?
- How much time does the solution require to install?
- Does the solution require any additional services, such as a Web server or Simple Network Management Protocol (SNMP) server, which might create an impact or security concern?

Your goal is not so much to record this information as a part of your evaluation of the product's suitability, but rather to find out how much impact the product will have if you install it on your production network for a more full-fledged evaluation.

Not safe to evaluate in production? Red flag! Be wary of products that create such a high impact that you don't feel comfortable even evaluating them on your production network. Even if you do select such a solution based on test network evaluations, the product might be difficult to remove from your network if you ever decide to quit using it, or even if a major new version comes out later. Ideal change management solutions should create a minimal impact, be safe for testing in your production environment, and be easily removed from service if desired.

Also, keep in mind that so far you're only working with an evaluation copy of the solution. Find out from the manufacturer what the differences are between the evaluation version and the "real" version you might purchase. The evaluation version might create a minimal impact on your network (perhaps it only changes the configuration of one device as part of its demonstration limitations), but the real version might not be so friendly—this difference is something that you'll want to make part of your shopping criteria. You might organize your test information into a basic grid, as Table 5.1 shows.

Criteria	Product A	Product B	Product C
Ease of installation	Very simple—Setup.exe	Very simple—Setup.exe	Requires manufacturer consultants
Ease of removal	Easy—Uninstalls cleanly	Somewhat—Uninstall routine doesn't put device configs back	Unclear—Manufacturer has no information
Changes required to devices	None	Device passwords must be changed; some privilege adjustments required	Device passwords and SNMP strings must be changed
Bandwidth for setup	Moderate—SNMP discovery can be planned for overnight run	Moderate—SNMP discover can be planned for overnight run	None—Configuration is manual and performed by consultants
Ongoing bandwidth requirements	Low—Uses mainly TFTP to collect data; can be scheduled for overnight	Higher—Re-discovers SNMP devices on a regular basis	Moderate—Uses mainly TFTP and SNMP to collect data, but does so several times each day (not configurable)
Additional services	Installs Apache; will need approval from Info Security dept.	Uses IIS on management workstation; will need approval from Info Security dept.	Uses customized server software; Info Security dept. will have to evaluate risks
Evaluation vs. production differences	Eval is fully functional and time limited	Eval supports max 10 devices; production bandwidth might be higher for more devices	Eval is fully functional and time limited
Overall risk of evaluating on production network	Low—Solution can be removed with no traces	Moderate—Device changes will have to be manually undone	High—Specialized skills required to install/remove and device changes are not automatically undone

Table 5.1: Change management product test information grid.

A chart like this one can help you formally determine which solutions offer a low-impact evaluation and which need to be constrained to a test network for evaluation purposes. Be aware, of course, that test networks rarely reflect the true day-to-day conditions of your production network, so you probably won't be able to accurately test factors such as workload and performance that will be affected by the conditions of your production network.

Evaluate for the Evaluation

Change management solutions are not, of course, created equal. After determining the impact a solution has on your test network, you can evaluate the risks of a full-fledged evaluation on your production network. Risks to consider include:

- The cost of the evaluation. Some solutions will require you to basically buy a full license that just includes a few devices or is time limited.
- The cost of installation. Some solutions require specialized consulting work to get up and running; others are ready after you run a short Setup routine on a test workstation.
- The impact on your devices. Some solutions operate in a completely passive manner, making no changes to your devices under normal circumstances; other solutions require you to modify your devices' configurations to support the solution.
- The impact on your network. Some solutions might need to conduct network-intensive searches and data transfers as a part of their setup; others might simply need to transfer a few configuration files from your devices, a task you can often schedule to occur during off hours.
- Ease of removal. This important consideration is related to network and device impact. If you decide to test a solution on your production network, you should be able to remove the solution through a simple uninstall routine, or at worst, by reformatting a test workstation. Under no circumstances should removal require changes to your network devices, as doing so would present a high risk for error and downtime.

The ideal solution, from an evaluation standpoint, is one that has the lowest cost and least impact. Often, the more complex and powerful the solution, the more impact it might have, although that's not always the case. Plenty of powerful, flexible solutions exist that meet a variety of common needs while presenting very few risks to a production network evaluation. Discuss your evaluation plans, concerns, and needs with a sales representative from the solution's manufacturer to find out what their product offers in exchange for the impact it might create on your production environment.

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 change 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 change 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 change 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 change 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 change 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 change management solutions, such as those from AlterPoint and Ecora, have a fairly easy deployment methodology:

- Install the software.
- Either use auto-discovery to add devices or add them manually.
- 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 change management solutions will make this process easy.

Q 5.4: Our network devices include load-balancing and network address translation devices that are difficult to connect to for management. How can we include them in change management?

A: Normally, I'd say it should just work. For example, consider the common configuration in Figure 5.1, which includes a Web farm connected to a hardware load-balancing device, such as a BigIP or LoadDirector box.

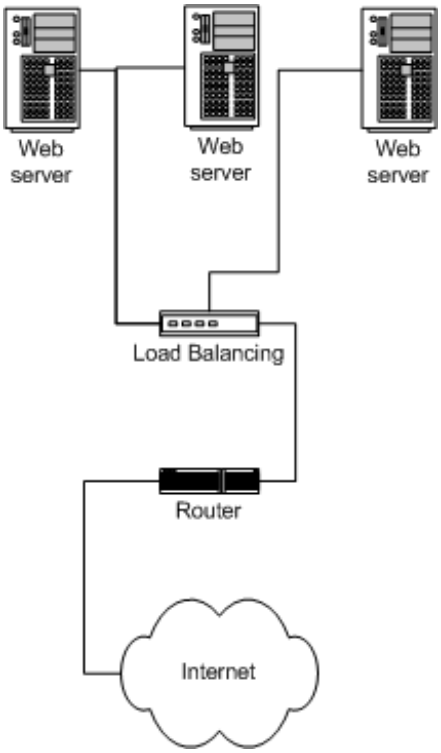


Figure 5.1: Common load-balancing scenario.

Suppose for the sake of argument that the router is also acting as a rudimentary port-filtering firewall, although in reality you'd likely find a dedicated firewall in the picture as well. Your network management tool, then, will need to log on to both the router and the load-balancing device. It should be able to do so with no problems as-is, even from across the Internet (assuming the appropriate ports are opened throughout). Keep in mind that your management solution will likely want to use Simple Network Management Protocol (SNMP) or Telnet to access the devices; those ports aren't typically load balanced. In Cisco IOS Server devices, for example, you have to specify which ports are included in load balancing, as in this example:

```
virtual 157.68.12.2 tcp 80 service http
```

It's rare that Telnet or SNMP would be enabled for load balancing. In fact, under most circumstances, you should even be able to connect to the servers on Telnet or SNMP by using their dedicated private IP addresses instead of the virtual IP used by the load-balancing device.

Everything's Load Balanced

However, let's assume that the load-balancing device is, in fact, load balancing everything. You should *still* be able to connect directly to the device using its own IP address rather than the virtual IP address that represents the Web farm. Having everything load balanced shouldn't even make it harder to access the Web servers. They should be using their own IP addresses, which should be accessible directly. Load balancing should only be affecting traffic sent to the virtual IP address set up on the load-balancing device.

Security Concerns

However, in this scenario, management might be a problem. Suppose that Figure 5.1 represents an off-site Web farm, and you want to perform network device management—such as change configuration management—on the router or the load-balancing device. As I mentioned earlier, there's likely to be a firewall in the picture, and you're not very likely to have SNMP or Telnet opened to the Internet. You've still got possible solutions.

First, you could configure the firewall to permit those ports inbound only from your company network's IP addresses. Doing so would enable your configuration management system to access the devices through the firewall to perform its work. I don't recommend this method, however, as it will have device configuration information transmitted in clear text across the Internet. That information, if captured, could be used to form an attack against the Web farm, which would have a greater probability of success thanks to the insider information.

Alternatively, as Figure 5.2 shows, you could establish a VPN connection.

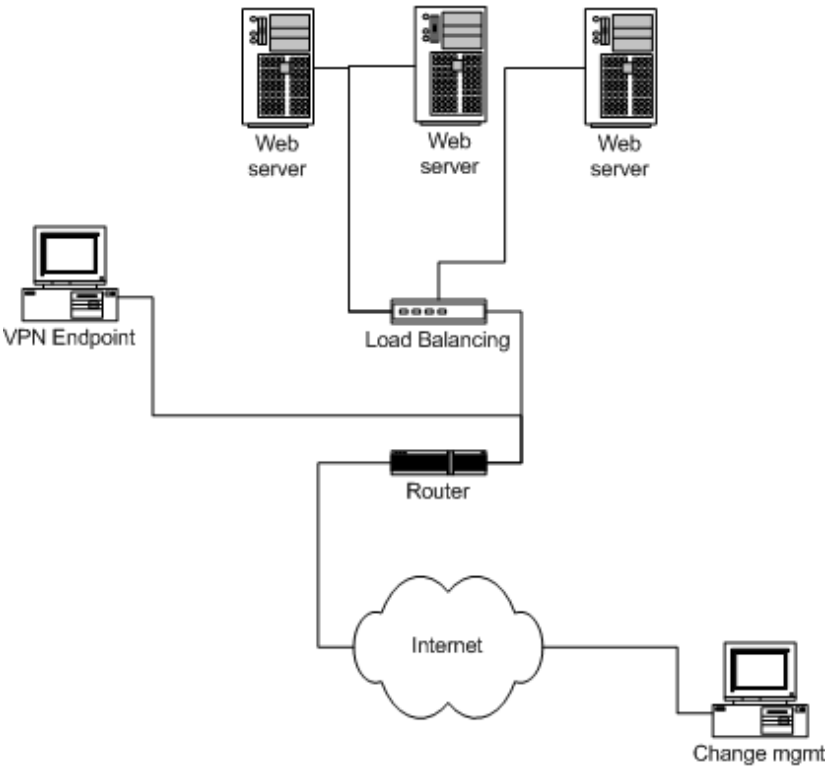


Figure 5.2: Adding a VPN to the Web farm network.

In this particular example, the VPN endpoint is inside the firewalled network (assume for a moment that the router is, in fact, the firewall). This setup ensures that your change management workstation, which would serve as the other end for the VPN, can securely access the configuration information in both the load-balancing device and the router.

Figure 5.3 illustrates another, more flexible, possibility. Here, your company network firewall forms a VPN connection to the remote firewall, securing all traffic between the two networks—the traditional use of a VPN. Your change management workstation is free to access the remote subnet as if it were local, logging on to the firewall/router's internal interface to collect management information and accessing the load-balancing device's front-end interface. The boldface line in the diagram represents the VPN-secured portion of the communications path.

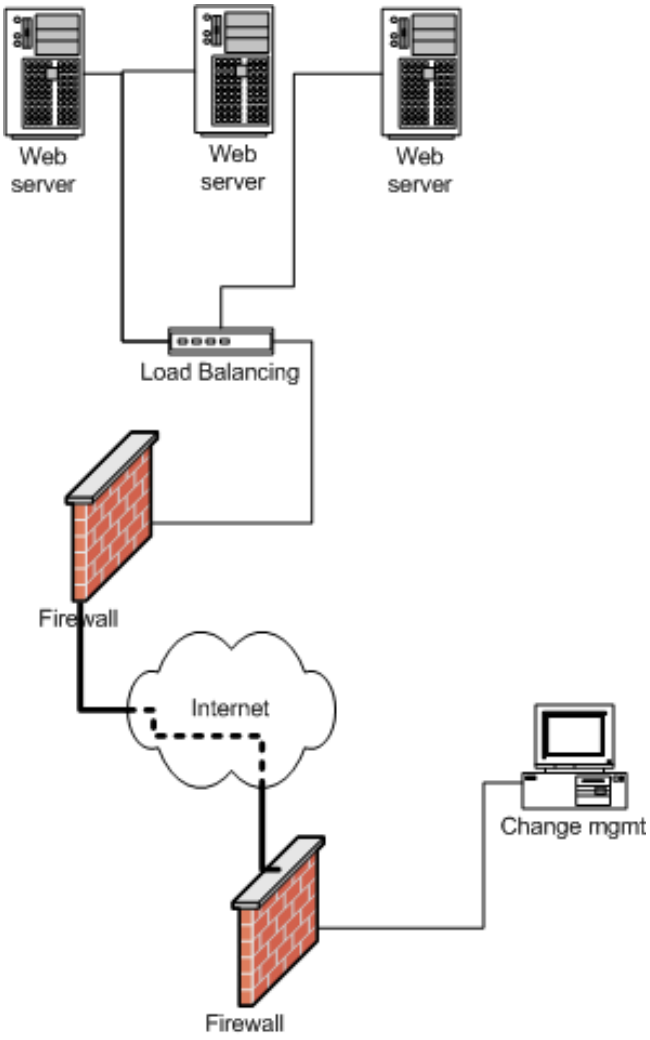


Figure 5.3: Firewalls as VPN endpoints.

The moral of this story is that management of remote network devices is not only possible but is fairly easy. It might take a little bit of time and effort to get everything set up, but in the end, it should be largely transparent and perfectly secure.

Topic 6: Enterprise Network Device Management

Q 6.1: Our enterprise has thousands of network devices and we're always adding new ones. How can we ensure that a change management solution will accommodate future devices?

A: A little bit of common sense and some investigation will generally provide a satisfactory answer to this question. What you need to do is evaluate each solution for its current breadth of device support, check up on how the solution is architected to accept changes, and make some educated guesses about how the solution will grow in the future.

Intended for Broad Support

On the common sense side, look at the ties each change management solution has. Take CiscoWorks, for example, which is made by Cisco. It's a good bet that any future Cisco equipment will be addressed by an update or add-on to CiscoWorks. It's also a good bet that any future Nortel equipment won't be handled at all. So, if you're truly in an all-Cisco shop, CiscoWorks might present an acceptable path for future support.

> 🕮 *Very* few companies use devices from only one vendor; see Question 5.1 for more details about the devices that are often forgotten.

Built for Device Expansion

On the investigation side, find out how each solution provides for future device support. Some solutions, for example, simply ship a new version of the product (upgrading you from 2.1 to 2.2) when a sufficient number of new devices are ready to be supported. Other solutions ship "packs" that add devices to the solution's list of supported devices. In general, the "pack" idea is preferable because the solution vendor can more easily provide support for one or two new devices rather than waiting until they've got enough to justify the release of a new version.

Evaluate the cost of device support, too. Some vendors ship base device support with their software and charge extra to provide support for other devices. Other vendors might offer additional device support on demand as part of a product maintenance agreement. Still others might simply accept requests for device support and post popularly requested device support for free on their Web sites.

Built for Long-Term Change

A larger, and more difficult to evaluate, question is how a change management solution will support radical new management technologies that come out in the future. Without question, every solution on the market today could be modified to support just about anything that comes along. It's the degree of modification that you should question. For example, suppose somebody invents BTEMP, the Better Than Ever Management Protocol. Device manufacturers universally jump on the BTEMP bandwagon and build it into their latest operating system (OS) updates. Now it's time to upgrade your management solution to take advantage of BTEMP's better security, lower bandwidth, or whatever. Some solutions will require a complete version upgrade, meaning you'll have to wait a while for it to be available, and then likely will have to deal with the bugs often associated with major changes in a software package. Other solutions might simply provide an add-on pack that takes advantage of BTEMP without modifying the base software.

You can get a feel for a solution's long-term change philosophy by looking at its current breadth of support. For example, a solution built entirely around Trivial File Transfer Protocol (TFTP) might not adapt well to entirely different technologies, although it could be built to easily accommodate new TFTP-based network devices. In contrast, a solution that already supports a broad variety of management technologies—TFTP, HyperText Transfer Protocol (HTTP), Simple Network Management Protocol (SNMP), or even proprietary management interfaces—is more likely to handle radical new management technologies without pain.

Evaluating Solutions' Growth Potential

As a part of your solution evaluation process, put together a grid that addresses future growth. Table 6.1 provides an example of such a grid.

Criteria	Product A	Product B	Product C
Existing breadth of support	Broad—12+ vendors	Moderate—8 vendors; mostly higher-end devices	Narrow—Vendor specific
Acceptance for new devices	Very good—Downloadable support packs	Very good—Downloadable support packs	OK—Ships a new version to support new device generations
Cost for new devices	Popular requests are added to the download site for free; moderate charges for immediate custom add-ons	Requires yearly maintenance agreement for popular requests; additional fees for custom add-ons	Included with new version; minor upgrades are free, major version upgrades have a cost
Major changes	Good—Already supports a broad range of management techniques, implying that support packs contain most of the device interface logic	Poor—Only supports TFTP/SNMP; manufacturer states that other techniques would require base code changes; device interface logic is all within the main program, not the support packs	Provides full support for vendor's existing hardware; will provide immediate support for any changes in vendor's management techniques

Table 6.1: Grid to evaluate solutions' growth potential.

With a chart like this one, you can easily evaluate solutions' growth capabilities against your specific needs and concerns. For example, if you're in a company that only uses network devices from one vendor, than that vendor's product (Product C in the chart that Table 6.1 shows) would be a great solution. It's likely that the product will always parallel the vendor's device technologies, providing a reasonable means of managing those devices well into the future.

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 change 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 underscore.

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.

Q 6.3: We're planning to use TACACS+ to consolidate authentication to hundreds of network devices. Is there anything that we need to be aware of?

A: TACACS+ is great for consolidating authentication. In fact, that's pretty much why it exists. However, you need to be aware of a few vulnerabilities and some limitations.

A Brief History

TACACS (without the +) was an early military protocol, first documented in Request for Comment (RFC) 1492. Cisco adopted TACACS as a sort of favorite protocol, and released XTACACS in 1990. This new version incorporated non-RFC proprietary extensions to TACACS, making it more suitable for use with Cisco devices. In 1998, Cisco released TACACS+. Support for TACACS+ was included in Cisco IOS 11.2 and later, and allowed the tasks of Authentication, Authorization, and Accounting (AAA) to be divided among several TACACS+ servers for better scalability.

TACACS+ is a proprietary Cisco protocol, although most of its details are publicly available. TACACS+ isn't fully backward-compatible with TACACS and XTACACS, so it's important to distinguish which you're using. The biggest caveat with TACACS+ is that it's primarily supported on Cisco devices; if you assumed that TACACS+ enjoys the wide support of similar protocols such as RADIUS, be sure you check your network devices to make sure they offer TACACS+ interoperability.

Pretty much nobody supports TACACS and XTACACS anymore, including Cisco, so I'll focus on TACACS+ for this tip.

How TACACS+ Works

TACACS+ is a client/server protocol. Clients, such as routers and network access devices, send AAA requests to a TACACS+ server, which is typically implemented as a UNIX daemon or a Windows background service. The server then processes and responds, if appropriate, to the request. The responding server represents the first vulnerable point in a TACACS+ implementation. First, whoever controls that service controls your enterprise-wide device authentication. On a Windows TACACS+ server, which will typically use the server's local user accounts or a domain for authentication, whoever controls the creation and modification of those accounts controls access to your network devices. Similarly, UNIX TACACS+ servers often keep their user lists in a configuration file, and whoever controls that file controls your network devices.

TACACS+ packets use an encryption technique to encrypt client/server messages, and the encryption key is a pre-shared secret, or password, entered on both the client and server. However, TACACS+ does not implement packet integrity checking, meaning packet capture and replay is a potential vulnerability, although the encryption limits the effectiveness of these attacks.

TACACS+ also uses session numbers to track AAA communications with clients. These session numbers are nonrandom and are in fact limited to 1, 2, or 3 for authentication, authorization, and accounting. Another portion of the session ID number is determined by a cryptographic algorithm, which is also nonrandom. With a large amount of TACACS+ traffic on the network, the randomness of these session IDs decreases, potentially making it easier for an intruder to mess with packets. For example, a packet could be switched from "authorization" type to "authentication" type. However, the server would find the remainder of the packet's structure incorrect for the new type (that structure having been based on the original type) and potentially crash, suffer a buffer overrun, or some other unexpected behavior. Note that only the body of the TACACS+ request is encrypted; the header is not, making it easy for someone with physical access to the network to capture, modify, and replay packets in an attempt to crash the server.

TACACS+ implementations before version 1.78 used User Datagram Protocol (UDP) packets to carry authentication requests, which is less reliable than the Transmission Control Protocol (TCP) connections used by newer versions. Ideally, you should configure your TACACS+ implementation to use a port other than TCP 49 (the default) to help obfuscate the service and thwart potential attackers.

Perhaps the biggest vulnerability in TACACS+ is the relative weakness of the overall encryption mechanism. It's possible for someone with physical network access to capture an authentication request from a client and modify it. This request could be submitted to the server and accepted; the server's reply would be encrypted, but because the clear text of that reply would be known, it would become much easier to break the encryption. Because the encryption is based on a shared secret that is rarely changed, breaking the encryption once would reveal the key necessary to break it from then on. Thus, a good practice is to regularly change the shared secrets used by TACACS+ clients to communicate with the server.

Accounting also presents replay vulnerabilities. Because the TACACS+ accounting packets aren't sequenced, they could be captured and replayed, essentially corrupting the TACACS+ accounting log and potentially even used as a denial-of-service (DoS) attack to prevent legitimate accounting packets from being processed—essentially disguising information.

Risk Analysis

How likely are any of these attacks? Probably not very, especially if your physical network containing the TACACS+ servers and clients is reasonably secure. However, it's always useful to know where the holes are so that you can at least watch out for exploits.

> 🕮 For more information, check out the Cisco TACACS+ document at ftp://ftpeng.cisco.com/pub/tacacs/tac-rfc.1.78.txt. There's also a good white paper about TACACS+ flaws and vulnerabilities at http://www.securityfocus.com/bid/1294, which provides more detail than I've given here. You can also read the TACACS FAQ at http://www.de.easynet.net/tacacs-faq/tacacs-faq.html.

Q 6.4: How can we ensure that our devices' software is consistent and up to date?

A: Managing your devices to ensure that they always contain the most recent version of their operating system (OS) software is becoming more important. Recent virus attacks that targeted Cisco routers directly have made it apparent that network devices—which were once largely ignored by virus authors and other hackers—are now fair game.

Interestingly, the very existence of a software patch for your routers makes them more vulnerable. The Blaster virus, which targeted Microsoft Windows-based computers, is an excellent example of how virus authors are taking advantage of lax patch management. A security firm notified Microsoft of a flaw in their remote procedure call (RPC) code that could allow an attacker to take complete control of a computer over TCP port 135. The notification was made quietly to Microsoft, which released a patch. The patch documentation did not describe in detail how the patch worked or what the original problem was. However, the patch contained only one replacement DLL file. The authors of the Blaster worm simply compared the new file to the old, then reverse-engineered the differences to determine what had been patched. They then wrote a worm to take advantage of the security hole on unpatched systems, knowing that most systems remain unpatched—often until the next service pack is released.

The Blaster worm has no direct effect on network devices, but the fact that virus authors are now using patches to develop their viruses should be a wake-up call for everyone in the IT industry: the minute a patch is released, apply it. This uncompromising approach means that you must ensure that all your network devices have a consistent, up-to-date software configuration.

Getting TFTP Ready

One way to ensure that your routers and other devices are up to date is to deploy a TFTP server on your network. This server can be used to house new updates; most devices can retrieve the update from the TFTP server. Start implementing this process by preparing the TFTP server. If you've got a spare Red Hat Linux server handy, you can run the following commands to make it a TFTP server:

```
# mkdir /tftpboot

# chmod 666 /tftpboot

edit file /etc/inetd.conf, uncomment line
tftp dgram udp wait root /usr/sbin/tcpd in.tftpd

# killall -1 inetd
```

Next, copy your update files—the binary files provided by your device manufacturer—into the TFTP folder and give them the appropriate permissions:

```
# cp update.bin /tftpboot/update.bin
# chmod 666 /tftpboot/update.bin
```

Preparing Your Devices

Take the time to make a backup of your devices before proceeding. You will want a backup of the entire NVRAM, especially the running configuration. After you finish upgrading the devices' firmware, reload the current configuration to get the device back up and running.

Upgrading the Firmware

The following code provides an example of commands to use to upgrade a Cisco 2501 router to a new version of the Cisco IOS:

```
Username: login
Password: *******
myrouter>en
Password: *******
myrouter#sh ver
```

The router would then display its version banner and other information. To begin the upload process, use the following commands:

```
myrouter#sh flash

System flash directory:
File  Length   Name/status
  1   5895768  c2500-is-l.112-14
[5895832 bytes used, 2492776 available, 8388608 total]
8192K bytes of processor board System flash (Read ONLY)
```

Next, tell the router to save its configuration via TFTP. In the following example, the TFTP server is at 192.168.1.100:

```
myrouter#wri n
Remote host [192.168.1.100]? 192.168.1.100
Name of configuration file to write [myrouter-confg]? myrouter-
confg
Write file myrouter-confg on host 192.168.1.100? [confirm]
Building configuration...

Writing myrouter-confg !! [OK]
```

With the configuration safely written, you can upload a new configuration into the router:

```
myrouter#conf n
Host or network configuration file [host]?
Address of remote host [192.168.1.100]? 192.168.1.100
Name of configuration file [myrouter-confg]? myrouter-confg
Configure using myrouter-confg from 192.168.1.100? [confirm]
Loading myrouter-confg from 192.168.1.100 (via Ethernet1): !
[OK - 1918/32723 bytes]
```

Next, write the current flash memory out to TFTP as well:

```
myrouter#copy flash tftp

System flash directory:
File  Length   Name/status
  1   5895768  c2500-is-l.112-14
[5895832 bytes used, 2492776 available, 8388608 total]
Address or name of remote host [255.255.255.255]? 192.168.1.100
Source file name? c2500-is-l.112-14
Destination file name [c2500-is-l.112-14]? c2500-is-l.112-14
Verifying checksum for 'c2500-is-l.112-14' (file # 1)...  OK
Copy 'c2500-is-l.112-14' from Flash to server
  as 'c2500-is-l.112-14'? [yes/no]yes
```

The router will display "!" as it uploads the configuration to the server, then it will provide the following output:

```
Upload to server done
Flash copy took 00:01:18 [hh:mm:ss]
myrouter#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
myrouter(config)#config-register 0x2101
myrouter(config)#^Z
myrouter#sh ver
```

Again, the router will display its configuration information, then you'll issue the write command to reload the router:

```
myrouter#wri
Building configuration...
[OK]
myrouter#reload
Proceed with reload? [confirm]
```

This data is what you will see as the router reloads and restarts. The router will be in boot mode, running the boot-ROM version of its OS. The following is the simplified version of the IOS and the flash memory in read/write mode:

```
Username: login
Password:
myrouter(boot)>en
Password:
myrouter(boot)#sh ver
Cisco Internetwork Operating System Software
IOS (tm) 3000 Bootstrap Software (IGS-BOOT-R), Version 11.0(10c),
RELEASE SOFTWA
RE (fc1)
Copyright (c) 1986-1996 by cisco Systems, Inc.
Compiled Fri 27-Dec-96 17:33 by loreilly
Image text-base: 0x01010000, data-base: 0x00001000

ROM: System Bootstrap, Version 11.0(10c), SOFTWARE
```

The important piece of the configuration information that you're looking for is:

```
myrouter uptime is 1 minute
System restarted by reload
Running default software

cisco 2500 (68030) processor (revision L) with 2048K/2048K bytes
of memory.
Processor board ID 08363366, with hardware revision 00000000
X.25 software, Version 2.0, NET2, BFE and GOSIP compliant.
2 Ethernet/IEEE 802.3 interfaces.
2 Serial network interfaces.
32K bytes of non-volatile configuration memory.
8192K bytes of processor board System flash (Read/Write)

Configuration register is 0x2101
```

This data tells you that you're running from the device's internal memory, and you can start working with the flash memory:

```
myrouter(boot)#copy tftp flash

System flash directory:
File  Length   Name/status
  1   5895768  c2500-is-l.112-14
[5895832 bytes used, 2492776 available, 8388608 total]
Address or name of remote host [255.255.255.255]? 192.168.1.100
Source file name? c2500-is-l.112-15.bin
Destination file name [c2500-is-l.112-15.bin]? c2500-is-l.112-
15.bin
Accessing file 'c2500-is-l.112-15.bin' on 192.168.1.100...
Loading c2500-is-l.112-15.bin from 192.168.1.100 (via Ethernet1):
! [OK]

Erase flash device before writing? [confirm]
Flash contains files. Are you sure you want to erase? [confirm]

Copy 'c2500-is-l.112-15.bin' from server
  as 'c2500-is-l.112-15.bin' into Flash WITH erase? [yes/no]yes
Erasing device... eeeeeeeeeeeeeeeeeeeeeeeeeeeeeee ...erased
Loading c2500-is-l.112-15.bin from 192.168.1.100 (via Ethernet1):
```

The router will display "!" as it loads the configuration file. Once it is finished, it will verify the file and allow you to work with it:

```
[OK - 5895436/8388608 bytes]

Verifying checksum...  OK (0x61A0)
Flash copy took 0:03:02 [hh:mm:ss]
myrouter(boot)#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
myrouter(boot)(config)#config-register 0x2102
myrouter(boot)(config)#^Z
myrouter(boot)#sh ver
Cisco Internetwork Operating System Software
IOS (tm) 3000 Bootstrap Software (IGS-BOOT-R), Version 11.0(10c),
RELEASE SOFTWA
RE (fc1)
Copyright (c) 1986-1996 by cisco Systems, Inc.
Compiled Fri 27-Dec-96 17:33 by loreilly
Image text-base: 0x01010000, data-base: 0x00001000

ROM: System Bootstrap, Version 11.0(10c), SOFTWARE

myrouter uptime is 5 minutes
System restarted by reload
Running default software

cisco 2500 (68030) processor (revision L) with 2048K/2048K bytes
of memory.
Processor board ID 08363366, with hardware revision 00000000
X.25 software, Version 2.0, NET2, BFE and GOSIP compliant.
2 Ethernet/IEEE 802.3 interfaces.
2 Serial network interfaces.
32K bytes of non-volatile configuration memory.
8192K bytes of processor board System flash (Read/Write)

Configuration register is 0x2101 (will be 0x2102 at next reload)
```

With everything in place, you can reload the router. The configuration has been changed, so the system will prompt you to save the configuration; don't bother—simply let the router restart:

```
myrouter(boot)#reload

System configuration has been modified. Save? [yes/no]: no
Proceed with reload? [confirm]
```

The router will come up normally; reload the configuration that the router should be using:

```
Username: login
Password:
myrouter>en
Password:
myrouter#sh ver
```

Again, the version information will be displayed. You can now load the configuration file from the TFTP server:

```
myrouter#conf n
Host or network configuration file [host]?
Address of remote host [255.255.255.255]? 192.168.1.100
Name of configuration file [myrouter-confg]? myrouter-confg
Configure using myrouter-confg from 192.168.1.100? [confirm]
Loading myrouter-confg from 192.168.1.100 (via Ethernet1): !
[OK - 1953/32723 bytes]

myrouter#wri
Building configuration...
[OK]
myrouter#exit
```

You've completed the software upgrade. This sequence is fairly complex and requires that you verify that each step has completed properly. In other words, this process is not easily scripted. However, software products such as AlterPoint DeviceAuthority and Cisco CiscoWorks have the ability to run through a script like this automatically, ensuring that each step completes properly. With such software, you can upload new software to multiple devices at the same time.

The benefit of using a dedicated software solution is that you don't have to worry about individually backing up each device, running the script, and verifying that everything completes properly. The software performs these tasks for you, making it easier to manage a large number of devices. In fact, in a large enterprise with hundreds of devices, an automated software solution can be the only feasible way to keep everything updated.

Q 6.5: How can we manage device changes in real-time?

A: Device change management in real-time is a tricky process. Very few devices offer any sort of events-based notification other than standard Simple Network Management Protocol (SNMP) traps sent to a management console. In other words, routers don't typically provide details of a configuration change; instead, they tend to notify a management console via SNMP trap that someone has logged on and done *something*. The notifications lack the detail that a change-management solution typical requires.

SNMP traps are a start—they at least provide *some* kind of notification that something has happened. To configure a Cisco IOS device (except some remote access devices) to send SNMP traps for most configuration-related operations, execute the following commands on the device:

```
snmp-server host 192.168.1.102 public config bgp tty
snmp-server enable traps config
snmp-server enable traps bgp
snmp-server enable traps tty
```

Be sure to provide the correct IP address for the SNMP workstation, which, in this example, is 192.168.1.102. Shortly, we'll explore how this process can help your configuration-management process.

Another technique is to use TACACS+ accounting information, which is available on most Cisco devices (Cisco owns the TACACS+ protocol) and on some other devices; Remote Authentication Dial-In User Service (RADIUS) accounting can also be used, although it tends to provide less-specific information. The following configuration file enables accounting:

```
aaa new-model

radius-server host  192.168.1.102  auth-port 1812  acct-port 1813
key secret
radius-server retransmit 3     !
radius-server timeout  6       !  default = 5 seconds
radius-server deadtime 1       ! default = ? minutes

tacacs-server host 192.168.1.102 key secret
tacacs-server timeout 6        ! default

aaa authentication login   default radius local
aaa authentication login   NO_AUTHEN none
aaa authentication enable  default group radius enable

aaa authorization network  default group radius if-authenticated
aaa authorization exec     default group  radius
if-authenticated
aaa authorization exec     NO_AUTHEN
aaa accounting exec        default start-stop tacacs+
aaa accounting commands 0  default start-stop tacacs+
aaa accounting commands 1  default start-stop tacacs+
aaa accounting commands 12 default start-stop tacacs+
aaa accounting commands 14 default start-stop tacacs+
aaa accounting commands 15 default start-stop tacacs+
aaa accounting network     default start-stop tacacs+
aaa accounting connection  default start-stop tacacs+
aaa accounting system      default start-stop tacacs+

line con 0
login authorization  NO_AUTHEN
```

The idea for using either TACACS+ or SNMP is that if you have a change-management solution capable of supporting the functionality, the solution can either receive SNMP traps or analyze the TACACS/RADIUS log on a periodic basis (or even directly receive TACACS/RADIUS accounting information). The traps or log information don't provide the actual change data, but they serve as a trigger to let the solution know that someone, for example, logged off of the device. This trigger tells the solution to log on via TFTP (or whatever means for which it is enabled) and retrieve the devices' current configuration.

By continually retrieving the configuration whenever someone logs off, you're sure to have a complete record of changes that were made over time. This ability contrasts with solutions that simply retrieve the device configuration on a scheduled basis, because such devices can easily miss incremental changes performed between the retrieval intervals.

Thus, there isn't really a way to perform real-time change management; however, with an automated change-management solution capable of scanning accounting logs or receiving SNMP traps, you *can* enable near-real-time change management by continually downloading and storing device configurations whenever anyone performs an action that might result in a configuration change. As you're evaluating change-management solutions, be sure to consider this capability in your features comparison.

Q 6.6: What is the best way to simplify the process of reconfiguring more than one hundred network devices?

A: The ability to reconfigure multiple network devices as easily as a single device—or even more easily, if possible—is important in organizations that support dozens of network devices. Some of the most basic security practices, in particular, can be more easily implemented through an automated configuration capability.

Failing to change devices' Simple Network Management Protocol (SNMP) community strings to something other than the default "public" is like failing to lock the doors on your car—eventually, someone's going to steal it or something in it. However, many companies change those SNMP strings only once—when the device is first placed into service. As a result of the complexity and time required to change the SNMP strings on literally hundreds of routers, switches, and other devices, organizations simply don't do it. However, the same organizations require users to change their passwords every 30 days because old passwords are more likely to be compromised. But old SNMP strings are just as likely to be compromised and are more likely to have a severe impact if they are compromised. Whether the changes are devices' administrator passwords, access permissions, or routing tables across a group of devices, the ability to make consistent changes to a large number of devices is a very real need in large organizations.

The complexity of this task requires more than simply write a script or two; you need third-party software to provide the correct combination of workflow, configuration deployment, change validation, and more. Solutions from AlterPoint, TripWire, Ecora, and Voyence can provide all or part of a mass-configuration-management solution. You'll need to carefully evaluate these companies' offerings to determine which solutions meet your business requirements.

For example, AlterPoint's DeviceAuthority provides an Update Module that allows you to automate the process of changing several devices' configurations at the same time. Figure 6.1 shows the module's overview page for a password-update job. You can see that only a portion of the *inventory*—the list of devices that the software knows about—will be modified; the Austin Inventory selection suggests that only devices in an Austin office will be modified by this job. The job also includes a workflow, which I'll examine in more detail in a bit. Notice that the job includes a notification capability, which will provide email notifications to selected users when the update is run, providing immediate status feedback.

Software vendors use different techniques for managing multiple device updates. DeviceAuthority uses a "job" to represent a single set of changes, and provides a "workflow" model to handle devices that are offline or not available when the update runs. Other software uses different terms. When comparing software packages, look for details like the ability to send email notifications, have update scripts created for you, and handle multiple devices simultaneously.

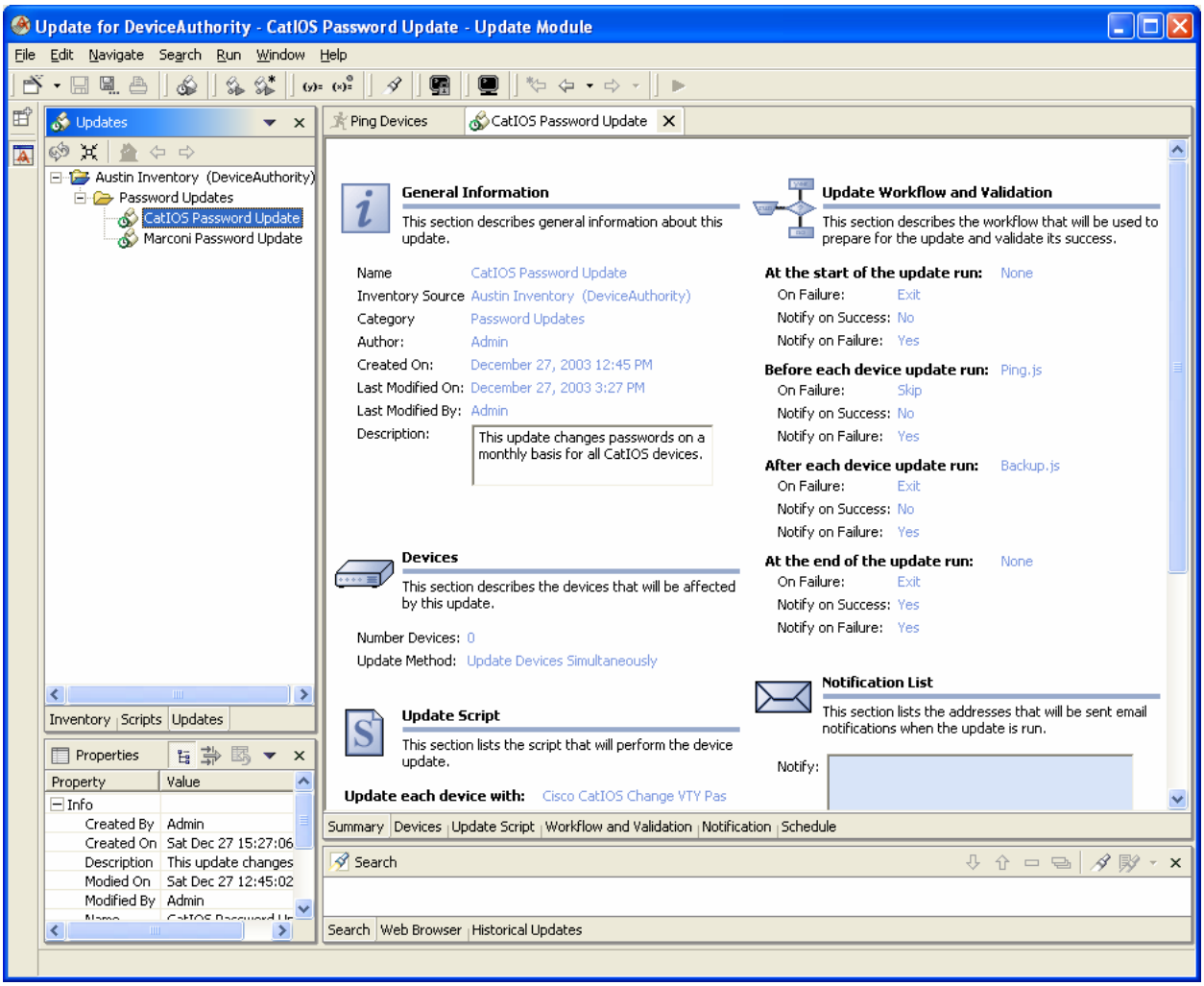


Figure 6.1: Deploying a configuration change to multiple devices at the same time.

A useful function is the ability to specify different actions for various parts of the update process. With DeviceAuthority, you can specify a different update script for the start of the update process, before each device is updated, after each device is updated, and after the entire job finishes. As Figure 6.2 shows, a short script can be used to ping each device before trying to update it; a failure will cause that device to be skipped and the failure to be logged. After updating each device, the device configuration is backed up, ensuring that the new configuration is safely stored and that the organization's overall configuration management process is maintained.

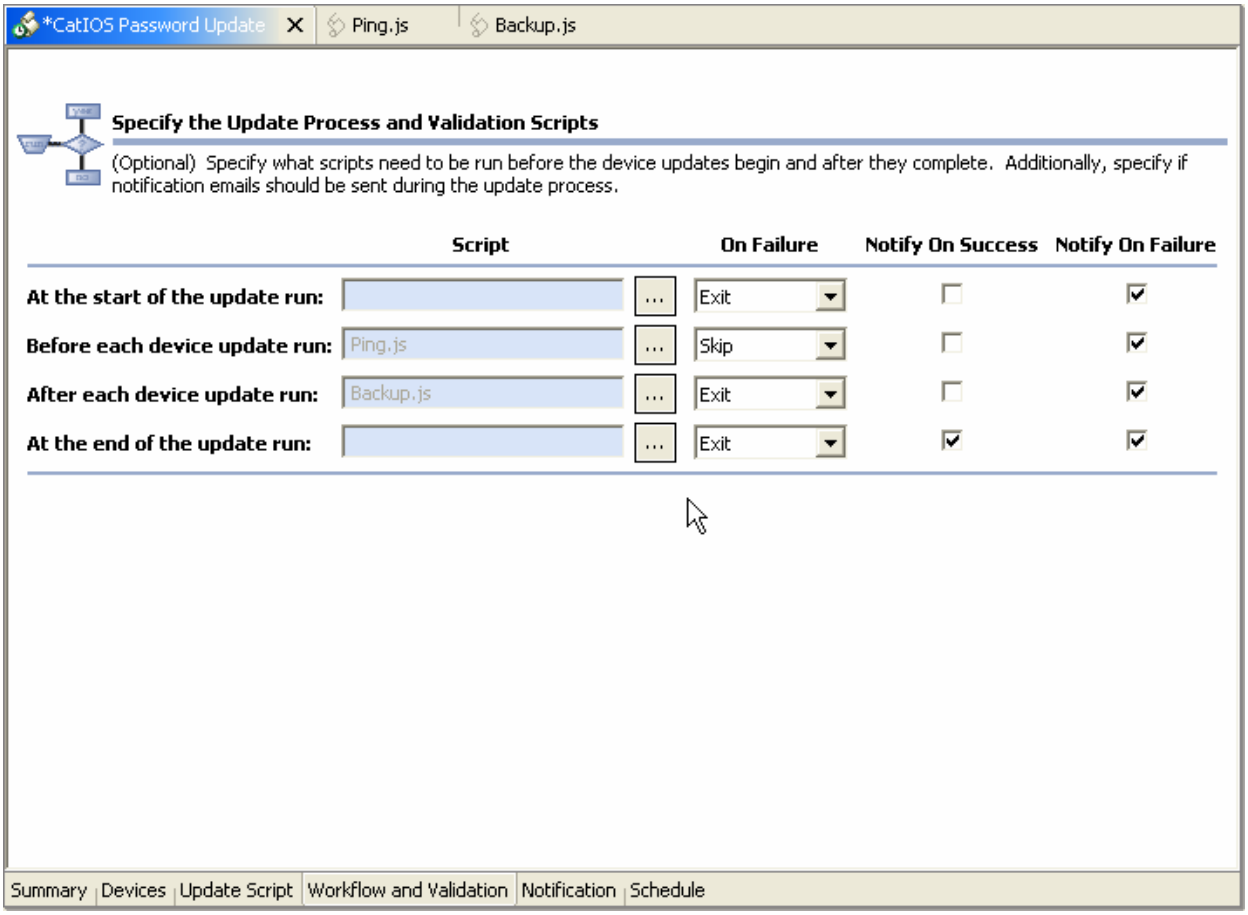


Figure 6.2: Customizing deployment workflow.

Figure 6.3 shows another important feature—the ability to provide detailed status reports for each step of the update process. Ideally, this status report is available both from the software's management interface, as Figure 6.3 shows, and through email notifications. The email notifications help provide real-time updates and status checks to administrators throughout your organization, helping ensure that everyone is kept up-to-date with the status of the change.

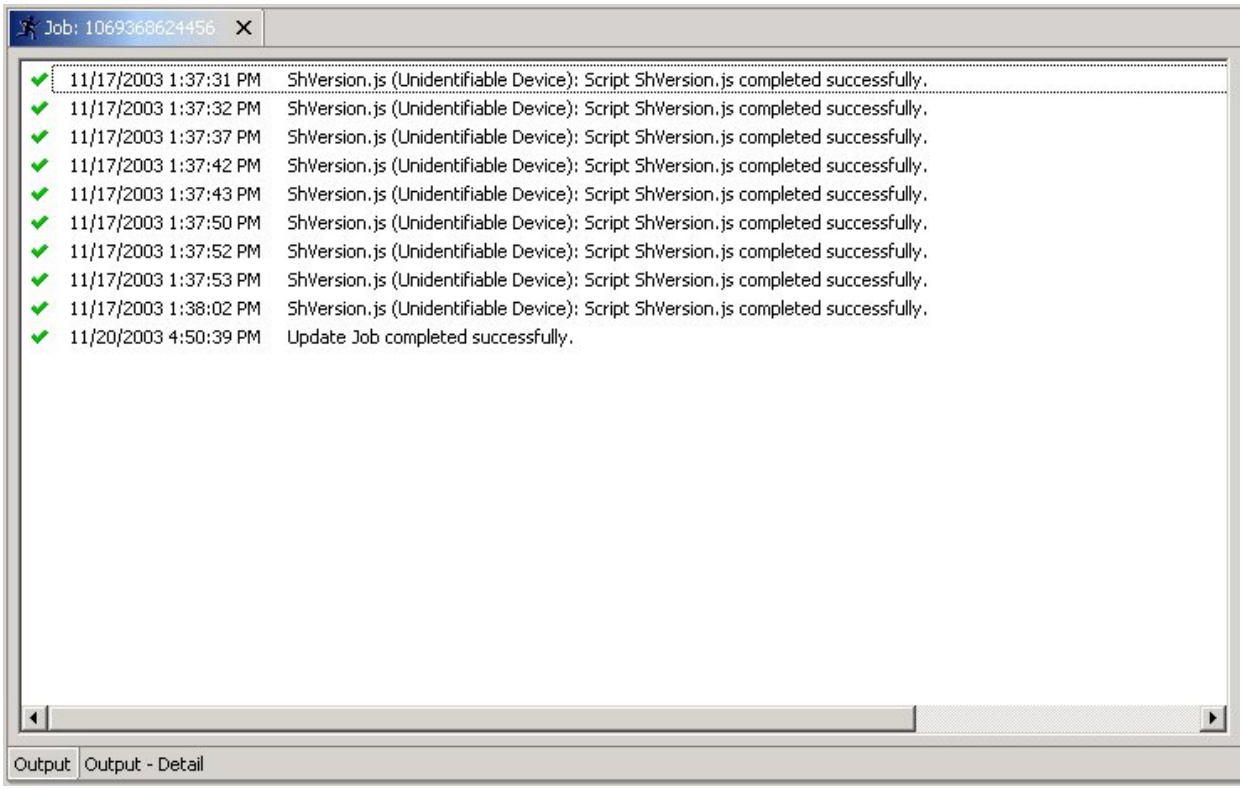


Figure 6.3: Viewing update results in DeviceAuthority.

The following list highlights key features to look for in a software package:

- The ability to target specific devices for an update. For example, you might want to modify only devices at a certain office or only devices that are running a certain operating system (OS) version.
- The ability to have rudimentary workflow in the update job. This workflow should allow actions that skip a device or cancel the job entirely, providing the job with some intelligence to handle real-life network conditions.
- The ability to incorporate management workflow into the process. For example, the software might allow an administrator to create a job, then require another administrator to review and approve the job before the job can actually execute.
- The ability to schedule jobs. You might want certain updates to run during off-hours. You should also be able to specify blackout times when no jobs are allowed to run, helping ensure that an administrator doesn't accidentally schedule a job that restarts devices to run during a period in which devices must be available. These scheduling capabilities can help you meet service level agreements (SLAs) and maintain a working production environment.
- A small built-in library of utility scripts. These scripts perform useful work such as pinging devices to check for availability. Alternatively, they might be example scripts to help you more quickly generate the appropriate update scripts.

- A known scripting language. Most software includes a scripting capability that allows you to perform complex tasks in an automated fashion. This scripting should use a known, documented language—such as JavaScript—or at least provide a well-documented proprietary language that includes plenty of examples. Ideally, the software monitors your actions on a device and creates a script based on those actions that can then be deployed to multiple other devices.

Selecting a solution that employs configuration templates—or developing your own templates, as I discussed in Tip 3.8—is another way to ensure that updates to multiple devices are made smoothly, consistently, and with less error. The ability to use configuration templates might be another selection criterion for a device management solution.

Q 6.7: How do you configure new devices to have the same consistent configuration as existing devices?

A: Ideally, you will create templates and scripts, which can be deployed to as many devices as possible. By eliminating manual configurations steps, you'll also eliminate error and ensure consistency.

> ☞ Using scripts and templates also makes your change-management process and a peer review much more important. Any errors that creep into your scripts or templates will be propagated to all of your devices. Thus, it is important that someone checks and tests scripts or templates before they are used to modify multiple devices.

Creating such scripts can be a complex, time-consuming process—one that you might not find the time for in a busy environment. One trick is to find a software utility that will automatically create scripts for you. For example, Figure 6.4 shows the script-creation tool provided by AlterPoint's DeviceAuthority. This tool provides you with a Telnet session, which you use to connect to a device such as a router or switch. You can see the Telnet session in the upper half of the window. However, this tool is no ordinary Telnet client—it watches everything you type and everything that the device sends to you in response. In the lower half of the window, the tool compiles a JavaScript file that mimics your actions. The practical upshot of this feature is that you manually configure only *one* device, then you have a JavaScript file that you can deploy to configure multiple other devices in the same way. You can even edit the script, if desired, so that you can change device-specific information such as IP addresses.

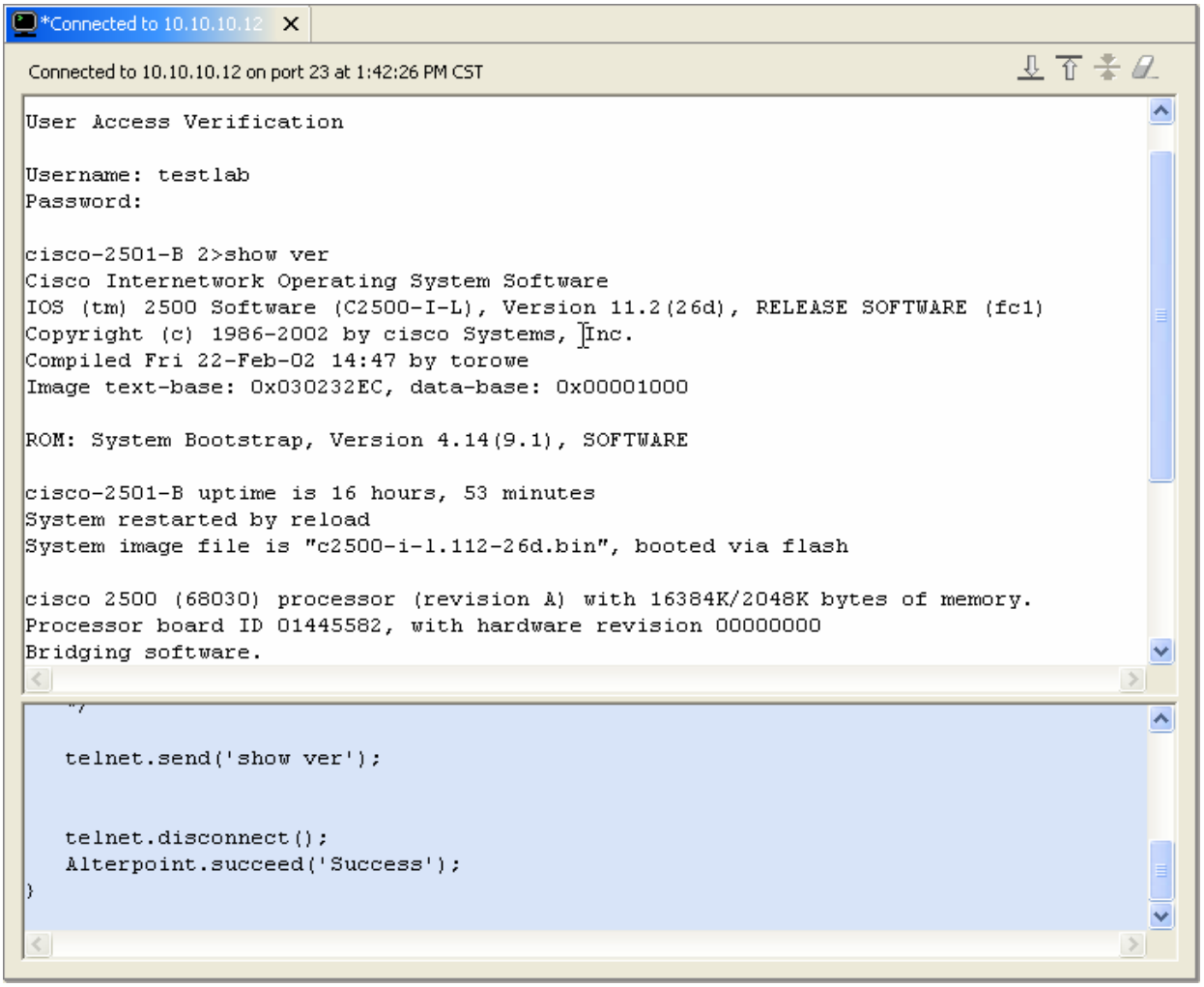


Figure 6.4: Employ a tool that records actions and creates a script automatically.

You can also create your own configuration templates without using complicated scripts simply by storing a configuration script in a text file. For example, I once worked with a company that deployed a dozen Cisco 1720 devices to their field offices and needed the devices configured to support a T1 or fractional T1 setup with an ISDN BRI backup interface. We developed a simple, standardized configuration script (or template) for these devices, which Listing 6.2 shows.

```
interface Serial0
backup delay 120 120
backup interface BRI0
no ip address
no ip directed-broadcast
encapsulation frame-relay IETF
keepalive 5
frame-relay lmi-type ansi
!
interface Serial0.1 point-to-point
description company.office.serialptp
ip address 172.24.5x.1 255.255.0.0
ip broadcast-address 172.24.5x.2
no ip directed-broadcast
frame-relay interface-dlci <DLCI> IETF
!
interface BRI0
description company.office.bri
ip unnumbered FastEthernet0
no ip directed-broadcast
encapsulation ppp
dialer idle-timeout 2147483
dialer wait-for-carrier-time 10
dialer map ip 38.1.1.1 name <PAP LOGIN> <DIAL NUMBER>
dialer load-threshold 1 either
dialer-group 1
isdn switch-type <SWITCH TYPE>
isdn spid1 <SPID1> <DN1>
isdn spid2 <SPID2> <DN2>
ppp pap sent-username <PAP LOGIN> password <PAP PASSWORD>
ppp multilink
!
interface FastEthernet0
ip address 192.168.1x.1 255.255.255.0
ip broadcast-address 192.168.1x.5
no ip directed-broadcast
!
ip classless
ip route 0.0.0.0 0.0.0.0 <DEFAULT GATEWAY> 1
ip route 0.0.0.0 0.0.0.0 38.1.1.1 2
ip route 38.1.1.1 255.255.255.255 BRI0
!
access-list 101 permit ip any any
dialer-list 1 protocol ip list 101
```

Listing 6.2: A simple device configuration script.

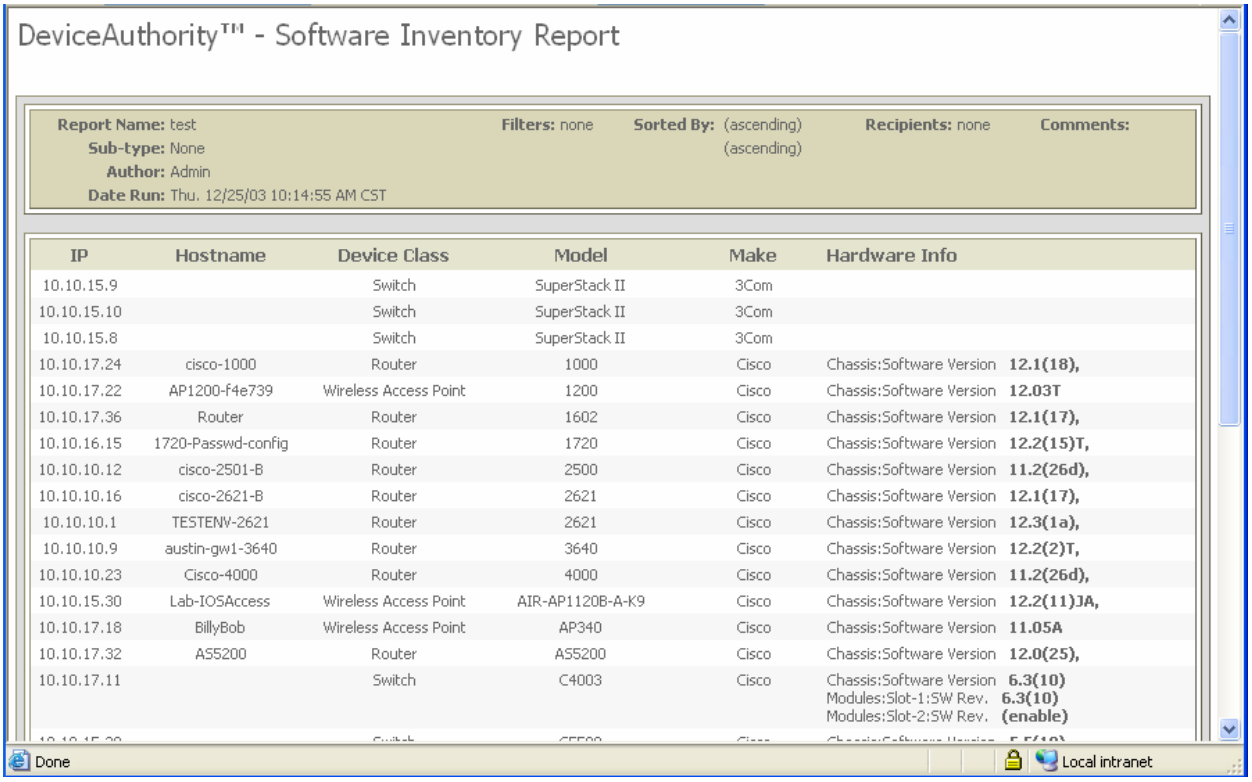
I've boldfaced the items you'll probably need to change in this template; italicized items are the names that this particular company changed from office to office.

Creating this type of template will help ensure that nothing is forgotten when configuring new devices, and that the new devices receive a configuration with standardized IP addressing, device names, interface names, and so forth.

📖 For more tips about how templates can save time and improve consistency, refer to Tip 3.8.

Q 6.8: Is there a way to quickly obtain serial numbers and other information from devices?

A: Most vendor-provided and third-party device-management software polls this information from devices and provides it for you. For example, Figure 6.5 shows a software inventory report, which you can use to see which chassis version each device is using. This information can be useful when determining which devices need updates or which devices might be affected by a particular problem that is unique to one chassis version.



DeviceAuthority™ - Software Inventory Report

Report Name: test
Sub-type: None
Author: Admin
Date Run: Thu. 12/25/03 10:14:55 AM CST

Filters: none **Sorted By:** (ascending) (ascending) **Recipients:** none **Comments:**

IP	Hostname	Device Class	Model	Make	Hardware Info
10.10.15.9		Switch	SuperStack II	3Com	
10.10.15.10		Switch	SuperStack II	3Com	
10.10.15.8		Switch	SuperStack II	3Com	
10.10.17.24	cisco-1000	Router	1000	Cisco	Chassis:Software Version **12.1(18),**
10.10.17.22	AP1200-f4e739	Wireless Access Point	1200	Cisco	Chassis:Software Version **12.03T**
10.10.17.36	Router	Router	1602	Cisco	Chassis:Software Version **12.1(17),**
10.10.16.15	1720-Passwd-config	Router	1720	Cisco	Chassis:Software Version **12.2(15)T,**
10.10.10.12	cisco-2501-B	Router	2500	Cisco	Chassis:Software Version **11.2(26d),**
10.10.10.16	cisco-2621-B	Router	2621	Cisco	Chassis:Software Version **12.1(17),**
10.10.10.1	TESTENV-2621	Router	2621	Cisco	Chassis:Software Version **12.3(1a),**
10.10.10.9	austin-gw1-3640	Router	3640	Cisco	Chassis:Software Version **12.2(2)T,**
10.10.10.23	Cisco-4000	Router	4000	Cisco	Chassis:Software Version **11.2(26d),**
10.10.15.30	Lab-IOSAccess	Wireless Access Point	AIR-AP1120B-A-K9	Cisco	Chassis:Software Version **12.2(11)JA,**
10.10.17.18	BillyBob	Wireless Access Point	AP340	Cisco	Chassis:Software Version **11.05A**
10.10.17.32	AS5200	Router	AS5200	Cisco	Chassis:Software Version **12.0(25),**
10.10.17.11		Switch	C4003	Cisco	Chassis:Software Version **6.3(10)** Modules:Slot-1:SW Rev. **6.3(10)** Modules:Slot-2:SW Rev. **(enable)**

Done Local intranet

Figure 6.5: Viewing devices' software versions using AlterPoint's DeviceAuthority.

If this inventory is incorporated into a change-management solution—as is the case with DeviceAuthority—the inventory is usually under change management as well. For example, suppose a company's routers were beginning to perform very poorly. Upon investigation, they realized that the routers simply didn't have enough memory installed to support the traffic they were trying to handle. The problem was that the routers had been purchased with fully populated memory slots, meaning that memory had at some point been removed. The company blamed a disgruntled, now-departed employee, but the damage was done and the memory was missing. Had the routers been under a change-managed inventory program, an administrator would have received immediate notification when the devices' hardware inventories changed, and the problem could have been caught—if not in time to save the memory, at least in time to fix the problem before it began impacting production operations.

Hardware inventory functionality, combined with change management, can also provide troubleshooting capabilities. For example, suppose a device suddenly starts to restart itself without warning. A glance at the hardware inventory report might reveal that the problem started shortly after adding a new interface card to a slot within the device. You might immediately suspect that the interface card was faulty, remove it, and have the problem fixed without a lengthy troubleshooting process.