



realtimepublishers.comtm

The Shortcut Guidetm To



Network Compliance and Security

AlterPoint

Don Jones

Chapter 2: Traditional Compliance Techniques	19
Compliance and IT.....	19
Foundation Technologies.....	20
Simple Network Management Protocol.....	20
TACACS and RADIUS	21
Syslog.....	23
Device Configurations	24
TFTP	26
Foundation Methodologies	27
ITIL Overview	27
Traditional Compliance Management.....	30
Monitoring Only	30
Point-in-Time Audits	30
Manual Configuration Review.....	31
Template-Based Provisioning.....	31
The Shortcomings of Traditional Compliance Management.....	34
Vendor-Specific	34
Lack of Reporting	34
Alerting, Not Enforcement.....	34
Lack of Logging and Auditing.....	34
Entirely Manual	35
Not Real-Time	35
Lack of Accountability	35
Not Continuous	36
Summary	36

Copyright Statement

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

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

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

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

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

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

Chapter 2: Traditional Compliance Techniques

How do we make sure we're compliant? It's an age-old question in the IT industry. As I mentioned in the previous chapter, *compliance* simply means obeying a set of rules; IT folks have been trying to obey rules long before legislative bodies such as the United States Congress and the European Union got into the act. Whether you're trying to comply with rules that relate to security, privacy, operational stability, or governance, knowing how to make your network compliant—and keep it that way—can be a complex task. In this chapter, we'll explore the traditional ways in which network administrators and engineers have dealt with compliance, and discuss how those ways help—and sometimes hinder—the overall compliance effort.

Compliance and IT

One of the key issues in compliance management is how you verify compliance. As I described in the previous chapter, testing the *end state* is usually the preferred method. For example, if you have a rule which states that only certain individuals should have access to a specific piece of information, you conduct a test to confirm that no other individuals are able to gain access. This idea is simple enough in theory, but even in non-IT areas, it can be difficult to actually implement. For example, suppose you have a room in your office to which only certain individuals should have access for security reasons. How do you *prove* that nobody else in the world has a key? Dealing with IT-related compliance can be even more complex because it's often extremely difficult, time-consuming, or even destructive to test the end state. For example, testing certain security technologies would require an attempt to *break* those technologies; if you're successful at doing so, you have not only proven that the technologies didn't work but also damaged your environment.

Thus, instead of testing the end state, companies often rely on policy-based compliance. They create policies and procedures that will, if followed, guarantee a compliant end state. Then they simply audit and test to make sure that the policies and procedures are being followed. For example, if you have a policy which states that only certain individuals have access to a given locked room, you can audit the number of keys that were made for the room's lock and inventory the keys that have and have not been issued. This method audits the policy of issuing keys only to authorized individuals; in theory, provided all keys are accounted for, the end state of a secure room will be guaranteed.

Most IT-based compliance, therefore, is based on compliant policies and procedures rather than on testing of end states. With network configuration management, this method of remaining in compliance is especially true, and several technologies exist to make policy auditing easier and more effective.

Foundation Technologies

Over the years, several technologies have been created to help make network device management easier. Many of these technologies also lend themselves in one way or another to compliance management, although their relationship to compliance can be less than obvious. Still, understanding how these core technologies work, and what they offer both to compliance and general management, is important to understanding how compliance management can be made more effective.

Simple Network Management Protocol

The Simple Network Management Protocol (SNMP) was designed to make centralized management of network devices easier. Generally, an SNMP-enabled network consists of one or more SNMP *management stations* and one or more SNMP-enabled devices, such as routers, switches, and so forth. Figure 2.1 illustrates a typical network.

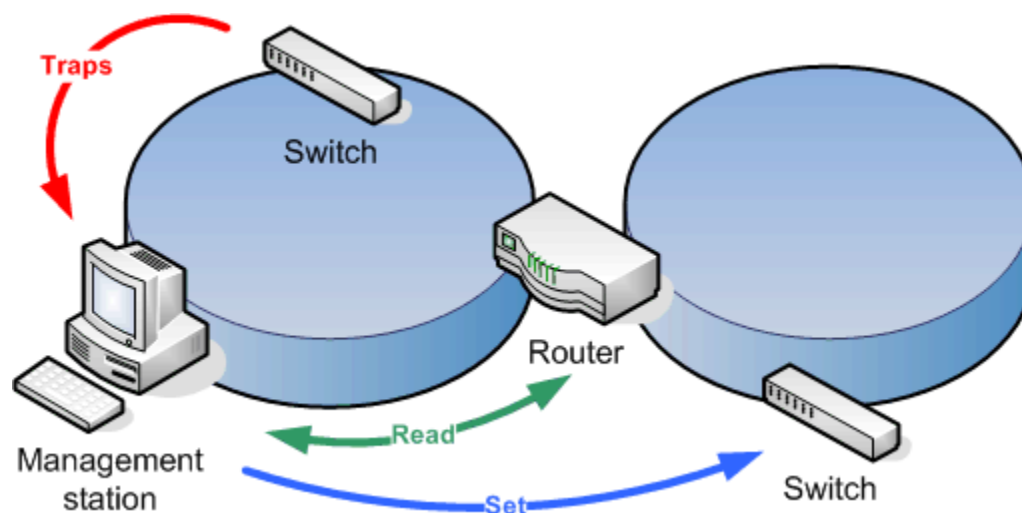


Figure 2.1: SNMP in a typical network.

As Figure 2.1 shows, the management station receives *traps*, or notifications, from managed devices. These traps often contain information about something that has just occurred, such as a configuration change, an error, and so forth. Management stations can process these traps and generate alerts for administrators or simply log the traps for future use. Management stations can also issue *reads* to devices, allowing the station to read certain configuration details from the device, or issue *sets*, which allow the station to change a device's configuration.

SNMP uses extremely simplistic security, primarily set through a *community string*. The community string is essentially a password; any management station possessing the correct community string can issue reads and sets to any managed device having the same community string. Newer versions of SNMP allow you to specify different community strings for read and set (or *write*) operations; this functionality provides for slightly more granular security control. For example, in a Cisco device's configuration you might see something similar to the following example:

```
Router#show running-config
....
....
snmp-server community public RO
snmp-server community private RW
....
....
```

This text specifies a community string of “public” for read operations, and a string of “private” for read-write operations. A drawback of these particular strings is that they're the defaults on almost every device in the world; thus, using “public” and “private” virtually guarantees that your devices' configurations will be available to anyone with SNMP software, regardless of whether a user is authorized.

From a compliance point of view, SNMP provides one important function—its traps allow devices to notify a central station when the devices' configuration might have changed. Any auditing activity can examine SNMP logs and, if traps are found, use those as a cue to perform a more detailed analysis on the devices involved.

TACACS and RADIUS

Terminal Access Controller Access Control System (TACACS) and Remote Access Dial-In User Service (RADIUS) were originally conceived as a means of centralizing remote user access to networks. In the world of network management, however, they've become an important way to control administrative access to devices and to log access to devices. A typical network containing TACACS or RADIUS (although TACACS and RADIUS are different, the two serve the same function and work similarly enough that they can be discussed as a single technology) might look something like the network that Figure 2.2 illustrates.

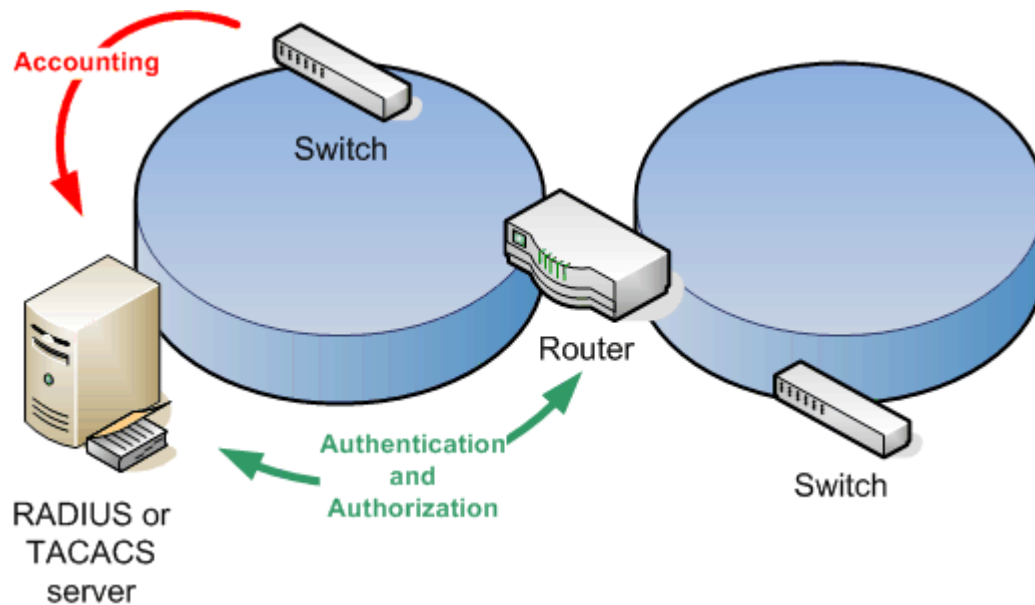



Figure 2.2: TACACS or RADIUS on a typical network.

When anyone attempts to gain access to a network device's configuration, the network device (if configured properly) passes the user's credentials to the RADIUS/TACACS server; the server responds with an *authentication* message indicating whether the user is who they claim to be (that is, the password is correct) and might also provide *authorization* information indicating which permissions the user has on the device. The goal of this process is centralization—rather than configuring each device with its own list of usernames and permissions, that information can be consolidated onto a single server and each device can simply look to that server for the information.

RADIUS/TACACS also provide valuable *accounting* features, allowing devices to send status information, security messages, and so forth to the RADIUS/TACACS server for long-term logging. This functionality is similar to that provided by SNMP traps, although most devices can generate more detailed RADIUS/TACACS messages than those provided by SNMP traps.

Devices must of course be configured to use TACACS or RADIUS. The following text shows an example of a script that configures a Cisco device to use a TACACS server.

```
aaa new-model
aaa authentication ppp default if-needed group tacacs+ local
tacacs-server host 10.1.2.3
tacacs-server key goaway
interface serial 0
ppp authentication default
```


 TACACS is popular in Cisco environments, as Cisco more or less owns the TACACS standard (which is currently referred to as TACACS+). RADIUS is a more vendor-neutral option.

TACACS/RADIUS plays an important role in compliance management. First, it is easier to maintain a compliant set of business policies (for example, user access permissions) when those policies are configured in one place—the TACACS/RADIUS server—than if they were configured in multiple places—on each individual device. Further, TACACS/RADIUS accounting logs can provide valuable auditing information, informing an auditor that a particular device might have been modified and leading them to review that device’s configuration in more detail.

Syslog


Syslog is a mature logging technology supported by almost all network devices. Because network devices rarely have their own hard drives or other mass-storage devices, they are unable to generate and maintain local log files. The Syslog protocol was developed so that devices could transmit log entries to a remote server, which stores them for long-term use. Syslog files generally contain more detailed information than TACACAS/RADIUS or SNMP logs; logging detail can often be configured within a device to log packet-level information for debugging purposes, if desired.

As with SNMP and TACACS/RADIUS, devices must be configured to utilize a Syslog server. The following examples shows a Cisco switch configuration, illustrating that logging is enabled and directed to a server at IP address 192.168.1.100.

 The logging level and severity are configurable, controlling the number of log messages that will be generated.

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

Like TACACS/RADIUS and SNMP, Syslog provides valuable logging and auditing information for compliance management efforts.

 It might seem like overkill for a single device to generate SNMP traps, TACACS/RADIUS accounting logs, *and* Syslog logs, but many organizations configure their devices to do just that. Syslog provides a continuous, low-level logging effort; TACACS/RADIUS accounting logs tend to focus on access control and administrative functions; and SNMP traps are often reserved for severe circumstances such as an error or possible device configuration change. There is a degree of overlap between the data captured in these technologies’ logs, but not enough to devalue each of their output.

Device Configurations

Most network devices maintain their configurations in flash memory, meaning the configuration persists even if the device is powered off. The configuration itself is simply a text file, full of keywords that make sense to the device's internal firmware. Listing 2.1 provides a sample configuration file.

```
service timestamps debug uptime
service timestamps log uptime
service password-encryption
!
hostname Router1
!
enable secret 0 IMPORTANT!InsertYourPasswordHere!
!
clock timezone est 10
clock summer-time est recurring
!
dial-peer voice 1 pots
  caller-id
  no forward-to-unused-port
  no call-waiting
  ring 0
  no silent-fax
  registered-caller ring 1
  port 1
  volume 3
  destination-pattern 0212345678
!
dial-peer voice 2 pots
  caller-id
  no forward-to-unused-port
  call-waiting
  ring 0
  no silent-fax
  registered-caller ring 1
  port 2
  volume 3
  destination-pattern 0287654321
!
pots country AU
!
ip subnet-zero
no ip source-route
!
ip domain-name insertyourdomainhere
ip name-server 139.134.5.51
ip name-server 139.134.2.190
isdn switch-type basic-net3
!
!
!
interface Ethernet0
  description connected to EthernetLAN
  ip address 192.168.1.1 255.255.255.0
  ip access-group 100 in
  no ip proxy-arp
```

```
ip nat inside
no ip route-cache
no ip mroute-cache
!
interface BRI0
description connected to Internet
bandwidth 128
no ip address
encapsulation ppp
no ip mroute-cache
dialer pool-member 1
isdn switch-type basic-net3
isdn voice-priority 0287654321 out off
isdn voice-priority 0287654321 in off
isdn voice-priority 0212345678 out always
isdn voice-priority 0212345678 in always
isdn incoming-voice modem
compress mppc
no cdp enable
!
interface Dialer1
description connected to Internet
ip address negotiated
ip access-group 100 in
ip nat outside
encapsulation ppp
no ip split-horizon
no ip mroute-cache
load-interval 30
dialer pool 1
dialer idle-timeout 3600
dialer string 0198308888
dialer hold-queue 10
dialer load-threshold 1 outbound
dialer-group 1
compress mppc
no cdp enable
ppp authentication pap callin
ppp chap hostname mybigpondaccount
ppp chap password 0 mybigpondpassword
ppp pap sent-username mybigpondaccount password 0 mybigpondpassword
ppp multilink
!
ip nat inside source list 1 interface Dialer1 overload
ip classless
ip route 0.0.0.0 0.0.0.0 Dialer1
no ip http server
ip pim bidir-enable
!
!
access-list 1 permit 192.168.1.0 0.0.0.255
access-list 1 permit 192.168.0.0 0.0.0.255
access-list 100 permit ip any any
access-list 100 deny    udp any eq netbios-dgm any
access-list 100 deny    udp any eq netbios-ns any
access-list 100 deny    udp any eq netbios-ss any
access-list 100 deny    tcp any eq 137 any
access-list 100 deny    tcp any eq 138 any
```

```
access-list 100 deny tcp any eq 139 any
dialer-list 1 protocol ip permit
!
banner motd ^CUnauthorized users prohibited^C
!
line con 0
  exec-timeout 0 0
  password 0 password
  login
  stopbits 1
line vty 0 4
  access-class 1 in
  password 0 password
  login
!
snmp server 207.46.130.100
no rcapi server
!
!
End
```

Listing 2.1: A sample device configuration file.

As you can see, the file that Listing 2.1 shows is long and complex, which highlights one of the difficulties of compliance management—auditing these configuration files for the proper configuration values is a time-consuming, detail-oriented task that is frankly boring. Even dedicated auditors are likely to miss something. In an environment in which your organization is relying on proper configurations to maintain compliance, manually dealing with configuration files at this level is almost a guarantee that some compliance detail will be overlooked at some point.

For example, suppose you need to ensure that all routers are configured to traffic on TCP port 139. Can you determine whether the configuration that Listing 2.1 shows is compliant with this rule? To make this determination requires training, patience, and searching; imagine how boring it would be to verify this setting on a dozen—or a hundred—identical devices. Many organizations use more than one model or devices from different manufacturers—imagine that you must verify this rule in a dozen *different* configuration files.

TFTP

Devices store their configuration files in an internal flash memory; working with this memory requires that you log on to the router through a Telnet or physical console session. This method is an inefficient way to work with device configurations, especially *en masse*. Fortunately, most devices also support the Trivial File Transfer Protocol (TFTP). By establishing a TFTP server on your network, you provide a place for devices to transmit—by using the TFTP protocol—their configuration files. The TFTP server can also act as a repository for new configuration files—devices can be commanded to load and use a configuration file that is located on the TFTP server.

Because TFTP represents one of the easiest and most common means of getting configuration files on and off of devices, it's a crucial technology in any network or compliance management effort. TFTP can be used to retrieve configurations for an audit, provide modified configurations to meet business rules, and back up and restore device configurations in the event of a disaster.

Foundation Methodologies

Developing a methodology for compliance management can be difficult. Where do you begin? One approach is to use an existing foundation methodology that has been developed from industry best practices. The Information Technology Infrastructure Library (ITIL) is one common foundation methodology that is widely recognized in the IT industry for its adherence to and promotion of best practices.

ITIL Overview

ITIL is a set of general IT best practices created by the United Kingdom Office of Government Commerce (OGC—<http://www.ogc.gov.uk/index.asp?id=2261>). ITIL addresses nearly every aspect of IT operations; of interest to compliance efforts is the ITIL sections on best practices for change and configuration management. The ITIL recommends a fairly comprehensive process of review, testing, deployment, and rollback, which are intended to prevent changes from having an adverse effect on the production environment. Figure 2.3 shows a sample business process developed from ITIL guidelines.

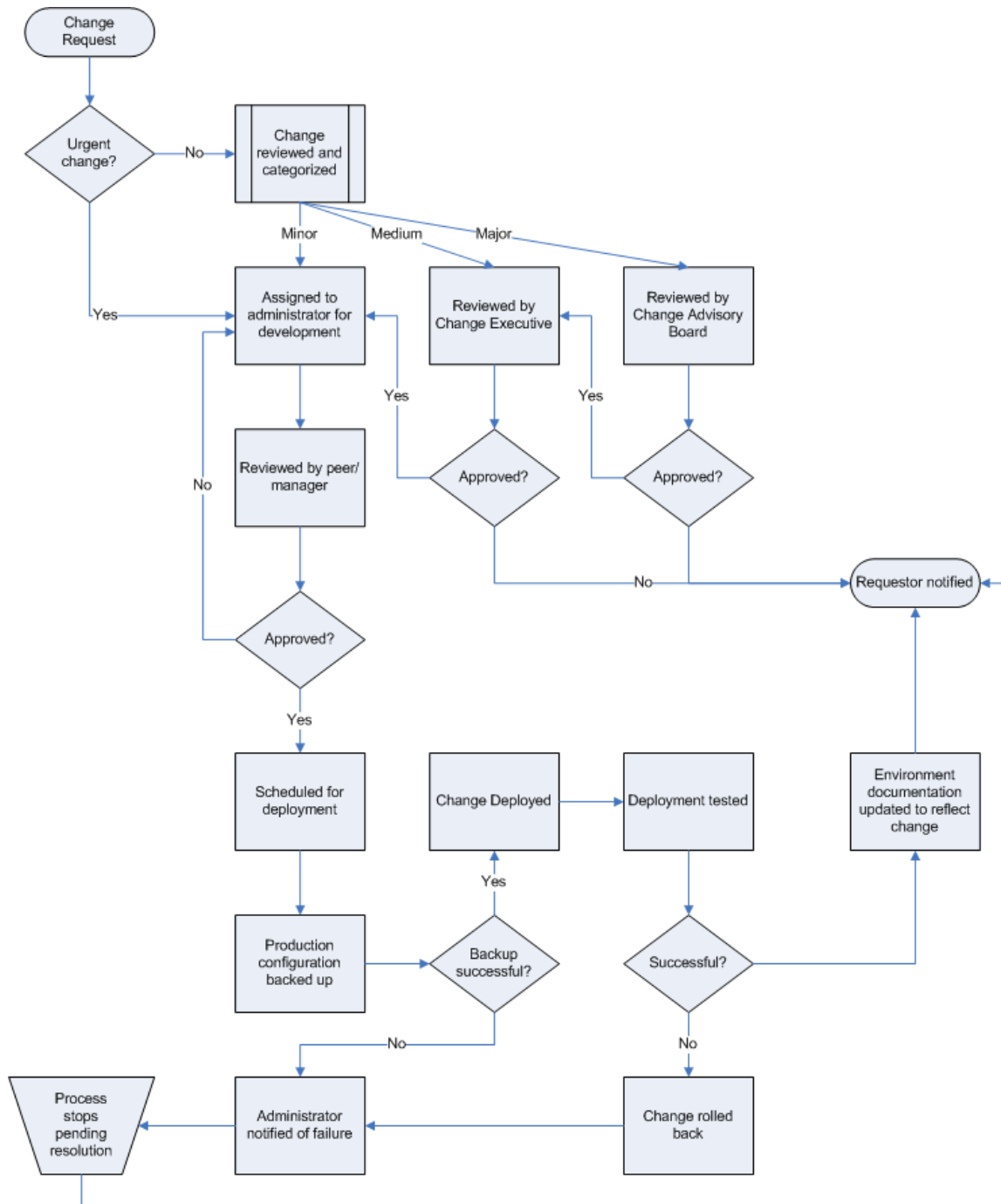


Figure 2.3: Sample change management process based on ITIL recommendations.

ITIL-based processes can form an excellent foundation for processes where compliance is a concern because the ITIL recognizes a few important facts about IT management in general (which happen to be especially true for network management):

- Changes to the production environment can often be easily made.
- Changes made to the production environment often take effect immediately.
- It is difficult to test an existing environment for compliance; it is easier to create configuration standards that are known to be compliant.
- By reviewing proposed configurations prior to their implementation, non-compliant configurations can be addressed prior to implementation, helping to maintain a compliant production environment.

ITIL also recognizes that IT management can become overburdened. ITIL therefore recommends a priority- and security-based categorization system that provides more thorough reviews for major changes as well as an expedited path for changes that are minor and less likely to have an adverse impact on either compliance or operations. In all cases, ITIL recommends peer or management review of changes to help ensure accuracy and mitigate simple human error.

Taking a moment to review the process in Figure 2.3, you can see that it offers several important aspects—both good and bad—to an organization concerned with compliance:

- The process focuses on catching non-compliant configurations *before* they are implemented.
- The process places an emphasis on post-deployment testing, including a rollback phase if the change's deployment doesn't go according to plan.
- The process does not focus on continual auditing of device configurations, a process which has already been identified as time-consuming and error-prone. Instead, devices are expected to be modified only through the process; of course, some means will need to be in place on devices to *enforce* the policy, ensuring that unauthorized (out-of-process) changes are detected and undone, or prevented entirely.


If ITIL has a weak point it's that it is a *process*, not a technology. In other words, the process must assume that you're using the process; no provisions are made for out-of-process events, which are the kind most likely to result in an out-of-compliance situation. Addressing out-of-process changes, however, isn't the job of a best practices system such as ITIL—it is your job to ensure that out-of-process changes simply cannot occur.

Traditional Compliance Management

To ensure that out-of-process changes are eliminated, what are the traditional common practices used to manage today's networks for compliance? Keep in mind that *compliance* simply means “following business rules,” and that those rules will cover a spectrum of concerns—security, reliability, operational, and so forth—and may come from a variety of sources—internal rules, legislative rules, and so on. Let's explore traditional means of compliance management.

Monitoring Only

Many organizations rely on monitoring to ensure that their network devices remain configured in a compliant condition. By *monitoring*, I don't mean periodic checks of the configuration—that would be *auditing*, which I'll discuss next. Monitoring is even more passive, simply waiting for red flags to be raised indicating that something is broken. Essentially, rather than checking the batteries on their smoke detectors, organizations are waiting for a fire to see whether the smoke detectors work. Of course, by then, you've got a *fire*. In other words, in order for monitoring to be effective, something has to be wrong, which means it's already too late.

 Monitoring is still, of course, an effective means of checking performance, activity levels, and other criteria; it's when monitoring is used as a compliance tool that it lacks value.

Point-in-Time Audits

Another common compliance management practice is auditing—periodic spot-checks of device configurations to make sure that everything is set up according to plan. However, as Chapter 1 discussed, this method is ineffective. For example, how many people break traffic laws and never get caught compared with those who are issued a ticket? The discrepancy between these numbers illustrates the inefficacy of auditing: Officers can't be everywhere all the time, so they rely on spot-checks—*auditing*—to enforce the law. Do you want your organization's compliance to be as ineffective as traffic law enforcement?

Auditing tells you that everything is—or is not—compliant *right now*. It says nothing of 10 minutes ago, 2 days from now, or at any other point in time despite the fact that networks are dynamic, constantly changing entities. The network that is audited today *will* be different tomorrow, yet today's audit won't be at all concerned with the network's state of compliance tomorrow.

However, there are situations in which auditing could be useful: If an auditor could be called in each and every time a device's configuration was changed, the auditor would then have the opportunity to audit the environment each time it changes, ensuring that each new iteration of the environment is as compliant as the last one. Obviously, such is not the case, and auditing remains an ineffective way to manage compliance.


Manual Configuration Review

Too often, auditing is based on manual reviews of network configuration files. As I've already described and illustrated, this time-consuming, error-prone process will rarely result in consistent audit results. Thus, in addition to the uselessness of auditing as a compliance tool, the audits aren't even accurate and consistent.

This is not to suggest that a review of configuration files isn't beneficial; it is the only practical way to create a compliant network because end-state testing is so impractical. What I am proposing is that the complexity of these files, and the room (and likelihood) for human error in a configuration file review makes a *manual* review less likely to ensure compliance.

Template-Based Provisioning

Template-based provisioning is a fairly new technique in network management and promises better compliance results. The idea is simple: Create a template of a known-good configuration that is compliant with all of your business rules. All devices are then configured based upon that template. Auditing can begin with a simple automated comparison of a live configuration file with the template that the file is supposed to be based upon; anything in the configuration that matches the template is compliant and can therefore be ignored; auditors can then focus on only the differences. The differences represent a much smaller area on which to focus, lessening the tedium of a manual review and increasing the level of accuracy and consistency.

 Manually performing this comparison is still a point-in-time audit; additional measures, including automated enforcement (which the next chapter will discuss), build on template-based provisioning to provide a more reliable compliance solution.

For example, suppose the text in Listing 2.2 is a part of an approved, known-to-be-compliant device configuration template.

```
ip classless
ip http server
!
dialer-list 1 protocol ip permit
dialer-list 1 protocol ipx permit
!
!
!
tftp-server flash:
snmp-server community corp_orate RO
snmp-server location HCC-Atlanta
snmp-server contact Joe,555-1212,joe@company.com

banner motd #Welcome#
!
line con 0
  exec-timeout 0 0
  password 7 094F471A1A0A
  login
  transport input none

line aux 0
  password 7 070C285F4D06
  login

line vty 0 4
  password 7 01100F175804
  login
!
```

Listing 2.2: A sample of an approved, known-to-be-compliant device configuration template.

Next, suppose that the text in Listing 2.3 is the same portion of an actual configuration file.

```
ip classless
ip http server
!
dialer-list 1 protocol ip permit
dialer-list 1 protocol ipx permit
!
!
!
tftp-server flash:
snmp-server community public RO
snmp-server location HCC-LasVegas
snmp-server contact Joe,555-1212,joe@company.com

banner motd #Welcome#
!
line con 0
  exec-timeout 0 0
  password 7 094F471A1A0A
  login
  transport input none

line aux 0
  password 7 070C285F4D06
  login

line vty 0 4
  password 7 01100F175804
  login
!
```

Listing 2.3: A sample portion of an actual configuration file that is supposed to be based on the template that Listing 2.2 shows.

A simple comparison utility would reveal the differences between the template and the live configuration:

Template:

```
snmp-server community corp_orate RO
snmp-server location HCC-Atlanta
```

Configuration:

```
snmp-server community public RO
snmp-server location HCC-LasVegas
```

One of these differences—the change of the location to LasVegas—might be documented as an acceptable deviation from the template. The other, however, is a security-sensitive setting that configures the SNMP community string. The ability to focus on only these two lines will enable an auditor to more easily catch a non-compliant configuration setting that might otherwise be lost in a file that contains hundreds of lines of configuration settings.

The Shortcomings of Traditional Compliance Management

Simply by reading the previous few sections, you've likely spotted a few problems with these traditional compliance techniques. However, there's more wrong than meets the eye.

Vendor-Specific

One major problem with many compliance efforts is that they're vendor-specific. Training auditors to look at configuration files, for example, is a vendor-specific task, meaning companies with more than one vendor's equipment will need to train auditors on each. Even management solutions provided by vendors are specific to that vendor's equipment.

To resolve this problem, companies need to employ a management solution that is vendor-neutral and supports a broad range of manufacturers' equipment. The ideal solution will create an abstract version of device configurations so that every configuration setting looks the same, regardless of which manufacturer's device it came from. This abstraction—or translation, if you will—helps to homogenize the configurations and make them more easily audited or modified.

Lack of Reporting

Because traditional compliance management is largely manual, no automated reporting is available. However, even most automated configuration management tools lack reporting capabilities. For example, a solution that backs up device configurations should be able to produce reports that list devices whose configurations have changed since the last backup; such solutions rarely provide this level of reporting, however. Foundation technologies that support compliance effort often lack reporting, too. Having a RADIUS/TACACS solution produce a report that lists all administrator access to a device might be a useful tool for judging whether the device needs to be re-audited; too few of these solutions lack such reporting capabilities, making them less supportive of techniques that would result in better compliance.

Alerting, Not Enforcement

Alerting is a common way for organizations to keep tabs on their environments, but alerting is too slow. By the time an alert has been produced, a problem already exists and must be corrected immediately. Enforcement combines alerting with an automated, immediate response—perhaps rolling back a device configuration to a known-compliant version—which *ensures* compliance rather than simply alerting to you to *lack* of compliance.

Lack of Logging and Auditing

Centralized network configuration tools often lack sufficient logging and auditing capabilities. Although these tools provide centralized control over devices, which is a crucial component for easier compliance management, the tools are not often designed to keep track of what centralized changes are made and by whom. This shortcoming essentially removes the tool's usefulness as a compliance management tool (although not as a network device management tool, which is in fact what most such tools are built as).

Entirely Manual

Most compliance efforts, in the end, are entirely manual. Manual processes are all subject to human error and inconsistency and are therefore less preferable than automated processes. Ideally, every manual process in your current compliance management efforts is replaced, at some level, by an equivalent automated process:

- The process of comparing device configurations to known-compliant templates should be automated.
- The process of rolling back configurations to a known-compliant state should be automated.
- The process of auditing devices each and every time their configurations are changed should be automated.

With automated processes, manpower, time, and money become non-issues, and compliance can be more consistently ensured across the network.

Not Real-Time

Traditional compliance management—which relies heavily on auditing—isn't real-time in nature. Instead, it tends to focus on point-in-time audits, which don't reflect all the changes that can occur from moment-to-moment on a production network. Capturing changes in real-time is critical because being out of compliance for even a moment can result in enormous losses; realizing that you have changes occurring can even help you redesign your network and security to help prevent those changes from occurring.

Lack of Accountability

Many traditional compliance management methods seek to establish accountability, in large part as a result of the many pieces of legislation that now mandate accountability (HIPAA, Sarbanes-Oxley, and so forth). Traditional compliance management, however, often fails to achieve this accountability at the network device level. Network devices are notoriously difficult when it comes to accountability because few of them lack any built-in means for tracking who makes what changes. Most devices support external technologies, such as TACACS or RADIUS, which can provide accountability, but these technologies often make it difficult to tie the *who* with the *what*: Although RADIUS can, for example, let you know that an administrator logged onto a router's console, RADIUS can't generally detail the exact changes that administrator made because RADIUS (and the network device) simply wasn't designed with that level of granularity. As a result, accountability for network management is often slipshod or inaccurate.

Not Continuous

Traditional compliance management is manual, lacks supportive reporting capabilities, and tends to be vendor-specific, so it cannot be continuous. Any compliance effort that isn't continuous—which, in other words, relies on spot-checks—is next to useless because networks simply change and evolve too quickly for spot-checks to have a hope of catching non-compliant conditions.

Your network needs to be non-compliant for only a few minutes in order for security breaches, operational problems, reliability issues, stability concerns, and other problems to occur. Fixing the problem quickly doesn't negate the fact that a problem occurred and damage was done; only *continuous* compliance management can truly be effective. The timeline in Figure 2.4 shows how non-continuous compliance management leaves plenty of room for problems to occur.

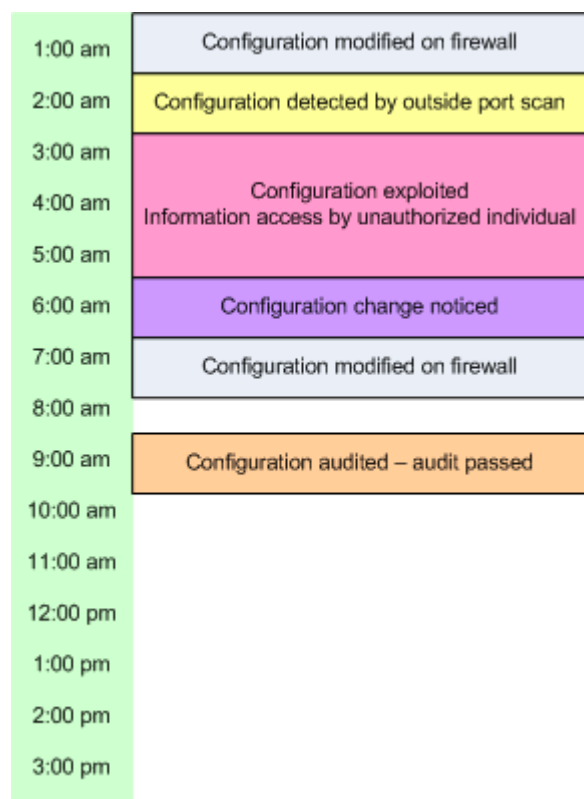


Figure 2.4: Timeline showing how an audit can miss a non-compliant configuration as well as the resulting damage.

Summary

Thus, if every traditional compliance management technique—monitoring, auditing, alerts, manual configuration reviews, and so forth—doesn't provide adequate compliance assurance, what does? Chapter 3 explores the answer to that question: Leading-edge techniques for providing real-time, information-rich, and highly automated compliance management. You'll rely on the same foundation technologies—SNMP, Syslog, TACACS/RADIUS, configuration files, and so forth—that this chapter has introduced, but you'll replace spot-checks, error-prone manual reviews, and other traditional techniques with new solutions that make compliance more automated and more consistent.