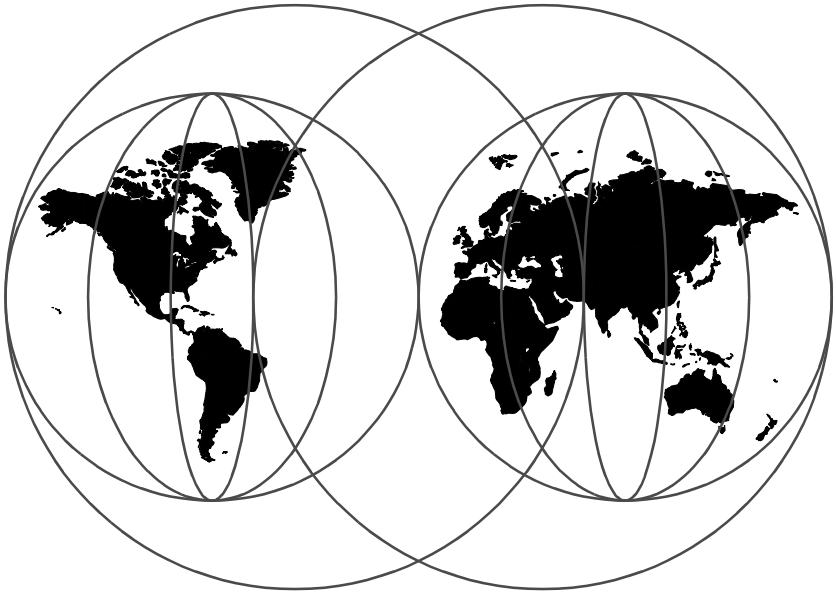
Tivoli Security Management Design Guide

Richard Hawes, Guenther Mayerhoffer, Thomas Schuster



International Technical Support Organization

http://www.redbooks.ibm.com

SG24-5101-00

SG24-5101-00

International Technical Support Organization

Tivoli Security Management Design Guide

July 1998

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix B, “Special Notices” on page 121.

First Edition (July 1998)

This edition applies to the Tivoli program product, "Tivoli Security Management". The contents are not intended to be version-specific (unless otherwise noted). It has been written with versions up to and including 3.6 in mind.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. DHHB Building 045 Internal Zip 2834
11400 Burnet Road
Austin, Texas 78758-3493

See also the preface section “Comments Welcome” on page xii about other ways to submit comments.

Contents

Figures

Tables

Preface

The first Security Management redbook, *Managing Access from Desktop to Datacenter: Introducing TME 10 Security Management*, SG24-2021, described the product and was aimed at helping the reader become familiar with the product features. However, an effective implementation of Tivoli Security Management involves defining appropriate access control to significant resources. Even if you are familiar with the product and its capabilities, determining what to protect and how to define appropriate access can be a difficult task. The defining document for a Tivoli Security Management installation should be a company's security policy. But in situations where the policy is incomplete or does not exist, how do we go about putting one together?

This publication provides a methodology for designing Tivoli Security Management installations. Starting from the high-level, organizational viewpoint, we show how to define what needs protecting, and how to implement the right levels of protection management.

As there is no book that describes the design of a security management implementation up to now, our aim was to publish this document quickly to help people in the field.

We welcome your comments (see "Comments Welcome" on page xii) as we intend to revise this publication in the future. This edition concentrates on Windows NT and UNIX platforms. Watch for further Redbooks on other platforms, including OS/390, in the upcoming book tentatively titled *Managing the OS/390 Security Server with Tivoli*.

The Team That Wrote This Redbook

This redbook was produced by a team of specialists working at the International Technical Support Organization (ITSO) Austin Center.

Richard Hawes is a Senior Systems Engineer at the ITSO, Austin Center. He writes extensively about the Tivoli Management Environment especially the Security discipline. Before joining the ITSO in July of 1997, Richard worked in the European Software Project Office (based in the UK), where he performed technical troubleshooting and critical situation-management on IBM products for OS/2 Warp Server and Windows NT.

Guenther Mayerhoffer is an Advisory Systems Engineer in IBM Germany's Professional Services Organization. He has several years of experience in

system management disciplines and client server technology in general. Guenther wrote sections of the redbook *TME 10 Cookbook for AIX*.

Thomas Schuster is an Advisory Systems Engineer in IBM Germany's Professional Services Organization. He has several years of experience in C/C++ and Visual Basic software development including user interface design and one year of experience in systems management. He has worked at IBM for one year. His areas of expertise include Tivoli Systems Management, Windows NT and UNIX operating systems.

The production of this material requires assistance from highly skilled people who already have many demands on their time. We would like to thank the following people for their invaluable contributions to this project:

Daniel Craun
Greg Fisher
Tivoli Professional Services

Gregg Wilson
Tim Moore
Tivoli Development

Tivoli Management Product Names

In an effort to eliminate any confusion about the names for Tivoli's expanding line of management products, Tivoli has recently been through a brand naming review. Those already familiar with the products mentioned in this publication will be used to seeing the names as TME 10 Security Management, TME 10 User Administration, and so on.

The new naming convention for these enterprise software management products will replace TME 10 with Tivoli. The new names are Tivoli Security Management and Tivoli User Administration. (This change may seem trivial, but the consistency comes from more dramatic changes on other products, such as Unison Destiny, which is now Tivoli Output Manager.)

Throughout this publication, we have endeavored to use the new names wherever practical. This includes references to the Tivoli Management Agent, often referred to in the past as the Lightweight Client Framework (LCF).

Comments Welcome

Congratulation or criticism, your comments are important to us!

We want our redbooks to be as helpful as possible. If you have implementation experiences that would benefit others or have any comments about this edition, please send them to us in one of the following ways:

- Fax the evaluation form found in “ITSO Redbook Evaluation” on page 137 to the fax number shown on the form.
- Use the electronic evaluation form found on the Redbooks Web sites:

  For Internet users `http://www.redbooks.ibm.com`
  For IBM/Tivoli Intranet users `http://w3.itso.ibm.com`

- Send us a note at the following e-mail address:

  `redbook@us.ibm.com`

Chapter 1. General Introduction

This chapter is an introduction to this book. It provides a short overview of what we mean by Security Management and how Tivoli Security Management fits into the picture. We then introduce an outline of the methodology proposed in this material.

1.1 Security Management Overview

Today, information technology has become an integral part of an organization's infrastructure. Almost any company cannot be competitive without it. It is also critical for an organization to keep their systems secure.

But what do we mean by secure? Is it enough to provide the user with an account and a password? The point of this document is that it certainly is not.

Computer security means that you can depend on the confidentiality, integrity, and availability of your machine and its software and data. You expect your sensitive data to be protected from unauthorized disclosure (confidentiality). You want to retrieve your data in an accurate and complete form (integrity) when you need it (availability).

As computer environments have become more distributed, it is no longer sufficient to rely on host-based security techniques to protect network data. The idea of traditional host-based security must be extended to local area networks (LANs) and wide area networks (WANs). Now, there is not only one critical point to be secured, but many systems and their connections. A network is only as secure as its weakest link. Therefore, all network components must be secured at an adequate minimum level.

What makes this task even more difficult is that we have to deal with a heterogeneous world. As every application has its preferred platform, there is usually more than one security control mechanism to manage.

Password policy, data and application protection, and remote login security are just a few of the aspects to consider in a distributed environment.

Terminology

Throughout this publication, we will use terms such as access rights, privileges, and permissions. Unless clear from the context, we are using these terms in a generic sense. These terms have specific and often different meanings on different platforms.

1.2 The Tivoli Security Management Approach

Up until now, every platform in an organization is administered by its native security mechanism (if it has one). Usually, this work is done on different machines, through different interfaces, and probably, in different locations. Tivoli Security Management centralizes this administration in a single interface running on one machine while retaining the ability to distribute management capability, if desired.

Before Tivoli Security Management, changing access to different resources meant that it was necessary to log on to the different systems and change the access rights. When working with a large number of machines, this often leads to human error, which can introduce security holes in the company. Centralized security management with the Tivoli Security Management product allows administrators to manage the access rights of all resources using one system that provides an overall view of the organization's security systems. They will not make mistakes through having to perform tasks repeatedly for numerous users over disparate platforms. Even if the structure of the enterprise calls for distributed management responsibility, the Tivoli product is flexible enough to cope with that requirement.

The Tivoli product answers the need to integrate with business processes. It allows the security administrator to implement a consistent security policy across the enterprise using a role-based security model. A role-based model maps job functions to roles. A group of users can then be defined according to sets of roles or job functions those users need to perform, not limited to a specific platform group architecture. So, as new employees are hired, all the administrator has to do is create user accounts and assign them to groups, which reflect their job in the company. Through integration with Tivoli User Administration, this creation of new users can also be handled in a centralized, uniform way.

The ability to implement uniform policies (for example, a uniform password policy) for all platforms is also provided by Tivoli Security Management.

1.3 Security Design and Implementation Methodology

As we have seen, it as a complex task to assure a secure computer environment in an enterprise. So, especially for large installations, there is a need for a methodology to ensure correct security through a comprehensive design and complete implementation.

You can start by securing critical resources by implementing them in the Tivoli Security Management software. You can arrange your access rights and set full auditing. In many environments, this covers what is already implemented in existing security products, and you may think that this is sufficient to be sure that your systems are secure. This may be true for smaller networks. But for bigger distributed heterogeneous environments, this can still lead to security holes and a continual requirement for reconfiguration. There will be no overall manageable view of the security configuration.

What you really need to have is an ordered step by step development of the security implementation. (And you should not only have a plan, you have to live it!)

In this publication, we define a plan of four major steps that should be used to establish security management for a company. An overview of these steps is shown in Figure 1.

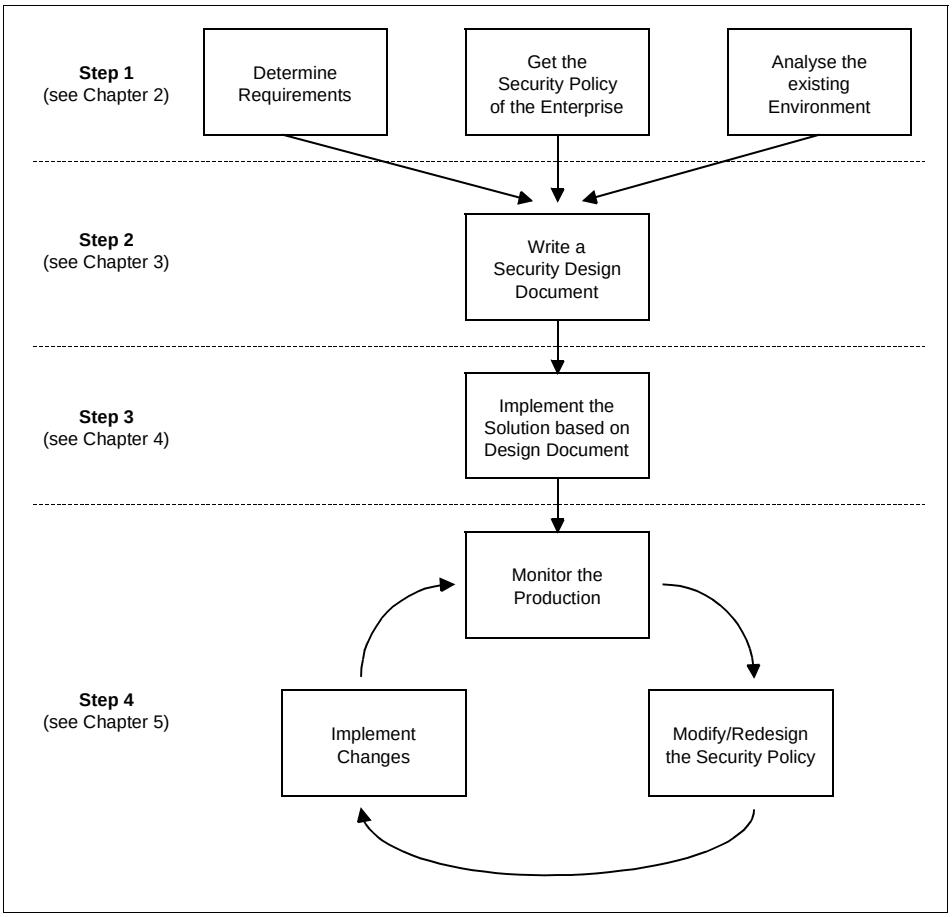


Figure 1. Security Design and Implementation Process

These steps can be summarized as follows:

1. **Analysis of the Target Environment**

   First of all, you will have to determine any specific requirements. By this we mean, what it is that those in the organization who are responsible for security management expect to accomplish. You need to find out what their aims are. For example, they could be satisfied with the existing security of the systems and just want to save money by centralizing security management. On the other hand, they may expect the implementation to prevent the recurrence of a security problem they have seen in the past.

Then, you will need to investigate the existing security environment of the customer. Ideally, this would mean simply getting the security policy document. Unfortunately, there are many companies that do not have an official document that says how they protect data and control network access. Usually, the security rules have grown with the company, but nobody standardized them and wrote an official paper. If a security policy document does not exist, then ideally, one should be written before implementing a product, such as Tivoli Security Management. In most environments, this is a non-trivial task and may be something that requires the services of a company specializing in writing such documents.

After that, you have to analyze the actual computer environment to get an image of what really goes on with the machines. Apart from cross-checking the real environment with the security policy document, this will determine what platforms are in use, what kinds and quantities of data there are, and how the existing security measures perform. During this step, you should also identify any points of integration required between Tivoli Security Management and other products, such as login applications.

2. **Writing a Security Architecture and Design Document**

   From the output of the previous step, we can now write a security architecture and design document. That means to translate the customer's expectations using the knowledge of their environment to a complete security management design oriented on the Tivoli Security Management Product. This also provides a good basis to review ideas and solutions and discuss them with the sponsor.

   Adequate completion of this step is essential. Proceeding to an implementation is likely to present problems if you have not finished the architecture and design document.

   The security architecture and design document also will contain the first two phases of the Security Policy Translation Process. This process is the method required to translate the corporate security policy gained from the Analysis of the Target Environment step into actual Tivoli Security Management configuration options. The three security policy translation process phases are:

   1. Identify all the appropriate action steps in the corporate security policy.
   2. Translate the action steps into a list of security processes.
   3. Translate the security processes into actual Tivoli Security Management configuration steps.

3. **Implementation and Installation**

On the basis of the written and reviewed architecture and design document, we now are able to implement the developed solution based on the security architecture and design document. That does not only mean to secure the resources, but also to test and validate that the systems are really secure. You will also have to document and transfer your knowledge of the implementation to the security administrators so that they can work with the security system. This step may be where a statement of work is published to define the scope of the project.

The third, and last, phase of the security policy translation process takes place in this step. The total number of Tivoli Security Management configuration steps involved will typically be listed in the statement of work. The actual list of configuration steps required should be compiled in something such as a post-deployment summary document delivered at the conclusion of the project.

4. **Monitoring and Maintenance**

   This is a step that is often ignored. Once you set up something, you do not watch it anymore, because you think this is fine. That may work with other subjects, but not with security. It is absolutely necessary to monitor the production environment of your security implementation. Discovering security holes or ways to make improvements will require you to modify the security policy and implement the changes. This is the only way the systems can be kept at the required level of protection.

The first two of these phases are the main emphasis of this redbook. Issues related to the physical installation are covered in the product manuals (especially the release notes) and in the *Managing Access from Desktop to Datacenter: Introducing TME 10 Security Management*, SG24-2021, redbook. Monitoring strategy is covered to some extent in this Redbook in the discussions on auditing. Future editions of this Redbook will also tackle the latter phases in more detail.

Except for the simplest of environments, you are unlikely to be successful establishing security management starting with a step other than a thorough analysis of the current environment. The steps in our procedure depend on the previous step being completed. Therefore, it makes no sense to skip steps.

This procedure will help you to set up a realistic and structured project plan. Keeping Figure 1 in mind will make it an easier task to perform.

This book describes the first two phases in order and follows with information of relevance to the last two phases. So, following the book from the beginning

to the end will guide you through a good evaluation and design for Tivoli Security Management.

Chapter 2. Analysis of the Target Environment

This chapter describes the first step in developing a security management implementation. It is divided into the three parts we first highlighted as phase one of our implementation method - see Figure 1 on page 4. First, we will determine what the customer expects from centralized security management, what the main aims are. We then talk about the security policy of an enterprise and its significance to the project. This will give us an overview of what is done with the machines, what kind of machines and platforms are in use, and if and how these machines are protected right now. Finally, we will analyze the existing computer environment to ensure the reality matches the documentation.

As you will see, this chapter gives you guidance in taking a snapshot of the environment to be secured. It should lead to the following outputs:

- A document showing the objectives concerning security management
- A security policy document
- A document describing the computer environment and the existing security implementation

The generated output will be used as input for the next step (Chapter 3, “Tivoli Environment Security Architecture” on page 47). The more work put into these documents, the easier the design pieces will be. That is why it is recommended to go through this step carefully and as completely as possible.

2.1 Determining Requirements

The first step to take is to determine the security requirements a company has. Different kinds of organizations require different security designs and implementations. The U.S. Department of Defense will need a security strategy that is very unlike that of a McDonald’s Restaurant, but both will need to implement some form of security policy.

Therefore, you will first have to understand the business of the organization. You will have to communicate a lot. This will include interviews with the departments to find out what their work is about. You will soon get an image of what is relevant for the company’s business, and what is not.

It may be appropriate to talk to the CIO and other senior managers to ask what is expected of a Tivoli Security Management implementation. It is a good idea is to ask for the priorities in protecting resources. At least, it should

be easy to identify the number one resource to be protected. And again, you find out what is relevant, and what is not. This will simplify taking the next steps.

It can be very interesting to find out more about what prompted a company to do a security management implementation. For example, you may be able to discover examples of security breaches in the past and their consequences.

This is also the right time to speak about the capabilities of the Tivoli Security Management product. You should be able to identify and explain what can be covered by the product, and what cannot. This helps you to avoid expectations that can not be fulfilled in time or in budget. It is important to make it clear that a satisfactory implementation of Tivoli Security Management depends on many other factors, such as the physical security of the computer systems and the network. The best design and implementation is worth nothing if there are security holes introduced in other places. The organization's entire network may only be as secure as its weakest link.

You will need to inform the management sponsor that implementing effective security management brings organizational changes. Processes are modified, and new job functions will be added and altered (for example, the job function *security administrator* must be defined and added).

Get all the information you think might be important for the security management implementation, and as with any major project, keep communicating throughout the process to avoid misunderstandings. Good practice includes formal review meetings throughout the timeframe of the design and implementation.

2.2 The Security Policy

The second part of analyzing an existing environment is to get the organization's security policy document. This document should describe what types of security measures are used, and how these should be implemented. It will inform you about the company's security standards and guidelines.

This document is the basis of any further work. The security of an organization's computer infrastructure depends on the quality of this document. Unfortunately, there are many companies that do not have an official document that says how they protect their distributed computer environment. This may be because there never was a serious incident that encouraged the management to work out a security plan for the whole organization. Usually, the security rules have grown with the company, but nobody standardized them and wrote an official paper.Often, the result of this

piecemeal development is that there are undocumented standards for computer security (if there are any at all). Even if there are documented topics, they are scattered across different departments, and these documents often contradict each other. For example, they may define the minimum level of security for a given resource differently.

Tivoli Security Management can implement a design allowing differences in implementation at local levels or a hierarchical administration structure. However, in order to centralize security management, different groups cannot dictate their own policy separate from everyone else. There has to be a *complete* and *consistent* policy throughout the company that encompasses the requirements of individual departments. Even if the organization has such a document, we must verify that it is complete and consistent. Experience shows that most organizations do not have a full-scale security plan. We will, therefore, give guidance in how an organization should develop a security policy. Even if you are not involved in the development of a policy, this information will help you to verify and adapt an existing document and translate it into activities required for implementation.

2.2.1 Security Policy Basics

Developing a security policy is a complex task. The effective security of a company's computer environment depends on the quality of the policy. This section gives you guidance in one method of developing a security policy. It will give you hints and tips and tell you what to consider when performing this task. This should not be considered a comprehensive treatment of this subject. There are many publications that deal with this subject in more detail (some of which are listed in Appendix C.3, "Other Publications" on page 125). Several points that the security policy document should contain are not part of this book, because they are not necessarily related to the Tivoli Security Management product. One example would be the control of physical access to computer systems, whether they should be locked in a room by themselves, should diskette drives be disabled, and so on.

As shown in Figure 2, a security policy can be divided into two parts.

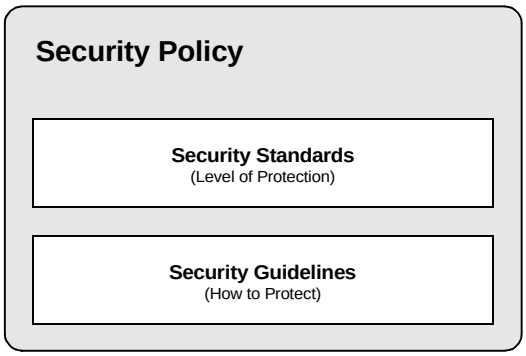


Figure 2. Contents of a Security Policy

The first part contains the *security standards*. These standards set the computer security measures for the organization and describe what to secure and at which level. Usually, they are phrased in terms of *shall*, as in the example below. Standards are generally platform and product independent. For this reason, this part of the security policy changes slowly over time. In order to see if a particular standard has been met, there has to be a metric applied. Standards cover such issues as which resource accesses to audit, what data to protect, and which applications have to be secured.

Here is an example for a standard to monitor login attempts:

All login attempts of all network-connected machines shall be monitored. That includes successful and unsuccessful logins. For every event, there shall be one record added to a storage system. The storage system shall reside locally on each machine. This data can be moved to a different machine or media if disk space is a concern. Unsuccessful logins must be reported to a central administration point. The login history has to be kept for 6 months.

As mentioned before, this standard is platform independent. It does not name the platform-specific mechanisms the login attempts are recorded with. But it does clearly say what has to be monitored, where the data shall reside, and for how long (metric) the login information has to be stored.

Here is another example of a standard. This one is for authentication:

Every user account on each multiuser machine shall have only one person authorized to use it. That user will be required to authenticate his or her identity to the system using some positive proof of identity. This proof of identity can be through the use of an approved authentication token or smart card, an approved one-time password mechanism, or an approved biometric

unit. Reusable passwords will not be used for primary authentication on any machine that is ever connected to a network or modem, that is portable and carried off company property, or that is used outside of a private office.

(Quoted from *Practical UNIX & Internet Security*, O'Reilly Press)

The second part of a security policy is made up of the *security guidelines*. The security guidelines define *how* to assure a secure computer environment. This implies that this part of the security policy is not platform independent. It is a detailed platform-specific interpretation of a certain security standard, ideally, using the *should* term. As platforms, software, and architecture change, this part of the security policy changes. It should be written by the persons with responsibility for all security issues of the organization. These persons not only need to have a strong skill in computer security, but also, in the platforms and applications in use.

Here is an example guideline for auditing logon attempts on the Windows NT platform:

Every administrative action on the Windows NT platform should be performed using Tivoli Security Management and/or Tivoli User Administration software, if there is not a functional requirement to do it by other means.

The event auditing of Windows NT must be enabled (it is disabled by default). Logon and logoff auditing should be turned on for successes and failures for all Tivoli groups.

By default, the security log of Windows NT is 512 kb in size, and events older than seven days are overwritten. Using the Log menu in the Event Viewer, logging parameters should be defined such that events are overwritten that are older than 180 days. The log data should be archived so that the logon attempts of the last 180 days can be reviewed. If disk space is a concern, you may move the log file data to a different partition or machine.

Unsuccessful logons should be reported to the appropriate Tivoli Enterprise Console (TEC) using the TEC Windows NT Event Log Adapter.

The configuration that should be applied is shown in appendix xy of this security policy document.

This guideline is concrete and specific for the Windows NT platform. It gives a detailed description of how to implement a standard. The system and/or security administrator now knows how to implement this particular standard using the guideline. The implementation is no longer a matter of individual judgement.

We have seen that the security standards and guidelines complement each other in one document. In practice, it may be preferable to have a separate document for the security guidelines, because they change as the computer environment changes.

Designing a security policy is not only a technical task. There are other considerations too. For example, suppose you chose to turn on auditing for all resource accesses. You will soon discover that this results in greater costs for hardware, software, and service. This is because the huge numbers of events generated will need bigger or more TECs, faster networks, and more data to be backed up. This example may be a little frivolous, but it shows the dependencies between security implementation and business practicality.

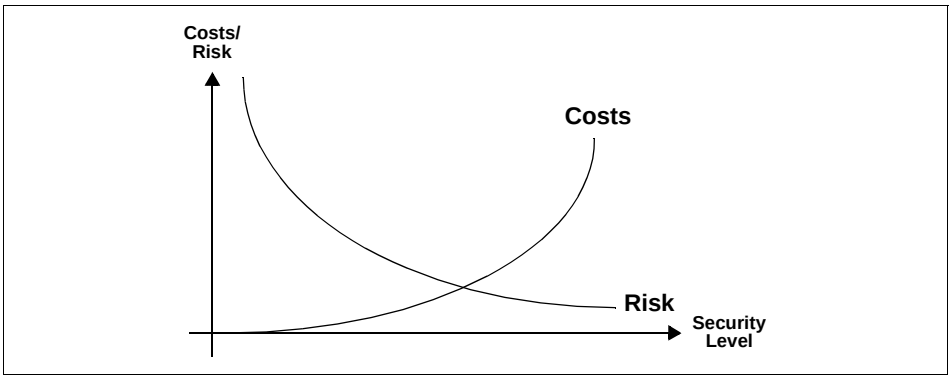


Figure 3. Relationship Between Security Level and Costs

Figure 3 illustrates the trade-off between the decreasing risk associated with higher levels of security versus the costs involved in achieving that security level. With a higher level of security, the costs for security management increases, and the risk of having security breaches decreases. Therefore, what is required is a compromise between the costs and the risks in a business risk assessment. Every security standard and guideline will need to be assessed concerning its use and costs for the company to come to a realistic and applicable security policy.

Note

Designing and implementing a new or modified security policy changes business processes and interfaces. Therefore, it is important to communicate with many departments in the company to assure a healthy security management implementation. The best security management is worth little if it does not fit in with the company's standard practices.

2.2.2 The Structure of a Security Policy Document

If you have the task of writing a security policy document, you may well wonder where to start. Once you know where to begin, you will still need to know how to go on and what aspects to include. That is exactly what is described in this section. There are many opinions on what makes a good security policy document. Here, we present suggestions for details that are used consistently in the best security policies.

Like any good technical management document, the security policy document has a general part, a kind of introduction. This part should give the reader an overview of the contents and the objectives of the document. Note that the target group is not only security staff. The standards (non technical) part should be understood by all other staff too, as it contains general security rules and guidelines for employees. Many organizations consider it a part of the employment contract that the employee has read and understood the security rules and guidelines.

2.2.2.1 Security Policy Objectives

First of all, you need to state the objectives of the document. The reader should have this information, even if they can find out about it after having read the document. The objective can be very short, but it must clearly describe the function of the document.

An objective could look like this:

- Objective

  Establish security implementation practices for the protection of the information assets on network and computing environments of the company XYZ Corporation.

2.2.2.2 Security Policy Scope

Next, you need to define the scope. Here, you set the boundaries of the document, so it is clear where it is applicable. This is a very important point and must be defined as precisely as possible in order to avoid misinterpretations of whether included security standards are applicable to a certain system or not. Usually, the scope is phrased very generally to be sure all the types of systems that are involved in the company's computer environment are included.

Here is an example:

- Scope

  This document defines the requirements for network infrastructures and system services, where XYZ Corporation internal business processing is

performed and where external availability of the infrastructure and system service is required.

You may define exceptions to this scope. There are usually systems that do not need to meet the defined standards and guidelines. For example, it may make sense to exclude systems used for education, demonstration, or test purposes, especially if those systems are not connected in any way to the corporate network.

2.2.2.3 Security Policy Basic Philosophy

After that, you can pick up a basic philosophy. You can save a lot of writing if you define a basic rule, such as Everything that is not specifically prohibited is allowed or Everything that is not specifically allowed is prohibited. In this case, you will have to take care of the consistency of the document. You may also need to redesign the whole document if you change the philosophy. You will have to decide whether to live a philosophy, or not, case by case.

Here is an example:

- Basic Philosophy

  XYZ Corporation computer systems should be extraordinarily restrictive and only allow the minimum access wherever possible.

It may be that documenting the philosophy is not that straightforward. Instead there may be some limited attempt to describe it, but it will influence decisions on all other aspects of the security policy.

2.2.2.4 Defining Asset Owners

Security management has a better chance of working if every piece of information and equipment is assigned an owner. Without having an owner for a resource, there is nobody to control it and assign access rights that are appropriate. In most organizations, you can find a resource whose ownership is not quite clear. This is why it is so important that this rule is defined in the general part of the security policy.

A ownership rule could look like this:

- Ownership

  All components and resources named in this document must have clearly identified owners. The owner must ensure that their infrastructure components comply with the standards and guidelines of this document.

Note

For the security policy document, it is enough to assign owners to the infrastructure components. But the person who is assigned the responsibility must also have the authority to set rules and deal with violations.

2.2.2.5 Good Practice Tip - Document Control

At the end of the general section or in the part following, you should include a document control piece and a change history. This is very important for the security policy document, because it is a document that constantly changes. There will be frequent modifications once the document is set up. Usually, there are periodic revisions to company security. One suggestion is to provide version numbers for the document to identify obsolete copies. On-line versions of the document that are likely to be printed out usually include a statement to the effect that the document is only current on the day it was printed, and a pointer to where the latest version of the document can be obtained.

Here is how a document control and change history may look:

- Document Control

  This document is reviewed annually and is re-issued when revisions are necessary and approved. Obsolete copies of this document should be destroyed by the document holder as soon as it is practical.

  All proposed revisions to this document must be reviewed and approved by the XYZ Corporation Security department.

- Change History

  Initial Release - Version 1.0 - July 1, 1998

  Version 2.0 - July 2, 1998

  - Changed Title of section “Network Environment” to “Distributed Network Environment”
  - The “Audit” Section has had added the sentence. “Unsuccessful logons must be reported to a central administration point”

You may add some more points to the general introduction, such as related documents or, as suggested before, where the latest version is available.

This concludes what should be included in the general part of the security policy document.

2.2.3 Security Policy Standards and Guidelines

You will now begin to write the security standards and guidelines. In 2.2.1, “Security Policy Basics” on page 11, we saw how they are defined and phrased. Standards describe what to secure, at what level, and use terms, such as *shall*. Guidelines are more specific to the platforms in use and define how secure the resource in question is. Guidelines are usually phrased using the word *should*.

We will now give you a list of content items. This list does not presume to be complete. We have concentrated on topics related to the Tivoli products, but it will give you an idea what to include when writing a security policy document:

- Identify and Authenticate Users

  This section must define the standards and guidelines to ensure that a unique identifier can be associated with a user on a system. There should be a section for user ID and password policies. This also covers invalid password attempts and password resetting.

  You can put the business use notice in this section, too, which might say that it is not permitted to use the systems for other than business purposes.

- Define and Protect Resources

  This section covers the treatment of all the resources in a company. It could be divided like this:

  - Classification of Data Objects
  - Protection of Confidential Information
  - User Resources
  - Operating System Resources
  - Virus Protection

- Auditing

  In this section, you define the audit strategy. It specifies whether you want to perform statistical analysis of the data, or whether you are just interested in reporting access failures. Name the events to audit and specify how long to keep the log data.

- System and Security Administration

  Here the roles of the system and security administrator are defined with their privileges and access rights. Note that these are different roles - whether they are assigned to separate people will depend on your security management philosophy.

- Security Status Check

  All systems have to be periodically checked for the health of their security. Here, you define that this should be done and how frequently. You should name the actions that constitute a check.

In small environments, single standards and guidelines can be very short.

There is one thing left to say. Try to write the security policy document, especially the guidelines, in a personal way. The document is an official document, but it is still dedicated to people. Employees will be more willing to meet the standards if they accept the style of the security policy document.

In the next section, we look at what is actually in place in the target environment to determine how well it matches the security policy document (if one existed) or what policy is in place.

2.3 Analyzing the Existing Environment

In this section, we are effectively defining a security policy. Even if a security policy exists and much of this information may be in there, this should still be verified against what is actually in place.

The granularity of analyzing the company's environment depends on the aims of the company concerning security management that have been discovered in "Determining Requirements" on page 9.

First, we start with the existing naming convention, so that we can be sure that resources are not defined twice, or more, because the same resource has different resource names. After this, we take a look at relevant corporate standards and all hardware and software resources, such as the Tivoli environment and the operating systems, in use.

In this and subsequent sections, we also suggest several forms that can be used to collect data from the environment. The blank forms are collected together in Appendix A, "Planning Forms for Security Management Design" on page 117. Depending on the security mechanisms that are already established in the company's environment, these forms should be expanded with additional attributes for resources, such as default access, further access time restrictions, and audit control.

After we have finished the analysis of the existing environment, we can use the results as input for writing the security design document. The design document provides the necessary detail to enable someone to implement Tivoli Security Management in this environment. Apart from ensuring that we

incorporate the existing security measures in our design, we will use this data to avoid breaking what is already in place. As we will see as we progress to the implementation phase, we have to be very careful about blanket access restrictions, as many processes may require accesses that may not be obvious.

2.3.1 Incorporate Standard Installation Requirements

Before we are able to design the structure of Tivoli policy regions, profile managers, and profiles, we have to view the configuration of client and server machines. We want to know if the operating system and the applications on these machines are installed in some predetermined, default way - such as a standard way of defining install directories. If there is no standard, the number of profiles we will need in order to manage these systems increases dramatically, because every site with different installation directories for applications or with different versions of operating systems would force the need for a different profile.

Every management system works more effectively and is easier to maintain if the company uses standard installations. This is similar to the requirement to standardize other areas, such as object and system names. You should ask the following questions:

- Can we classify the computer systems by different types of functions, for example file server, database server, or a workstation for an end user?
- What is the equipment in use for these system types?
- Is the installation of the operating system and applications based on standard installation and standard configuration?

As an example, if Windows NT application servers always had the office applications stored in the same directory path (including drive letter), we would only need to specify one resource record in a security profile to manage that same resource on many different Windows NT application servers.

2.3.2 Naming Conventions

Users and applications often access computer resources and objects, such as computers, shared directories, and printers, by using the name assigned to the resource. If there is a naming convention where the names are consistent for all variations of resource types, it helps the user to select quickly the proper resource for their work and simplifies administration.

There are several requirements to defining an appropriate name for computer resources. From the point of view of a user, resource names should be easy to keep in mind, and the type of resource should be easily determinable from the name. In order to automatically execute business tasks implemented through programs and scripts, the names may also have to be configured within programming variables.

Several points must be covered in the context of naming conventions:

- Identify the group or individual within the organization responsible for defining the naming convention.
- The name must refer to only one resource; it must be unique.
- If the name should be built in some standard way, identify each part and the delimiter to be used for separation. For example, a user login name may be made up of the user's first initial and the first seven characters of their last name, or a resource name may include some indicator of its location, its type, and a unique part, each separated by an underscore (_) or a period (.).
- Establish whether there should be a mechanism to validate the uniformity of names for newly defined resources. In the Tivoli enterprise software environment, we can use validation policies to enforce this.
- Check for any dependencies on other naming conventions - such as may already be in use in some other context within the organization.
- Consider limitations for the characters used and length of names.
- Consider the need for platform-specific naming conventions for certain resources.

This section continues by listing commonly-used objects in a computer environment you should consider when figuring out the naming convention. These objects are:

- TCP/IP names
- Microsoft-style computer and domain names
- User and group names
- Tivoli enterprise software objects

2.3.2.1 TCP/IP Names

A TCP/IP name indentifies a computer that can be connected to/from another machine with the TCP/IP protocol. It is easier for the user to address a computer with the TCP/IP name instead of the IP address.

The structure of a TCP/IP name consists of two parts. The first part, often referred to as the host name, usually represents the computer‘s machine name, then there is a period character, and the second part is the domain name suffix. The domain name is used to assign the computer to an organization and usually incorporates some hierarchy, separating levels in the hierarchy with a period.

RFC952 states that every TCP/IP name must start with a letter and RFC 1123 extends this to allow a number. These are followed by further letters, numbers, or two special characters, the hyphen (-) and underscore (_). The case is typically ignored by TCP/IP applications.

In the Tivoli Management Framework, the host name without the domain name suffix is used during installation as a name for the management database and the directory where the database is stored. In addition, it represents the resource name of managed nodes, because it is better to have short names for often used resources. In order for this to work well for managed nodes, it is best to ensure that the hostname without domain suffix is unique.

2.3.2.2 Microsoft-Style Computer and Domain Names

The Microsoft-style computer and domain name are applied if you use windows applications to access the network through the NetBEUI protocol (the NetBIOS extended user interface). NetBIOS is the standard API for Microsoft network products.

These names are necessary to know if the resources of a computer will be shared with many end users. Typically, a user will browse the network looking for the name of a server known to contain a particular resource, or the user will directly connect to a share from the server if the name and resource name are known.

Microsoft allows a computer name to have a maximum length of 15 characters, including letters, numbers, and the special characters ()_`{}~^&%!#$. Again, the case is ignored for most operations.

2.3.2.3 User and Group Names

Every user logged into a system has a login name, and the login name can be assigned to one or more system-specific group names. Outside of Tivoli Security Management, the users’ membership in a selection of system groups defines their access rights to resources, because the access to resources is typically granted to groups, rather than the individual users.

- Check the organization structure.

Quite often, companies use a structure for user and group names to describe the organizational relationship.

- Equal user names on all platforms.

  Should/can a user have the same login name for all supported platforms?

Note that when we move to Tivoli Security Management, we will not necessarily be concerned with the use of system groups. System groups require the administrator to collect individuals who will require the same access to specific resources on that system into those groups. If the same user needs access to resources on another system, even one of the same type, they will need to be a member of another group with the appropriate access. In Tivoli Security Management, we will group users according to their job title. This will usually mean a group structure much more closely aligned to the organizational structure than with system-based groups. The access to resources will be defined for security management groups through security roles. This topic is discussed further in 2.3.5, “Groups and Roles” on page 32.

2.3.2.4 Tivoli Enterprise Software Objects

If Tivoli enterprise software products have already been installed, there will hopefully be a naming convention for all current resource types. Each product may have many different resource types, for example:

- Framework

  TMR, Generic Collection, Administrator, Policy Region, Subregion, Managed Node, Endpoint, Profile Manager, Task Library, Task, Job

- Distributed Monitoring

  Sentry Profile, Indicator, Indicator Collection

- Tivoli/Enterprise Console

  Event Source, Event Group, Event Classes

- Logfile Adapter

  Adapter Configuration Profile (ACP)

- User Administration

  User Profile, Group Profile

- Security Management

  Security Profile

A new object can be created for each resource type. The TMR server guarantees that the object identifier of this resource is unique in the whole Tivoli Managed Region (TMR) even if other TMRs are connected (so long as

resources are exchanged between connected TMRs). However, the name of objects represented by the field label can reside several times in the Tivoli environment. Therefore, it is possible to have various resources on the desktop with identical names. For example, it is possible to have a profile manager called Development, as well as a Tivoli User Administration group profile called Development. The type of object in use may or may not be obvious from the context of the desktop, but it may not be clear when working with the command line. For this reason, it is recommended to implement a naming convention that specifies the object type within the name, such as Development-PM, where -PM indicates a profile manager.

You can choose any letter or numeral for the resource name. Spaces and special characters like hyphen (-) and underscore (_) are also allowed. Note that for all operations in Tivoli, the names *are* case sensitive. Be aware that when spaces are included in names, it becomes necessary to surround parameters in quotation marks when referring to them on the command line.

2.3.3 Tivoli Environment

If the company already operates with Tivoli, we need full documentation of the existing management environment. The topics in this chapter help us to collect the necessary information.

For each Tivoli product in use, you can start the investigation with the following questions:

- What Tivoli product components are in use?
- What version has been installed?
- Who are the responsible product managers?

  You will need to talk with managers or delegated staff responsible for the Tivoli products to answer questions about the Tivoli environment.

For this assessment, we will consider only Tivoli products that are relevant for security management issues. These products are:

- Tivoli Management Framework
- Tivoli Distributed Monitoring
- Tivoli Enterprise Console
  - Logfile Adapter
  - Windows NT Event Log Adapter
- Tivoli User Administration
- Tivoli Security Management

Please remember that the commands listed in the next chapters can only be executed if the Tivoli environment has been set. The environment can be set with these shell scripts (paths shown are the defaults):

- `/etc/Tivoli/setup_env.sh` for UNIX platforms - default shell
- `\WINNT\system32\drivers\etc\Tivoli\setup_env.cmd` for NT platforms

On Windows NT, people often include the `setup_env.cmd` in a directory that is always in the path, or they set the same environment variables permanently through the Control Panel.

2.3.3.1 Tivoli Management Framework

As you should know, the Tivoli Management Framework, or in some cases, the Tivoli Management Agent is a prerequisite for the Tivoli products below. It must be installed on UNIX and Windows NT machines in order to protect computer resources with Tivoli Security Management.

Note

Even with the new endpoint support provided by later versions of Tivoli Security Management (the Tivoli Management Agent), UNIX systems will still need to have the full TACF product installed on them to be protected. Use `winsttacf` or the GUI to perform this installation.

Find out how many Tivoli Management Regions (TMRs) are installed. Each server can have several one- or two-way-connections to other TMRs. Sometimes, they are arranged in a two-layer hierarchy, with a top tier TMR connected with one- and two-way connections to all the other TMRs (see Figure 4). These inter-region connections are responsible for the exchange of management information between them. It is a good idea to show these dependencies in a map.

Because a TMRs resource data can only be exchanged with an adjacent TMR, multi-TMR installations often result in one TMR being connected to all the others. This configuration is sometimes known as hub and spoke and is shown in Figure 4.

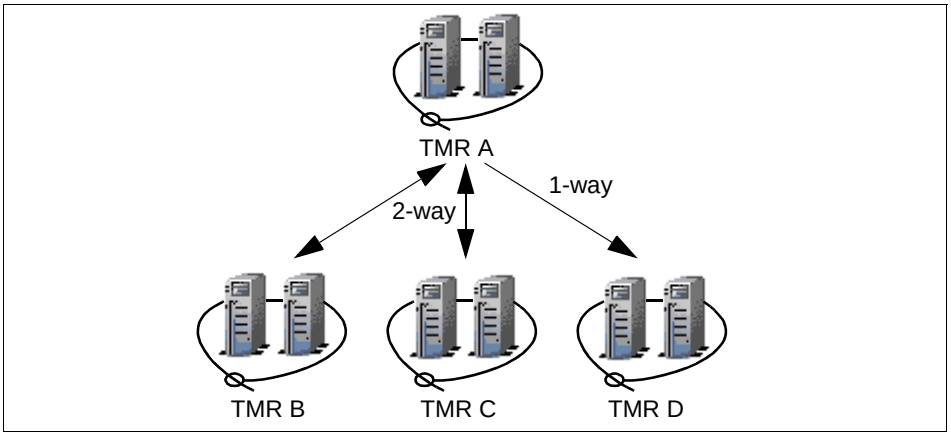


Figure 4. TMR Hierarchy and Connections

Table 1 provides a suggested list of information to gather for each TMR server.

Table 1. Collecting Information: Tivoli Management Framework

Information	How To Get the Answer
Directories used, and the file permissions of the Tivoli database, binary, and library files	On UNIX use the command: `ls -ld $BINDIR $LIBDIR $DBDIR`
	Use the Windows NT Explorer on Windows NT platforms to determine the directory access rights for results of: echo `%BINDIR% %DBDIR%`
Determine the password to install products on the TMR server (if in use)	It was typed in during the installation
Check if remote client login allowed	`odadmin odinfo 1`
Check if Kerberos is in use	`odadmin odinfo 1`
Encryption level for communication between object dispatchers within the local region	`odadmin odinfo 1`
Encryption level for communication between Tivoli servers	Tivoli Management Framework will use the level of the target server to communicate with it.
List of inter-region connections to Tivoli servers	`wlsconn`

Information	How To Get the Answer
List of Tivoli Administrators and their properties	`wlookup -r Administrator -a` `wgetadmin <administrator>`
List of Managed Nodes	`wlookup -r ManagedNode -a`
List of Profile Managers to which each Managed Node subscribes	`wlssub @ManagedNode:<label>`
List of Task Libraries and their contents (Tasks and Jobs)	`wlstlib -a`
Note: It is recommended that the commands be used on each TMR. The `wlookup` will list objects known to all connected TMRs but only if the resource exchanges are up to date.	

2.3.3.2 Tivoli Distributed Monitoring

To improve the availability of your computer environment, it is necessary to monitor its resources. This can be done with Tivoli Distributed Monitoring. The values determining how to monitor a resource are stored in the distributed monitoring profiles, the profile type being SentryProfile.

We need a full description of all monitor definitions distributed to managed nodes. It is not enough to consider just what we might think of as security-relevant resources. We should find out about all monitored resources, because monitors are normally defined for critical resources.

Table 2 shows the information required in the context of monitors and how to get the information.

Table 2. Collecting Information: Tivoli Distributed Monitoring

Information	How To Get the Answer
Determine the structure of Policy Regions and Profile Managers in which the SentryProfiles are stored.	Ideally, this will be documented by an administrator. Otherwise, you will need a description that you should then verify.
Installed Monitoring Collections	`wlsinst -a`
Defined Sentry Monitors and their properties	`wlsmon <sentry profile>`

The output about the properties of monitors is very unstructured. Therefore, we suggest you extract the information for each monitor on a sheet and add the following information:

- What is the purpose of the monitor?

- Who is responsible for maintaining the monitor, especially the thresholds?
- The list of managed nodes that subscribe to the monitor.
- Document each failure notification and actions that are executed automatically with a process diagram.
- Who takes care of the monitored resource if a threshold is passed?

2.3.3.3 Tivoli Enterprise Console (TEC)

This product is responsible for the event management. It receives the events from the whole Tivoli management environment.

The TEC components, Logfile Adapter and Windows NT Event Log Adapter, described in sections 2.3.3.4 and 2.3.3.5, respectively, are able to generate the security events. (Note that Tivoli Security Management provides TEC rules for security-related events that we will probably wish to employ. The *Tivoli Security Management User's Guide* has a chapter on Events and Rules.)

In order to define new events, we need an overview of all defined event classes and the configuration of the event console, as listed in Table 3.

Table 3. Collecting Information: Tivoli Enterprise Console

Information	How To Get the Answer
Name and directory of the Rulebase	`wlsrb -d`
File permission of directories	On UNIX use the command: `ls -ld <rulebase directory>`
	Use the Windows NT Explorer for Windows NT platforms to determine the directory access rights for: echo `%BINDIR% %DBDIR%`
Event Classes	`wlsrbclass <rulebase name>`
Event Sources	`wlssrc`
Event Groups	`wlseg`
Event Filter	`wlseg -f`

2.3.3.4 Logfile Adapter

The TEC logfile adapter can be used to observe messages from the operating system, or applications that are written in log files, and to forward events to TEC from those files.

In most cases, the logfile adapter monitors all messages from the UNIX syslog daemon and scans for particular messages. In order to decide what is important, the adapter has a format file that contains search strings and statements to extract information from the log file. Then, the adapter generates an event object and sends it to the Enterprise Console. The same functionality for monitor messages applies to every kind of log file.

This product is only available for UNIX platforms. On Windows NT, there is comparable functionality to monitor event messages. See 2.3.3.5, “Windows NT Event Log Adapter” on page 29, for more details.

We need a description of every logfile that is already monitored. Table 4 contains attributes of a logfile adapter we need to know.

Table 4. Collecting Information: TEC Logfile Adapter

Information	How To Get the Answer
Logfile Adapter Path	`echo $TECADHOME`
Adapter Configuration File	`ls $TECADHOME/etc/*.conf`
Adapter Format File	`ls $TECADHOME/etc/*.fmt`
Adapter Object Indentifier File	`ls $TECADHOME/etc/*.oid`
Adapter Configuration Profile	`wlookup -r ACP -a`

If Adapter Configuration Profiles (ACP) are in use, you can save a lot of time. In this case, all desired information about logfile adapters is defined in ACP records.

2.3.3.5 Windows NT Event Log Adapter

The TEC Windows NT Event Log Adapter is used to extract information about the events from the three standard logs.

System log General system-related events

Application log Application-related events

Security log Windows NT security events (logon, logoff, audited objects, and so on)

We need a list of events that are already monitored with the NT Event Log Adapter (see 2.3.12, “Auditing” on page 44, for details on how to get this list). If this adapter is configured with an ACP, determine the properties from the ACP record.

2.3.3.6 Tivoli User Administration

Although it is not a pre-requisite, it is very likely in a Tivoli Security Management environment that Tivoli User Administration will be in use to administer user accounts. Table 5 lists suggested information to gather.

Table 5. Collecting Information: Tivoli User Administration

Information	How To Get the Answer
Custom categories and attributes for user records	Ask the administrator and/or use: `wlsusrcat` and `wlsusrsubcat`
Default policies for user and group profiles	Select edit -> default policies in the GUI or use `wlspol -d`
Validation policies for user and group profiles	Select edit -> validation policies in the GUI or use `wlspol -v`
Distribution actions	Ask the administrator or use `wlsactions`
Organizational assignment of user and groups to profile manager(s) and their policy regions	Ask the administrator
Name of user and group profiles	`wlookup -r UserProfile -a` `wlookup -r GroupProfile -a`
Subscribers of user and group profiles	`wgetsub <profile manager>`
List of user records of each user profile	`wlsusrs`
List of group records of each user profile	`wlsgrps`

Many attributes of a user record have security-related aspects. Therefore, identify the security policy for the following user attributes:

- Time restrictions for login
- Password Aging
- Audit Control

2.3.3.7 Tivoli Security Management

This is what you will use to assign access rights of computer resources to user accounts through group associations. It may be, of course, that Tivoli Security Management has already been deployed to some extent. The information about resources we want to protect and the assessment of

auditing and resource controls are stored in the Security Profiles. Table 6 shows what to collect.

Table 6. Collecting Information: Tivoli Security Management

Information	How To Get the Answer
List of security profiles	`wlookup -r SecurityProfile -a`
Attributes of each security profile	`wlssec <security profile>`
Default policies for user and group profiles	Select edit -> default policies or use `wlspol -d`
Validation policies for user and group profiles	Select edit -> validation policies or use `wlspol -v`

2.3.4 System Policies

System policies define the rules for identification and authentication of the users to the systems in the networked environment and set the standard or default access to the business resources.

In order to design a system policy, we need to know the current policies that are implemented in the environment. This section contains tables that can be used as forms to fill in the required policy for different subjects. Note that these tables are for the standard supported platforms of Windows NT and UNIX. You may wish to adapt these tables as other platform support is enabled, or at least, use them as a guideline for the areas that need consideration. The system policy attributes are listed and described in the *Tivoli Security Management User's Guide*. Note also, that not all attributes are supported on all platforms. Part of the policy decision-making process will be to decide whether to use a lowest-common-denominator approach, where the same policy will be applied to all systems as far as that is practical. Alternatively, the policy may be to use as high a level of security as each platform allows.

We need to know how inactive computer accounts are handled, and what should happen if a user repeatedly enters the wrong password. These and similar attributes are summarized as *login policy*. Table 7 lists them all (you can use it as a form and fill in the required values). Attributes that are not supported by a operating system are marked n/a.

Table 7. Login Policy Attributes

Attribute	UNIX	NT
Suspend inactive accounts		n/a
Lock account upon multiple logon failures		
Limit grace logins		n/a
Limit concurrent logins		n/a

We also need to get the information about the characteristics of a user's password. This defines things, such as how many and what type of characters the password must contain. These and more attributes are included in the *password policy*. Table 8 lists all the topics we may need to cover in our design.

Table 8. Password Policy Attributes

Attribute	UNIX	NT
Minimum days between password changes	n/a	
Maximum days between password changes		
Minimum password length		
Minimum alphabetic characters		n/a
Minimum alphanumeric characters		n/a
Minimum numeric characters		n/a
Minimum uppercase characters		n/a
Minimum lowercase characters		n/a
Maximum repeated characters		n/a
Minimum special characters		n/a
Password history depth		

2.3.5 Groups and Roles

The security resources that we will discover in later sections are assigned to groups of users to provide the proper access rights. In Tivoli Security Management, we will use groups in a different way than in any existing configuration. In Tivoli Security Management, groups of users can have access control defined to a wide variety of resources across multiple

platforms. The access capability is defined in security roles that are assigned to the groups. This is unlike the typical use of groups at a system level where access capability to resources may be assigned directly to either groups or users. In Tivoli Security Management, a group should represent a real-life group, department, or collection of individuals with a similar job description, and is not simply a way of grouping system user records to define access.

We, therefore, need to know which real-life groups exist in the company. For every group the following questions must be answered:

- What function do members of the group represent?
- Who are the members of that group?

In addition, you need to collect information about existing system-based groups to determine access requirements that will be fed into the roles and groups in Tivoli Security Management. See the following discussion on Roles for more details.

Tivoli Security Management introduces a third tier in the usual group or user and resource access relationship. This tier is called the *role*. The role is a collection of definitions of access capabilities to resources. The idea is that the access required to resources usually revolves around the role, or job function, a user has. For example, if a function of your job (a role) is to review programs, you are likely to be allowed to read what the programmers develop but you may not have the access to change their code. Any one person in an organization may have many different roles or job functions as part of their job. Someone who reviews programs will have other roles such as writing reports or organizing the team meetings. The overall job title for this person will usually determine which group they are in, and the job functions a person with that job title performs will determine what roles that group needs to have in order to access the right resources. So in this case, a person with the job title Code Reviewer will have the roles Review Program, Write Code Reports, and Meeting Coordinator.

To write your review about the current program release, you need the capability to read and write the review reports directory resource. This example is shown in Figure 5. A member of a group with the Develop_Program role has no rights over the /ReviewReports tree but can read and write the /sourcecode/prg1 resource. A member of a group with the Review_Program role can read from the /sourcecode/prg1 resource and read and write the /ReviewReports/prg1 resource.

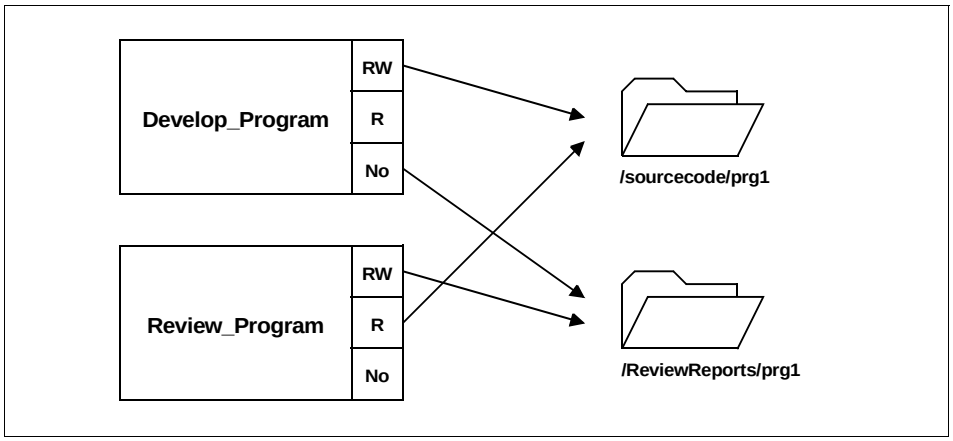


Figure 5. Example Role Definition

We will have to determine the roles that are defined in the company at this granularity in order to implement security management. It makes no sense to assign the resources to the users, or even groups directly, because that will circumvent the ability to determine access across a range of resources on different platforms and will be too complex to manage centrally.

If the roles are not defined in the granularity of the above example, there will need to be some sort of workshop or brain-storming session to determine what they should be.

The primary input for such a workshop would be an organization chart. This will need to include the job descriptions of all jobs that need to have access to computer resources. You will also need the output of the other sections in this chapter, which determined the computer resources themselves. As participants, you should invite people who know what every job is about. For big companies, there may need to be a multi-step approach involving managers or team-leaders of the different departments. They should know best how to describe the jobs of the people in the department. In some cases, it may be necessary to invite an employee directly to obtain a description of the roles for a job title.

An important input when determining appropriate access for roles will be the existing group configurations. If an organization makes extensive use of access control through system groups, you can use them to determine the types of accesses required. The starting point is to determine on which systems the existing groups are defined, and then, list out the groups and memberships.

Because it can be difficult to find out which group(s) the users belong to, we suggest the following approaches for the platforms AIX and NT:

- On NT, the tool `findgrp.exe` is a part of the Windows NT Server Resource Kit. This is quicker and easier to use than the `NET LOCALGROUP` and `NET GROUP` commands.
- For AIX systems, there is the command `lsgroup ALL`.
- Substitute the relevant commands for your platforms.

If the company already uses Tivoli User Administration to manage user accounts, it is much easier to get the group membership for the entire company at once. The `wlsgrps` command delivers the group names of the entire company and the users that belong to them, for example:

```
wlookup -r GroupProfile -a |
while read GROUP_LABEL GROUP_ID
do # Get a list of group names and users that belong to

   wlsgrps -l @GroupProfile:$GROUP_LABEL
done
```

We suggest the form in Table 9 for collecting the appropriate information from the user groups.

Table 9. System Groups Attributes

Attributes of a User Group	Sample Values
Group name[1]	`system`
Description[2]	`Maintaining critical system resources`
Name of the system on which it resides[3]	`rh2430b.itsc.austin.ibm.com`
Operating system and level[4]	`AIX 4.2.0.0`
Responsible job role[5]	`System Administrator`
Group members[6]	
User members[7]	`root`

1 Name the group.
2 Describe the job function of the group members.
3 On what platform is the group defined.
4 Enter the operating system and version.
5 Identify the job role responsible for maintaining the group, such as assigning users.
6 In the case of Windows NT, a local group may have a global groups as members.
7 Users that are members of the group.

To show you what is required or what a workshop should achieve we will now go through an example. For this example, we use a simplified view of the ITSO organization. The simplified organization chart is shown in Figure 6.

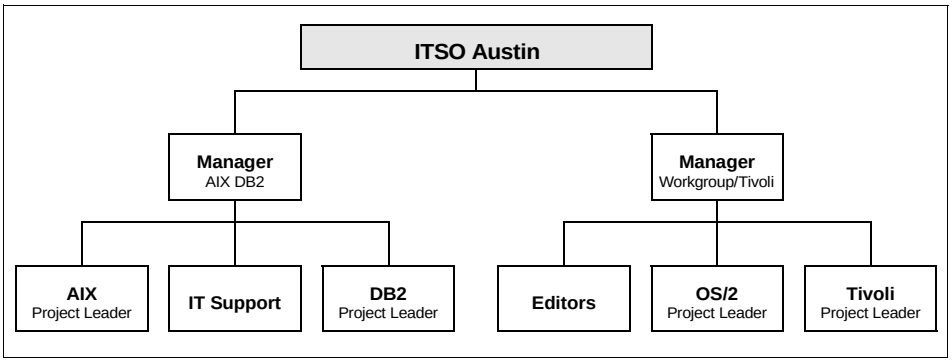


Figure 6. Simplified Organization Chart of the ITSO Austin

You will need to take two steps.

1. Brainstorm the roles.

   You take every job description, one after the other, and brainstorm the roles. Do not forget to define the role term to the participants first. We are looking for security related roles. Therefore, when defining a role, verify if this role really needs one of the resources you determined in the other sections of this chapter, for example, “Data Resources” on page 38 or “Terminal Access” on page 42. We are only concerned with those roles, because this assessment is for implementing security management.

   Looking at the organization chart of the ITSO, we see a lot of different job titles. Let us take the job of the Tivoli Project Leaders. Brainstorming the security related roles or job functions of a Tivoli Project Leader might result in the following:

   - Edit Redbook Files.
   - Print Redbook.
   - Exchange Files with Tivoli.
   - Send E-Mail.
   - Update Project Database.
   - Read ITSO Public Database.
   - Maintain Project Homepage.
   - Read Framemaker Application Image.

This list is not complete, but gives you an idea of the roles to define for a certain function. Note that every role is related to access requirements for resources. Brainstorming the next job functions, you will discover that there are many common roles to different job functions - everyone in the ITSO will probably print redbooks for example. This is the idea of roles. Once we define all the roles that exist in an organization, giving groups access to the right resources simply becomes a matter of adding the correct roles to the group definition.

2. Define resource access rights.

   Take every role description you found in step one and define the resources the role needs. You do not have to figure out the specific name of the resource and the system it resides on at this stage, but a uniquely identifiable name such as customer database, product image files, or 3rd floor laser printers, will suffice. Define the access of the role to the resources, such as update the customer database.

   During this process, you may find roles defined in previous steps that are not granular enough. It will be necessary, then to break them down into smaller roles.

   The Tivoli Project Leader's job description (and, therefore, security group) might have the role assignments listed in Table 10.

Table 10. Role-Based Resource Access Example

Role	Access	Resource
Edit Redbook Files	Read/Write	Redbook Files
Print Redbook	Print	Printer Pool ITSO Austin
Exchange Files with Tivoli	Read/Write	Shared Tivoli/ITSO Filesystem
Send E-Mail	Read	Lotus Notes Program Image
	Read/Write	Mail Database
Update Project Database	Read/Write	Project Database
Read ITSO Public Databases	Read	ITSO Public Databases
Maintain Project Homepage	Read/Write	Project Web Filesystem
Read Framemaker Image	Read	FrameMaker Program Image

While going through this process, you might find some resources you did not think of in the other steps. Of course, you can now go back and add that resource to the list of resources to be protected.

2.3.6 Data Resources

One of the most important types of resources in a company are data resources. Data resources include files, directories, UNIX file systems, and Windows NT shares. In order to manage the access rights to the data from a central point with the Tivoli Security Management product, we need an overview of files, directories, and shares that are supposed to be managed. You have to find out where the business resources are located and for what they use each file or directory. As there are many things to consider when listing those resources, we provide, in Table 11, a form that shows the attributes to write down or generate electronically. This makes it easy to record the data resources in a structured way. The following table shows an example data resource. Note that there is a blank version of the table to copy or adapt for your own use in Appendix A, “Planning Forms for Security Management Design” on page 117.

Table 11. Data Resource Attributes

Data Resource Attributes	Value
Resource name [1]	d:\public\images\wordpro
Name of owning system [2]	FILESVR1
Resource type [3]	[ ] File [x] Directory [ ] Share
Operating system	[ ] UNIX [ ] NT
Owner [4]	RH2430C\Administrators
Access [5]	Everyone Read (RX) RH2430C\Administrators Full Control (All) SYSTEM Full Control (All)
Audit	[ ] Success [x] Failure [ ] None
Responsible job function [6]	WordPro application administrator
Function / description [7]	In this directory, you will find the official and approved release of the Lotus WordPro application installation image for use in the entire organization.

1. Name of the resource. Example: /home/ausres47 (UNIX), c:\winnt\system32 (NT).
2. Example: rh2430b.itsc.austin.ibm.com (TCP/IP hostname), rh2430c (NetBIOS computer name) If this is a standard resource, such as a temporary directory, then use the type of machines (for example fileserver).
3. For the resource type Share, see note number seven.
4. The owner of the resource as reported by the file system. Example: root (UNIX), Administrator (NT).
5. Access rights as reported by the file system. Example: rwxr-x--- (UNIX), Everyone Read(RX)/ Administrators Full Control (All)(NT).
6. Describe who is responsible for the resource and job function.
7. Describe the purpose of the resource. This could be: accounting data, database containing employee data, or user directory of user ausres46. If the resource is a Share on a Windows operating system, name the local resource here.

You might start by highlighting the most important resources. Depending on the level of protection you are trying to achieve (see 3.1, “Protection Levels” on page 47), you may be able to concentrate on a comparatively small number of data resources (say in the 10s). Using one sheet of paper for every resource makes it easy to sort and classify them. You could even organize them into roles and groups by sticking them on a wall and using a rubber band or string to make connections.

If it is necessary to deal with larger quantities of data resources, it may make sense to modify Table 11 to a list format with a column for each value and a row for each resource.

Please note that this is an assessment of the current settings. Fill in the values as they are now.

At the time of writing, there was a really helpful tool from Somar Software for working with ACLs on Windows NT. It is a piece of shareware called `dumpacl`. We found it at `http://somarsoft.com`. If you need a thorough examination of Windows NT resources, we encourage you to use this or a similar tool. With this tool, you can scan entire drives for file and directory permissions. It also dumps the audit configuration. It has a GUI and a command line interface. Figure 7 is a screen-shot of the `dumpacl` GUI.

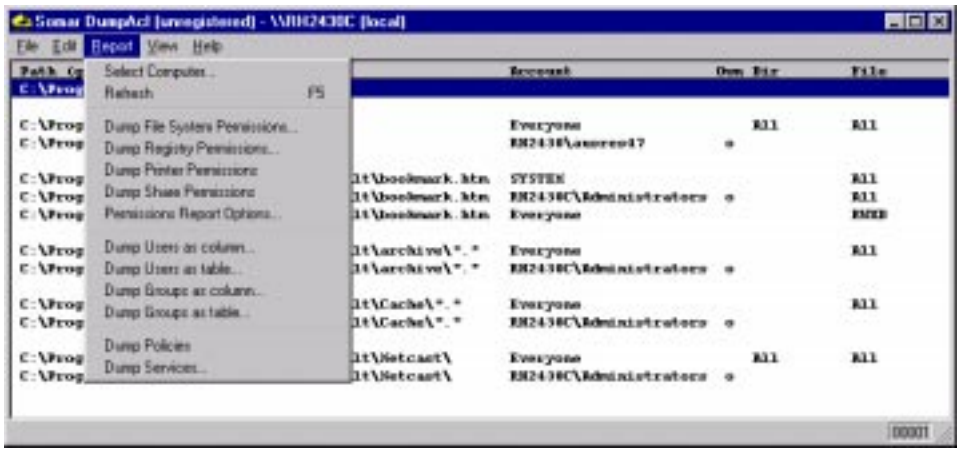


Figure 7. Dumping File Permission Data on Windows NT

We would recommend extracting the relevant data from the output of this sort of tool and putting it into our table. The standard Windows NT command for dealing with ACLs is `cacls.exe`.

2.3.7 Network Connectivity

It is usual that every computer is connected to a network. Without any security, everybody who has access to that network would be able to work with that remote host using telnet or remote login services. Apart from these remote access facilities, applications could use Remote Procedure Calls (RPC) to gain access to system resources.

If we can limit the ways hosts can operate with a server, we can reduce the openings for either accidental or malicious attacks on the system. Tivoli Security Management gives us the capability to block incoming requests from another network computer on all supported UNIX platforms. It is similar to protecting resources by restricting who can log in locally. They may not establish a connection to any computer they want. It may also be desirable to limit outgoing requests, too.

For every UNIX machine, we have to identify the allowed TCP/IP services used for remote access and the list of hosts that may connect remotely. Services we do not really need to fulfil the hosts function can then be deactivated or restricted.

The TCP/IP requests can be classified with the direction of the communication:

- Incoming TCP/IP requests from remote hosts
- Outgoing remote connection requests of local users

Note

Restricting inbound or outbound connections on a UNIX platform requires a detailed understanding of what is involved. For example, by default, FTP from a client to a server will usually require that the server can make a connection back to the client. Other examples include the use of specific TCP/IP ports.

2.3.7.1 Incoming TCP/IP Requests

Determine if there are TCP/IP services (/etc/services) that are necessary to use, and then, create a list of hosts that may operate with those services. In addition, define the default access, the audit mode, and the access date/time

restriction required for the resource. Table 12 is a suggested form for collecting the appropriate information.

Table 12. TCP/IP Services Attributes

Attributes of TCP/IP Services	Values
Resource name[1]	`telnet`
Description[2]	`Remote terminal access for system administrators`
Name of the system on which it resides[3]	`rh2430b.itsc.austin.ibm.com`
Operating system[4]	`AIX 4.2.0.0`
Responsible job function[5]	`System Administrator`
Host accessors[6]	`9.3.1.211,` `*austin.ibm.com`

[1] Enter the service name or port number defined in /etc/services or /etc/rpc.
[2] Describe the TCP/IP service and explain the reason for using it.
[3] The hostname or IP address of the host.
[4] Enter the operating system and version of the system.
[5] Identify the job function responsible for maintaining this TCP/IP service.
[6] List of hosts that may have access to the TCP/IP service described above.

2.3.7.2 Outgoing Remote Connection Requests

Obtain a list of hosts that users have to connect to/from a managed node. In addition, define the default access, the audit mode, and the access date/time restriction for the resource, if required by the policy. Table 13 is a suggested form for collecting the appropriate information.

Table 13. Remote Connection Attributes

Attributes of Remote Connection	Values
Resource name[1]	`rh2430a.itsc.austin.ibm.com`
Description[2]	`Remote terminal access for system administrators`
Name of the system on which it resides[3]	`rh2430b.itsc.austin.ibm.com`
Operating system[4]	`AIX 4.2.0.0`
Responsible job function[5]	`System Administrators`
User group accessors[7]	`Administrators,` `Security_Administrator`

Attributes of Remote Connection	Values
[1] Enter the hostname that may be connected to/from the local host. [2] Explain the reason to connect to that remote host. [3] The hostname or IP address of the local host. [4] Enter the operating system and version of the local host. [5] Identify the job function responsible for maintaining this remote connection. [6] List of user groups that may have access to the remote host described above.	

2.3.8 Terminal Access

It may be required to restrict the workstations from which terminal sessions can be made in the company's computer environment (supported for UNIX in Tivoli Security Management). This is often used to enable employees to work only at their workstation or to restrict the root access from all machines but the system itself. We will have to find out if the company implemented such security features and how they are realized. There can be a native UNIX implementation, for example, for root access by adding `secure` to the terminal line in the file `/etc/ttys`, or there may be a third party product implementation.

For the Windows NT platform, there is also the possibility to restrict the user to logon only from a limited number of workstations. In Tivoli, this is managed through Tivoli User Administration though, not Tivoli Security Management. If the company uses native Windows NT user administration, we can find this information in the **User Manager for Domains**. Select one or a collection of users and show their properties by clicking on the menu item **User->Properties...**. The user administration dialog comes up. Click the **Logon To** button, and you will receive the dialog shown in Figure 8.

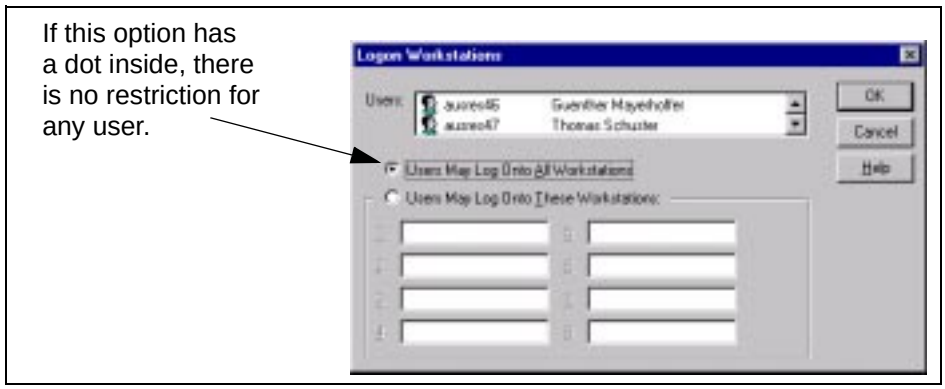


Figure 8. Determining Workstation Restrictions on Windows NT

If the option **Users May Log Onto All Workstations** is selected, there is no restriction for any user. If no option is selected, there is at least one user in the list of multiple users you selected with a workstation restriction. You will have to go through each of the users to find out which one(s). You can also use the `NET USER username` command to display whether the user is only allowed to log on at specific workstations.

If the NT users are already managed by Tivoli User Administration, you can find out in the **User Properties** of a Windows NT user in the relevant user profile. Select the **Category NT** in the drop-down box, and then, select **NT Workstations** from the list. In the right side of the dialog, a list of workstations appears. If this list is empty, the user has no workstation restrictions.

2.3.9 Applications

In order to manage the access to the applications, we need a snapshot of all the programs that are in use that need protecting. We have to figure out if and how these applications are secured right now. The following questions will help find that out:

- Is any application security currently implemented?
- If yes, how is it realized (for example, through some add-on product)?
- Are the applications only secured at the file level, or do the applications provide their own security mechanism?
- Which users, groups, and roles have access to a particular application?
- Are the applications protected against modification?

You should consult every application administrator to determine how the application is currently secured and that the security is satisfactory. It may be that it is not necessary to incorporate that application into Tivoli Security Management protection mechanisms - or it may be that the current method of protection is best replaced by some feature of Tivoli Security Management.

2.3.10 Critical Files

Tivoli Security Management will allow us to maintain information about critical files in a system in a similar way to watching an application executable. We can identify files as being critical and be alerted to changes to those files. This is unlikely to be implemented in an existing environment, except perhaps, with some add-on product. A security policy may specify certain files as being critical that can only be trusted if the file details remain constant.

2.3.11 Machine and Operating System Types

For every implementation of a system management product, you need to develop a strategy to deploy the solution. For this, it is necessary to have a detailed plan of the computer environment that shows the quantity and names of systems, and any standard naming practices in use. This means determining the following things:

- How many and which system types have to be managed?
- How many and which system types are there for each particular operating system?
- How many and which system types meet a particular standard (for example, file server, database server, user workstation)? (See also Chapter 2.3.1, “Incorporate Standard Installation Requirements” on page 20.)
- The ownership of the systems (list quantity and types by department or business unit).
- How many and which system types are already Tivoli Managed Nodes?

Normally companies hold databases for this information, so it may be possible to obtain these lists in an automatic way. That makes it easier to process the information in order to deploy the security solution.

With this data, you have a good basis for deciding how to roll out a security implementation. For example, a roll out may be better performed platform by platform, or perhaps, department by department.

2.3.12 Auditing

Auditing is a very effective way of building statistics, finding an intruder, or just locating the person who performed some action of interest, such as deleting a file system. As we want to design and implement a smart auditing solution, it is necessary to know what events are currently generated and logged.

We need to ask the company what their logging strategy is up to now. This should include the following aspects:

- Is auditing already implemented, if so which events are logged?
- Often systems perform local logging. If the goal is to centralize the audit data, what events shall be logged on that central point?
- What are the log files and where are they located?
- How are these log files structured?

- What are the access rights to the log files?
- Who uses the log data?
- Do they use the log data for statistics or just for security purposes?
- Is there an action plan for responding to events?

Note

On the Windows NT platform, it is important to check whether the system's auditing is turned on. It is disabled by default.

Check Windows NT auditing in the User Manager or the User Manager for Domains. Select the menu item **Policies->Audit...**. The dialog is shown in Figure 9.

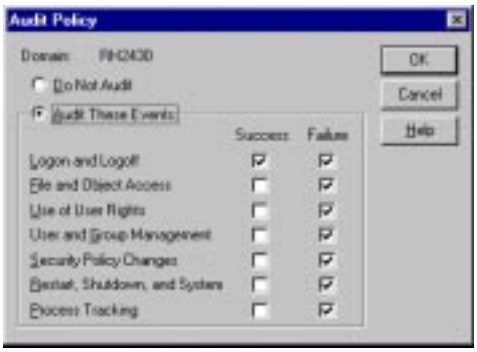


Figure 9. Windows NT Audit Policy

This shows you if, and which, events are audited. **Do Not Audit** is the default.

Now, you will have to generate a list of the resources that are audited in the company. If auditing is deliberately enabled, there should be a list of resources that are observed. If not, you will have to find out which logging level is used for each resource. To get a brief overview, it is a good idea to look into the security log of every system. To make this task a little easier in Windows NT, you can use the tool `dumpel.exe` from the Windows NT Resource Kit. You can even automate this job, because it allows you to specify the server you want the log to be dumped from.

Here is the syntax to generate a complete dump of the security log of a system:

```
dumpel -f dumpfile.txt -s <computer name> -l security
```

There may still be some resource audits that are not included, because no one ever accessed them in the way the auditing was configured. But at least, this gives you an idea what is audited on a system.

You can also use the shareware tool `dumpacl` which is able to generate complete audit lists over Windows NT file systems. For a description see Chapter 2.3.6, “Data Resources” on page 38.

2.3.13 Physical Security

Physical Security is an area which should certainly be covered in a security policy document but is something we have little control over in Tivoli Security Management. The factors we have to consider about physical security are what effects the physical security of systems have on logical security requirements. For example, if important data servers are placed throughout the enterprise in relatively unsecured locations, we will need greater protection against local use such as logons, than if the servers are stored in locked rooms with limited access.

Chapter 3. Tivoli Environment Security Architecture

We now know what the company currently achieves, and what they would like to achieve with a security implementation. We have the security policy document and an overview of the company's computer environment. With these inputs, we are now able to take the second big step from the method we showed in Figure 1 on page 4. Through the next three chapters, we will design the security management implementation for the company and write it down in a security architecture and design document. This document will be the principle resource for those performing the implementation. With the knowledge of the Tivoli Security Management product and with the architecture and design document, a person should be able to realize the given design. This chapter deals with those areas of the design related to the Tivoli Management Architecture and related topics, such as naming conventions. Chapter 4, "Resource Security Design" on page 67, continues by looking at the specific types of resources we will use Tivoli Security Management to protect, and Chapter 5, "Security Auditing" on page 99, will complete the architecture and design topic with a discussion on auditing as it relates to security management.

What we do now can be called *translating*. We translate the input we worked out in the previous step to something that can be implemented with Tivoli Security Management by going through all the necessary topics.

Throughout these next three chapters, we will give recommendations for implementing a certain policy. These recommendations are based on experiments, experiences, and recommendations from those who have performed some implementations.

It is important to verify if the suggestions are appropriate to your environment, the requirements of the company and the given security policy. Even if they are not directly appropriate, these suggestions should at least help you to understand the issues involved in the design of a security management implementation.

3.1 Protection Levels

The requirements discovered through phase one of our process will illustrate what level of protection we are trying to implement. By level of protection, we mean how strictly we will enforce rules and how deeply we will protect resources. This can vary, from a simple increase in auditing, to a complete lock-down of a server allowing only specifically defined requests to complete.

In a complex environment where strict levels of security are required, one implementation method would be to build up the protection gradually through different levels. This allows fast implementation of critical security measures and the phased introduction of the more detailed requirements until the full level required has been achieved.

Possible phases include:

- Implement a good auditing strategy on all platforms. Compared to the remainder of the deployment, this is perhaps the simplest measure with the largest benefits in terms of understanding what is happening and detecting problems as they occur.
- Implement System Policies. Simple strategies that apply to whole groups of platforms.
- Select and secure files, directories, shares, and printers.
- Add high-security measures, such as su protection, root delegation, and network connection restriction.

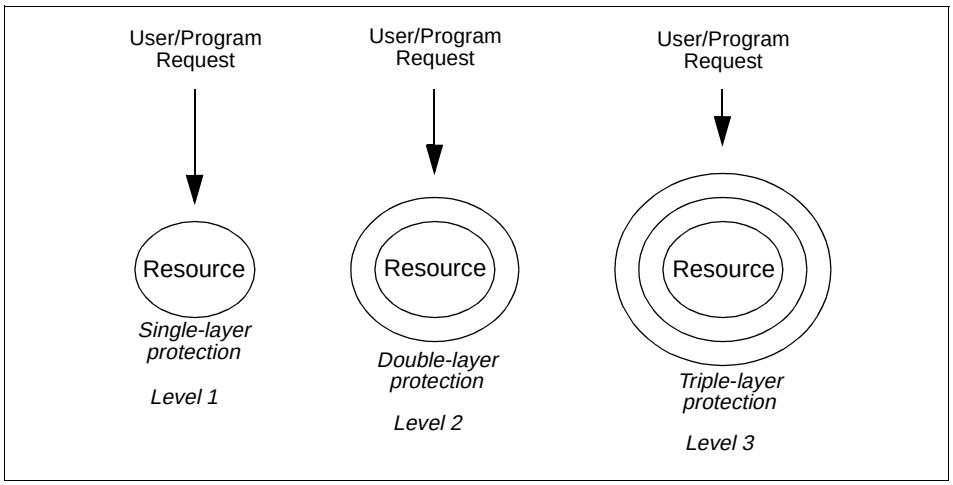


Figure 10. Increasing Levels of Security

Figure 10 illustrates tiered levels of security. The first or lowest level might be requiring user passwords to access a system. The second level might be that the resource can only be accessed by an authorized user from a system directly connected to the internal network. The third level might be that access is only available to an authorized user on a local system using a particular application and port (a capability of UNIX systems with TACF).

Designing and documenting a detailed level of security for a large number of resources at the outset would take a very long time. Instead, basic security measures could be implemented while the next phase of protection is being worked out.

3.2 Naming Conventions

Naming conventions represent a crucial planning detail. With precise planning, you can avoid any requirement to change names in the testing environment or - even worse - in production. A well designed naming convention makes resources easy to recognise and names easy to remember. What will work best in your environment will depend on many factors. This section discusses a number of suggestions.

When defining the naming conventions, we suggest the following guidelines to take into consideration:

- Keep the names as short as possible.
- An object's name should immediately identify the type of object.
- Use all same-case letters.
- Avoid special characters except for one basic sign, such as dot (.) or dash (-) to separate parts of a name.

3.2.1 TCP/IP Hostnames and NetBIOS names

The aim is to try to avoid administering two names for the same machine, such as the hostname for the TCP/IP protocol and the computer name for the NetBIOS interface. To do that we have to keep in mind the rules described in 2.3.2.1, "TCP/IP Names" on page 21 and 2.3.2.2, "Microsoft-Style Computer and Domain Names" on page 22. Combining the restriction of both names results in a string of up to 15 characters in which the first character must be a letter or number followed by further letters, numbers, or underscore characters. The case is ignored, but consider other platforms and uses of machine names when deciding what to do about case and name length.

We suggest a naming convention for machines that is as short as possible, easy for users to remember, and that also contains basic information about the machine itself. The first part of the machine name can be informative about the physical location, described with an acronym of the country, division, or department. The second part could represent the function of the machine. Further information, for example, the installed operation system, is not recommended because we would have to change the name if the

configuration is changed. Those sorts of details can be gathered with a configuration management tool, such as Tivoli Inventory.

Here are the details of a suggested naming convention for hostnames and computer names; you may not need to use examples from all of these suggestions in your environment:

1. Letters or numbers representing the country. This obviously makes sense for international companies. Even if an organization is not comprised of international components now, consideration should be given to the possibility of future changes. To represent the country with just one letter would be not enough for a large international company. Many countries adopt a system used in Europe, and elsewhere, to indicate the origin of a vehicle with an owner-applied sticker. This system designates countries by using up to three letters - GB for Great Britain, F for France, AUS for Austria, and so on. Another option might be to use the international dialling code for a country. This would require numbers up to three digits. To remain consistent, those that use less than three digits could be padded out with leading zeros. So, the United States becomes 001, the United Kingdom 044, Germany 033, and so on. Lastly, of course, you could devise a system of your own. Two letters would give over 600 (26 x 26) combinations, or using a letter or numeral in each of two digits, could locate over 1000 (36 x 36) places a machine resides.

   You can assign the machine to a division or location of the company, for example, the name of the city, or specify the location with a letter and number as already suggested. That could also be required if the company has multiple locations in one city.

2. One-letter function identifier.

   The function letter describes what the machine is used for. This is very helpful for users and automated processes to identify the task of the computer. Here are some possible classifications:

   w for Workstation
   a for Application Server
   d for Database Server
   P for Proxy Server

3. Numbers can be used to assure the name of the machine is unique.

   A range of four numbers, for example, is enough to identify up to 10,000 machines of the same type in each location. If you are looking for consistent names, you will need to identify the highest number of machines of any one type likely in any one location.

An example of one possible implementation follows:

A user in Germany in the Munich location has access to the following machines of the computer environment:

- `gmuw0105` is her workstation.
- Her application server is `gmua0001` in Munich.
- Her data is stored in `gmud0001` in Munich and also in `uaud0001` in Austin, USA.

As new resources are installed, we have to ensure that the name is chosen correctly. For the validation of hostnames, computer names, and other resources, our suggested system uses the following tables. Part of the design document should contain a similar description of the naming convention:

1. Country codes, where the company has locations or divisions

Table 14. Valid Country Codes for the Naming Convention

Country Code	Full Country Name
a	Australia
g	Germany
u	United States of America

2. Code for locations or divisions in the company

Table 15. Valid Location Codes for the Naming Convention

Location Code	Full Name of the Location or Division
au	Austin
sy	Sydney
mu	Munich

3. Code for machine types in the company

Table 16. Valid Machine Types for the Naming Convention

Machine Type	Description of Machine Type
a	Application Server
d	Database Server
p	Proxy Server
w	Workstation

Note that in the implementation of Tivoli Security Management, we may choose to restrict the naming convention to that determined here through the use of validation policies. Validation policies are discussed in the Tivoli Framework manuals and in the *Tivoli Security Management User's Guide*.

3.2.2 User Names

Before a user can work with a computer on which UNIX or Windows NT is running, they must identify themselves by entering their user or login name in the login facility of the operating system. The identity of a user is normally checked with a password. In order to use the same identification on all systems installed in the company, we have to define a common naming convention for every user that is independent of the division of the company they belongs to.

Although the convention for user names in Windows NT can be up to 20 characters, including letters, numbers, and special characters, other environments, such as most UNIX platforms or host access through a security system, such as the Resource Access Control Facility (RACF) of OS/390, may limit us. We may have to decide that a user name can be a string up to 7 characters, including letters, numbers, and some special characters.

One option is to use numbers only for the user name, possibly preceded by a single letter if systems require it. Many companies use some form of employee serial numbers, and this could be used for the login name. We need no letters or special characters to identify organizational structures, because the user name is not dependent on organizational units. Taking a single letter followed by a six digit number to identify a user company-wide could handle nearly twenty-six million employees!

An alternative strategy is to make some other rule that derives a login name from the person's real name. For example, their first initial followed by the first six characters of their last name - this sort of convention will need to include some contingency for the case where different people would result in the same login name.

As system login procedures change, more and more use can be made of the person's full name, such as in an e-mail address, or to access a system, such as Lotus Notes.

3.2.3 System Group Names

The standard naming convention for user groups is more flexible on Windows NT than UNIX and many other systems. In order to support a variety of

operating systems, one suggested approach is to use the naming convention of UNIX.

In most UNIX systems, the group name can be a string of one to eight characters, including letters, numbers, and some special characters with various excluded characters. In addition, the first character may be restricted, and the name may not contain the key words ALL or default.

In a mixed Windows NT and UNIX environment, we might use only lower case letters for group names. The reason is that in Windows NT the case is ignored, but not in UNIX.

Here is a possible convention for system group names:

1. A similar notation to that of machine names representing the country and department or location.

   The country code is the first classification to group users. Valid country codes for our system are defined in Table 14 on page 51.

   If we want to define groups independent of any country, we could use a reserved character such as `x` instead of a specific country code.

2. Two lower case letters or the combination one lower case letter and one number describe the geography where the machine resides.

   The second classification to group users is the location or division they belong to, for example, based on the name of the city. This is similar to the convention used for machine names listed in Table 15 on page 51.

   Again, if we want to group users independent of any location or division, we use the string `xx` instead of a specific location or division code.

3. Five letters or numbers describe the group of users.

   Of course, it is a strong limitation to use only five characters for the description. If the full description of a group name consisted of several words, for example, Security Administrators, you should create a short name by using the first letters of the words, such as `secad`. The aim is to remain consistent so that sec would always relate to security.

Here are some examples for generating names for user groups:

- Munich Security Administrators are represented by `smusecad`.
- German Security Administrators are represented by `gxxsecad`.
- All Security Administrators are represented by `xxxsecad`.

3.2.3.1 Security Group Names

When we define security groups in Tivoli Security Management, we can choose to stick to system-compatible names, as described in the previous section. Alternatively, as the group names are only used and stored within the Tivoli environment and are not specific to any one system-type, we could adopt a more liberal naming convention. We should still aim for consistency, using the same words or word parts all the time, such as Admin for administrators of any kind, Mgr or something similar for managers, and so on. The next section discusses naming conventions for Tivoli objects in more detail.

3.2.4 Tivoli Objects

Every component that is managed with Tivoli is represented by an object in the Tivoli database. If a Tivoli administrator creates a new object of a supported Tivoli resource class, for example, a managed node or a security profile, the TMR server automatically assigns a new object identifier (OID) to the object. This identifier is a unique number, even if all TMRs of the company are connected with one- or two-way connections.

It would seem, therefore, that we do not need a naming convention for Tivoli objects. Actually, we need a convention, because the administrator normally would not work with an object identifier. The administrator usually works with the resource name of a Tivoli object that is linked to their management desktop. The resource name is the *label attribute* of the resource class the object belongs to. A unique resource name is the basis on which the resources of different TMR's are distinguished.

The standard naming convention for the resource label is very flexible in Tivoli. The string can be longer than any administrator would wish to make it and can include almost any special character excluding the pound sign (#). The names are case sensitive.

We suggest the following guidelines for defining Tivoli resource names, in addition to the basic recommendations described in the introduction of this chapter:

- Using lower case letters for the names only is more practical for administrators.
- Do not use special characters except one proper sign, for example, dot (.) or dash (-) to separate parts of the name.

In order to distinguish resource names used in the graphical user interface and command line, we added some organization information to the normal

description of the function of the resource. Here is our suggestion on how to structure the names of Tivoli resources:

- Two letters (either leading or trailing the name) describe the resource type.

  Every object belongs to a specific resource class. We append a code representing the class to every resource name. Some suggested codes for the resource classes are listed for some security-related resources in the sections following this one. An example might be `pr` for a policy region. This would give the name the form of pr-objectname or objectname-pr, which is instantly recognizable as a policy region. This helps with recognition and use of resource names as it is possible to have multiple resource types using the same name (although we would suggest that a good naming convention should avoid this).

- Three letters describe the scope of the resource.

  This part of the name can be used to show which TMR the resource belongs to. We need this distinction when using different management policies in each TMR. This allows us, for example, to define a system security policy for several countries, where the rest of the resource name is equal:

  - `sp-gxx-system` is the system policy security profile in Germany.
  - `sp-uxx-system` is the system policy security profile in the USA.

  The scope of the resource is built of the codes for the country and the location or division. Possible codes for both are defined in Table 14 on page 51 and Table 15 on page 51.

  If a resource applies for all TMRs, we can use the string `xxx` as scope or omit it altogether - as long as we remain consistent. Similarly, if the resource has been defined for all TMRs of a country, we use the code of the country together with the string `xx`.

- Two letters specify the operating system, if necessary

  If a management resource contains platform specific information, we may have to add the code of the operating system as part of the resource name. This procedure is very helpful for the administrators to choose the correct targets for the distribution of profiles. In addition, this allows us, for example, to define a system security policy for several operating systems, where the rest of the resource name is equal:

  - `sp-xxx-ai-system` is the system policy security profile for AIX that applies to all regions.
  - `sp-xxx-nt-system` is the system policy security profile for Windows NT that applies to all regions.

If the resource is platform independent, we can use the string `xx` instead of the code for a operating system or omit such markings altogether. Again, we must remain consistent - if no marking is used and the resource is platform specific, this could introduce problems.

We suggest the definition of a table for use by administrators. This would contain the codes for all supported platforms. This table can be used to validate resource names or could potentially be used in a validation policy.

- One or more letters describe the function or task of the resource.

  The description about the management function or task of the resource should be as short as possible, while maintaining an obvious meaning for every administrator.

  Avoid the use of superfluous words, for example, `sp-xxx-ai-security system policy for aix`. The information resource type and platform are already included in first and third part of the name.

This naming convention is just an example. You can add additional information to the resource names, for example, the name of the application that is managed with the Tivoli object.

In the following sections, we show an example of how to apply the naming convention to the security relevant resource types of Tivoli.

3.2.4.1 Tivoli Management Region Name

Every TMR server can either be accessed with the region number that is automatically assigned during the installation or with the TMR name. You can think of this name as an alias for the server.

In many cases, it may be useful to incorporate the country code and certainly the location or division code where the server resides. For example, `gmu` for a TMR server in Munich, Germany. Examples of codes are defined in Table 14 on page 51 and Table 15 on page 51.

This approach for determining the TMR name has an advantage when creating local resources. The command `wtmrname` returns the name of the TMR and can be used in a shell script when assigning a resource name to the Tivoli object. The following statement is an example of how to create a new Security Profile from a standard profile, where the TMR name is incorporated into the name of the new security profile - note that this script can be run on any TMR without modification, and it will correctly use the name of the local TMR:

```
wcrtprf -c @SecurityProfile:sp-xxx-xx-standard \
   @ProfileManager:pm-‘wtmrname‘-xx-system \
```

```
SecurityProfile sp-‘wtmrname‘-xx-system
```

3.2.4.2 Tivoli Management Framework Naming

The naming convention can not be applied for every Tivoli resource. There are some types that use the same fixed name from the TMR server. The following resources have names that are equal in every management region:

- The Desktop resource type (TME Desktop).
- Administrators is the name of the type AdministratorCollection.
- Notices is the name of the type BulletinBoard.
- Scheduler is the name of the type Scheduler.

Because we want to add the type in the resource name, we have to define a code for the framework resources. Table 17 can also be used to validate the names of resources:

Table 17. Framework Resource Naming Codes

Resource Class	Code	Example
PolicyRegion	pr	pr-gmu-xx-users
PolicySubregion	pr	pr-gmu-xx-users development
ProfileManager	pm	pm-gmu-xx-users development
TaskLibrary	tl	tl-xxx-ai-users
Task	ta	ta-xxx-ai-users check security
Job	jo	jo-gxx-ai-users check security

There are several resource types for which we do not suggest applying the naming convention for Tivoli objects:

- Administrator

  It is more convenient to use the full user name for the resource type Administrator. One possible exception may be a situation where multiple administrators use the same Tivoli administrator ID. For example, all those charged with adding users may use a name, such as `ad-gmu-xx-adduser`. Shared administrator IDs are discouraged in Tivoli for accountability reasons.

- GenericCollection

  A generic collection is normally used by administrators to group resources on their own desktop. That name is only relevant for the owner of the generic collection, the administrator.

- Endpoints

  For these resources, we have already defined a naming convention in this chapter. It makes sense to assign the TCP/IP hostname or NetBIOS computer name to the Tivoli resource name.

3.2.4.3 Tivoli Distributed Monitoring

Table 18 contains suggested codes for some resource types from the Distributed Monitoring application to be used as part of the naming convention for Tivoli objects.

Table 18. Distributed Monitoring Resource Naming Codes

Resource Class	Code	Example
SentryProfile	mo	mo-xxx-ai-system disk space
IndicatorCollection	ic	ic-gmu-ai-system disk space

3.2.4.4 Adapter Configuration Facility

An Adapter Configuration Profile (ACP) can contain records of configurations for the TEC Logfile Adapter and the TEC Windows NT Event Log Adapter.

Table 19 contains the suggested code of the resource type ACP to be used as part of the naming convention for Tivoli objects.

Table 19. Logfile Adapter Resource Naming Codes

Resource Class	Code	Example
ACP	ac	ac-xxx-ai-tivoli database logfile

3.2.4.5 Tivoli Enterprise Console

Table 20 contains the codes of several resource types from the TEC to be used as part of the naming convention for Tivoli objects.

In this case, it makes no sense to include the information about the platform in the naming convention because event processing is platform independent. Therefore, we can use only the parts of the naming convention defined in this chapter. These parts are the code, the scope, and the description of the resource.

Table 20. TEC Resource Naming Codes

Resource Class	Code	Example
Rule Base	rb	rb-xxx-security
Event Group	eg	eg-xxx-security

Note that for the names of event classes, we can not apply the naming convention. These class names have to be globally defined for the whole company and need no organizational classification. However, we suggest using the name of the application the class belongs to as a prefix for the name.

3.2.4.6 Tivoli User Administration

Table 21 contains the codes of resource types from Tivoli User Administration to be used as part of the naming convention for Tivoli objects.

Table 21. User Administration Resource Naming Codes

Resource Class	Code	Example
UserProfile	up	up-gmu-ai-development
GroupProfile	gp	gp-gmu-ai-development

3.2.4.7 Tivoli Security Management

The following table contains the code of the resource type Security Profile to be used as part of the naming convention for Tivoli objects.

Table 22. Security Management Resource Naming Codes

Resource Class	Code	Example
SecurityProfile	sp	sp-xxx-ai-system

For the definition of roles, we can not apply the naming convention of Tivoli objects described in the beginning of this chapter. We do not need the code for the resource class and a specific type of platform.

Here are details of a suggested naming convention for role names:

- One lower case letter representing the country.

  The country code is the first classification to roles. Valid country codes are defined in Table 14 on page 51.

  If a role is independent of any country, we can use a wild-card character, such as x instead of a specific country code.

- Two lower case letters or the combination one lower case letter and one number describe the geography where the role applies.

  Valid location or division codes are defined in Table 15 on page 51.

  If the role is independent of any location or division, we use the string xx instead of a specific location or division code.

- Several lower case letters or numbers describe the membership of the role to a organization, department, or project.

  The resources defined to a role are related to an organizational element, usually a job function. If we add the name of an organization to the role name, it may help us to assign roles to security groups. For example, you can express that relationship with a project or department number.

- One or more letters describe the function.

  The description about the role should be as short as possible, while the meaning must be obvious for every administrator.

For example, the role name `uau-sg24nnnn-print redbook` contains the necessary access rights to a printer and directory of a host in Austin, USA in order to print redbooks of the form SG24nnnn.

3.3 Organizing the Tivoli Environment

In order to manage the security of target machines, we have to define a set of profiles in Tivoli and distribute these to the machines. Theoretically, this can be achieved by creating a policy region that contains all managed nodes and a single profile manager with security profiles and subscribers. But this approach becomes unmanageable when handling the thousands of profiles and subscribers that may be needed for a big implementation. Therefore, we want to define a hierarchical structure consisting of policy regions, sub-regions, and profile managers that are able to arrange the profiles in a logical way. This structure may also be required to restrict the access of administrators managing particular applications.

Of course, you can use more policy regions and profile managers to classify the subscribers in a more granular fashion.

3.3.1 Using Policy Regions

The use of policy regions will depend on the administration policy. For example, if security administrators are not the same as distribution administrators, then in each TMR, you may specify a different policy region or subregion to contain profiles for each Tivoli application. Then, each region can restrict access only to their correct administrators, as illustrated in Figure 11.

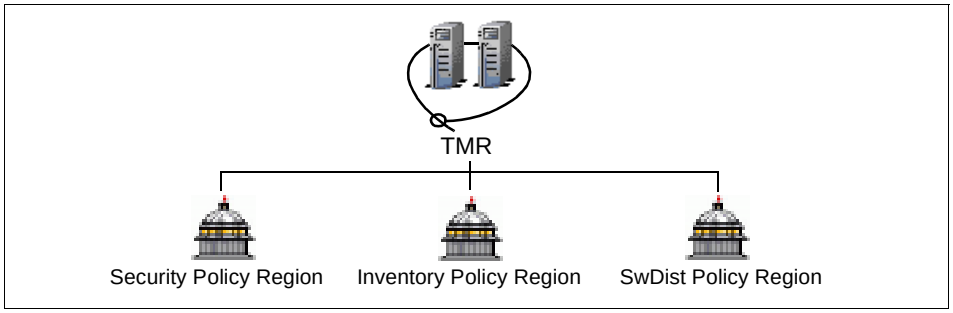


Figure 11. Policy Region Separation by Tivoli Application

It may be necessary to apply a similar technique to the grouping of system-types. For example, you may wish to have different administrators managing different platform types, as illustrated in Figure 12.

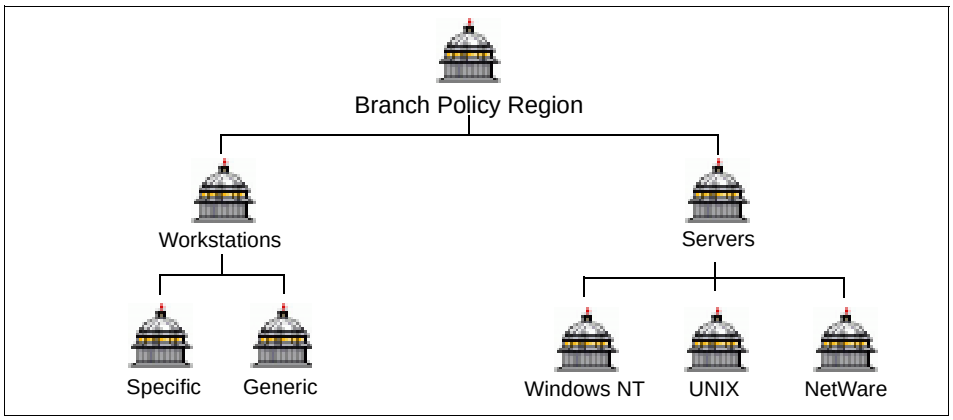


Figure 12. Policy Region Separation by Platform Type

The level of division depends on the required separation of administration and the manageability based on the likely number of components in each region.

3.3.2 Using Profile Managers

The figure below shows an example of the profile managers for the application, Lotus Notes. Because the profiles for security, sentry, and adapter configuration can be different for different operating systems, such as AIX and Windows NT, we will define a profile manager for each.

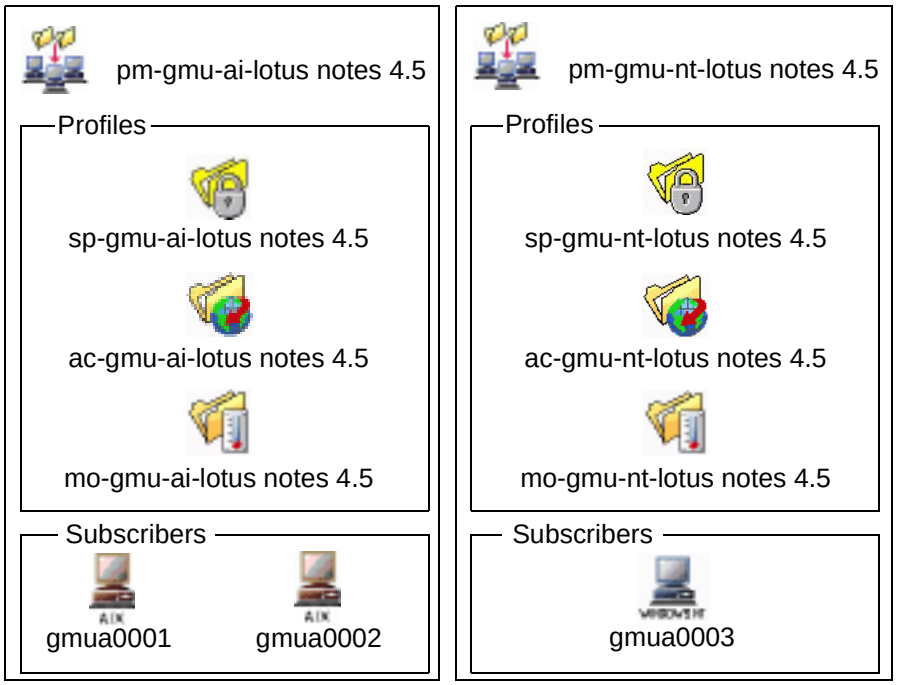


Figure 13. Platform-Specific Application Profile Manager Example

3.3.3 Configuring Multiple Management Regions

We described in the last section how to create profiles in a TMR to manage applications. In a configuration of multiple distinct TMRs, these profiles can only be distributed to the nodes local to the TMR server. But what if we want to be able to define profiles applicable to a range of TMRs? For example, we may wish to specify the security system policy for logins or the password for every Lotus Notes server (assuming every Lotus Notes server has the same security configuration).

The Tivoli Management Environment allows for an administrator in one TMR to control the Tivoli resources of another TMR. This allows for the formation of a form of hierarchy, although as there is no pass through of management from one TMR to another, a TMR has to be directly connected to any other TMR in order for cross-TMR management to take place. This is illustrated in Figure 14.

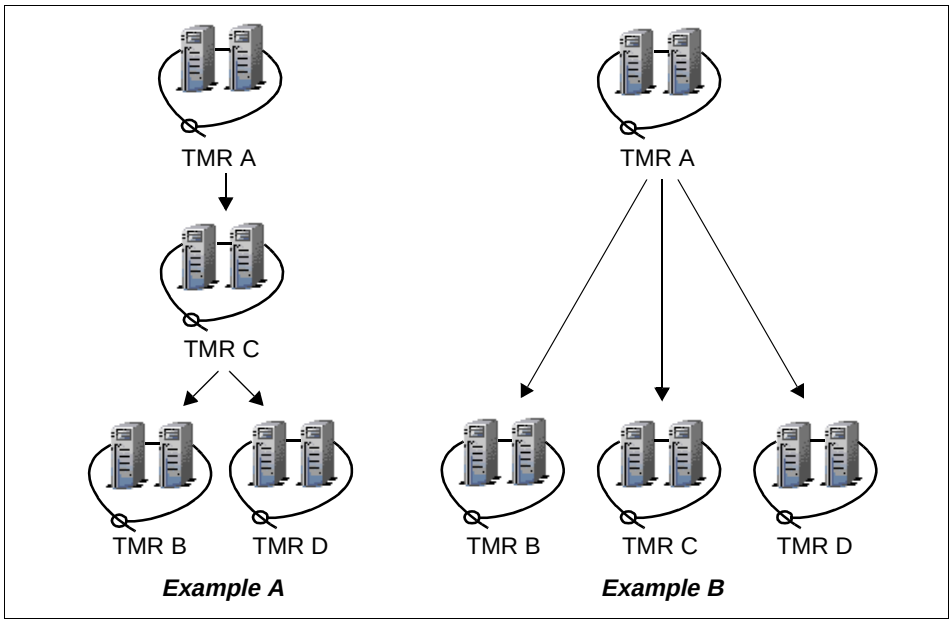


Figure 14. TMR Management Hierarchy

Example A, in the above figure, shows an invalid TMR management structure. TMR A could be configured to manage TMR C, and TMR C can be configured to manage TMRs B and D, but in this setup, an administrator at TMR A has no control over TMRs B and D. Example B shows a valid single-level hierarchy, where an administrator at TMR A could control resources in TMRs B, C, and D. You might also picture a single-level hierarchy as a hub and spokes configuration, where one system acts as a hub (TMR A in Example B), and the others are all spokes fanning out from the hub. The administrator only needs to connect to the hub to manage all the systems in the wheel.

In order to define profiles in one TMR to be distributed to many others, one or more top-level TMRs are connected to the TMRs in the locations with one- or two-way connections. Refer to the *Tivoli Framework User's Guide* for how to establish connections between TMR's.

We will review this with a more specific example.

In order to manage the TMRs in all locations in Germany, we install a new TMR server. We apply the naming convention described in section 3.2.4, "Tivoli Objects" on page 54 and assign the name gxx for that TMR. After this, a connection needs to be established to each of the TMRs in the company's German locations.

Because we want to manage the security of the Lotus Notes server in Germany, we create the policy region `pr-gxx-xx-lotus notes 4.5` for this application in the top-level TMR. The function of that policy region is comparable with all policy regions created for the same application in the locations. The new policy region includes the profile manager `pm-gxx-ai-lotus notes 4.5` for the Lotus Notes application.

The following figure shows the distribution of the security profile `se-gxx-ai-lotus notes 4.5` to the subscriber `pm-gmu-ai-lotus notes 4.5`. After the distribution, the profile manager in Munich received a copy of the security profile, which can be distributed further to the application server. Whether the distribution goes all the way to the application server depends on the options chosen during a distribution. If the policy requires local administrators, the distribution will occur to the next level of subscribers, and the local administrator could then be responsible for implementing the profile locally.

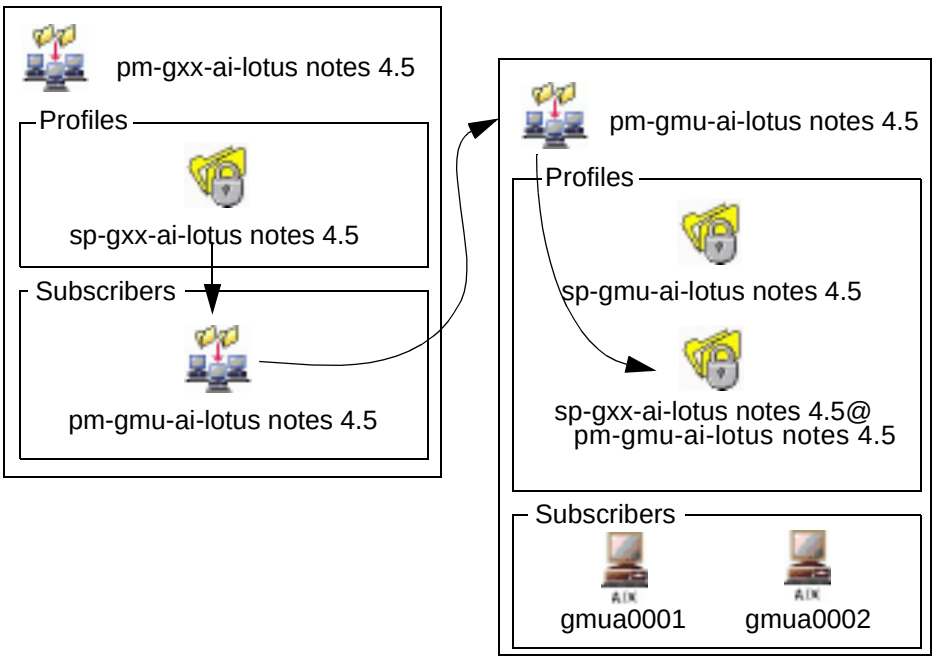


Figure 15. Hierarchical Application Profile Manager Example

The above example shows how security profiles can be defined at a singe point of control for hundreds of target machines. Of course, the TMR servers are suitable for providing many other management functions. For most Tivoli Security Management implementations, we will also need to install Tivoli Enterprise Console in one or more TMRs that represent a focal management

point in order to receive event messages from every Logfile Adapter and Sentry Monitor.

3.4 Groups and Roles

As explained in “Groups and Roles” on page 32, a Tivoli Security Management implementation requires a complete rethinking of the way groups are used. You should re-read those sections and be sure to understand the way security groups and roles determine access rights. In this discussion, we are talking about security roles as a way of managing user access to resources, rather than the Tivoli roles that are assigned to administrators.

With security groups and roles, we have an extra tier in the way we determine which users get access to which resources. In Tivoli Security Management, users belong to security groups that align more with their job title in an organization. That group then has roles assigned to it which determine the access rights the members of the group have over resources.

In 2.3.5, “Groups and Roles” on page 32, you should have determined the roles people perform within the organization, and the resource accesses required for those people to fulfill those roles. Now, you need to use security groups to bring together lists of users who perform similar functions so that they can all be given the same set of roles.

In order to reduce future administration, it is preferable to make users members of as few security groups as possible. Membership of the group gives them access to all the resources they need to perform their job. If they only belong to one group, an employee changing jobs just needs to be moved from one security group to another to make sure all the old access rights are removed, and all new ones are created. One action of moving a user from one group to another takes care of all the required modifications on all the systems and system types that have resource accesses defined in that role. You should expect to have very large number of resources defined, as access will be refined in a smaller number of roles that group those resources together, and users will be collected in an even smaller number of groups.

The groups will contain users based on an organization chart, and may need to be quite specific in order to ensure all the members only have the access rights they need. Examples might include Zingo Development-Permanent Staff, Zingo Development-Contractors, Zango Development-Permanent Staff, Development Managers, and so on.

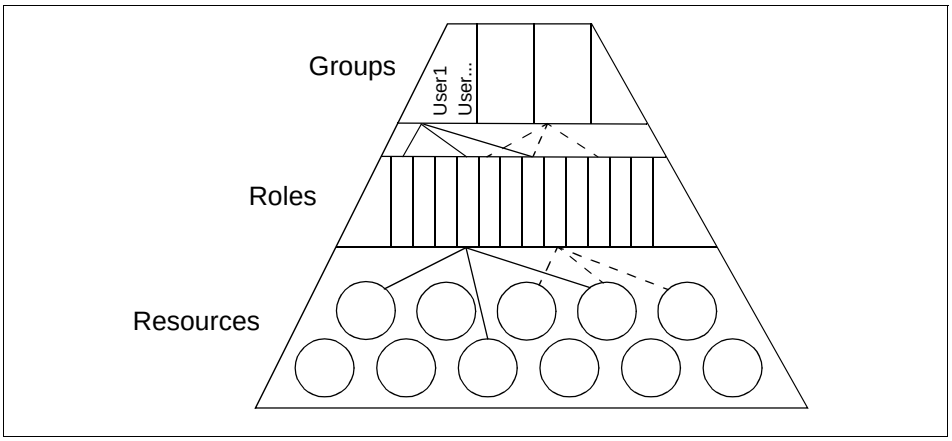


Figure 16. Group, Role and Resource Hierarchy

Figure 16 gives some idea about how the management is performed through the tiers of security groups, roles, and resources. Note that different access to the same resources can be defined in multiple roles, and that roles can be assigned to multiple groups. One other consideration not shown in the diagram is that roles can inherit their definition from other roles. You may have the access to 20 resources from six systems defined in a role called Source Management and want similar, but slightly different, access rights for the role Source Backup. One role can inherit its properties from the other and then modify them to provide the required access controls.

Those familiar with the Windows NT security model will be most familiar with this groups and roles concept. A security role maps to a Window NT local group, and a security group maps to a Windows NT global group. By being a member of the correct global group, you have access to all the correct Windows NT resources by virtue of your global group being a member of the right local groups on the resource servers - and those local groups define the access to the resources. With Tivoli Security Management, however, there is no reliance on trusted and trusting domains to implement a security design across multiple domains - and the same design will simultaneously manage resources for the same users on other platforms.

Chapter 4. Resource Security Design

This chapter continues the discussion on architecture and design by looking at the actual resources we are out to protect. Starting at the system level with policies that apply to whole machines, we then move on through the various individual resource types, such as files and network connectivity.

4.1 System Policies

By system policies, we mean a policy that is applies to the whole target operating system, not something that would be specific to individual groups or resources. That means that every user of the system has to follow the policy or rules. In this section, we will discuss the way Tivoli Security Management enables the management of password and login settings that can be used for the different operating systems.

4.1.1 Password Policy

Passwords grant the access to computer systems. Therefore, tightening password policy is a key action, where the result is the improvement of the security of a computer environment. Stories sometimes make the headlines about intruders accessing network resources of certain companies. This is usually not because these intruders are expert hackers. They use the *open doors,* such as blank or inadequate passwords. In fact, most intruders are successful because system administrators have been careless in setting security policies - either due to maintenance difficulties, or perhaps, an ignorance of platform-specific security exposures.

Providing a password policy helps to assure that these open doors are closed. With the password policy, you control the user's behavior concerning passwords.

The password policies are split up into the following topics:

- Maximum days between password changes
- Minimum days between password changes
- Minimum password length
- Password history depth
- Allowable characters (not available on Windows NT)

We will define, discuss, and offer a suggestion for each of these.

4.1.1.1 Maximum Days between Password Changes

This setting defines how long passwords can be used before they expire. So, this is the lifetime of a password. Users whose passwords are older than the time allowed are forced to change their passwords the next time they log in. This is often referred to as *password aging*.

With this feature, you can improve the security so that even if an unauthorized person knows someone else's password, you can at least restrict the maximum time they are able to login as that user before the original user must change the password. Using this option reduces the chances of their being multiple users accessing the resources under the same account (so long as the account owner does not spread the password after being forced to change it).

The older a password gets, the more likely it is that it could be disclosed to an undesirable entity. Password security decreases with age.

So, wouldn't it be a nice idea to force the users to change their password as often as possible, for example every 2 days? This is not likely to help. Changing the passwords too often could result in users having difficulty remembering the current one. So, there is a greater tendency to write down the password, which is, of course, very insecure. Therefore, password security is also lower with very short lifetimes. This is represented graphically in Figure 17.

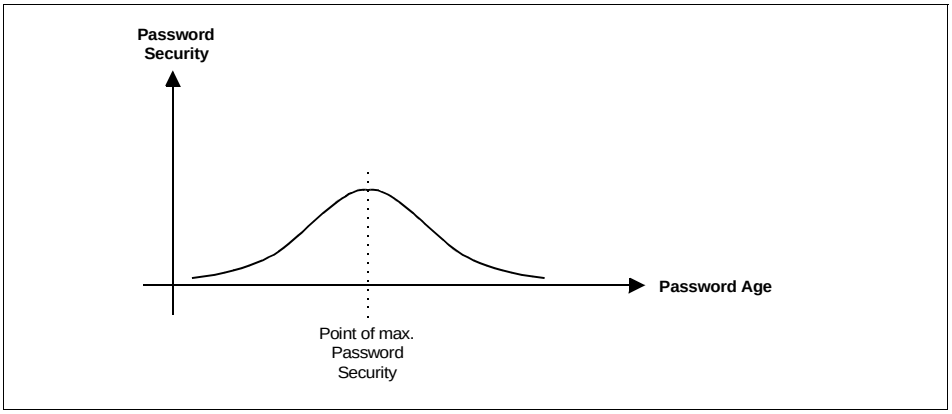


Figure 17. Relationship Between Password Age and Security

The point of maximum password security also depends on how sensitive the data in the environment is. For non-sensitive environments, the typical suggestion is 30-60 days for the maximum password age. For more sensitive, environments changes obviously should occur more often.

Your work with the system policy in 2.3.4, “System Policies” on page 31, should result in a figure for the number of days. Note that not all platforms will necessarily implement every password policy attribute. TACF on UNIX and Windows NT both allow the specification of a maximum password age.

4.1.1.2 Minimum Days between Password Changes

This setting defines the minimum amount of time between password changes. This is the minimum lifetime of a password. Users are not allowed to change their password more frequently than this setting allows.

It may be that if a user is forced to change a password after some maximum time, they may try to change it to a temporary one and then change it back to the original. Specifying a minimum time for which a user must keep a new password is one way of restricting this. Using a password history and preventing users from using a previously used password is another (see 4.1.1.3, “Password History Depth”). The minimum days between password changes alone will not prevent them from cycling between two favorite passwords. If this is to be implemented, but has not been specified in a security policy, we recommend a minimum password age of seven days.

4.1.1.3 Password History Depth

This setting defines the number of passwords that must be used before a user can reuse an old password. If a user is forced to change their password at login, and they enter a combination that is still stored in the password history, they will be asked to choose another password.

Without establishing a password history, users can cycle between two passwords. If the history is too small, a user can enter dummy passwords until they can reuse their favorite old password.

When setting the password history depth, you may also consider the minimum time after which a password can be reused (minimum password reuse time or password aging - see 4.1.1.2, “Minimum Days between Password Changes”). The following equation shows how the minimum reuse time can be calculated:

```
min. password age x password history depth = min. password reuse time
```

If we have a minimum password age of seven days and a history depth of 15 passwords, a user can reuse a password at the earliest after 105 days. If no password aging is specified, the user could just change the password the number of times required to get past the history depth and then revert to the original password.

If the security policy does not determine the password history depth, we recommend 10 passwords or more.

The maximum history depth can depend on the platform. For Windows NT and TACF, it is 24 password entries.

4.1.1.4 Minimum Password Length

This setting defines the minimum number of characters a password must contain. Password changes will be rejected unless the provided string has at least the minimum password length.

Increasing the minimum password length brings more security to your systems, because it is harder to guess or otherwise find out longer passwords than shorter ones. But longer passwords are harder to remember. As a result, users may tend to write their passwords down, which is less secure than a shorter password.

Note

As administrator in the Windows NT operating system or as root on AIX, you can override the minimum password length. You can set up new users with a password with fewer characters than the minimum password length or even a blank password. When the user is forced to change their password, the minimum password length restriction is applied.

If the security policy does not dictate minimum password lengths but they are to be used, we recommend a minimum password length of around eight characters for all systems.

4.1.1.5 Restricting Password Composition

You can improve the password security by restricting the composition of passwords. That means that you apply rules for the characters that are included in a password. If a user tries or is forced to change their password, the new password is not accepted unless it obeys the set rules. The following descriptions and recommendations are based on rules available within TACF on UNIX systems and may differ for other platforms, as platform support extends beyond UNIX and Windows NT. Windows NT support does not currently include password composition rules. Note that the policy for the composition of the password on UNIX is currently enforced during password changes only when using the `sepass` command or the TACF command line commands, such as the `chusr` utility. If `wpasswd` or `passwd` are used, the password policy checks are not performed.

A password's structure can be ruled with the following settings:

Minimum Alphabetic The minimum number of alphabetic characters (A-Z or a-z) a password must contain. Our recommendation is 1. Note that the overall password length may still be required to be higher.

Minimum Numeric The minimum number of digits (0-9) a password must contain. Our recommendation is 1.

Minimum Alphanumeric The minimum number of alphanumeric characters (A-Z, a-z, or 0-9) a password must contain. Our recommendation is 1. Using the previous recommendations, a value of one does not change the password policy.

Minimum uppercase The minimum number of uppercase characters (A-Z) a password must contain. Our recommendation is 0.

Minimum Lowercase The minimum number of lowercase characters (a-z) a password must contain. Our recommendation is 0.

Maximum Repeated The maximum number of times a character can appear consecutively within a password. Our recommendation is 3.

Minimum Special The minimum number of special characters (any non-alphanumeric) a password must contain. Our recommendation is 0 for normal environments or 1 for more sensitive installations.

Summarizing the recommendations for constraining passwords leads to the following rules.

A password must have:

- At least one numeric character (0-9)
- At least one alphabetic character (A-Z or a-z)
- Not more than three consecutively repeated characters
- At least one special character (non-alphanumeric) in especially sensitive environments

Assuming a minimum password length of eight characters and applying the above recommendations, valid passwords would be:

- Tecis4me

- omsi:luli1 (assuming special character usage)
- luv-tme10

4.1.2 Login Policy

Tivoli Security Management specifies login policy through the system policy record in a security profile.

4.1.2.1 Account Lockout

Any password can be guessed, if you give an intruder enough chances to try different passwords. The *account lockout* feature gives you the ability to prevent someone from using a trial-and-error method to break into the systems.

It lets the systems lock out an account for a period of time if a specified number of failed login attempts are encountered.

If the network is accessible from outside the organization through a WAN or dial-up connection, it is especially important to provide a restrictive account lockout policy.

You will have to set the number of failed login attempts before the user accounts are locked. If a value is not derived from a security policy, we recommend three failed logins. It is not recommended to choose a value of less than three. A user must be permitted a couple of chances to reenter a password due to typing errors. You would reduce the productivity of the users and administrators with a very low setting.

Note

Solaris operating systems provide a maximum number of failed login attempts of five. You can still choose higher values for all systems. Tivoli Security Management will set this value on all operating systems but Solaris. Here the value five is set.

SunOS does not support user account lockouts at all.

If an intruder tried to guess a password, and the account they were using locked up, how long should this account be locked? Too short a time could allow the intruder more guesses. Too long a time will probably prevent the intruder from trying, but a user is not able to work for this long lockout duration.

The lockout feature can prevent unauthorized use, but it can also be used for attacks of other kinds. It could be used to maliciously lock out accounts. Intruders can use it to lock out selected users, perhaps to prevent discovery of their actions for the time the user is locked out.

Note

The root user on UNIX systems and the Administrator user on Windows NT (if not renamed or assigned to another user) can not be locked out.

In Windows NT, it is possible to lockout an account forever, that means until the administrator unlocks it. We do not recommend this for previously described reasons. Our suggestion for the lockout duration is 60 minutes. That should meet the needs of most environments.

There is also a Time Span setting for lockouts. For TACF, Time Span determines the time within which the maximum number of failed logins must occur for the lockout to happen. For example, setting this to 10 minutes means with a maximum number of failed login attempts set to five, that if a login fails, and it is the fifth failed attempt within the last ten minutes, the lockout will occur. For Windows NT, this is slightly different: this parameter specifies how long after the last failed attempt to reset the failed count to zero. If this is set to 10 minutes for Windows NT, you could have four failed attempts five minutes apart and the fifth attempt - if still within 10 minutes of attempt four - will lockout the user. For TACF, Time Span is the whole period for counting failed logins, and for Windows NT, Time Span is how long between each login to wait before resetting the count.

For UNIX, a typical setting might be 60 minutes. For Windows NT, you might choose something like 30 minutes. The larger the value, the fewer chances a hacker has to try different passwords each day. However, the larger the value, the greater chance there is of a genuine user mis-typing passwords five times (or whatever) within the resultant time period.

4.2 Data Resources

The data of a company is one of the most important things to protect. It contains vital assets, such as balance sheets, statistics, development, and production information. You must protect data against deletion by unauthorized persons and modification by someone who is not supposed to change a file. There is also the need to ensure that the information stored in your network can only be viewed by users that are supposed to have access. This is not only assured through proper access rights. Full availability means

including backup and archives of the data on a regular basis. Backup policy is not a part of Tivoli Security Management and is not covered here.

This section concentrates on Windows NT and UNIX data types. The topics covered will demonstrate the sorts of considerations that should be applied to other platforms. Windows NT and UNIX are treated separately here, as there are some differences. Whether you choose to manage Windows NT and UNIX resources separately will depend on the complexity of the environment and the administrative policy. One of the great advantages of management through Tivoli Security Management is that multiple resource types from different platforms can be managed centrally through the same records. It is possible to keep resources defined in different profile managers in platform-specific policy regions but still assign them to the same role if your administrative requirements dictated such a division.

Tivoli Security Management specifies data resources through the resource record in a security profile.

Note that in this section, we are including Printer and Registry resources as Windows NT Data resources, as they are managed in a very similar fashion to files and directories.

4.2.1 Windows NT Data Resources

Windows NT employs a group-based file and directory security with NTFS. In this publication, we strongly recommend and assume that NTFS is being used with the Windows NT operating system. Comparing it to the FAT file system it brings some additional features, such as local folder and file level permissions, ownership, and auditing.

There are many NTFS-specific facts you should know when designing your data security.

Every directory and file has an owner who has a special status. By default, the owner of that resource can manage the permissions like an administrator. The owner is even allowed to prevent administrators from accessing the owner’s files and directories. This enables users to configure their personal directories for complete privacy. However, administrators can take ownership of any file or directory on the network. This is because the administrator must be able to access the problem files or deal with files whose owner is no longer in the company. However, the administrator can not return the ownership to the original owner. So taking this action will leave some evidence behind.

Note

Up to and including version 3.6, Tivoli Security Management does not provide any facility to centrally alter the way Windows NT treats the permissions of file owners (specified through the Windows NT group CREATOR OWNER). If a modification is desirable, native Windows NT tools should be used to modify the CREATOR OWNER group.

Now that we have the data resource list from 2.3.6, “Data Resources” on page 38, we can check if the access permissions that are currently applied are suitable for the resources. There will be some resources that do not have proper access restrictions or resources that will be created soon. For that reason, we will be designing the access permissions to Windows NT files, directories, and shares, now.

The Tivoli security model with groups and roles maps closely to the one that is implemented in the Windows NT operating system. The Tivoli groups correspond to the Windows NT global groups, and the Tivoli roles correspond to the Windows NT local groups.

It is a common recommendation that you should not assign any resources to a user directly (with a few exceptions like the user directory), and that recommendation is usually enforced in Tivoli Security Management. Resources are accessed through the role (NT local group), and users have access using roles through the groups (security groups map to global groups, which become members of local groups in Windows NT). Often you will find NT global groups assigned directly to resources. This does not fit in to the role-based security model either.

Therefore, what we need to do is to take the tables we filled out in 2.3.6, “Data Resources” on page 38, go through all resources, and perform the following. Note that these actions are taken on the paper, not on the resource itself.

1. **No direct user assignments.**

   Verify that there is no user assigned to a resource directly. If you find a user in the access control list (ACL), delete the user from the Access row of the resource table. Users should obtain the correct access through their security group memberships. The groups gain the correct access to resources through the security roles they are assigned.

2. **No direct NT global group assignments.**

Verify that there is no NT global group assigned to a resource directly. If you find a NT global group in the ACL, delete it from the Access row of the resource table. As with users, global group accesses should be covered through user membership of security groups.

3. **Verify the role assignments.**

   Check that every resource can be accessed by one or more appropriate roles with the appropriate access permissions.

4. **Ensure administrative access.**

   Ensure that there is an access control entry (ACE) for the NT global group Domain Admins with Full Control. (Assuming some other administrative group has not been set up.)

5. **Make backups possible.**

   Ensure that there is an ACE for the backup role. Even if you are granting default access (NT global group Everyone) to a resource, this is still a good thing to do. That is because you do not have to worry about the backups if the default access permissions need to change. And again, if you think the roles that you defined are not granular enough, add the required ones to your list. For example, you could find out here that it would better meet the company's requirements to have a backup role for every type or location of data (for example, backup correspondence or backup development data).

6. **Do not touch SYSTEM group entries whenever possible.**

   If you have an ACE for the NT AUTHORITY group SYSTEM, you should seldom, if ever, need to manipulate the permissions. Remember that critical resources such as the Tivoli Object Dispatcher service are running under the SYSTEM account too.

This is the last time we go through the access permissions of the resources. *Now* is the time to decide who has which access to a resource. So, define it now and write it down. The design document you produce is very important, because the person implementing it will normally not think about the access permissions they find in the design document. Even if you will implement the security yourself, you will not necessarily be thinking about all the access permissions again.

This work can be simplified by using abbreviations for the permissions. Table 23 lists the generic abbreviations used in Tivoli Security Management and the equivalent in Windows NT.

Table 23. Tivoli Security Management Abbreviations for NT Permissions

Permission	Tivoli Abbreviation (CLI)	NT Equivalent
Read	R	R
Write	W	W
Execute	X	X
Delete	D	D
Update	U	RWXD
Full Control	F	All
Change Ownership	O	O
Change Permissions	P	P
No Access	N	No Access

The following sections provide specific information on a range of Windows NT resources, including files, directories, shares, printers, Windows NT system files, and the registry.

4.2.1.1 Windows NT File Resource Considerations

Using Tivoli Security Management, you can assign access permissions to individual files. In order to secure a file, you must have the full path, including the drive letter. We suggest limiting the use of security applied directly to files with the security FILE resource only for important single files for the following reasons:

- It is too much work to administer every file separately. You would have to define one security profile record of the type FILE for each file.
- When you define access permissions directly for a single file, they will be set when the Tivoli profile is distributed to the Managed Node. If you locally set the file permissions of the directory in which the file resides, the previous set of file access permissions can be overridden, because Windows NT provides an option to apply new directory permissions to all files included in the directory.

Therefore, if you are setting access permissions on single files, you have to make sure that the default permissions of the directory in which the files reside are not modified locally on the Windows NT system, as they could potentially be applied to all the files in the directory, overwriting permissions set on individual files.

Note

As of the 3.6 release, you can not use wild-cards to apply access restrictions to a group of existing files, such as `D:\Correspondence\1998\*.lwp`, with Tivoli Security Management. You can only specify the default access for newly created files through the DIRECTORY resource type (See the next section). This function could be handled through Tivoli via scripts or tasks using native Windows NT commands, such as `cacls.exe`.

Table 24 shows the capabilities a user has when granted Tivoli Security Management permissions on files. For example, someone granted Execute (X) permissions for a file will be able to Display the file's owner and permissions and run the file if it is a program.

Table 24. User Capability on NT Files by Access Permission

Capability Enabled by Security Permission	R	W	X	D	U	F	O	P	N
Display the file's owner and permissions	✓	✓	✓		✓	✓			
Display the file's data	✓				✓	✓			
Display the file's attributes	✓		✓		✓	✓			
Change the file's attributes		✓			✓	✓			
Change data in and append data to the file		✓			✓	✓			
Run the file if it is a program			✓		✓	✓			
Delete the file				✓	✓	✓			
Change the file's permissions						✓		✓	
Take ownership of the file						✓	✓		

The Windows NT file resource record can be specified from the command line as endpoint and resource type: NT:FILE.

4.2.1.2 Windows NT Directory Resource Considerations

Using Tivoli Security Management, you can assign access permissions to individual directories. Controlling access to Windows NT data resources by setting the directory permissions, in contrast to setting permissions to single files, is the recommended approach as it simplifies administration.

Windows NT stores two kinds of permissions for a directory:

Directory Permissions Permissions to the directory itself.

Default Creation Mask These are the default permissions for files created in the directory. If you create a new file in or copy a file into the directory, these access permissions are set.

Figure 18 shows how these two access types are displayed in the Windows NT directory permissions dialog.

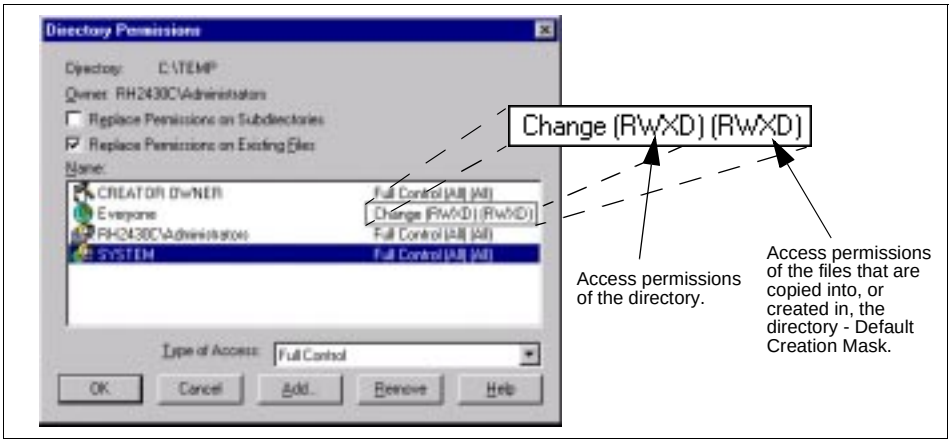


Figure 18. Windows NT Directory and Contained Files Access Rights

Table 25 shows the capabilities a user has when granted Tivoli Security Management permissions on directories.

Table 25. User Capability on NT Directories by Access Permission

Capability Enabled by Security Permission	R	W	X	D	U	F	O	P	N
Display filenames of the directory	✓				✓	✓			
Display the directory's attributes	✓		✓		✓	✓			
Add files and subdirectories		✓			✓	✓			
Change directory's attributes		✓			✓	✓			
Go to the directory's subdirectories			✓		✓	✓			
Display directory's owner and permissions	✓	✓	✓		✓	✓			
Delete the directory				✓	✓	✓			
Change the directory's permissions						✓		✓	
Take ownership of the directory						✓	✓		

The mapping of Tivoli Security Management access permissions to Windows NT permissions are listed in Table 24 on page 78.

Note

As of the 3.6 release of Tivoli Security Management, the DIRECTORY resource does *not* provide the option of applying directory permissions to the permissions of files already in the directory. It sets permissions for the directory itself, and the FILE resource sets the default creation mask of the directory. Applying directory permissions to existing files would have to be done through normal Windows NT administration.

The Windows NT directory resource record can be specified from the command line as endpoint and resource type: NT:DIRECTORY.

4.2.1.3 Windows NT Share Resource Considerations

Shares are the essential part of Windows NT that make resources accessible over the network. Every share has its own permissions set just like a file or a directory. The Tivoli software enables you to set these access permissions. When choosing the permissions for a share, you have to consider that the permissions associated with the files and directories that can be accessed through the share are still active. For example, you could provide a share with permissions granting everyone full control. However, the directory and file permissions may still restrict the access to the information.

That means that the access rights to a resource through a share are made up of the share's ACL and the NTFS's ACL. The user can access a network resource only to the extent that both sets of permissions agree. Figure 19 shows this relationship in the form of an example.

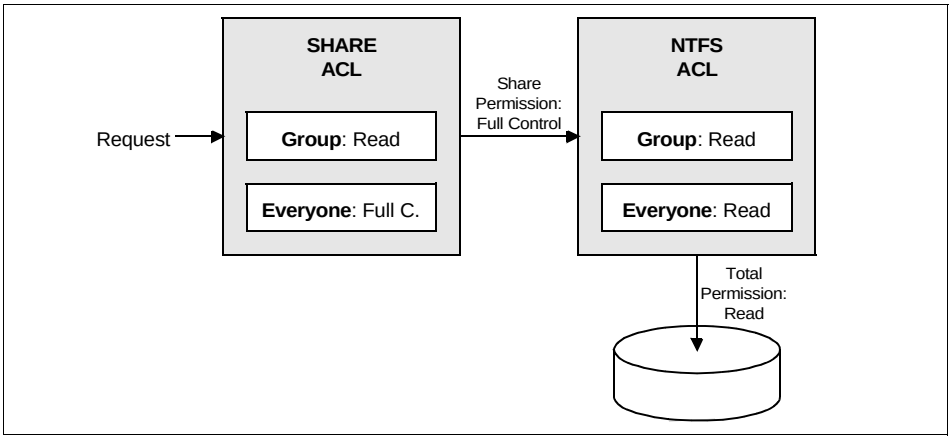


Figure 19. Windows NT Share versus NTFS Permissions

Permissions that are set to a share apply to the directory, its files, and subdirectories. If you want to have a different access type to the subdirectories or files, you have to apply NTFS restrictions to the appropriate resources. For subdirectories, you can also set up a new share with the appropriate access rights.

Table 26 shows the capabilities a user has when granted Tivoli Security Management permissions on shares.

Table 26. User Capability on NT Shares by Access Permission

Capability Enabled by Security Permission	Read R	Update U	Full C. F	No Acc. N
Display subdirectory and filenames	✓	✓	✓	
Display file data and attributes	✓	✓	✓	
Run program files	✓	✓	✓	
Go to the directory's subdirectories	✓	✓	✓	
Create subdirectories and add files		✓	✓	
Change and append file data		✓	✓	
Change file attributes		✓	✓	
Delete subdirectories and files		✓	✓	
Change permissions			✓	
Take ownership			✓	

When setting the access permissions to shares, we recommend having the table defining the file or directory resources from Chapter 2, "Analysis of the Target Environment" on page 9, on hand. Compare the permissions and check the access of a representative sample of users. This will help you to find out if your permissions are appropriate before you go into production.

Note

As of version 3.6 of Tivoli Security Management, you can administer shares but not create them. Creating a share must be done with Windows NT tools.

The Windows NT share resource record can be specified from the command line as endpoint and resource type: NT:SHARE

4.2.1.4 Windows NT Printer Resource Considerations

In Windows NT, you can restrict the access to the attached printers. Normally, you would think that all users should be able to print on a certain printer. But

there are some good reasons to restrict the access. If you have staff that print letters and editors printing books, it is good to have two kinds printers, one for big print jobs and one for the quick letters. Therefore, staff printing letters do not need to have access to the high performance book printer, and editors do not block a printer intended for letters by printing books.

Color laser printers are also often protected. As the supplies for this kind of printer are expensive and a print job can take a lot of time, it's common practice to restrict access.

In Windows NT, there are four different access types for printers:

- No Access
- Print
- Manage Documents
- Full Control

As of version 3.6, Tivoli Security Management does not include the Manage Documents permission, which is used to alter whether users can delete their own jobs. Table 27 shows the capabilities a user has when granted Tivoli Security Management permissions on printers.

Table 27. User Capability on NT Printers by Access Permission

Capability Enabled by Security Permission	**Access A**	**Full C. F**	**No Acc. N**
Print documents	✓	✓	
Control settings for documents		✓	
Pause, resume, restart, and delete documents		✓	
Change the printing order of the documents		✓	
Pause, resume, and purge the printer		✓	
Change printer properties		✓	
Delete printer		✓	
Change printer permissions		✓	

Note

As of version 3.6 of Tivoli Security Management, you can administer the access rights to a printer connected to a Windows NT system, but as with other shares, you cannot create the share for the printer.

By default, Windows NT gives the CREATOR OWNER group the “Manage Documents” permission, enabling users, for example, to delete their own jobs. If this is not altered, groups managed through Tivoli Security Management will have this function. This is usually what is desired.

The Windows NT printer resource record can be specified from the command line as endpoint and resource type: NT:PRINTER.

4.2.1.5 Windows NT Operating System Files

To ensure the availability of the systems services the resources of the operating system should be protected. An ordinary user should not be allowed to delete or change files, for example, in the `\winnt\system32\drivers` directory. When Windows NT is installed, default permissions are assigned to the system directories on the machine. In most cases, the default protection in Windows NT is likely to be sufficient. However, you need to be familiar with what the defaults are to ensure they are in place and to modify them if that is called for by the security policy.

The default permissions are summarized in Table 28 (Permissions of additional groups like Server Operator, Power User, or Print Operator are not shown).

Table 28. Default Access Permissions to the Windows NT System Resources

Directory (relative to %SystemRoot%)	Administrator	SYSTEM	CREATOR OWNER	Everyone
SYSTEM32	Full Control	Full Control	Full Control	Change
SYSTEM32\CONFIG	Full Control	Full Control	Full Control	List
SYSTEM32\DRIVERS	Full Control	Full Control	Full Control	Read
SYSTEM32\SPOOL	Full Control	Full Control	Full Control	Read
SYSTEM32\REPL	Full Control	Full Control	Full Control	Read

Directory (relative to %SystemRoot%)	Administrator	SYSTEM	CREATOR OWNER	Everyone
SYSTEM32\REPL\IMPORT	Full Control	Full Control	Full Control	Read
SYSTEM32\REPL\EXPORT	Full Control	Full Control	Full Control	Read

For Windows NT Servers, these permissions to the system resources are reasonable for most environments. You may be surprised by that statement when you see that groups, such as CREATOR OWNER and Everyone, have access rights to all of the directories. But remember, resources can only be accessed through a share. If that share does not give the permissions to the Everyone group, the directory can not be accessed by everyone. There is one exception where the share restrictions do not apply. That is when a user logs on to the machine locally. We, therefore, recommend that no user except the administrators should be able to log on to a Windows NT server. Other persons shouldn't even be able to access the servers physically.

For Windows NT Workstations, the default access rights to the operating system resources are also reasonable. Here, users do log on locally, of course. You may want to define additional protection for these resources against local modification and deletion. You could for example set the following ACL on the system root directory:

- Administrators Full Control
- CREATOR OWNER Full Control
- SYSTEM Full Control
- Everyone Read

But restricting access to the operating system resources must be tested carefully to retain a working operating system. There are many files and directories that need to have Change permission for the Everyone group. For example, applications need to write to the %SystemRoot%\system32 directory when installing. The company's strategy could be that any customizing can only be done by the administrator and that applications can only be installed by a centralized software distribution. In this case, it makes sense to restrict access to certain directories and files. But this depends on the security policy.

You might want to restore or set the permissions of the operating system resources to provide the same access rights to all NT stations in the company. There are two ways of doing that:

- Using Tivoli Security Management

  You provide records of the type FILE and DIRECTORY for the NT operating system files with the desired permissions and place them in a separate Security Management profile. Test these permissions carefully after distributing the profile to test systems. Then, apply the access rights to all NT systems by distributing the profile to the production environment.

- Using Tivoli Software Distribution

  There is a tool from Microsoft called `FIXACLS.EXE`. With this tool, you can restore the default permissions of the operating system files and directories. At the time of writing, this was available on Microsoft's FTP site at:

  `ftp://ftp.microsoft.com/bussys/winnt/winnt-public/reskit/nt40/i386.`

  Figure 20 shows the GUI of the program.

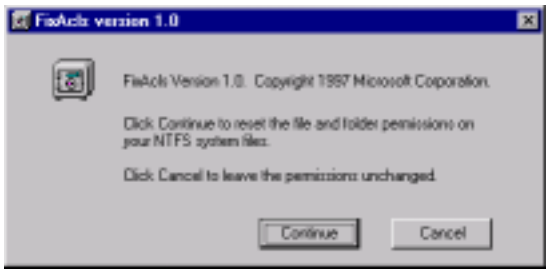


Figure 20. Windows NT FIXACLS.EXE GUI

The tool uses the information contained in the file:

`%SystemRoot%\INF\PERMS.INF`

so, it will need access to it. You have to run the utility with a privileged account to modify the access rights of the system files and directories (groups Administrators or SYSTEM). You can make a Software Distribution package out of the (modified) INF file and the tool and distribute it to the NT systems. In the After Script, you run the tool. Note that you will need two different packages for servers and workstations. This is because the `PERMS.INF` files of Windows NT Server and Windows NT Workstation are different. When setting other than the default permissions, verify the function on test systems before distributing to the whole environment.

For Windows NT Workstations, you should protect the files needed for the operating system boot process. Table 29 shows those files and the default permissions after installation.

Table 29. Default Access Permissions for Windows NT Workstation Boot Files

File	Administrators	SYSTEM	Everyone
BOOT.INI	Full Control	Full Control	Read
NTBOOTDD.SYS (SCSI)	Full Control	Full Control	Read
NTLDR	Full Control	Full Control	Read
NTDETECT.COM	Full Control	Full Control	Read

As the files are protected against deletion, this is quite secure. If you additionally want to restrict the users from reading the BOOT.INI file or executing NTDETECT.COM, you can deny any access to these files for the Everyone group.

4.2.1.6 Example for File and Directory Permissions

Here, we want to give you an example of setting directory and file permissions. Imagine a project in a company located in Austin, Texas that has the project number P891. All members of the project should maintain their weekly status reports in the directory `\Status Reports`. They should be permitted (through a role) to create and change their own reports and read, but not change, the reports of other members in that directory. Only project leaders should be able to read and change all reports (again through a role). This example is depicted in Figure 21.

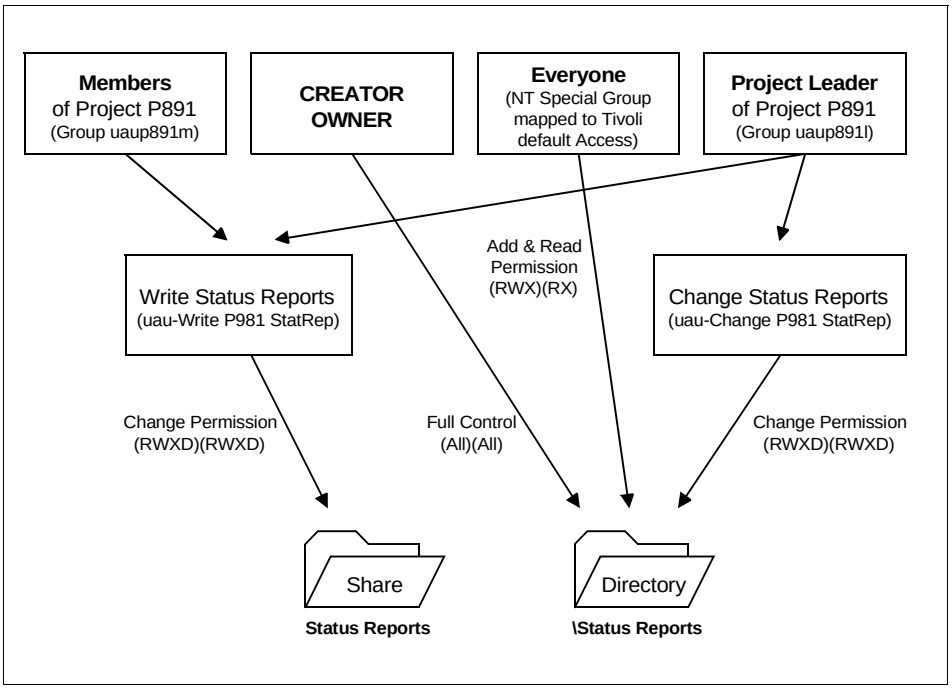


Figure 21. Windows NT File and Directory Access Control Example

We need two groups:

- Members of the Project P891 (uaup891m)
- Project Leader of Project P981 (uaup891l)

And we need two roles:

- Write Status Reports (uau-Write P981 StatRep)

  This role grants access to the share, through which the report can be accessed. We grant the access Change (RWXD)(RWXD). It may look like all members of the project have Change rights to the files, but remember, that the final access rights are made up of the share *and* the file/directory rights.

- Change Status Reports (uau-Change P981 StatRep)

  This role grants Change (RWXD)(RWXD) access to the `\Status Reports` directory and its files.

The Everyone group has RWX rights to the directory. That means that they can list the directory contents, create new files in it, and execute programs (if

there are any). They can not change permissions or take ownership of the directory.

Any file created in this directory will have read and execute permission for the Everyone group (which is mapped to the Tivoli default access), because the default creation mask is set to RX. Only the owner of a report is allowed to modify it. This is realized by the access control entry, CREATOR OWNER Full Control (All)(All), which is implemented by default when a Windows NT directory is created.

The project leader group is allowed to read and change the status reports through the role `uau-Change P981 StatRep`.

4.2.1.7 Windows NT Registry

The Windows NT registry is the central database where configuration data for the operating system and applications is stored. It contains information for applications, hardware and device drivers, network configuration, and card settings. In the majority of cases, changes to the registry are made by means of utilities in the Windows NT Control Panel or Setup Applications. There is also the possibility to change values of the registry directly through the Windows NT registry Editor (`REGEDIT.EXE` or `REGEDT32.EXE`). Additionally, there are many shareware tools that make changes to the registry. Therefore, it is of little use to protect the registry by restricting the access to the Windows NT registry editor. Users and crackers will find other ways to do their modifications. The Windows NT registry has to be secured with its native security.

The registry is organized by hives. Here are some important hives of the database with their default permissions. (We don't cover the structure of the Windows NT registry here. Use a Windows NT system book for more details.):

- **HKEY_LOCAL_MACHINE\SAM** Stores the security information for user and group accounts. This hive is read protected even for administrators.
- **HKEY_LOCAL_MACHINE\SECURITY** Stores security information. This hive is read protected even for administrators.
- **HKEY_LOCAL_MACHINE\SOFTWARE** Stores all information concerning installed software. Everyone can read the program configuration. The CREATOR OWNER group has full access.
- **HKEY_LOCAL_MACHINE\SYSTEM** Stores information for the system startup. The system can not start without this information. Everyone is allowed to read the data by default.

- **HKEY_USERS\DEFAULT** Stores the default user profile. Everyone can read the information.

Every hive in the NT registry has its own Access Control List (ACL). The ACL regulates the access to the stored data under that hive. As you can see in the list above, even the administrator is not allowed to view or control every part of the registry. This assures, for example, that administrators can not see the user passwords or other security information.

The default permissions of the hives are set reasonably well from Windows NT v4.0 onwards. With some exceptions, the Everyone group has read access to all the data. If the company does not want ordinary users to see particular registry information, you can restrict the access of the hives for the Everyone group. We recommend to change access permissions to the registry rarely.

Be careful not to modify the access for the SYSTEM group. The operating system needs to have access to the information.

4.2.2 UNIX Data Resources

Users have different requirements for accessing files, directories, and devices, such as printers. It depends on the job function, whether, for example, the read permission to these resources is enough, or the writing privilege is also needed. In 2.3.5, “Groups and Roles” on page 32 and in 3.4, “Groups and Roles” on page 65, we assessed these dependencies between users and resources in roles. In Tivoli Security Management, the members of a security group receive access permissions to resources through security roles.

But the concept of roles represents a problem for the standard file protection of the majority of UNIX file systems. In UNIX, we are only able to assign the read, write, and execute permissions to the file owner, the members of a group, and all other users (world). However, this access method is not granular enough to apply the roles concept. Quite often, roles apply to members of several different groups, but we are not able to assign more than one group to files or directories in UNIX.

Some UNIX systems have extended the familiar file protection modes to Access Control Lists (ACL’s) in order to assign the permissions to several groups or directly to particular users. In theory, the security roles definition could be transformed into a list of access control entries assigned to a resource. Every entry consists of a group name and permissions. But ACLs are rarely used in the field, because they are not implemented in a standard

way and not supported at all from many UNIX tools. We do not recommend adding the use of ACLs, if they have not been already implemented in the company.

The solution to handling role definitions in UNIX is the Tivoli Access Control Facility (TACF) delivered with Tivoli Security Management. This extends the security mechanism of the UNIX operating system. If TACF is running, every attempt by a user to access a resource is validated first by TACF before the request is passed to the operating system.

Figure 22 shows an example of the access validation of TACF on UNIX. Three users defined on a UNIX system want to access a file resource. These requests are validated by two security mechanisms: first TACF, and if passed by TACF, then the UNIX native security mechanism. Both are represented with a filter symbol in the figure. Only the write-request from user-3 was granted. The request-1 was denied by TACF, and the request-2 was denied by UNIX. Note that user-1 would have access to the file if TACF had been disabled.

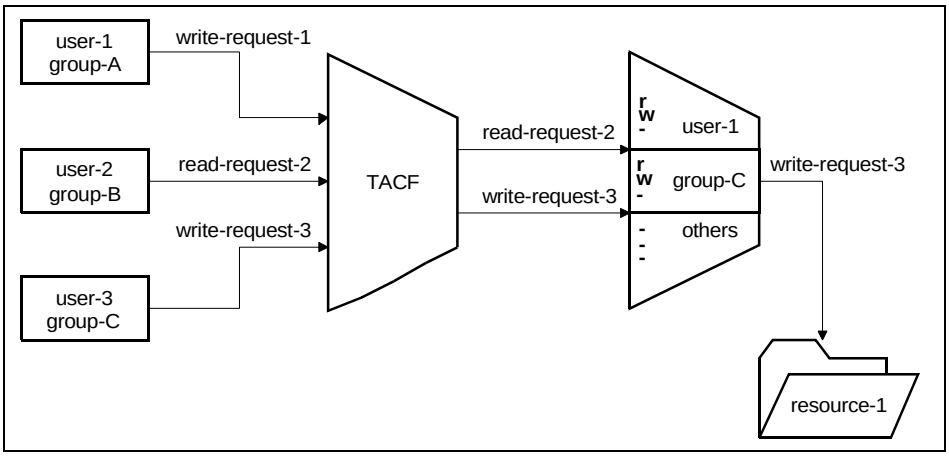


Figure 22. Example of Accessing a Resource with TACF

The access validation by TACF is done with the Tivoli Security Management user, group, role, and resource entries that are distributed to the UNIX system and stored in the equivalent TACF records in the TACF security database (see Figure 23). Notice that the group names of this security database are not defined as group names in UNIX, they are used for grouping users internally to TACF.

Looking at Figure 23, we can see why user-2 and user-3 were granted the described permissions. TACF denied the write-request of user-1, because the group this user belongs to has not been assigned a role that has access to resource-1, and the default access for the resource has been defined as no access. The access of the other users was permitted, because there is a proper role (print-resource-1 and edit-resource-1) defined, which grants access.

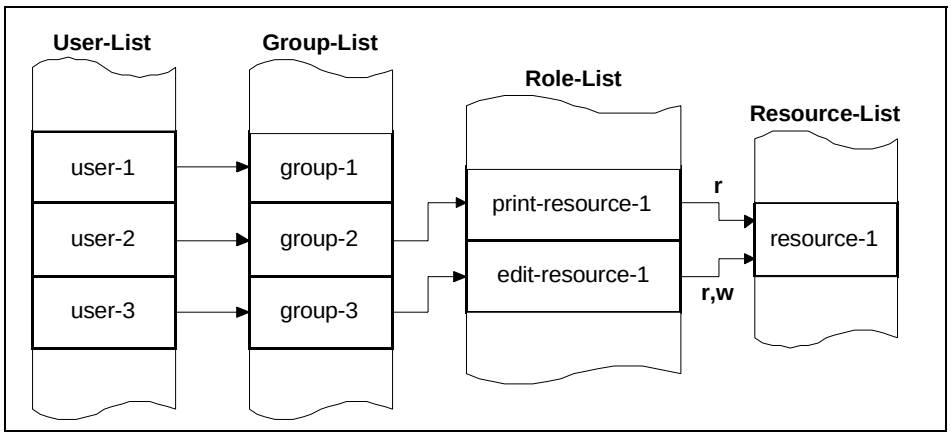


Figure 23. Visualization of ACL's in TACF

The evaluation of access requests is more complex than in the figure above. Full details about TACF resources and access rules are documented in the *Tivoli Security Management User's Guide*, the *Tivoli Security Management Reference Manual for TACF*, and in the Redbook *Managing Access from Desktop to Datacenter: Introducing Tivoli Security Management* (IBM form number SG24-2101).

Synchronization between UNIX and TACF

Notice that TACF maintains its own database and does not create users, groups, or resources in the UNIX operating system that are defined in the TACF database. In addition, TACF does not change by default the file permissions for the owner, group, or others.

The users and groups defined in UNIX can be imported into the TACF database with the task Synchronize TACF/UNIX Users & Groups, which is available if you have installed the TACF Tasks, Monitors, and Event Integration piece of Tivoli Security Management.

TACF is able to create ACL's automatically for files on several supported UNIX systems, such as Sun Solaris. For these operating systems, you can turn on the TACF synchronization feature to have the permissions mapped. See the description of the SynchUnixFilePerms token for the SEOS.INI file in the *Tivoli Security Management Reference Manual for TACF*.

In operating systems that do not support ACLs or where the use of ACLs is not wanted, we cannot fully synchronize TACF protection with UNIX. TACF can be made to modify group permissions, but in general, TACF is responsible for the actual security by acting as a filter on top of the operating system. *However*, if TACF is not available for any reason, the access to those resources reverts to the currently set file protection mode of UNIX.

TACF employs a number of methods to ensure that it is not disabled by anyone other than an authorized person. As always with any security system, the greatest threat is posed by those who have physical access to the TACF server.

Besides the ability to define roles, we also gain a strong auditing function with TACF. We can generate events when a user accesses security-relevant resources, such as the system files /etc/passwd and /etc/services or user confidential data. The auditing of resource accesses can be engaged in case of success and/or failure, and the event can be forwarded to a Tivoli Enterprise Console for which security-related rules are provided.

TACF is also useful for delegating and reducing the access rights of the super user, for example, the root login. The root user is treated no differently by TACF in the validation of access rights for users. For example, a super user cannot change the ownership of a file if it is not allowed by TACF.

In the following sections, we will first review the protection of files with the UNIX file system, and then, describe how to use the additional protection afforded by TACF.

The protection of files and directories is done in two stages. The first is the setting of the permissions in the UNIX file system. We do this even though TACF is used, because if TACF approves a request, it must still be granted through UNIX for the request to proceed. In most existing installations, these permissions have been built up over a period of time and are, therefore, already in place. As TACF will use its own roles and groups to form access rules, we need to leave the UNIX configuration as uncomplicated as possible, possibly combining smaller system groups into larger ones and using TACF to implement restrictions through more specific security groups.

The second stage is to further define the privileges users have over resources by defining entries in Tivoli Security Management that are mapped to the TACF database after a distribution of the security profile. The rules for validation of a request are also mapped into the security database of TACF.

4.2.2.1 UNIX File and Directory Permissions

If the correct UNIX permissions are not already in place, we must start by assigning file permissions to the three classes: owner, group, and other. File permissions can only be changed by the file's owner or a super user.

Because the meaning of the three file permissions read, write, and execute are a little different for files and directories, we regard them separately. Table 30 shows the meaning of file permissions for directories:

Table 30. Directory Permissions Generally Used in UNIX

Character	Permission	Meaning
r	READ	Allows you to find out which files are in the directory.
w	WRITE	Allows you to add, rename, or remove entries in that directory (but see x).
x	EXECUTE	Allows you to get full details about the contents of the directory, for example, the file sizes or owners. The user needs this permission to make this directory their current working directory or to open files contained in the directory or any of the directory's subdirectories.

The file permissions shown in Table 31 apply to devices, named sockets, and so on exactly as they do for regular files, but they do not apply for symbolic links.

Table 31. File Permissions in UNIX

Character	Permission	Meaning
r	READ	Allows you to open and read the contents of a file.
w	WRITE	Allows you to overwrite the file with a new one, modify the contents, or make it longer or shorter.
x	EXECUTE	The binary file or executable script can be run. If it is a script, you also need the read permission for execution because the file content has to be read. The read permission for binary files is not necessary. Notice that you also have to have the access permission to the full directory path that contains the file.

For example, suppose the ITSO in Austin wants to store the Redbooks of the current projects in the directory `/redbooks`. We would take the following steps to set the file privileges for that resource:

1. Create the UNIX group, for example `aubooks`.

   The members of this group are all persons that need to have access to the Redbooks, for example, the managers, editors, project leaders, and residents.

2. Set the ownership of files and directories.

   ```
   chown -R root:aubooks
   ```

3. Set the file permissions for all files.

   ```
   chmod -R 660 /redbooks
   ```

   or

   ```
   chmod -R u=rw,g=rw,o= /redbooks
   ```

4. Set the file permission for the directory and subdirectories.

   ```
   find /redbooks -type d -exec chmod 770 {} \;
   ```

   or

   ```
   find /redbooks -type d -exec chmod u=rwx,g=rwx,o= {} \;
   ```

We now have adjusted the appropriate permissions to files and directories that already exist. But we haven't covered the access rights of newly created files. Therefore, we have to think about the user file-creation mode mask set by umask. This mask is set for particular users by default in the `.login`, `.cshrc`

or `.profile`, or in the system `/etc/profile` file. In some UNIX systems, you can also define the umask value for all users. It is recommended to set the umask globally, where available.

4.2.2.2 TACF File and Directory Access Control

We use TACF to define additional access restrictions for users that already have permissions to resources in the UNIX file system. In the above example, the members of the group aubooks have different reasons for accessing Redbooks. For example, residents need write and print privileges for those files representing their own Redbooks, and a manager needs privileges to all Redbook files. Before we can define the roles that handle these different privileges to resources, we have to add new groups and resource entries in the security database of TACF.

UNIX (TACF) file and directory resources are defined in the Tivoli Security Management through the command line, using the endpoint and resource type UX:FILE.

For each of the UNIX file and directory resources for which we filled out tables in 2.3.6, “Data Resources” on page 38, we must create a security resource record.

Next in our example, the security groups are needed for the ITSO employees, such as department managers, editors, project leaders, and residents. The groups that are defined on the UNIX operating system are unlikely to match what is required as security groups, although populating security groups from system groups is sometimes an option as a starting point.

The next step is to assign the group members by selecting user accounts either directly from the UNIX node(s) or from Tivoli User Administration user profiles. It is obviously preferable to use Tivoli User Administration to manage user account information although this is not a prerequisite to using Tivoli Security Management.

Tivoli Product Integration

Each new release of Tivoli User Administration and Tivoli Security Management will improve the integration between them. Already, the membership of a user in security groups is also stored in the user record of a user profile. Tivoli Security Management can modify groups automatically if a user is deleted or moved through Tivoli User Administration.

4.3 Network Connectivity

UNIX systems, through TACF, have the capability to define restrictions on network connectivity. Two resource types (TCP and CONNECT) define these restrictions for incoming and outgoing traffic. The data used to determine what resources to define was gathered in 2.3.7, “Network Connectivity” on page 40. As with other TACF resources, connections can have resource records defined but with the warning mode set. This performs the security check and always allows the request (as far as TACF is concerned), logging any that would fail the check.

4.3.1 Incoming

The requirements for restricting incoming requests will depend on the very specific needs of each implementation. Other protection mechanisms may already be in place, but common restrictions defined in this category include restricting telnet connections to a specific list or range of hosts.

Another common use of this resource type is to provide auditing of connections made through a variety of services.

Command-line activity of this resource is done using the UX:TCP endpoint and resource type.

4.3.2 Outgoing

As with incoming restrictions, the use of this resource type will be very dependent on the aims of the implementation. While the incoming resource type restricts connections per TCP service or dynamic port names assigned by the portmapper, the outgoing resource type simply identifies which remote hosts the TACF node can connect to. You specify the host, and whether this system has access to it, or not, together with any auditing that you require.

Command-line activity for this resource is done using the UX:CONNECT endpoint and resource type.

4.4 Terminal Access

Terminal access information was gathered in 2.3.8, “Terminal Access” on page 42. TACF enables UNIX systems to restrict logins from the local host, another host on the network, or X-terminals. This would typically be used to implement security policy regarding administrator-only login capability to server systems.

Note

You should not define 0.0.0.0 as a terminal resource. This terminal is used by the Tivoli object dispatcher (oserv) daemon. Restricting this terminal will cause the oserv to fail.

An equivalent login restriction function is provided in Windows NT. As it is implemented as a user attribute, this can be managed through Tivoli User Administration.

The Tivoli endpoint and resource type are UX:TERMINAL. As with other resources, a primary requirement driving the use of terminal resources are the needs for auditing who is logging on and from where.

4.5 Applications

In 2.3.9, “Applications” on page 43, we looked at analyzing the existing management of application security in the enterprise. The Tivoli endpoint and resource type UX:PROGRAM allows us to audit executable use and define restrictions on who has access to run the program. Programs defined in this resource are monitored by the TACF watchdog daemon to ensure that they are not modified.

Note that this resource applies to records that are marked as setgid or setuid in the UNIX file system. Use the UX:FILE resource to restrict access for programs that do not set user or group IDs.

4.6 Critical Files

Refer to 2.3.10, “Critical Files” on page 43, where we determined that if any files are sensitive and static enough to be watched, to alert an administrator in case of modification. The UX:SECFILE resource defines a file to be watched by the TACF watchdog daemon. This is likely to be a requirement specific to each implementation.

Chapter 5. Security Auditing

This chapter completes the discussion on architecture and design by looking at the very important topic of auditing. Auditing is a very useful feature for monitoring the resource accesses in the network. You can track any access to a file, directory, printer, or a system itself. This can be done for successes, failures, or for both failures and successes. This will bring you the following benefits:

- You can identify potential intruders trying to login to the system.
- If a breach was detected, you can follow the intruder's actions taken on the system (if they affected audited resources).
- You can collect statistics on a resource. With this data, you can, for example, optimize the hardware usage. You will need to store often used data on machines with a high data throughput.

  You can identify rarely or never used data. So you can put it on a less expensive media, such as a CD-ROM or backup tape.
- You can recover (or at least have an idea what to recover) minor disasters produced by a user mistake by watching the resources he/she touched in a certain action.
- You can use the audit data as legal evidence (depending on specifics of law) to show the login times or the resources affected.

To have better statistics and more granularity in the logged events, it would be good to have as many events logged as possible. But this leads to the potential for the following consequences:

- High network traffic
- A lack of visibility for essential events
- Overloading the Tivoli Enterprise Console (TEC)
- Time consuming event data backups
- Large amounts of storage required for logs and events

Therefore, what we need to do is find the middle ground between event granularity and system load. Our approach is to reduce the Tivoli event generation at the endpoint to the minimum and let only specific and needed events get to the TEC. Remember, there are also other sources for events besides security that have to be sent over the network and processed by the TEC.

Designing an auditing implementation is an ongoing process. You do not design it once and implement it, and that is it. You will have to analyze the

collected event data frequently and make design changes. So, if you see that there are masses of events coming from a particular source, you should verify your auditing strategy on that point. If you have resources that do not produce any events at all, this may require an alteration to settings, too.

While there are some differences in the way auditing is handled on the different platforms, there is some common ground, particularly in the advice to employ the Tivoli Enterprise Console to manage security audited events. This chapter is based around experiences with Windows NT, but many of the principles can be applied to any platform supported by Tivoli Security Management.

Note

In the first release of Security Management for the OS/390 endpoint, the Audit Log Report Task does not support the OS/390 endpoint. Also, integration with TEC must be done through Netview until such time as TEC integration is supported directly.

5.1 Windows NT Auditing Example

What we aim to describe here is effective centralized auditing for Windows NT platforms. For that task, we will use the Tivoli Enterprise Console to receive the events from the Windows NT Managed Nodes. On the clients, we use the TEC Windows NT Event Log Adapter to convert the NT events to the TEC data structure, filter the events, and forward them to the TEC.

In the first step, we look at the data structure of an NT event to be able to design appropriate TEC event classes and write a corresponding format file for the NT Event Adapter. This step is a very important one, because we can only apply filters and run effective queries on the event repository of the TEC if the event class structure is well designed.

After that, we will go through the different Windows NT resource types and give suggestions on how to monitor them.

5.1.1 Windows NT Event Structure

Windows NT logs all auditing events to the security part of the Windows NT Event Log. The Event Log is not a plain text file. Therefore, you cannot view the logged events with an editor. However, the TEC Windows NT Event Log Adapter can generate a Tivoli event out of a Windows NT event. In

Windows NT, the events can be viewed with the Windows NT Event Viewer, shown in Figure 24. This is located in the Administrative Tools in the Start Menu.

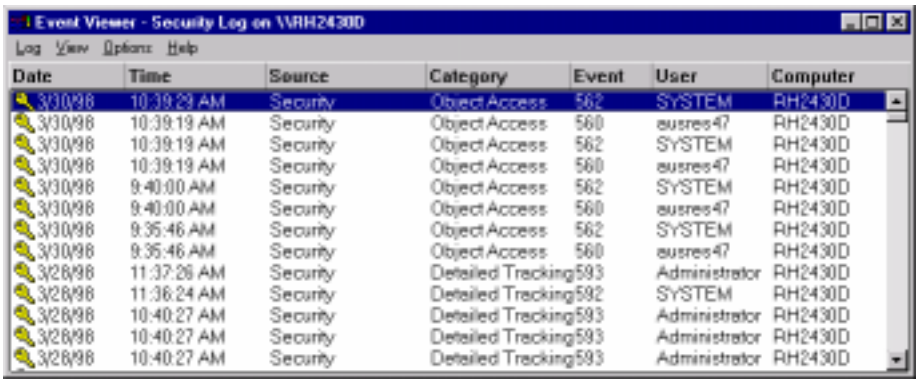


Figure 24. Windows NT Event Viewer

Here, all events are displayed in a list, with the most important fields, such as the date, time, or the user who is associated with that event. But the data shown is not all the data that is recorded. If you open the detail view of an event, you see all the information that is logged by the system, as in Figure 25.

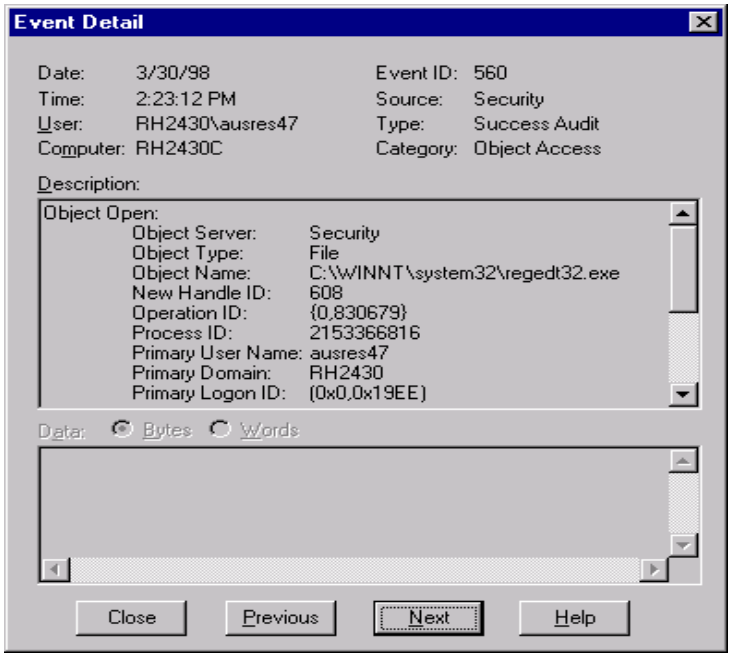


Figure 25. Windows NT Event Viewer Detail View

The information shown in the Description box is dependent on the *Event Category*. For example, for the category shown, Object Access, there are fields that describe how the object was accessed (reading, executing, and so on) that are not necessary for other event categories. So, every event has its event category. The Windows NT event categories are:

- System Event Category
- Logon/Logoff Category
- Object Access Category
- Privilege Use Category
- Account Management Category
- Policy Change Category
- Detailed Tracking Category

We will focus our examples on the Logon/Logoff and the Object Access categories. We will describe the details of those two categories in the following sections, but first, we will look at the common information both categories share.

The common fields are shown in the upper part of the dialog in Figure 25, from Date to Category. The following field description includes the slot in the TEC class the field is mapped to (if it is already mapped). This mapping is based on the TEC Windows NT Event Log Adapter format file (tecad_nt.fmt) created at installation time.

Date and Time This is the date and time when the event was generated. These fields are mapped to the slot *date* of the TEC class *NT_Base*.

User This is the name of the account that was logged on when the event was generated. This field is mapped to the slot *sid* of the class *NT_Base*.

Computer This is the computer name where the event occurred. This field is not mapped to a slot. Instead, Tivoli fills the *hostname* slot of the *EVENT* class.

Event ID This is the identifier for a particular event. For example, the ID 560 represents an Object Open event. That means that either an object has been successfully opened, or the request to open an object was rejected. (At the end of this list, there is a list of references where you can find more information on event IDs.) This field is mapped to the slot *id* of the class *NT_Base*.

Source The source field contains the name of the software that logged the event. This could be an application, but in security, it is a component of Windows NT. The field is mapped to the slot *sub_source* of the class *EVENT* from which the *NT_Base* class is inherited.

Type The Type field describes the type of the event. The possible values here are Information, Warning, Critical Error, Success Audit and Failure Audit. The field is mapped to the *eventType* slot of the *NT_Base* class.

Category The category field classifies the event to one of the Windows NT Event Categories. This field is mapped to the *category* slot of the *NT_Base* class.

5.1.1.1 Windows NT Event Structure Resources

The following resources will give you more information about the Windows NT event structure:

- *Windows NT Resource Kit* Audit Categories help file: AUDITCAT.HLP. Here you can find a description of all Audit Categories.

- *Windows NT Resource Kit* messages help file: NTMSGS.HLP. This help file contains a list of most of the error and system-information messages that are implemented in Windows NT. You can list the contents by event ID, event source, or in alphabetical order. The contents of the help file is very useful when performing the Windows NT event field to Tivoli slot mapping, as it provides the exact string that is generated by Windows NT.

We now come to the event-specific part which is shown in the lower part of the dialog that was shown in Figure 25.

5.1.1.2 The Object Access Category

The Object Access Category is the most important one for us. This category tracks all object accesses for files, directories, printers, and the registry. The following is an example event for a Windows NT file access:

```
Date:     3/27/98              Event ID:    560
Time:     20:23:12 PM          Source:      Security
User:     RH2430\ausres47      Type:        Success Audit
Computer: RH2430C              Category:    Object Access

Description:
   Object Open:
      Object Server:        Security
      Object Type:          File
      Object Name:          C:\WINNT\system32\regedt32.exe
      New Handle ID:        892
      Operation ID:         {0,915100}
      Process ID:           2153366816
      Primary User Name:    ausres47
      Primary Domain:       RH2430
      Primary Logon ID:     (0x0,0x19EE)
      Client User Name:     -
      Client Domain:        -
      Client Logon ID:      -
      Accesses:             SYNCHRONIZE
                            Execute/Traverse
      Privileges:           -
```

Here, the file `C:\WINNT\system32\regedt32.exe` was accessed (executed) on the computer named RH2430C by the user ausres47 of the domain RH2430.

We will now give descriptions of all the Object Open event-specific fields that are in the Windows NT record. This event here is named *Object Open*. The specific data for that event begins with the words Object Open and ends with `Privileges: -`. Here, we will describe all the fields of the event *Object Open* of the category *Object Access*:

Object Server The name of the subsystem server process that logged the event.

Object Type	The type of the object that is being accessed. For example, this could be a file or directory. This can be used to group the events in the TEC and assign these groups to different administrators.
Object Name	The name of the object that is being accessed. For example, for a file resource, this could be `C:\WINNT\system32\regedt32.exe`.
New Handle ID	The handle identifier of the object. If the access to the object failed, this is a dash (-). This is the second way to determine whether the access to a resource was successful or not. (The first one is the event type.)
Operation ID	A unique identifier that associates multiple events that are the result of a single operation. This can be used for filtering and correlation of events in the TEC.
Process ID	The identifier of the client process accessing the object.
Primary User Name	The user name of the user requesting the object access. This can also be the user name with which the server process is logged on, for example, if a user accesses a resource through the network.
Primary Domain	The name of the computer where the event was generated. If the computer is a member of a Windows NT Server domain, this can also be the name of the domain.
Primary Logon ID	A unique identifier assigned by the operating system when the primary user logged on.
Client User Name	The user name of the user on whose behalf the primary user is accessing the object. If this is specified, it is usually the user name of a user logged on interactively or remotely. If this is not specified, the field contains a dash (-).
Client Domain	The name of the computer. If the computer is a member of a Windows NT Server domain, this can also be the name of the domain containing the client user's account.
Client Logon ID	A unique identifier assigned by the operating system when the client user logged on.
Accesses	The types of access with which the user tried to access the object.

Privileges Special user privileges invoked to perform an object access. If there are none this field contains a dash (-).

A knowledge of these descriptions is essential when constructing classes, writing TEC adapter format files, and creating queries on the TEC database.

5.1.1.3 The Logon/Logoff Category

The Windows NT Logon/Logoff category contains events that describe single successful or unsuccessful logons and logoffs. We can group the contained events as follows:

Successful Logon (Windows NT event - Successful Logon.) This type of event is mapped to the TEC class *NT_Logon_Successful*.

Logon Failure (Windows NT events - Unknown User name or Bad Password, Account Currently Disabled, Logon Type Restricted, Password Expired, and Unsuccessful Logon.) This type of event is mapped to the T/EC class *NT_Logon_Failure*.

User Logoff (Windows NT event - User Logoff.) This type of event is mapped to the TEC class *NT_User_Logoff*.

This is an example event of a Windows NT logon attempt:

```
Date:     3/31/98              Event ID:    529
Time:     10:57:43 AM          Source:      Security
User:     NT AUTHORITY\SYSTEM  Type:        Failure Audit
Computer: RH2430C              Category:    Logon/Logoff

Description:
   Logon Failure:
      Reason:                 Unknown user name or bad password
      User Name:              ausres47
      Domain:                 RH2430
      Logon Type:             2
      Logon Process:          User32
      Authentication Package: MICROSOFT_AUTHENTICATION_PACKAGE_V1_0
      Workstation Name:       RH2430C
```

Here, the user ausres47 tried to logon to the Workstation RH2430C. Either the user name was unknown, or the user provided a bad password.

We will now give descriptions of all the Logon Failure event-specific fields that are in the Windows NT record. The event here is named *Logon Failure*. The specific data for that event begins with the words Logon Failure and ends with

Workstation Name: RH2430C. We will describe here all the fields of the event Logon Failure of the category *Logon/Logoff*. The description includes the slot in the TEC class and the field it is mapped to (if it is already mapped). This mapping is based on the TEC Windows NT Event Log Adapter format file (tecad_nt.fmt) created at installation time.

Authentication Package The system component that authenticates logon attempts. The standard Windows NT authentication package is: `MICROSOFT_AUTHENTICATION_PACKAGE_V1_0`.

Domain The name of the computer. If the computer is a member of a Windows NT Server domain, this can also be the name of the domain containing the user's account.

Logon ID A unique session identifier given when a user is logged on.

Logon Process The process submitting the logon/logoff request. For example, an interactive logon is requested by User32, and a network logon can be requested by NTLanMan.

Logon Type Indicates the method of logging on. For example, a value of 2 indicates a normal interactive logon, and 3 indicates a remote (network) logon.

Reason The cause of a failed logon or logoff attempt. For example, this could be Unknown user name or bad password.

User Name The user name of the user who logged on or off.

A knowledge of these descriptions is essential when constructing classes, writing TEC adapter format files, and creating queries to the TEC database.

5.1.2 Event Classes

The Tivoli Enterprise Console is a very flexible application that can be integrated to collect any kind of event information throughout an enterprise. The software lets you easily integrate new event sources and types.

The TEC Windows NT Event Log Adapter already comes with class definitions for the generation and processing of a Windows NT event. It covers all events for the three NT logs: System, Application, and Security. We will improve the quality of the event data for queries on the TEC database and a better display on the GUI of the Event Console.

The standard implementation of the Windows NT object open event results in the following TEC event:

```
NT_Base;
    server_handle=1;
    date_reception=891374612;
    event_handle=1;
    source=NT;
    source=Security;
    origin=9.53.65.158;
    sub_origin=N/A;
    hostname=rh2430c;
    adapter_host=N/A;
    status=OPEN;
    administrator=' ';
    acl=[ admin];
    severity=WARNING;
    date='Mar 31 14:01:56 1998';
    duration=0;
    msg='Object Open: Object Server: Security Object Type: File Object Name: C:
        \Temp\SampleFax.txt New Handle ID: 104 Operation ID: {0,2136507} Proces
        s ID: 2160807072 Primary User Name: ausres47 Primary Domain: RH2430 Pri
        mary Logon ID: (0x0,0x1F488F) Client User N';
    msg_catalog=none;
    msg_index=0;
    num_actions=0;
    credibility=0;
    repeat_count=0;
    cause_date_reception=0;
    cause_event_handle=0;
    category=3;
    eventType=AuditSuccess;
    id=560;
    sid=RH2430\ausres47;
END
```

Reading this result, we notice that the complete information that is provided in the NT Description field of the event details dialog (see Figure 25 on page 102) is mapped as one string to the *msg* slot of the class *NT_Base*. In addition, this string is even cut off because of the maximum string length of a slot in the TEC. Looking at this event in the Event Console makes it hard for the administrator to understand what happened without viewing the details dialog. For example, you cannot determine the type and name of the resource that was accessed, and you can not run a query to find out all accesses of a certain user (without string processing).

The solution for this problem is to assign every piece of information that NT provides to a dedicated slot in a TEC class at event generation time. Therefore, we need to define the classes to which we want to assign the values first. We suggest the following definitions:

```
TEC_CLASS :
    NT_Specific_Object_Open ISA NT_Base
    DEFINES {
        objectServer: STRING;
        objectName: STRING;
        newHandleID: STRING;
        operationID: STRING;
        processID: STRING;
        primaryUserName: STRING;
        primaryDomain: STRING;
        primaryLogonID: STRING;
        clientUserName: STRING;
        clientDomain: STRING;
        clientLogonID: STRING;
        accesses: STRING;
        privileges: STRING;
        };
END
TEC_CLASS :
    NT_FileDir_Open ISA NT_Specific_Object_Open;
END
TEC_CLASS :
    NT_Reg_Open ISA NT_Specific_Object_Open;
END
TEC_CLASS :
    NT_Printer_Open ISA NT_Specific_Object_Open;
END
```

The class *NT_Specific_Object_Open* is inherited from the class *NT_Base,* and therefore, includes all slots of that class. The specific slots for the object access events are added. The list corresponds to the fields delivered by Windows NT. We will cover all events that are generated by accesses of the resources: file, directory, printer, and the registry. Resources not covered will be processed as an event of the standard class, *NT_Object_Open*, which is not as detailed as this definition. If you have other resources that are important to have monitored with the same quality, add them to the class definitions.

The classes *NT_FileDir_Open*, *NT_Reg_Open*, and *NT_Printer_Open* are inherited from the class *NT_Specific_Object_Open,* and therefore, have all the slots defined in that class. You may be asking why we do not implement a single class for all object accesses and define a slot for the resource type. This is possible, as well, and is a good solution in certain environments. However, the solution with one class per object type has an advantage; you can define event groups in the TEC based on the class name and assign these event groups to different administrators, if you wish.

These definitions can be at the end of the tecad_nt.baroc class definition file without changing any existing classes.

The adapter classes for auditing implemented at installation time, (*NT_Logon_Failure*, *NT_Logon_Successful*, and *NT_User_Logoff*), give the same room for improvement. Here, most of the information is also mapped to the message slot of the class. To have the maximum granularity, you will have to define the classes like this:

```
TEC_CLASS :
    NT_Logon_Failure ISA NT_Base
    DEFINES {
        reason: STRING;
        userName: STRING;
        domain: STRING;
        logonType: STRING;
        logonProcess: STRING;
        authenticationPackage: STRING;
        workstationName: STRING;
        };
END
TEC_CLASS :
    NT_Logon_Successful ISA NT_Base
    DEFINES {
        userName: STRING;
        domain: STRING;
        logonID: STRING;
        logonType: STRING;
        logonProcess: STRING;
        authenticationPackage: STRING;
        workstationName: STRING;
        };
END
TEC_CLASS :
    NT_User_Logoff ISA NT_Base
    DEFINES {
        userName: STRING;
        domain: STRING;
        logonID: STRING;
        logonType: STRING;
        };
END
```

The slots of the three classes are named and ordered according to the Windows NT event data structure.

5.1.3 Format File

Now that we have designed the TEC classes for object accesses and logon/logoff events, we can create the corresponding format file entries to map the Windows NT event fields to the TEC event class slots. Always have the Administrators in mind who will be observing, evaluating, and reacting to the events (if not automated). They must be able to see which is the most important event that should be analyzed first. We suggest putting a brief, but concise, description in the msg slot that tells the administrator *what* happened to *which* resource. If something else is important for you to see on the first view, put it in a slot and display it on the administrator's console. We next show our suggestions for the format file entry for the three object access

classes *NT_FileDir_Open, NT_Printer_Open,* and *NT_Reg_Open* that we designed in the previous section.

```
FORMAT NT_FileDir_Open FOLLOWS NT_Base
%t %s %s %s %s %s %s Object Open: Object Server: %s Object Type: File Object Name: %s+
New Handle ID: %s+ Operation ID: %s+ Process ID: %s+ Primary User Name: %s+ Primary
Domain: %s+ Primary Logon ID: %s+ Client User Name: %s+ Client Domain: %s+ Client Logon
ID: %s+ Accesses %s+ Privileges %s*
objectServer $8
objectName $9
newHandleID $10
operationID $11
processID $12
primaryUserName $13
primaryDomain $14
primaryLogonID $15
clientUserName $16
clientDomain $17
clientLogonID $18
accesses $19
privileges $20
-tmp $9
msg PRINTF(“File/Dir: %s access attempt”, tmp)
END

FORMAT NT_Printer_Open FOLLOWS NT_Base
%t %s %s %s %s %s %s Object Open: Object Server: %s Object Type: Printer Object
Name: %s+ New Handle ID: %s+ Operation ID: %s+ Process ID: %s+ Primary User Name: %s+
Primary Domain: %s+ Primary Logon ID: %s+ Client User Name: %s+ Client Domain: %s+
Client Logon ID: %s+ Accesses %s+ Privileges %s*
objectServer $8
objectName $9
newHandleID $10
operationID $11
processID $12
primaryUserName $13
primaryDomain $14
primaryLogonID $15
clientUserName $16
clientDomain $17
clientLogonID $18
accesses $19
privileges $20
-tmp $9
msg PRINTF(“Printer: %s access attempt”, tmp)
END

FORMAT NT_Reg_Open FOLLOWS NT_Base
%t %s %s %s %s %s %s Object Open: Object Server: %s Object Type: Key Object Name: %s+ New
Handle ID: %s+ Operation ID: %s+ Process ID: %s+ Primary User Name: %s+ Primary Domain:
%s+ Primary Logon ID: %s+ Client User Name: %s+ Client Domain: %s+ Client Logon ID: %s+
Accesses %s+ Privileges %s*
objectServer $8
objectName $9
newHandleID $10
operationID $11
processID $12
primaryUserName $13
primaryDomain $14
primaryLogonID $15
clientUserName $16
clientDomain $17
clientLogonID $18
```

```
accesses $19
privileges $20
-tmp $9
msg PRINTF("Registry: %s access attempt", tmp)
END
```

When creating the Windows NT event field to Tivoli slot mapping, it is very useful to have a look in the *Windows NT Resource Kit* messages help file (NTMSGS.HLP). This help file contains a list of most of the error and system-information messages that are implemented in Windows NT. It shows the exact string that is generated by Windows NT. You can list the contents by event ID, event source, or in alphabetical order.

5.1.4 Reducing the Event System Load

The events that are logged by Windows NT produce different kinds of load on several systems. For example, generating events for an object access means system load on the Workstation or Server where the object resides. Windows NT writes the events to the Event Log where it consumes hard disk space. The TEC Windows NT Event Log Adapter runs as a service and translates the generated events from NT to the format that the TEC understands. This translated event is sent through the Tivoli Framework, which is based on TCP/IP communication. So, every piece of information causes network traffic. The TEC host receives the event and processes it. That means system load for the TEC host. If there are automated actions on the events, this produces even more system load. All the events are stored in the TEC database (which does not have to be on the same host), which grows with every event. This process is shown in Figure 26.

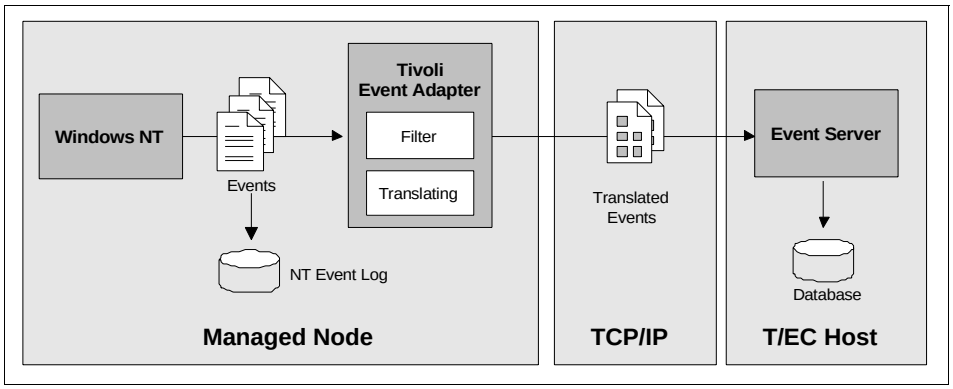


Figure 26. Routing of a Windows NT Event to the TEC Database

Note that there can be many Managed Nodes, sometimes hundreds, and many resources being audited. This multiplies the load for the network, the

TEC and database host, and the database. Consider the time and cost consumed by backups of the event data as well.

As you can see, it is very important to be careful of the amount of events that are processed. We suggest to begin at the source, which is the event generation of Windows NT. That means to deciding which resources to audit for success, failure, or both. Remember, an event that is not generated produces no load to any system. We treat that topic in the following sections when we get specific for every resource type.

The TEC Windows NT Event Log Adapter has a feature to filter the events generated by Windows NT. That means that Windows NT still does its native auditing as before, but specified events in the adapter configuration file are not translated (less CPU load) and not sent to the TEC (less network and TEC load). As you can imagine, this is a very effective way of reducing load.

What would be the strategy to design this filter? We suggest identifying the events needed by the company (some of this information is covered in 2.3.12, “Auditing” on page 44) and block every other event from being translated and forwarded to the TEC. In the case of a problem (such as intruders or data loss), you can still view the Windows NT event Log. If you are only interested in security events being forwarded to the TEC, you can also start the event adapter specifying the security log (with the command `tecad_nt -L Security`).

The next point where you can reduce load is by reducing the size of events. You can reduce the event size by designing TEC classes that have dedicated slots for single short pieces of information. It does not make sense, for example, to map one string containing a tangled mass of information pieces to the *msg* slot of the *EVENT* class. You not only reduce the event size with a dedicated slot strategy, you also get a better event quality. You can run more meaningful queries or do automated reactions. And again, think of every single piece of information a Windows NT event gives you. If you think you do not need it for your purposes, do not include this slot in a class and do not map the field to another slot with the adapter’s configuration file.

In the sections 5.1.2, “Event Classes” and 5.1.3, “Format File”, we implemented all the pieces of information that Window NT provides for the events. This is because we do not know your implementation priorities or what this data will be used for. It is part of the process of writing the design document to determine that and adapt the definitions.

Here is an example: If we decided that we are not interested in the ID of a process that is associated with an object access, we would save 21 bytes per event (The additional string part of the event could be

“processID=2194120192;”). Let us assume that we have 50 resources in the computer environment that have auditing turned on for read and write accesses and that they have an average hit number of 2,000 per day. If you multiply that, you will come to a saving of around 2.1 Mb per day that need not be transferred over the network, not processed by the TEC, and not stored in the event database.

Therefore, take some time in deciding which events to generate, which to process by the adapter, and with what content.

5.2 Auditing Files and Directories

When auditing files and directories, you want to be informed of actions that pose a security risk. In certain cases, you want to log all accesses because you are interested in how frequently a resource is used (for statistical purposes). We recommend that, in general, you turn on auditing for sensitive data only, rather than auditing the access to publicly available information. Examples for sensitive date are accounting information, databases, development data, and mail data.

For high level security files and directories, we recommend failure and success auditing. For less sensitive resources, apply only unsuccessful accesses.

5.3 Auditing Logins

When auditing user logins to workstations and servers, we can monitor who was logged in and at what time. In case of any security problems, you can first create a list of who was logged on at the time of the breach. This helps a lot in reducing the work of searching for the account that is responsible for a certain action.

We recommend turning on the auditing for all logins and logoffs and forwarding that data to the Tivoli Enterprise Console.

5.4 Auditing Printers

Auditing printers (particularly for Windows NT, which has an NT:PRINTER resource) can help when planning the purchase of new equipment, designing printer pooling, and defining which users should print to which printer. You can see if a printer is used only rarely, or if it is overloaded.

You can make queries, for example, to find out which user submits the most, or the largest, print jobs.

If you are interested in these statistics, you need to turn on the printer auditing for successes and failures. If you are only interested in detecting unauthorized use attempts for the printers, you can restrict our auditing to failure events.

Appendix A. Planning Forms for Security Management Design

This Appendix lists some forms that may be useful as a basis for determining the environment of the company where a security management implementation is to be done. These forms represent a suggestion for taking a snapshot of the current security implementation. You may be able to use the forms as they are, or you may wish to adapt them to your needs.

A.1 Data Resource Information Form

This form can be used to collect information about specific data items (files, directories, UNIX file systems, and Windows NT shares) that need to be protected. See Table 11 on page 38 for an example of how this form might be used.

Table 32. Data Resource Attributes - Blank Form

Data Resource Attributes	Value
Resource name [1]	
Name of owning system [2]	
Resource type [3]	❑ File ❑ Directory ❑ Share
Operating system	❑ UNIX ❑ NT
Owner [4]	
Access [5]	
Audit	❑ Success ❑ Failure ❑ None
Responsible job function [6]	
Function / description [7]	

1. Name of the resource. Example: `/home/ausres47` (UNIX), `c:\winnt\system32` (NT).
2. Example: `rh2430b.itsc.austin.ibm.com` (TCP/IP host name), `rh2430c` (NetBIOS computer name). If this is a standard resource, such as a temporary directory, then use the type of machines (for example file server).
3. For the resource type Share, see description number seven.
4. The owner of the resource as reported by the file system. Example: `root` (UNIX), `Administrator` (NT).
5. Access rights, as reported by the file system. Example: `rwxr-x---` (UNIX), `Everyone Read(RX)/Administrators Full Control (All)`(NT).
6. Describe who is responsible for the resource and what job function.
7. Describe the purpose of the resource. This could be: accounting data, database containing employee data, user directory of user ausres46. If the resource is a Share on a Windows operating system, name the local resource here.

A.2 System Policy Information Forms

Use these forms to summarize what attributes are required for Login and Password policy. See 2.3.4, “System Policies” on page 31 for further information.

Table 33. Login Policy Attributes - Blank Form

Attribute	UNIX	NT
Suspend inactive accounts		n/a
Lock account upon multiple logon failures		
Limit grace logins		n/a
Limit concurrent logins		n/a

Use copies of Table 34 to record information about password policy attributes.

Table 34. Password Policy Attributes - Blank Form

Attribute	UNIX	NT
Minimum days between password changes	n/a	
Maximum days between password changes		
Minimum password length		
Minimum alphabetic characters		n/a
Minimum alphanumeric characters		n/a
Minimum numeric characters		n/a
Minimum uppercase characters		n/a
Minimum lowercase characters		n/a
Maximum repeated characters		n/a
Minimum special characters		n/a
Password history depth		

A.3 System Group Information Form

Use copies of this form to keep information about system groups that are in use. See 2.3.5, “Groups and Roles” on page 32 for more information.

Table 35. System Groups Attributes - Blank Form

Attributes of a User Group	Sample Values
Group name[1]	
Description[2]	
Name of the system on which it resides[3]	
Operating system and level[4]	
Responsible job role[5]	
Group members[6]	
User members[7]	

1 Name the group.
2 Describe the job function of the group members.
3 On what platform is the group defined.
4 Enter the operating system and version.
5 Identify the job role responsible for maintaining the group, such as assigning users.
6 In the case of Windows NT, a local group may have a global groups as members.
7 Users that are members of the group.

A.4 Network Connectivity Resource Information Forms

Use these forms for building network connectivity configuration data. See 2.3.7, “Network Connectivity” on page 40 for more information.

Table 36. TCP/IP Services Attributes - Blank Form

Attributes of TCP/IP Services	Values
Resource name[1]	
Description[2]	
Name of the system on which it resides[3]	
Operating system[4]	
Responsible job function[5]	
Host accessors[6]	

Attributes of TCP/IP Services	Values
[1] Enter the service name or port number defined in /etc/services or /etc/rpc. [2] Describe the TCP/IP service and explain the reason for using it. [3] The host name or IP address of the host. [4] Enter the operating system and version of the system. [5] Identify the job function responsible for maintaining this TCP/IP service. [6] List of hosts that may have access to the TCP/IP service described above.	

Use Table 37 to record data about the attributes of remote connection.

Table 37. Remote Connection Attributes - Blank Form

Attributes of Remote Connection	Values
Resource name[1]	
Description[2]	
Name of the system on which it resides[3]	
Operating system[4]	
Responsible job function[5]	
User group accessors[6]	
[1] Enter the host name that may be connected to/from the local host. [2] Explain to reason to connect to that remote host. [3] The host name or IP address of the local host. [4] Enter the operating system and version of the local host. [5] Identify the job function responsible for maintaining this remote connection. [6] List of user groups that may have access to the remote host described above.	

Appendix B. Special Notices

This publication is intended to help those involved at any stage of the implementation of a TME 10 Security Management installation. It is intended to help in the design, planning, and rollout of an installation. The information in this publication is not intended as the specification of any programming interfaces that are provided by the Tivoli Management Environment. See the PUBLICATIONS section of the IBM Programming Announcement for TME 10 Security Management for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program, or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, NY 10594 USA.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The information about non-IBM (vendor) products in this manual has been supplied by the vendor, and IBM assumes no responsibility for its accuracy or completeness. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate

them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any performance data contained in this document was determined in a controlled environment, and therefore, the results that may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environment.

This document contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples contain the names of individuals, companies, brands, and products. All of these names are fictitious, and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

Reference to PTF numbers that have not been released through the normal distribution process does not imply general availability. The purpose of including these reference numbers is to alert IBM customers to specific information relative to the implementation of the PTF when it becomes available to each customer according to the normal IBM PTF distribution process.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

AIX	IBM
OS/2	OS/390
RACF	

The following terms are trademarks of other companies:

C-bus is a trademark of Corollary, Inc.

Java and HotJava are trademarks of Sun Microsystems, Incorporated.

Microsoft, Windows, Windows NT, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

PC Direct is a trademark of Ziff Communications Company and is used by IBM Corporation under license.

Pentium, MMX, ProShare, LANDesk, and ActionMedia are trademarks or registered trademarks of Intel Corporation in the U.S. and other

countries.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Other company, product, and service names may be trademarks or service marks of others.

Appendix C. Related Publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

C.1 International Technical Support Organization Publications

For information on ordering these ITSO publications, see “How to Get ITSO Redbooks” on page 127.

- *Managing Access from Desktop to Datacenter: Introducing TME 10 Security Management*, SG24-2021
- *TME 10 Internals and Problem Determination*, SG24-2034
- *Getting Started with TME 10 User Administration*, SG24-2015
- Also look out for a new Redbook on *Managing the OS/390 Security Server with Tivoli* - due for publication late 1998.

C.2 Redbooks on CD-ROMs

Redbooks are also available on CD-ROMs. **Order a subscription** and receive updates 2-4 times a year at significant savings.

CD-ROM Title	Subscription Number	Collection Kit Number
Tivoli Redbooks Collection (HTML, PDF)	SBOF-6898	SK2T-8044
System/390 Redbooks Collection	SBOF-7201	SK2T-2177
Networking and Systems Management Redbooks Collection	SBOF-7370	SK2T-6022
Transaction Processing and Data Management Redbook	SBOF-7240	SK2T-8038
Lotus Redbook Collection	SBOF-6899	SK2T-8039
AS/400 Redbooks Collection	SBOF-7270	SK2T-2849
RS/6000 Redbooks Collection (HTML, BkMgr)	SBOF-7230	SK2T-8040
RS/6000 Redbooks Collection (PostScript)	SBOF-7205	SK2T-8041
RS/6000 Redbooks Collection (PDF Format)	SBOF-8700	SK2T-8043
Application Development Redbooks Collection	SBOF-7290	SK2T-8037

C.3 Other Publications

These publications are also relevant as further information sources:

- *Inside Windows NT Server 4*, New Riders, ISBN 1-56205-789-8

- *Windows NT 3.5 Guidelines for Security, Audit, and Control*, Microsoft Press, ISBN 1-55615-814-9
- *Securing the Open Client/Server distributed Enterprise*, IBM, SC28-8135
- *Practical UNIX & Internet Security*, O'Reilly,ISBN 1-56592-148

How to Get ITSO Redbooks

This section explains how both customers and IBM employees can find out about ITSO redbooks, CD-ROMs, workshops, and residencies. A form for ordering books and CD-ROMs is also provided.

This information was current at the time of publication, but is continually subject to change. The latest information may be found at `http://www.redbooks.ibm.com/`.

How IBM Employees Can Get ITSO Redbooks

Employees may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **Redbooks Web Site on the World Wide Web**

  `http://w3.itso.ibm.com/`

- **PUBORDER** – to order hardcopies in the United States
- **Tools Disks**

  To get LIST3820s of redbooks, type one of the following commands:

  ```
  TOOLCAT REDPRINT
  TOOLS SENDTO EHONE4 TOOLS2 REDPRINT GET SG24xxxx PACKAGE
  TOOLS SENDTO CANVM2 TOOLS REDPRINT GET SG24xxxx PACKAGE (Canadian users only)
  ```

  To get BokkManager BOOKs of redbooks, type the following command:

  ```
  TOOLCAT REDBOOKS
  ```

  To get lists of redbooks, type the following command:

  ```
  TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET ITSOCAT TXT
  ```

  To register for information on workshops, residencies, and redbooks, type the following command:

  ```
  TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ITSOREGI 1998
  ```

- **REDBOOKS Category on INEWS**
- **Online** – send orders to: USIB6FPL at IBMMAIL or DKIBMBSH at IBMMAIL

Redpieces

For information so current it is still in the process of being written, look at "Redpieces" on the Redbooks Web Site (`http://www.redbooks.ibm.com/redpieces.html`). Redpieces are redbooks in progress; not all redpieces become redbooks, and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

How Customers Can Get ITSO Redbooks

Customers may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **Online Orders** – send orders to:

	IBMMAIL	**Internet**
In United States	usib6fpl at ibmmail	usib6fpl@ibmmail.com
In Canada	caibmbkz at ibmmail	lmannix@vnet.ibm.com
Outside North America	dkibmbsh at ibmmail	bookshop@dk.ibm.com

- **Telephone Orders**

United States (toll free)	1-800-879-2755
Canada (toll free)	1-800-IBM-4YOU
Outside North America	(long distance charges apply)
(+45) 4810-1320 - Danish	(+45) 4810-1020 - German
(+45) 4810-1420 - Dutch	(+45) 4810-1620 - Italian
(+45) 4810-1540 - English	(+45) 4810-1270 - Norwegian
(+45) 4810-1670 - Finnish	(+45) 4810-1120 - Spanish
(+45) 4810-1220 - French	(+45) 4810-1170 - Swedish

- **Mail Orders** – send orders to:

IBM Publications	IBM Publications	IBM Direct Services
Publications Customer Support	144-4th Avenue, S.W.	Sortemosevej 21
P.O. Box 29570	Calgary, Alberta T2P 3N5	DK-3450 Allerød
Raleigh, NC 27626-0570	Canada	Denmark
USA		

- **Fax** – send orders to:

United States (toll free)	1-800-445-9269
Canada	1-800-267-4455
Outside North America	(+45) 48 14 2207 (long distance charge)

- **1-800-IBM-4FAX (United States)** or **(+1) 408 256 5422 (Outside USA)** – ask for:

  Index # 4421 Abstracts of new redbooks
  Index # 4422 IBM redbooks
  Index # 4420 Redbooks for last six months

- **On the World Wide Web**

Redbooks Web Site	http://www.redbooks.ibm.com
IBM Direct Publications Catalog	http://www.elink.ibmlink.ibm.com/pbl/pbl

Redpieces

For information so current it is still in the process of being written, look at "Redpieces" on the Redbooks Web Site (`http://www.redbooks.ibm.com/redpieces.html`). Redpieces are redbooks in progress; not all redbooks become redpieces, and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

IBM Redbook Order Form

Please send me the following:

Title	Order Number	Quantity

First name	Last name	
Company		
Address		
City	Postal code	Country
Telephone number	Telefax number	VAT number

☐ Invoice to customer number _____

☐ Credit card number _____

Credit card expiration date	Card issued to	Signature

We accept American Express, Diners, Eurocard, Master Card, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.

List of Abbreviations

ACE	Access Control Entry
ACL	Access Control List
ACP	(logfile) Adapter Configuration Profile
AIX	Advanced Interactive eXecutive (IBM UNIX)
CLI	Command Line Interface
GUI	Graphical User Interface
IBM	International Business Machines Corporation
ITSO	International Technical Support Organization
LAN	Local Area Network
LCF	Lightweight Client Framework - Now known as Tivoli Management Agent
NetBEUI	NetBIOS Extended User Interface
NetBIOS	Network Basic Input Output System
OID	Object Identifier
PROFS	Professional Office System
RFC	Request For Comments
RPC	Remote Procedure Calls
TACF	Tivoli Access Control Facility
TCP/IP	Transmission Control Protocol/Internet Protocol
TEC	Tivoli Enterprise Console
TMR	Tivoli Management Region
WAN	Wide Area Network

Index

Symbols

A

B

C

D

E

F

I

L

N

O

P

ITSO Redbook Evaluation

Tivoli Security Management Design Guide
SG24-5101-00

Your feedback is very important to help us maintain the quality of ITSO redbooks. **Please complete this questionnaire and return it using one of the following methods:**

- Use the online evaluation form found at http://www.redbooks.ibm.com
- Fax this form to: USA International Access Code + 1 914 432 8264
- Send your comments in an Internet note to redbook@us.ibm.com

Which of the following best describes you?
_ **Customer** _ **Business Partner** _ **Solution Developer** _ **IBM employee**
_ **None of the above**

Please rate your overall satisfaction with this book using the scale:
(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)

Overall Satisfaction _____

Please answer the following questions:

Was this redbook published in time for your needs? Yes___ No___

If no, please explain:

_____

_____

_____

_____

What other redbooks would you like to see published?

_____

_____

_____

Comments/Suggestions: (THANK YOU FOR YOUR FEEDBACK!)

_____

_____

_____

_____

SG24-5101-00
Printed in the U.S.A.

Tivoli Security Management Design Guide

SG24-5101-00