6

Solaris Security

BUSINESSES AROUND THE WORLD ARE MAKING as much use as possible of Internet technologies to break into new global markets. As companies become ever more dependent on the Internet for their business relationships, the importance of computer security increases as well. More and more people and businesses are connecting every day, and with that comes more potential for security threats.

Single-site, centralized data centers, with equally centralized security requirements, are fast disappearing and are being replaced by new, modern, distributed corporations with sites not just in different towns and cities, but in different countries as well, all communicating together and sharing confidential company information. It is a system cracker's paradise if there is no security policy in place—they can just march in and wreak havoc.

Hacker or Cracker

The term *hacker* is usually referred to in the news as an external malicious attacker of computer systems. Within the IT industry however, the term *cracker* is used to define the malicious attacker or virus perpetrator. A *hacker* is someone who is not malicious but who, through unsupervised access, can often cause unintentional damage to a computer system, normally because of inexperience rather than a deliberate action. That distinction between the two concepts is used in this text.

But it's not the external cracker who actually causes the most damage; those instances are merely the most widely reported. Perhaps surprisingly, various studies have shown that by far the largest cause of security incidents comes from within—that is, someone already authorized to use the computing resources of a company, such as a disgruntled employee who was passed over for promotion or was refused a raise. This employee could do something as simple as delete important files, for example, but a more sophisticated employee could write and introduce a destructive virus into the system. The possibilities are many.

The Computer Crime and Security Survey, carried out each year in the United States by the Computer Security Institute/Federal Bureau of Investigation (CSI/FBI), analyzes and highlights the effect of computer crime/abuse. The findings clearly show a marked increase each year. To put this into perspective, Figure 6.1 shows the average financial loss suffered for the last four years as a result of computer crime. The average loss is taken from the number of companies that were able to put a dollar value on the loss encountered.

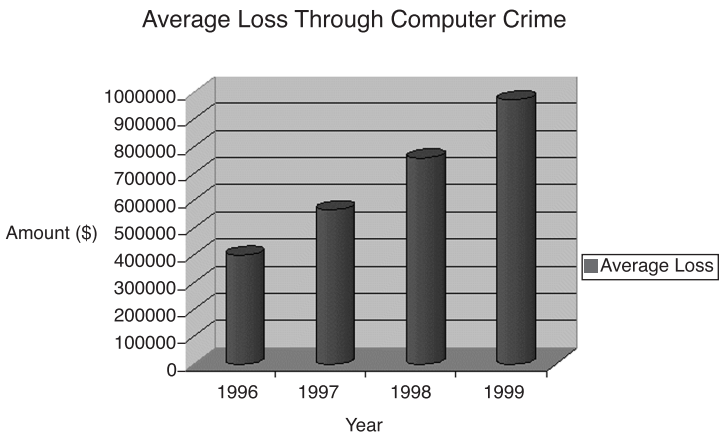

Figure 6.1 The disturbing upward trend toward computer crime highlights the need for increased computer security and incident reporting.

The system manager is responsible for maintaining the integrity of the systems that he manages and for ensuring that there is no unauthorized access to confidential data. In larger corporations, the system manager might be assisted by a dedicated computer security section, whose role is to implement a company security policy. Sadly, a large number of businesses do not have a corporate policy, and various departments implement their own ad hoc computer security, often lacking consistency and the capability to ensure that it is enforced.

This chapter aims to address some of the issues surrounding the need for better security in a Solaris environment, and it takes a brief look at some of the security products that are available—both with the Solaris operating environment and also from the public domain.

Strategic Planning and Techniques

Computer security is a strategic issue, although some tactical measures can be taken on a per-system basis; these are discussed at the end of the chapter. Computer security requirements should be addressed as part of the IT strategy and should be budgeted for accordingly. Too little value is placed on security, probably because there is no "tangible" benefit, or return from investing in it, unlike the launch of a new product. The threat from both inside and outside the company is very real and is growing. A company should address the issue from the perspective of not being able to carry out business. For example, suppose that an external cracker penetrates the system and trashes the customer and orders databases, or perhaps a malicious e-mail is sent to each of the customers in the database stating that the company is no longer trading, and recommending a competitor instead. The losses incurred could undoubtedly cause bankruptcy in some cases.

A good computer security strategy needs to be initiated by professionals who have experience in dealing with such an important issue. When the strategy has been devised and implemented, it needs to be regularly assessed to see if any modifications or enhancements are required. Each version of the Solaris operating environment is becoming more secure, providing a good reason for carrying out upgrades as soon as it is practical to do so, in line with the corporate IT policy regarding the stability of products (refer back to Chapter 5, "Solaris Installation and Upgrades").

This section discusses how the strategic security policy can be implemented in the Solaris environment, and examines a number of security products that are available to assist the system manager in securing the systems under his control. The products include those shipped with the Solaris operating environment as well as unbundled security packages from Sun Microsystems and others that are available in the public domain.

System Security Policy

When dealing with security issues in a network environment, it is the security policy that pulls everything together and identifies how the security of a network and the systems contained within it is managed.

Most companies implement a security policy, some on a per-system basis and others on a corporate basis. Larger organizations use a dedicated computer security section to manage the entire security issue, taking a perspective from a much higher level. The policy is then implemented for each system that is installed and connected to the network—indeed, a condition of connecting to the corporate network is that compliance to the security policy is demonstrated and verified. The system manager must ensure that security issues are taken into account when implementing a new system. He also serves as a regular liaison with the computer security section.

The development of a good security policy, however, is not solely about protecting the systems at all costs. It needs to be a careful balance between the security itself and the restrictions that it imposes on running the business. The user community must be

considered as part of the bigger picture, particularly the effect and inconvenience that the security measures have on them. Of course, the best security policy is to deny everything to everyone, but this is not a practical solution. The system manager (and the computer security section, if applicable) must take into account the needs of the business and the users to see if the proposed policy is more of a dictatorship than a preventive measure. After all, the users are carrying out the core business functions, and they will often provide a valuable insight into what is practical and realistic and what is unworkable. It is no use having a brilliant security policy if the company cannot function as a result.

There are a number of important steps to take in the formulation of a security policy; these are discussed briefly here:

- **Identify what needs to be protected**—This includes but is not limited to hardware, software, data, and documentation.
- **Carry out a risk assessment**—Each item identified in the previous section should be examined to see the associated risks and then should be ranked by the level of severity attached.
- **Apply weightings**—After assembling a list of resources that require protection, each item needs to have a weighting associated with it so that it can be prioritized as to its importance. This, in turn, defines how much is spent on protecting the resource.
- **Keep the policy acceptable and realistic**—Despite the risks and severity, the security policy needs to be workable and acceptable. There is no point in having a stringent security policy that stops the company from carrying out its business in the most effective manner. The cost-effectiveness of a security policy can often be determined by the effect that it has on the operation of the business.
- **Establish a violation procedure**—While trying to protect the resources, a security policy also has to consider what to do if the network or system is breached. The type of incident reporting, the level of auditing, and the escalation procedure for when to inform the authorities are all examples of things to be addressed here.
- **Educate users**—One of the best ways to protect the resources is through educating the users in matters relating to security. A formal training program coupled with regular bulletins keeps the user aware of security considerations. It also demonstrates to the users that the company takes security seriously.

Unauthorized User Warning

Whenever a user logs in to a computer system, a statement should be displayed that says something like, "Only authorized users can use this system." Otherwise, you are providing an intruder with an escape route when prosecutions start—on the grounds that it never said he couldn't or shouldn't have access!

A more local issue for the system manager to address as part of the security policy is that of managing the access for internal users, or employees of the company. It has to be recognized that in any computer network environment, there must be an element of trust; otherwise, nobody would be allowed to do anything, and the business would not function. Consider, for example, how you would protect against a corrupt system administrator with full superuser privileges. The answer is that, realistically, this is virtually impossible, which is where the element of trust comes in. For the rest of the users of the system, though, the system manager must analyze the potential risks to the security of the system (and the data) and determine how to provide them with sufficient access to carry out their jobs without compromising the system. The system manager needs to ask himself a few fundamental questions about each user (or group of users):

- Does the user need to have command-line access (a shell)?
- Does the user have potential access to sensitive data?
- Does the user require any special privileges?

These questions are deliberately broad in their scope, and companies should be able to ask more direct questions depending on the type of business being carried out. Still, these cover the major risks that a user could potentially gain access to more than they are legitimately entitled. The system manager could consider three general solutions: a restricted shell, captive accounts, or nonprivileged user accounts with enhanced auditing facilities. Each of these is discussed briefly here:

- **The restricted shell**—This option is realistically useful only when a user carries out an extremely limited function, such as logging in to collect or deliver data. Using the restricted shell, there can be no traversing of the system directory structure. The command set that is available is very limited, too.
- **Captive accounts**—For users who run specific applications, this is a more popular option. For example, the sales staff might use the system to access the sales and customer databases. In this case, when logging in to the system, the user is taken straight into the application, with no opportunity to enter commands at the prompt. When the session is finished and the user quits the application, he is immediately logged off.
- **Nonprivileged accounts with auditing**—When access to the command line is required, the best solution is to provide a nonprivileged account, reducing the amount of damage that the user can cause. For functions requiring greater access to the system or enhanced privilege status, the use of software such as Sudo provides the necessary privilege while also limiting the access and auditing everything that is done. (Sudo is discussed in detail later in this chapter in the section "Products Available in the Public Domain.")

Shared User Accounts
Never allow users to share user accounts. Every user of a system must always be assigned a unique account that is to be used only by that person. As soon as users are allowed to share logins, accountability is diminished and the security is compromised.

For further references on security issues, two excellent sites on the Internet provide regular bulletins and newsletters, as well as a wealth of security-related information:

- The SANS Institute, at `http://www.sans.org`
- CERT Coordination Center (originally called the Computer Emergency Response Team), at `http://www.cert.org`

See also the Appendix, "Resources," for recommended further reading and other Internet sites relating to Solaris security.

Out-of-Hours Policy

When computer systems are running continuously, 24 hours a day, 7 days a week, there needs to be a policy in force for the hours when offices are unattended. Consider the scenario in which Sun workstations configured as diskless clients are left in open office space. If an intruder can gain physical access to the client, then it can be turned on. It will automatically boot across the network and end up displaying the login prompt. Meanwhile, the intruder, watching the boot sequence, can gain all sorts of information concerning the network setup, such as the hostname, the IP address, the Ethernet address, the daemon processes that are running, and so on. If left long enough, he may be able to log in!

One way of stopping this with clients that boot across the network is to disable them from the server. This is accomplished simply by renaming a single file in the directory /tftpboot on the operating system server that provides the client's resources. The client's boot filename is made up of the hexadecimal equivalent of its IP address and the kernel architecture. For example, a Sparc Ultra client with an IP address of 210.127.8.10 would have a file named D27F080A.sun4u. By appending, say, .old to the filename, it will fail to boot. When the file is renamed back to its original name, the boot will succeed.

Setting such a policy is simple to implement and can be done automatically through the use of a shell script, even using cron to schedule its execution at the required times.

Firewalls

Most businesses allowing external connections make use of firewalls. Many suppliers produce firewall hardware and software, so it is perhaps suitable that a short discussion of the concept of firewalls appears here.

There is a point where the company's private network and an external network, such as the Internet, meet. This is the point at which a firewall is inserted. A firewall system consists of a hardware and software configuration (collectively known as a firewall). It resides at this point and controls access both into and out of the company's network, much like a security guard at the entrance to a company's premises.

A firewall need not be used only for external connections; it can also be implemented to restrict access to specific systems within the company, such as the accounting system. In this instance, the company might want to separate this system from others on the network and allow only a specified list of users to access the system.

Sun's Firewall-1

Sun Microsystems produced a product called Firewall-1 which has now been retired and incorporated into the Solstice Sunscreen security product. Solstice Sunscreen is discussed later in the "Unbundled Products" section.

There are two main types of firewalls—a packet-filtering firewall and an application firewall (also known as a proxy server). Both are described in the following sections.

Packet-Filtering Firewall

This kind of firewall "filters" packets that it receives according to a predefined set of rules, allowing only packets that match the rule criteria to proceed through to the company network. Two slightly different policies might be applied to this kind of firewall, either to allow all packets unless specifically denied by the rule set, or to deny all packets unless specifically allowed by the rule set. The choice will depend on the overall security policy for the company, but obviously, the latter option is more secure.

A packet-filtering firewall offers less protection than an application-level firewall, but it is cheaper and might be appropriate for a low-risk environment. It usually doesn't support authentication, nor does it have any reasonable logging facility. Packet filtering can be implemented on the Internet router(s) or, if a Sun system is connected to the Internet, by using the freely available package ipf. For further reference on this package, see the Appendix, "Resources."

Application-Level Firewall

This is a much more sophisticated kind of firewall, demanding a separate system to run server programs (called proxies). These programs receive external requests and analyze them for authenticity, forwarding only legitimate requests to the internal network.

A major distinction between the two kinds of firewall described here is that the application-level firewall supports full authentication and logging. It is a more secure option than a packet-filtering firewall and is suitable for medium-high-risk environments. It also has the capability to "hide" the name and address of any computer communicating through the firewall. This means, for example, that the IP address of a computer is concealed from any potential intruder "listening" on the other side; all communications "appear" to have originated from the firewall.

Creating a DMZ

You can combine packet-filtering and application-level firewalls in series, one after the other. This has the advantage of increased protection from a greater variety of attacks. If you then take this one step further, using two packet-filtering routers and an application-level firewall in the middle, the most secure firewall can be created through the use of a *demilitarized zone* (DMZ). This creates a separate, isolated network between the private company network and the external network (the Internet).

An advantage of this type of setup is that direct transmission across the DMZ is prohibited, denying any attacker direct access to the company network. It is in the DMZ that a company might place public data or a download area because external users can still legitimately access the publicly available company information, but the private company network is protected, while still retaining the capability to communicate on the Internet.

Security Products

This section provides an insight into some of the security products that are available for Solaris. It is not intended to be an in-depth discussion; it will serve more to raise awareness that will prompt system managers to investigate products of interest more thoroughly. Some additional resources are shown in the Appendix. This section covers three main aspects: those that come as part of the Solaris operating environment, those that are provided by Sun Microsystems as unbundled products, and, finally, some products that are available from third-party vendors or in the public domain.

Bundled with Solaris

The first list comprises those that come bundled with the Solaris operating environment and that are available immediately following the installation of the operating environment. An advantage that these products possess is that no additional configuration of the system is required to make use of them right away.

Password Management Features

Solaris, by default, provides a number of password management features. These are discussed briefly here:

- **Validation of passwords**—The password entered by a user at login is compared with the password stored in the file /etc/shadow; see the following bullet "Shadow password file" for an explanation of this file. If the two match, then the user is allowed to proceed with the login.
- **Aging of passwords**—The system administrator can force the user to change his password after a specified period of time, warn the user that a change is imminent, and prohibit him from immediately changing it back to the old password. When selecting a new password, the user also is prevented from using a previously selected password. The mechanism ensures that a newly selected password meets the specified criteria—that is, the correct number or characters or symbols.

- **Shadow password file**—Earlier implementations of SunOS contained the encrypted password as part of the file /etc/passwd, so it was visible to non-privileged users. The file /etc/shadow now contains the encrypted passwords for users and is also a hidden file, which is readable only by the root user.
- **Expiration of user accounts**—The system administrator can set the expiration date of a user account. This feature automatically disables the user account when the expiration date is reached. One of the most frequently overlooked password administration duties is disabling user accounts when a member of staff leaves the company or no longer requires access to the system. This feature goes some way to closing this potential loophole.

Automatic Security Enhancement Tool (ASET)

ASET is a set of utilities that can be used by system administrators to check the basic security of the Solaris system. It provides warnings where potential security loopholes are detected and, depending on the level of security selected—low, medium, or high—makes corrections.

When you run ASET, a number of reports are generated, which you can inspect to see the results. Details of any actions taken can also be found here.

As an example, I ran ASET on a Solaris 7 machine with deliberate security holes. The results are shown in Listing 6.1 as a concatenation of all of the *.rpt files in the directory /usr/aset/reports/latest.

Listing 6.1 **The Output Produced from Running the ASET Command**

```
#cat *.rpt
*** Begin Checklist Task ***

No checklist master - comparison not performed.
... Checklist master is being created now. Wait ...
... Checklist master created.

*** End Checklist Task ***

*** Begin EEPROM Check ***

EEPROM security option currently set to "none".
<< There is no security on the boot prom. This
                                                          Should at
least be set to 'command'.
*** End EEPROM Check ***

*** Begin Environment Check ***

Warning! umask set to umask 022 in /.profile - not recommended.
      << should be set to 077 so that the default is no
                                                          permission for
```

continues

Listing 6.1 **Continued**

```
anyone other than the owner
Warning! umask set to umask 022 in /etc/profile - not recommended.

Warning! "." is in path variable!
<< path variable is open to malicious attack
Check /.profile file.

*** End Environment Check ***

*** Begin Firewall Task ***

Task skipped for security levels other than high.

*** Begin System Scripts Check ***

Warning! /etc/hosts.equiv contains a line with a single +
<< the system trusts all remote hosts
This makes every known host a trusted host, and is therefore
not recommended for system security.

Warning! The use of /.rhosts file is not recommended for system security.
      << If this has to be used, keep it at individual
                                                                  user level,
certainly not in the root directory.
Warning! Shared resources file (/etc/dfs/dfstab) , line 12, file system exported
with no restrictions:
      share -F nfs /usr/opt/oracle                                          <<
blindly sharing file systems as read/write

*** End System Scripts Check ***

*** Begin Tune Task ***

... setting attributes on the system objects defined in
    /usr/aset/masters/tune.low

*** End Tune Task ***

*** Begin User And Group Checking ***

Checking /etc/passwd ...

Checking /etc/shadow ...

Warning!  Shadow file, line 15, no password:
<< Users with no passwords assigned
      john::11146::::::
```

```
Warning!  Shadow file, line 16, no password:
      bill::::::::

Warning!  Shadow file, line 17, no password:
      frank::::::::

... end user check.

Checking /etc/group ...

... end group check.

*** End User And Group Checking ***
```

The example output shows that significant security holes have been found, and these have been annotated accordingly on the text, but the system has not attempted to modify any of them. This is because, by default, the security level is set to low. Experienced system administrators can modify the configuration for each of the security levels to suit their own installations. The relevant files can be found in the directory /usr/aset/masters.

Access Control Lists

The standard UNIX file and directory permissions allow three categories of access to be specified: the owner of the file or directory, the group, and everyone else (known as "other"). Sometimes this is insufficient, particularly when individual, specific access is required. Consider the following example:

The system manager (user: john) is compiling a management report file named monthly, but he requires input from another manager, the network manager (user: bill). The file is owned by user john and is group-readable by the group managers. The requirement is for the other manager to be able to write to the file, but some managers in the group should not be able to write to it, so the option of allowing group write permissions is not acceptable. Of course, one alternative is to give the user bill his own copy of the file. This is inefficient, though, because it creates multiple copies of the same file, both of which will end up being different.

The access control list (ACL) provides the answer to the problem. ACLs provide a finer level of file and directory access permission and easily solve the problem outlined. User john can now give explicit write access to user bill in the form of an ACL, as shown here.

The original listing shows the permissions as originally stated:

```
$ ls -la monthly
-rwxr-----  1 john     managers 2872175 Jul  8 15:24 monthly
$
```

User john executes the command to give user bill the required access, to be able to write to the file:

```
$setfacl -s user:bill:rw-,user::rwx,group::r--,other:---,mask:rw- monthly
$
```

The listing now shows a "+" at the end of the permissions listing, indicating that an ACL is in force:

```
$ ls -la monthly
-rwxr-----+  1 john     managers 2872175 Jul  8 15:24 monthly
$
```

By executing the command `getfacl`, the precise permissions can be displayed, as shown in Listing 6.2.

Listing 6.2 **Sample Output Showing the Full ACL for a Given File**

```
$ getfacl monthly

# file: monthly
# owner: john
# group: managers
user::rwx
user:bill:rw-           #effective:rw-
group::r--              #effective:r--
mask:rwx
other:---
```

ACLs can be extremely useful when explicit, limited access is provided to a restricted subset of users, while retaining the desired level of security. The ACL facility is automatically included when Solaris is installed, so it can be used immediately without any further configuration necessary.

Pluggable Authentication Module (PAM)

The PAM software comprises several authentication modules that are dynamically loaded—that is, loaded into the kernel as they are required. One of the benefits of PAM is that the system manager can implement a separate authentication process for each of a number of services (login, su, and telnet being good examples). In this way, a more extensive security authentication procedure can be put in place, depending on the perceived security risk that the service displays. Also, the authentication of a user can be covered by more than one method and in a flexible order. For example, consider the following sample lines from the PAM configuration file /etc/pam.conf:

```
su      auth  requisite   /usr/lib/security/pam_inhouse.so.1
su      auth  required    /usr/lib/security/pam_unix.so.1
login   auth  required    /usr/lib/security/pam_unix.so.1
login   auth  optional    /usr/lib/security/pam_inhouse.so.1
```

Notice, for example, that the `su` command is authenticated twice, once using an in-house authentication method and then also using the standard UNIX authentication. The third column defines the severity of the authentication, so again for `su`, if the requisite authentication fails, then the other authentication will not even be carried out, and the `su` request will fail.

A final advantage of the PAM feature is that further modules can be plugged in and configured without needing to modify the applications.

SunShield Basic Security Module (BSM)

The Basic security module is enabled by a simple script. It makes a modification to the kernel, which disables the STOP-A key combination (used to halt a running system), disables the volume management facility, and installs a full auditing facility that is highly configurable, depending on the level of audit required. The system manager can choose to audit functions, such as deletion of files, and also users. BSM includes utilities that enable the audit trail data to be analyzed, even generating reports of events that have been logged.

Figure 6.2 shows the default audit_class file, which displays the standard auditing categories that can be used immediately.

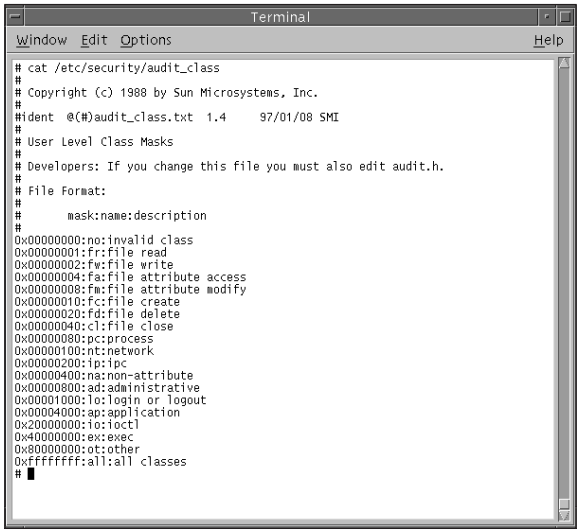

Figure 6.2 The categories available for auditing allow the majority of events to be captured, producing full accountability. Auditing is highly configurable for specific requirements.

As an example, suppose that the Oracle database has crashed. Upon initial investigation, it is found that the main tablespace data file is missing—this is called system.dbf and is an essential part of the database. By interrogating the audit logs, which contain the attribute `fd`, it is possible to ascertain exactly what happened. The record in question is reproduced here, with the fields of interest highlighted in bold:

```
header,129,2,unlink(2),,Sat Jul 15 00:45:59 2000, + 509999500
➥msec,path,/usr/opt/oracle/oracle/system.dbf,attribute,100600,oracle,dba,838
➥8621,25540,0,subject,jephilc,root,other,root,other,701,350,0 0
➥aries,return,success,0
```

The scenario goes like this:

- The file in question was /usr/opt/oracle/oracle/system.dbf.
- It was owned by user oracle and group dba.
- The operation carried out was `unlink`, which is the `rm` command, executed on Saturday, July 15, 2000, at 00:45:59, on the host named `aries`.
- The user in question turned out to be `jephilc`, who had managed to become the superuser root.
- The result of the operation was `success`—it worked, and the file was successfully deleted.

This example demonstrates how useful the auditing facility can be, particularly when considering that the majority of security incidents are actually carried out by members of staff within the company, as mentioned in the introductory paragraphs of this chapter.

> **Audit Only as Needed**
>
> Be very careful configuring the file audit_user, where the categories for auditing the users are selected. If `all` is selected, then vast amounts of disk space will be consumed—approximately 20–30Mb when a user logs in and starts up the window system. If the system does not have a separate /var file system, then there is a risk that the root (/) file system will fill to capacity.

Secure Network File System (NFS)

Sharing file systems across the network is potentially dangerous, particularly when shared globally with no restrictions. Solaris provides a mechanism for using enhanced authentication of users when trying to mount file systems across the network using NFS. An option to the `share` command is included, `sec=mode`. Currently, Solaris provides support for three modes (the mode none is not supported for NFS mounts): sys, dh, and krb4. Each is briefly discussed here:

- **sys**—This is the default authentication method used by Solaris. It uses the AUTH_SYS authentication, in which the user ID and group ID are checked by the NFS server before allowing the mount to proceed.
- **dh**—This is the Diffie-Hellman public key encryption system. This is a standard public key system that creates a secret key between two hosts.
- **krb4**—This method uses the Kerberos version 4 system for encryption and is freely available from the Massachusetts Institute of Technology (MIT). It is also available as a product from several different vendors. Kerberos is described later in this chapter in the section "Products Available in the Public Domain."

Unbundled Products

The security products listed in the next sections are available from Sun Microsystems but are not part of the Solaris operating environment distribution. They must be purchased as separate products and installed as packages. The list is not exhaustive, but it is intended to give you good insight into the products and facilities that can be implemented if a company requires additional security functionality to that provided by the standard Solaris release.

Trusted Solaris

This is a security-enhanced implementation of the Solaris operating environment that provides a fully configurable security policy that is incorporated within the software. Trusted Solaris extends the security that is normally provided with UNIX, such as sensitivity labels. These determine a level of security that a user possesses when logging in to the system and, in turn, can restrict the access available to the user.

Trusted Solaris also limits the use of the root user in traditional UNIX systems. Instead, there are other user accounts, such as secadmin, the security administrator, and admin, the system administrator, who sets up, for example, nonsecurity-related portions of user accounts.

Further information on Trusted Solaris can be found on Sun's Web site or gained from your Sun sales department.

Sunscreen Secure Net

Solstice Sunscreen is a package that provides the combined functionality of a firewall along with network-level authentication, also known as Simple Key management for IP (SKIP).

Sunscreen consists of two main components, a screen and an administration station. The screen component carries out the firewall functionality, screening packets and performing the necessary encryption and decryption, while the administration component is carried out by the administration station, based on configured security policies.

Sunscreen employs a stealth architecture that allows companies to effectively set up a virtual secure private network across a public network, such as the Internet. The stealth aspect of this product relates to the fact that the screen can be configured so that it cannot be accessed using an IP address. Therefore, it can pass packets across the network without recording any indication that it ever existed because there is no IP address. Because of this, a potential intruder cannot access the machine running Sunscreen, which makes it very difficult for it to be attacked. For internal network connections, the stealth mode need not be used, so a second mode, routing, is also available, which is much faster and would normally be used to segment the network internally within the company. The stealth mode screen would be used for the external perimeter—that is, the point where communications meet the outside world, the Internet.

Figure 6.3 displays a company network employing the use of Sunscreen Secure Net.

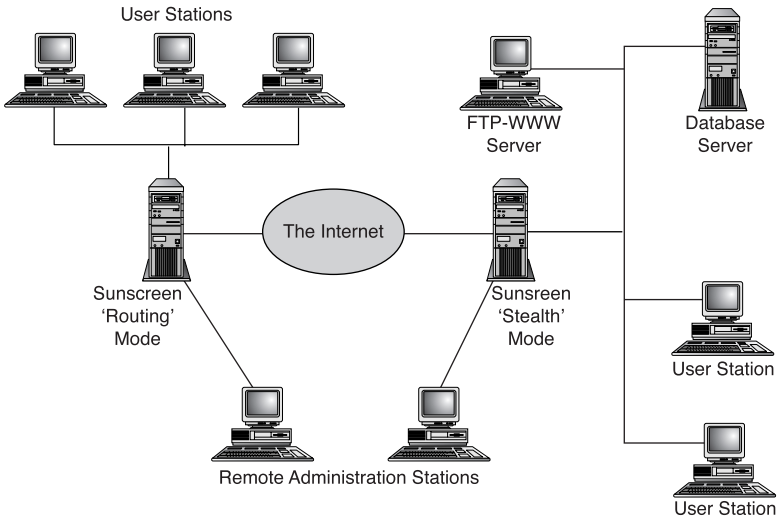

Figure 6.3 The flexibility of Sunscreen is demonstrated by the added functionality of remote administration.

Sun Security Manager (SSM)

SSM provides the type of security that corporations require when running mission-critical applications in a distributed environment. It delivers advanced access control for clients, servers and applications, as well as a centralized management facility.

One interesting aspect of SSM is that it supports *Secure Single Sign On (SSSO)*. This facility is important in today's multi-system distributed networks. Users are being forced to remember more passwords as they have access to more systems and applications. A consequence of this is that there is a higher probability that they will be written down or made easy to remember, which makes them less secure. The capability to use a single password for all access means that the user can remember the password, and the fact that it may have to change regularly is not a major issue.

Sun Security Manager also provides support for high availability in a 24×7 environment. A master server holds the security database, while a number of slave (redundant) servers hold read-only copies of the security database. Any of the security servers can respond to requests, and updates are automatically distributed to the slaves when made to the master database. If the master security server fails, then any of the slave serves can easily be converted into a master.

Products Available in the Public Domain

The following products are available in the public domain from a variety of sources. Refer to the Appendix for a list of the more popular sites where these products can be obtained.

Sudo

Sudo is a freely available software package that provides the facility for specific users (or defined groups of users) to run either some or all commands as the superuser (root). Additionally, it can be configured so that commands are run as another user. All commands and their arguments are written via the syslog process, which can be regularly audited. This can be modified so that all Sudo messages are written to a specific file that can be closely monitored. Configuration of syslog for such purposes is discussed in more detail in Chapter 9, "Tactical Management," in the section "System Logging."

Sudo presents a number of advantages for the system manager:

- It is an ideal way to give enhanced access to specified users for a particular purpose.
- Limited superuser (root) access can be granted without having to reveal the superuser password, which would allow blanket access to the whole system.
- The audit logs can be used to track privileged command usage and hence see what was done, by whom, and when it was done, providing increased accountability.
- It is an excellent tool for junior system administrators who are learning the job because it eliminates the risk of them making an error while logged on as user root.
- A double-check is made when users run Sudo commands. First, a user is required to provide his own password when using Sudo commands. After the user is authenticated, no password is required unless a Sudo command is not entered for 5 minutes (although this interval is also configurable). As a second check, the program verifies that the user is allowed to run the required program.
- Sudo is simple and effective, and its use can be easily modified through the use of a single configuration file, the sudoers file.

Control of privileged commands is carried out through the use of the sudoers file, which is installed by default into the directory /usr/local/etc, although it is a good idea to configure the installation to place this file in the /etc/ directory. A sample sudoers file is shown in Listing 6.3:

Listing 6.3 **A Sample Configuration of the sudoers File, Containing a Fundamental Flaw**

```
# sudoers file.

#


# This file MUST be edited with the 'visudo' command as root.
#
```

continues

Listing 6.3 **Continued**

```
# See the man page for the details on how to write a sudoers file.
#

# Host alias specification

# User alias specification
User Alias  OPS=jephilc,operator,frank,bill
User Alias  DBA=jephilc,oracle
User Alias  SADM=jephilc

# Cmnd alias specification
Cmnd Alias  ADMIN DB=/app/oracle/product/8.1.6/bin/svrmgr1
Cmnd Alias  EDITHOSTS=/usr/bin/vi /etc/inet/hosts
Cmnd Alias  LISTALL=/usr/bin/ls
Cmnd Alias  KILL=/usr/bin/kill
Cmnd Alias  BACKUPS=/usr/sbin/ufsdump,/usr/sbin/ufsrestore
Cmnd Alias  MOUNT=sbin/mount,/sbin/umount
Cmnd Alias  SHUTDOWN=/sbin/shutdown

# User privilege specification
root  ALL=(ALL) ALL
SADM  ALL=(ALL) ALL
Jephilc     ALL=EDITHOSTS
DBA   ALL=ADMIN DB
OPS   ALL=SHUTDOWN,KILL,MOUNT,BACKUPS
```

This file shows that several user aliases have been set up along with a number of command aliases. The section of the file headed "User privilege specification" dictates exactly who can run what.

The listing declares that there is a fundamental flaw in the file, namely the line `Cmnd_Alias EDITHOSTS=/usr/bin/vi /etc/inet/hosts`. By providing access to the vi editor, the whole system has been left wide open because vi has a shell escape option. In this case, user `jephilc` could gain access to the command line as user root and have full superuser access. For this reason, the administrator of the sudoers file must be extremely careful when considering the entries to be added and also determining who will be allowed to use them.

Kerberos

Kerberos is a network authentication system that allows entities communicating over networks to prove their identity. It also prevents eavesdropping by potential intruders because the communications are encrypted.

Kerberos works by providing users or services (known as principals) with tickets that are used to identify themselves to other principals. It also provides secret cryptographic keys for secure communication with other principals.

When a user wants to use a service, these steps are followed (they would normally be carried out as a script or as part of the service requests, not manually):

1. The user runs kinit and requests a ticket-granting ticket (TGT) and a session key from the Key Distribution Service (KDS), which is the Kerberos server.
2. KDS receives the request and returns an encrypted TGT and the session key, which are encrypted using the user's password. The user enters the password to decrypt the TGT and the session key.
3. User now runs klogin, which requests a service ticket from the Ticket Granting Service (TGS). The user sends the TGT received from KDS, the session key, and a service request.
4. The TGS authenticates the request by examining the session key. The TGS then sends back a service ticket and a service session key to the user, who decrypts them both.
5. The user sends the service request to the provider of the service, along with the service ticket obtained from the TGS and the authenticator, similar to a reference, proving that he is who he says he is.
6. The service provider decrypts the TGS ticket and authenticates the request. The service now starts.

Figure 6.4 shows the kerberos authentication process in action. The numbers on the diagram refer to the steps in the process just listed.

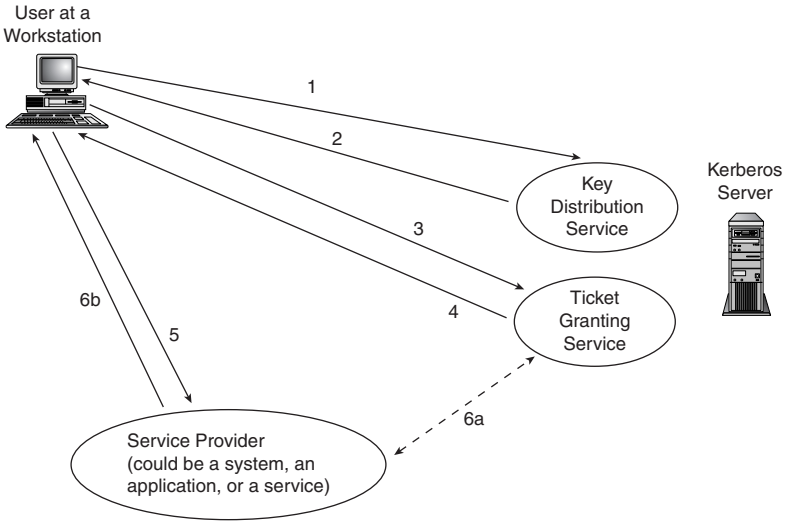

Figure 6.4 The authentication process of kerberos is similar to obtaining a reference from a trusted person known to both parties.

Crack

A number of system administrators and system managers are reluctant to use tools such as Crack because they are utilized by crackers to try to gain unauthorized entry to computer systems. Crack attempts to obtain passwords by comparing them to dictionary words. It has been stated that more than 30% of the average password file entries can be broken the first time this program is run.

If the system manager arranges for the program to be run against password files, then weaknesses can be identified and rectified, possibly before the cracker tries. The old saying, "Forewarned is forearmed" comes to mind, and in this case, it is particularly true.

Crack is successful only on systems in which the encrypted password is held in the file /etc/passwd along with all of the other user details. Solaris 2 versions utilize the shadow password entry, which is secure and beyond the reach of programs such as Crack. However, older Solaris systems running Solaris 1.x are at risk from this program, as are systems that are part of an NIS domain.

SATAN

SATAN is an acronym for Security Analysis Tool for Auditing Networks. Used to detect security risks, SATAN is run remotely from a computer against either another system or an entire network. As with Crack, some companies are reluctant to use this kind of tool because it is not necessary to have access to the system or network being tested. However, the same reasons for using Crack apply here.

An extremely useful feature of SATAN is that when a security risk is identified, a tutorial is provided that not only explains the detail of the risk, but also offers advice on how to fix it. The tutorial also details the potential impact of not fixing it.

Sinning with SATAN

Be aware that there are severe legal implications for using a tool such as SATAN to assess a network or system to which you do not have legitimate access. As a security prevention tool, it can be useful in identifying weaknesses in your own system or network.

Tcp Wrapper

Tcp Wrapper is a freely available tool that provides monitoring and control of network services. It is installed and configured so that when certain communications ports are connected to, the Wrapper program runs instead of the intended daemon, such as Telnet. Authentication can be carried out in addition to detailed logging before passing successful connections to the original daemon. Tcp Wrapper delivers an extra level of security, which is essential when allowing external connections via the Internet to the corporate systems.

Tripwire

Tripwire is a company producing security software designed to maintain the integrity of a system or a number of systems. The majority of companies rely heavily on the integrity of their data and need to know immediately if it has been compromised in any way. Tripwire can be used in a variety of applications; the main uses are described here:

- **Intrusion detection**—Tripwire software notifies the user when an intrusion is detected. It does this by noticing that files have changed or been tampered with. When Tripwire is run for the first time, it takes a snapshot of the files or file systems that it has been configured to monitor. This counts as a baseline for future, regular comparisons to be made, effectively creating a digital fingerprint of the system. As soon as the software notices that a file has been changed, it notifies the user and identifies exactly what has changed and what needs to be done to undo the damage.
- **Unauthorized software check**—In the same way that the software checks for an intruder modifying or altering files on the system, Tripwire also acts as a verification tool that a system has not had any unauthorized software installed on it. This is something that a cracker might want to do to install a virus.
- **Assessment and assistance with recovery**—If an attack is taking place, the Tripwire software produces violation reports so that the files requiring repair or replacement can be quickly identified.
- **Evidence reporting**—The reports produced by the Tripwire software can be utilized in forming a chain of events for data intrusions, specifying what happened and at what time. They are useful for providing formal evidence of the intrusion.
- **Configuration verification**—The Tripwire software can also be used to verify configurations across several different systems. Suppose that there are 10 servers with similar configurations. Tripwire makes a database using as a template a configured server. It then checks against the other servers to see that they are using the same applications, for example, or are using up-to-date versions of software. In this way, the software can monitor systems, watch for the abuse of software licenses, and monitor which applications are installed. It also has the advantage of noticing when untested modifications are made to the system because the difference between the modified system and the template will be flagged to the user.

The Tripwire software is extremely useful for maintaining the integrity of software and data. In addition to the uses provided here, Tripwire can also be used across the network to centrally manage systems throughout the enterprise. Tripwire HQ Connector contains the integrity software along with a communications agent, allowing it to communicate with a management console—this is known as HQ Manager. Using HQ Manager, a single position can act as a central point for integrity management for hundreds of different systems.

Tactical Options

A number of tactical options are available to the system manager of a Solaris network to help tighten up the security and make it much more difficult for unauthorized access to be gained. Some of these are discussed in the next sections. This is not an exhaustive discussion, but it provides a good indication of the type of action that can be taken to protect a business from malicious and, in some cases, accidental incidents.

Routing Options

By default, Solaris runs the routing daemon (in.routed). It allows routes to be dynamically added to the routing table instead of statically routing through a known gateway. Greater security and control of network routes can be achieved by creating the file /etc/defaultrouter and adding an entry containing either the name or the IP address of the primary router that the system uses. When the system is next rebooted, only the default route defined in this file will be used. Further routes can be manually added using the `route` command.

setuid and setgid Programs

Quite a number of programs make use of the setuid and setgid features of Solaris. Some, such as the `passwd` command, are needed, but others are not. setuid programs are not normally recommended because they pose added security risk. Also, when there are a large number of them on the system, it is difficult to spot a potential cracker adding one more!

The system manager should have a policy whereby setuid programs are closely managed. One good way of doing this is to collect a list of setuid and setgid programs and then compare them regularly to see if any unauthorized ones have been added.

To obtain a list of programs that have the setuid bit set, use the following command:

```
find / -perm 4000 -print
```

Or, another way:

```
find / -perm -u+s -print
```

The lists should be saved in a secure location and be used as a baseline template for comparisons to be made.

If any programs have the setuid or setgid bit set, then they should be reviewed to see if this is really necessary. Those that are not deemed necessary should have the bit(s) removed. As an example, consider the program ufsdump in the directory /usr/lib/fs/ufs. The only time that this program is used is by the superuser (root), and the program is already owned by user root. The setuid bit can easily be removed from this program without causing adverse affects on anything else.

Logging of Repeated Failed Login Attempts

Detailed logging of failed login attempts can be obtained by creating a file called loginlog. These three commands set this up correctly:

```
touch /var/adm/loginlog
chmod 600 /var/adm/loginlog
chgrp sys /var/adm/loginlog
```

The addition of this extra logging causes an entry to be written to the log file if five failed login attempts occur. The majority of the time, it is simply a user that has forgotten the password, but it could also alert the system manager (or administrators) to an intruder trying to guess passwords.

Disabling or Removing Unnecessary Services

If a particular service is not required, it should be disabled and then removed. By leaving it, the company could be exposing itself to an avoidable risk of intrusion. For example, if NFS is not being used, then comment out or remove the relevant entries in the /etc/services file, and rename or remove the scripts in /etc/rc2.d and /etc/rc3.d that automatically start NFS at system startup.

Many network facilities are started via the inetd process, its configuration file being /etc/inetd.conf. Each of the facilities listed in this file should be inspected to see whether it is appropriate for the specific environment—if not, then disable it by commenting out the entry or removing it. For example, if a company requires a more secure system, then it is a good idea to replace the standard /etc/inetd.conf file with one that just allows Telnet and FTP services (if these are required). A very good example of a network service to disable is finger because it has been known to have security problems.

Making Use of Groups

When administering user accounts, many companies create the accounts with a group of staff or general. This is less secure and potentially allows user accounts to have access to files and data that should be restricted. It is a better policy to create groups for relevant sections or functions, and then modify them as necessary when further access is required. For finer management of permissions, a combination of groups and ACLs (mentioned earlier in the "Bundled with Solaris" section) can be used.

For example, when defining groups for sections, possible candidates could be sales, marketing, or personnel. Here, the accessibility of unauthorized data is immediately reduced, preventing staff from sections outside of personnel from reading sensitive files. The other popular alternative for grouping user accounts is by function, such as sysadmin, which provides greater privilege for system administrators yet does not require them to always use the superuser account (root). Management would be a further function grouping allowing managers from various sections to share information requiring limited access, such as sensitive reports or financial data.

Summary

Network and system security have become increasingly important as more companies make use of public networks such as the Internet. The increased business and profits must offset the increased risk from unauthorized intruders; the latter could prove to be extremely costly.

Several products are available from Sun Microsystems, either included with the Solaris operating environment or as unbundled products. There are also an increasing number of products available to run on Solaris from third-party suppliers and in the public domain. Some tools available publicly are also those used by crackers to try to gain unauthorized access to systems and networks. System managers are well advised to be aware of these products so that any potential security holes can be filled at the earliest opportunity.

A central part of dealing with the computer security issue is formulating a security policy. This is the document that draws everything together, the resources that need protection, the risks associated with them, and the way in which they should be protected. It also contains the procedure to follow if an attack on a system or network is detected.