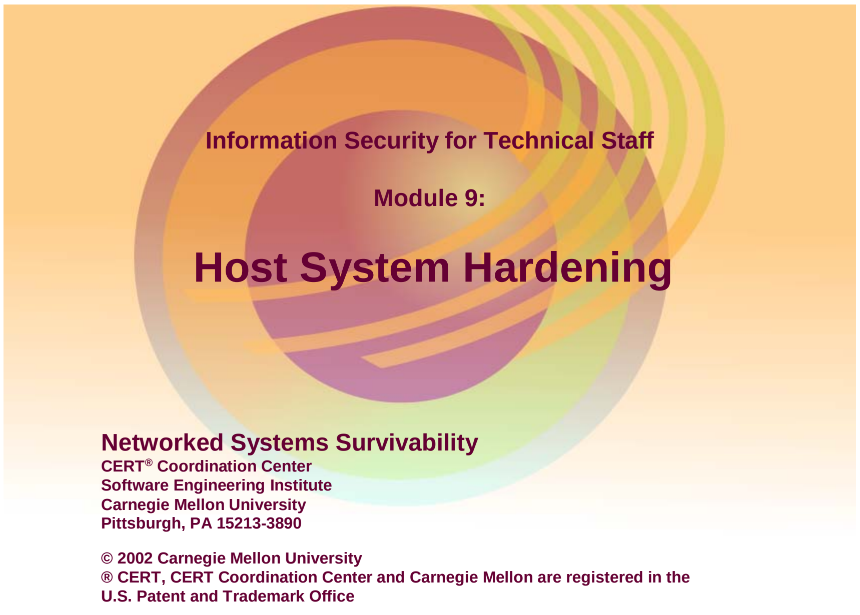


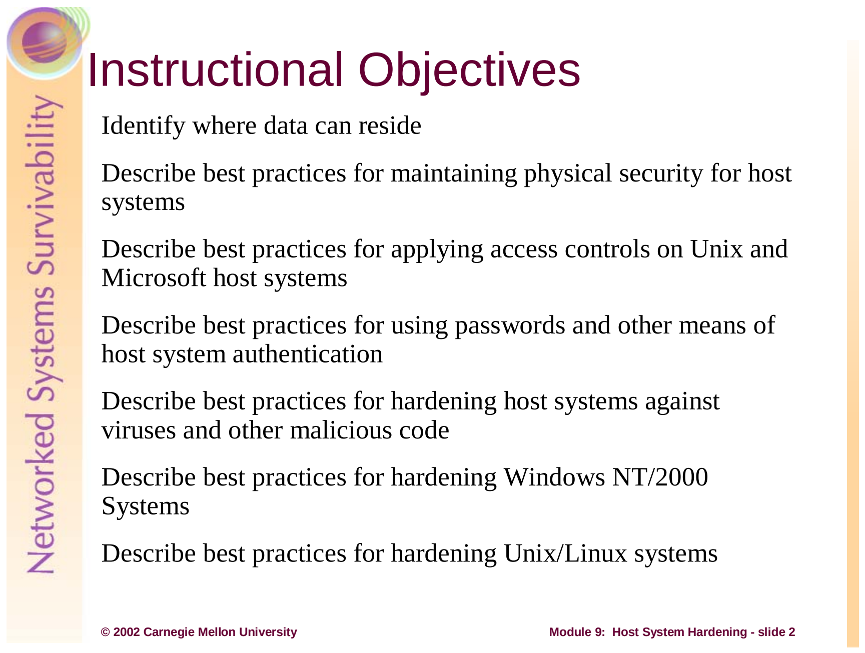
Information Security for Technical Staff

Module 9:

Host System Hardening

Networked Systems Survivability
CERT® Coordination Center
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213-3890

The first step in the SKiP Method (described in Module 4) is hardening that system against known threats and vulnerabilities. This module describes a set of tasks that the administrator performs as well as the philosophies behind those tasks. The philosophies are important because the current list of threats and vulnerabilities continues to grow almost daily. This means that the administrator must supplement this information presented here to address the latest tasks when hardening a system.



Networked Systems Survivability

Instructional Objectives

Identify where data can reside

Describe best practices for maintaining physical security for host systems

Describe best practices for applying access controls on Unix and Microsoft host systems

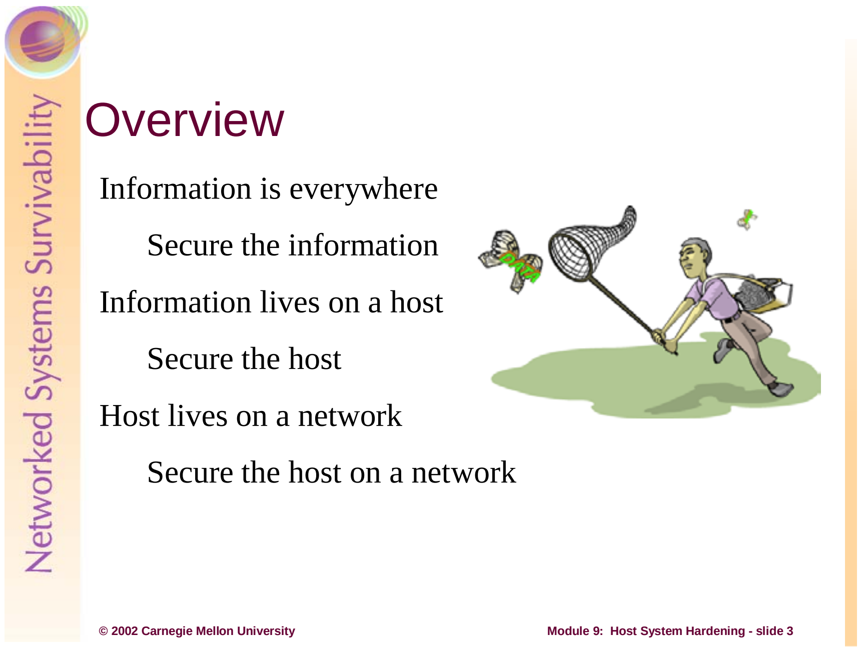
Describe best practices for using passwords and other means of host system authentication

Describe best practices for hardening host systems against viruses and other malicious code

Describe best practices for hardening Windows NT/2000 Systems

Describe best practices for hardening Unix/Linux systems

© 2002 Carnegie Mellon University　　Module 9: Host System Hardening - slide 2



Networked Systems Survivability

Overview

Information is everywhere

- Secure the information

Information lives on a host

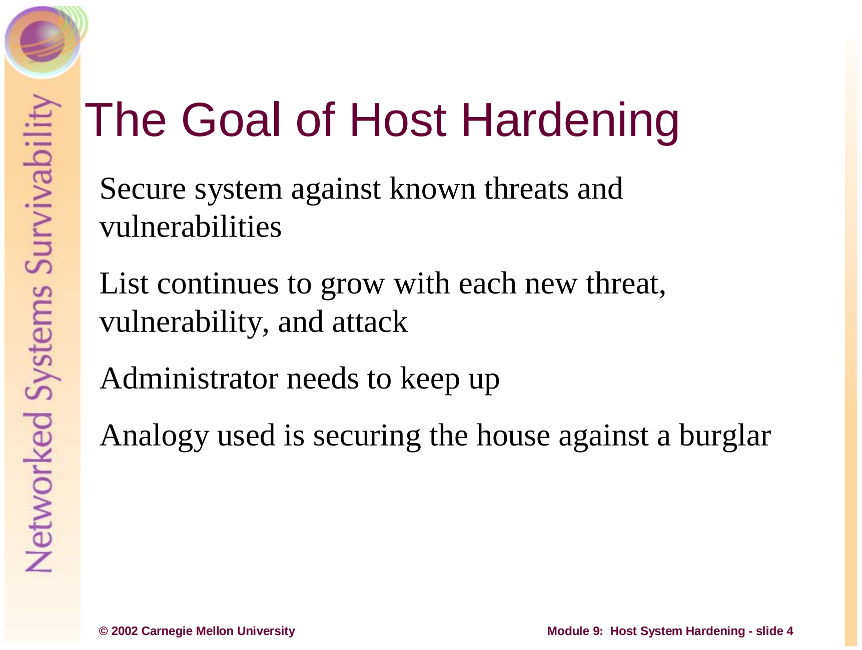
- Secure the host

Host lives on a network

- Secure the host on a network

Module 9: Host System Hardening - slide 3

To help understand the process of hardening a computer system, we need a method for thinking about that task. Imagine then that in your house, you've got some very sensitive information that you want to guard against prying eyes, perhaps others in your immediate family, and likely anyone outside of that same group. Some members of your family – your spouse for example – can see that information and maybe even change it. Others shouldn't be able to see it let alone change it. Finally, you are concerned with that information being stolen and you need to take steps to address that type of threat. We'll use this analogy to describe the process of hardening a host computer system.



The Goal of Host Hardening

Secure system against known threats and vulnerabilities

List continues to grow with each new threat, vulnerability, and attack

Administrator needs to keep up

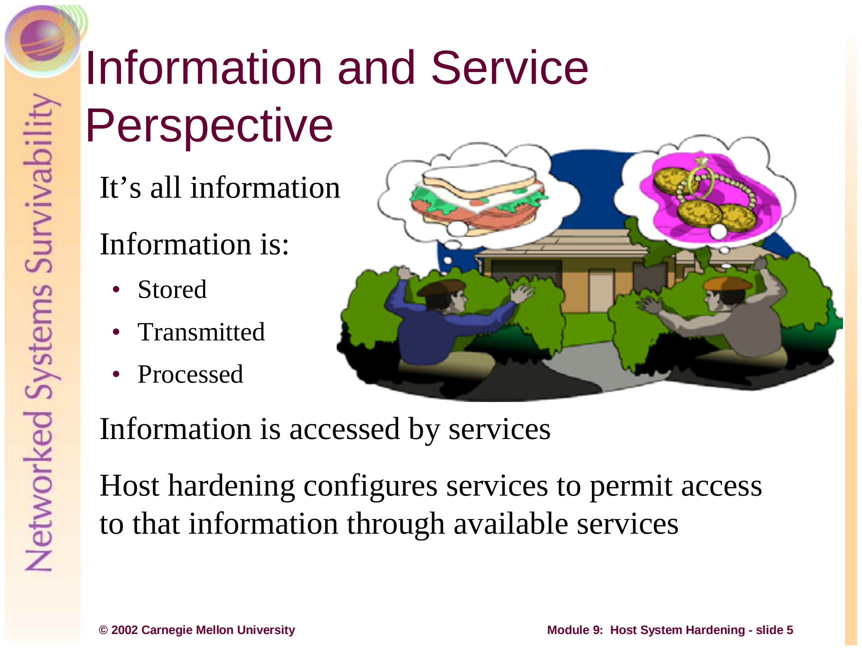
Analogy used is securing the house against a burglar

Networked Systems Survivability
© 2002 Carnegie Mellon University
Module 9: Host System Hardening - slide 4

Your sensitive information is in a file cabinet that is locked with a key or a combination lock. It's in a room with a door that's at least closed and perhaps locked. You've told your family members who is allowed to be in that room and when. That room is in your house, and that house has locks on the doors, perhaps even with dead bolts, and also on the windows. You may have purchased and installed a security system to keep unwanted persons out. You've probably told the other family members who can be in the house and how they should identify themselves when greeted at the door. Why did you do these things?

All of these steps are designed to address known ways that burglars break into a house. We've all learned over time what burglars do and the steps we can take to secure our valuables against them. We've purchased and had installed the defenses that we can afford and that we feel are appropriate for what we're trying to protect. For example, we could have purchased and installed heavy steel doors with combinations similar to those found in a bank, but that's likely too expensive and beyond what we really need to keep our sensitive information safe.

Host hardening is very similar to the house hardening we've described above. Its goals are to address the ever-growing list of threats, vulnerabilities, and attacks against a host computer system (or for that matter a router or a sub-infrastructure such as remote access, email, Web service, etc.). It is the first major step in the SKiP Method as defined in Module 4 *Security Knowledge in Practice* and it is performed when the system being installed is in a highly controlled environment, for example, when that host is disconnected from an active production network or when it is attached to a test network where resources on that network are used to ease the installation process. Once a host has been hardened, it is ready to be characterized or base-lined to learn how it operates in a production setting. This latter activity is geared toward anomaly-based intrusion detection, a topic discussed elsewhere.



Information and Service Perspective

It's all information

Information is:

- Stored
- Transmitted
- Processed

Information is accessed by services

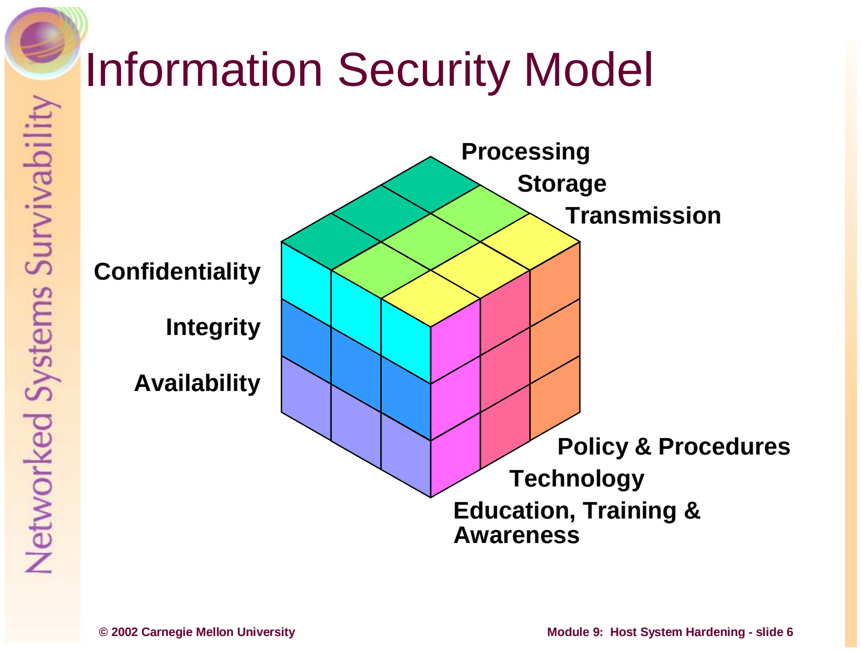
Host hardening configures services to permit access to that information through available services

Networked Systems Survivability

Module 9: Host System Hardening - slide 5

In your house, everything is valuable to someone, even your most insignificant property. For example, to a jewel thief, your jewelry is valuable; to a person who has not eaten in a while, the contents of your refrigerator are valuable. The bottom line is that all of your property is important to someone and it all needs to be secured against various threats. The key here is to recognize that everything in your house is property and worthy of your security attention.

Similarly, everything on a computer system can be thought of as a type of property. Specifically, the idea here is to recognize that every file, every bit of data within the domain of computer and network security, is **information**. That information is accessible through **services**, be they the file system service of an operating system or other network-based services, such as a Web server or an email server. That information is produced and consumed by programs that run under some circumstances. If you think about restricting access to that information by controlling the services that access that information, then you've adopted what we will call the *Information and Service Perspective* to host system hardening. We'll use this philosophy throughout this module to do host system hardening.



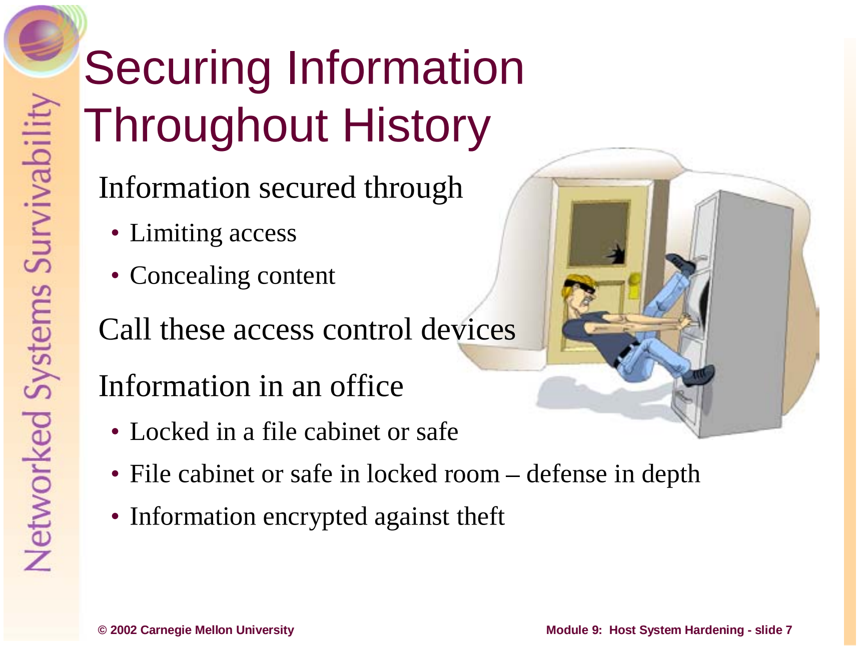
Information Security Model

Networked Systems Survivability
Processing
Storage
Transmission
Confidentiality
Integrity
Availability
Policy & Procedures
Technology
Education, Training & Awareness
© 2002 Carnegie Mellon University
Module 9: Host System Hardening - slide 6

We already know that a file used by an application is considered information. How then do we secure that information against an intruder? To answer this question, we need a way to think about securing information in general, and the following illustration, adapted from the National Training Standard for Information Systems Security Professionals [NST94], gives us such a way. This topic was discussed in Module 1, but it's important to review it here.

Recall that at the heart of this Rubik's cube is information. It has attributes of confidentiality, integrity, and availability. Information exists in the state of processing, storage, and transmission, and policies and procedures, technology, and education, training, and awareness secure it.

All of these match well to our sensitive-information-at-home analogy. That information must be kept confidential, we are concerned about it's integrity, and it must be available to us when we need it. We need to secure it when we're using it, when it is stored, and perhaps when we are sending it through the mail and transferring it by other means. Finally, we guard it through policy and procedures (who can go into the room and house and who cannot), technology (locks on the file cabinet, room, house, windows, etc), and education, training, and awareness (perhaps the parent disguising themselves as an untoward person arriving at the door to see how the children react). If you have a good grasp on securing your property in your house, then host hardening ought to be straightforward.



Securing Information Throughout History

Information secured through

- Limiting access
- Concealing content

Call these access control devices

Information in an office

- Locked in a file cabinet or safe
- File cabinet or safe in locked room – defense in depth
- Information encrypted against theft

Networked Systems Survivability
© 2002 Carnegie Mellon University
Module 9: Host System Hardening - slide 7

Given that information is the focal point for this approach to information security and the attributes of confidentiality, integrity, and availability, what must the administrator do to secure information assets against those who seek to disrupt them in some way? Let's begin by looking at the attribute of confidentiality.

Before computers were available, users would address information security by locking that information – which probably existed as paper – in a filing cabinet or a safe. Those who should have access were given a key or the combination to the safe, and those who shouldn't have access were not. By controlling access with a key or the combination, the information could be kept confidential, and these methods were pretty successful.

The filing cabinet and safe stand between the information and its producer(s)/consumer(s). They can be thought of *access control devices*. Their job is to arbitrate who has access to the information. Throughout this module, there will be other pieces of technology that are in fact access control devices, and the challenge of securing computers and networks is simplified by thinking of those technologies in this way.

Returning to the time before computers, along came lock picks and more sophisticated tools for opening a safe. Since the lock on the filing cabinet and safe controls didn't always work because the information could still be accessed and subsequently viewed, something else needed to be done to secure those valuable information assets.

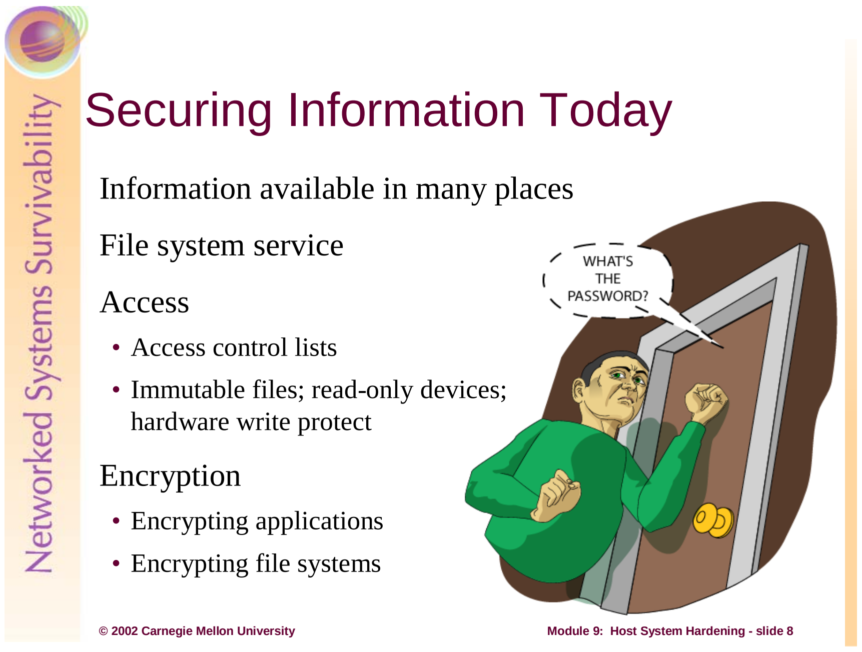
To combat this new problem, the assets were concealed from view by some means. The idea was that if the assets fell into the wrong hands, they would be useless.

As discussed in the cryptography module, Julius Caesar concealed information by using a *cipher*. A cipher is a means of transforming text to conceal its meaning. His cipher [Laubenbacher] transformed each letter in the alphabet by a constant, 3 for example. This means that to convey the letter A, he would write the letter D. The word *data* would be written as *gdwd*. If his enemies didn't know or couldn't figure out the scheme, the information was useless to them.

In more modern times, another such scheme used during World War II was the Navajo language as depicted in the movie *Windtalkers* [MGM 02]. Concealing information through this type of encryption is another type of access control device.

This combination of restricting access and concealing information through encryption has served us well for a long time. Walk around any modern office building today and you'll see many locked file cabinets. In special settings, you'd also find that the most sensitive information is encrypted to guard it against concerted attempts to gain access.

In our securing-information-at-home analogy, this is exactly what would be done. The sensitive information would be locked in a cabinet and perhaps encrypted to protect against theft, if we felt it was important enough to merit such measures.



Networked Systems Survivability

Securing Information Today

Information available in many places

File system service

Access

- Access control lists
- Immutable files; read-only devices; hardware write protect

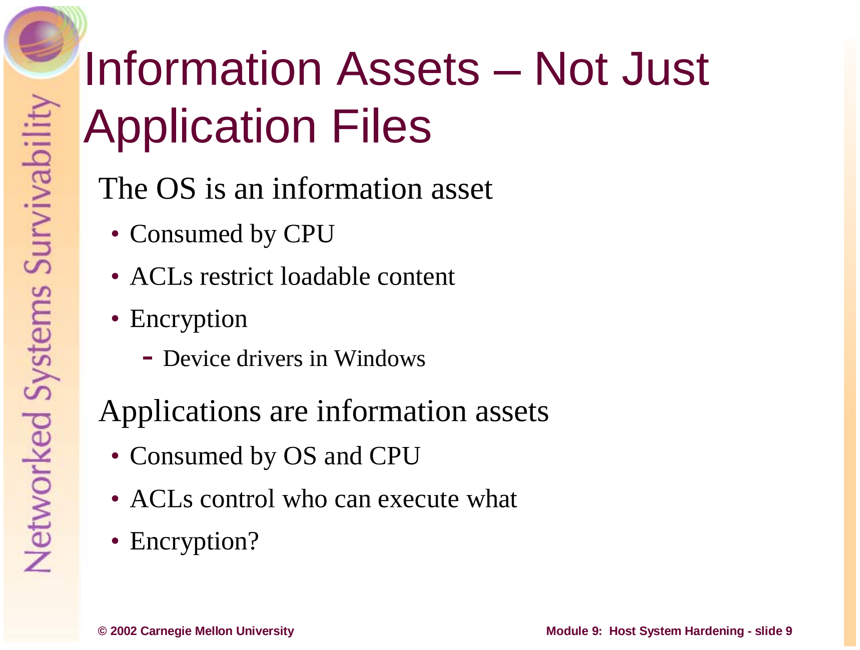
Encryption

- Encrypting applications
- Encrypting file systems

© 2002 Carnegie Mellon University

Module 9: Host System Hardening - slide 8

On a computer system, access control and encryption help information keep its confidentiality attribute. For controlling access to a file, most modern computer systems provide some form of *access control*. Only the set of identities that need access to an asset can have the access through these controls. Note that access control granularity varies from operating system to operating system as do the specifics of the type of access and how long that access is granted. The administrator's challenge is to tune the available access controls for each and every file and asset on a computer system to match its organizationally specified use. It is a hard but increasingly necessary job.


Networked Systems Survivability

Information Assets – Not Just Application Files

The OS is an information asset

- Consumed by CPU
- ACLs restrict loadable content
- Encryption
  - Device drivers in Windows

Applications are information assets

- Consumed by OS and CPU
- ACLs control who can execute what
- Encryption?

© 2002 Carnegie Mellon University Module 9: Host System Hardening - slide 9

In much the same way that the homeowner only thinks about jewelry, the CD player, and the television as the assets to be protected, administrators usually stop thinking about data when they move beyond disk files operated on by applications. In reality, the full range of what should be considered data – and therefore benefit from these access control devices – has not been completely explored. Let’s dig deeper into other sources of data and apply these access control and encryption concepts to those new places in new ways.

For example, the operating system (OS) is an information asset that is consumed by the CPU part of a computer system. The process of booting an OS loads that special-purpose program into the computer’s memory and then executes it. Now, some OSes only load the operating system code and other code such as device drivers from specific places on a file system. This means that file system access controls are the access control devices that restrict the range of what can be booted and then subsequently loaded into an operating system.

To further improve security, vendors could apply encryption techniques in concert with the CPU, thereby allowing only well defined operating system code and dynamic content to be installed into the OS. While there yet aren’t any widely deployed OSes that encrypt this type of content, Windows 2000 and XP do approximate this functionality by using driver signing as a requirement for loading a driver into the kernel [Microsoft 01a].

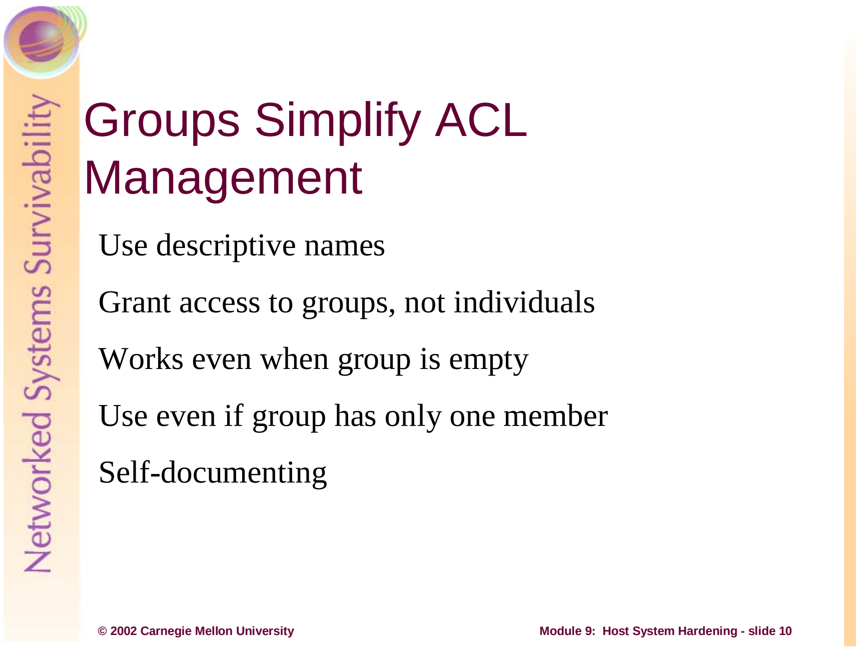
Continuing on with this example, applications are also information assets. The OS and the CPU consumes them when they run. Access controls in the file system also restrict which applications users can execute. Again, there could be cooperation between the OS and applications so that the OS would only run properly encrypted applications.

In fact, if those applications were licensed, copying an encrypted application from one system to another wouldn’t be effective because the application would not be properly encrypted for that new target system. Currently, license servers act as access control devices, but the operating system in concert with encryption technologies could do the same job.

On a computer system then, every file is an information asset, no matter its place in that system or its usage. The access control devices used to secure assets are access control lists and encryption. These technologies stand between the assets and its producer(s)/consumer(s), arbitrating access as necessary.

Consider one more example of an information asset to which these access controls can be applied. Backups are the heart of recovery of a computer system. Frequently, backup media – tapes, whatever – are stored in locked cabinets or rooms so that they cannot be accessed except under special circumstances.

Imagine then that someone does pick the lock and gain access to those tapes. Now, if the assets on those tapes where encrypted, it wouldn’t matter if the tapes were stolen. So, for defense in depth, put backup takes behind a locked door and encrypt their contents. These are additional instances of the two simple principles of limiting access and concealing through encryption.



Networked Systems Survivability

Groups Simplify ACL Management

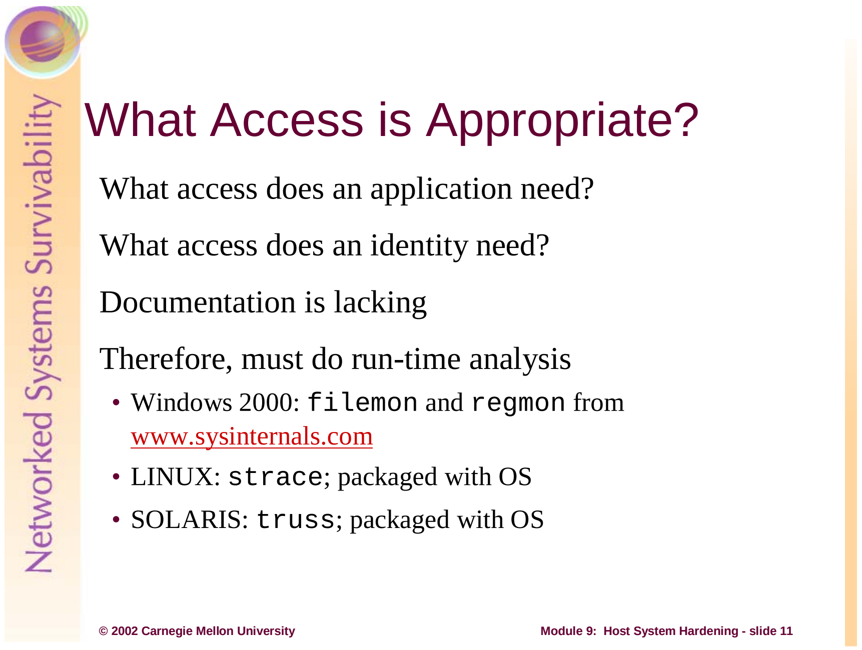
- Use descriptive names
- Grant access to groups, not individuals
- Works even when group is empty
- Use even if group has only one member
- Self-documenting

© 2002 Carnegie Mellon University

Module 9: Host System Hardening - slide 10

The first layer of defense against the would-be intruder, or more generally those who shouldn't have access to data, is *access control lists,* also known as ACLs. ACLs control which users – actually the identities that they are logged in as – can access information in a file system. Before explaining ACLs, let's first discuss the nature of those identities that are granted or denied access to data.

From a pragmatic point of view, it is strongly recommended that access to information be discriminated based on groups, not individuals. For example, this means that instead of giving read-only access to the Master Auto Parts File to the identities `Manny`, `Moe`, and `Jack`, the administrator would first create a group descriptively named `Sales_Clerks` and place the identities `Manny`, `Moe`, and `Jack` into that group. The access control lists associated with the Master Auto Parts File would then be changed to grant read-only access to the group `Sales_Clerks`. If that ever changes, only the group definition needs to change; the file need never change, even if no one is a member of the `Sales_Clerks` group. Plus, the group's name helps to document the access granted to those who are members. It is strongly recommended that you use groups in ACLs even if they contain only one member.



What Access is Appropriate?

What access does an application need?

What access does an identity need?

Documentation is lacking

Therefore, must do run-time analysis

- Windows 2000: `filemon` and `regmon` from www.sysinternals.com
- LINUX: `strace`; packaged with OS
- SOLARIS: `truss`; packaged with OS

© 2002 Carnegie Mellon University Module 9: Host System Hardening - slide 11

The biggest challenge in setting up access controls in an operating system's file system is usually NOT the use of interfaces to set those permissions. Instead the challenge is discovering what level access is appropriate. The question quickly becomes how does the administrator know which identities need what type of access to properly use the data items in question?

As will be discussed in more detail later in this module, a program runs with a set of credentials that identify the entity (logged in user) and a set of rights and privileges to which that user is entitled. The key is to minimize the access that a program, operating on behalf of the user, has to information resources on a computer system (e.g., the file system, functions within the kernel, access to the network, etc.). For now, the issue is the task of determining the set of information resources accessed by a program.

Unfortunately, vendor documentation usually does not provide enough guidance to help the administrator answer these questions. The information resources that programs use (files, directories/folders, and registry entries on Windows-based systems) and the way they are used (read, write, execute) must be determined through brute-force techniques. Fortunately, there are tools available to learn how programs interact with these resources.

Specifically, under Windows-based operating systems, the `filemon` and `regmon` tools from SysInternals [Sysinternals 02] are free and can show the file, directory, and registry resources used by a running program.

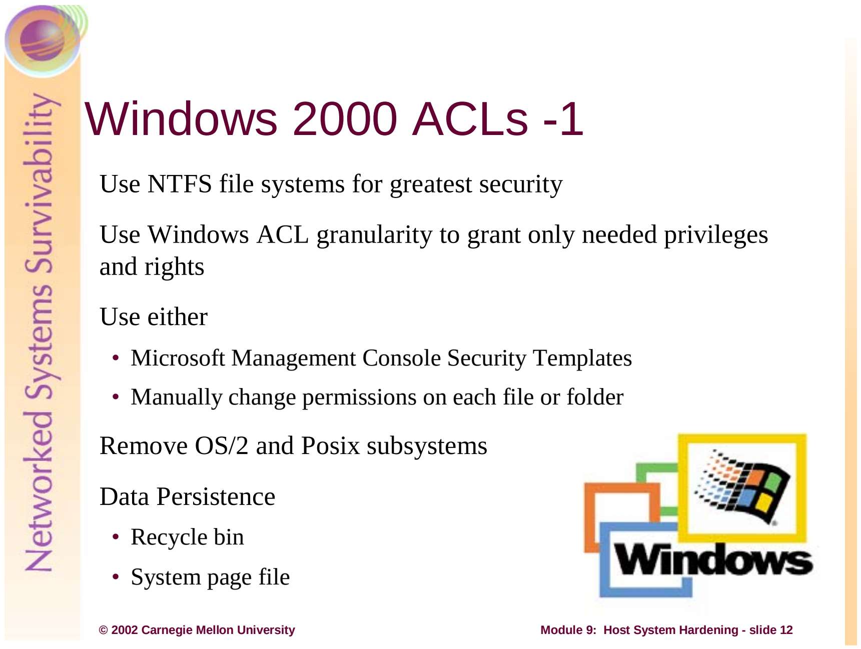
For LINUX-based systems, `strace` [Ackerman] shows the system call patterns. The following command shows all of the files opened by the `/bin/date` program:

```
    strace /bin/date |& grep '^open('
```

Note that that this command does not show all of the files operated upon by a program. To determine that list, the administrator must determine all system calls that operate on files by name and then change the `grep` search pattern to look for all of them. For example, the command above can be changed to the following to find more file name accesses:

```
    strace /bin/date |& egrep '^open(|^access('
```

Finally, SOLARIS works the same way as Linux except that `truss` [Lucy 00] is used instead of `strace`.



Networked Systems Survivability

Windows 2000 ACLs -1

Use NTFS file systems for greatest security

Use Windows ACL granularity to grant only needed privileges and rights

Use either

- Microsoft Management Console Security Templates
- Manually change permissions on each file or folder

Remove OS/2 and Posix subsystems

Data Persistence

- Recycle bin
- System page file

© 2002 Carnegie Mellon University   Module 9: Host System Hardening - slide 12

Windows 2000 ACLs provide sufficient controls and granularity to adjust access to information assets as needed by the systems administrator. This section is derived from [NSA 01h].

Windows 2000 New Technology File System (NTFS) version 5 is a secure file system that provides a reliable way to safeguard valuable information. NTFS works in concert with the Windows 2000 user account system to allow authenticated users access to files**.**

The following notation will be used in this section:

- *%SystemDrive%* - the drive letter on which Windows 2000 is installed, e.g. *C:\*
- *%SystemRoot%* - the folder in which Windows 2000 is installed, e.g. *C:\winnt*
- *%SystemDirectory%* - the folder in which Windows 2000 system files are installed e.g. *C:\winnt\system32*

All volumes must use NTFS to achieve the highest level of security. Under Windows 2000, only NTFS supports Discretionary Access Control to the directories and files. NTFS volumes provide secure and auditable access to the files. Therefore, any File Allocation Table (FAT16/FAT32) partitions should be converted to NTFS.

A non-NTFS volume can be converted at any time using the **convert.exe** program *(%SystemRoot%\system32\convert.exe*). The **convert.exe** program must be executed from a command prompt window using an administrative account.

The steps needed to convert a drive to NTFS are as follows:

- Select *Start→Run→*`cmd.exe` to open a command prompt
- At the command prompt, type the command below. Substitute the drive letter of the partition to be converted for `volume` (i.e. *C:*). The `/v` switch is optional and runs the program in verbose mode

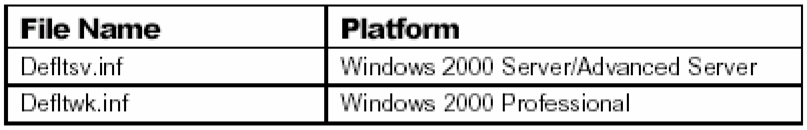
```
convert volume /FS:NTFS [/V]
```

- Restart the system

Note that the conversion will not take effect immediately on the system drive or any drives being used for page swapping. In this case, conversion is performed when the system is restarted. This process should not destroy any data, but backing up valuable data is always advisable.

Once the partition is converted, the `Everyone` group will have full control of the entire partition. Since the `Everyone` group consists of all users, including anonymous users (null connections), it is critical that stricter file permissions be set before any users are added to the system.

There are several security template files that contain the default security settings applied to a clean-installed (non-upgraded) Windows 2000 machine. These files reside in the *%SystemRoot%\inf folder*. The table below shows a list of the default security templates.



File Name	Platform
Defltsv.inf	Windows 2000 Server/Advanced Server
Defltwk.inf	Windows 2000 Professional

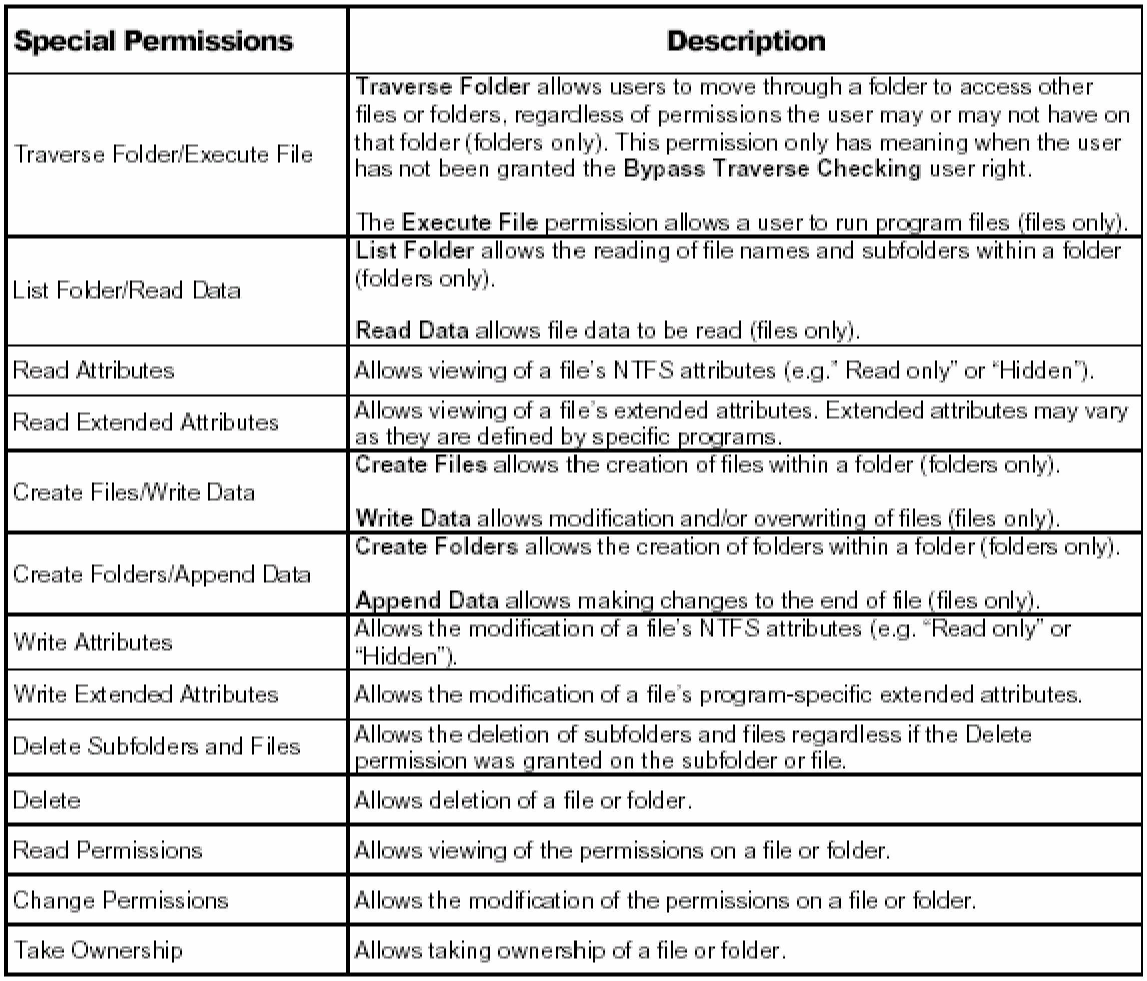
The default security templates are especially useful when converting from a FAT or FAT32 file system to NTFS. To obtain the file system security settings that would have been present if NTFS had been the original file system, the File System portion of the default security templates can be applied [NSA 01c].

NTFS allows for varying levels of file access permissions to users or groups of users. Coupled with file access permissions is the concept of *inheritance*. By default, newly created files or folders inherit the parent folder's file access permissions.

To manually view permissions on a specific file or folder:

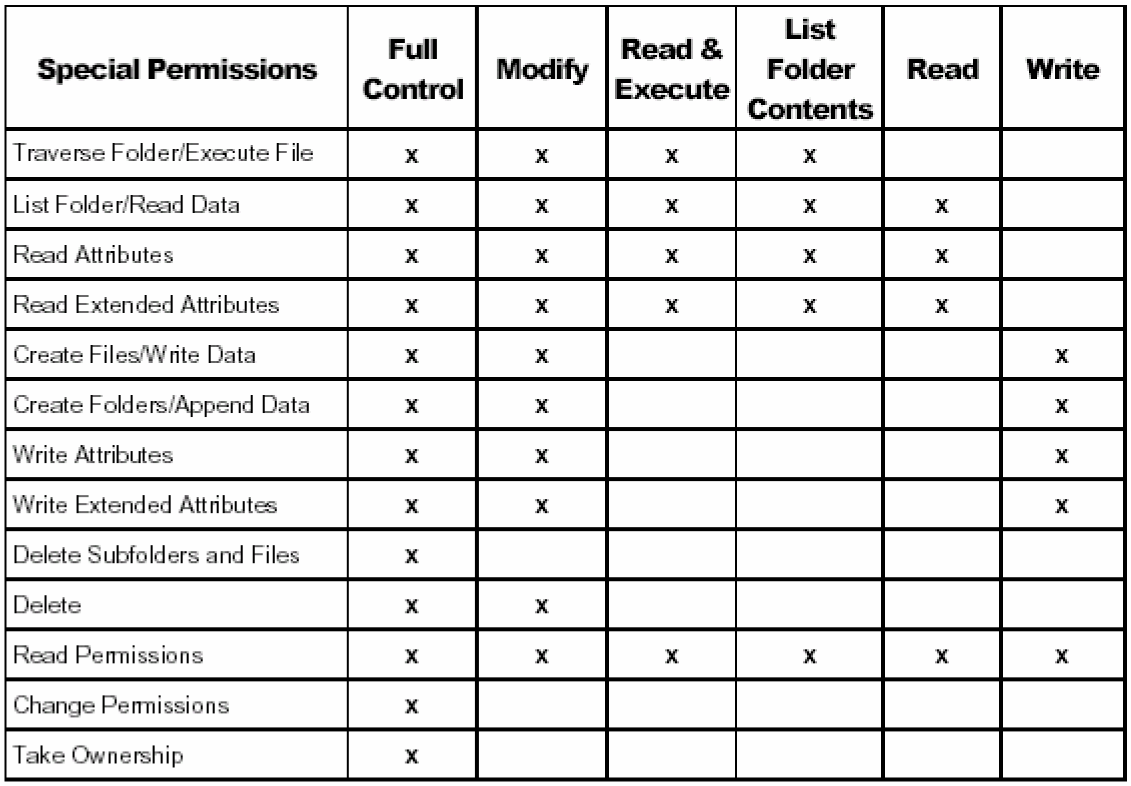
- In Windows Explorer right-click on the file or folder
- Select *Properties* from the pull-down menu
- Click the *Security* tab
- Click *Advanced* to see more detailed permission information

File permissions may be set with more granularity than those listed in the *Permissions* dialog box by clicking the *Advanced* button. The first table below shows a list of high-level file permissions, and next two tables show which low-level permissions to select in order to achieve these higher-level permissions for folders and files, respectively.



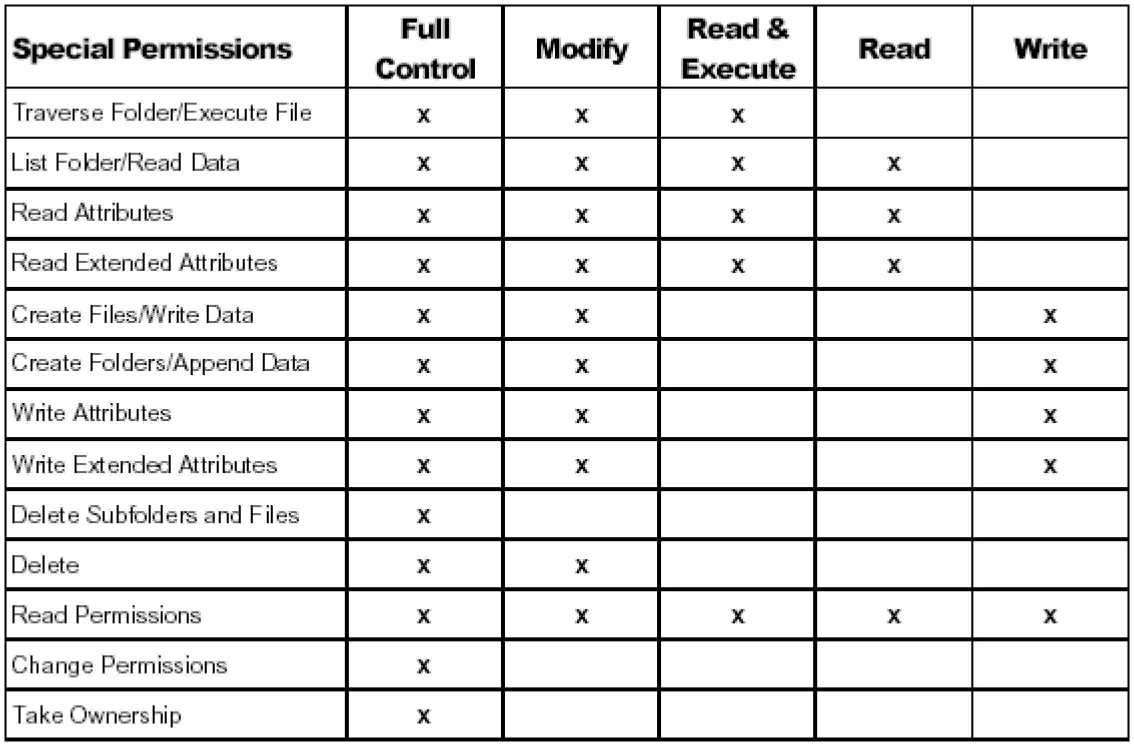
Special Permissions	**Description**
Traverse Folder/Execute File	**Traverse Folder** allows users to move through a folder to access other files or folders, regardless of permissions the user may or may not have on that folder (folders only). This permission only has meaning when the user has not been granted the **Bypass Traverse Checking** user right. The **Execute File** permission allows a user to run program files (files only).
List Folder/Read Data	**List Folder** allows the reading of file names and subfolders within a folder (folders only). **Read Data** allows file data to be read (files only).
Read Attributes	Allows viewing of a file's NTFS attributes (e.g." Read only" or "Hidden").
Read Extended Attributes	Allows viewing of a file's extended attributes. Extended attributes may vary as they are defined by specific programs.
Create Files/Write Data	**Create Files** allows the creation of files within a folder (folders only). **Write Data** allows modification and/or overwriting of files (files only).
Create Folders/Append Data	**Create Folders** allows the creation of folders within a folder (folders only). **Append Data** allows making changes to the end of file (files only).
Write Attributes	Allows the modification of a file's NTFS attributes (e.g. "Read only" or "Hidden").
Write Extended Attributes	Allows the modification of a file's program-specific extended attributes.
Delete Subfolders and Files	Allows the deletion of subfolders and files regardless if the Delete permission was granted on the subfolder or file.
Delete	Allows deletion of a file or folder.
Read Permissions	Allows viewing of the permissions on a file or folder.
Change Permissions	Allows the modification of the permissions on a file or folder.
Take Ownership	Allows taking ownership of a file or folder.

File Permissions



Special Permissions	Full Control	Modify	Read & Execute	List Folder Contents	Read	Write
Traverse Folder/Execute File	x	x	x	x		
List Folder/Read Data	x	x	x	x	x	
Read Attributes	x	x	x	x	x	
Read Extended Attributes	x	x	x	x	x	
Create Files/Write Data	x	x				x
Create Folders/Append Data	x	x				x
Write Attributes	x	x				x
Write Extended Attributes	x	x				x
Delete Subfolders and Files	x					
Delete	x	x				
Read Permissions	x	x	x	x	x	x
Change Permissions	x					
Take Ownership	x					

Folder Permissions Options



Special Permissions	Full Control	Modify	Read & Execute	Read	Write
Traverse Folder/Execute File	x	x	x		
List Folder/Read Data	x	x	x	x	
Read Attributes	x	x	x	x	
Read Extended Attributes	x	x	x	x	
Create Files/Write Data	x	x			x
Create Folders/Append Data	x	x			x
Write Attributes	x	x			x
Write Extended Attributes	x	x			x
Delete Subfolders and Files	x				
Delete	x	x			
Read Permissions	x	x	x	x	x
Change Permissions	x				
Take Ownership	x				

File Permission Options

Good file and folder security is critical in maintaining the overall security of a Windows 2000 system. Windows 2000 provides improved default file security over Windows NT 4.0, but additional file permissions are recommended to obtain a higher level of security.

The least privilege principle should be used when deciding how to implement ACLs. In other words, grant access to those users who need to have access and then allow those users only the access levels they absolutely require. For example, if a group needs Read access to a folder, resist the temptation to give the group Full control and only grant Read access.

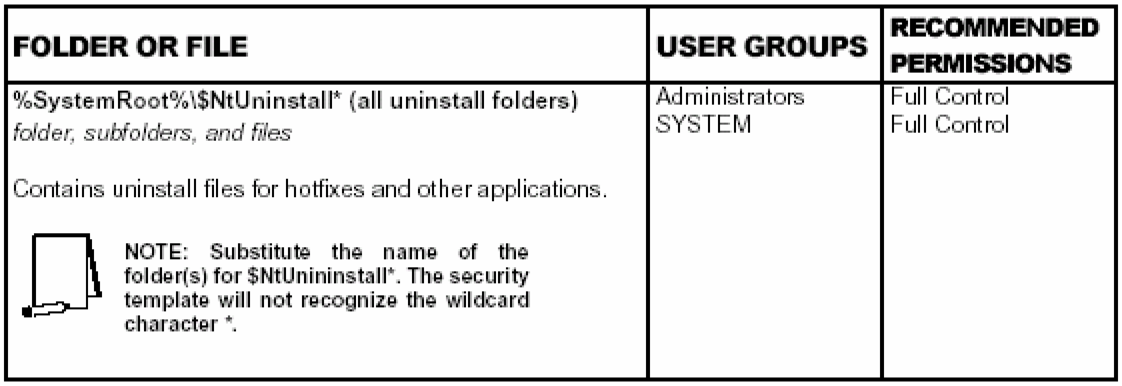
Changes to file system ACLs can be made in one of two ways. The first method is to use the Microsoft Management Console (MMC) Security Templates snap-in and the provided template to apply the recommended file and folder permissions. The alternative and more time-consuming method is to manually change permissions on each file and folder.

The Security Templates snap-in is the recommended tool for creating a configuration file for setting security on Windows 2000 File Resources. See [NSA 01c] for more information on using the Security Templates snap-in as well as adding, modifying, and excluding files and folders from the security policy. The recommended changes to system files and folders are also listed in the Security Configuration Toolset mini-guide.

To view file system settings using the Security Templates Snap-in do the following:

- In the Security Templates snap-in, select the default file directory (*%SystemRoot%\Security\Templates*)
- Select the specific configuration file
- Select *File System*

It is recommended that the folder(s) shown below be added to the security configuration template:



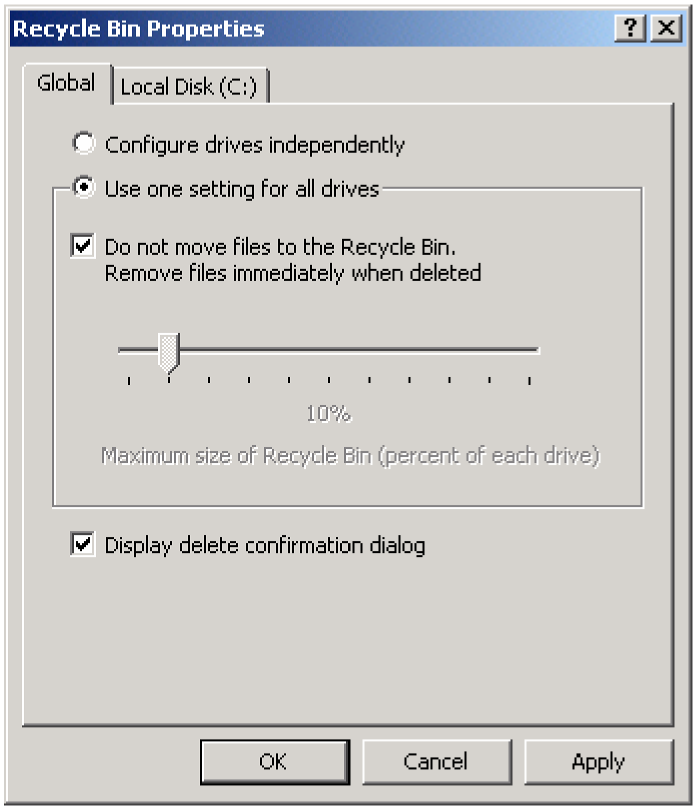
FOLDER OR FILE	USER GROUPS	RECOMMENDED PERMISSIONS
%SystemRoot%\$NtUninstall* (all uninstall folders) *folder, subfolders, and files* Contains uninstall files for hotfixes and other applications. **NOTE: Substitute the name of the folder(s) for $NtUnininstall*. The security template will not recognize the wildcard character *.**	Administrators SYSTEM	Full Control Full Control

Data remnants relates to images of data remaining on a Windows 2000 platform after the data should no longer be available. This includes data left in the system page file and the recycle bin. Each of these areas could have sensitive data that is open to being read by unauthorized or malicious users.

The Recycle Bin saves a copy of a file when it is deleted through Windows 2000 Explorer. On critical servers and key workstations, this could pose a security risk. A sensitive file may be deleted, yet a copy of that file would remain in the Recycle Bin. To configure the Recycle Bin to prevent deleted files from being saved, use the following procedure:

- Right click the **Recycle Bin** icon on the desktop, and select *Properties*.
- Check the box labeled "*Do not move files to the Recycle Bin. Remove files immediately on delete.*"
- Click *OK*.

- Empty the Recycle Bin of any pre-existing files.



○ Configure drives independently

● Use one setting for all drives

[✓] Do not move files to the Recycle Bin.
Remove files immediately when deleted

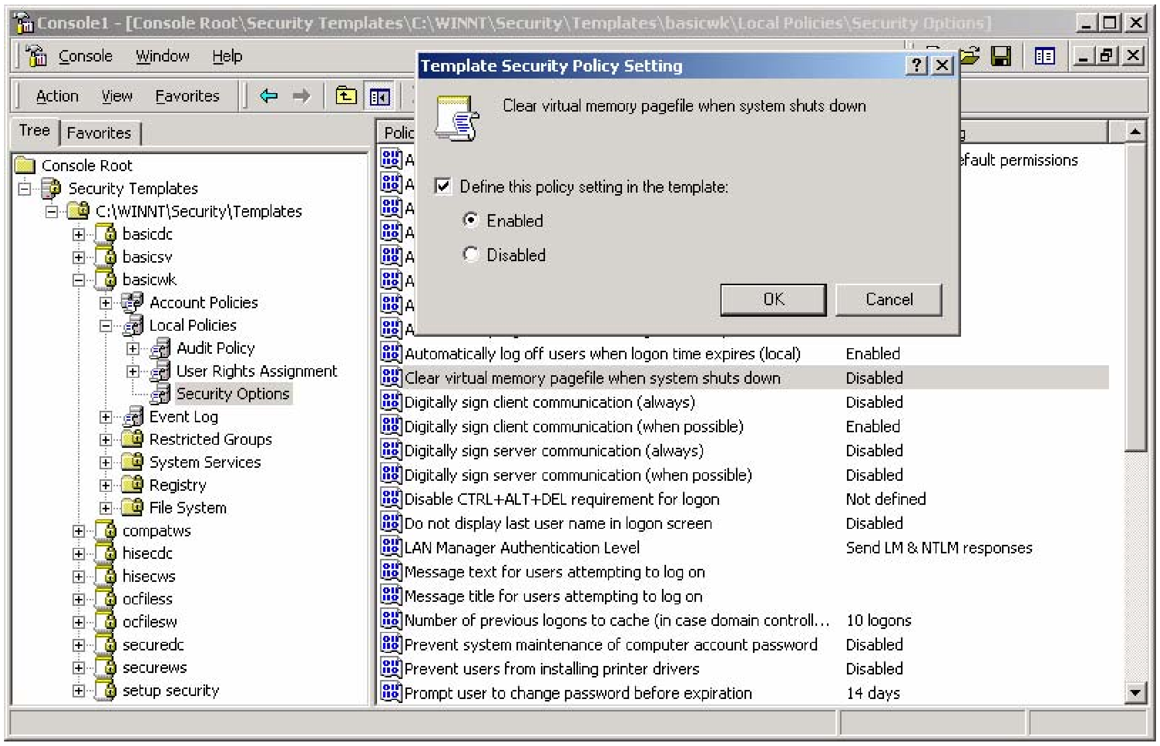
10%
Maximum size of Recycle Bin (percent of each drive)

[✓] Display delete confirmation dialog

Virtual Memory support in Windows 2000 uses a system pagefile to swap pages from memory when they are not being actively used. On a running system, this pagefile is opened exclusively by the operating system and hence is well protected. However, to implement a secure Windows 2000 environment, the system page file should be wiped clean when Windows 2000 shuts down. This action ensures sensitive information, which may be in the page file, is not available to a malicious user.

To view file system settings using the Security Templates Snap-in do the following:

- Select the default file directory (*%SystemRoot%\Security\Templates*)
- Select the specific configuration file (*deflsv.inf* or *defltwk.inf*)
- Select *Local Policies*
- Select *Security Options*
- Enable the setting for "*Clear virtual memory pagefile when system shuts down*"


Console1 - [Console Root\Security Templates\C:\WINNT\Security\Templates\basicwk\Local Policies\Security Options]
Console Window Help
Action View Favorites
Tree Favorites

Console Root
- Security Templates
  - C:\WINNT\Security\Templates
    - basicdc
    - basicsv
    - basicwk
      - Account Policies
      - Local Policies
        - Audit Policy
        - User Rights Assignment
        - Security Options
      - Event Log
      - Restricted Groups
      - System Services
      - Registry
      - File System
    - compatws
    - hisecdc
    - hisecws
    - ocfiless
    - ocfilesw
    - securedc
    - securews
    - setup security

Template Security Policy Setting

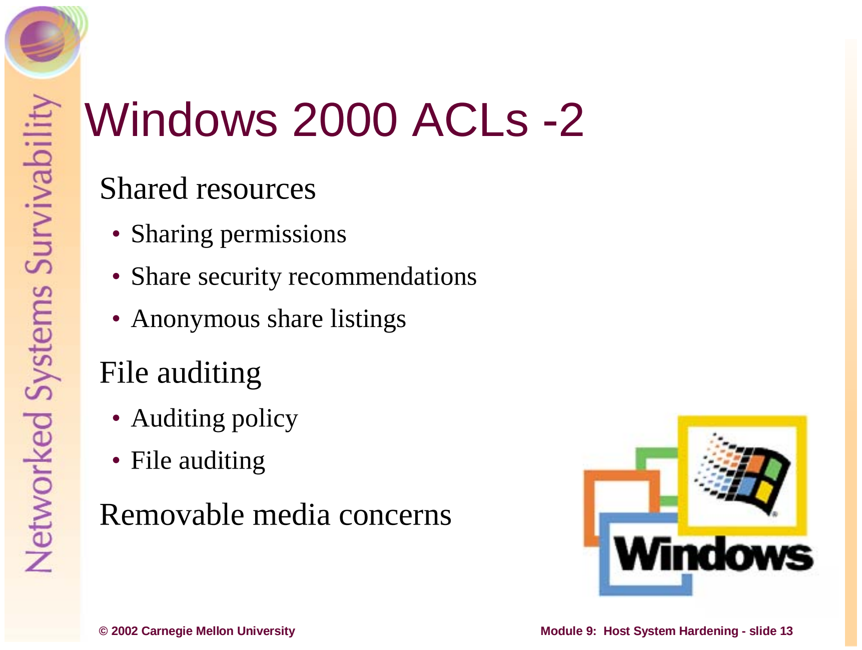
Clear virtual memory pagefile when system shuts down

[✓] Define this policy setting in the template:

[✓ Enabled] Disabled

OK Cancel

Polic	
A	efault permissions
A	
A	
A	
A	
A	
A	
A	
Automatically log off users when logon time expires (local)	Enabled
Clear virtual memory pagefile when system shuts down	Disabled
Digitally sign client communication (always)	Disabled
Digitally sign client communication (when possible)	Enabled
Digitally sign server communication (always)	Disabled
Digitally sign server communication (when possible)	Disabled
Disable CTRL+ALT+DEL requirement for logon	Not defined
Do not display last user name in logon screen	Disabled
LAN Manager Authentication Level	Send LM & NTLM responses
Message text for users attempting to log on	
Message title for users attempting to log on	
Number of previous logons to cache (in case domain controll...	10 logons
Prevent system maintenance of computer account password	Disabled
Prevent users from installing printer drivers	Disabled
Prompt user to change password before expiration	14 days



Windows 2000 ACLs -2

Shared resources

- Sharing permissions
- Share security recommendations
- Anonymous share listings

File auditing

- Auditing policy
- File auditing

Removable media concerns

Networked Systems Survivability
Windows
© 2002 Carnegie Mellon University
Module 9: Host System Hardening - slide 13

(This text is a continuation of information derived from [NSA 01h].)

Windows 2000 shares are a means by which files, folders, printers, and other resources can be published for network users to remotely access. Regular users cannot create shares on their local machines; only `Administrators` and `Power Users` have this ability and they must have at least List permission on the folder to do so. Since shares may contain important data and are a window into the local system, care must be taken to ensure proper security settings on shared resources.

The following share permissions can be granted or denied to users or groups:

- *Full Control*
- *Change*
- *Read*

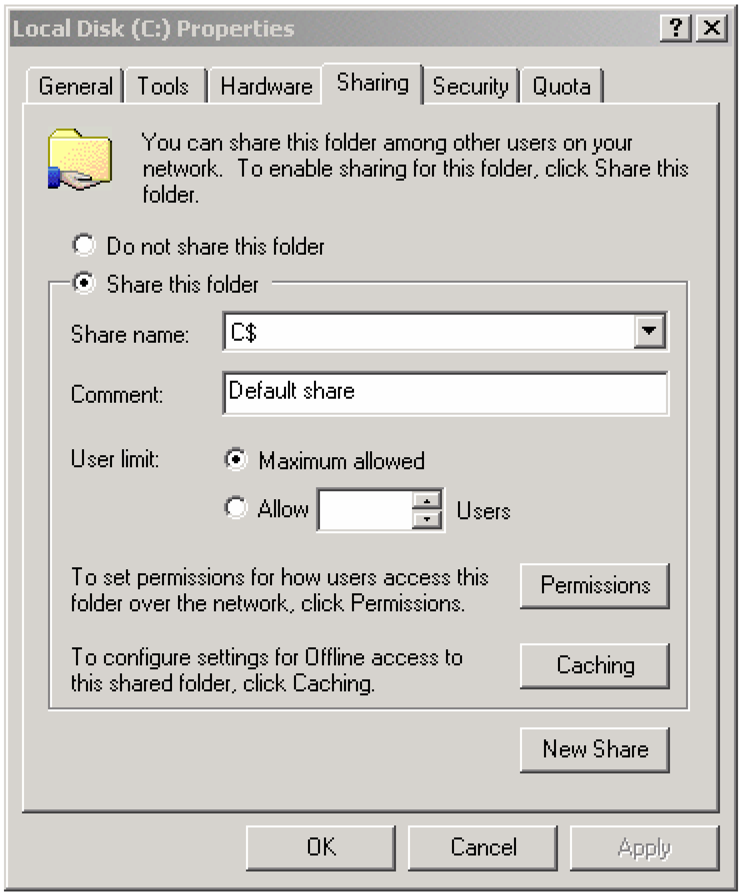
Share permissions are granted independent of NTFS permissions. However, share permissions act aggregately with NTFS permissions. When accessing a remote share, the more restrictive permissions of the two apply. For example, if a user accesses a share remotely and has *Full Control* over a shared folder, but only NTFS Read access to that folder on the local file system, he will only have *Read* access to the share.

The default permissions on a share give the `Everyone` group *Full Control*; therefore, you must explicitly edit security permissions on shared resources to limit share access.

Create A Share and Set Security Permissions

To create a share and set security permissions:

- In explorer, right mouse-click on the folder that is to be shared.
- Select the *Sharing…* menu option
- Click the *Share this folder* radio button.
- Specify the *Share Name*.
- Click the *Permissions* button.
- Add, remove, or edit the users and/or groups in the access control list for the share.



You can share this folder among other users on your network. To enable sharing for this folder, click Share this folder.

Do not share this folder [✓ Share this folder]

Share name: C$

Comment: Default share

User limit: [✓ Maximum allowed] Allow Users

To set permissions for how users access this folder over the network, click Permissions.

To configure settings for Offline access to this shared folder, click Caching.

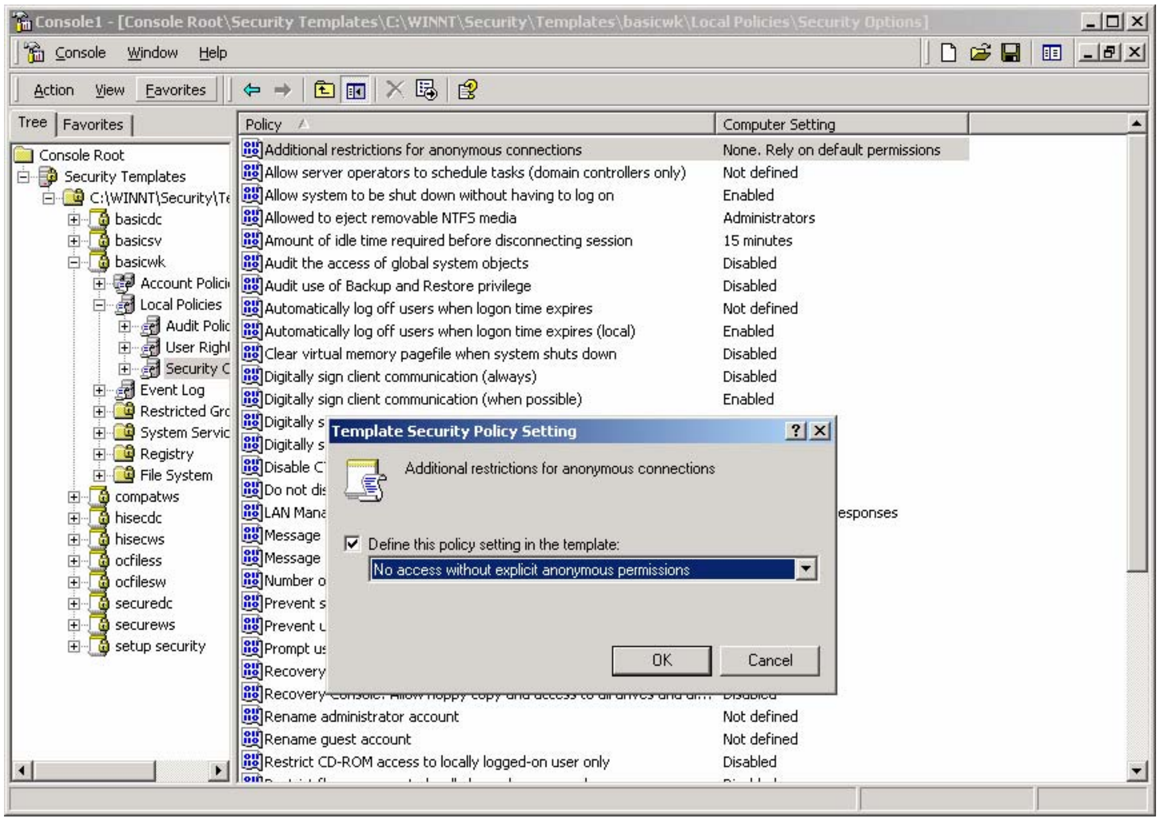
When creating shares and share permissions, adhere to the following criteria when possible:

- Ensure that the `Everyone` group is not given permissions on any shares.
- Use the `Authenticated Users` or `Users` groups in place of the `Everyone` group.
- Give users and/or groups the minimum amount of permissions needed on a share.
- To protect highly sensitive shares not for general use, hide shares by placing a $ after the share name when creating a share. Users can still connect to hidden shares, but must explicitly enter the full path to the share (i.e. the share will not be visible in Network Neighborhood).

Restrict Anonymous Logon Users

To restrict the ability for anonymous logon users (also known as NULL session connections) to list account names and enumerate share names, using the Security Templates Snap-in do the following:

- Select the default file directory (*%SystemRoot%\Security\Templates*)
- Select the specific configuration file
- Select *Local Policies*
- Select *Security Options*
- Set *"Additional restrictions for anonymous connections"* to*" No access without explicit anonymous permissions"*.


Console1 - [Console Root\Security Templates\C:\WINNT\Security\Templates\basicwk\Local Policies\Security Options]
Console
Window
Help
Action
View
Favorites
Tree
Favorites

Console Root
- Security Templates
  - C:\WINNT\Security\Te
    - basicdc
    - basicsv
    - basicwk
      - Account Polici
      - Local Policies
        - Audit Polic
        - User Righ
        - Security C
      - Event Log
      - Restricted Grc
      - System Servic
      - Registry
      - File System
    - compatws
    - hisecdc
    - hisecws
    - ocfiless
    - ocfilesw
    - securedc
    - securews
    - setup security

Policy	Computer Setting
Additional restrictions for anonymous connections	None. Rely on default permissions
Allow server operators to schedule tasks (domain controllers only)	Not defined
Allow system to be shut down without having to log on	Enabled
Allowed to eject removable NTFS media	Administrators
Amount of idle time required before disconnecting session	15 minutes
Audit the access of global system objects	Disabled
Audit use of Backup and Restore privilege	Disabled
Automatically log off users when logon time expires	Not defined
Automatically log off users when logon time expires (local)	Enabled
Clear virtual memory pagefile when system shuts down	Disabled
Digitally sign client communication (always)	Disabled
Digitally sign client communication (when possible)	Enabled
Digitally s	
Digitally s	
Disable C	
Do not dis	
LAN Mana	esponses
Message	
Message	
Number o	
Prevent s	
Prevent u	
Prompt us	
Recovery	
Recovery Console: Allow floppy copy and access to all drives and all ...	Disabled
Rename administrator account	Not defined
Rename guest account	Not defined
Restrict CD-ROM access to locally logged-on user only	Disabled

Template Security Policy Setting

Additional restrictions for anonymous connections

☑ Define this policy setting in the template:

No access without explicit anonymous permissions

OK | Cancel

File auditing is critical to maintaining file system security. Windows 2000 includes auditing capabilities that collect information about the use of file resources. For file auditing to function, system auditing must be enabled by configuring the auditing policy correctly.

On Windows 2000 systems, auditing is not enabled by default, and audit policies are set on a per-system basis via the Security Configuration Tool Set and the Security Templates snap-in. Each Windows 2000 system includes auditing capabilities that collect information about individual system usage. The logs collect information on applications, system, and security events. The three types of auditing are User Account, File System Auditing, and System Registry Auditing. Once System Auditing is configured, use the Security Templates snap-in to configure File Auditing. Windows 2000 Explorer can also be used to set File System Auditing.

Because of the importance of the Security Event Log in recording unauthorized accesses to the system, control of it should be limited to a select few. It is recommended that an “`Auditors`” group be created and given *Full Control* permissions to that log. Only individuals without administrator duties should be members of this group. This group should be given the User Right to “*Manage auditing and security log*” and the Administrators group should be removed from that right.

WARNING: Auditing can consume a large amount of processor time and disk space. It is highly recommended that administrators/auditors check, save, and clear audit logs daily/weekly to reduce the chances of system degradation or save audit logs to a separate machine. It is also recommended that logs be kept on a separate partition.

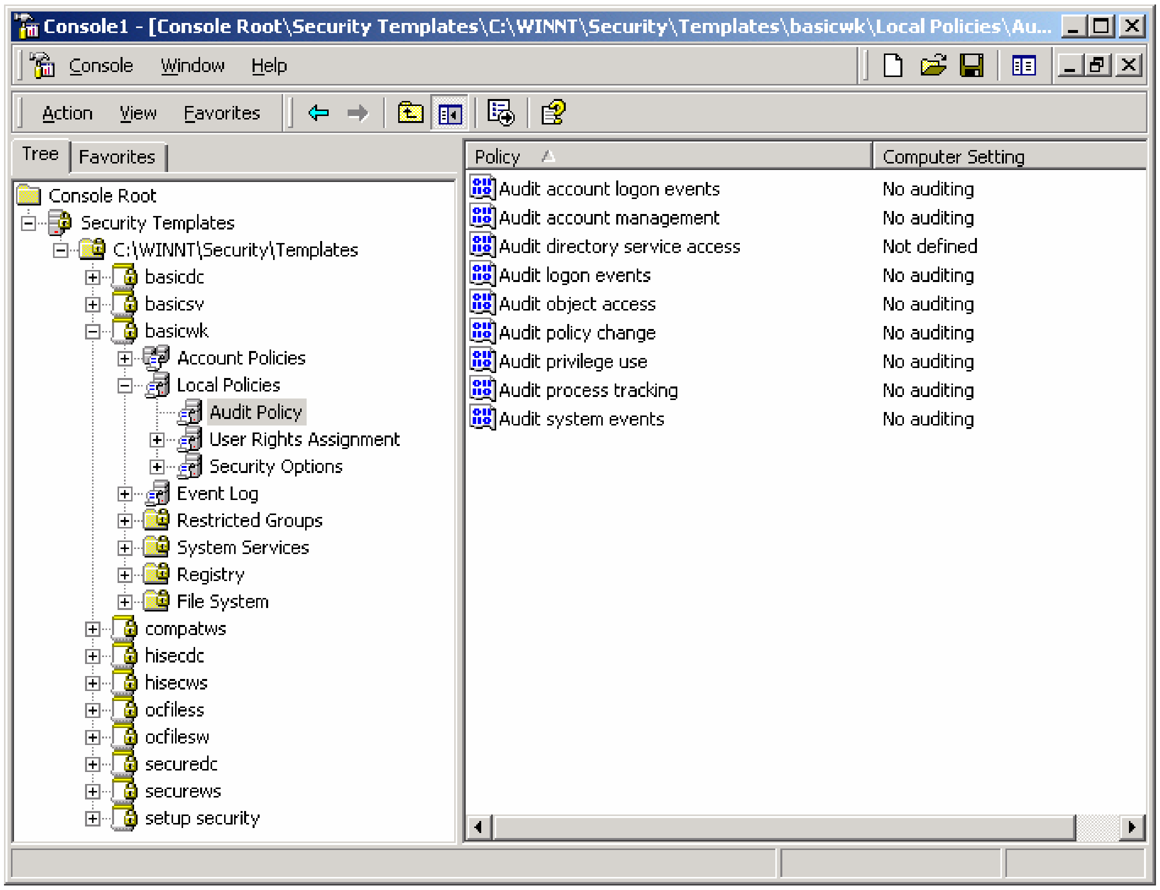
See the Security Configuration Toolset mini-guide for more information on setting audit policy.

Each event that is audited in an audit policy is written to the security event log. The security event log can be viewed with the Event Viewer.

Enable Audit Policy Setting To Capture File Events

To enable the audit policy setting to capture file events using the Security Templates Snap-in, do the following:

- Select the default file directory (*%SystemRoot%\Security\Templates*)
- Select the specific configuration file
- Select ***Local Policies* → *Audit Policy***
- Set *Audit Object Access* to enable auditing of failure events



Console1 - [Console Root\Security Templates\C:\WINNT\Security\Templates\basicwk\Local Policies\Au...

- Console Root
  - Security Templates
    - C:\WINNT\Security\Templates
      - basicdc
      - basicsv
      - basicwk
        - Account Policies
        - Local Policies
          - Audit Policy
          - User Rights Assignment
          - Security Options
        - Event Log
        - Restricted Groups
        - System Services
        - Registry
        - File System
      - compatws
      - hisecdc
      - hisecws
      - ocfiless
      - ocfilesw
      - securedc
      - securews
      - setup security

Policy	Computer Setting
Audit account logon events	No auditing
Audit account management	No auditing
Audit directory service access	Not defined
Audit logon events	No auditing
Audit object access	No auditing
Audit policy change	No auditing
Audit privilege use	No auditing
Audit process tracking	No auditing
Audit system events	No auditing

File System Auditing tracks a user's access to a specific directory or file. Auditing of sensitive files or directories may prove useful in identifying a system compromise or unauthorized use of resources.

View File System Settings

To view file system settings using the Security Templates Snap-in so the following:

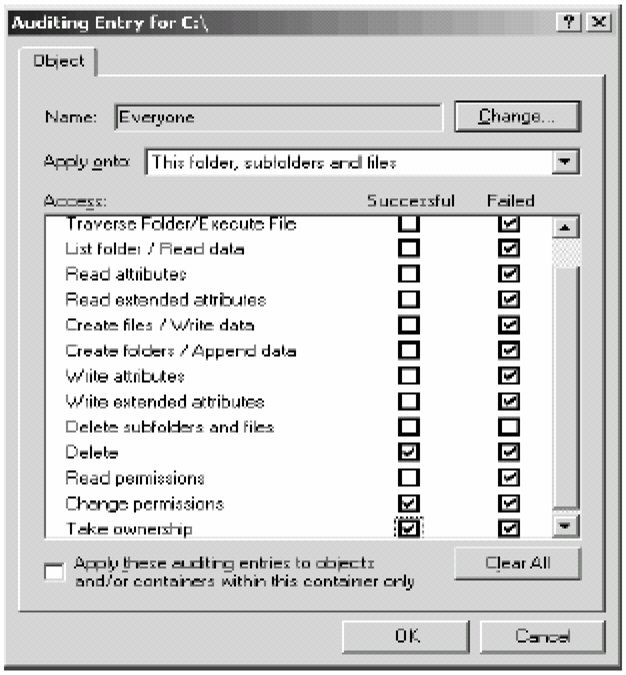
- Select the default file directory (*%SystemRoot%\Security\Templates*)
- Select the specific configuration file
- Select **File System**

Add Audit Settings on File or Folder

To add the audit settings on a particular file or folder through the Security Templates snap-in:

- In the right system object frame, double-click on the file or folder to be changed

- Ensure that the **“***Replace existing permissions on all subfolders and files with inheritable permissions***”** radio button is selected
- Click ***Edit Security***
- Click the *Advanced* button.
- Select the Auditing tab.
- Uncheck the “*Allow inheritable auditing entries from parent to propagate to this object***”** checkbox. Click on the *Remove* button in the *Security* dialog box.
- Click *Add* and select the group or user whose access will be audited
- Click *OK*.
- Select the desired audit settings. The figure below shows a sample auditing scheme.



Name: Everyone

Apply onto: This folder, subfolders and files

Access:	Successful	Failed
Traverse Folder/Execute File	☐	✓
List folder / Read data	☐	✓
Read attributes	☐	✓
Read extended attributes	☐	✓
Create files / Write data	☐	✓
Create folders / Append data	☐	✓
Write attributes	☐	✓
Write extended attributes	☐	✓
Delete subfolders and files	☐	☐
Delete	✓	✓
Read permissions	☐	✓
Change permissions	✓	✓
Take ownership	✓	✓

☐ Apply these auditing entries to objects and/or containers within this container only

- Click *OK*.
- Click *Apply→OK→OK→OK*

Modify Audit Settings on File or Folder

To modify the audit settings on a particular file or folder through the Security Templates snap-in:

- In the right frame, double-click on the file or folder to be changed
- Ensure that the **“***Replace existing permissions on all subfolders and files with inheritable permissions***”** radio button is selected
- Click *Edit Security*
- Click the *Advanced* button.
- Select the *Auditing* tab.
- Highlight the *Type* selection you want to modify.
- Click *View/Edit*.

- Edit the settings in the Audit Entry Window and click *OK*.
- Click *Apply→OK→OK→OK*

By default, only `Administrators` can eject removable NTFS media from the computer. Ensure that this option has not been changed.

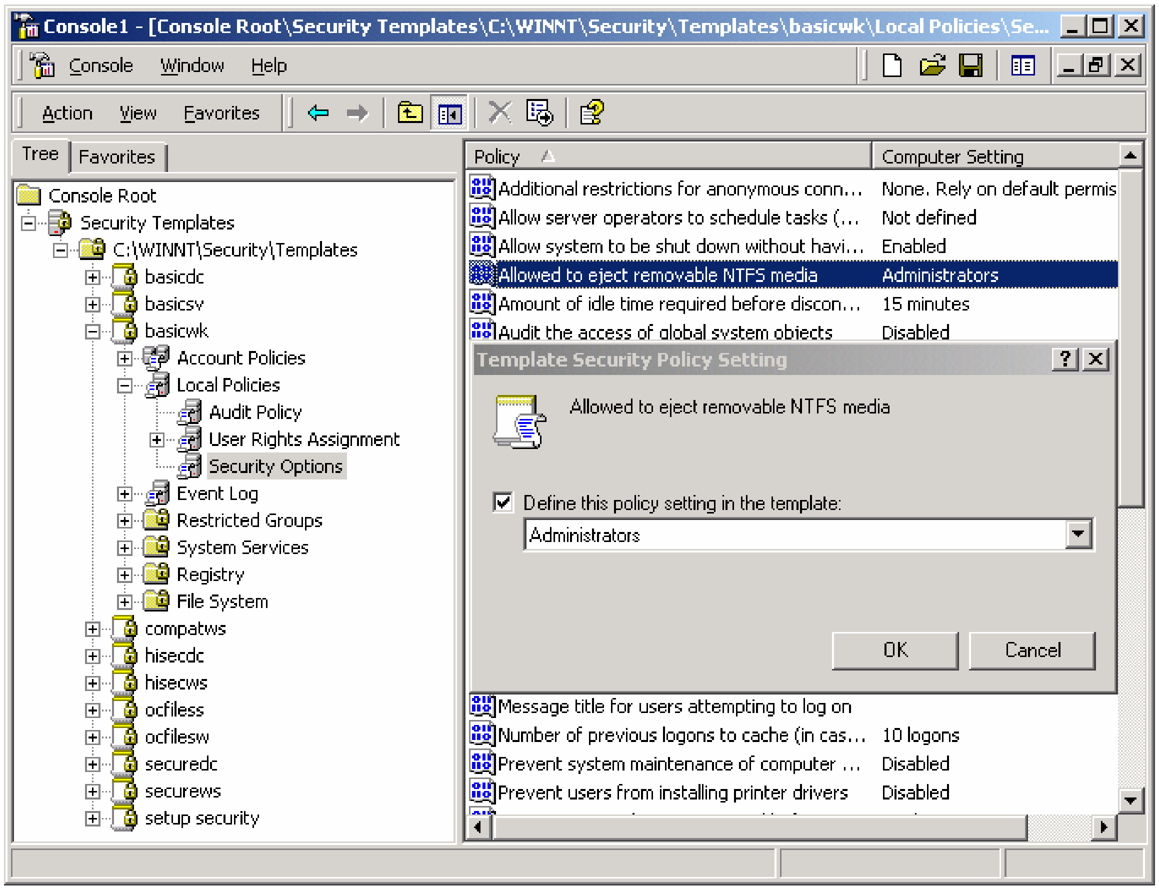
View Disk Resource Settings

To view related disk resource system setting for this option using the Security Templates Snap-in do the following:

- Select the default file directory (*%SystemRoot%\Security\Templates*)
- Select the specific configuration file

  Select *Local Policies*

  Select *Security Options*
- Ensure that the policy "*Allowed to eject removable NTFS media*" is set to `Administrators`



Console1 - [Console Root\Security Templates\C:\WINNT\Security\Templates\basicwk\Local Policies\Se...

Console Window Help

Action View Favorites

Tree | Favorites

- Console Root
  - Security Templates
    - C:\WINNT\Security\Templates
      - basicdc
      - basicsv
      - basicwk
        - Account Policies
        - Local Policies
          - Audit Policy
          - User Rights Assignment
          - Security Options
        - Event Log
        - Restricted Groups
        - System Services
        - Registry
        - File System
      - compatws
      - hisecdc
      - hisecws
      - ocfiless
      - ocfilesw
      - securedc
      - securews
      - setup security

Policy	Computer Setting
Additional restrictions for anonymous conn...	None. Rely on default permis
Allow server operators to schedule tasks (...	Not defined
Allow system to be shut down without havi...	Enabled
Allowed to eject removable NTFS media	Administrators
Amount of idle time required before discon...	15 minutes
Audit the access of global system objects	Disabled
Message title for users attempting to log on	
Number of previous logons to cache (in cas...	10 logons
Prevent system maintenance of computer ...	Disabled
Prevent users from installing printer drivers	Disabled

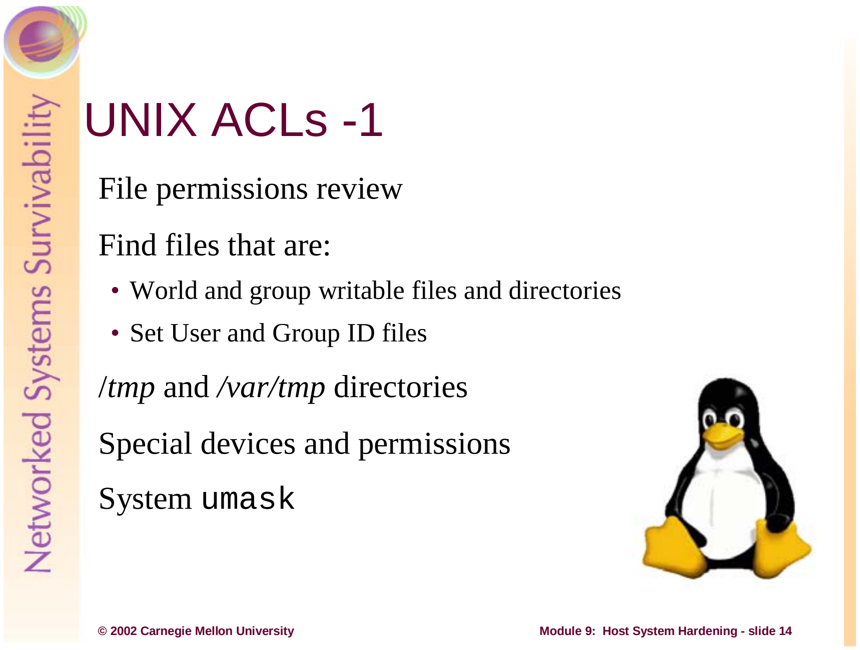
Template Security Policy Setting

Allowed to eject removable NTFS media

☑ Define this policy setting in the template:

Administrators

OK Cancel



Networked Systems Survivability

UNIX ACLs -1

File permissions review

Find files that are:

- World and group writable files and directories
- Set User and Group ID files

/tmp and */var/tmp* directories

Special devices and permissions

System `umask`

© 2002 Carnegie Mellon University Module 9: Host System Hardening - slide 14

The goals of adjusting ACLs on a UNIX-based file system are comparable to those on a Windows system, namely grant no access beyond what is minimally needed by user and applications. The challenge is again learning what access is appropriate. Before going into the security aspects, let's begin with a brief primer on UNIX file permissions, taken from [CTSSN 01].

Every file and directory in UNIX/LINUX has read, write, and execute permissions. Read permission means that the file can be read but not modified or deleted. Write permission means that the file can be created, modified, or deleted. Execute permission means that the file can be executed. The execute permission is similar to a .exe file in Windows or DOS. For directories, execute permission means you can open the directory.

Permissions
There are 3 sets of permissions for every file or directory – owner, group, and world. For each set, there are separate read, write, and execute permissions. The owner permissions are for the owner of the file or directory. The group permissions are for everyone in the group. The global permissions are for anyone. To see the current permissions, owner, and group for a file or directory, type the following commands:

```
cd
ls -l
```

This will display the contents of your home directory in long format.

The following is an example of what you might have seen with the above 2 commands:

```
% cd
% ls –l
drwxr--r-- 5 you public 4096 Feb 7 14:33 Desktop
drwxr--r-- 2 you public 4096 Apr 4 10:55 linuxpractice
-rwxr-xr-x 1 you public  223 Apr 6 11:37 myfirstscript
drwxr-xr-x 4 you public 4096 Apr 9 12:03 public_html
```

The third column (`you`) tells the owner of the file or directory, and the fourth column (`public`) is the name of the group for the file or directory. The permissions are listed in the first column. The first letter is whether the item is a directory or a file. If the first letter is a "`d`", then the item is a directory as in the first item listed above, *Desktop*. Notice, for the file *myfirstscript*, the first letter is "`-`". The next three letters are the permissions for the owner of the file, the next three letters apply to everyone in the group, and the last three letters are for everyone else. The read, write, and execute permissions are referred to as `r,w`, and `x` respectively. Thus, for the directory *Desktop* above, the owner `you` has read, write, and execute permissions to the directory *Desktop*, everyone in the group `public` has read permissions, and everyone else has read permissions. The only one who can modify or delete any file in this directory is the owner you (or the superuser- "`root`").

chmod(1) is a standard UNIX/LINUX command that allows you to change the permissions of a file or directory. There are two arguments for **chmod**: the permissions and the file/directory name. The permission argument for **chmod** is based on numbers.

`1` stands for execute
`2` stands for write
`4` stands for read

To set more than one permission on a file or directory, you just add up the permissions. For example, 7 means read, write, and execute permissions. **chmod** takes the permissions as the first argument in the order user, group, global. Thus, the command `chmod 777 hello` will change the permissions of the file *hello* to read, write and execute by user, group, and everyone else. Note: To change the permissions of a file/directory, you must be the owner of that file/directory. However, `root` can change permissions on any file or directory.

One last topic that we should touch on here is the ownership commands. If you are `root`, you can change the ownership of files using the `chown` *`username filename`* command. As with most commands, there are many options. You can learn about those options by typing `man chown.`

Examples

- `chmod 777 Desktop` - allows all logged in users full access to read, write and execute
- `chmod 644 Desktop` - all the owner (you) read and write access, the group read access, and everybody else (world) read access too
- `chmod 700 Desktop` - gives yourself full access while giving everybody else no access.

As distributed, most UNIX-based systems are fairly secure, that is to say that the permissions are mostly appropriate for secure operation. However, there are some specific permissions to look at to improve the level of security. Most of them can be shown with the **find**(1) program. The **find** program traverses the file system searching for files and directories that match a criterion. To that end, it is good to know how to operate the **find** program. Here are some examples:

First, let's find all files and directories in a file system that are group- or world-writable. Files with these permissions may allow writing by more users than those intended. As the administrator, you must list these files and then decide about adjusting the permissions. To help you find these files, use the following commands:

```
/bin/find / -type f \( -perm -2 -o -perm -20 \) -exec ls -lg {} \;
/bin/find / -type d \( -perm -2 -o -perm -20 \) -exec ls -ldg {} \;
```

The first command finds all files starting at the root of the file system (/) that are regular files (`-type f`) and that have either the world writable (`-perm –2`) or the group writable bit (`-perm –20`) turned on. If such a file is found, the list files (`ls`) program is executed with the options to provide a long listing and the group ownership of the file (`-lg`). The brackets (`{}`) are replaced by the matching file name, and the escaped semi-colon (`\;`) ends the command that **find** executes upon a match.

In the second command, **find** searches for directories instead of files. Also, the way in which a match is listed is different. The command does not list the directory's contents (`-ldg` instead of `–lg`).

By finding all files and directories that are either group- or world-writable, the administrator can adjust their ACLs so their permissions are not as lenient as they would have been had they not been adjusted. The challenge here is to know what permissions are appropriate for a program to operate correctly.

Similarly, finding all Set User and Group ID executable files provides a list of programs that take on additional permissions when they run. Set User and Group ID executable files take on the identity as specified in the User or Group ID fields of the file. In this way, temporary permissions can be given to a user and those permissions last for the duration of the execution of that program. This is a simple privilege-granting technique that achieves its goal – granting temporary permissions with a limited lifetime – at the expense of security. The burden on using those privileges properly lies with the programmer. Unfortunately, Set User and Group ID executables are frequent targets for intruders. It is strongly suggested that the list of files that have either Set User or Group ID privileges be reviewed often to see if programs with these extra permissions truly need those permissions. The command to execute in this case is the following or a variant:

```
/bin/find / -type f \( -perm -004000 -o -perm -002000 \) \
        -exec ls -lg {} \;
```

This command files all regular files, starting at the root of the file system, where the file in question has either the Set User ID (`-perm –004000`) or Set Group ID (`-perm –002000`) bits turned on. Those found are listed for the administrator's perusal.

The */tmp* and */var/tmp* (aka */usr/tmp*) directories are somewhat special in that they are intended to be group- and world-writable. To improve their security, an additional permission bit takes on special meaning when applied to a directory. This bit is called the sticky bit, a remnant from the early days of UNIX systems. It was originally only intended for regular files, but when applied to directories, it has the meaning that says that only the owner of the file can remove the file, even though the directory permissions would otherwise allow it by being group- or world-writable. This means that the user who created them can only remove the files in those system-wide temporary directories. To enable this capability in a directory, do the following:

```
chmod 1777 directory
```

The 1 enables the stick bit and the 777 sets the permissions to all operations are allowed.

Contained in the device directory (*/dev*) should be special files – either block- or character-special – and perhaps some directories and symbolic link files. Special files allow users to gain access to devices on a system, such as disk, memory, a CDRom, and a floppy. Care must be taken on the files in these directories because permissive settings allow a user to bypass the controls placed on devices and operate on those devices directly. For example, if the character- and block-special entries for a disk allow reading by all, then the file system access controls enforced by the file system service in the operating system can be bypassed. It is as though there are no ACLs on any file on the disk.

Finally, the **umask**(1) program defines the set of permissions that are withheld when new files and directories are created. Care must be taken so that newly created files and directories have only the permissions needed and nothing more. Consult your system documentation on how to establish these values. For example, on a RedHat LINUX 7.1 system, the default **umask** value is set in */etc/rc.d/init.d/functions*, a shell script run by every system service. The value established by that script is 022, meaning that no file or directory created by the system will be group- or world-writable.

Understanding what permissions are correct is a challenge, as has been previously started. To help the administrator, the AusCERT UNIX Checklist [AUSCERT 01] gives several recommendations of permissions to set and tasks to do when adjusting ACLs on a UNIX/LINUX-based system. Here are highlights from that document that reflect ACLs. Note that on the specific system being hardened, some of the pathnames may be different.

- ENSURE that the permissions of */etc/utmp* and */var/adm/wtmp* are set to 644.
- ENSURE that the permissions of */etc/motd* and */etc/mtab* are set to 644.
- ENSURE that the permissions of */etc/syslog.pid* (*/var/run/syslog.pid* on some systems) are set to 644. [NOTE: This may be reset each time you restart **syslog**.]
- CONSIDER removing read access to system configuration files that users do not need to access, for example `rc.*` startup files and authentication files like */etc/hosts.allow*.
- ENSURE that logfiles (usually in */var/log/*) are only writable by root.
- CONSIDER using file system security features if your operating system supports them - system binaries may be made immutable, log files append-only.
- ENSURE that the kernel (e.g., */vmunix* or */boot/vmlinuz*) is owned by root, has group set to 0 (wheel on SunOS) and permissions set to 644.
- ENSURE that */etc*, */usr/etc*, */bin*, */usr/bin*, */sbin*, */usr/sbin*, */tmp* and */var/tmp* are owned by root and that the sticky-bit is set on */tmp* and on */var/tmp*. Do the following:

  ```
  /bin/chown root /tmp
  /bin/chgrp 0 /tmp
  /bin/chmod 1777 /tmp
  ```

  Refer to the AusCERT Advisory:

  ftp://ftp.auscert.org.au/pub/auscert/advisory/AA-95.07.Incorrect.Permissions.on.tmp.may.allow.root.access
- ENSURE that there are no unexpected world writable files or directories on your system. The following commands find group and world writable files and directories.

  ```
  /bin/find / -type f \( -perm -2 -o -perm -20 \) -exec ls -lg {} \;
  /bin/find / -type d \( -perm -2 -o -perm -20 \) -exec ls -ldg {} \;
  ```

- ENSURE that files that have the SUID or SGID bit set, need to have it that way. Remove this bit from programs that do not require elevated privileges to function successfully.
- ENSURE the **umask** value for each user is set to something sensible like 027 or 077. (The shell script below checks this).

  ```
  #!/bin/sh
  PATH=/bin:/usr/bin:/usr/etc:/usr/ucb
  HOMEDIRS=`cat /etc/passwd |
          awk -F":" 'length($6) > 0 {print $6}' | sort -u`
  FILES=".cshrc .login .profile"
  ```

```
for dir in $HOMEDIRS
do
    for file in $FILES
    do
    grep -s umask /dev/null $dir/$file
    done
done
```

- ENSURE all files in */dev* are special files. Special files are identified with a letter (usually *c* for 'character' or *b* for 'block') in the first position of the permissions bits. The following command finds files in */dev* which are not special files or directories. [NOTE: Some systems have directories and a shell script in */dev* which may be legitimate.] Please check the manual pages for more information.

```
# /bin/find /dev -type f -exec ls -l {} \;
```

- ENSURE that there are no unexpected special files outside `/dev`. To find any block special or character special files, use the following command:

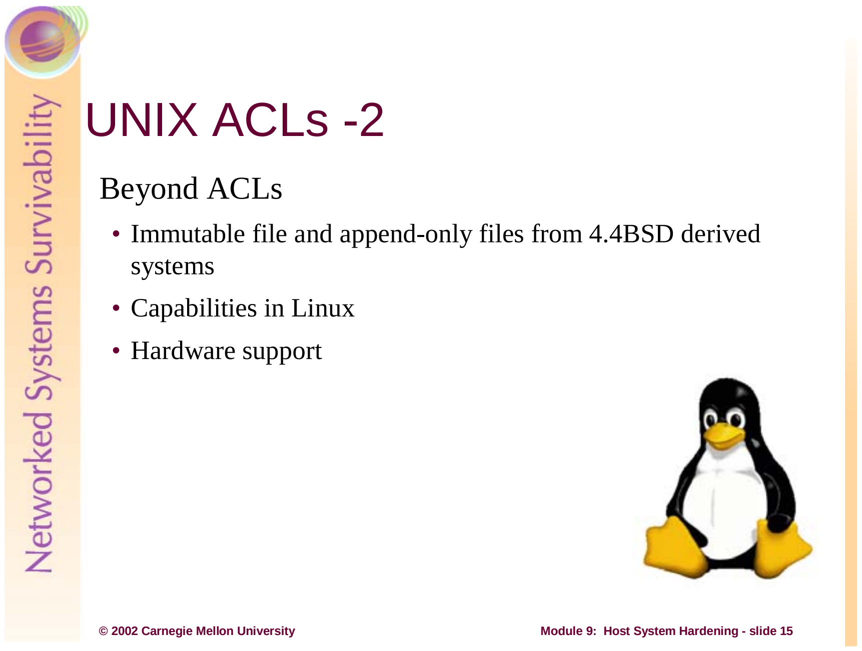
```
# /bin/find / \( -type b -o -type c \) -print | \
   grep -v '^/dev/'
```

AusCERT recommends that any script or binary to be executed by root should be owned by root and should not be world or group writable. Additionally, this file should only be located in a directory for which every parent directory is owned by root and is not group or world writable.

- CHECK the contents of the following files for the root account. Any programs or scripts referenced in these files should meet the above requirements:
    - *~/.login*, *~/.profile* and similar login initialization files
    - *~/.exrc* and similar program initialization files
    - *~/.logout* and similar session cleanup files
    - **crontab** and **at** entries
    - files on NFS partitions
    - */etc/rc** and similar system startup and shutdown files
- If any programs or scripts referenced in these files source further programs or scripts they also need to be verified.

Many systems ship files and directories owned by the `bin` (or `sys`) users. This varies from system to system and may have serious security implications.

- CHANGE all non-setuid files and all non-setgid files and directories that are world readable but not world or group writable and that are owned by bin to ownership of root, with group id 0.Please note that under SOLARIS 2.X changing ownership of system files can cause warning messages during installation of patches and system packages. One utility available to help Solaris administrators avoid this problem is **fix-modes** [SUN 01]. Any other values should be verified with the vendor.



UNIX ACLs -2

Beyond ACLs

- Immutable file and append-only files from 4.4BSD derived systems
- Capabilities in Linux
- Hardware support

Networked Systems Survivability

© 2002 Carnegie Mellon University

Module 9: Host System Hardening - slide 15

There are other types of access controls too. Among them are:

- Immutable files: Files marked immutable may not be written to, even by root or Administrator, regardless of the permissions. Immutable files are available on some versions of LINUX and derivatives of 4.4BSD UNIX.

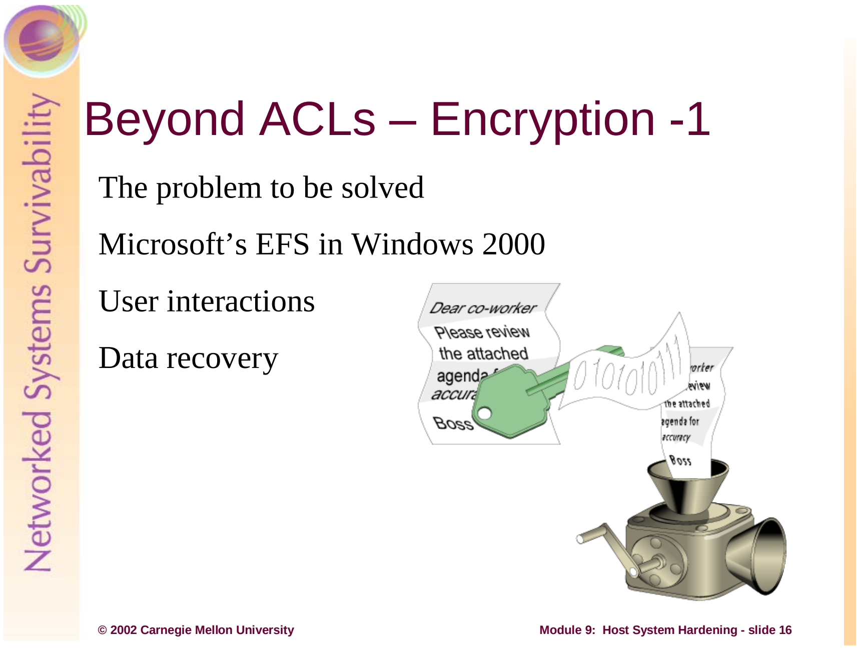
  Not all immutable capabilities are the same. With the 4.4BSD strain of UNIX systems, immutable means that no one – repeat NO ONE – can write to files except when the system is in a maintenance mode, often referred to as single-user mode. In that case, the issues of physical security come into play because the administrator needs to be physically at the console of the computer system. The security of that device can be improved by age-old physical security methods.

  On LINUX systems, files can be marked immutable with the **chattr** program, but the root user can change that attribute. This capability only prevents the accidental overwriting of a file's contents but does not defeat the more skilled intruder. So, immutable does not mean the same thing in all settings.

- Append-only files: Files marked append-only may be written to only in append mode. Log files are good candidates for append-only files, allowing only new log entries to be added. Append-only files are also available on some versions of LINUX and derivatives of 4.4BSD UNIX.

  Again, the administrator needs to determine how append-only is implemented. On some LINUX systems, root can change that attribute as with immutable files above, so be aware.

- Hardware protected or read-only media: Files residing on disks that have a write-protect mechanism or on read-only media, CDROM for example, use hardware access controls. Since these features are available in hardware, all operating systems can benefit from them.



Beyond ACLs – Encryption -1

The problem to be solved

Microsoft’s EFS in Windows 2000

User interactions

Data recovery

Networked Systems Survivability
Dear co-worker
Please review
the attached
agenda for
accuracy
Boss
010101011

© 2002 Carnegie Mellon University Module 9: Host System Hardening - slide 16

When an asset is vitally important to an organization’s mission, it can be further secured through encryption. Encryption obscures information by mathematically changing that information. If that changed form falls into the wrong hands, it is of no use without access to the algorithms and the keys used to perform the encryption since the algorithms are normally available. The challenge then is to safeguard the keys used by the encryption algorithms. In fact, since encryption algorithms have become so computationally complex, intruders have turned their efforts to stealing the keys and away from brute-force cracking.

Examples of encryption schemes are DES, the Data Encryption Standard [TropSoft 01], and AES, the Advanced Encryption Standard [NIST 01]. Indeed, entire disk drives can be obscured from view by encrypting a file system. Windows 2000 supports the Encrypting File System (EFS) [WinNTMag 99], [WinNTMag 00] and there are many similar file systems for LINUX and UNIX-based operating systems [LinuxJournal 98]. This section describes EFS for Windows 2000 and it is derived from [Microsoft 00].

The Encrypting File System (EFS) included with the Microsoft Windows 2000 operating system is based on public-key encryption and takes advantage of the CryptoAPI [Microsoft 01b] architecture in Windows 2000. Each file is encrypted using a randomly generated file encryption key, which is independent of a user’s public/private key pair; thereby stifling many forms of cryptanalysis-based attack.

File encryption can use any symmetric encryption algorithm. This release of EFS uses the Data Encryption Standard X, or DESX (128 bit in North America and 40 bit International), as the encryption algorithm. Future releases will allow alternate encryption schemes.

EFS supports encryption and decryption of files stored on local drives as well as those stored on remote file servers. It is very important to remember that EFS is only a component of the NTFS v.5 file system.

Note: *In the case of remote servers, you can encrypt files and folders on the server but your data is not protected if you access a file over the network. Windows 2000 provides network protocols such as Secure Sockets Layer/Private Communication Technology (SSL/PCT) and IPSEC to encrypt data access over the network.*

User Interaction

The default configuration of EFS allows users to start encrypting files with no administrative effort. EFS automatically generates a public-key pair and file encryption certificate for file encryption for a user the first time a user encrypts a file.

File encryption and decryption is supported per file or for an entire folder. Folder encryption is transparently enforced. All files (and folders) created in a folder marked for encryption are automatically encrypted. Each file has a unique file encryption key, making it safe to rename. If you rename a file from an encrypted folder to an unencrypted folder on the same volume, the file remains encrypted. However, if you copy an unencrypted file into an encrypted folder, the file remains in the state that it was in—in this case the file remains unencrypted. Command-line tools and administrative interfaces are provided for advanced users and recovery agents.

You don't have to decrypt a file to open it and use it. EFS automatically detects an encrypted file and locates a user's file encryption key from the system's key store. Since the key storage mechanism is based on CryptoAPI, in the future, users will have the flexibility of storing keys on secure devices, such as smart cards.

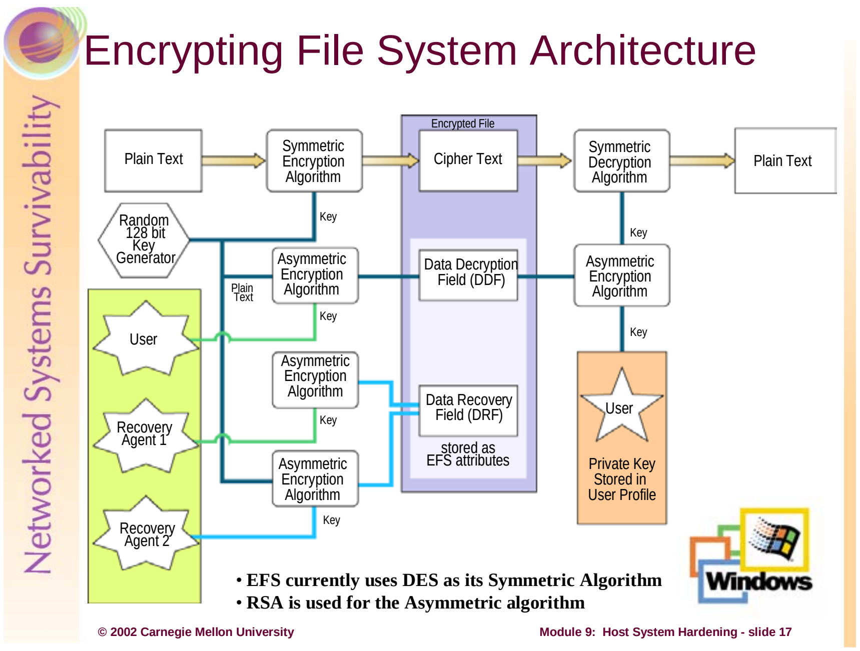
The initial release of EFS does not support file sharing. However, the EFS architecture is designed to allow file sharing between any number of people by the simple use of their file encryption certificates. Users can then independently decrypt files using their own private keys. Users can be added easily (if they have a configured public key pair) or removed from a group of permitted sharers.

Data Recovery

EFS provides built-in data recovery support. The Windows 2000 security infrastructure enforces the configuration of data recovery keys. You can use file encryption only if the system is configured with one or more recovery keys. EFS allows recovery agents to configure public keys that are used to recover encrypted data if a user leaves the company. Only the file encryption key is available using the recovery key, not a user's private key. This ensures that no other private information is revealed to the recovery agent.

Data recovery is intended for business environments where the organization expects to be able to recover data encrypted by an employee after an employee leaves or if file encryption keys are lost.

EFS automatically generates recovery keys and self-signed certificates when the local administrator logs on, making the local administrator the default recovery agent. Users can also use the command line tool to recover data using the administrator's account. This reduces the administrative overhead for a user.



Encrypting File System Architecture

Networked Systems Survivability
Plain Text
Symmetric Encryption Algorithm
Encrypted File
Cipher Text
Symmetric Decryption Algorithm
Plain Text
Random 128 bit Key Generator
Key
Key
Plain Text
Asymmetric Encryption Algorithm
Data Decryption Field (DDF)
Asymmetric Encryption Algorithm
Key
Key
User
Asymmetric Encryption Algorithm
Data Recovery Field (DRF)
User
Key
stored as EFS attributes
Private Key Stored in User Profile
Recovery Agent 1
Asymmetric Encryption Algorithm
Key
Recovery Agent 2
Windows

- EFS currently uses DES as its Symmetric Algorithm
- RSA is used for the Asymmetric algorithm

© 2002 Carnegie Mellon University Module 9: Host System Hardening - slide 17

The following is excerpted from the Encrypting File System for Windows 2000 white paper:[1]

EFS implements data encryption and decryption, using a public key–based scheme. File data is encrypted using a fast symmetric algorithm with a file encryption key (FEK). The FEK is a randomly generated key of a certain length required by the algorithm, or by law if the algorithm supports variable length keys.

The FEK is encrypted using one or more key encryption public keys to generate a list of encrypted FEKs. The public portion of a user's key pair is used to encrypt FEKs. The list of encrypted FEKs is stored along with this encrypted file in a special EFS attribute called the Data Decryption Field (DDF). The file encryption information is tightly bound to the file. The private portion of the user's key pair is used during decryption. The FEK is decrypted using the private portion of the key pair. The private portion of a user's key pair is stored safely elsewhere in smart cards or other secure storage devices.

The FEK is also encrypted using one or more recovery key encryption public keys. Again, the public portion of each key pair is used to encrypt FEKs. This list of encrypted FEKs is also stored along with the file in a special EFS attribute called the Data Recovery Field (DRF). Only public portions of the recovery key pairs are needed for encryption of the FEK in the DRF. These public recovery keys are required to be present at all times on an EFS system for normal file system operations. Recovery itself is expected to be a rare operation required only when users leave organizations or lose keys. Because of this, recovery agents can store the private portions of the keys safely elsewhere (on smart cards and other secure storage devices).

The user's plaintext file is encrypted using a randomly generated FEK. This file encryption key is stored along with the file, encrypted under a user's public key in the DDF and encrypted under the recovery agent's public key in the DRF.

A user's private key is used to decrypt the FEK, using the corresponding encrypted FEK item in the DDF. The FEK is used to decrypt file data reads on a block-by-block basis. Random access to a large file decrypts only the specific blocks read from disk for that file. The entire file does not have to be decrypted.

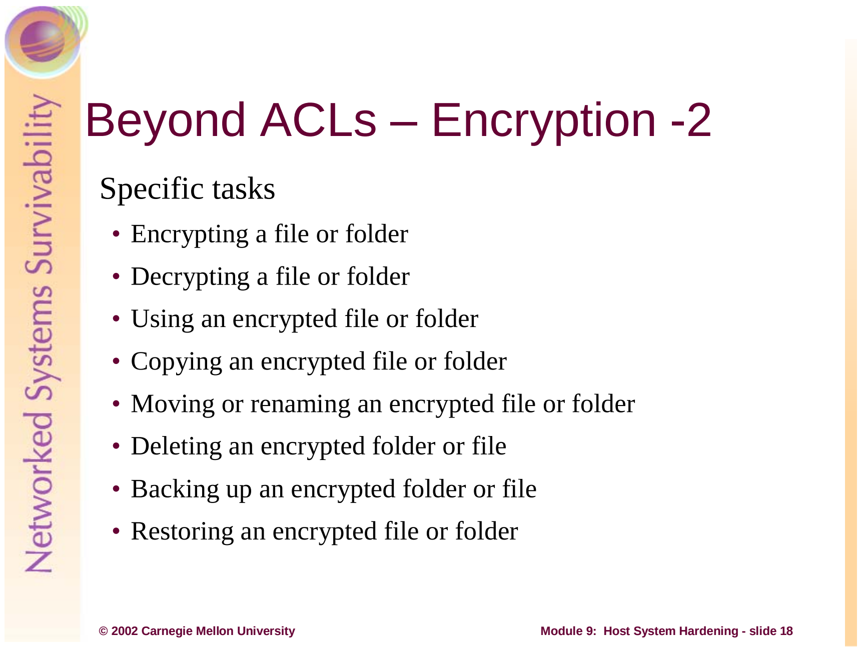
[1]http://www.microsoft.com/technet/treeview/default.asp?url=/technet/prodtechnol/windows2000serv/deploy/confeat/nt5efs.asp

EFS Architecture Key points:

- Uses both symmetric and asymmetric cryptography
- A symmetric key encrypts each file
- Copies of this symmetric key is stored as a file attribute
    - An encrypted copy of the symmetric key for each security principal (user and each recovery agent)
    - Encrypted with principal’s public key
- Principal’s private key is used to decrypt the appropriate copy of the symmetric key which can then decrypt the file

EFS Recovery Agent Key Points:

- Generally, only the user who encrypts a file should be able to decrypted it
- Recovery agents can be enabled that allow authorized security principals to decrypt any user’s files
    - Useful when user decryption process fails
    - Useful when user leaves an organization
- Normally, the domain admin and/or the local admin are designated as recovery agents



Beyond ACLs – Encryption -2

Specific tasks

- Encrypting a file or folder
- Decrypting a file or folder
- Using an encrypted file or folder
- Copying an encrypted file or folder
- Moving or renaming an encrypted file or folder
- Deleting an encrypted folder or file
- Backing up an encrypted folder or file
- Restoring an encrypted file or folder

Networked Systems Survivability
© 2002 Carnegie Mellon University
Module 9: Host System Hardening - slide 18

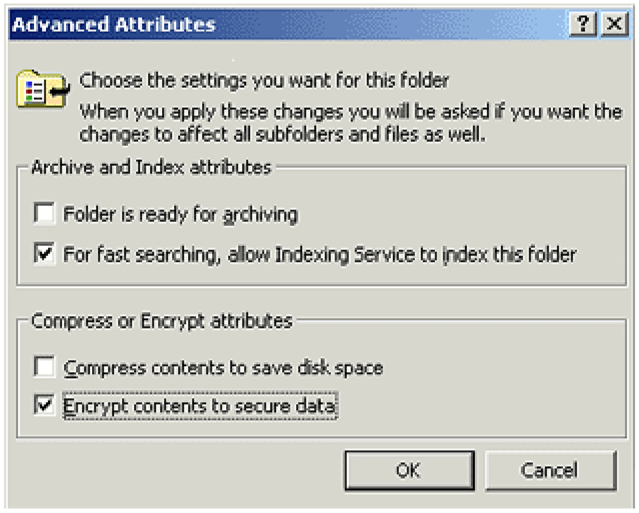
The following sections detail how to work with EFS on specific operations and is also derived from [Microsoft 00].

Encrypting a File or Folder

When encrypting a folder or file, you can use Windows Explorer or you can use the command-line utility, **cipher.exe**. Both procedures are described next. It is assumed that these steps are being performed on a computer running Windows 2000 Professional.

To use Windows Explorer to encrypt a file or folder, do the following:

1. Click **Start**, point to **Programs**, point to **Accessories**, and click **Windows Explorer**.
2. Right-click the folder or file name you wish to work with (in this example a folder we created under **My Documents** called **Encrypted Files**), and choose **Properties**.
3. On the **General** tab in the **Encrypted Files Properties** dialog box, click **Advanced**.



Advanced Attributes

Choose the settings you want for this folder

When you apply these changes you will be asked if you want the changes to affect all subfolders and files as well.

Archive and Index attributes

- [ ] Folder is ready for archiving
- [✓] For fast searching, allow Indexing Service to index this folder

Compress or Encrypt attributes

- [ ] Compress contents to save disk space
- [✓] Encrypt contents to secure data

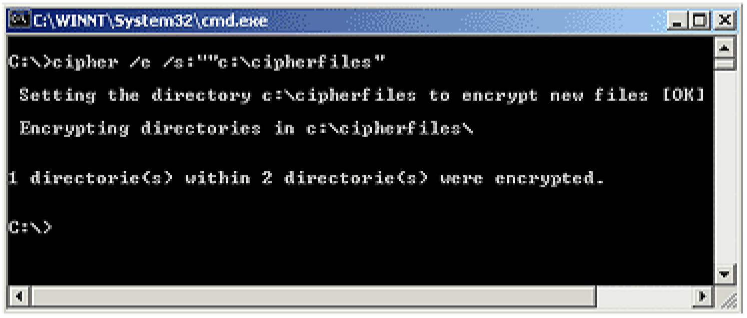
4. On the Advanced Attributes dialog box, select Encrypt contents to secure data and click OK. This returns you to the Encrypted Files Properties dialog box.
5. Click OK in the Encrypted Files Properties dialog box.
6. You are asked to choose between encrypting the folder and all its contents or just the folder itself. If the folder is empty, choose to encrypt the folder only; otherwise, choose the folder and its contents, and click OK.
7. A dialog box shows you the status of encrypting the folder or file. Click **OK** again to make this change, and close the snap-in.

To use **cipher.exe** to encrypt a folder or file, do the following:

1. To encrypt the folder, **D:\Encrypted Files** (substitute this with the drive letter and folder that you are working with), click **Start**, click **Run**, type **cmd** and click **OK**. For example, at the command prompt, type:

   ```
   D:\>cipher /e /s:"D:\Encrypted Files"
   ```

2. Press **ENTER**.



```
C:\>cipher /e /s:""c:\cipherfiles"
 Setting the directory c:\cipherfiles to encrypt new files [OK]
 Encrypting directories in c:\cipherfiles\

1 directorie(s) within 2 directorie(s) were encrypted.

C:\>
```

The folder's attributes report that this folder is encrypted.

Decrypting a File or Folder

As with encryption, you can use Windows Explorer or a command-line utility to decrypt a folder or file. Both procedures are described next. Note that you do not need to decrypt a file to open the file and edit it. Decrypt a file that you want to make accessible to others.

To use Windows Explore to decrypt a folder or file, do the following:

1. Click **Start**, point to **Programs**, point to **Accessories**, and select **Windows Explorer**.
2. Right-click the folder or file name, and choose **Properties**.
3. On the **General** tab of the **Properties** dialog box, click **Advanced**.
4. On the **Advanced Attributes** dialog box, clear the **Encrypt contents to secure data** check box, and click **OK**.
5. Click **OK** in the **Encrypted Files Properties** dialog box.
6. You are asked to choose between decrypting the folder and all its contents or just the folder itself. The default is to decrypt the folder only, and then click **OK**.

Note: *It is recommended that you encrypt folders and not individual files. This is because many existing applications are not aware of encryption and can therefore render the file in clear text.*

To use **cipher.exe** to decrypt a folder or file, do the following:

1. To decrypt the folder, D:\Encrypted Files, type

```
C:\>cipher /d /s:"C:\Encrypted Files"
```

2. Press **Enter**.

Using an Encrypted File or Folder

If you are the user who encrypted a file, you can use that file as before. You can open, edit, copy, and rename the file. If you are *not* the user who encrypted the file, you cannot do any of those things and you receive an *Access Denied* error message if you attempt to access the file. Only the user who encrypted the file and the recovery agent can decrypt it—making it available again to others in its unencrypted (plain text) form.

With an encrypted folder, if you had access to that folder before it was encrypted, you can still open it. Folders are only marked as encrypted so that all files in them are encrypted as they are created, and sub-folders are marked encrypted at creation.

Note: *An encrypted folder can be decrypted only by the user who encrypted it.*

Copying an Encrypted File or Folder

The following explains the procedures and limitations for copying encrypted folders or files on the same volume and from one volume to another.

- **To copy a file or folder on the same computer from one NTFS partition in a Windows 2000 location to another NTFS partition in a Windows 2000 location**. Copy the file or folder as you would an unencrypted file. Use Windows Explorer or the command prompt. The copy is encrypted.
- **To copy a file or folder on the same computer from an NTFS partition in a Windows 2000 volume to a FAT partition**. Copy the file or folder as you would an unencrypted file. Use Windows Explorer or the command prompt. Because the destination file system does not support encryption, the copy is in clear text.
- **To copy a file or folder to a different computer where both use the NTFS partitions in Windows 2000**. Copy the file or folder as you would an unencrypted file. Use Windows Explorer or the command prompt. If the destination is a location where encryption is supported and trusted, then the copy is encrypted. Otherwise, it is stored in unencrypted.
- **To copy a file or folder to a different computer from an NTFS partition in a Windows 2000 location to a FAT or NTFS in a Windows NT® 4.0 location**. Copy the file or folder as you would an unencrypted file. Use Windows Explorer or the command prompt. Because the destination file system does not support encryption, the copy is in clear text.

Note: *If your original file was encrypted, Microsoft recommends that you use the File Properties, Advanced option to confirm the status of the destination file.*

Moving or Renaming an Encrypted File or Folder

The following explains the procedures and limitations for moving encrypted folders or files on the same volume and from one volume to another.

- **To move or rename a file or folder within the same volume**. Move the file as you would an unencrypted file. Use Windows Explorer, the context menu, or the command prompt. The destination file or folder remains encrypted.
- **To move a file or folder between volumes**. This is essentially a copy operation. Review the previous section, *Copying an Encrypted Folder or File.*

Deleting an Encrypted Folder or File

If you have access to delete the file or folder, you can delete it as you could an unencrypted file.

Note: *Deleting an encrypted folder or file is not restricted to the user who originally encrypted the file.*

Backing Up an Encrypted Folder or File

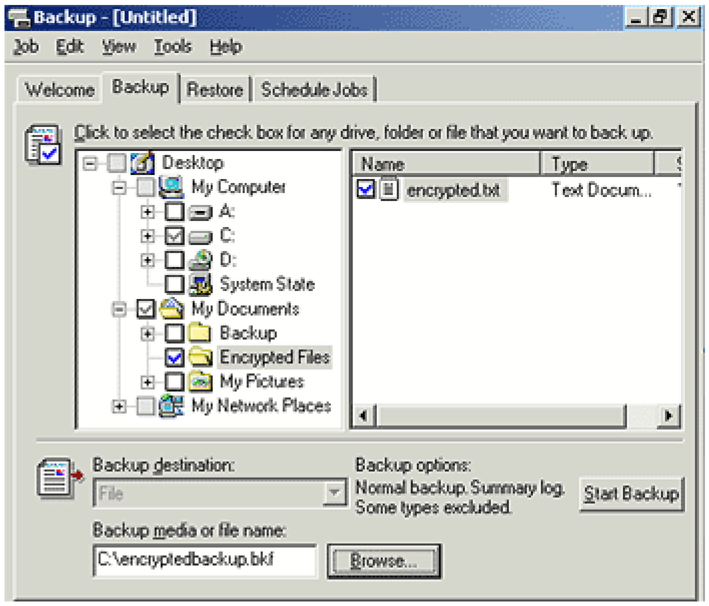
The following explains the procedures and limitations for backing up encrypted folders or files.

Backing up by copying. Backup created using the **Copy** command or menu selection can end up in clear text, as explained previously in the section, **Copying an Encrypted Folder or Folder**.

Backing up using Backup in Windows 2000 or any backup utility that supports Windows 2000 features. This is the recommended way to back up encrypted files. The backup operation maintains the file encryption, and the backup operator does not need access to private keys to do the backup; they only need access to the file or folder to complete the task.

To use Backup to back up a file, folder, or drive, do the following

1. Click **Start**, point to **Programs**, point to **Accessories**, point to **System Tools**, and then click **Backup**. The **Backup** wizard appears.
2. Click the **Backup** tab.
3. Select the drive, files, or folders that you want to back up. (in this case **My Documents\Encrypted Files**).



4. Select the destination in the Backup media or file name list. Click **Browse** to locate a pre-existing backup file.

5. Click **Start Backup**.
6. In the **Backup Job Information** dialog box, make selections, and then click **Start Backup**. When the backup process is complete, click **Close** in the **Backup Progress** dialog box.

Backup copies the entire encrypted file, folder, or drive to the backup file you selected. This file can be copied to FAT media, such as floppy disks, and is secure because its contents remain encrypted.

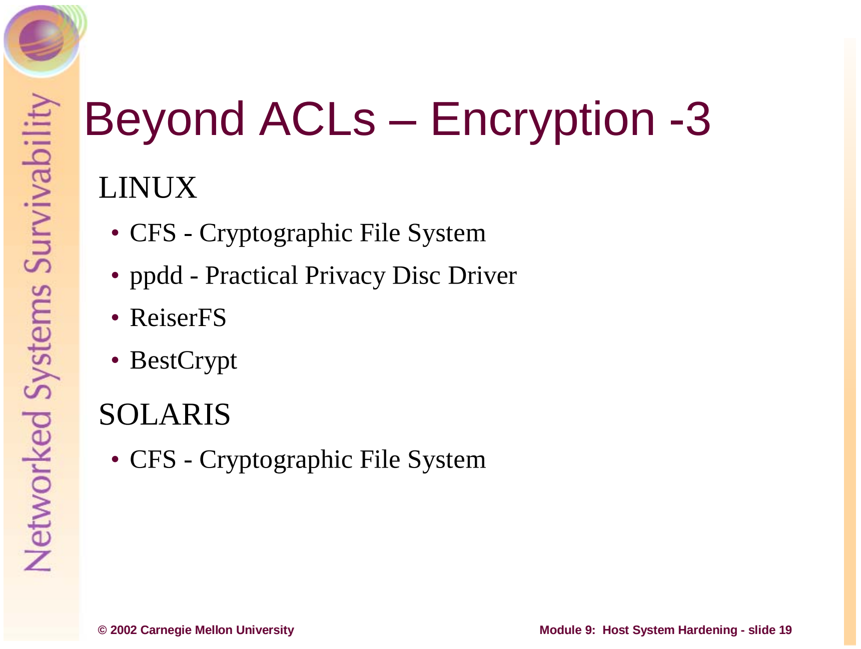
Restoring an Encrypted File or Folder

Restore operations parallel those used for backing up encrypted files. The following explains the procedures and limitations for restoring backed up encrypted files to the computer where the backup was performed and to a computer other than the one where the files were backed up.

Restoring by Copying. Restored files created using the **Copy** command or menu selection can end up in clear text, as explained previously in the section, **Copying an Encrypted Folder or File**.

Restoring using Backup in Windows 2000 or any backup utility that supports Windows 2000 features. This is the recommended way to restore encrypted files. The restore operation maintains the file encryption, and the restoring agent doesn't need access to private keys to restore the files. After the restoration is complete, the user with the private key can use the file normally.

For more specific operations, see [Microsoft 00] which contains a wealth of information and examples for both users and administrators. In addition, [NSA 01g] contains similar concepts but from a different perspective than that of Microsoft's. Both are valuable resources.



Beyond ACLs – Encryption -3

LINUX

- CFS - Cryptographic File System
- ppdd - Practical Privacy Disc Driver
- ReiserFS
- BestCrypt

SOLARIS

- CFS - Cryptographic File System

Networked Systems Survivability
© 2002 Carnegie Mellon University
Module 9: Host System Hardening - slide 19

UNIX/LINUX-based operating systems also have support for cryptographic file systems. [Tzeck 99] contains a list of many of those available today.

Among the first to support UNIX systems and subsequently LINUX was the Cryptographic File System [Yuriev 96] originally written by Matt Blaze. CFS rules exclusively in user mode and does not require kernel patches. Unfortunately, this design also slows its performance. Nonetheless, it is worth the effort. CFS is also available for SOLARIS. CFS is free.

Encryption information is on a directory-by-directory basis and the encrypted versions of files reside on the computer system doing the decryption. Once of the benefits of CFS is that it can also operate on files residing on other servers where those files are stored using NFS, AFS, or other file sharing methods.

Another file system encryption package for LINUX systems is **pppd**, the Practical Privacy Disc Driver [Tzeck 99a]. It encrypts an entire file system and it does require kernel patches. It is free.

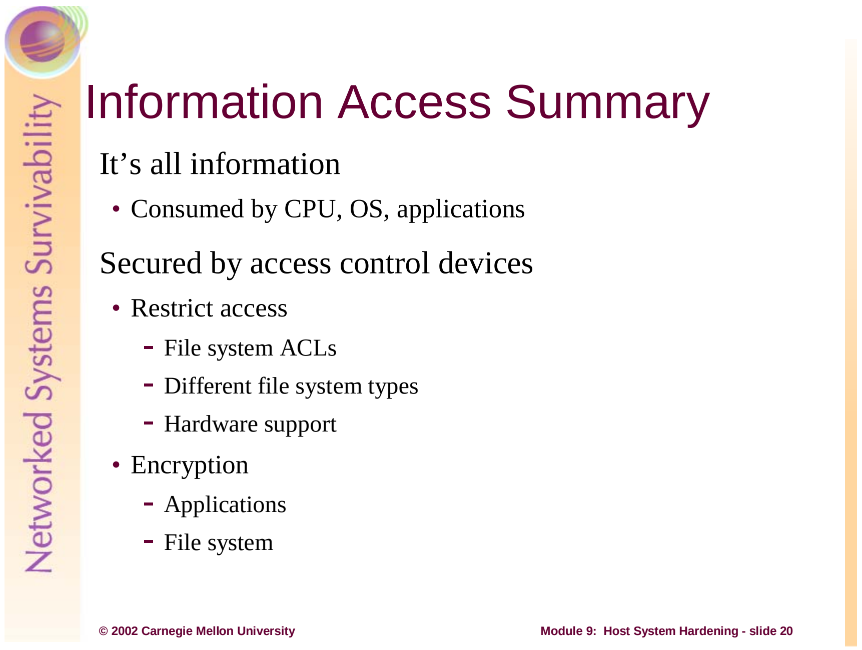
Currently under development is ReiserFS Version 4 [Reiser]. ReiserFS is a file system using a plug-in based object oriented variant on classical balanced tree algorithms. The results when compared to the ext2fs conventional block allocation based file system, running under the same operating system and employing the same buffering code, suggest that these algorithms are overall more efficient and are becoming yet more so.

Key to this file system is plug-ins, including those for security. These plug-ins will address the concerns of file auditing, aggregated files when different ACLs for the parts of a file, not just the file as a whole, and efficient encryption (where file contents are encrypted on flushes to disk, not when a buffer is written).

ReiserFS shows lots of promise and version 4 is due to be released on September 30, 2002. It is a free product, sponsored by DARPA.

Finally, for LINUX there are some commercial products. Among them are BestCrypt [Jetco 01]. Their 1.0b version works with the latest LINUX kernel versions and does not require kernel rebuilds to work.

Encrypting file systems are available for LINUX and UNIX-based systems. Some are free and some are commercial. Unfortunately none are as well integrated into the operating system as is EFS under Windows 2000 and its successors.



Information Access Summary

It’s all information

- Consumed by CPU, OS, applications

Secured by access control devices

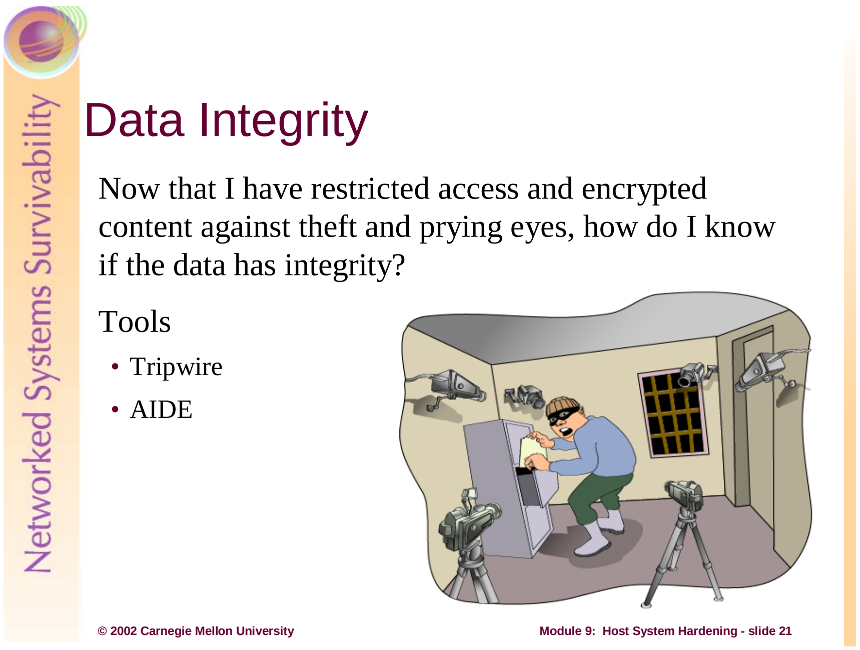
- Restrict access
  - File system ACLs
  - Different file system types
  - Hardware support
- Encryption
  - Applications
  - File system

Networked Systems Survivability
© 2002 Carnegie Mellon University
Module 9: Host System Hardening - slide 20

In summary, this combination of access controls and encryption are the major ways to secure the traditional information assets against intruder attack. Notice also that both schemes can be applied to a single asset. This is an example of *defense in depth*.

To simplify the increasingly complex task of securing information assets, think of all information as input that is consumed by a information processor (i.e., the operating system or a program) through a service (the file system, a Web server, email server, etc). That information can be protected in two fundamental ways:

- Restricting access – Only those who can operate on the information (i.e., read, write, append, execute, etc.) are allowed, and all other operations by all other users are excluded. On a computer system, this is achieved with the access control devices known as access lists. They come in a variety of types, from those traditionally associated with the file system, to special types of files and hardware controls. Access controls are also used at the network level, where network data is exposed to only those hosts and network segments as needed. All of these access control devices restrict access to the information assets they are guarding.
- Concealing information – With this type of access control device, other users may be able to see the concealed form of the asset but only a few can translate it into a usable form. On a computer system, this is achieved through encryption, be they assets that reside on storage media, transiently on the network, or in the address space of a running application. The information must be strongly encrypted such that the time to decrypt it require more time elapsed time than the useful lifetime of that asset. This means that by the time that the asset is decrypted, its unencrypted form is no longer useful.



Networked Systems Survivability

Data Integrity

Now that I have restricted access and encrypted content against theft and prying eyes, how do I know if the data has integrity?

Tools

- Tripwire
- AIDE

© 2002 Carnegie Mellon University

Module 9: Host System Hardening - slide 21

As that homeowner protecting assets in the locked file cabinet, how do you know if those assets have been changed, replaced, or in some way tampered with? That's a tough question. Some suggest that you take pictures all around your house to help jog your memory should there be a break-in.

To know where literally everything is or was around your house, you'd need to photograph every inch of everything you own, inside and out. That's a nearly impossible task. Moreover, every time you leave for a few days or even a few hours, you'd have to re-take every picture so that recent changes you've made would be caught on film. It might be fun to do once or twice, but the process would get old soon after.

Video surveillance is another way to record the goings on of the places where the camera is pointed. Assuming that everything works as designed and installed, and that the video tape does not run out, this video *log file* can help you understand what happened while you were away. Again, constantly video taping every room in your house is a daunting task, and it can be fairly expensive.

Back into the realm of information assets, how does the administrator determine if an asset has been changed or removed? Now that the administrator has tuned the file system so that the access to resources is appropriate, the next issue to deal with is the integrity of information assets.

There are at least two ways to discover if an information asset's integrity has been compromised. One way is to record all changes to the file in a transaction log. This log shows what change, when it changed and the identity of program and user who changed it.

Another method, and by far the most popular, is to record what a file used to be and then compare that previous state to the current state. Should they be the same, the file did not change. However, if they are different, the asset's integrity has been violated.

Clearly, keeping mirror images of every file in a system is inefficient and time-consuming. From Module 6 on Cryptography, we know that hashing a file's contents using the MD5 or SHA-1 algorithms creates a fix-sized descriptor of what a file's contents used to be. By hashing its current contents, we can compare the two hashes and conclude whether an asset has been altered.

This is the fundamental premise upon with the Tripwire integrity checking software suite is based. Here is a brief summary taken from [Tripwire]:

Data and network integrity depends on your ability to quickly determine if—and how—your data and network infrastructure has changed. Tripwire for Servers automatically monitors changes to files and system attributes, including file size, access flags, write time, and much more.

Identifies Changes

Tripwire for Servers determines how UNIX and Microsoft Windows NT file systems, and Windows NT registry keys have changed. It begins by creating a baseline database of files, directories, and the NT Registry. It monitors file content integrity and 24 system and registry attributes on Windows NT:

- File adds, deletes, modifications
- Flags-archive, read-only, hidden, offline, temporary, system, director
- Last access time
- Last write time
- Create time
- File size
- MS-DOS 8.3 name
- NTFS Compressed flag, NTFS Owner SID, NTFS Group SID, NTFS DACL, NTFS SACL
- Security descriptor control and size of security descriptor for this object
- Number of alternate data streams
- Hash checking- CRC-32, POSIX 1003.2 compliant 32-bit Cyclic Redundancy Check; MD5, the RSA Data Security, Inc.® Message Digest Algorithm; SHA, part of the SHS/SHA algorithm; HAVAL, a strong 128-bit signature algorithm

Tripwire for Servers monitors 14 system and registry attributes on UNIX systems:

- File adds, deletes, modifications
- File permissions and properties-ignore, record and check
- Inode number, number of links
- User id of owner, group id of owner
- File type, file size
- Device number of the disk on which the inode associated with the file is stored
- Device number of the device to which the inode points. Valid only for device objects.
- Number of blocks allocated
- Modification timestamp
- Inode creation/modification timestamp
- Growing files-indicates that the file is expected to grow.
- Shrinking files
- Access timestamp
- Hash checking- CRC-32, POSIX 1003.2 compliant 32-bit Cyclic Redundancy Check; MD5, the RSA Data Security, Inc.® Message Digest Algorithm; SHA, part of the SHS/SHA algorithm; HAVAL, a strong 128-bit signature algorithm

The Tripwire for Servers software engine conducts subsequent file checks, automatically comparing the state of the system with the baseline database. Any inconsistencies are reported to Tripwire Manager and to the host systems log file. Reports can also be emailed to an administrator.

If a violation is actually an authorized change (such as installing an upgrade or new application), a user can update the database so changes no longer show up as violations.

Flexible Policy Language

The power behind Tripwire technology lies in its highly configurable policy language. Not only can you define which files or directories to monitor, you can also define the attributes of each object monitored. For example, it makes sense to monitor data integrity of a system binary file. It doesn't make sense to monitor the contents of a log file, but you do want to ensure that certain aspects of a log file (such as permissions or ownerships) don't change. You can configure Tripwire for Servers to monitor only those things that shouldn't change.

Tripwire policy languages also allow you to group objects around easy-to-understand rule names and then prioritize them based on relative "severity." Grouping and prioritization enables you to focus on the most important changes first-performing "triage" in a time- critical situation. Tripwire for Servers includes a pre-defined policy file for the platform on which it will run, enabling you to quickly install and deploy your Tripwire solution.

Secure Management

Tripwire for Servers software communicates with the Tripwire Manager management console via Secure Sockets Layer (SSL) protocol. Commands and data exchanged with the console are run as a daemon or service. Tripwire for Servers and Tripwire Manager allows you to manage data and network integrity from one central location.

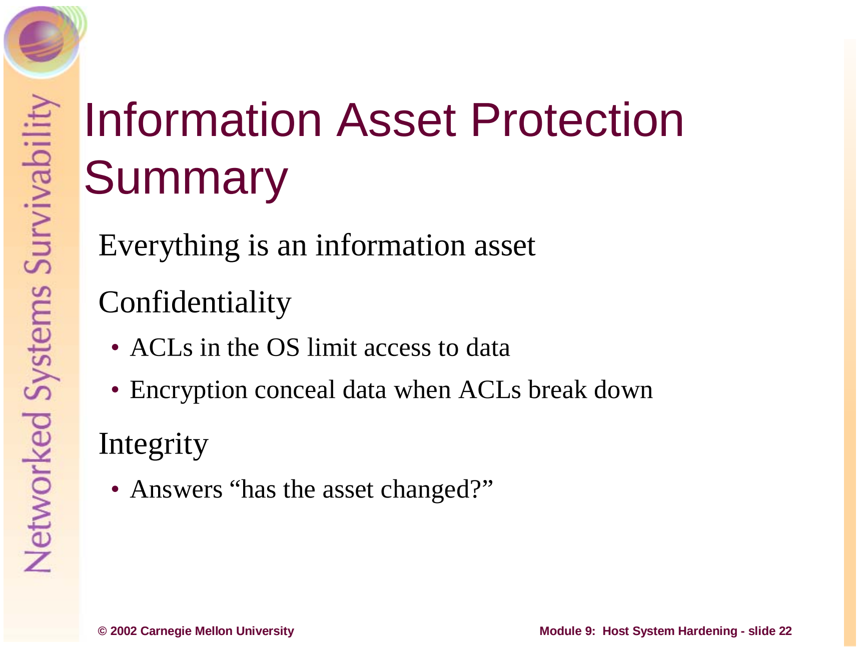
Tripwire is a commercial product for Windows-based platforms and most UNIX platforms. There is a free version for LINUX[2]. In addition there are Tripwire tools for Cisco Routers and Switches[3] and for Apache-based Web servers[4].

AIDE (Advanced Intrusion Detection Environment) is a free replacement for Tripwire. It does the same things as the semi-free Tripwire and more. There are other free replacements available so why build a new one? All the other replacements do not achieve the level of Tripwire. And I wanted a program that would exceed the limitations of Tripwire.

It creates a database from the regular expression rules that it finds from the config file. Once this database is initialized it can be used to verify the integrity of the files. It has several message digest algorithms (md5,sha1,rmd160,tiger,haval,etc.) that are used to check the integrity of the file. More algorithms can be added with relative ease. All of the usual file attributes can also be checked for inconsistencies. It can read databases from older or newer versions. See the manual pages within the distribution for further info.

AIDE is available for free download at: http://www.cs.tut.fi/~rammer/aide.html

[2] http://sourceforge.net/projects/tripwire
[3] http://www.tripwire.com/products/routers
[4] http://www.tripwire.com/products/web_pages



Information Asset Protection Summary

Everything is an information asset

Confidentiality

- ACLs in the OS limit access to data
- Encryption conceal data when ACLs break down

Integrity

- Answers “has the asset changed?”

© 2002 Carnegie Mellon University

Module 9: Host System Hardening - slide 22

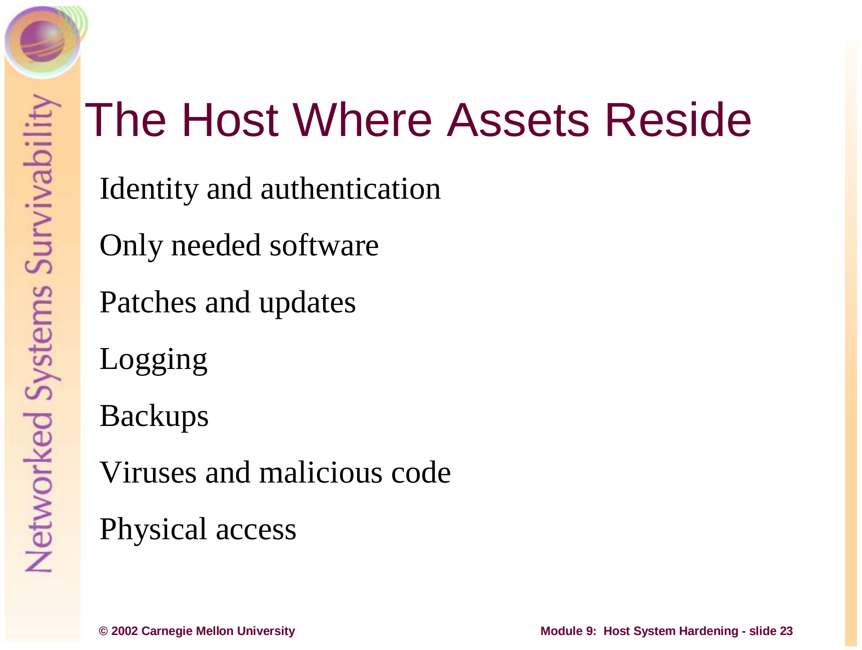
In your home, everything is property and has value to someone. In many cases, you have chosen to minimize their value and so you leave that property only minimally protected. In other cases, you are very concerned, even to the point of putting that property into a locked cabinet, and perhaps encrypting it to safeguard against theft or inadvertent disclosure.

On today’s computer systems, everything is also property – an information asset – be it the traditional application-specific files stored in the file system, the application executable files themselves, or even the operating system. Given this information asset-centric perspective, the first way to secure that asset against those who should not have access – that is *confidentiality* – is through the access control device known as access control lists or ACLs. ACLs are an operating system-provided capability that limits the access granted to identities as assigned by that operating system at login time.

The second way to limit access is through the access control device known as encryption. Through encryption, an information asset is concealed from its viewer through techniques steeped in advanced mathematics and computer algorithms. They are designed to address a breakdown in access where someone not authorized to view an asset is indeed able view that asset. Through encryption, that asset is useless because the cryptographic algorithms hide content by turning it into unintelligible gibberish. To those with access to the encrypting/decrypting algorithms and the necessary keys, the encrypted form can be decrypted for their use. A strong encryption algorithm must render all brute force attacks ineffective for the useful lifetime of the asset they conceal. This means that as computing resources (CPU speed, algorithms, and magnitude of amassable resources) grow, algorithms once thought to be strong may become weak and the information they once guarded become vulnerable to disclosure. See Module 6 on Cyptography for a more detailed explanation.

Finally, all information assets have an attribute of integrity. Through still more advanced mathematics and computer algorithms, alterations even as small as a single bit can be detected in the largest of files. Administrators and other users can have confidence in the information assets they use by first establishing their integrity through appropriate software tools.

So in summary, everything is an information asset. Their access can be restricted; their content can be concealed; their integrity can be verified.



The Host Where Assets Reside

Identity and authentication

Only needed software

Patches and updates

Logging

Backups

Viruses and malicious code

Physical access

Networked Systems Survivability

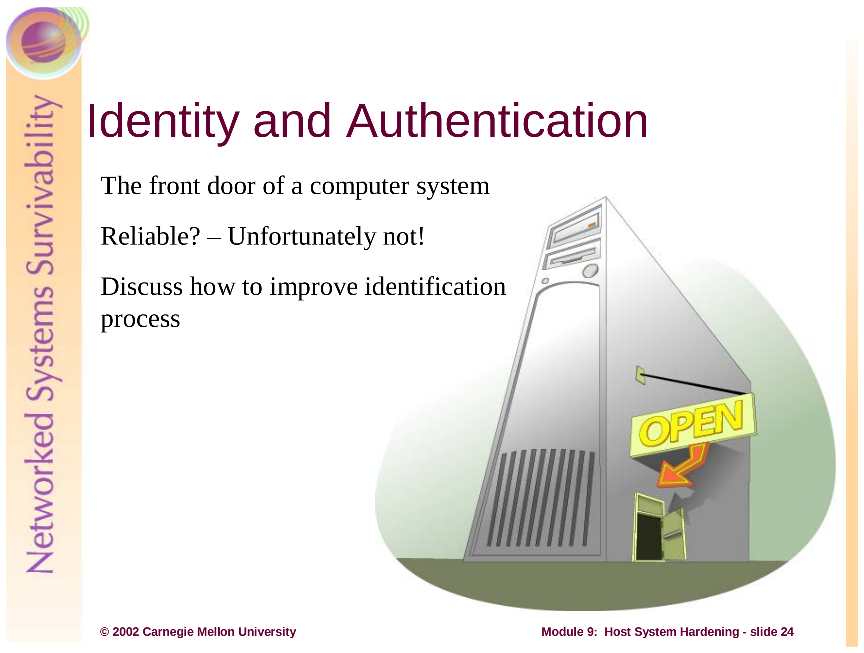
© 2002 Carnegie Mellon University Module 9: Host System Hardening - slide 23

In your home, you've got rooms and the rooms have doors and the house also has a door. In many cases, these doors are locked to guard against entry by those who should not have access. You have secured the house as a way of protecting your property.

Similarly on a computer system, information assets are likely stored on one or more computer systems, and their security is of paramount importance. The previous topics discussed discriminating access to them based upon access control lists and encryption. Further, their accuracy can be ascertained through integrity checking tools.

The next point of focus when hardening a host computer system is the host where those information assets reside. Here are the topics addressed in this section:

- Identity and Authentication: This topic discusses reusable passwords, one-time passwords, and other forms of authenticating users to a computer system.
- Install only the needed software: Install only that software needed by a computer system to do the job it's been selected to do.
- Patches and Updates: Keep operating systems and applications software up to date.
- Logging: Identify and enable system logging mechanisms.
- Backups: Configure systems for file backups.
- Virus and malicious code: Protect systems from viruses and similar programmed threats.
- Physical access: Protect computers from a physical perspective.



Identity and Authentication

The front door of a computer system

Reliable? – Unfortunately not!

Discuss how to improve identification process

Networked Systems Survivability
OPEN

Module 9: Host System Hardening - slide 24

Most access to your house is through the front door, and that door is usually locked. Family members have their own keys and they can come and go as they please. Others allowed in must first past inspection at the front door where they are either recognized or their identity verified in some other way (ID card, official papers, etc.). There is usually only one front door and it is usually clearly identified for all who seek entry.

Once someone has gained access to the house, they usually have access to everything in the house, though in our analogy, a file cabinet and encryption further guard some very sensitive information. This defense in depth approach grants layers of access to those who have any access at all. There are outsiders, insiders, and trusted insiders.

Access to the house is usually regulated by policy as well. Those with keys are told to guard those keys and not loan them to others. Procedures also constrain access to keys. For example, a key is not prominently displayed for all to use in or around the front door, though looking under the front door mat may produce a key available for any who finds it.

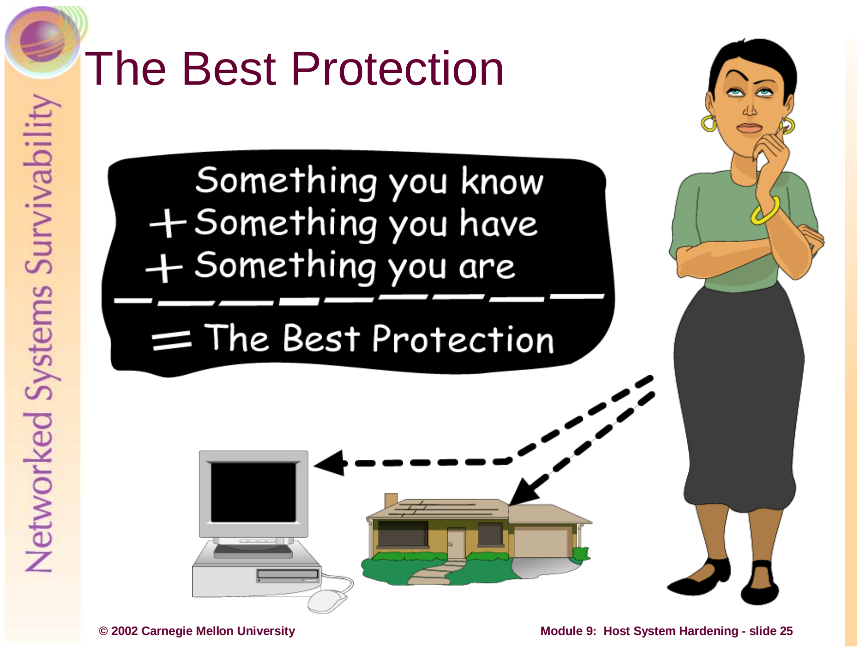
Access to a computer system is similar to access to a house through the front door. That access is usually through a login procedure that traditionally prompts you for a name and some identifying information, typically a password. This is the "door" to the system that most users must go through. Those with a valid login/password pair (keys to the house) are granted access. Note that just as a burglar can pick a lock, so too can an intruder by obtaining a login/password pair by any of the well-known means (sniffing, shoulder surfing, keystroke monitoring, or brute force cracking).

There are policies and procedures governing logins and passwords as well. For example, users are often strongly cautioned against sharing their login and password pair. Posting your passwords on the infamous yellow sticky on your computer's monitor is also strongly discouraged by policy.

In general, access to a computer system through the front door is based upon the reliable identification and authentication of people to that computer system. Once identified and authenticated, those people are considered *users* and are granted certain privileges and rights to carry out tasks on that system. Users carry with them a set of credentials that identify who they are and that define the set of rights that have been granted to them. These credentials are carried as part of every program that runs on a computer system. They are presented to the operating system as a way to access information assets and services provided by the operating system.

If the process by which a person identifies himself or herself to a computer system was truly reliable, this discussion would be at an end. Unfortunately, this is not the case as the traditional methods used by most modern day computer systems by which people identify themselves is eminently unreliable. This discussion centers on the problem of identification and authentication of people to computer systems.

The discussion on access control devices, specifically ACLs, is based upon that reliable identification of a person to a computer system. If the identification scheme can be breached, then ACLs cannot properly guard the information assets in question. Therefore, it is fundamentally important to accurately identify a person to a computer system so that the credentials they acquire through that identification process reflect the correct user.



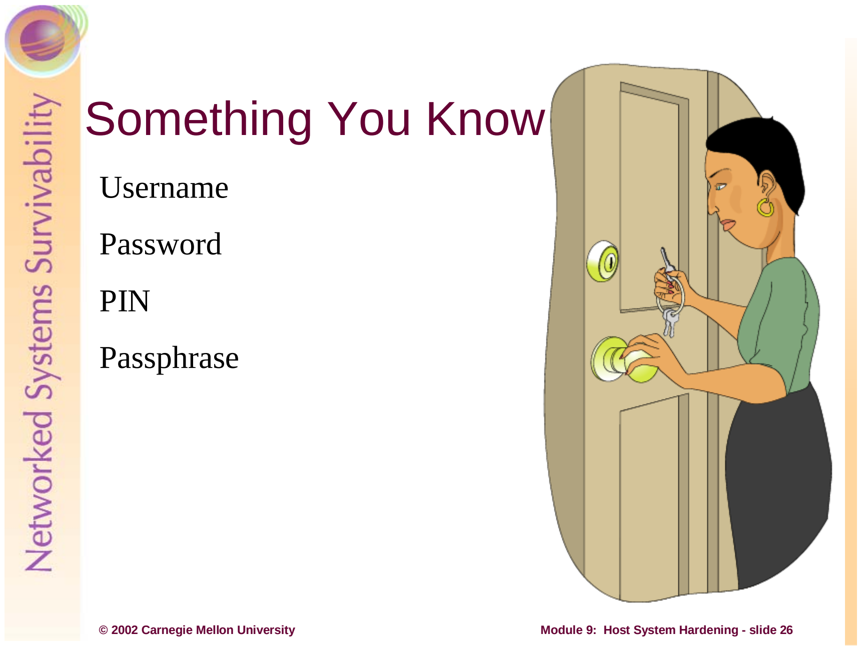
The Best Protection

Something you know
+ Something you have
+ Something you are

= The Best Protection

Networked Systems Survivability
© 2002 Carnegie Mellon University
Module 9: Host System Hardening - slide 25

In the best possible protection scheme, identifying a person to either a house or to a computer system combines something you know, something you have, and something you are. However, forcing people to use all three at each login may be cumbersome to the point of rendering a computer system unusable. In lieu of using all three, many advanced identification schemes use what is called *two-factor authentication*, where only two of the three are required to successfully identify a person. The next section discusses examples of these three attributes as well as products that combine two of them to produce two-factor authentication.



Networked Systems Survivability

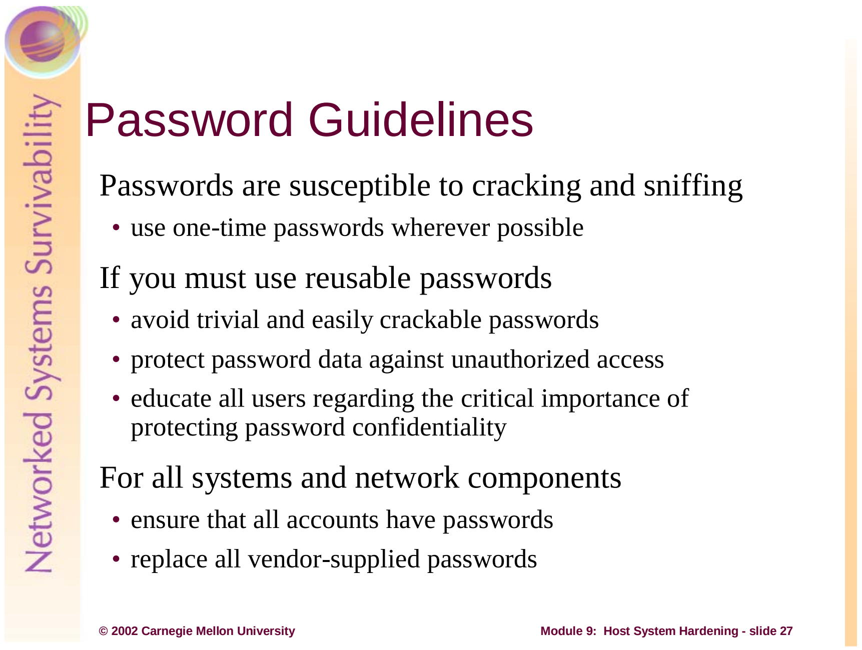
Something You Know

- Username
- Password
- PIN
- Passphrase

Most access to a house does not use much that you know, with the possible exception of which door a key opens. In almost all cases, the key in question opens the one and only front door.

A user name is the login part of the login/password scheme that is the backbone of most computer system identification subsystems. It is often unchanging and is therefore a likely candidate for capture and reuse. Passwords are also something that you know and are traditionally six to ten characters long. PINs are usually shorter, and passphrases are usually longer.

Most computer systems limit their requirement for identifying information to something you know, and that is traditionally a username/password pair.


Networked Systems Survivability

Password Guidelines

Passwords are susceptible to cracking and sniffing

- use one-time passwords wherever possible

If you must use reusable passwords

- avoid trivial and easily crackable passwords
- protect password data against unauthorized access
- educate all users regarding the critical importance of protecting password confidentiality

For all systems and network components

- ensure that all accounts have passwords
- replace all vendor-supplied passwords

© 2002 Carnegie Mellon University Module 9: Host System Hardening - slide 27

The bottom line is that the something you know – passwords – are susceptible to discovery through brute force cracking and through sniffing, where either network packets the keystrokes entered at a computer system are captured. Note that no matter what policy and rules used by an organization, if reusable passwords can be sniffed, an intruder can use them to gain access to an account. If at all possible, stay away from reusable passwords.

If that is not an option, then the best practice is to establish policy backed up by technology that enforces the use of strong passwords. On most LINUX/UNIX systems, the standard password cracking support library – *libcrack* – is installed and integrated into the password-changing scheme. This means that users cannot select passwords that would be discovered through standard brute force means.

To require strong passwords in Windows 2000, you'll need to open the Group Policy snap-in within a Microsoft Management Console. Begin by opening a `Microsoft Management Console`: from the `Start` menu and then choose Run. Type `MMC` and click `OK`. Next, choose `Console` → `Add/Remove Snap-in`. Click `Add`. From the list of Snap-ins, choose `Group Policy` and accept the default settings.

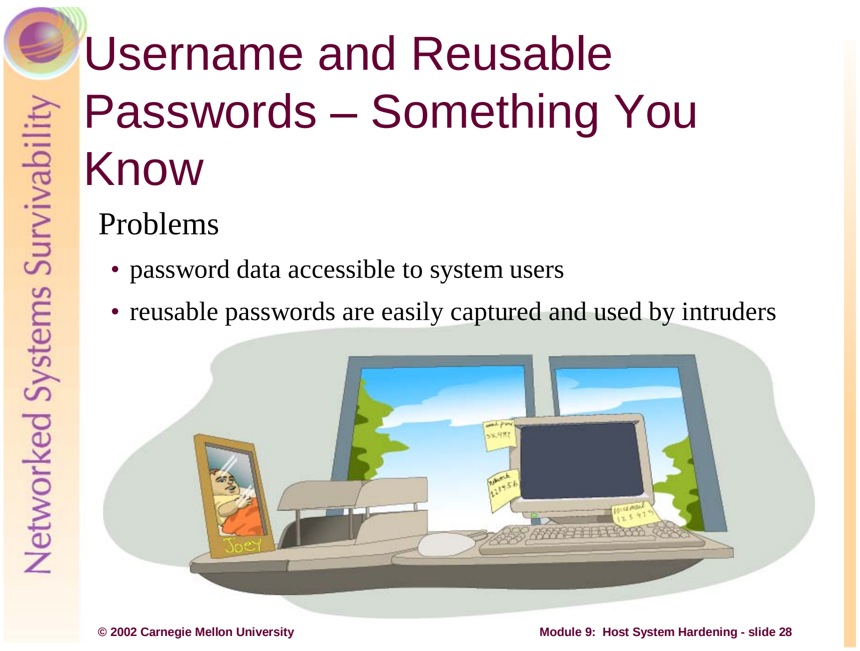
In the console tree, expand `Local Computer Policy` → `Computer Configuration` → `Windows Settings` → `Security Settings` → `Account Policies`. Select the `Password Policy` object. In the details pane, enable `the Passwords Must Meet Complexity Requirements`. By default, Windows 2000 disables this setting. To enable it, double-click on the setting and choose `Enabled`. The system will then require people to use strong passwords when they login [ZDNET 00].

No matter which operating system you use, make sure that the password data has the highest level of protections available. This means setting all related ACLs appropriately and concealing password data where possible.

Finally, people need to know that intruders can easily attack reusable password schemes by sniffing networks and keystrokes. They need to be aware of the communication path over which passwords travel and the security of the hosts being used. Often the best thing to do may be to not login to a system because of the information exposed along that pathway.

Administrators should be certain that all accounts have passwords and that any vendor-provided accounts get new passwords immediately after activation.

In summary, reusable passwords are just that – reusable. That reuse comes traditionally from the person logging in but it continues to come from intruders who discover passwords through brute force cracking and sniffing techniques. If reusable passwords are avoidable, then avoid them wherever possible. They continue to be a frequently used method of attack.



Networked Systems Survivability

Username and Reusable Passwords – Something You Know

Problems

- password data accessible to system users
- reusable passwords are easily captured and used by intruders

© 2002 Carnegie Mellon University

Module 9: Host System Hardening - slide 28

The key to the front door of your house is similar to a reusable password. Reusable means that the same method is used each time access is desired. If that reusable key can be captured and subsequently copied, then somebody else can use it to gain entry. All it takes is an image of that key and the crafty burglar can get it whenever they want.

On most computer systems today, the user identification scheme relies on reusable passwords. This scheme suffers from two major flaws. They are:

- The information asset associated with the password is available to more users than is necessary. This means that some form of the password, normally an encrypted version, is stored somewhere on the system where the person logs in. Examples are in files and in the address space of a process validating identity. If those locations can be somehow breached, then the encrypted version can be viewed and used in an attack.

  Recall the previous discussion on securing information assets through access control devices. Given that this password data is information, the same techniques for securing it also apply. The strongest ACLs should be used, and encryption beyond that. From the Rubik’s Cube model, assets must to be secured where possible in all of the places where it resides.

  On LINUX/UNIX-based systems, the encrypted form of passwords resides in either the world-readable */etc/passwd* file or the root-only readable */etc/shadow* file. In the former case, any user can read the file whereas in the latter, the root privileges must be gained first. In either case, once the encrypted form is read, brute force techniques can be used to discover the clear text form of passwords. While most modern systems support shadow passwords, they may need to be enabled on your system. Consult your system documentation to learn how to enable shadow passwords. For example, on a RedHat Linux 7.2 system, shadow passwords are enabled on through the system installation tool.

  On Windows-based systems, a form of passwords appears in the address space of the **lsass** (the Local Security Authority Subsystem) process. If the SAM (Security Access Manager) database in the registry is obscured, then the form that appears in the **lsass** is a non-obscured version. If the **lsass**’s address space can be compromised, then the version found there can be used to discover the clear text form of passwords.

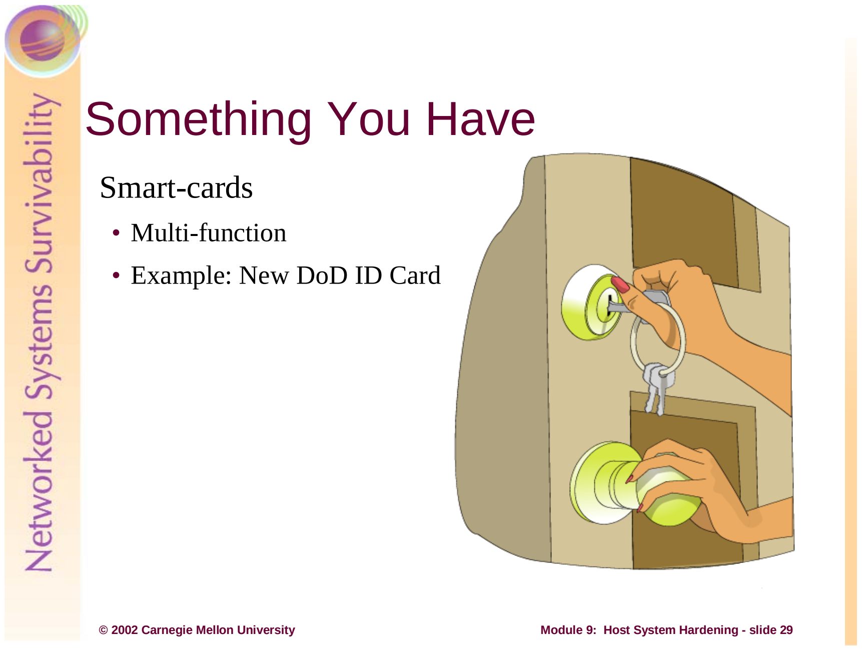
  As you secure these user identification information assets against intruders, make sure that the strongest possible ACLs are in place, and beyond that, the strongest possible encryption that your system provides. Consider all of the places that identification information resides. In the case of Windows-based systems, obscured forms of passwords appear in the virtual address space of a process. On LINUX/UNIX-based systems, the same is also true, where the password may actually appear in the clear in processes such as **login** and any other program that prompts for a password. If access to the virtual memory of those processes or to the physical memory of the host computer can be compromised and the data read, then passwords can be captured. The key is to recognize that this identification data is information and must be protected as such.

- Reusable passwords are easily captured and used by intruders. In many cases, applications such as **telnet** and email clients transmit their login and password information in the clear across a computer network. This means that they can be captured through sniffing and replayed, giving unwanted access to others. One example of such a sniffer is **dsniff**[5] that listens on the network for keywords such as login and password and displays the contents of the reassembled packets that directly follow them.

  Through techniques called *key logging*, keystrokes entered at the keyboard, all mouse clicks, etc. can be captured. One such product is called Stealth Keyboard Interceptor Pro[6]. It silently monitors your PC, intercepting keystrokes and saving them to a text-based log file. The log is complete, retrieving time/dates, application and dialog names, file names, pasted text, and keystroke actions.

  The result is a clear picture of all activity that has transpired on a PC. Users won't see the program in the task list, task bar, or system-tray area. For added security, the administrator can choose to encrypt the log file. Several settings are available to customize the behavior of the program: Mouse clicks, control keys, keyup events, and keydown events can be recorded.

[5] http://www.monkey.org/~dugsong/dsniff/
[6] http://www.keyloggers.com/skinpro.html

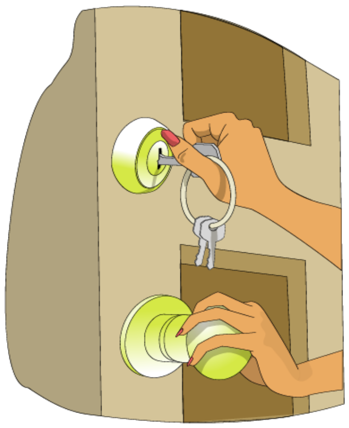


Networked Systems Survivability

Something You Have

Smart-cards

- Multi-function
- Example: New DoD ID Card



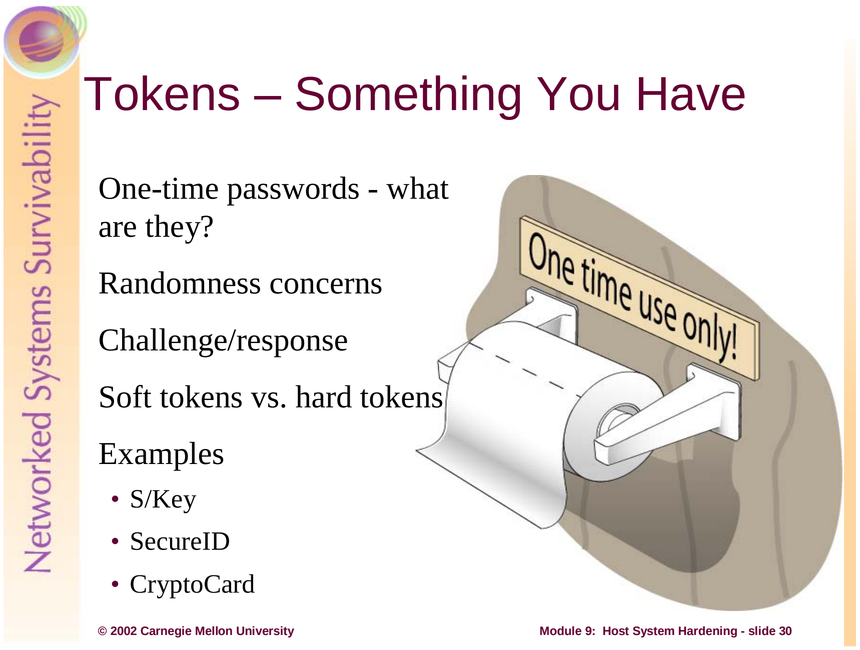
© 2002 Carnegie Mellon University

Module 9: Host System Hardening - slide 29

At home, the something you have is the house key and all who have it have access to the house independent of any other means of identification.

Some computer systems use smart-cards. A smart-card is a device that is about credit card sized and may have a processor of its own. There are many types and the use of this technology is increasing. To learn more about smart-cards, here’s a web site that has lots of background and product information: http://www.scia.org/.

Many smart-cards support multiple functions. Some cards can provide authenticated access into the physical security of buildings and rooms as well as computer systems. Others have been integrated into time-card systems for keeping track of hours worked, etc.



Networked Systems Survivability

Tokens – Something You Have

- One-time passwords - what are they?
- Randomness concerns
- Challenge/response
- Soft tokens vs. hard tokens
- Examples
  - S/Key
  - SecureID
  - CryptoCard

One time use only!

© 2002 Carnegie Mellon University

Module 9: Host System Hardening - slide 30

A person wishing to gain access to information assets must first identify himself or herself to the computer system where those assets reside. Usually this requires that the person log into that computer system with a password. If that system supports one-time passwords, then the password part of the login/password pair works only once and is then no longer valid. One-time password methods defeat sniffing because the captured password doesn't work a second time. This means that there is little or no concern about the communication channel over which they travel.

Key to one-time passwords is true randomness. Imagine that the one-time scheme was weak, meaning that the next one-time password in the sequence could be computed or guessed. For example, if the first password in a one-time password sequence was `AAAAA` and the second was `bbbbb`, it seems reasonable to conclude that the next password in this sequence could be `CCCCC`. While these are indeed one-time passwords – they are only good one time – they are predictable and therefore weak. So, one-timeness is also a characteristic of one-time passwords.

One-time passwords frequently use a challenge/response scheme where the computer system being logged in to issues a challenge to which the prospective user must respond. The means to manufacture that response often comes from some type of hand-held device – the something you have. It uses the same algorithms as the host computer system. When the host challenges the person, they use this device to compute the response, and the host does the same. The person then provides the response and if they match, the person is properly identified as the user in question and then authenticated to the system. Most of the hand-held devices also require a login so that simply having the device is not sufficient to compromise the integrity of the login process. It is this combination of something you know (the login to the hand-held) and the something you have (the hand-held itself) that improves the quality of the login scheme.

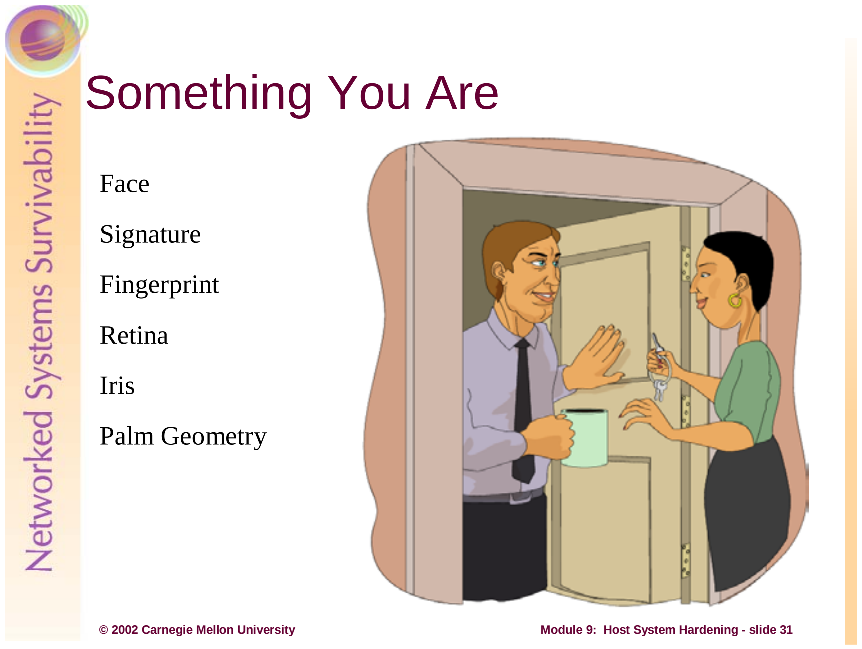
One-time password implementations come in two general flavors – soft tokens and hard tokens. Soft tokens – tokens computed through software – are cheaper to operate. Some implementations even use Java as the programming language. Since they traditionally operate on a computer system, they can be compromised by the traditional techniques used to compromise other software systems.

To improve the level of security of one-time passwords, use a hard token-based scheme. There are many products in the marketplace that do one-time password schemes. Among them are:

- S/Key: http://www.ece.nwu.edu/CSEL/skey/skey_eecs.html

- SecureID: http://www.rsasecurity.com/products/securid
- CryptoCard: http://www.cryptocard.com

One-time password methods use software and hardware to improve security by more strongly identifying the person attempting to login to a computer system. Through a combination of what you have and what you know, those allowed access to information assets are more clearly separated from those not allowed such access. The technologies available today are cost effective, user-friendly, and more easily integrated into most modern operating systems.


Networked Systems Survivability

Something You Are

- Face
- Signature
- Fingerprint
- Retina
- Iris
- Palm Geometry

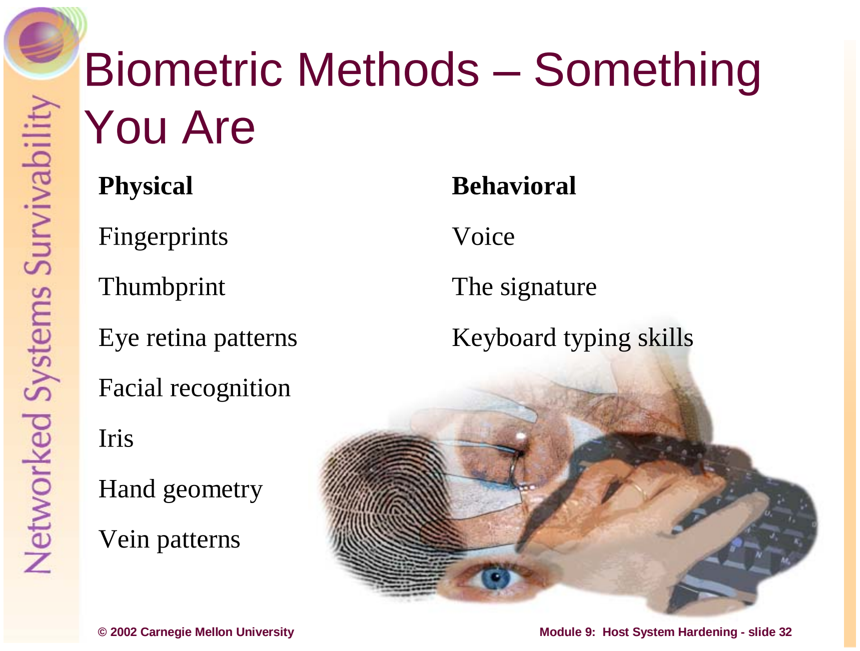
© 2002 Carnegie Mellon University

Module 9: Host System Hardening - slide 31

If you don't have a key to the house, one means of gaining entry is by being recognized by the person who opens the door. Your face conveys who you are and it is presumed to be difficult to copy and therefore a reliable means of identification.

Signatures are an interesting means of identification. Have you ever purchased something from a store and used a credit card? What did the clerk do with the card? Did they even bother to check your signature to see if they matched or did they just hand the card back to you and not even bother to compare the handwriting?

On a computer system, the something you are uses people's physical attributes as another means for identifying them to a computer system. We'll talk more about these in light of biometrics.



Networked Systems Survivability

Biometric Methods – Something You Are

Physical	**Behavioral**
Fingerprints	Voice
Thumbprint	The signature
Eye retina patterns	Keyboard typing skills
Facial recognition	
Iris	
Hand geometry	
Vein patterns	

© 2002 Carnegie Mellon University   Module 9: Host System Hardening - slide 32

Biometrics identification methods represent an opportunity to use the unique features of a person as the way to identify and subsequently authenticate people to computer systems. The fundamental premise of biometric methods is that we are all unique somehow and these unique features can be used as a reliable means of identification.

Biometrics information can be classified in two ways: physical characteristics of a person and behavioral or patterns-of-action characteristics. Physical attributes consist of fingerprints, thumbprints, retinal patterns, facial recognition, the iris, hand, and patterns of veins. These attributes are presumed to be difficult if not impossible to counterfeit, though in the James Bond movie *Never Say Never Again*, even a retinal scan technology was defeated.

Behavioral characteristics attempt to uniquely identify a person based on how they act, again presuming their actions to be unique or difficult to reproduce. Among the attributes sampled are voice, signature, and keyboard typing skills.

At home, we use biometrics to identify the person at the door. Perhaps we recognize them, so we are using their face, their voice, or other physical attributes before granting access. If we don't recognize them, we probably do subconsciously categorize them by their physical appearance before taking a chance to let them in.

On computer systems, there are products that use biometrics coming to market. Some are affordable and therefore practical alternatives for identifying people to computer systems. Examples are:

- Digital Persona's U.are.U[7] fingerprinting recognition product. This comes in both personal and professional versions and is integrated into Windows XP's fast user switching feature.
- Sony's FIU-700[8] is a credit card-sized device that verifies the user's fingerprints to allow access to networks, computers or individual applications. The add-on device can be connected to a computer via the USB port and works with either Windows 98 or Windows 2000.

In fact, support for biometric devices is now part of Windows XP. Here's an excerpt from a news story on this topic [CNET 00]:

[7] http://www.digitalpersona.com
[8] http://www.sel.sony.com/SEL/corpcomm/news/bandp/654.html

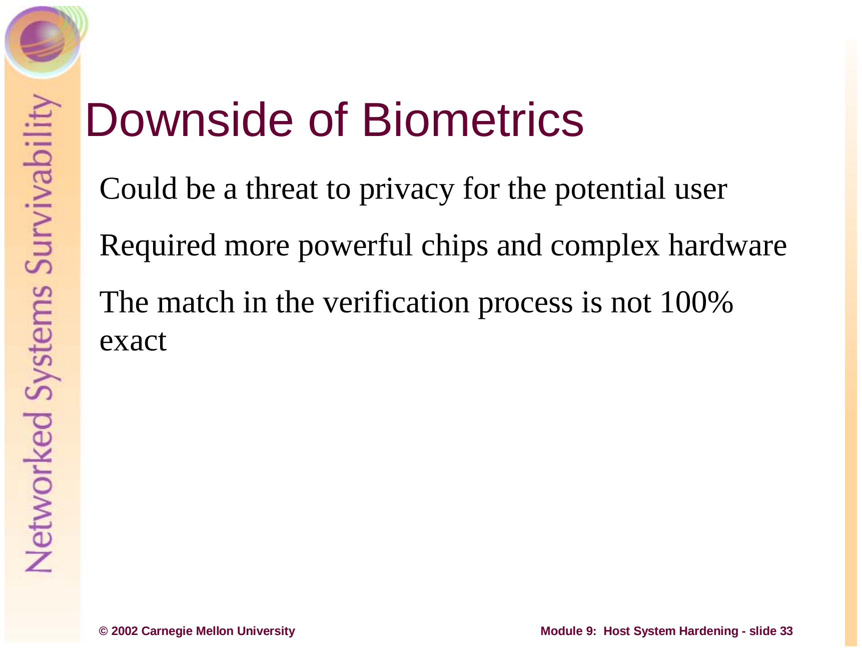
Microsoft is swapping passwords for fingerprint readers and retinal scanners.

The software giant today said it has entered a pact with security software firm I/O Software[9] to integrate the company's Biometric API (application programming interface) directly into Windows operating systems. Windows users could opt to have their identities verified through fingertips or other physical characteristics, rather than traditional passwords.

Riverside, Calif.-based I/O Software offers security products that verify identities--and correspondingly limit or allow network access--through fingerprints, irises, retinas, voices and faces. The technology is available with Windows 95, Windows 98 and Windows NT; Windows 2000 support is expected this summer, the companies said.

Biometric technology matches stored 2D and 3D images of an individual's body parts with the person. These systems often can compensate for aging, weight gain and other variations. The sheer volume of topographic data captured on a typical stored image also serves to deter fraud: Wearing a Bill Clinton mask won't likely get someone past a biometric firewall.

[9] http://www.iosoftware.com



Downside of Biometrics

Could be a threat to privacy for the potential user

Required more powerful chips and complex hardware

The match in the verification process is not 100% exact

Networked Systems Survivability
© 2002 Carnegie Mellon University
Module 9: Host System Hardening - slide 33

No identification and authentication scheme is foolproof and biometrics is no exception. In addition to the computational requirements needed to boost accuracy, the information captured challenges privacy issues and personal rights. For example, from a fingerprint alone, you may be able to determine a person’s daily activities (calluses obscuring the prints may mean a laborer, smooth prints may mean an office worker), their health, their sex, or their race. The key is to simply be aware that biometric information may convey much more than the identification information that the participants intended. Here is an excerpt from [RAND 01] on this issue:

Information Privacy

By far the greatest concern was about an individual's ability to control personal information. Not surprisingly, misuse of the information ranked high on the list of information privacy issues. Widely publicized accounts of identity theft and financial loss stemming from it have made people nervous about any information that could enable someone to assume their identity. Biometric data, though more complex than most passwords, can also be stolen or copied, so any attempt to develop a system using it must take these concerns into account.

Misuse is not the only information privacy concern. Some worry about "function creep" – that is, using information collected for one purpose to do something else. These secondary purposes may be perfectly valid, but if people have not been informed about the new use and have not consented to having information about them applied to it, they could object. Another worry is that biometric information will enable a third party to track an individual's actions or search databases to get information about them. Using a biometric to participate in routine daily activities (e.g., entering or exiting buildings, making financial transactions) leaves a detailed record. This capability bothers many people, who worry that some biometrics may allow their activities to be monitored without their knowledge or consent.

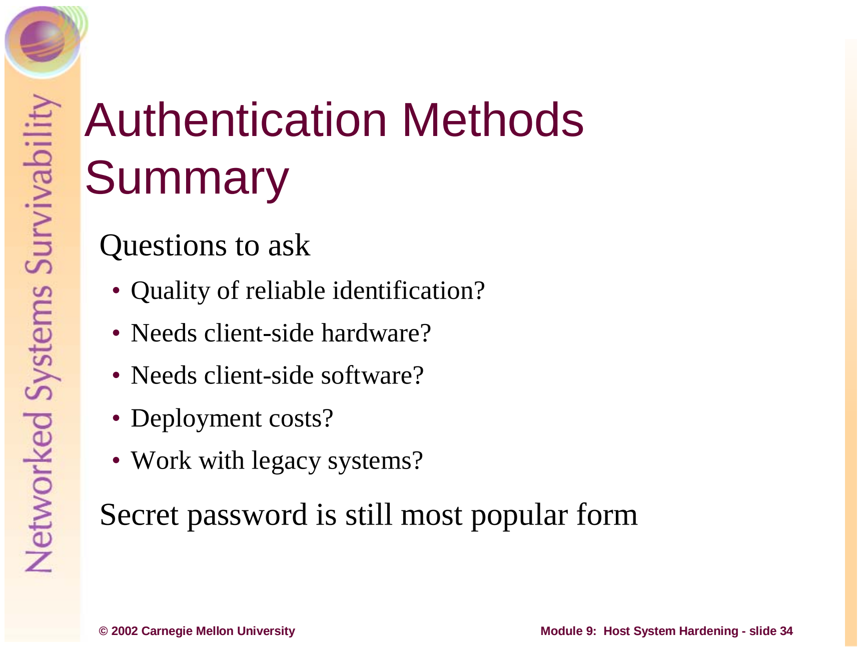
Physical Privacy

Unlike some other identifiers--a Social Security number, for example--biometrics raise issues of physical privacy. One is that certain biometrics carry a stigma. Fingerprints, for example, are widely associated with criminal activity. Another issue is concern about actual physical harm. While the research team found no biometric that causes such harm, it nonetheless troubles some. For example, retinal scanning techniques require individuals to place their eyes close to a camera lens, a requirement that makes some people uneasy. Hygiene issues also concern some. The need to place a hand or finger on a sensor plate can prompt fear about the spread of disease.

Religious Objections

Certain Christian sects have objected to the use of biometrics based on the "Mark of the Beast" language in the Book of Revelation. Although the number of such believers is relatively small, some members of the Army community might hold similar beliefs. The Army needs to be prepared to address such objections.

To learn more about biometrics, including the issues related to errors rates, see [Computer 01].



Networked Systems Survivability

Authentication Methods Summary

Questions to ask

- Quality of reliable identification?
- Needs client-side hardware?
- Needs client-side software?
- Deployment costs?
- Work with legacy systems?

Secret password is still most popular form

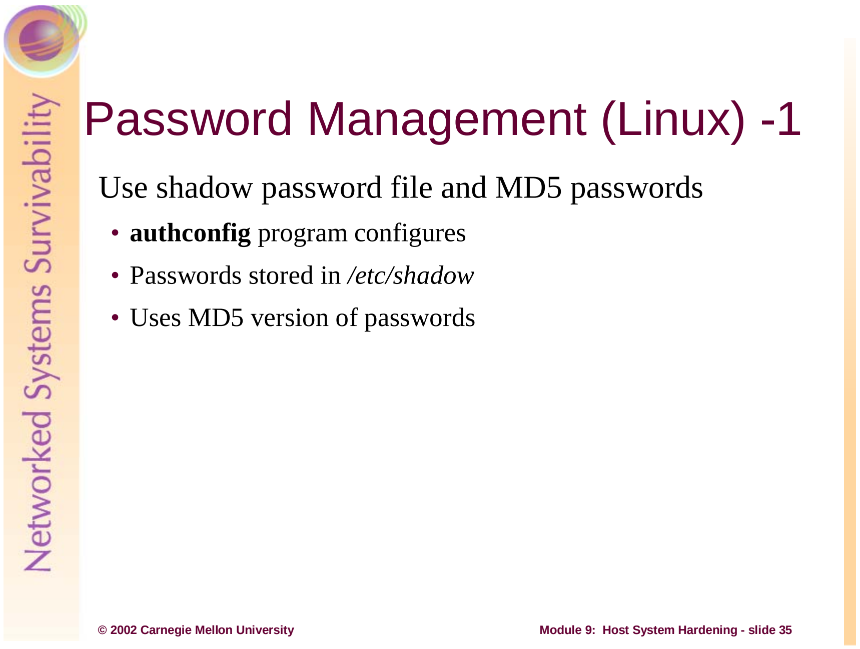
© 2002 Carnegie Mellon University

Module 9: Host System Hardening - slide 34

In general, it is more secure to use multiple forms of authentication, that is, a security token combined with a password. These types of authentication may be further characterized as follows:

Characteristic	**Something You Know - Secrets**	**Something You Have - Tokens**	**Something You Are - Biometrics**
Reliable identifcation/authentication?	Good	Very good	Excellent
Requires client-side hardware?	No	Sometimes	Yes
Requires client-side software	No	Sometimes	Yes
Typical deployment cost/user	0	$50	$100
Works with legacy systems	Yes	Sometimes	Sometimes

Due to cost and compatibility with legacy systems, the most popular form of user authentication continues to be a secret password. The next section details how to use the authentication methods of LINUX/UNIX and Windows 2000.



Password Management (Linux) -1

Use shadow password file and MD5 passwords

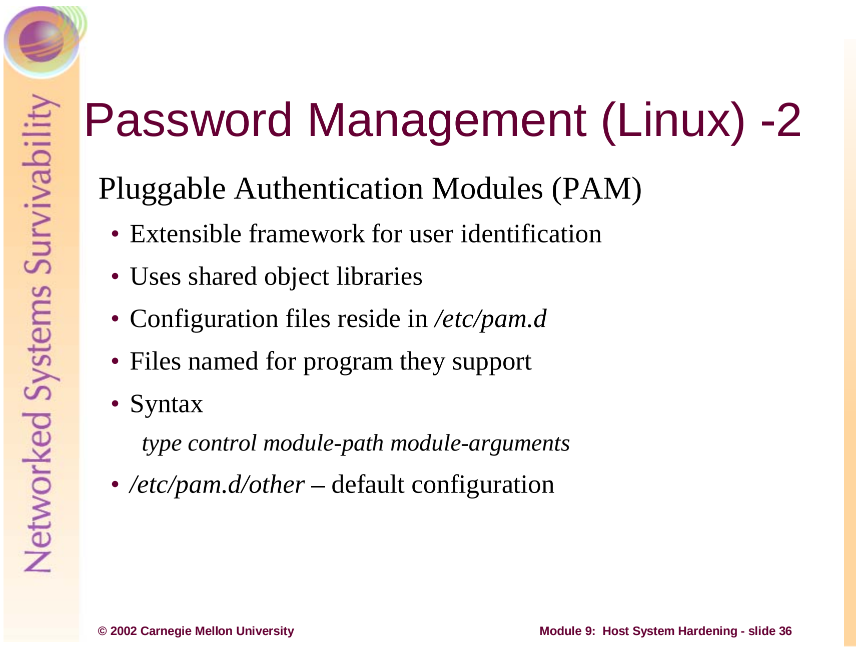
- **authconfig** program configures
- Passwords stored in */etc/shadow*
- Uses MD5 version of passwords

Networked Systems Survivability
© 2002 Carnegie Mellon University
Module 9: Host System Hardening - slide 35

On a LINUX system and most modern UNIX variants, the encrypted password information is stored in the */etc/shadow* file. This technique is used so that the information that most users need – the mapping between login names and the operating system's internal user identifier for example – is accessible through the */etc/passwd* file without compromising the encrypted passwords.

In addition, an MD5 hash of the password can be configured and it replaces the encrypted version of the password using the standard **crypt**(3) subroutine. The **crypt** subroutine is the password encryption function that is based on the Data Encryption Standard algorithm with variations intended (among other things) to discourage use of hardware implementations of a key search. This scheme makes discovering passwords through brute-force methods very unlikely to be successful.

Both the use of the shadow password file and MD5 hashes of passwords are configured by the **authconfig** program on a LINUX system. See it's manual page for a complete description.



Password Management (Linux) -2

Pluggable Authentication Modules (PAM)

- Extensible framework for user identification
- Uses shared object libraries
- Configuration files reside in */etc/pam.d*
- Files named for program they support
- Syntax

  type control module-path module-arguments
- */etc/pam.d/other* – default configuration

© 2002 Carnegie Mellon University Module 9: Host System Hardening - slide 36

Most LINUX systems use Pluggable Authentication Modules (PAM). PAM provides a framework for using different authentication mechanisms on a computer system. For example, if a biometric device is to be integrated into the identification and authorization scheme, the vendor will likely provide the libraries referenced by the PAM configuration files. These libraries are invoked as needed by directives in the PAM configuration files that tell when and where to use that device and how to evaluate its results. For example, the biometric device could be considered sufficient for identifying a person or it could be considered only part of the larger login scheme. PAM provides the framework for supporting these schemes and more. The discussion that follows is from [Hernberg 00].

PAM configuration files are stored in the `/etc/pam.d` directory. On a LINUX system, that directory contains the following:

```
~$ cd /etc/pam.d
/etc/pam.d/$ ls
chfn      chsh    login   other   passwd  su      xlock
```

A given system may have a few more or a few less files in this directory, depending on what's installed. Whatever the details, notice that there is a file for each program on the system that authenticate users. Each file contains the PAM authentication configuration for the program it's named after (except for the `other` file, which is discussed about in a little bit).

To begin, here are the contents of the PAM configuration file for **login**:

```
[pam.d]$ cat login
# PAM configuration for login
auth        requisite  pam_securetty.so
auth        required   pam_nologin.so
auth        required   pam_env.so
auth        required   pam_unix.so nulok
account     required   pam_unix.so
session     required   pam_unix.so
session     optional   pam_lastlog.so
password    required   pam_unix.so nullok obscure min=4 max=8
```

PAM configuration files have the following syntax:

type *control* *module-path* *module-arguments*

Using the login configuration file (see above) as an example let's take a look a the syntax for PAM configuration files:

type

The *type* token tells PAM what type of authentication is to be used for this module. Modules of the same type can be "stacked", requiring a user to meet multiple requirements to be authenticated. PAM recognizes four types:

`account`: Determines whether the user is allowed to access the service, whether their passwords has expired, etc.

`auth`: Determines whether the user is who they claim to be, usually by a password, but perhaps by a more sophisticated means, such as biometrics.

`password`: Provides a mechanism for the user to change their authentication. Again, this is usually their password.

`session`: Things that should be done before and/or after the user is authenticated. This might include things such as mounting or unmounting the user home directory, logging their login and logout, and restricting or unrestricting the services available to the user.

In the login configuration file example above, there is at least one entry for each type. Since this the program that allows user to login (hence the name), it's understandable that it needs to access all of the different types of authentication.

control

The *control* token tells PAM what should be done if authentication by this module fails. PAM recognizes four control types:

`requisite`: Failure to authenticate via this module results in immediate denial of authentication.

`required`: Failure also results in denial of authentication, although PAM will still call all the other modules listed for this service before denying authentication.

`sufficient`: If authentication by this module is successful, PAM will grant authentication, even if a previous required module failed.

`optional`: Whether this module succeeds or fails is only significant if it is the only module of its type for this service.

In the configuration file for login, there are nearly always one of the different control types. Most of the required modules are *pam_unix.so* (the main authentication module), the single requisite module is *pam_securetty.so* (checks make sure the user is logging in on a secure console), and the only optional module is *pam_lastlogin.so* (the module that retrieves information on the user's most recent login).

module-path

The *module-path* tells PAM which module to use and (optionally) where to find it. Most configurations only contain the module's name, as is the case in our login configuration file. When this is the case, PAM looks for the modules in the default PAM module directory, normally `/usr/lib/security`. However, if your LINUX distribution conforms to the LINUX Filesystem standard, PAM modules can be found in `/lib/security`.

module-arguments

The *module-arguments* are arguments to be passed to the module. Each module has its own arguments. For example, in the example login configuration, the "`nullok`" ("null ok") argument is being passed to *pam_unix.so* module, indicating the a blank ("null") password is acceptable ("ok").

If the PAM configuration is stored in `/etc/pam.conf` rather than `/etc/pam.d directory`, the PAM configuration lines are a bit different. Rather than each service having its own configuration file, all configurations are stored in `/etc/pam.conf` with the service name as the first token in a configuration line. For example, the following line in `/etc/pam.d/login`:

```
auth        required   pam_unix.so nullok
```

becomes the following line in `/etc/pam.conf`:

```
login      auth        required   pam_unix.so nullok
```

Unfortunately, many LINUX distributions ship with user authentication that is not adequately secure. This section discusses some of the ways to make user authentication more secure. As always, do not be so naive as to think they make it invulnerable.

All of the files in `/etc/pam.d` contain the configuration information for a particular service. The notable exception to this rule is the `/etc/pam.d/other` file. This file contains the configuration information for any services that do not have their own configuration file. For example, is the (imaginary) **xyz** service attempted authentication, PAM would look for a `/etc/pam.d/xyz` file. Not finding one, authentication for **xyz** would be determined by the `/etc/pam.d/other` file. Since `/etc/pam.d/other` is the configuration to which PAM services fallback, it is important that it is secure. The following discusses two secure configurations of `/etc/pam.d/other`, one that is quite paranoid and the other of which is gentler.

A paranoid configuration of `/etc/pam.d/other` is as follows:

```
auth      required      pam_deny.so
auth      required      pam_warn.so
account   required      pam_deny.so
account   required      pam_warn.so
password  required      pam_deny.so
password  required      pam_warn.so
session   required      pam_deny.so
session   required      pam_warn.so
```

With this configuration, whenever an unknown service attempts to access any of the four configuration types, PAM denies authentication (via the *pam_deny.so* module) and then logs a **syslog** warning (via the *pam_warn.so* module). Short of a bug in PAM, this configuration is brutally secure. The only problem with that brutality is it may cause problems if the configuration information of another service is accidentally deleted. If the `/etc/pam.d/login` is mistakenly deleted, no one would be able to login to your system

Here's configuration that isn't quite so mean:

```
auth      required      pam_unix.so
auth      required      pam_warn.so
```

```
            account   required      pam_unix.so
            account   required      pam_warn.so
            password  required      pam_deny.so
            password  required      pam_warn.so
            session   required      pam_unix.so
            session   required      pam_warn.so
```

This configuration will allow an unknown service to authenticate (via the *pam_unix.so* module), though it will not allow it to change the user's password. Although it allows authentication by unknown services, it logs a **syslog** warning whenever such a service attempts authentication.

So, when configuring the PAM system, the first task is to implement – or at least review – the `/etc/pam.d/other` configuration file. If there is a need to grant new service authentication privileges, simply create a PAM configuration file for that service.

On most LINUX systems, there are many "dummy" user accounts, created to assign privileges to certain system services like ftp, webservers, and mail gateways. Having these accounts secures a system because if these services are compromised, an attacker will only gain the limited privileges available to the dummy account, rather than the full privileges of a service running as root. However, allowing these dummy account login privileges is a security risk, as they usually have blank (null) passwords.

The PAM configuration option that enables null passwords is the `nullok` module-argument. Remove this argument from any modules of 'auth' type for services that allow login. This is usually the login service, but it may also include services like **rlogin** and **ssh**. Hence, the following line in `/etc/pam.d/login`:

```
            auth       required   pam_unix.so     nullok
```

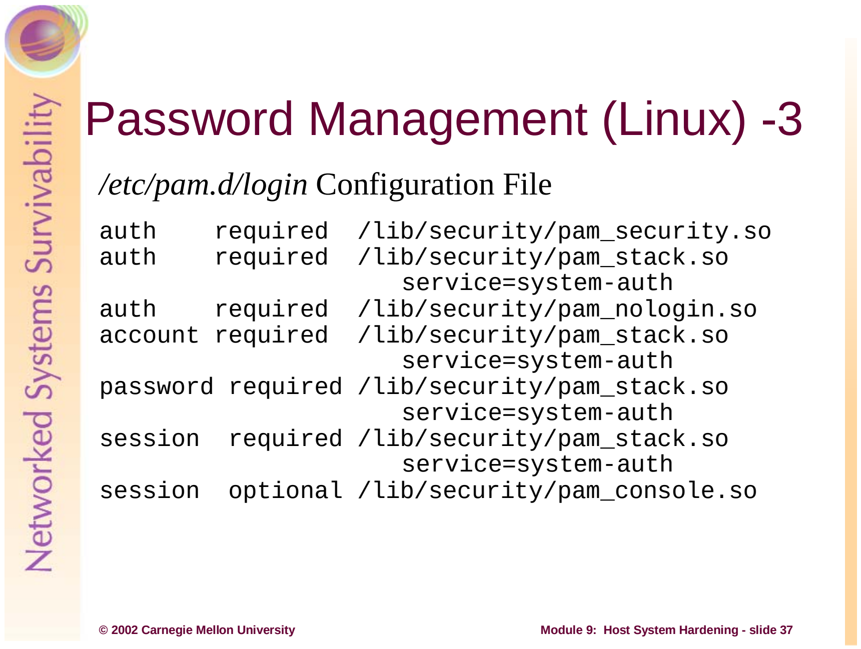
should be changed to:

```
            auth       required   pam_unix.so
```

Looking at the files in `/etc/pam.d`; notice that directory listing for the files */etc/pam.d* contain several configuration files for programs users don’t typically use and likely several you've never heard of. Although allowing authentication to these services probably won't open any huge security holes, a more secure approach is to deny them authentication. The best way to disable PAM authentication for these programs is to rename these files. By not finding the file named after the service requesting authentication, PAM will fallback to the secure `/etc/pam.d/other` default configuration file. If it is later found that one of these programs is needed, the file can be simply renamed the to its original name and everything will work as it was intended.

While password-cracking tools can be by attackers used to compromise a system, system administrators can also use them as proactive tool to ensure the strength of passwords on their system. There is a PAM module that uses a library from the **Crack**[10] password cracking system to check the strength of a users password whenever it changed. When this module is installed, the user can only change their password to one that meets the minimum password strength.

[10] http://www.users.dircon.co.uk/~crypto



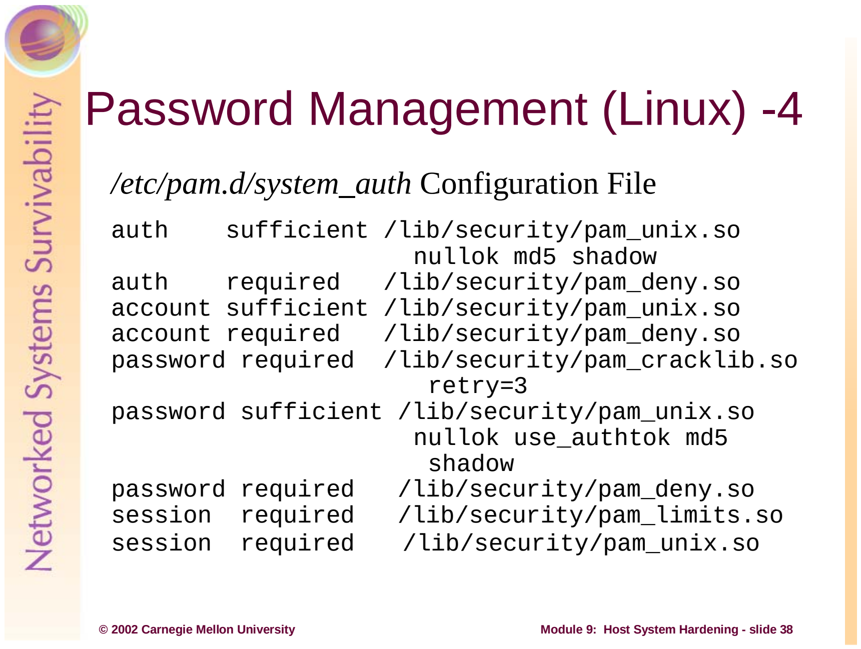
Networked Systems Survivability

Password Management (Linux) -3

/etc/pam.d/login Configuration File

```
auth    required  /lib/security/pam_security.so
auth    required  /lib/security/pam_stack.so
                     service=system-auth
auth    required  /lib/security/pam_nologin.so
account required  /lib/security/pam_stack.so
                     service=system-auth
password required /lib/security/pam_stack.so
                     service=system-auth
session  required /lib/security/pam_stack.so
                     service=system-auth
session  optional /lib/security/pam_console.so
```

© 2002 Carnegie Mellon University   Module 9: Host System Hardening - slide 37



Networked Systems Survivability

Password Management (Linux) -4

/etc/pam.d/system_auth Configuration File

```
auth    sufficient /lib/security/pam_unix.so
                     nullok md5 shadow
auth    required   /lib/security/pam_deny.so
account sufficient /lib/security/pam_unix.so
account required   /lib/security/pam_deny.so
password required  /lib/security/pam_cracklib.so
                      retry=3
password sufficient /lib/security/pam_unix.so
                     nullok use_authtok md5
                      shadow
password required   /lib/security/pam_deny.so
session  required   /lib/security/pam_limits.so
session  required    /lib/security/pam_unix.so
```

© 2002 Carnegie Mellon University   Module 9: Host System Hardening - slide 38

Let’s look at a more involved example. The files shown here are from RedHat LINUX 7.2. First, here is the */etc/pam.d/login* configuration file:

```
auth        required     /lib/security/pam_securetty.so
auth        required     /lib/security/pam_stack.so service=system-auth
auth        required     /lib/security/pam_nologin.so
account     required     /lib/security/pam_stack.so service=system-auth
password    required     /lib/security/pam_stack.so service=system-auth
session     required     /lib/security/pam_stack.so service=system-auth
session     optional     /lib/security/pam_console.so
```

and because it is incorporated through the stacking module, here is *the /etc/pam.d/system_auth* configuration file:

```
auth        sufficient  /lib/security/pam_unix.so nullok md5 shadow
auth        required    /lib/security/pam_deny.so
account     sufficient  /lib/security/pam_unix.so
account     required    /lib/security/pam_deny.so
password    required    /lib/security/pam_cracklib.so retry=3
password    sufficient  /lib/security/pam_unix.so nullok use_authtok \
                                        md5 shadow
password    required    /lib/security/pam_deny.so
session     required    /lib/security/pam_limits.so
session     required    /lib/security/pam_unix.so
```

- The `auth` configuration rules are used when a person tries to identify themselves to the computer system for the purposes of logging in. Normally this happens with a username/password pair, but PAM is sufficiently flexible that other identification schemes can also be configured and used.
  1. A root login must be occurring on a secure terminal, that is one of the terminals specified in the */etc/securettys* file. For regular users, this always fails. Since this is required, if the root login is not on a secure terminal, the login fails
  2. The *pam_stack.so* rule provides the capability of calling a subroutine. In this case, the rules used come from the */etc/pam.d/system_auth* configuration file.
     a. The *pam_unix.so* rule is the standard UNIX authentication module. It uses standard calls from the system's libraries to retrieve and set account information as well as authentication. Usually this is obtained from the */etc/passwd* and the */etc/shadow* file as well if shadow is enabled. Note that if the user can be logged in, this rule is sufficient.

        The default action of this module is to not permit the user access to a service if their *official* password is blank. The `nullok` argument overrides this default.

        In the case of conventional UNIX databases (which store the password encrypted) the `md5` argument is used to do the encryption with the MD5 function as opposed to the *conventional* `crypt(3)` call. It has no meaning in the `auth` context.

        The `shadow` argument says to use the /etc/shadow file if present.
     b. The *pam_deny.so* rule says that if the previous was not successful, the login should be denied. This rule is required so no further rules are tried.
  3. The *pam_nologin.so* rule checks the */etc/nologin* file to see if logins are allowed. For root, this always succeeds; for non-root, it can fail if */etc/nologin* is present.
- The `account` configuration rule performs non-authentication based account management. It is typically used to restrict/permit access to a service based on the time of day, currently available system resources (maximum number of users) or perhaps the location of the applicant user - root login only on the console.
  1. The *pam_stack.so* rule provides the capability of calling a subroutine. In this case, the rules used come from the */etc/pam.d/system_auth* configuration file.
     a. The *pam_unix.so* module in the account context performs the task of establishing the status of the user's account and password. In the case of the latter, it may offer advice to the user on changing their password or, through the

PAM_AUTHTOKEN_REQD return, delay giving service to the user until they have established a new password.

- The password configuration rule is required for updating the authentication token associated with the user. Typically, there is one module for each `challenge/response' based authentication (auth) module-type.

  1. The *pam_stack.so* rule provides the capability of calling a subroutine. In this case, the rules used come from the */etc/pam.d/system_auth* configuration file.

     a. The *pam_cracklib.so* module works in the following manner: it first calls the *Cracklib* routine to check the strength of the password; if crack likes the password, the module does an additional set of strength checks. These checks are:

        - **Palindrome -** Is the new password a palindrome of the old one?
        - **Case Change Only -** Is the new password the old one with only a change of case?
        - **Similar -** Is the new password too much like the old one? This is primarily controlled by one argument, difok which is a number of characters that if different between the old and new are enough to accept the new password, this defaults to 10 or 1/2 the size of the new password whichever is smaller. To avoid the lockup associated with trying to change a long and complicated password, difignore is available. This argument can be used to specify the minimum length a new password needs to be before the difok value is ignored. The default value for difignore is 23.
        - **Simple -** Is the new password too small? This is controlled by 5 arguments minlen, dcredit, ucredit, lcredit, and ocredit. See the section on the arguments for the details of how these work and there defaults.
        - **Rotated -** Is the new password a rotated version of the old password?
        - **Already used -** Was the password used in the past? Previously used passwords are to be found in */etc/security/opasswd*.

        This module with no arguments will work well for standard UNIX password encryption. With md5 encryption, passwords can be longer than 8 characters and the default settings for this module can make it hard for the user to choose a satisfactory new password. Notably, the requirement that the new password contain no more than 1/2 of the characters in the old password becomes a non-trivial constraint. For example, an old password of the form "the quick brown fox jumped over the lazy dogs" would be difficult to change. In addition, the default action is to allow passwords as small as 5 characters in length. For md5 systems it can be a good idea to increase the required minimum size of a password. One can then allow more credit for different kinds of characters but accept that the new password may share most of these characters with the old password.

        This rule has to like the password selected because it is required.

        retry=N: The default number of times this module will request a new password (for strength-checking) from the user is 1. Using this argument this can be increased to N.

     b. The *pam_unix.so* module in the password context performs the task of updating the user's password.

        The nullok argument is used to permit the changing of a password *from* an empty one. Without this argument, empty passwords are treated as account-locking ones.

The argument `use_authtok` is used to *force* this module to set the new password to the one provided by the previously stacked `password` module (this is used in an example of the stacking of the *Cracklib* module documented above).

c. The *pam_deny.so* rule says that if the previous was not successful, the password change was not successful. This rule is required so no further rules are tried.

- The `session` configuration rules do things that need to be done for the user before/after they can be given service. Such things include the logging of information concerning the opening/closing of some data exchange with a user, mounting directories, etc.

  1. The *pam_stack.so* rule provides the capability in essence of calling a subroutine. In this case, the rules used come from the */etc/pam.d/system_auth* configuration file.

     a. The *pam_limits.so* module places resource limits on users' sessions through the contents of the configuration file, `/etc/security/limits.conf`. Users with a `uid=0` are not affected by this restriction.

     b. The *pam_unix.so* module logs the username and service-type to `syslog(3)`. Messages are logged at the beginning and end of the user's session.

  2. The *pam_console.so* module gives users at the physical console (virtual terminals and local **xdm**-managed X sessions by default, but that is configurable) capabilities that they would not otherwise have, and to take those capabilities away when they are no longer logged in at the console. It provides two main kinds of capabilities: file permissions and authentication.

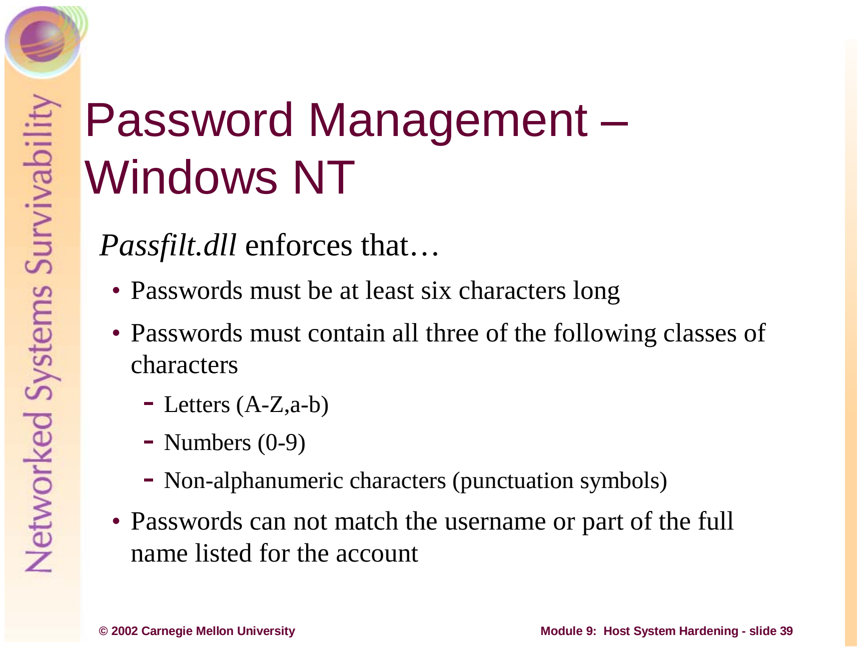
     When a user logs in at the console and no other user is currently logged in at the console, *pam_console.so* will change permissions and ownership of files as described in the file */etc/security/console.perms*. That user may then log in on other terminals that are considered part of the console, and as long as the user is still logged in at any one of those terminals, that user will own those devices. When the user logs out of the last terminal, the console may be taken by the next user to log in. Other users who have logged in at the console during the time that the first user was logged in will not be given ownership of the devices unless they log in on one of the terminals; having done so on any one terminal, the next user will own those devices until he or she has logged out of every terminal that is part of the physical console. Then the race can start for the next user. In practice, this is not a problem; the physical console is not generally in use by many people at the same time, and *pam_console.so* justtries to do the right thing in weird cases.

To find out if a given executable uses PAM, execute the command **ldd**. For example, the resulting output for */bin/login* on a RedHat Linux 7.2 system:

```
% ldd /bin/login
        libcrypt.so.1 => /lib/libcrypt.so.1 (0x40027000)
        libpam.so.0 => /lib/libpam.so.0 (0x40054000)
        libdl.so.2 => /lib/libdl.so.2 (0x4005c000)
        libpam_misc.so.0 => /lib/libpam_misc.so.0 (0x40060000)
        libc.so.6 => /lib/i686/libc.so.6 (0x40063000)
        /lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
```

Note the *libpam.so.0*; this is a PAM module and it tells the administrator that the */bin/login* program does use PAM.

More information about PAM is available at [Morgan 01] and [SUN 02]. Finally, see [CMU 01] for an explanation of a biometrics device – a camera in this case – that has been added to a LINUX system as an ancillary identification device.



Password Management – Windows NT

Passfilt.dll enforces that…

- Passwords must be at least six characters long
- Passwords must contain all three of the following classes of characters
  - Letters (A-Z,a-b)
  - Numbers (0-9)
  - Non-alphanumeric characters (punctuation symbols)
- Passwords can not match the username or part of the full name listed for the account

Networked Systems Survivability

© 2002 Carnegie Mellon University

Module 9: Host System Hardening - slide 39

In Windows NT Server 4.0, administrators must install a new *Passfilt.dll* library and place an appropriate entry in the registry in order for this functionality to be enabled. *Passfilt.dll* implements the following password policy:

1. Passwords must be at least six (6) characters long.
2. Passwords must contain characters from at least three (3) of the following four (4) classes:

```
Description                                Examples
------------------------------------------------------
English upper case letters                 A, B, C, ... Z
English lower case letters                 a, b, c, ... z
Westernized Arabic numerals                0, 1, 2, ... 9
Non-alphanumeric ("special characters")
        such as punctuation symbols
```

3. Passwords may not contain your user name or any part of your full name.

These requirements are hard-coded in the *Passfilt.dll* file and cannot be changed through the user interface or registry. If you wish to raise or lower these requirements, you must write your own .dll and implement it in the same fashion as the Microsoft version that is available with Windows NT 4.0 Service Pack 2.

How to Install Strong Password Filtering: To ensure Strong Password functionality occurs throughout your domain structure, make the following changes on all primary domain controllers (or stand-alone servers, where needed).

Passfilt.dll is not necessary on backup domain controllers since the PDC is the only machine where changes to the domain accounts database are made. However, it should be installed on all BDCs because they can be promoted to PDC. If a BDC without *Passfilt.dll* is promoted to PDC, then strong password enforcement will be lost but there will be no other adverse effects.

WARNING: Using the Registry Editor incorrectly can cause serious problems that may require you to reinstall your operating system. Microsoft cannot guarantee that problems resulting from the incorrect use of Registry Editor can be solved. Use Registry Editor at your own risk.

For information about how to edit the registry, view the "*Changing Keys and Values*" Help topic in Registry Editor (**Regedit.exe**) or the "*Add and Delete Information in the Registry*" and "*Edit Registry Data*" Help topics in **Regedt32.exe**. Note that you should back up the registry before you edit it. If you are running Windows NT or Windows 2000, you should also update your Emergency Repair Disk (ERD).

1. Install the latest Windows NT 4.0 service pack.
2. Copy *Passfilt.dll* to the *%SYSTEMROOT%\SYSTEM32* folder.
3. Start Registry Editor (**Regedt32.exe**).
4. Locate and click the following key in the registry:

   `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa`

5. On the *Edit* menu, click *Add Key***,** and then add the following registry key:

   `Notification Packages`

   NOTE: If the `Notification Packages` key already exists, proceed to the next step.

6. Click the `Notification Packages` key.
7. On the *Edit* menu, click *Add Value***,** and then add the following registry value:

   Value name: `FPNWCLNT`
   Data type: `REG_MULTI_SZ`
   Value data: `PASSFILT`

   NOTE: If the `FPNWCLNT` value is already present:

   a. Click the `FPNWCLNT` value.
   b. On the *Edit* menu, click *Multi String***.**
   c. Type `PASSFILT`, and then click **OK.**

8. Quit Registry Editor.
9. Restart the computer.

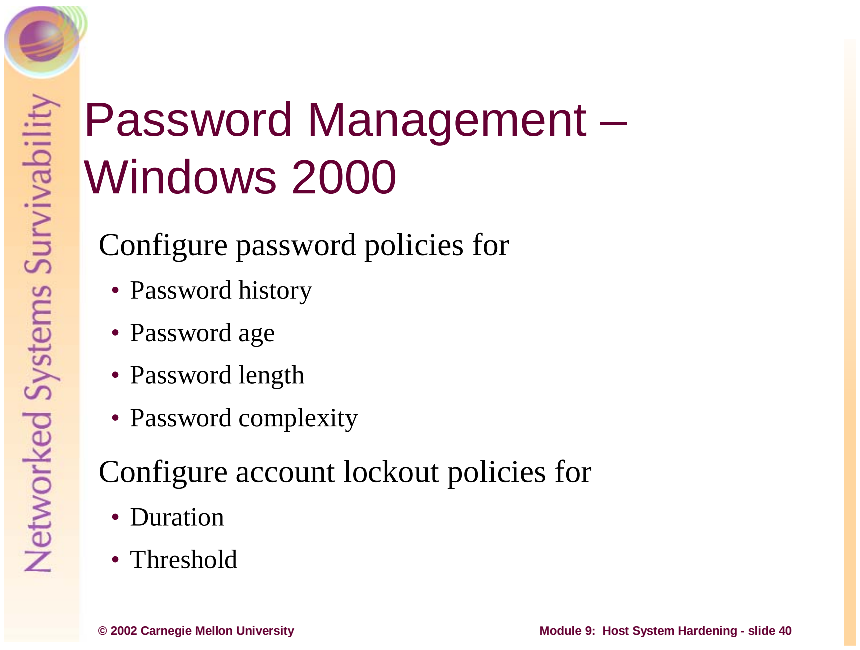
For additional information, click the article numbers below to view the articles in the Microsoft Knowledge Base:

Q151082[11] Password Change Filtering & Notification in Windows NT

Q174075[12] Strong Passwords With Passfilt.dll Are Not Enforced

Q174076[13] Invalid Password Message When Strong Passwords Are Required

[11] http://support.microsoft.com/default.aspx?scid=kb;en-us;Q151082
[12] http://support.microsoft.com/default.aspx?scid=kb;en-us;Q174075
[13] http://support.microsoft.com/default.aspx?scid=kb;en-us;Q174076



Password Management – Windows 2000

Configure password policies for

- Password history
- Password age
- Password length
- Password complexity

Configure account lockout policies for

- Duration
- Threshold

Networked Systems Survivability
© 2002 Carnegie Mellon University
Module 9: Host System Hardening - slide 40

Windows 2000 Server includes the strong password functionality first provided in Microsoft Windows NT Server 4.0 Service Pack 2 (SP2). Administrators can use strong password requirements by using a Group Policy Object (GPO) because the required dynamic-link library (DLL) is included with Windows 2000 [Microsoft 00].

To implement strong password requirements for your domain, configure a group policy object linked to your domain:

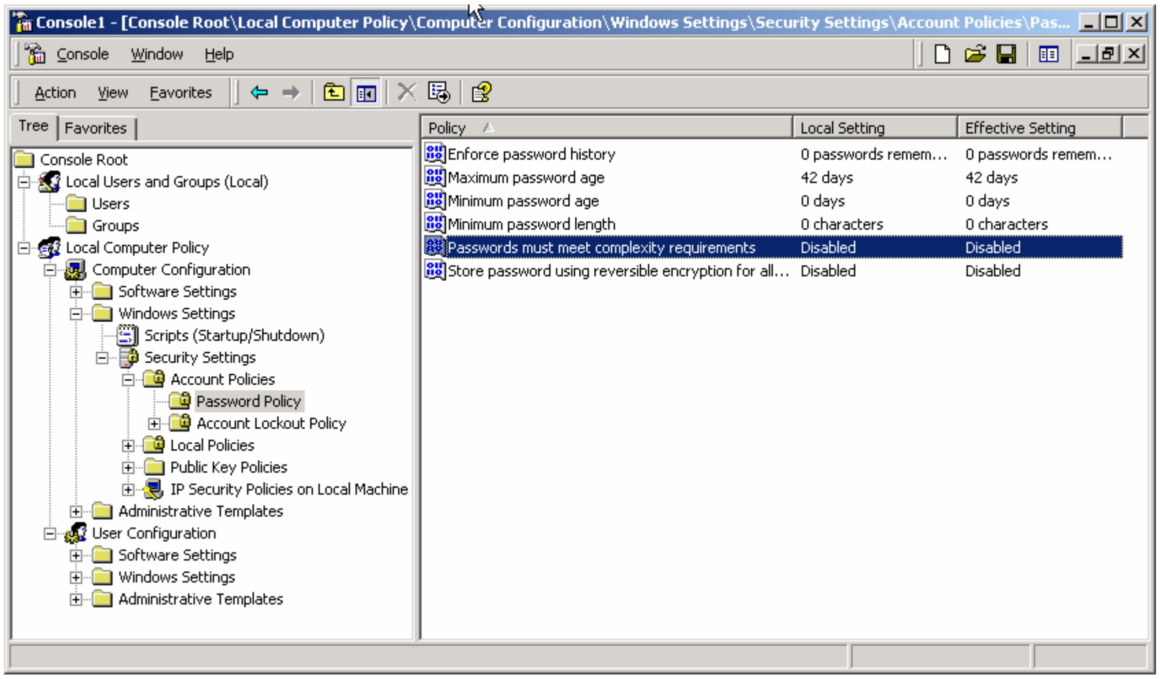
1. Start the Group Policy Editor Microsoft Management Console (MMC) snap-in, focused on the appropriate group policy object. This must be a group policy object linked to the entire domain. You can directly edit the Default Domain Controllers Policy GPO if you are using this policy object.

   A quick method of starting the Group Policy Editor MMC snap-in with its focus pointed to this GPO is:

   a. Right-click the domain object in the Active Directory Users and Computers MMC snap-in, and then click *Properties* .
   b. Click the *Group Policy* tab.
   c. Click the *Default Domain Policy* GPO link, and then click *Edit***.**

2. Navigate to the following node in the group policy object:

```
Group Policy Object Policy
 Computer Configuration
  Windows Settings
   Security Settings
    Account Policies
     Password Policy
```



Policy	Local Setting	Effective Setting
Enforce password history	0 passwords remem...	0 passwords remem...
Maximum password age	42 days	42 days
Minimum password age	0 days	0 days
Minimum password length	0 characters	0 characters
Passwords must meet complexity requirements	Disabled	Disabled
Store password using reversible encryption for all...	Disabled	Disabled

3. To configure the password history, double click on *Enforce password history* and configure how many passwords to remember.

4. To configure the password age, double click on *Maximum password age* and *Minimum password age* and configure them to meet your needs.

5. To configure the password length, double click on *Minimum password length* and configure your minimum password size Note that the minimum length cannot exceed the system's maximum length, which is 14 characters.

6. To enable additional checks on password quality, double click on *Passwords must meet complexity requirements of installed password filter* to produce the application dialog box. Change the template setting to *Enabled* to activate the strong password requirement. Note that these complexity requirements are enforced upon password change or creation.

   By enabling this policy, passwords will be required to pass the following tests following:

   a. Does not contain all or part of the user's account name
   b. Is at least six characters in length
   c. Contains characters from three of the following four categories:
      - English upper case characters (A..Z)
      - English lower case characters (a..z)
      - Base 10 digits (0..9)

- No alphanumeric (For example, !,$#,%)

As with all policy settings, the change is not applies until the next time group policy objects are applied to your domain controllers [Microsoft 00a].

To configure the account lockout policies, do the following:

1. Start the Group Policy Editor Microsoft Management Console (MMC) snap-in, focused on the appropriate group policy object. This must be a group policy object linked to the entire domain. You can directly edit the Default Domain Controllers Policy GPO if you are using this policy object.

   A quick method of starting the Group Policy Editor MMC snap-in with its focus pointed to this GPO is:

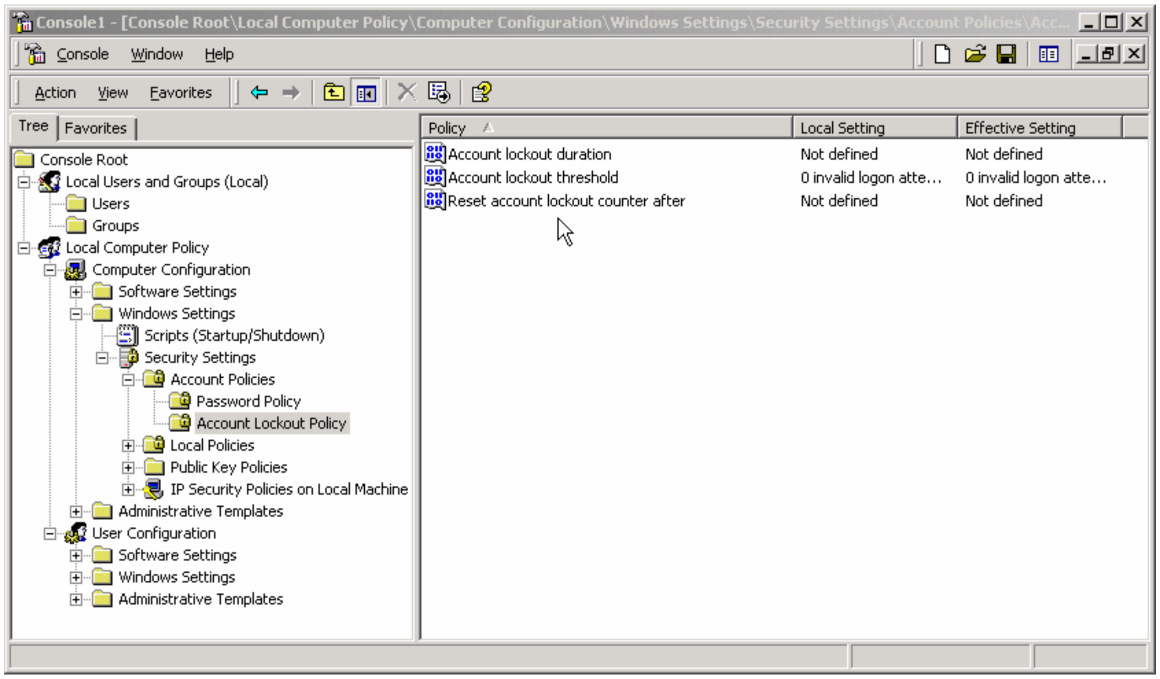
   a. Right-click the domain object in the Active Directory Users and Computers MMC snap-in, and then click *Properties***.**

   b. Click the *Group Policy* tab**.**

   c. Click the *Default Domain Policy* GPO link, and then click *Edit***.**

2. Navigate to the following node in the group policy object:

```
Group Policy Object Policy
 Computer Configuration
  Windows Settings
   Security Settings
    Account Policies
      Account Lockout Policies
```


Console1 - [Console Root\Local Computer Policy\Computer Configuration\Windows Settings\Security Settings\Account Policies\Acc...
Console
Window
Help
Action
View
Favorites
Tree
Favorites

```
Console Root
Local Users and Groups (Local)
  Users
  Groups
Local Computer Policy
  Computer Configuration
    Software Settings
    Windows Settings
      Scripts (Startup/Shutdown)
      Security Settings
        Account Policies
          Password Policy
          Account Lockout Policy
        Local Policies
        Public Key Policies
        IP Security Policies on Local Machine
    Administrative Templates
  User Configuration
    Software Settings
    Windows Settings
    Administrative Templates
```

Policy	Local Setting	Effective Setting
Account lockout duration	Not defined	Not defined
Account lockout threshold	0 invalid logon atte...	0 invalid logon atte...
Reset account lockout counter after	Not defined	Not defined

3. To configure the account lockout duration, double click on *Account lockout duration* and configure how long in minutes the account is locked out before automatically becoming

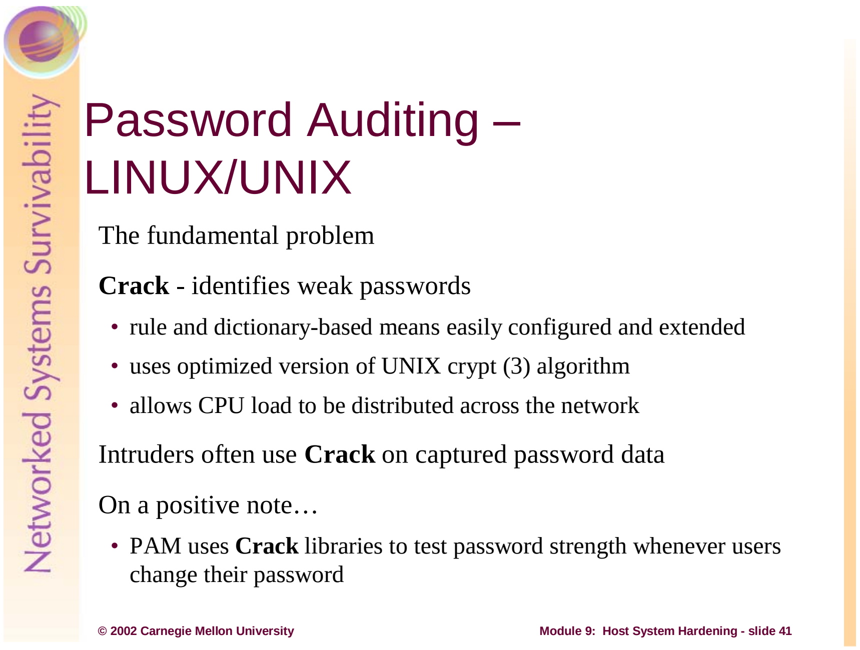
unlocked. The range is 1 to 99999 minutes. You can specify that the account will be locked out until an administrator explicitly unlocks it by setting the value to 0.

4. To configure the account lockout threshold, double click on the *Account lockout threshold* and configure the number of failed logon attempts that will cause a user account to be locked out. A locked out account cannot be used until an administrator resets it or the account lockout duration has expired. You can set values between 1 and 999 failed logon attempts, or you can specify that the account will never be locked out by setting the value to 0.

5. To configure when the count of failed logins is reset, double click on Reset account lockout after and configure the number of minutes that must elapse after a failed logon attempt before the bad logon attempt counter is reset to 0 bad logons. The range is 1 to 99999 minutes.

   By default, this policy is not defined, since it only has meaning when an Account lockout threshold is specified.

Again, as with all policy settings, the change is not applies until the next time group policy objects are applied to your domain controllers



Password Auditing – LINUX/UNIX

The fundamental problem

Crack - identifies weak passwords

- rule and dictionary-based means easily configured and extended
- uses optimized version of UNIX crypt (3) algorithm
- allows CPU load to be distributed across the network

Intruders often use **Crack** on captured password data

On a positive note…

- PAM uses **Crack** libraries to test password strength whenever users change their password

Networked Systems Survivability
© 2002 Carnegie Mellon University
Module 9: Host System Hardening - slide 41

There is a fundamental problem with password encryption in most operating systems in use today. The problem is that an encrypted version of passwords as well as the encryption algorithms is all too easily available to those with access to either the system or the network to which it is connected. While these algorithms are sufficiently strong so that the encrypted version of a password cannot be decrypted, today's computing power provides the surreptitious user with sufficient resources to attack that encrypted data through brute-force techniques. In other words, encrypted password information cannot be decrypted, but CPU speed provides enough power that dictionaries and their variations can be encrypted fast enough that the clear text equivalent of account passwords can be discovered in a timely manner.

This is the premise upon which several free and commercial password auditing tools have been built. Unfortunately, it is still the case that unwanted access to computer systems is achieved by exploiting weak passwords. It is the responsibility of the systems administrator to audit passwords in an attempt to weed out those considered to be weak.

The premiere free password-auditing tool for LINUX/UNIX-based systems is named **crack**[14]. It is distributed in source code form and runs on virtually every LINUX/UNIX version. It is network-aware, which means that the task of auditing passwords can be spread out over several computer systems in an attempt to discover passwords more quickly. **Crack** forms the basis for the *cracklib* subroutines used by the PAM authentication modules previously described.

Simply put, **crack** is a rule-based tool that alters clear text words from various dictionaries – and there are many, many dictionaries available – and summarily encrypts them with an optimized version of **crypt**. **Crack** then compares the encrypted result with that found in the */etc/password* or */etc/shadow* file on a LINUX/UNIX system, declaring success when the two match.

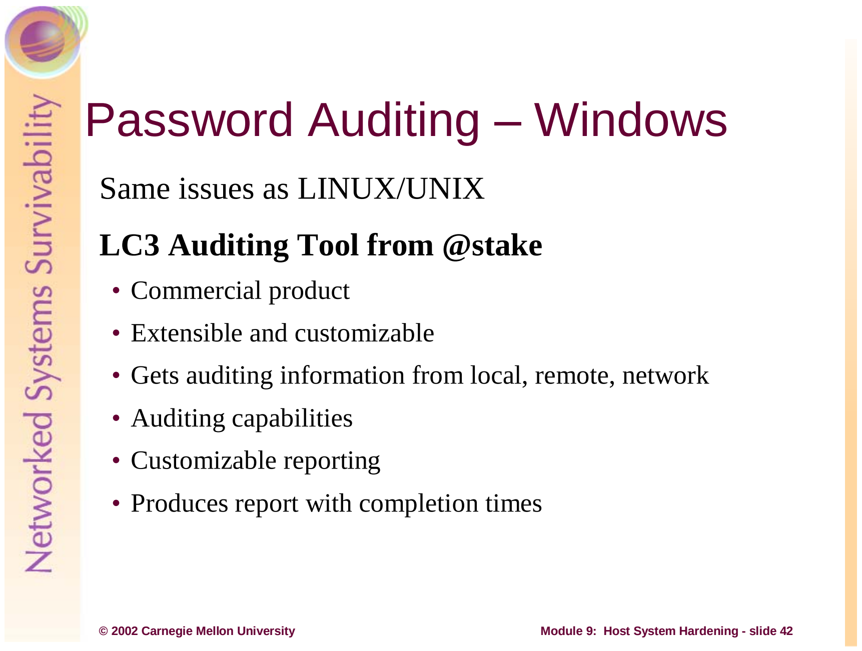
Here are two simple examples of the language that **crack** uses to alter words in its dictionaries:

- `/?ul` – For all dictionary words that have upper case letters, convert them to lower case.
- `/ese3u` – For all dictionary words which contain the letter "e", replace it with the digit "3", and force the rest of the word to uppercase.

[14] http://www.users.dircon.co.uk/~crypto

The *cracklib* used in PAM modifies dictionary entries with its rules to see if the user-proposed password matches one of those modifications, rejecting the candidate upon finding a match.

The **crack** password-auditing tool is a LINUX/UNIX tool used to discover accounts with weak passwords. It is rule-driven and dictionary-driven, meaning that new rules can be added to aid in transforming dictionaries, and environment-specific dictionaries can be added to augment those already provided. For example, at a university, an administrator could add a dictionary containing all faculty surnames, building names, and local street names.



Password Auditing – Windows

Same issues as LINUX/UNIX

LC3 Auditing Tool from @stake

- Commercial product
- Extensible and customizable
- Gets auditing information from local, remote, network
- Auditing capabilities
- Customizable reporting
- Produces report with completion times

© 2002 Carnegie Mellon University Module 9: Host System Hardening - slide 42

Networked Systems Survivability

Windows can also fall victim to password cracking attacks. Fundamentally the issues are the same: the encrypted versions of passwords and the encryption algorithms are easily available to a would-be intruder. The same kind of brute-force attacks can also be waged against Windows systems as well.

One of the premier password auditing tools for Windows systems is **LC3**[15]. It is a derivative of L0pht Crack, the name by which it was originally known, which is a variation on **Crack** for LINUX/UNIX systems. The following sections are taken from its documentation.

Introduction

Passwords have received increasing attention as a source of vulnerability in individual machines and entire networks. Security experts from industry, government, and academia agree that weak passwords represent one of the ten most critical Internet security threats. **LC3** offers an easy-to-use and adaptable way to identify and assess password vulnerability.

LC3 uses a variety of sources and methods to retrieve passwords from the operating system, and provides feedback about the strength of passwords based on the types of audit required to recover the password and the length of time required to do so. The end result is a state of the art tool for password auditing and recovery that serves to guide organizational policies and procedures.

There are many uses for auditing user passwords. First and foremost is for a system administrator to audit the strength of the passwords their users are using. Weak passwords represent vulnerability points for any organization, and uncovering these vulnerabilities is the first step toward reducing them. Administrators can use corporate password policies, and for some environments, filtered password generators to improve the quality of passwords used in their organizations. But without testing the passwords chosen by users or generated against a real world password auditor, the administrator is taking a chance at the time required for an external attacker or malicious insider to uncover the passwords.

In other cases **LC3** can be used to streamline the migration or upgrading of users from one authentication system to another. Lastly, **LC3** can serve to recover a lost or forgotten password to permit re-entry to a system from which an administrator is locked out.

[15] http://www.atstake.com/research/lc3

What’s New in LC3

LC3 adds numerous enhancements to the critically acclaimed capabilities of its L0phtCrack predecessors:

Support for Windows 2000: LC3 now runs cleanly on Windows 2000. It can extract unencrypted password hashes from systems that use Microsoft's *SYSKEY* protection, and it uses an updated packet sniffer that supports Windows 2000 systems.

Works with International Character Sets: LC3 supports international language input locales, allowing it to work with operating systems and passwords based on single byte character sets including those for European, Cyrillic, Greek, Hebrew, Arabic, and other languages.

Distributed Cracking: LC3 lets an administrator speed up a time-consuming password audit by breaking it into parts that can be run simultaneously on multiple machines.

Hide Cracked Passwords: LC3 gives administrators the option to know whether or not a password was cracked without knowing the password itself.

Audit Time: Password auditors get a quantitative comparison of password strength from **LC3**'s report on the time required to crack each password.

Wizard: LC3 offers a Wizard to help new password auditors configure and run their first audits quickly and easily.

Export: It's easier than ever to manipulate the results of a password audit by exporting results to a tab-delimited file.

Improved Product Support: Registered **LC3** users get email support with 24 hour or shorter response time.

New Dictionary: LC3 now includes a 250,000 word English dictionary for comprehensive English dictionary audits.

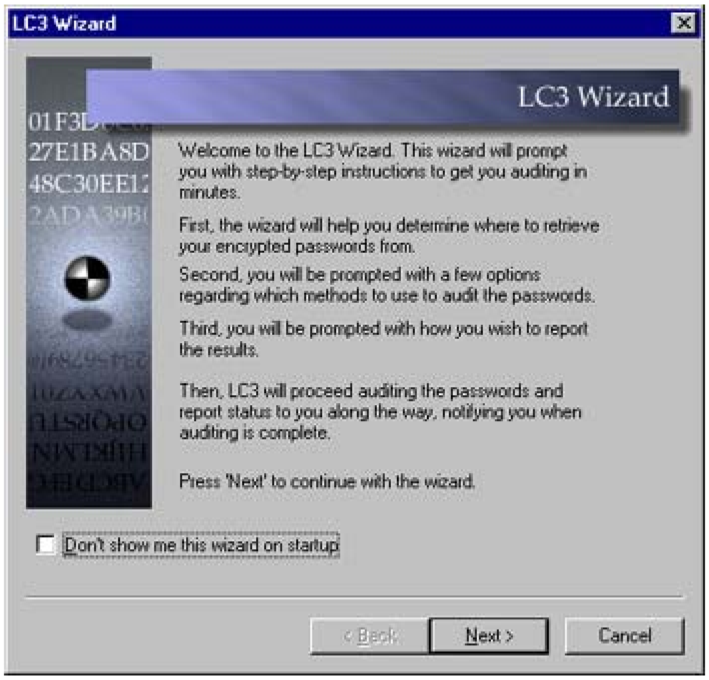
Manipulate Imported Passwords: Easily delete encrypted passwords directly from the **LC3** window to focus an audit on the passwords you want to audit.

Quick Start with the LC3 Wizard

The **LC3** Wizard, shown below, helps you quickly configure the settings needed to retrieve and audit passwords by the most common means, and provides a quick overview of the password auditing process.

The Wizard opens by default the first time you run **LC3**. Those already familiar with **LC3** password auditing may prefer to administer **LC3** without the Wizard, and can check the 'Don't show me this wizard on startup' checkbox. If you want to check out the Wizard at a later time, you may launch it from the **LC3** toolbar.

Click the Next button to use the Wizard.



Welcome to the LC3 Wizard. This wizard will prompt you with step-by-step instructions to get you auditing in minutes.

First, the wizard will help you determine where to retrieve your encrypted passwords from.

Second, you will be prompted with a few options regarding which methods to use to audit the passwords.

Third, you will be prompted with how you wish to report the results.

Then, LC3 will proceed auditing the passwords and report status to you along the way, notifying you when auditing is complete.

Press 'Next' to continue with the wizard.

Get Encrypted Passwords

The first step is to obtain encrypted passwords to audit. The wizard's dialog shown below lets you choose the source of encrypted passwords.

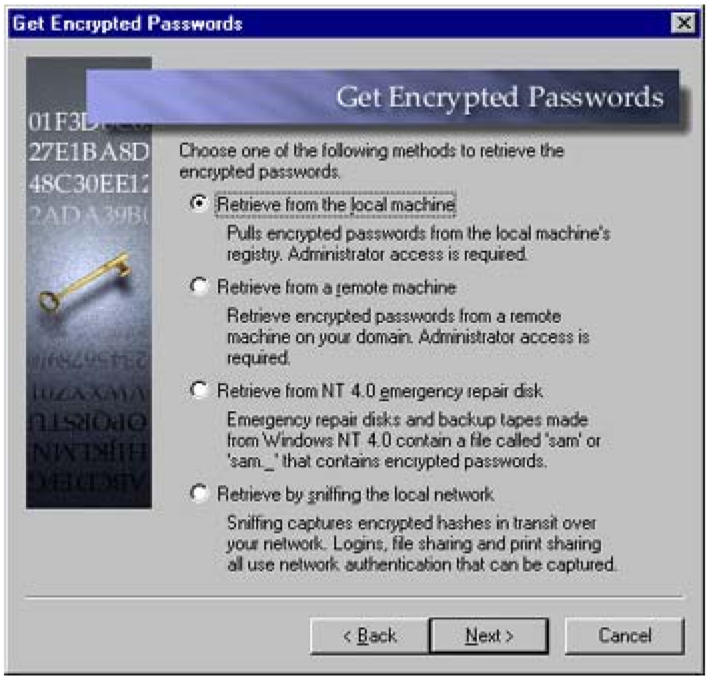
The first and most straightforward option extracts password hashes from the machine you're currently on.

The second option attempts to retrieve them from another network-accessible machine on which you have administrator privileges. (Note: password hashes retrieved with this approach will not be cracked if *SYSKEY* is enabled as is the default on Windows 2000, or if it is an NT system with *SYSKEY* enabled ...more about that later.)

The third option retrieves encrypted passwords from an NT emergency repair disk (note: Windows 2000 Emergency Repair Disks will not provide encrypted passwords).

The final option sniffs the network for password hashes that are traversing it.

After you've made your selection, click **Next.**


Get Encrypted Passwords

Get Encrypted Passwords

Choose one of the following methods to retrieve the encrypted passwords.

[✓ Retrieve from the local machine]

Pulls encrypted passwords from the local machine's registry. Administrator access is required.

Retrieve from a remote machine

Retrieve encrypted passwords from a remote machine on your domain. Administrator access is required.

Retrieve from NT 4.0 emergency repair disk

Emergency repair disks and backup tapes made from Windows NT 4.0 contain a file called 'sam' or 'sam._' that contains encrypted passwords.

Retrieve by sniffing the local network

Sniffing captures encrypted hashes in transit over your network. Logins, file sharing and print sharing all use network authentication that can be captured.

Choose Auditing Method

There's a tradeoff involved in the rigor with which **LC3** audits your passwords: the more rigorous the audit, the longer it takes to complete.

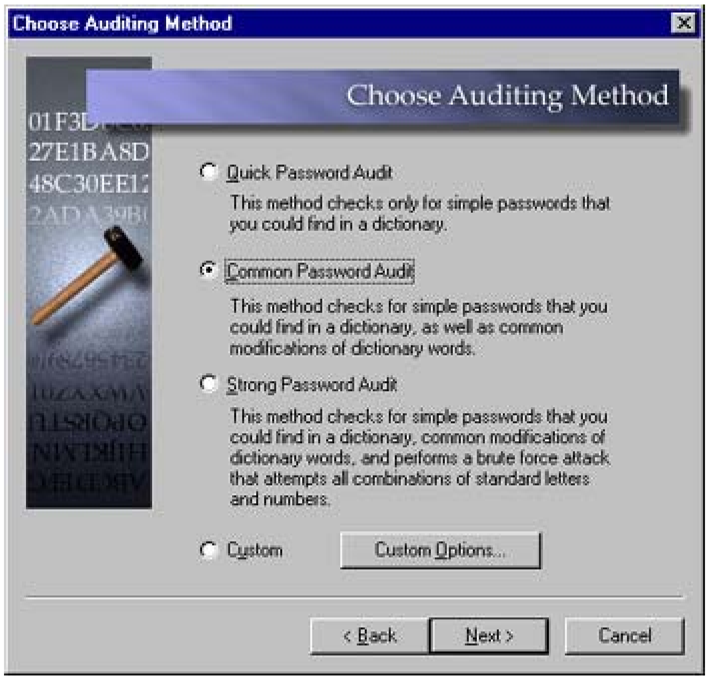
The Quick Password Audit takes only a minute to perform and tries every word in a 26,000 word dictionary file included with **LC3** to see if any words match the passwords you're examining.

Since many users comply with corporate password policies by slightly modifying dictionary words, the Common Password Crack programmatically varies the dictionary words by a chosen number of characters to see if any fit. This takes a couple minutes longer to complete.

The Strong Password Audit adds a brute force audit, trying all combinations of letters and numbers in seeking to compute passwords. This approach may take longer than a day to perform.

The Custom Audit lets you configure your audit more precisely. For example, you can change word files or choose a different character set for the brute force audit.

After you've made your selection in the window shown below, click Next.


Choose Auditing Method

Choose Auditing Method

( ) Quick Password Audit

This method checks only for simple passwords that you could find in a dictionary.

(●) Common Password Audit

This method checks for simple passwords that you could find in a dictionary, as well as common modifications of dictionary words.

( ) Strong Password Audit

This method checks for simple passwords that you could find in a dictionary, common modifications of dictionary words, and performs a brute force attack that attempts all combinations of standard letters and numbers.

( ) Custom

Pick Reporting Style

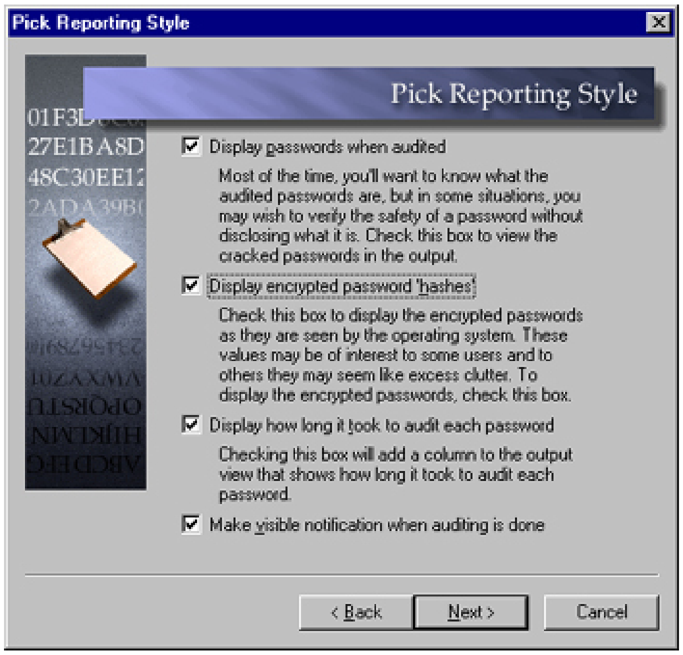
The next Wizard dialog lets you configure **LC3**'s reporting style.

You may hide the audited passwords so that your audit identifies whether or not a password is unacceptably weak without revealing what the password is. When choose to hide audited passwords, you must use the Audit Time to determine whether or not a particular password was successfully cracked.

You may list the time required to retrieve each password, in order to have a quantitative estimate of each password's comparative strength.

If you choose 'visible notification when auditing is done,' your computer will show an alert dialog when the audit completes, even if you're working in another application.

Make your selections, and then click Next.



☑ Display passwords when audited

Most of the time, you'll want to know what the audited passwords are, but in some situations, you may wish to verify the safety of a password without disclosing what it is. Check this box to view the cracked passwords in the output.

☑ Display encrypted password 'hashes'

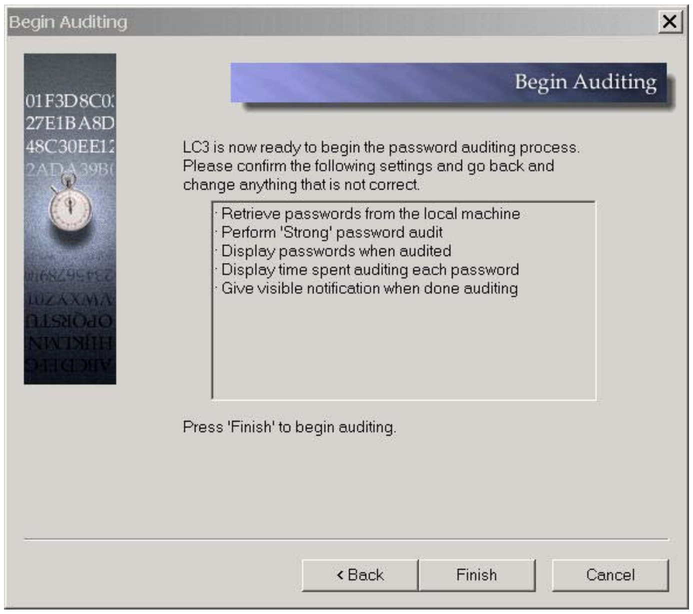
Check this box to display the encrypted passwords as they are seen by the operating system. These values may be of interest to some users and to others they may seem like excess clutter. To display the encrypted passwords, check this box.

☑ Display how long it took to audit each password

Checking this box will add a column to the output view that shows how long it took to audit each password.

☑ Make visible notification when auditing is done

Begin Auditing: Once you've finished the Reporting options, **LC3** is ready to audit. The Wizard's final dialog summarizes the settings you've chosen. When you click Finish, the retrieval and audit begins.



LC3 is now ready to begin the password auditing process. Please confirm the following settings and go back and change anything that is not correct.

- Retrieve passwords from the local machine
- Perform 'Strong' password audit
- Display passwords when audited
- Display time spent auditing each password
- Give visible notification when done auditing

Press 'Finish' to begin auditing.

Now let's take a closer look at some of the basic elements of password auditing with **LC3**.

Using LC3

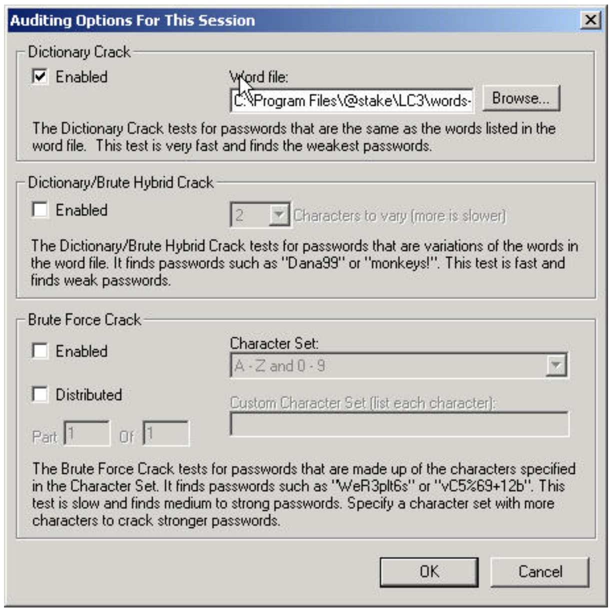
The operating system does not store users' passwords in their original clear-text form for security reasons. The actual user passwords are encrypted into hashed forms because they are sensitive information that can be used to impersonate any user, including the operating system administrator. The original password cannot be derived directly from a hashed password, so **LC3** does what any hacker does: it guesses. By automating the 'guessing' process, **LC3** reveals for administrators how tough a password is to crack.

LC3 works by first obtaining password hashes from the operating system, and then hashing possible password values. When there's a match between a target hash and a possible password hash, a password is found. Therefore, to do its thing, **LC3** must first obtain password hashes from the target system, and then use various cracking methods to retrieve the passwords.

Obtaining the Password Hashes: There are several different approaches to obtaining password hashes, depending on where they reside, and your access to them. **LC3** can obtain password hashes directly from the registry, from the file system, from backup tapes and repair disks, from Active Directory, or by recovering them as they traverse the network. This process is not always straightforward, so read the documentation on this subject carefully (it has been omitted here).

Cracking the Password Hashes: The cracking processes that generate password values provide several options that balance audit rigor against the time required to crack. Effective auditing therefore requires an understanding the underlying business goals, and the security thresholds necessary to meet them.

To configure the cracking methods applied in your session, choose Session .. Session Options or simply click the toolbar's Session Options button to open the 'Auditing Options For This Session' dialog. From there you can configure the auditing options for the password hashes you have retrieved. The figure below shows this.



Auditing Options For This Session

Dictionary Crack
[✓] Enabled
Word file: C:\Program Files\@stake\LC3\words-
The Dictionary Crack tests for passwords that are the same as the words listed in the word file. This test is very fast and finds the weakest passwords.

Dictionary/Brute Hybrid Crack
[ ] Enabled
2 Characters to vary (more is slower)
The Dictionary/Brute Hybrid Crack tests for passwords that are variations of the words in the word file. It finds passwords such as "Dana99" or "monkeys!". This test is fast and finds weak passwords.

Brute Force Crack
[ ] Enabled
[ ] Distributed
Part 1 Of 1
Character Set: A - Z and 0 - 9
Custom Character Set (list each character):
The Brute Force Crack tests for passwords that are made up of the characters specified in the Character Set. It finds passwords such as "WeR3plt6s" or "vC5%69+12b". This test is slow and finds medium to strong passwords. Specify a character set with more characters to crack stronger passwords.

The UserName Crack: The first cracking method **LC3** employs checks to see if any accounts have used the username as the password. You'll want to know about these weak passwords right away, and since this crack is nearly instantaneous, it is performed first in every audit.

Dictionary Crack: The fastest method for retrieving simple passwords is a dictionary crack. In a dictionary crack, **LC3** tests all the words in a dictionary or word file against the password hashes. When it finds a correct password, it displays the result. The dictionary crack will try words of any length, up to the 14-character limit (which Windows NT imposes, but Windows 2000 does not).

LC3 defaults to using a 25,000-word dictionary file named words-English that contains the most common English words. **LC3** also ships with a 250,000 dictionary called words-English-big, which can be used for more comprehensive dictionary audits. This file or any other word file you select is loaded into **LC3** based on the settings in the Session Options dialog.

Because of the structure of the LM hash, the cracking process analyzes the first and last seven characters of a possible password independently. So if the first seven characters match those of a word found in the dictionary, **LC3** will report these, even if subsequent characters do not match those in the dictionary word. Likewise if eighth character through the end of the word matches the corresponding characters in any dictionary word, **LC3** will identify those. When one half of a password is cracked, but the other is not, question marks ('???????') fill the un-cracked half. When neither half is cracked, the results in **LC3** are left blank.

This approach explains partial results **LC3** returns when one part of a password matches a dictionary word and the other does not. For example, consider the following passwords and their results in a Dictionary attack:

Password	Dictionary Attack Result	Comments
biochemistry	biochemistry	Standard word, found in LC3's words-english dictionary, and cracked in full.
biochemist7y	biochem???????	the first 7 characters match those in 'biochemistry.'
b#^chemistry	???????istry	The 8th character, through the end of the password match the corresponding characters in 'biochemistry,' but the first seven do not.
accomplistry	accomplistry	The password is not a dictionary word. Because both the first seven characters and characters 8-12 happen to match dictionary words, LC3's Dictionary crack finds the whole password, even though different dictionary words matched each part.
severecrimp	[LC3's Dictionary crack will not recover this password]	Although the password is formed from two dictionary words, neither the first 7 characters nor the 8-11th characters match words in the dictionary, thus the dictionary crack does not find this password. <u>You must use the brute force crack to recover a password such as this.</u>

Hybrid Crack: The second method **LC3** can use is called a hybrid crack. This builds upon the dictionary method (and its results display in the 'Dictionary Status' area) by appending numeric and symbol characters to dictionary words. Many users choose passwords such as "bogus1!" in an attempt to create a memorable, yet harder to crack password, based on dictionary words slightly modified with additional numbers and symbols. These are the types of passwords that will pass through many password filters and policies yet still pose organizational vulnerability because they are so easily cracked.

LC3 can guess these passwords in much less time than it would take for a brute force attack. **LC3**'s Hybrid crack checks to see if any number of number and symbol characters is appended to each word in the word file you have selected. The default number of number and symbol characters is 2 but can be changed according to your preference.

Note: Selecting 3 'Characters to Vary' makes the Hybrid attack take much longer than with just 2 Characters to Vary. If you use a dictionary that's much bigger than the one **LC3** uses by default, it may take a prohibitively long time to finish.

Brute Force Crack: The most comprehensive cracking method is the brute force method. This method will recover any password up to 14 characters (which is Windows NT's password length limit).

Because the brute force crack tries every combination of characters it's configured to use, your choice of character sets determines how long the brute force crack will take. Common passwords, based on letters and numbers can typically be recovered in about a day using the default character set A-Z and 0-9. Complex passwords, on the other hand, that use characters such as #_}* may take up to a hundred days to crack on the same machine, using a comprehensive character set.

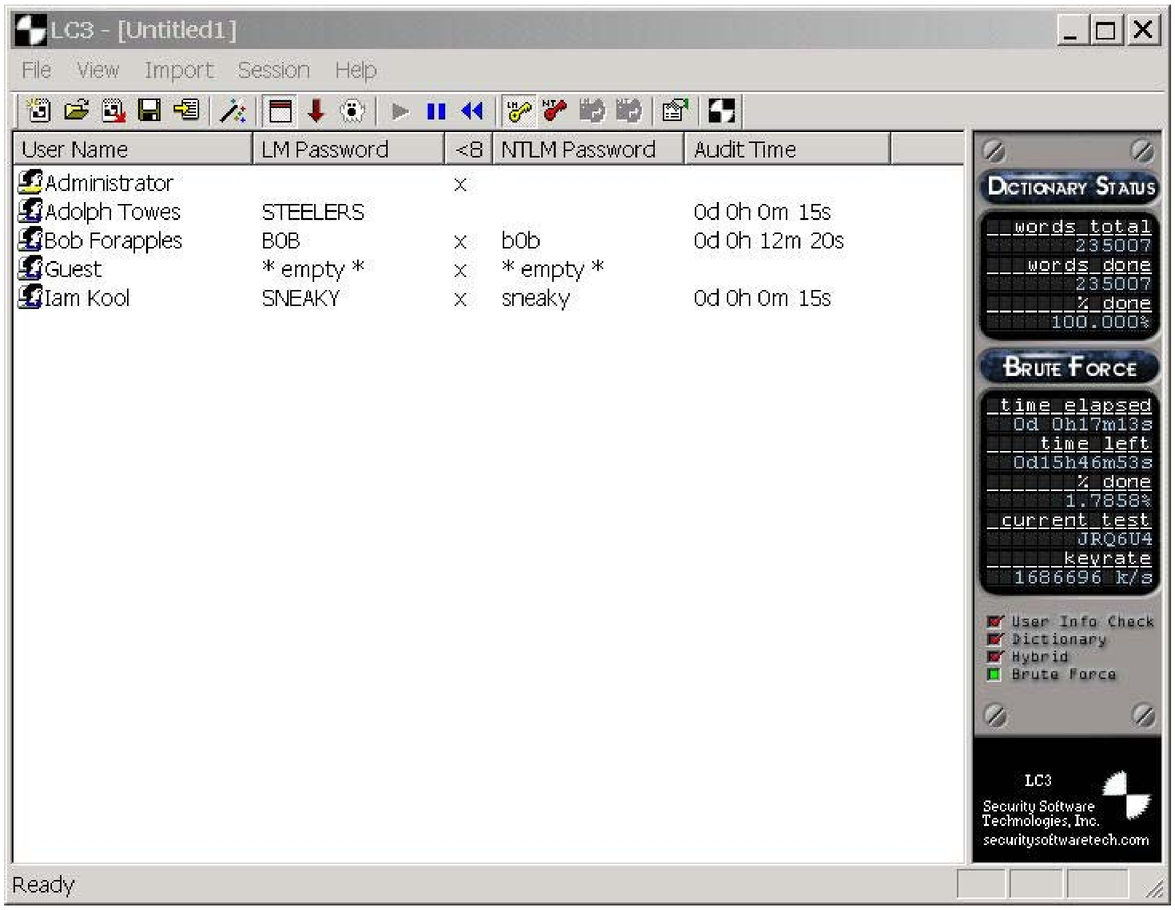
This difference between the strengths of weak versus strong passwords demonstrates the value of strong passwords in protecting your organization or machine. Using a real-world password-auditing tool is the only real way to discover the strength of passwords in your organization, and gauge policy decisions such as:

- the password policies users are expected to follow
- the compliance rate or non-compliance instances with such policies
- the effectiveness of a password filter, or
- what length time one should set for password expirations.

Finally, the graphic below shows LC3 in operation. Notice that **LC3** has found three passwords:

- Adolph Towes – `STEELERS`
- Bob Forapples – `b0b`
- Iam Kool – `sneaky`

Notice also that it has not yet found the administrator's password. Lastly, notice that the time it took to discover those passwords is also displayed



User Name	LM Password	<8	NTLM Password	Audit Time
Administrator		x		
Adolph Towes	STEELERS			0d 0h 0m 15s
Bob Forapples	BOB	x	b0b	0d 0h 12m 20s
Guest	* empty *	x	* empty *	
Iam Kool	SNEAKY	x	sneaky	0d 0h 0m 15s

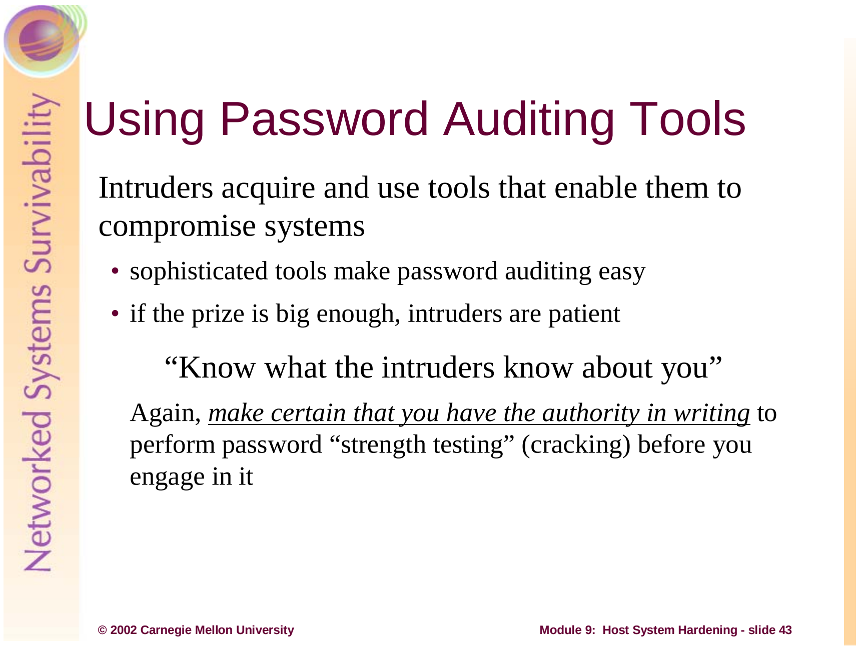
```
words total
     235007
words done
     235007
     % done
   100.000%
```

```
time elapsed
 0d 0h17m13s
   time left
 0d15h46m53s
      % done
    1.7858%
current test
      JRQ6U4
     keyrate
1686696 k/s
```

- 🟥 User Info Check
- 🟥 Dictionary
- 🟥 Hybrid
- 🟩 Brute Force

Ready

The bottom line is that encrypted passwords can be captured from a variety of sources, including from an operational network, and cracked, or at least try to crack them. In some cases, that process gets results quickly whereas in others, it may take days to attempt brute-force all of the guesses.



Using Password Auditing Tools

Intruders acquire and use tools that enable them to compromise systems

- sophisticated tools make password auditing easy
- if the prize is big enough, intruders are patient

“Know what the intruders know about you”

Again, *make certain that you have the authority in writing* to perform password “strength testing” (cracking) before you engage in it

© 2002 Carnegie Mellon University   Module 9: Host System Hardening - slide 43

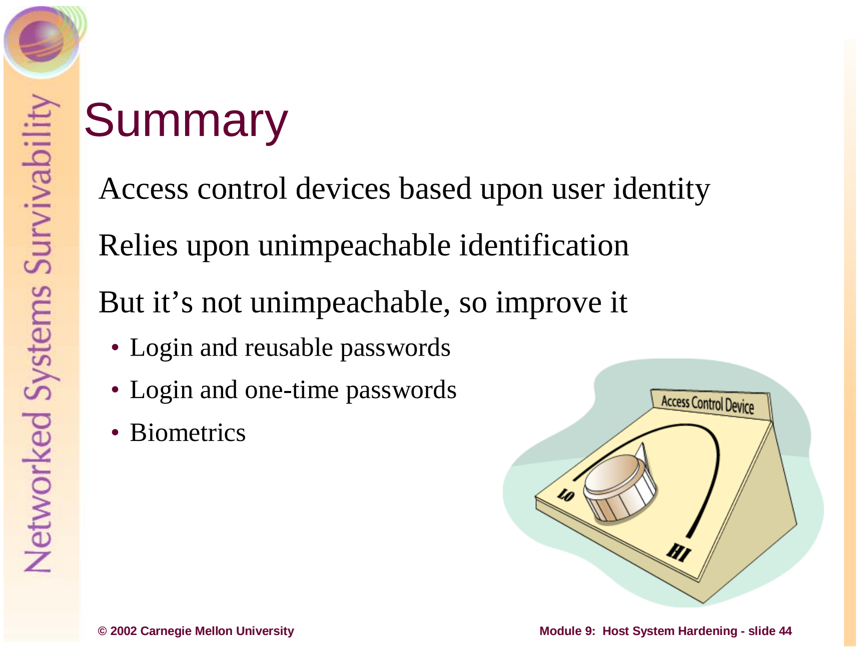
Password auditing tools are available, sophisticated, and extensible. Some are free and those that are commercial are affordable. Auditing passwords is no longer bounded by the intellectual capacity of an intruder – anyone can do it.

Even though brute force methods may take days or even weeks to achieve success, this should not be viewed as a large enough deterrent to a would-be intruder. If the prize to be claimed by gaining access through a password audit is sufficiently big, then the intruder will likely be patient enough to wait until the audit finishes.

The time it takes to discover a password should be longer than the useful lifetime of the information assets it guards. For example, if an intruder takes six months to crack a password at a bank, chances are good that after that period, the bank will still be a bank and the discovered password will still be useful. In contrast, decrypting a session key used for an e-commerce transaction is time-critical. Today’s technology may be insufficient to help the intruder use the information gleaned by cracking the network traffic generated by that transaction.

It is important that a systems administrator know about their systems what the intruders may also know about their systems. This gives rise to the whole concept of running proactive vulnerability testing tools. An organization’s test environment provides an excellent means to conduct carry out this task without harming the production infrastructure.

Finally, learn from history. Read about a well-meaning systems administrator who felt that password auditing was in his purvey but his employer did not [EFF 99]. The lesson to learn is to have the authority in writing to conduct an audit. If you don’t have that authority, don’t do it, tempting, as it may be to improve the security of your site.



Networked Systems Survivability

Summary

Access control devices based upon user identity

Relies upon unimpeachable identification

But it's not unimpeachable, so improve it

- Login and reusable passwords
- Login and one-time passwords
- Biometrics

Access Control Device
LO
HI

© 2002 Carnegie Mellon University

Module 9: Host System Hardening - slide 44

The main problem that systems administrators need to address is discriminating access to information assets as defined by an organization's asset usage policies. As part of the solution to that problem, administrators must adjust the available controls that limit access to those resources. At the basic level, those adjustments are contained within the operating system. Fortunately, most modern operating systems provide almost all of the necessary controls to limit that access. The controls used most often are the access control devices known as access control lists. Setting ACLs in accordance with usage policies is a tedious and sometimes difficult job, but through the techniques described earlier in this module, it can be made more manageable.

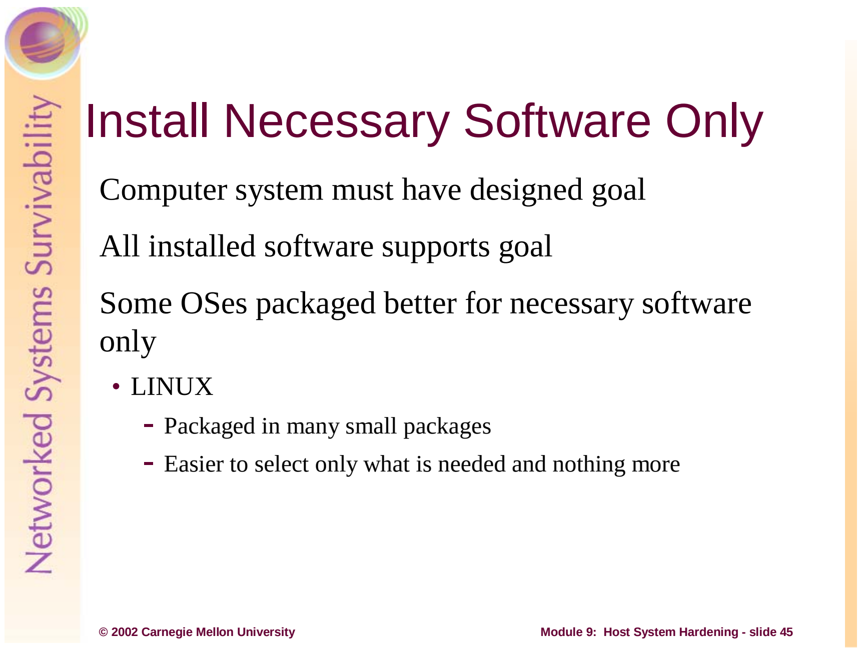
Key to using ACLs is the assumption that people have reliably identified themselves to the computer system. Once identified, these users are granted privileges and rights that enable them to carry out their duties and responsibilities on that computer system.

If it is possible for one person or user to electronically impersonate another person or user, then administrators cannot rely on the access-to-information-assets model. This means that the information assets in question cannot be secured in accordance with an organization's policies.

Therefore, it is profoundly important that the methods and technologies used to discriminate access to information assets be anchored by as strong an identification scheme as the organization can pragmatically and financially tolerate. This section of this module has discussed how traditional identifications schemes (login and reusable passwords) work and their weaknesses. The next concept discussed was one-time passwords as an improvement over reusable passwords. This section ended with a discussion of biometrics.

So far in our house analogy, we've got some information assets in a locked file cabinet in one of the rooms in the house. We know that that house has a front door that is usually locked. Those who have keys have access to the house, but they need a key or the combination to the file cabinet to gain access to the guarded assets. Indeed, those assets may be of such importance that they are encrypted to render them useless to all but those who know the specifics of the decryption operation.

We will continue to build our house and shore up its defenses and at the same time relate those defenses to computer systems.



Install Necessary Software Only

Computer system must have designed goal

All installed software supports goal

Some OSes packaged better for necessary software only

- LINUX
  - Packaged in many small packages
  - Easier to select only what is needed and nothing more

Networked Systems Survivability

© 2002 Carnegie Mellon University

Module 9: Host System Hardening - slide 45

Most houses usually have more than one door, though only one is considered the “front” door. Because of this, homeowners have more than one entrance they need to secure. Certainly this security task would be easier if there were only one door. However, that presents other problems, such as having alternate exits in case of fire. More doors represents a trade-off between a more secure configuration – only one door – and a more useable or practical configuration – multiple doors.

Computer systems have some of these same issues. For example, having all available software and packages on a system may make it easier to diagnose and repair problems and they provide more options when configuring a system. But, the more software there is, the more opportunities for vulnerabilities that can be exploited by an intruder to gain unwanted or undeserved access. So, the key point here is to install only the minimum software necessary for a computer system to achieve is designed goal, and nothing more.

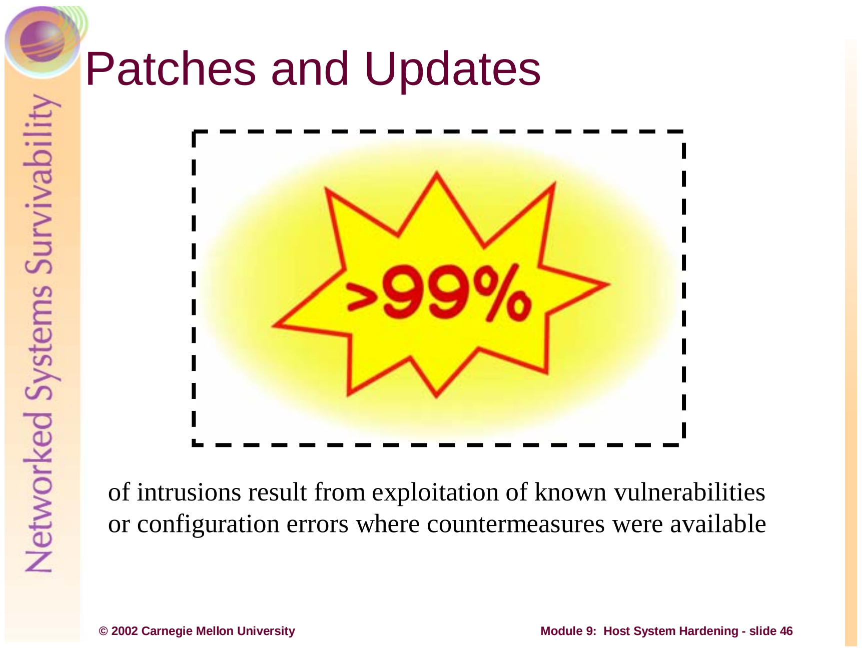
To install only the minimal software in accordance with a design goal presupposes that there is a design goal. Imagine a computer system designed to be a Web server. The collection of practices and procedures to achieve this goal are defined in [CERT 01a]. Those practices define what tasks the Web server should perform and how to configure such a machine to perform them and only them. So, the first step is to know what the system will do.

Once the system has a goal, the next issue to tackle is what is the software necessary to achieve that goal. There’s no easy answer to that question. One way to approach this is to run the system in a production mode and see the list of programs and applications used and then eliminate all else. Another way is to load the absolute minimum and subsequently add what is missing. It is a bit of an art and can be laborious, but remember that the fewer the doors, the fewer things to secure and the more secure the system is.

Because of their packaging schemes, some operating systems make installing only the minimally required software easier to achieve. For example, most LINUX-based systems are packaged as several small (10-30 member) elements (called packages) that provide the administrator the luxury of installing only those tools and applications needed to meet the goals of the system. Continuing this example, on RedHat’s version of LINUX, the **telnet** client and **telnet** server are packaged separately so that they can be installed independently of each other, or not installed at all. The client package provides only the

telnet client and it’s documentation whereas the server package provides only the server and it’s documentation. The administrator can install exactly what is needed and nothing more.

In summary, where possible, select operating systems that provide the tailoring capabilities that allow for only the needed software to be installed. No matter what operating system is selected, use the facilities available to install only the tools and applications that are needed to operate that system.


Networked Systems Survivability

Patches and Updates

>99%

of intrusions result from exploitation of known vulnerabilities or configuration errors where countermeasures were available

© 2002 Carnegie Mellon University   Module 9: Host System Hardening - slide 46

If you learned that an ordinary screwdriver could easily defeat the lock on your front door, what would you do? Likely, you'd replace it with a different lock that could withstand that type of threat, especially if you felt that it was a true concern. Hopefully, you wouldn't wait until someone took advantage of this vulnerability in your house before you responded.

The same principle also applies to a computer system. In fact, ninety-five percent of all network intrusions could be avoided by keeping a computer system up to date by applying patches proactively, that is before an intruder takes advantage of them.

There's no question that keeping patches up to date is often a hard and time-consuming task, especially if patching a system means visiting each computer system. Imagine walking around a site, visiting each and every workstation, installing needed patches after cajoling each user to log off. At 20 minutes per computer system, it would take more than 40 days to patch 1,000 computers. It's no wonder that organizations' systems whose patches must be installed at the workstation often go unpatched. Even if patches aren't installed separately for each user, just keeping current with available patches and finding time to install them is a challenge.

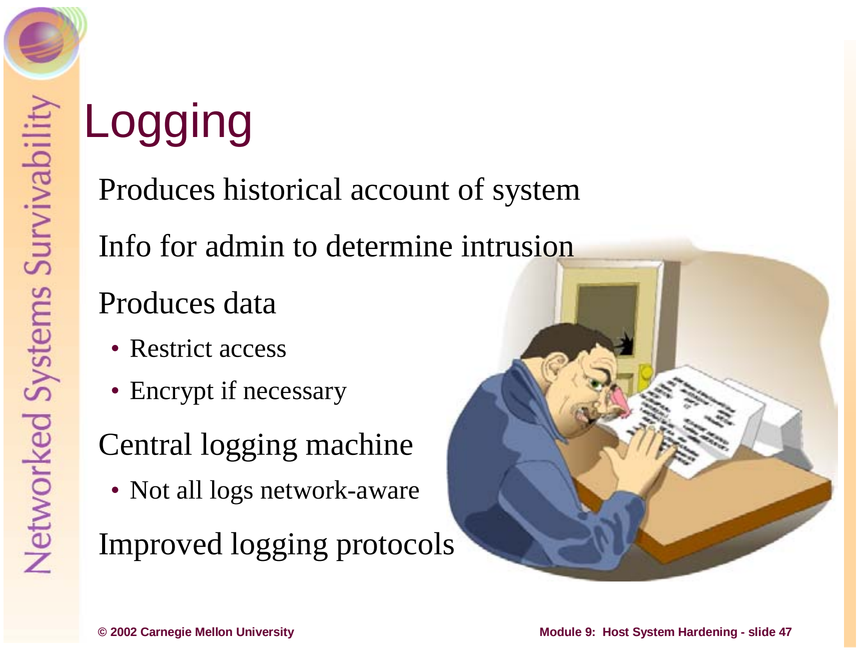
What happens on those systems once they are patched? Do the applications that worked before still work? Can this be known in advance? If there is a problem, can the offending patch be successfully "back out"? Unfortunately, this question is hard to answer because application vendors don't always provide enough information. Sometimes it seems easier to avoid patching systems altogether and just tighten perimeter defenses instead.

Since applications are the lifeblood of business, security is often sacrificed to keep the business running. This may mean that patches are not installed. Ultimately, the business will suffer, though perhaps not in the short term. In the long term, the potential harm that an organization will experience may be greater than the harm caused by installing patches earlier and discovering incompatibilities sooner.

Remember the old oil filter commercial tag line: "You can pay me now or pay me later"? The commercial reminded people that the low cost of an oil filter—less than $10—could save them the cost of a new engine later—often over $1,000. Contrast the cost of installing a patch and perhaps discovering an application incompatibility with the cost of managing and recovering from an intrusion. At least when a new engine is installed in a vehicle, it will be drivable again and the driver will be able to get back to the business at hand. Recovering from an intrusion is not so straightforward. It is possible that an organization won't be able to recover from an intrusion quickly or completely. It may even put them out of business!

Nonetheless, patch wherever and whenever possible. When it's not obvious that a patch can be installed without serious repercussions, contact the vendor and ask. The more administrators who ask these questions, the more likely the vendors will make their products work on patched systems and, perhaps, publicize their efforts - the best of both worlds.

New vulnerabilities are discovered and exploited every day. For the systems, network components, and software used in your organization, it is important to stay abreast of reports regarding vulnerabilities and their solutions. Applicable security patches and workarounds should be applied and deployed as soon as possible.



Networked Systems Survivability

Logging

Produces historical account of system

Info for admin to determine intrusion

Produces data

- Restrict access
- Encrypt if necessary

Central logging machine

- Not all logs network-aware

Improved logging protocols

© 2002 Carnegie Mellon University

Module 9: Host System Hardening - slide 47

Logging produces information assets that describe – to varying degrees of completeness – the history of activities on a computing system, router, etc. Logging does not prevent an intrusion per se, but it does provide information that the administrator must interpret to recognize the signs of an intrusion.

Logging produces information. That information should be secured through all available access control devices – ACLs and encryption – and in all of the states where resides – processing, transmission, and storage. To this list of access control devices, we can add the notion of a dedicated logging machine. This machine, located somewhere on your network, contains all of the logging information from all machines in your organization. It is a highly secured machine, including being physically secured against unwanted access through its console and other hardware communication ports. By consolidating all logs of all computers in one place, the administrator can more easily track the sum total of an intruder's activities throughout their entire network.

This central logging machine should have sufficient disk space to hold all of the log information produced by computer systems throughout your network. These days, disk space is inexpensive and the value of having all of the logs available for perusal likely far outweighs the capital costs of the technology needed to support this activity. For $1,500 or less, you can build a central logging machine using LINUX running on an INTEL-based processor with 40Gb or more of disk space. Costs no longer constrain this task.

Unfortunately, not all applications can forward their logging information to another computer system. For example, on LINUX/UNIX–based systems, process accounting information, which shows the log of processes run and other resource consumption information, is only available in a local file on the system where it is produced. That information would need to be moved or copied onto the network-logging machine to achieve the one-stop-shopping logging goal previously noted.

LINUX/UNIX-based machines come with a network-capable logging mechanism based upon **syslog**(8). **Syslog** is a UDP-based scheme that uses neither strong authentication nor an encrypted data channel to secure logging information assets. There are **syslog** replacements, most notably **syslog-ng**,[16] that address the authentication and encryption issues.

[16] http://www.balabit.hu/en/downloads/syslog-ng.

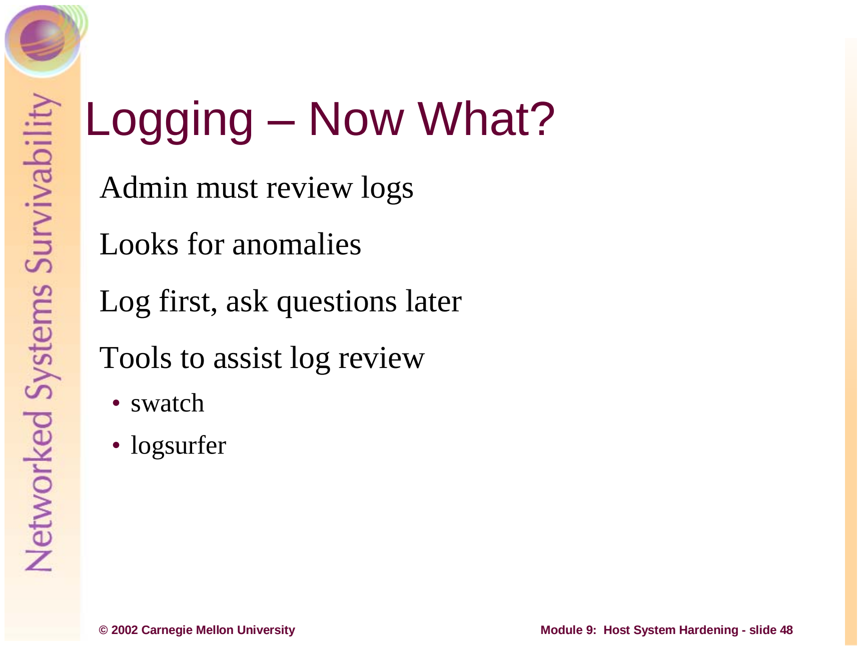
Windows-based machines are not **syslog** compatible, but there are packages that can copy information from the Windows Event Log into **syslog**. One such tool is **slogger** from Core Security Technologies[17].

To make the most use of the logging information sent to a logging server, it is important to have a reliable time-ordered sequence of logging entries. This means that the clocks on all of the machines sending logging information must be synchronized. There are packages for LINUX/UNIX and Windows [Geosoft 02] that use the NTP (Network Time Protocol) [IETF 92]. The key point here is to synchronize the clocks with one another, and better still to one of the many network timeservers [Mills 02].

In summary, the goals of these and any other comparable products are:

- Synchronize the clocks of the log server and all logging clients with NTP.
- Consolidate all logs into a single place – this requires network-aware technologies.
- Strongly authenticate the communication – this requires protocols and software that support strong authentication.
- Encrypt the logged data – this requires that the data be encrypted in all places where it resides, namely transmission, processing, and storage.

[17] http://www.core-sdi.com/download/download1.html



Logging – Now What?

Admin must review logs

Looks for anomalies

Log first, ask questions later

Tools to assist log review

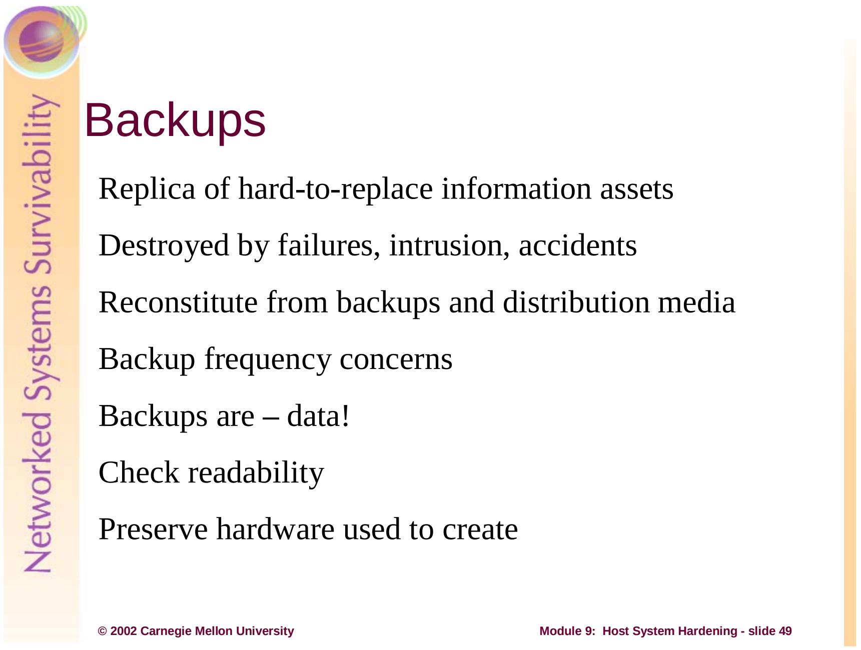
- swatch
- logsurfer

Networked Systems Survivability
© 2002 Carnegie Mellon University
Module 9: Host System Hardening - slide 48

Now that you’ve built this logging infrastructure that consolidates all logs onto a single machine where the connections are strongly authenticated and the data encrypted against theft and unwanted viewing, what happens next? Easy – someone (the administrator) needs to look at the logs often enough that they can detect abnormalities, some of which may be signs of an intrusion.

What is anomalous behavior? It is anything that is unexpected, but if you don’t know what expected is, then it is difficult to define unexpected. This topic will be discussed in much greater detail in Module 13 on Intrusion Detection.

For now, the key is to harden a host computer system by configuring it to log everything that it is capable of logging. That logging information should be securely consolidated in one computer system on your network to facility its review. The review process can be automated through analysis and notification tools such as **logsurfer** [CERT01b] and **swatch** [CERT 00]. The challenge of this activity is to configure these tools to look for specific entries that are indicators of intrusion attempts or that represent abnormal behavior.



Backups

Replica of hard-to-replace information assets

Destroyed by failures, intrusion, accidents

Reconstitute from backups and distribution media

Backup frequency concerns

Backups are – data!

Check readability

Preserve hardware used to create

Networked Systems Survivability
© 2002 Carnegie Mellon University
Module 9: Host System Hardening - slide 49

Most homeowners have homeowners insurance. Its purpose is to allow the homeowner to replace items lost through theft, fire, other natural disasters, or acts of God. In most cases, the property around a house can be replaced by something comparable, that is, if a thief steals your television, your homeowners insurance helps you to buy another television. If the new TV is not identical to the old one, that's probably OK – after all, they are just televisions.

Likely you do have things around the house that cannot be replaced. To address this, you may have stored them in a fireproof container or even in a lockbox at a local bank. Insurance policies, securities, and perhaps special jewelry fit into this category. You may also have insurance on them, but their loss is certainly more devastating. Some things are indeed irreplaceable.

On a computer system, you have many of the same issues. In most cases, the hardware can be replaced with what's available today (and in some cases, that would be a good thing, given the age of some production systems these days). The software and other information assets are a different story. Those are much more difficult to either replace or reconfigure to match your production environment. Because of that, you routinely make copies of those assets, called *backups*.

Information assets are destroyed because of hardware or software failures, an intrusion or any other malicious activity, or just plain old accidents. No matter why the assets were destroyed, the administrator's job is to reconstitute them from a combination of backups and the original operating system and application distribution media. The resulting assets should be comparable to the assets they replace.

Most organizations backup their information assets once a day, usually over night when system usage is lower and the assets are fairly static. This helps to insure that the backups represent the assets in a consistent state. Other organizations backup assets more frequently so that they can capture more changes should an asset be damaged. The backup frequency depends upon the importance of the changes to the assets; the more important they are, the more often backups should be done.

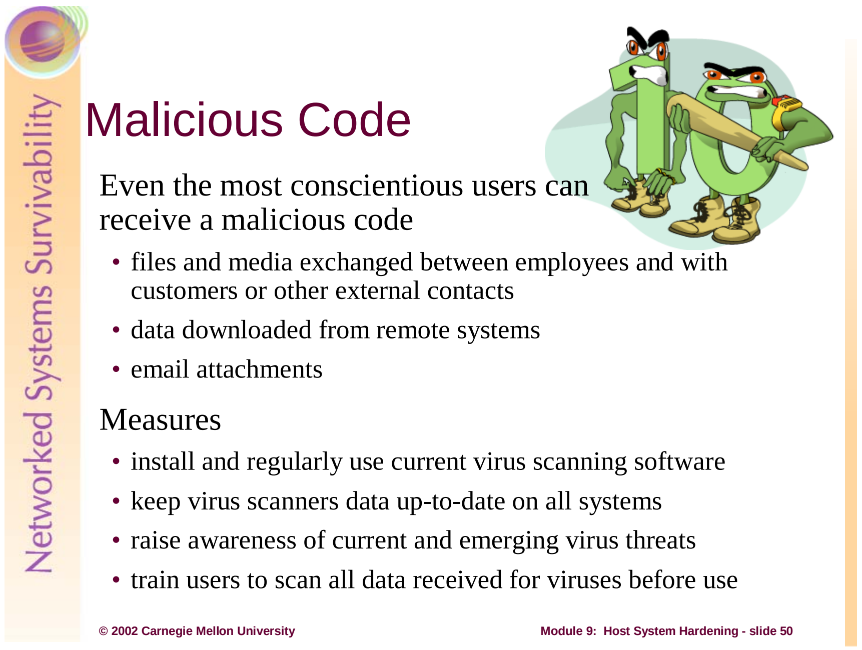
The process of doing backups creates new information assets, and those assets must be secured just like any other asset. This means that the backup media should be securely stored from a physical perspective, using the locked cabinet or separate, locked room access control device. In addition, backups can be encrypted to guard against theft. Two sites that describe systems that encrypt backups are [Flipchip] and [Bird 01]. These are but two examples of technology of which there are many. Encrypting your backups is the key.

You should also be certain that your backups can be read and that you can in fact recreate an information asset should it become damaged. If you have a spare machine, try to rebuild an asset from the backups and distribution media to build confidence in your backup procedures.

Finally, you'll need to keep hardware around and functional as long as you have backups that rely on that hardware. This may mean that you keep that 9-track 1600BPI tape drive around until the assets on those types of tapes are no longer needed. Be sure to factor the maintenance cost of the hardware into your thinking. It may be cost-effective to copy the assets from one medium to another.

The key points for backups are:

- Backup as often as the importance of the assets dictate
- Maintain the physical security backup media
- Encrypt the assets on the backup media
- Be certain you can read the media that you've written
- Maintain all hardware necessary to read all backup media


Networked Systems Survivability

Malicious Code

Even the most conscientious users can receive a malicious code

- files and media exchanged between employees and with customers or other external contacts
- data downloaded from remote systems
- email attachments

Measures

- install and regularly use current virus scanning software
- keep virus scanners data up-to-date on all systems
- raise awareness of current and emerging virus threats
- train users to scan all data received for viruses before use

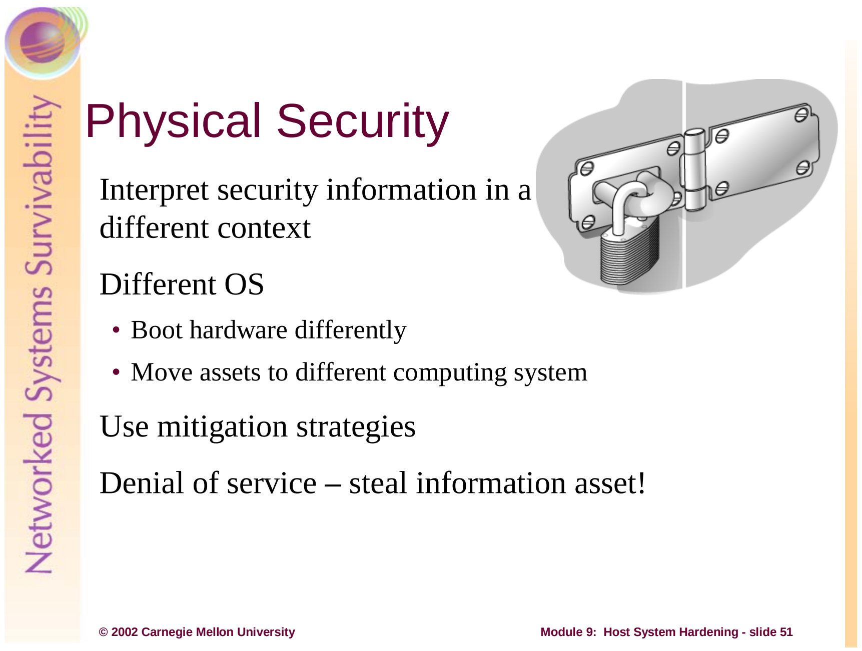
© 2002 Carnegie Mellon University Module 9: Host System Hardening - slide 50

Any time data is introduced into a computer system, malicious code (e.g. viruses) can also be introduced. The pathway used is much less a concern than is the nature of the code. In short, if the bits can get to a computer, they need to be inspected for malicious content.

Most often, malicious code is targeted against Windows-based systems simply because of the damage potential. No other system is so widely deployed that infecting it can have an effect of the same magnitude. This doesn't mean that you shouldn't consider malicious code on other systems; rather it means that you should concentrate your efforts on Windows-based systems.

Combating malicious code is addressed through technology, policy, and procedures. The technological state-of-the-practice is to install malicious code scanners and configure them to keep their signature files up-to-date. These signature files contain the information needed by a scanner to recognize malicious code should it find its way to your computer systems. Since new instances of malicious code are discovered daily, keeping up with the latest signatures is the best way to secure your systems.

The policy and procedures part of the state-of-the-practice is to inform your users about malicious code, signatures, and the ways that such code can enter your organization's computer systems.



Networked Systems Survivability

Physical Security

Interpret security information in a different context

Different OS

- Boot hardware differently
- Move assets to different computing system

Use mitigation strategies

Denial of service – steal information asset!

© 2002 Carnegie Mellon University   Module 9: Host System Hardening - slide 51

You've taken the time to secure your information assets in your house by locking them in a cabinet and locking the room where they are. You have family policies and procedures in place, and you've made an encrypted copy of those assets and placed them in a safety deposit box at the local bank. You have installed a good front door lock, and you have more family policies and procedures that define who can enter the house and under what circumstances.

Imagine then that something catastrophic happens, such as an intruder using their car to break down a wall, or if your home is a mobile home, the intruder attaching it to a truck and carting it away. The physical security beyond the doors and windows is also a concern. You need to at least think about those issues and mitigate where possible.

Within the domain of a computer system, physical security must also be a concern to the administrator. For example, if you've followed the advice given in this module about ACLs but an intruder can boot a different operating system on your computer – one that disregards the ACLs – then they are useless. ACLs are most relevant when they are interpreted by the operating system that they are designed for. More specifically, if your system is Windows-based and you set the ACLs for that system, and then LINUX is run on that same hardware, those ACLs have no meaning.

The example above can be generalized to the following: if the fundamental computing paradigm can be changed, then controls used by the system administrator to secure information assets can be rendered meaningless. Earlier we discussed identification and authentication as a principle important to ACLs – if people are improperly identified as users, then the ACLs that discriminate asset access have no meaning. In this module, we consider the physical security of a computing system as a way that an intruder changes this fundamental computing paradigm.

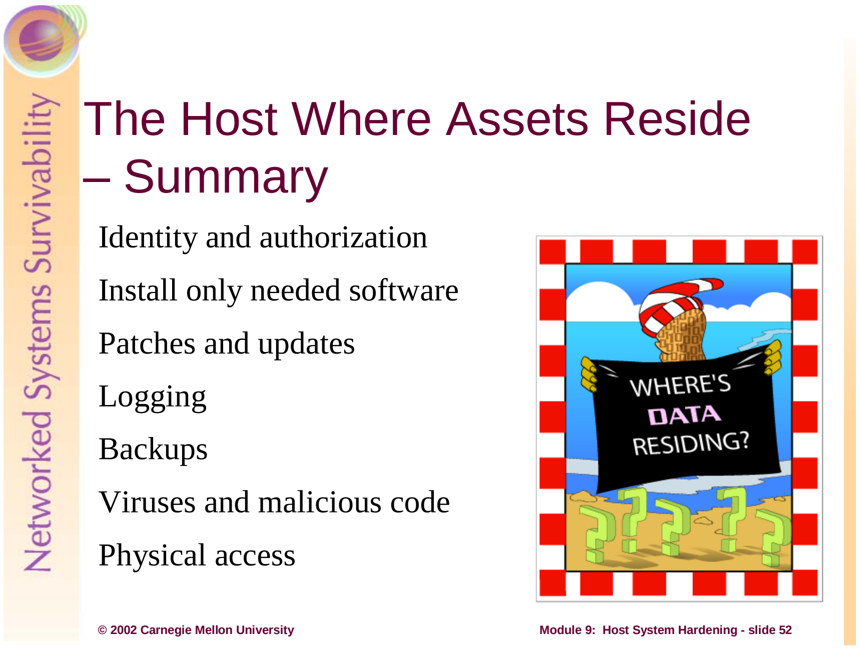
If an intruder seeks to change this paradigm from a physical security perspective, the administrator needs to consider the following:

- Interpreting storage information in a different context. This means disk contents are read and acted upon by an operating system that assigns a different meaning to those contents. This can happen in several ways:
  - Booting a different operating system on the same hardware: This can happen when the BIOS [BIOSCentral] or EEPROM is improperly secured such that the hardware loads and executes a different operating system. Mitigation strategies include:

        - Passwords – Assigning a BIOS or EEPROM password means that the system cannot be booted differently without that password. Note that some operating systems provide a way to set these passwords from software. Note also that physical access to the motherboard of the computer system may provide a way to clear the BIOS or EEPROM password, therefore rendering those systems vulnerable. You need to know how your system works and where to concentrate your efforts.
        - Changing the boot device order – If you can change the boot order, then change it to reflect from where the operating system should boot. The boot order usually requires the BIOS or EEPROM password.
        - Locking hardware to prevent tampering – Some motherboards contain circuitry that allows the CMOS RAM (which contains BIOS/EEPROM configuration information such as boot order and passwords) to be cleared. An intruder can gain access to that motherboard if the computer's cabinet can be opened. Again, check the specifics of your systems to see what is possible and therefore what mitigation strategies to apply.
        - Removing unneeded devices so that they cannot participate in the boot process – This is a more drastic step that may make routine maintenance on your systems more difficult. One of the challenges of systems administration is balancing the security of a system against its usability. By removing the floppy disk and CDROM from a system, you have eliminated those boot options and have improved the security of the system.

          At the same time, you have also eliminated them as fallback boot methods that can be useful in the event of a corrupted hard disk, for example. Only the administrator can select the appropriate choice – to remove or not to remove – given the requirements of the system in its production setting.
        - Locating computer systems in a secure facility – By placing a computer system in a secure facility, some physical security concerns can be addressed. Much as your house could be in a private community with gates and fencing designed to keep out traffic, your computer systems could be placed behind layers of locked doors.
    - Removing the hard drive: If the hard drive can be removed from the computer system and installed elsewhere, the paradigm can again be changed. This happens most often with a portable computing system such as a laptop computer. The mitigation strategies of locking the hardware to prevent tampering and locating computing systems in a secure facility also apply here.
- Denying service. This means that the computer system where the information assets are stored is rendered useless through damage or theft. If an intruder's intent is to deprive users of the benefits of accessing an information asset, then destroying it or stealing it achieves that goal. Even if the computer system is locked against tampering with the hardware, it may be possible to steal it or destroy it. Care should be taken so that that cannot happen either.

In summary, concerns about physical access revolve around changing the paradigm by which information stored on the hard disk is changed by processing that information through a different operating system. The hardware is either made to run another operating system or the hard disk is relocated to another computer system where it is processed differently. There are various mitigation strategies available to the administrator. Finally, the computer system itself can be stolen or destroyed, rendering the information unavailable for its intended audience.



Networked Systems Survivability

The Host Where Assets Reside – Summary

- Identity and authorization
- Install only needed software
- Patches and updates
- Logging
- Backups
- Viruses and malicious code
- Physical access

© 2002 Carnegie Mellon University

Module 9: Host System Hardening - slide 52

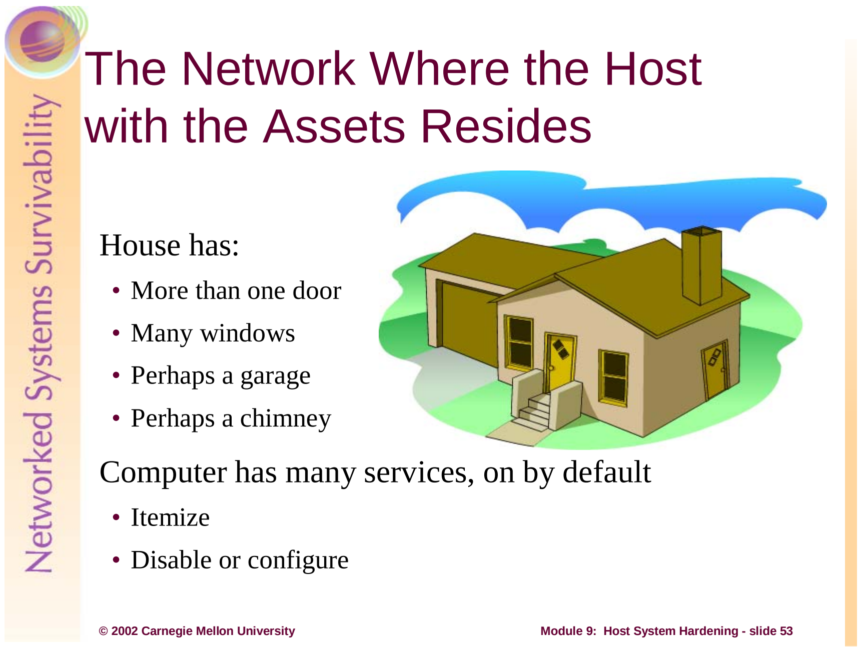
Assets reside on a computer system, and that system needs to be secured beyond the access control devices applied to the assets themselves. The areas that the administrator must concentrate on are:

- Identity and Authentication: Administrators must strengthen the scheme used by people to identify themselves to computer systems. On identified, these users are given rights and privileges to carry out activities on those systems. As long as the identification scheme is unaltered, the access control devices can do its job as designed and configured.
- Install only the needed software: A necessary prerequisite is a clear definition of the role a computer system is to play in an organization. Once that has been established, the administrator installs only the software needed by that system to realize that role, and nothing more. Not every operating system can be easily slimmed down to only the minimum required software. If possible, select one that provides that freedom.
- Patches and Updates: Once the system has been installed, slimmed down, and otherwise configured, the administrator should, wherever and whenever possible, keep that system up to date with patches and updates. The challenge of patches and updates is to determine their effect on the mission critical applications and then to decide whether to proceed or hold back. Having a test environment and the ability to back out patches can be a significant help when considering how to proceed.
- Logging: Logging helps to understand what is happening or has happened. Administrators must look at logs as often as they want to know what their systems are doing. By consolidating logs onto a single machine, the administrator benefits from a one-stop-shopping approach to learning what is happening on their systems. If this is your plan, then you must include clock synchronization so that the log of events can be properly ordered in time.
- Backups: Backups are an essential part of a recovery strategy where intruders, faulty software, or accidents damage assets. Backups are also information assets, and as such, all available access control devices, including encryption to guard against unauthorized disclosure, should be used to secure them.
- Virus and malicious code: These programs are often written with hostile intent. They are spread most often by email, though Web pages and other downloaded information are also popular

methods. Mitigation strategies include virus checkers and vigilance when retrieving executable content.

- Physical access: By changing the computing paradigm, intruders can circumvent the security features provided by an operating system. They are able to achieve this and more when they are able to get physical access to a computer system. In addition to several computer-based mitigation tactics, the old-fashioned, well-established strategies used to secure physical assets also apply.

Now that the security of the host where the assets resides has been addressed, the next area of concern is the network where the host where the assets reside.



The Network Where the Host with the Assets Resides

Networked Systems Survivability

House has:

- More than one door
- Many windows
- Perhaps a garage
- Perhaps a chimney

Computer has many services, on by default

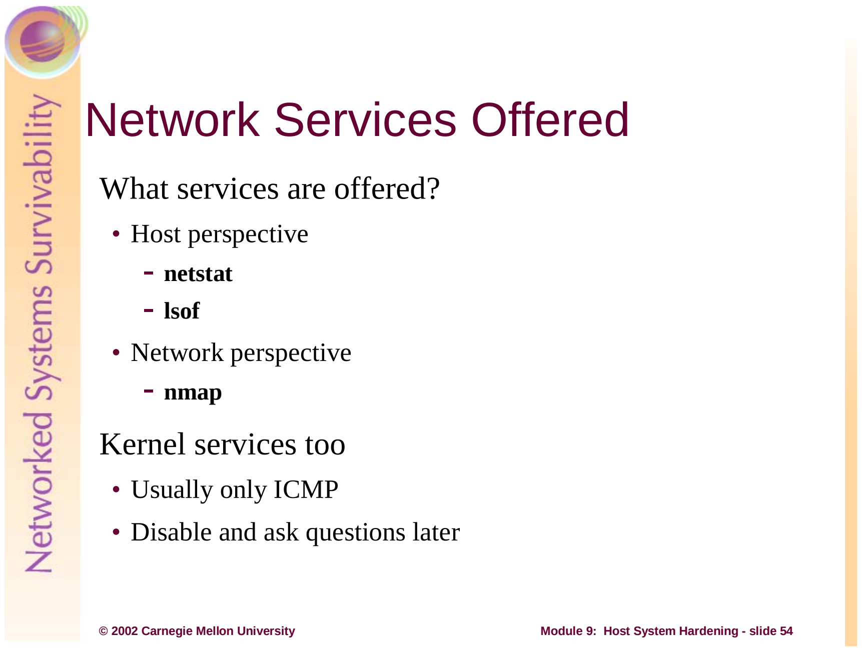
- Itemize
- Disable or configure

© 2002 Carnegie Mellon University

Module 9: Host System Hardening - slide 53

Returning to our house analogy, that house has windows, more doors, a garage, and perhaps a chimney. Each represents a way for a burglar to gain access to the sensitive information we've locked in our file cabinet. We need to recognize that entranceways exist and then think about them in the same way that we think about the front door. That means that we must be sure that these other "entrances" are appropriately secured, security reviewed regularly, and there are family policies and procedures that define who can use any of these alternative entrances.

A computer system attached to a network may also have more doors, windows and other points of access beside the front door of the standard terminal login and password. Each of these entrances – called *services* in networking parlance – needs to be identified and appropriately secured.



Network Services Offered

What services are offered?

- Host perspective
  - **netstat**
  - **lsof**
- Network perspective
  - **nmap**

Kernel services too

- Usually only ICMP
- Disable and ask questions later

© 2002 Carnegie Mellon University   Module 9: Host System Hardening - slide 54

For a house, the complete inventory of doors, windows, garage doors, chimneys, etc is easy to create. For a computer system, it's not nearly as easy. How then does the administrator learn what services are offered by a computer system? There are two ways to answer this question: one is from the perspective of that system and the other is from the perspective of another system on the same network.

From the perspective of that system, the two programs an administrator uses are **netstat** and **lsof** [CERT 00a]. First **netstat**. With the `–a` option, under Windows or LINUX/UNIX variants, **netstat** shows all established connections and offered services. Here are some examples. Note that the output is filtered so that only the listening connections for both the TCP and UDP protocols are shown:

```
% netstat -a| fgrep '*'
tcp        0      0 *:printer              *:*                     LISTEN
tcp        0      0 host.net:domain        *:*                     LISTEN
tcp        0      0 localhost:domain       *:*                     LISTEN
tcp        0      0 *:ssh                  *:*                     LISTEN
tcp        0      0 *:x11:10               *:*                     LISTEN
tcp        0      0 *:x11:11               *:*                     LISTEN
udp        0      0 *:filenet-tms          *:*
udp     3456      0 *:filenet-rpc          *:*
udp     1728      0 *:32923                *:*
udp     1728      0 *:32924                *:*
udp        0      0 host.net:domain        *:*
udp        0      0 localhost:domain       *:*
udp        0      0 *:bootps               *:*
raw        0      0 *:icmp                 *:*                     7
raw        0      0 *:icmp                 *:*
```

The next program to use is **lsof**. **lsof** is a LINUX/UNIX-based program that, when run with the `–i` option, shows all of the network services offered by the local computer system. Here again is an example:

```
% lsof -i
COMMAND     PID USER   FD   TYPE  DEVICE SIZE NODE NAME
named      1100 root   20u  IPv4    1704       UDP localhost:domain
named      1100 root   21u  IPv4    1705       TCP localhost:domain (LISTEN)
named      1100 root   22u  IPv4    1706       UDP host.net:domain
named      1100 root   23u  IPv4    1707       TCP host.net:domain (LISTEN)
sshd       1111 root    3u  IPv4    1724       TCP *:ssh (LISTEN)
lpd        1201 root    5u  IPv4    1903       TCP *:printer (LISTEN)
dhcpd      1212 root    8u  IPv4    1935       UDP *:bootps
login      1425 root    3u  IPv4    4203       UDP *:filenet-rpc
sshd       3441 root    6u  IPv4    4818       TCP *:x11:10 (LISTEN)
sshd      16225 root    3u  IPv4 2378065       TCP *:x11:11 (LISTEN)
```

With these programs, the administrator can see what services are offered from the host computer system's perspective.

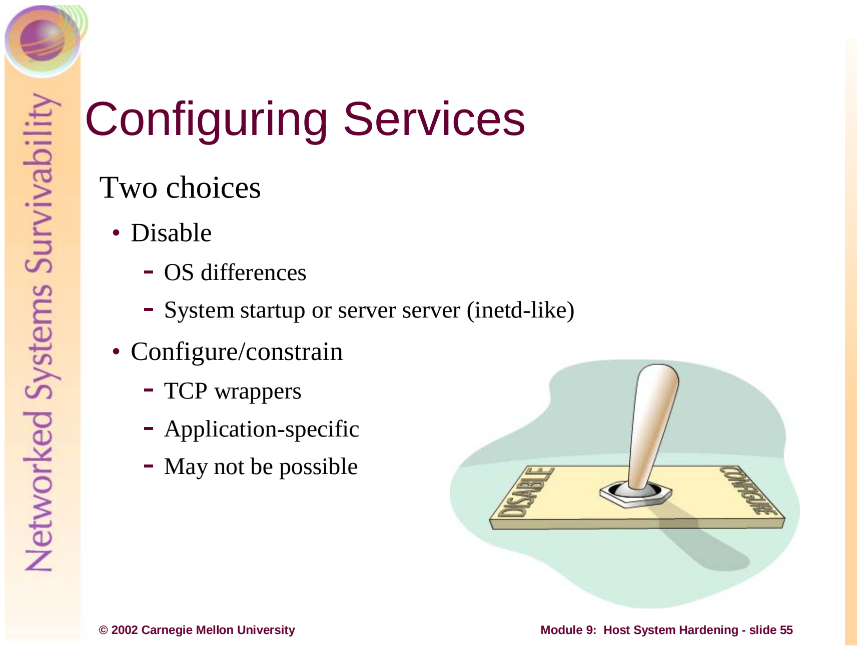
From the network perspective, the program of choice is **nmap** [Insecure]. **Nmap** runs on most LINUX/UNIX systems and support for Windows-based systems is being developed. By using the loopback network connection, available by the host name `localhost`, **nmap** can also be used to learn the services offered on the local computer system. Here is an example:

```
% nmap localhost
Starting nmap V. 2.54BETA7 ( www.insecure.org/nmap/ )
Interesting ports on localhost (127.0.0.1):
(The 1531 ports scanned but not shown below are in state: closed)
Port       State       Service
22/tcp     open        ssh
53/tcp     open        domain
515/tcp    open        printer

Nmap run completed -- 1 IP address (1 host up) scanned in 1 second
```

Nmap shows the port in use, the protocol, the service state, and the services offered.

These programs show the services offered by a computer system. These services typically run as user-level processes outside of the operating system. The operating system also offers a set of services to systems attached to the network, and these too must be secured. Among those services are ICMP-based services [IETF 81], such as the current network's netmask and echo and its response (used by the **ping**(8) program). It is not as easy to determine the list of operating system-based services. Since most of them are ICMP-based, the administrator's best response is to disable them where possible.



Configuring Services

Two choices

- Disable
  - OS differences
  - System startup or server server (inetd-like)
- Configure/constrain
  - TCP wrappers
  - Application-specific
  - May not be possible

Networked Systems Survivability
DISABLE
CONFIGURE

© 2002 Carnegie Mellon University

Module 9: Host System Hardening - slide 55

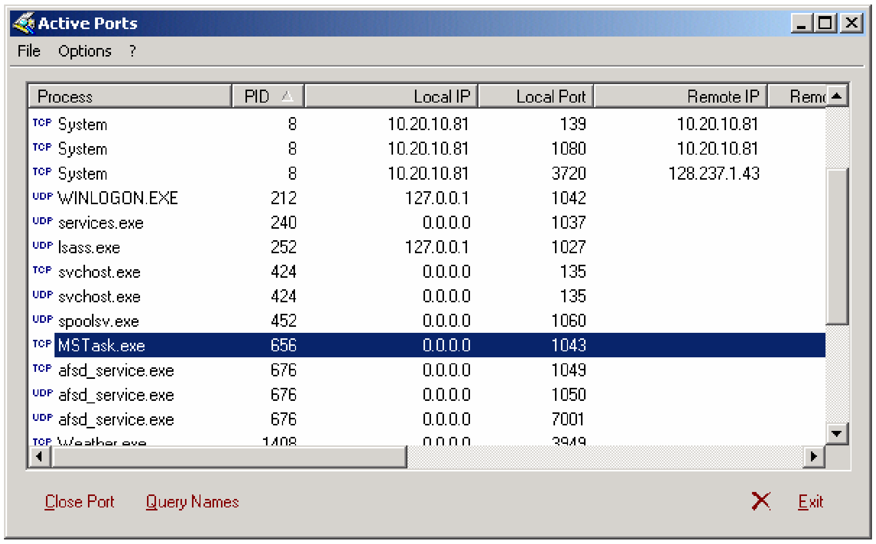
Securing entrances in a house is at least straightforward. Doors, windows, and the garage can be locked and the chimney can be capped so that only small items can pass.

On a computer system, network services must be either disabled or enabled but constrained in their use. This section describes how to do those tasks.

To disable a service, the administrator needs to do the following operating system specific tasks. No matter which OS is used, all of the programs that comprise the service as well as any ancillary configuration information should be completely removed from the system to guard against them accidentally being turned back on.

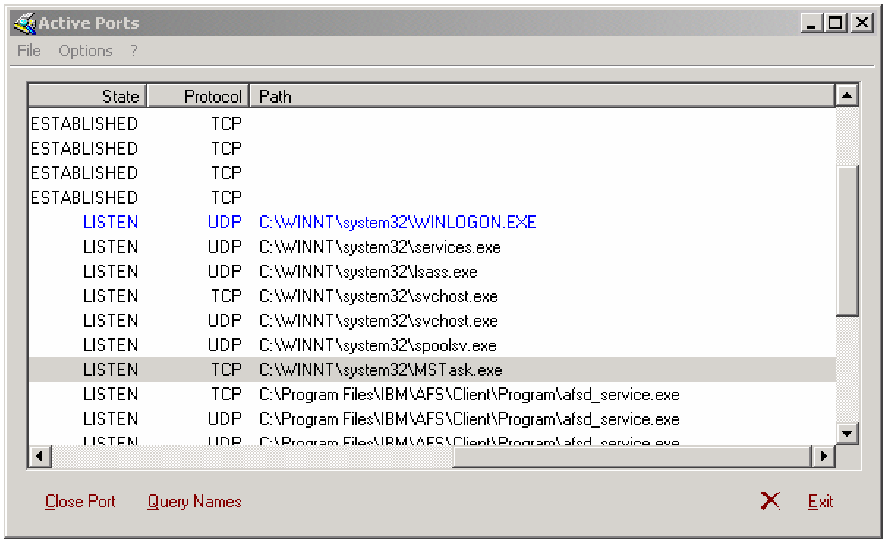
- UNIX/LINUX systems: Determine the program that is providing that service. In most cases this is the program named in the `COMMAND` name column of the output from the **lsof** program. Determine the way that that program is being started, and there are two basic choices:
  1. The program in question is being started from the Internet Services Daemon, **inetd**, or the Extended Internet Services Daemon, **xinetd** [XINETD]. Read the documentation to learn how to disable a service started by this daemon.
  2. The program in question is being started directly by the system. On some variants (LINUX), use the **chkconfig** program to disable the service. On other variants (SOLARIS), the files in the */etc/rc[1234546].d* directories must be removed so that that program in not started at the specified run level, in this case run levels 1 through 6. Some systems, Mandrake 8.1, has a tool named SysV-Init Editor which is a GUI interface to configuring services and various levels.
- Windows-based systems: Determine the program that is providing the service. One way to do this is to use the freeware tool Active Ports[18] [SmartLine]. Active Ports shows the list of ports in use by processes on your Windows systems, including those where a server is listening for both TCP- and UDP-based services. Active Ports also lists the full path name of the program providing that service. The two following figures show its output:

[18] http://www.bhs.com/downloads/download.asp?filename=aports%2Ezip



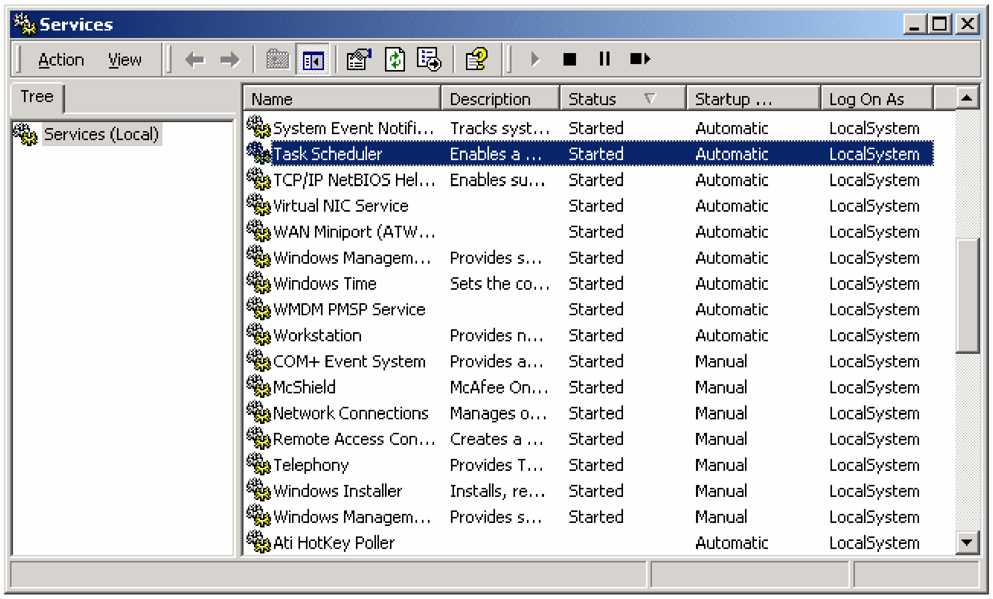
Process	PID	Local IP	Local Port	Remote IP	Rem
TCP System	8	10.20.10.81	139	10.20.10.81	
TCP System	8	10.20.10.81	1080	10.20.10.81	
TCP System	8	10.20.10.81	3720	128.237.1.43	
UDP WINLOGON.EXE	212	127.0.0.1	1042		
UDP services.exe	240	0.0.0.0	1037		
UDP lsass.exe	252	127.0.0.1	1027		
TCP svchost.exe	424	0.0.0.0	135		
UDP svchost.exe	424	0.0.0.0	135		
UDP spoolsv.exe	452	0.0.0.0	1060		
TCP MSTask.exe	656	0.0.0.0	1043		
TCP afsd_service.exe	676	0.0.0.0	1049		
UDP afsd_service.exe	676	0.0.0.0	1050		
UDP afsd_service.exe	676	0.0.0.0	7001		
TCP Weather.exe	1408	0.0.0.0	3949		

This part of the display shows that the **MSTask.exe** program is listening for connections on TCP port 1043 and that those connections can come from anywhere (0.0.0.0).



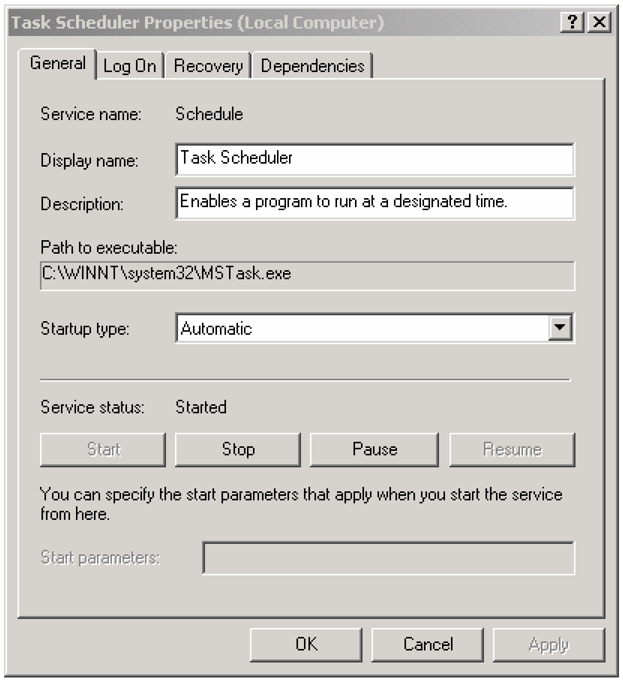
State	Protocol	Path
ESTABLISHED	TCP	
ESTABLISHED	TCP	
ESTABLISHED	TCP	
ESTABLISHED	TCP	
LISTEN	UDP	C:\WINNT\system32\WINLOGON.EXE
LISTEN	UDP	C:\WINNT\system32\services.exe
LISTEN	UDP	C:\WINNT\system32\lsass.exe
LISTEN	TCP	C:\WINNT\system32\svchost.exe
LISTEN	UDP	C:\WINNT\system32\svchost.exe
LISTEN	UDP	C:\WINNT\system32\spoolsv.exe
LISTEN	TCP	C:\WINNT\system32\MSTask.exe
LISTEN	TCP	C:\Program Files\IBM\AFS\Client\Program\afsd_service.exe
LISTEN	UDP	C:\Program Files\IBM\AFS\Client\Program\afsd_service.exe
LISTEN	UDP	C:\Program Files\IBM\AFS\Client\Program\afsd_service.exe

This part of the display shows that the **MSTask.exe** executable has the full pathname of *C:\WINNT\system32\MSTask.exe*.The next step is to determine how **MSTask.exe** is started by the system, and to do that, select *Start→Programs→Administrative Tools→Services*. Sort on the Status column to see all services that are started. The figure below shows the results.



Name	Description	Status	Startup ...	Log On As
System Event Notifi...	Tracks syst...	Started	Automatic	LocalSystem
Task Scheduler	Enables a ...	Started	Automatic	LocalSystem
TCP/IP NetBIOS Hel...	Enables su...	Started	Automatic	LocalSystem
Virtual NIC Service		Started	Automatic	LocalSystem
WAN Miniport (ATW...		Started	Automatic	LocalSystem
Windows Managem...	Provides s...	Started	Automatic	LocalSystem
Windows Time	Sets the co...	Started	Automatic	LocalSystem
WMDM PMSP Service		Started	Automatic	LocalSystem
Workstation	Provides n...	Started	Automatic	LocalSystem
COM+ Event System	Provides a...	Started	Manual	LocalSystem
McShield	McAfee On...	Started	Manual	LocalSystem
Network Connections	Manages o...	Started	Manual	LocalSystem
Remote Access Con...	Creates a ...	Started	Manual	LocalSystem
Telephony	Provides T...	Started	Manual	LocalSystem
Windows Installer	Installs, re...	Started	Manual	LocalSystem
Windows Managem...	Provides s...	Started	Manual	LocalSystem
Ati HotKey Poller			Automatic	LocalSystem

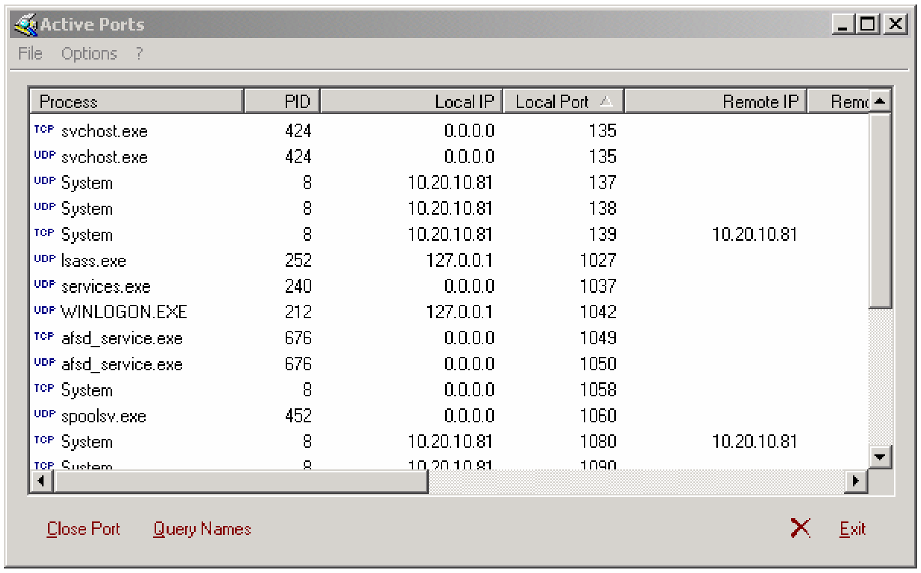
From this list and the fact that the program name has the word *task* in it, we guess that the line in question is the one highlighted. By right-clicking on that line and selecting **Properties**, we get the following:



Service name: Schedule
Display name: Task Scheduler
Description: Enables a program to run at a designated time.
Path to executable:
C:\WINNT\system32\MSTask.exe
Startup type: Automatic
Service status: Started
You can specify the start parameters that apply when you start the service from here.
Start parameters:

Notice that the path to the executable matches what we saw from the Active Ports display.

To stop this service, click on *Stop* button. To ensure that this service does not restart, change the *Startup type* to either *Manual* or *Disabled*. Returning to Active Ports, we see the following that shows that the service has been stopped and the port no longer active. Note that this display is sorted based upon *Local Port*.



	Process	PID	Local IP	Local Port	Remote IP	Remo
TCP	svchost.exe	424	0.0.0.0	135		
UDP	svchost.exe	424	0.0.0.0	135		
UDP	System	8	10.20.10.81	137		
UDP	System	8	10.20.10.81	138		
TCP	System	8	10.20.10.81	139	10.20.10.81	
UDP	lsass.exe	252	127.0.0.1	1027		
UDP	services.exe	240	0.0.0.0	1037		
UDP	WINLOGON.EXE	212	127.0.0.1	1042		
TCP	afsd_service.exe	676	0.0.0.0	1049		
UDP	afsd_service.exe	676	0.0.0.0	1050		
TCP	System	8	0.0.0.0	1058		
UDP	spoolsv.exe	452	0.0.0.0	1060		
TCP	System	8	10.20.10.81	1080	10.20.10.81	
TCP	System	8	10.20.10.81	1090		

To constrain a service, the administrator needs to do the following operating system specific tasks:

- UNIX/LINUX systems: Again, determine the program that is providing that service. In most cases this is the program named in the `COMMAND` name column of the output from the **lsof** program. Again, determine the way that that program is being started, and there are two basic choices:

  1. The program in question is being started from the Internet Services Daemon, **inetd**, or the Extended Internet Services Daemon, **xinetd**.

     a. If **inetd** starts the program, then the TCP Wrappers [SANS 01c] should be used to define the set of computer systems that have access to that service. Examples of how to do this for various systems are in [CERT 00b].

     b. If **xinetd** starts the program, then it has its own built-in service access language. Consult its documentation to see how to configure it. In some cases, the **xinetd** program may be compiled with support for the TCP Wrappers. If this is the case, use its configuration documentation [SANS 01c].

  2. The program in question is being started directly by the system. If this is the case, then the way to constrain the set of hosts that connect to that service varies from application to application.

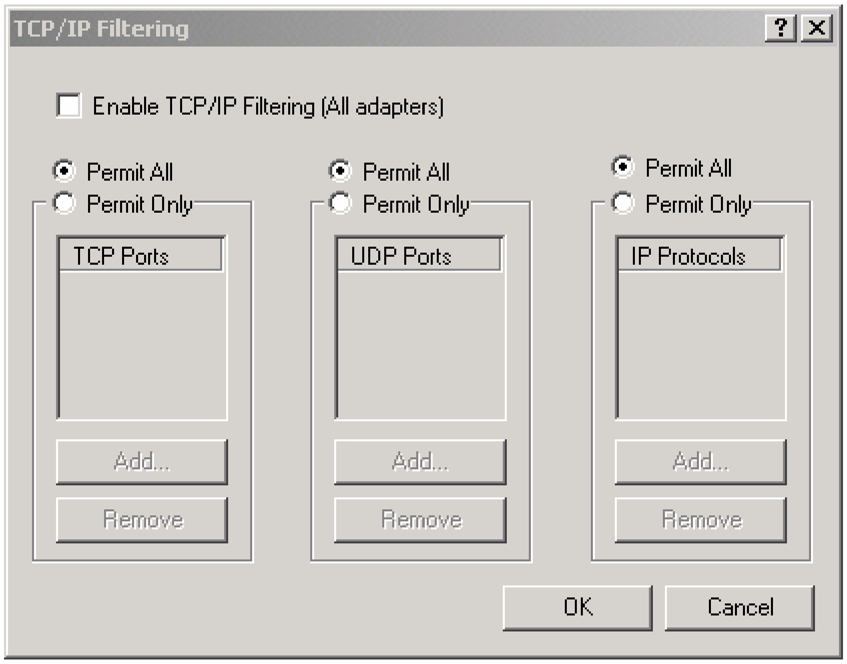
     For example, if the application is the **sendmail**[19] mail transportation agent, it may have been built with the TCP Wrappers. If this is the case, then they can be used to limit connections. Similarly, if the Apache Web Server[20] is the application being secured, then it has its own language for specifying acceptable client access. In other cases, the application in question may have no capability for discriminating access.

- Windows-based systems: Windows systems do not have TCP Wrapper-like functionality, meaning that is no functionality where both endpoints of a network connection can be constrained as with the TCP Wrappers. Where possible, the administrator must use the capabilities of each application to restrict access to that application from the network.

  There is a capability built into Windows whereby connections to specific ports can be allowed or denied. It is available from *Start→Settings→Network and Dial-up Connections*. Next, click on

[19] http://www.sendmail.org
[20] http://www.apache.org

Local Area Network, right click on *Properties*, then *Internet Protocol (TCP/IP)* and then *Properties* again, and then click the *Advanced* button. Under the *Options* tab, select *TCP/IP filtering*, and then click the *Properties* button. You should then see a window similar to the following:



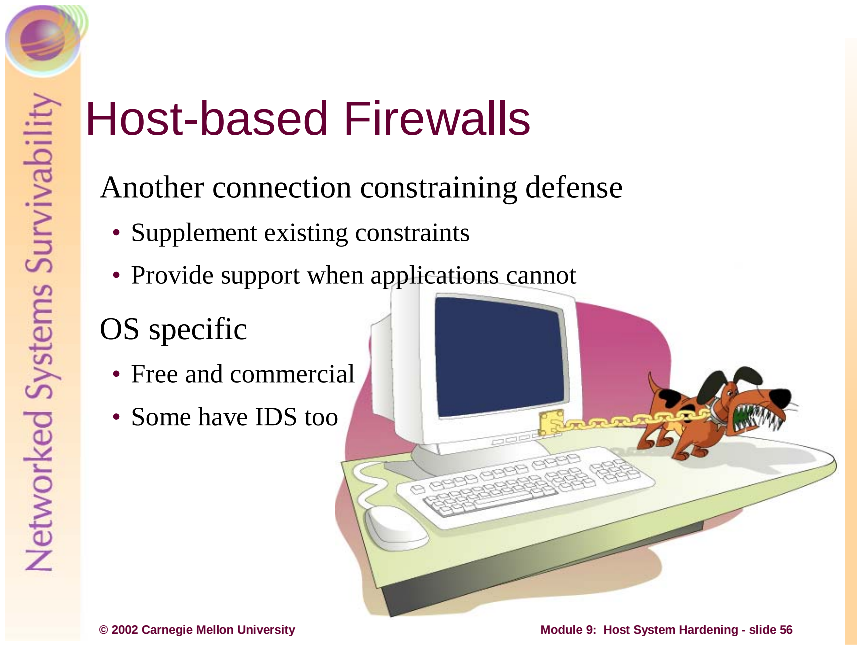
TCP/IP Filtering

☐ Enable TCP/IP Filtering (All adapters)

TCP Ports	UDP Ports	IP Protocols
● Permit All	● Permit All	● Permit All
○ Permit Only	○ Permit Only	○ Permit Only

You would click the various radio buttons and select the ports allowed. Unfortunately, you cannot specify the set of computer systems that can make connections.

This section has described disabling and constraining network access to network services that run outside of the operating system. Once the set of services provided has been established, the goals of the computer system define which of those services remain enabled and to which other computer systems they are available. Note that while almost all services that need to be disabled can be disabled, not all services can be constrained as to the set other computer systems that can connect. The next section discuses host-based firewall systems that can be used to overcome any deficiencies in constraining allowable connections.



Networked Systems Survivability

Host-based Firewalls

Another connection constraining defense

- Supplement existing constraints
- Provide support when applications cannot

OS specific

- Free and commercial
- Some have IDS too

© 2002 Carnegie Mellon University

Module 9: Host System Hardening - slide 56

Our house may have a security system that extends beyond locks, latches, garage door openers, etc. In fact, the system provides another layer of security which to us means that if one of the locks, latches or other devices fails or can be by-passed, the security system is still there to alert its listeners about an intrusion. In some cases, where no lock, latch or other device could be installed, the security system is the only means of defense.

On a computer system, the host-based firewall is analogous to the home security system. It is another defense beyond the specific connection constraints installed at the host level as just discussed. Firewalls can also add constraints at the operating system level when none can be added at the application level. Finally, with host-based firewalls, the administrator can eliminate connections to services that have been disabled and subsequently removed as another way to insure that the service remains disabled, even if the supporting files are reinstalled and the service restarted.

There are many host-based firewall products available. Here is a brief list:

- Windows
  - Zone Alarm[21] - Zone Alarm comes in free and commercial versions, where those that cost have more features and more support.
  - Tiny Personal Firewall[22] - Tiny Personal Firewall (TPF) also comes in free and commercial versions, and those that cost have more features and more support. TPF has a good capability for adding rules with restrictions to further constrain both endpoints of a connection.
  - Blackice Defender[23] - Blackice Defender is a commercial product only. It has intrusion detection capabilities that are tied into Mitre’s Common Vulnerabilities and Exposures (CVE) database[24]. This structure provides a way for the administrator to learn more about vulnerabilities and an attack that may be in progress or soon to start.

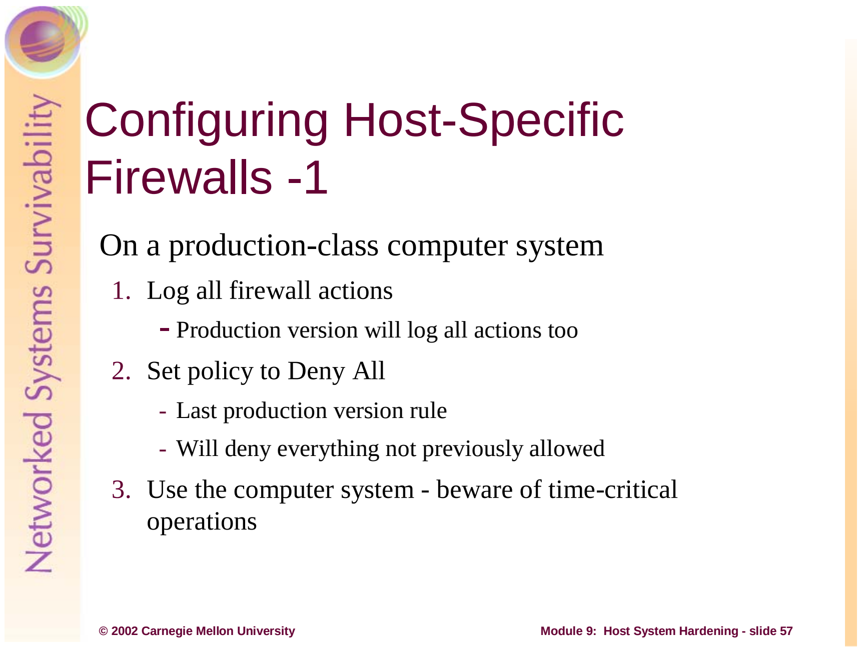
---

[21] http://www.zonelabs.com
[22] http://www.tinysoftware.com/home/tiny?la=EN&va=aa
[23] http://www.networkice.com
[24] http://www.cve.mitre.org

- LINUX-based systems – LINUX has its own internal firewall capabilities, called **IPChains** (stateless packet filtering) or **IPTables** (stateful packet filtering), depending upon the version of the kernel. There are many graphical front ends that simplify the process of setting up a firewall. In addition, there are many sites that describe how to set up the firewall and also contain many compendiums of frequently asked questions (FAQs). One with many such links is [LinuxSec] and it is extensive.
- SOLARIS – There is a public domain firewall system for SOLARIS named IP Filter, written by Darren Reed[25]. It has many features and provides the same core set of firewall services as the other firewalls noted above.

No matter which operating system or firewall system has been selected, the key is to configure that firewall with as restrictive a set of constraints as the computer system and its applications can tolerate. Since the production configuration of a computer system can vary widely, there is no specific set of rules. The next section describes a method for configuring a host firewall for a specific computer system.

[25] http://coombs.anu.edu.au/~avalon/ip-filter.html



Networked Systems Survivability

Configuring Host-Specific Firewalls -1

On a production-class computer system

1. Log all firewall actions
   - Production version will log all actions too
2. Set policy to Deny All
   - Last production version rule
   - Will deny everything not previously allowed
3. Use the computer system - beware of time-critical operations

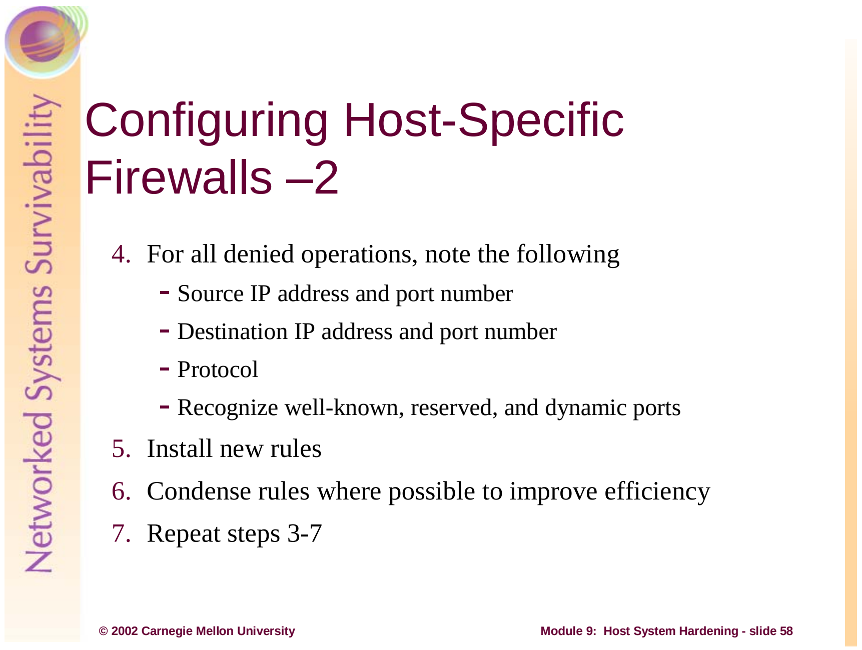
© 2002 Carnegie Mellon University Module 9: Host System Hardening - slide 57

The key to configuring a host-specific firewall is to install and operate the firewall on a production-class computer system. That system is either running in a test environment, where the characteristics mirror those of the eventual production environment, or the real production environment. A test environment is best because configuring the firewall may cause applications to malfunction, harming true production information assets. It is recognized that that may not be possible, so configuration in a real production environment can also be done.

The first step is to enabling logging of all of the firewall's actions. This will also be the production setting so that the administrator can review all firewall actions at a later time. It is extremely important to understand what the firewall is doing with all network traffic presented to it. Logging provides this information.

Next, configure a single rule to deny all access to all services from all sources, no matter whether the computer system generates the network traffic or consumes it. This will cause the firewall to not let anything pass through. This will be the final rule of the production version of the rule set. Additional rules will be added before this *deny all* and they will allow specific traffic. Because of the *deny all* final rule, anything not specifically allowed will be denied.

Finally, use the computer system in as near a production manner as possible. All network traffic produced and intended for the computer system will be denied and logged by the firewall rules. The administrator needs to review the logs frequently and allow specific connections that complement the system's usage. It is important to review these logs in near real-time as some applications may have built-in timers that expire if an operation cannot be completed in a predetermine amount of time. Operations may have to be attempted several times until the proper rules are configured into the firewall.



Configuring Host-Specific Firewalls –2

4. For all denied operations, note the following
   - Source IP address and port number
   - Destination IP address and port number
   - Protocol
   - Recognize well-known, reserved, and dynamic ports
5. Install new rules
6. Condense rules where possible to improve efficiency
7. Repeat steps 3-7

Networked Systems Survivability

© 2002 Carnegie Mellon University Module 9: Host System Hardening - slide 58

For each denied operation, make specific note of the following:

- Source IP address
- Source port
- Destination IP address
- Destination port
- Protocol (i.e., UDP, TCP, ICMP)

One key to the successful configuration of a firewall – host-specific or otherwise – is the efficient configuration of the rules. This means that any condensing of two or more rules into a single rule will likely improve packet throughput. If you can detect patterns in any of the information captured above, then it is probable that rules can be condensed and efficiencies improved.

For example, the Dynamic Host Control Protocol (DHCP) UDP-based service always connects the server's port 67 to the client's port 68. If your network provides several DHCP servers, then individual rules can all be grouped together to define a class of addresses that produces traffic on port 67 intended for your computer system on port 68. By condensing several rules – ostensibly one for each producer – into a single rule, you can increase the performance through the firewall system.

With respect to port numbers, you need to be aware of well-known ports, registered ports, and dynamic ports. The first two of these are defined in [IANA] and the third is operating system-dependent. For example, on LINUX, the dynamic ports used by the operating system are defined with the following:

```
/sbin/sysctl -w net.ipv4.ip_local_port_range='32768 65535'
```

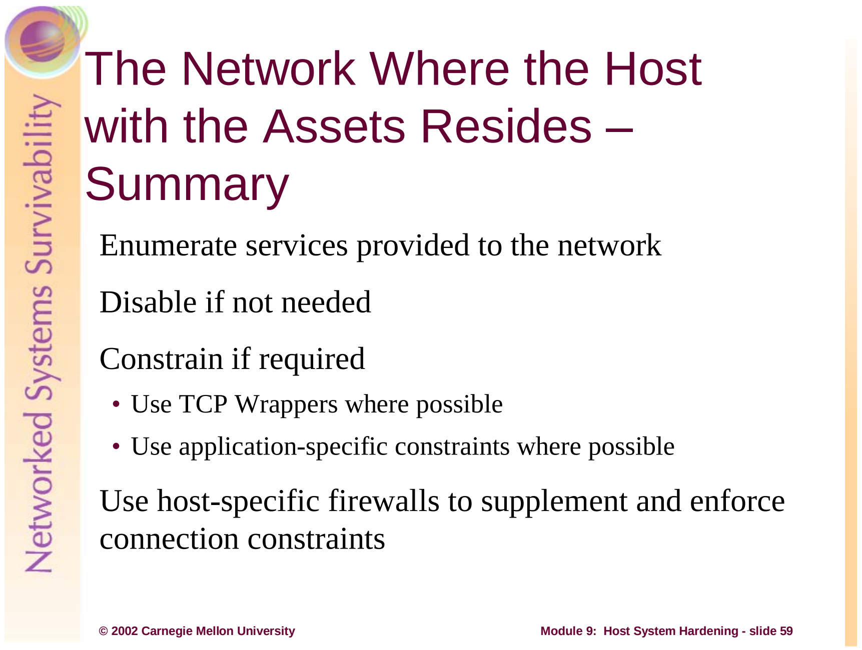
This means that any port in the range of 32K to 64K-1 will change from connection to connection and therefore the specific port used should not appear in a firewall configuration rule unless the administrator knows some additional information that contradicts this. In general, ports in the dynamic range are indeed that – dynamic – and should be treated as such when configuring firewall rules.

In summary, do the following:

1. Log all firewall actions
2. Install single *deny all* rule

3. Use the computer system
4. Note firewall exceptions
5. Install new rules
6. Condense rules where possible
7. Repeat steps 3-7

This procedure yields an efficient firewall rule set where the default rule is to deny the connection while allowing only that which has been explicitly learned. Note that as the computer system matures in its production setting, the firewall rules will likely need to be adjusted as conditions warrant.


Networked Systems Survivability

The Network Where the Host with the Assets Resides – Summary

Enumerate services provided to the network

Disable if not needed

Constrain if required

- Use TCP Wrappers where possible
- Use application-specific constraints where possible

Use host-specific firewalls to supplement and enforce connection constraints

© 2002 Carnegie Mellon University

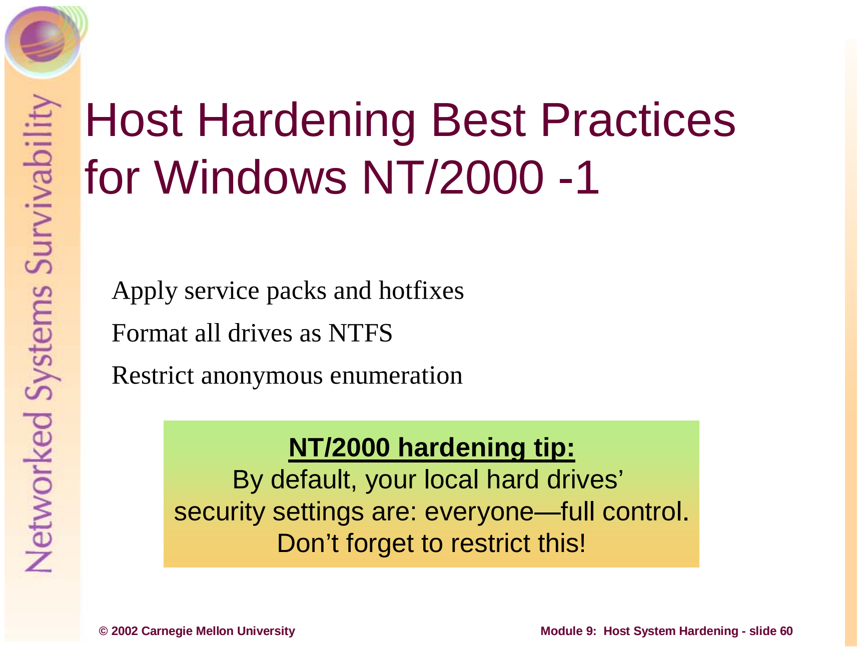
Module 9: Host System Hardening - slide 59

When the host resides on a network, that host computer system must be evaluated with respect to the services it provides to that network. Every service must be accounted for, and either disabled and removed from the system, or configured to constrain the set of computer systems that can use that service. There are various programs that can be used to enumerate the set of services that a host provides, from both the host and the network perspective.

No matter the disposition of the service, the way in which it is started must be clearly identified. If the service is to be disabled, the instantiation method must be changed to disable the service. If it is to be constrained, the instantiation method may be changed depending on the service. In some cases, those changes will suffice, whereas in other cases, the application that provides the service may need to be configured directly. Finally, some services have no capability to constrain the set of hosts to which they provide a service.

Independent of whether services are disabled or constrained, use host-specific firewalls as a defense-in-depth measure to guard against services being re-enabled and to supplement other constraining mechanisms. Finally, when no other mechanisms are available, host-specific firewalls provide the only means to constrain the computer systems seeking service. This section provided a method for configuring such a firewall.

The house has many doors, windows, and other access points. So does a computer system – through its offered services – and attention must be given to restrict access to only that set of networked computer systems that are allowed to receive service.



Host Hardening Best Practices for Windows NT/2000 -1

Apply service packs and hotfixes

Format all drives as NTFS

Restrict anonymous enumeration

NT/2000 hardening tip:
By default, your local hard drives' security settings are: everyone—full control. Don't forget to restrict this!

Networked Systems Survivability

© 2002 Carnegie Mellon University — Module 9: Host System Hardening - slide 60

Service Packs and Hotfixes

A service pack is a periodic update to the operating system that contains fixes to vulnerabilities and bugs. As of this writing, Microsoft has released six service packs for Windows NT 4.0 and two service packs for Windows 2000. Updates addressing specific vulnerabilities and bugs introduced between Service Packs are called *hotfixes*. Service packs are cumulative, meaning they include all hotfixes from previous service packs, as well as new fixes. In addition to installing the latest service packs, it is important to install new hotfixes, as these patches will often address current attacks that are proliferating throughout networks. Although Microsoft recommends applying a hotfix only if a system experiences the specific problem, it is recommended that all security-related hotfixes be installed immediately after installation of the latest service pack. If a service pack is reapplied at any time, the hotfixes must also be re-installed. [NSA1]

Applying Service packs and Hotfixes on larger networks can be a significant challenge. Using tools like Systems Management Server (SMS), Windows 2000 Installation Services, and even using the embedded Windows Update feature can makes this a little more manageable. Microsoft has also released a free command-line tool called **Qchain.exe** that gives system administrators the ability to safely chain hotfixes together. Hotfix chaining involves installing multiple hotfixes without rebooting between each installation.

Format Drives as NTFS

NTFS partitions offer access controls and protections that aren't available with the FAT, FAT32, or FAT32x file systems. Directory and File-level access controls can be set with NTFS as well as the additional capabilities of encryption and compression (NTFS v.5). Make sure that all partitions on your computer are formatted using NTFS. If necessary, use the convert utility to non-destructively convert your FAT partitions to NTFS.

Restrict Anonymous Enumeration

Windows NT 4 and Windows 2000 have a feature that allows anonymous logon users to list system information, including usernames and details, account policies, and share names. Users who want enhanced security can restrict this enumeration function.

A registry key called `RestrictAnonymous` controls the level of enumeration granted to an anonymous user. Often referred to as Null Connections, they are included in the built-in `Everyone` security group; thus, anonymous users have access to any resources that the Everyone group has access to. If you keep `RestrictAnonymous` at its default setting of 0, any user can obtain system information, including: usernames and details, account policies, and share names. Account policy includes such information as minimum password length, whether blank passwords are permitted, maximum password age, and password history. This information could be used in an attack against your system. The list of usernames and share names could help potential attackers identify who is an administrator, which machines have weak account protection, and which machines are sharing information with the network.

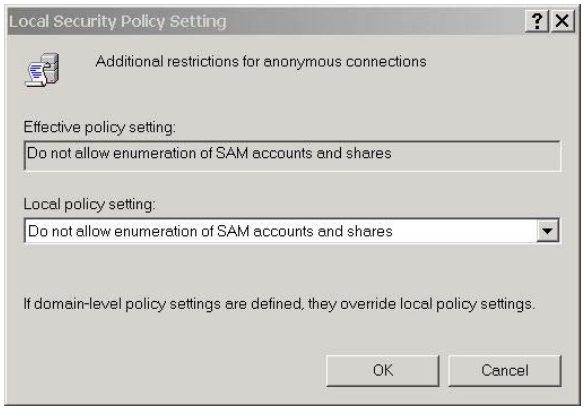
To restrict anonymous connections from enumerating this information, change the `RestrictAnonymous` security settings. This can be done in Windows 2000 or Windows NT4 SP4 by changing the registry key from 0 to 1 in NT4 or from 0 to 1 or 2 in Windows 2000. These numbers correspond to the following settings:

0 – "None. Rely on default permissions"
1 – "Do not allow enumeration of SAM accounts and names"
2 – "No access without explicit anonymous permissions" (not available on NT 4)

In NT 4: Set the following registry key:[26]

Hive: `HKEY_LOCAL_MACHINE`
Key: `System\CurrentControlSet\Control\Lsa`
Name: `RestrictAnonymous`
Type: `REG_DWORD`
Value: 1

In Windows 2000, use the Local Security Policy MMC(under Local Policies/Security Options) to do this:[27]


Local Security Policy Setting

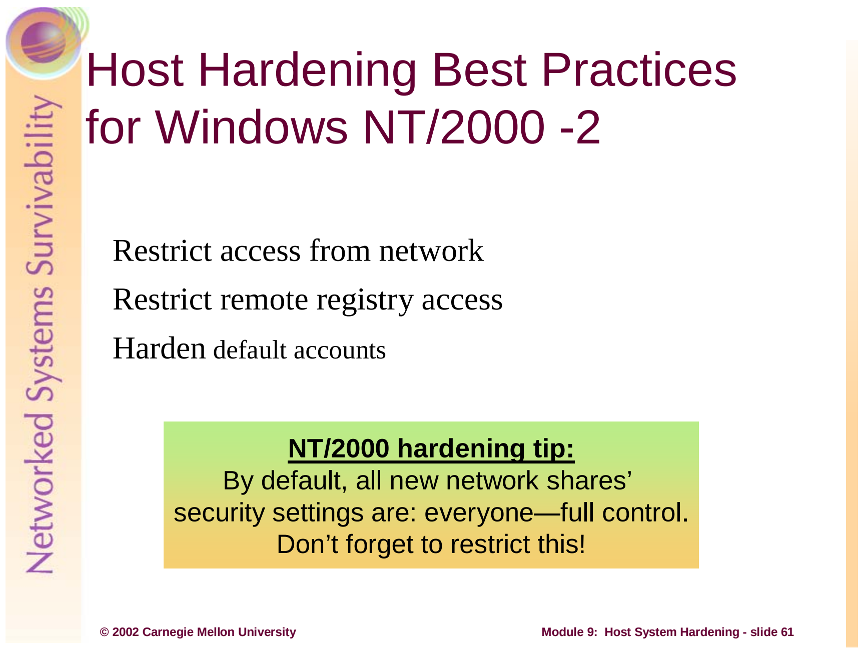
Additional restrictions for anonymous connections

Effective policy setting:
Do not allow enumeration of SAM accounts and shares

Local policy setting:
Do not allow enumeration of SAM accounts and shares

If domain-level policy settings are defined, they override local policy settings.

[26] http://support.microsoft.com/support/kb/articles/Q246/2/61.ASP
[27] http://support.microsoft.com/support/kb/articles/Q246/2/61.ASP



Networked Systems Survivability

Host Hardening Best Practices for Windows NT/2000 -2

Restrict access from network

Restrict remote registry access

Harden default accounts

<u>NT/2000 hardening tip:</u>
By default, all new network shares' security settings are: everyone—full control. Don't forget to restrict this!

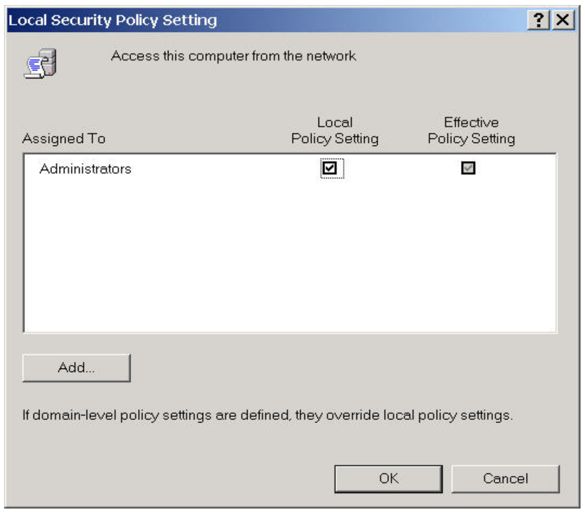
© 2002 Carnegie Mellon University

Module 9: Host System Hardening - slide 61

<u>Restrict Access from Network</u>

Remove the `Everyone` group (there by default) from the *Access this Computer from the Network* user right. Replace it with the group or tighter restrictions. In Windows NT 4.0, this can be accomplished under *User Manager* → *Policies* → *User Rights*. In Windows 2000, this can be done via the Local Security Policy MMC (*Local Security Policies/User Rights Assignment*).

This can also be accomplished in a Windows 2000 Domain using the Security Configuration Toolset and Group Policy.



Local Security Policy Setting

Access this computer from the network

Assigned To	Local Policy Setting	Effective Policy Setting
Administrators	✓	✓

If domain-level policy settings are defined, they override local policy settings.

Remote Registry Access

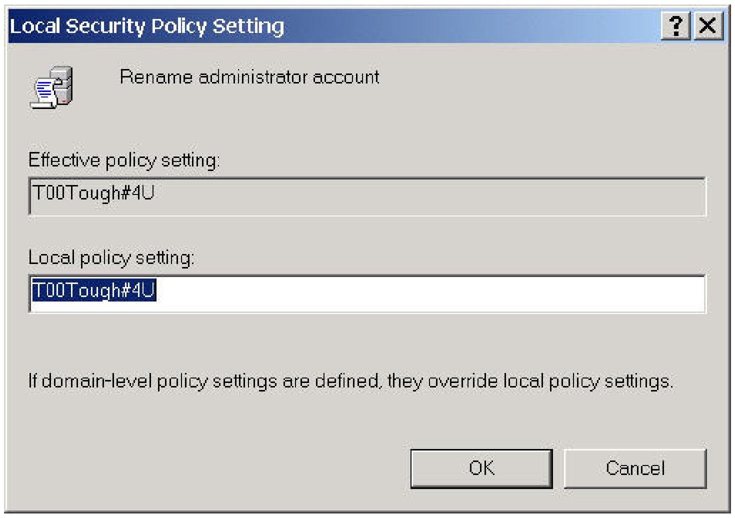
Do not allow remote registry access. There are many registry keys that allow the `Everyone` group, and therefore anonymous users, read and/or set value permissions. If an unauthorized user was able to remotely edit the registry, he could modify registry keys in an attempt to gain elevated privileges. Restricting remote registry access is accomplished by setting security permissions on the :`HKLM\SYSTEM\CurrentControlSet\Control\SecurePipeServers\winreg` key. It is highly recommended that only Administrators and System have remote access to the registry.

Harden Default Accounts

Because the Administrator account is built in to every copy of Windows NT/2000, it presents a well-known objective for attackers. To make it more difficult to attack the Administrator account, do the following both for the local Administrator account on each computer:

- Rename the account to a nonobvious name (e.g., not `admin`, `root`, etc.)
- Establish a decoy account named `Administrator` with no privileges. Scan the event log regularly looking for attempts to use this account.
- Enable account lockout on the real Administrator accounts by using the passprop utility
- Disable the local computer's Administrator account.

In Windows 2000, use the Local Security Policy MMC (under *Local Policies*/*Security Options*) or through User Manager (NT 4 as well)


Local Security Policy Setting

Rename administrator account

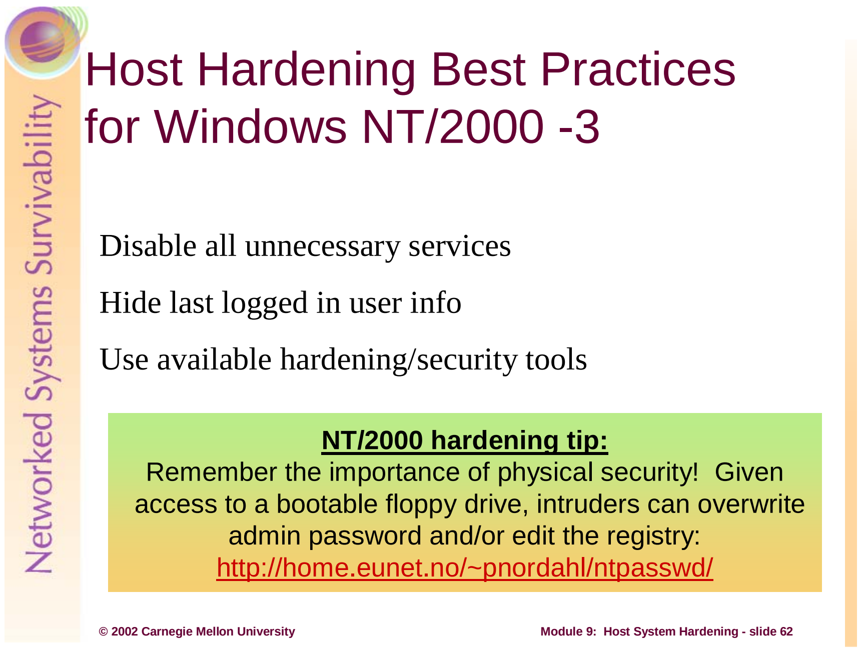
Effective policy setting:
T00Tough#4U

Local policy setting:
T00Tough#4U

If domain-level policy settings are defined, they override local policy settings.

For maximum protection, make sure the Administrator account password is at least eight characters long and that it includes at least one punctuation mark or nonprinting ASCII character in the first seven characters. In addition, the Administrator account password should not be synchronized across multiple computers. Different passwords should be used on each computer to raise the level of security in the workgroup or domain.

Ensure that the Guest Account is disabled (disabled by default). Ensure that all accounts (service and user) have passwords regardless if the account is enabled or disabled.



Host Hardening Best Practices for Windows NT/2000 -3

Disable all unnecessary services

Hide last logged in user info

Use available hardening/security tools

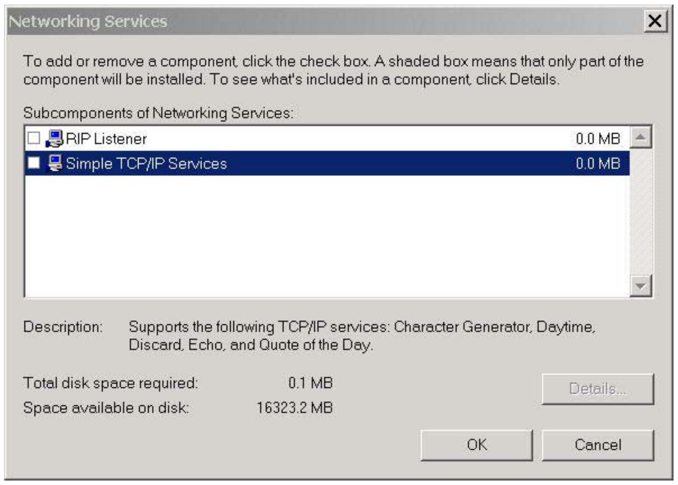
NT/2000 hardening tip:
Remember the importance of physical security! Given access to a bootable floppy drive, intruders can overwrite admin password and/or edit the registry:
http://home.eunet.no/~pnordahl/ntpasswd/

Networked Systems Survivability
© 2002 Carnegie Mellon University
Module 9: Host System Hardening - slide 62

Disable Unnecessary Services

From a security standpoint, it is important to run only the minimum required essential services. These can include application services (like personal web server or Internet Information Server) as well as networking services. By default, Windows NT installs NWLink (Novell Netware networking) and NetBEUI, which is Microsoft's NETBIOS networking protocol. Since most clients only require TCP/IP, all other protocols should be disabled.

Additionally, Simple TCP/IP Services should be uninstalled, as these are commonly used as the basis of attacks.



To add or remove a component, click the check box. A shaded box means that only part of the component will be installed. To see what's included in a component, click Details.

Subcomponents of Networking Services:

☐ RIP Listener	0.0 MB
☐ Simple TCP/IP Services	0.0 MB

Description: Supports the following TCP/IP services: Character Generator, Daytime, Discard, Echo, and Quote of the Day.

Total disk space required: 0.1 MB
Space available on disk: 16323.2 MB

Some other services that should be disabled on most hosts:

- Telnet Server
- FTP Server
- Remote Access
- File and Print Sharing for Microsoft Networks (if not essential)

Hide Last Logged in User

Always showing a valid username in the User Login box gives away 50% of the required information (all he/she needs further is their password) to potential intruders.

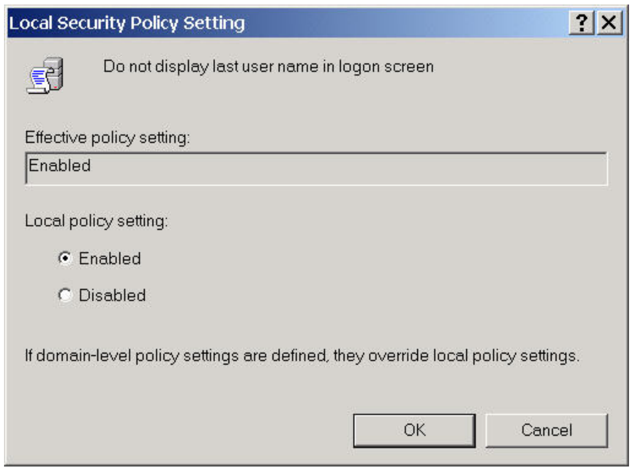
To hide the last logged in user in NT 4, make the following Registry change:

Start **Regedt32.exe** and locate the following registry key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon
```

Add a `REG_SZ` value named `DontDisplayLastUserName`. Set the data value to 1 to enable the new feature. A data value of 0 disables the feature.

In Windows 2000, use the Local Security Policy MMC (under *Local Policies*/*Security Options*) to set this:



Do not display last user name in logon screen

Effective policy setting:
Enabled

Local policy setting:
(●) Enabled
( ) Disabled

If domain-level policy settings are defined, they override local policy settings.

Use Available Hardening/Security Tools

Microsoft has provided many new security related tools that can be used to identify problems and harden Windows hosts.[28] Some of the more useful of these are:

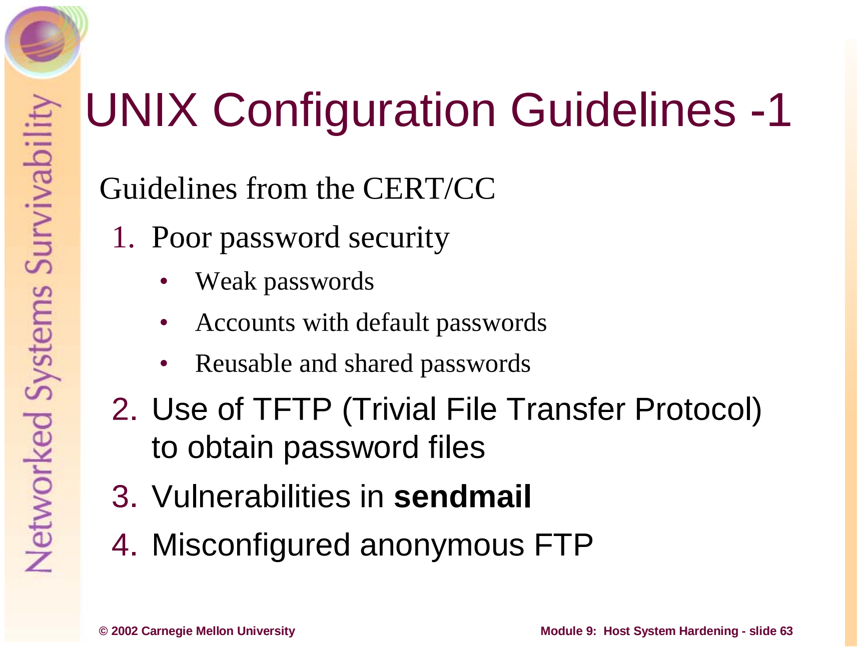
- Microsoft Personal Security Advisor – Web-based--checks system for holes/bad configs.
- `HFNetCHk`- Allows for local and remote checking on currency of hotfixes applied
- IIS Lockdown Tool – Securely configures IIS 4.0 or 5.0 Web Servers

Gibson Research has a number of tools that can be helpful in securing Windows systems.[29] Some of the most used are:

- ShieldsUp! – Tests how susceptible systems are to Microsoft Networking based attacks
- LeakTest – Checks the effectiveness of Personal Firewalls (including the XP Firewall)

Finally, Sysinternals.com offers too many Freeware Windows tools to list.[30] Many of these can be used within a security context.

[28] http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/tools/w2kprocl.asp
[29] http://grc.com/default.htm
[30] http://www.sysinternals.com/



UNIX Configuration Guidelines -1

Guidelines from the CERT/CC

1. Poor password security
   - Weak passwords
   - Accounts with default passwords
   - Reusable and shared passwords
2. Use of TFTP (Trivial File Transfer Protocol) to obtain password files
3. Vulnerabilities in **sendmail**
4. Misconfigured anonymous FTP

Networked Systems Survivability

© 2002 Carnegie Mellon University   Module 9: Host System Hardening - slide 63

The configuration guidelines in this section are from [CERT 97].

1. Poor Password Security
   The basic form of authentication used to control access to a UNIX host is a username and password combination. Intruders have established mechanisms and tools to compromise password information by leveraging a variety of common problems.

   1) Weak passwords

      Encourage your users to choose passwords that are difficult to guess (for example, words that are not in any dictionary of any language; no proper nouns, including names of "famous" real or fictitious characters; no acronyms that are commonly used by computer professionals; no simple variations of first or last names.) Furthermore, inform your users not to leave any clear text username/password information in files on any system.

      A good heuristic for choosing a password is to choose an easy-to-remember phrase, such as "By The Dawn's Early Light", and use the first letters to form a password. Add some punctuation or mix case letters as well. For the phrase above, one example password might be: `bt}DeL{.` (DO NOT use this sample phrase for your password.)

      If intruders can get a password file, they usually move or copy it to another machine and run password-guessing programs on it. These programs involve large dictionary searches, and they run quickly even on slow machines. Most systems that do not put any controls of the type of passwords used probably have at least one password that can be easily guessed. CERT Incident Note IN-98.03 describes intruder activity that is based on a stolen password file.

      http://www.cert.org/incident_notes/IN-98.03.html

      If you believe that your password file may have been taken, change all the passwords on the system. At the very least, you should change all system passwords because an intruder may concentrate on those and may be able to guess even a reasonably "good" password. Intruders often use compromised accounts to attempt to gain privileged access on vulnerable systems, so we encourage you to follow the steps in

- http://www.cert.org/tech_tips/intruder_detection_checklist.html
- http://www.cert.org/tech_tips/win-UNIX-system_compromise.html

For further information about protecting your system from password-based attacks, see

- http://www.cert.org/tech_tips/passwd_file_protection.html

2) Accounts with default passwords

Intruders exploit system default passwords that have not been changed since installation, including accounts with vendor-supplied default passwords. In some cases, accounts do not have a password assigned by default. CERT Incident Note IN-98.01 describes intruder activity that is based on exploitations of accounts without passwords.

http://www.cert.org/incident_notes/IN-98.01.irix.html

Be sure to change all default passwords on computer systems and networking equipment prior to deployment. Also, be aware that product upgrades can quietly change account passwords to a new default. It is best to change the passwords of default accounts after applying updates.

Scan your password file for extra UID 0 accounts, accounts with no password, or new entries in the password file. Do not allow any accounts without passwords. Remove entries for unused accounts from the password file. To disable an account, change the password field in the */etc/passwd* file to an asterisk '*' and change the login shell to */bin/false* to ensure that an intruder cannot login to the account from a trusted system on the network.

3) Reusable and shared passwords

Even excellent passwords are not safe. Programs such as packet sniffers can capture them if the passwords are sent across networks in clear text (whether on a subnet, a local network, or the Internet). It is common for intruders to use packet sniffers on compromised systems to harvest passwords.

CERT Incident Note IN-99-06 describes widespread intruder activity involving distributed sniffers used to harvest username and password information from a network.

http://www.cert.org/incident_notes/IN-99-06.html

At the very least, a single password should not be used to protect multiple accounts. If an intruder is able to compromise a shared password just once, all of the accounts sharing the password are compromised. Each account, or resource, protected by a password should have it's own unique password.

To overcome the threat posed by packet sniffers, we recommend using one-time passwords, especially for authenticated access from external networks and for access to sensitive resources like name servers and routers. For more information, see Appendix B of the following advisory:

http://www.cert.org/advisories/CA-94.01.ongoing.network.monitoring.attacks.html

Another approach is to use strong authentication mechanisms such as secure shell, SSL, or kerberos. Secure shell, or ssh, is widely available for many different platforms. For more information about secure shell, see

- http://www.ssh.com/index.html
- http://www.openssh.com/

2. Use of TFTP (Trivial File Transfer Protocol) to obtain password files

   To test your system for this vulnerability, connect to your system using tftp and try

   ```
   get /etc/motd
   ```

   If you can do this, anyone else on the network can probably get your password file. To avoid the problem, disable **tftpd**. If you must have **tftpd**, ensure that it is configured with restricted access. For further information, see

   http://www.cert.org/advisories/CA-91.18.Active.Internet.tftp.Attacks.html

   As mentioned in Section 1 above, if you believe your password file may have been taken, the safest course is to change all passwords in the system.

3. Vulnerabilities in **sendmail**

   There have been a number of vulnerabilities identified over the years in **sendmail**(8). To the best of our knowledge, the current version of **sendmail** addresses those known vulnerabilities.

   To determine which version of **sendmail** is running, use telnet to connect to the SMTP port (25) on your system:

   ```
   telnet hostname 25
   ```

   We encourage you to keep up to date with the latest version of **sendmail** from your vendor, and ensure that it is up to date with security patches or workarounds detailed in CERT advisories and bulletins. For information about the latest version of *sendmail*, see

   ftp://info.cert.org/pub/latest_sw_versions/sendmail

   In addition, we encourage you to use the following tools, both of which are distributed with the latest versions of **sendmail**:

   a. **smrsh**, the **sendmail** restricted shell, controls the way o that incoming mail messages can interact with your operating system. For instance, when configured correctly, **smrsh** can prevent an intruder from using pipes to execute arbitrary commands on your system.

   b. **mail.local** can be used to control the way in which the */bin/mail* program is used on your system.

      ftp://info.cert.org/pub/cert_advisories/CA-95:02.binmail.vulnerabilities

   If you are not running a recent version of **sendmail**, the above tools may also be obtained individually from a number of sources, including

   c. ftp://info.cert.org/pub/tools/mail.local/

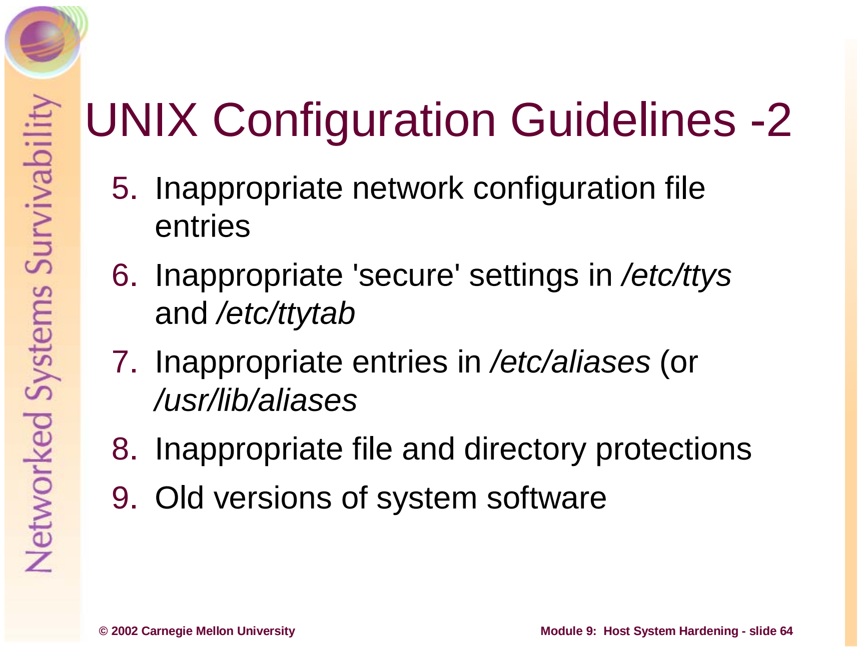
   d. ftp://info.cert.org/pub/tools/smrsh/

   Warning: If you are running such an old version of **sendmail** that you must install these tools separately, intruders will continue to be able to exploit vulnerabilities that were fixed in later versions of **sendmail**. We urge you to upgrade to the current version of **sendmail** mail and then run the tools, which are included with the distribution.

4. Misconfigured anonymous FTP

   In addition to making sure that you are running the most recent version of **ftpd**, check your anonymous FTP configuration. It is important to follow the instructions provided with the operating system to properly configure the files and directories available through anonymous FTP (for example, file and directory permissions, ownership and group). Note that you should not use your system's standard password file or group file as the password file or group file for FTP. The anonymous FTP root directory and its two subdirectories, etc and bin, should not be owned by ftp. For more information about configuring anonymous FTP, see

ftp://info.cert.org/pub/tech_tips/anonymous_ftp_config



UNIX Configuration Guidelines -2

5. Inappropriate network configuration file entries
6. Inappropriate 'secure' settings in */etc/ttys* and */etc/ttytab*
7. Inappropriate entries in */etc/aliases* (or */usr/lib/aliases*
8. Inappropriate file and directory protections
9. Old versions of system software

Networked Systems Survivability
© 2002 Carnegie Mellon University
Module 9: Host System Hardening - slide 64

5. Inappropriate network configuration file entries

   Several vendors supply */etc/hosts.equiv* files with a '+' (plus sign) entry. The '+' entry should be removed from this file because it means that your system will trust all other systems. Other files that should not contain a '+' entry include all *.rhosts* files on the system. These files should not be world-writable.

   If your */usr/lib/X11/xdm/Xsession* file includes an **xhost** command with a '+' entry, such as

   ```
   /usr/bin/X11/xhost +
   ```

   remove that line. (Note that the **xhost** command may be in a different directory tree on your system.) If such a line remains intact, anyone on the network can talk to the X server and potentially insert commands into windows or read console keystrokes.

6. Inappropriate 'secure' settings in */etc/ttys* and */etc/ttytab*

   Check the file */etc/ttys* or */etc/ttytab* (depending on the release of UNIX being used). The ONLY terminal that should be set to 'secure' should be the console.

7. Inappropriate entries in */etc/aliases* (or */usr/lib/aliases*)

   Examine the */etc/aliases* (or */usr/lib/aliases*) mail alias file for inappropriate entries. Some alias files include an alias named `uudecode` or just `decode`. If this alias exists on your system and you are not explicitly using it, then you should remove it.

8. Inappropriate file and directory protections

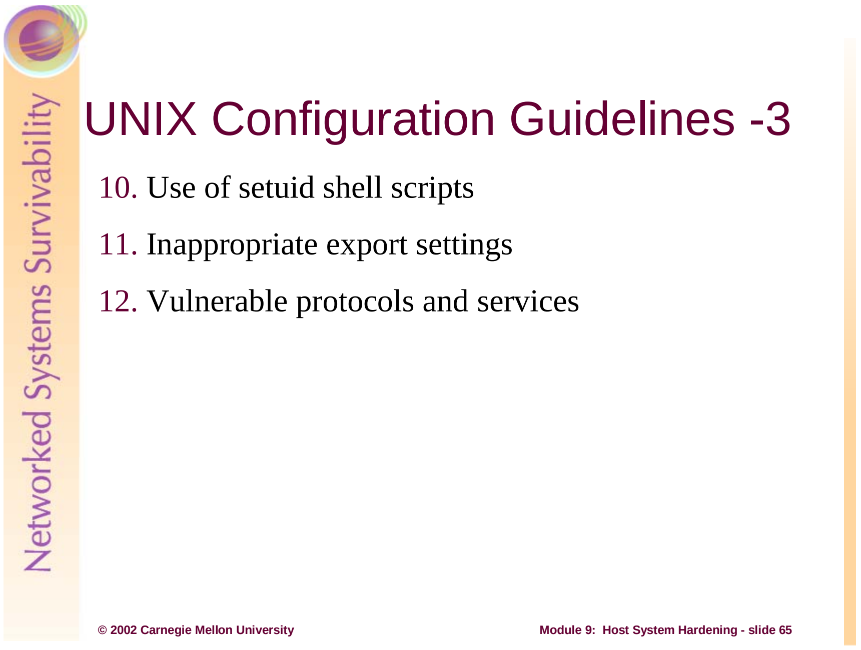
   Check your system documentation to establish the correct file and directory protections and ownership for system files and directories. In particular, check the / (root) and */etc* directories, and all system and network configuration files. Examine file and directory protections before and after installing software or running verification utilities. These procedures can cause file and directory protections to change.

9. Old versions of system software

   Older versions of operating systems often have security vulnerabilities that are well known to intruders. To minimize your vulnerability to attacks, keep the version of your operating system up to date and apply security patches appropriate to your system(s) as soon as they become available.

   For information about software upgrades that fix security problems, their sources, and their MD5 checksums, see

   ftp://info.cert.org/pub/latest_sw_versions/



UNIX Configuration Guidelines -3

10. Use of setuid shell scripts
11. Inappropriate export settings
12. Vulnerable protocols and services

Networked Systems Survivability

Module 9: Host System Hardening - slide 65

10. Use of setuid shell scripts

    Setuid shell scripts (especially setuid root) can pose potential security problems, a fact that has been well documented in many UNIX system administration texts. Do not create or allow setuid shell scripts, especially setuid root.

11. Inappropriate export settings

    Use the **showmount**(8) utility to check that the configuration of the */etc/exports* files on your hosts are correct.

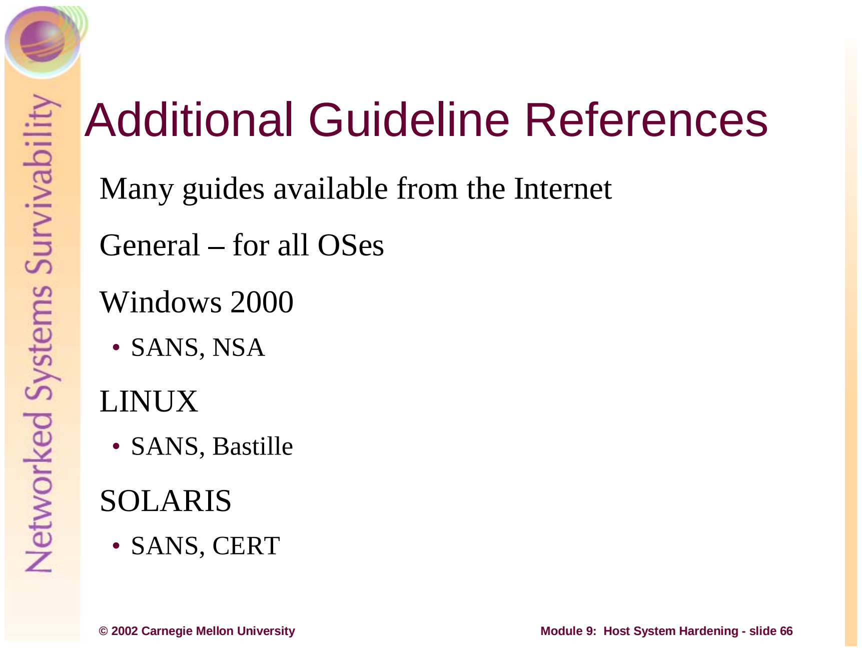
    - Wherever possible, file systems should be exported read-only.
    - Do not self-reference an NFS server in its own exports file. That is, the exports file should not export an NFS server to itself nor to any netgroups that include the NFS server.
    - Do not allow the exports file to contain a "localhost" entry.
    - Export file systems only to hosts that require them.
    - Export only to fully qualified hostnames.
    - Ensure that export lists do not exceed 256 characters (after the aliases have been expanded) or that all security patches relating to this problem have been applied.

    The CERT Coordination Center is aware that intruders are using tools that exploit a number of NFS vulnerabilities. This can result in a root compromise, depending on the vulnerability being exploited. We encourage you to limit your exposure to these attacks by implementing the security measures outlined in CERT advisory CA-94:15. For this and other information about the NFS vulnerability, see

    ftp://info.cert.org/pub/cert_advisories/CA-94:15.NFS.Vulnerabilities

12. Vulnerable protocols and services

    You may want to consider filtering certain TCP/IP services at your firewall or router. For some related suggestions, please refer to *Packet Filtering For Firewall Systems*, available from ftp://info.cert.org/pub/tech_tips/packet_filtering



Additional Guideline References

Many guides available from the Internet

General – for all OSes

Windows 2000

- SANS, NSA

LINUX

- SANS, Bastille

SOLARIS

- SANS, CERT

© 2002 Carnegie Mellon University Module 9: Host System Hardening - slide 66

Networked Systems Survivability

There are many guides available from other sources that further describe the process of hardening host computer systems. These should also be considered when an administrator decides upon a strategy. Here is a list:

- General
  - **The CERT Guide to System and Network Security Practices** [Allen 01] **-** The following is excerpted from that site:

    Networked systems and the sensitive information they contain can be compromised despite an administrator's best efforts. What is the best way to protect computer networks and systems? Administrators need a clear and comprehensive set of security practices that are easy to find and follow.

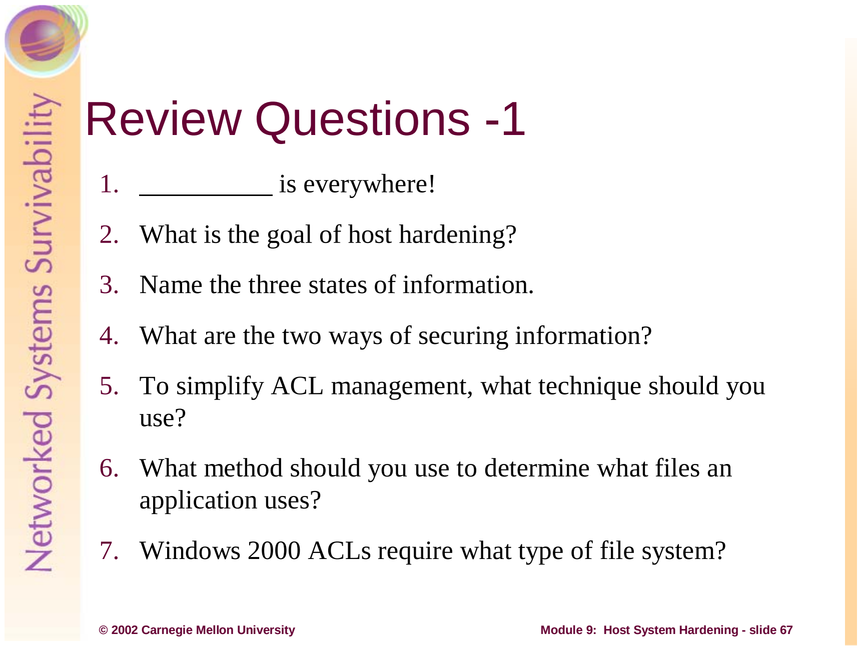
    The SEI is proud to announce the publication of The CERT® Guide to System and Network Security Practices, written by Julia Allen, a senior member of the technical staff, and published by Addison-Wesley. The book is available at walk-in and online bookstores. It presents more than fifty best practices that address all phases of securing systems and networks. Using a practical, phased approach, the book shows administrators how to protect systems and networks against malicious and inadvertent compromise based on security incidents reported to the CERT/CC. Whether you manage systems and networks or configure, operate, and maintain them, you will find guidance in The CERT® Guide to System and Network Security Practices that is easy to implement and protects your information infrastructure.
  - *The 60 Minute Network Security Guide (First Steps Towards a Secure Network Environment* [NSA 01] - The topics covered are:
    - General Guidance
    - Perimeter Routers and Firewalls
    - Windows NT 4.0 and Windows 2000
    - Microsoft Applications
    - UNIX Networks
    - UNIX Web Servers
    - Intrusion Detection Systems

- Windows 2000
  - *Windows 2000 Security: Step by Step* [SANS 01a] by SANS. This is a complete guide that describes all aspects of securing a Windows 2000 system. The topics covered are:
    - General Security Guidelines
    - Physical Data Security
    - Windows 2000 Security Policy Configuration
    - Additional Utilities
    - File, Folder, and Registry Permissions
    - Windows 2000 Security Configuration and Analysis Tool
    - Windows 2000 Recovery Options
    - Windows 2000 Domains: Active Directory Services
    - Windows 2000 Application Security
  - NSA Security Guides - These guides are very complete and detailed in their description of security issues, practices, and techniques for Windows 2000 systems. There are 19 guides in the set. [NSA01a] [NSA 01b] [NSA 01c] [NSA 01d] [NSA 01e] [NSA 01f] [NSA 01f] [NSA 01h] [NSA 01i] [NSA 01j] [NSA 01k] [NSA 01l] [NSA 01m] [NSA 01n] [NSA 01o] [NSA 01p] [NSA 01q] [NSA 01r] [NSA 01s]
- UNIX
  - See the AusCERT *UNIX Security Checklist v2.0* [AUSCERT 01]. Since this is presented in the form of a checklist, it is easier for the administrator to apply to their computer system.
- LINUX
  - *Securing Linux: Step-by-Step* by SANS [SANS 01b]. This is a complete guide that describes all aspects of securing a LINUX system. The topics covered are:
    - Before Installation
    - Install Linux
    - Securing Workstation Network Configurations
    - Securing Server Network Configurations
    - Tuning and Packet Firewalls
    - Tools
  - Bastille-Linux [Bastille 02]: If a RedHat[31] LINUX is the system to be secured, the Bastille–LINUX documented tools that give the administrator the resources needed to secure their systems. The support for other versions of is increasing and eventually Debian[32], SuSE[33], TurboLinux and HP-UX[34] will also be included.
- SOLARIS
  - *Solaris Security: Step-by-Step* [SANS 99] - by SANS. This is a complete guide that describes all aspects of securing a Solaris system. The topics covered are:
    - Boot-Time Configuration
    - OS Modifications
    - Installing SSH with TCP Wrappers Support
    - Putting the System into Production

[31] http://www.redhat.com
[32] http://www.debian.org
[33] http://www.suse.com/index_us.html
[34] http://www.hp.com/products1/linux

- CERT Implementations for UNIX Systems [CERT 01] - These implementations from the CERT Coordination Center and are directly primarily at Solaris 2.6 but are applicable to other versions too.

There are many sites and many sources of information. Those given here come from organizations with credibility in the field of computer system security. They are not the only sites that are credible, and as an administrator, you should seek out additional sources and perspectives to further improve the security of your systems.



Review Questions -1

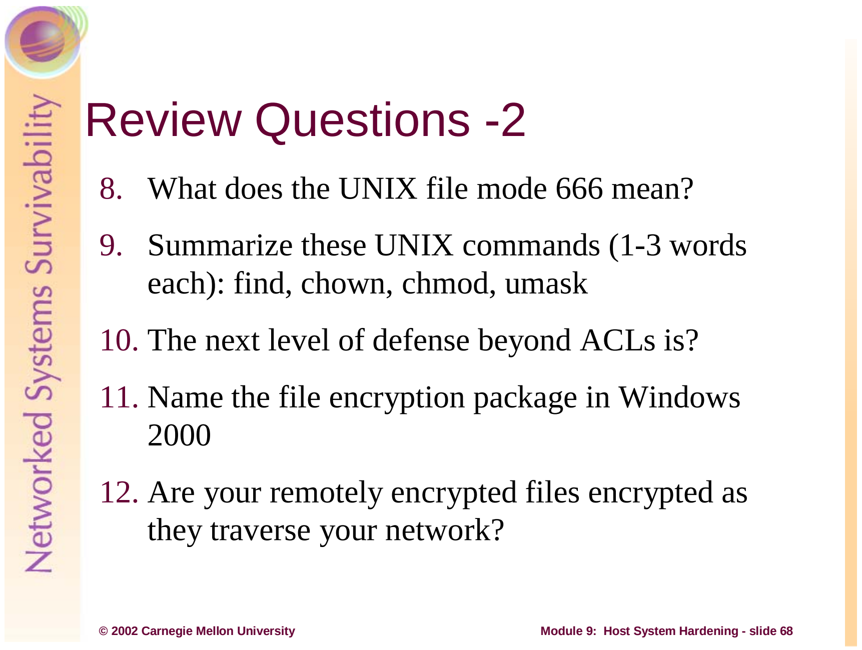
1. __________ is everywhere!
2. What is the goal of host hardening?
3. Name the three states of information.
4. What are the two ways of securing information?
5. To simplify ACL management, what technique should you use?
6. What method should you use to determine what files an application uses?
7. Windows 2000 ACLs require what type of file system?

Networked Systems Survivability

© 2002 Carnegie Mellon University

Module 9: Host System Hardening - slide 67

1. Information
2. Secure computer systems against known threads and vulnerabilities
3. Stored, transmitted, and processed
4. Limiting access and concealing content
5. Groups
6. Run-time analysis
7. NTFS


Networked Systems Survivability

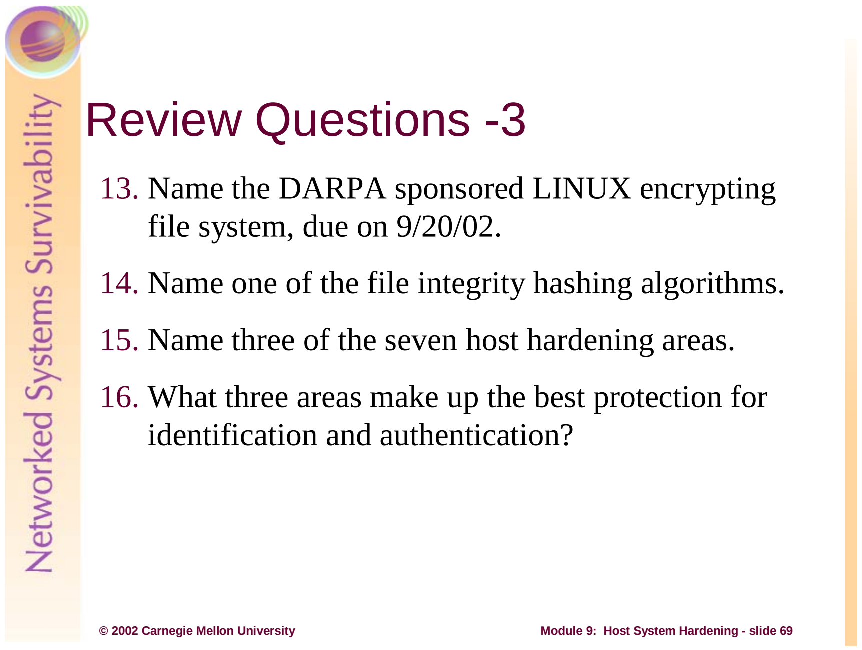
Review Questions -2

8. What does the UNIX file mode 666 mean?
9. Summarize these UNIX commands (1-3 words each): find, chown, chmod, umask
10. The next level of defense beyond ACLs is?
11. Name the file encryption package in Windows 2000
12. Are your remotely encrypted files encrypted as they traverse your network?

© 2002 Carnegie Mellon University

Module 9: Host System Hardening - slide 68

8. The file is readable and writable by the owner, the group, and everyone else
9. **find** – Find files in the file system; **chown** – change file ownership; **chmod** – change file modes; **umask** – define permissions not granted to newly created files and directories
10. Encryption
11. EFS – Encrypting File System
12. No



Networked Systems Survivability

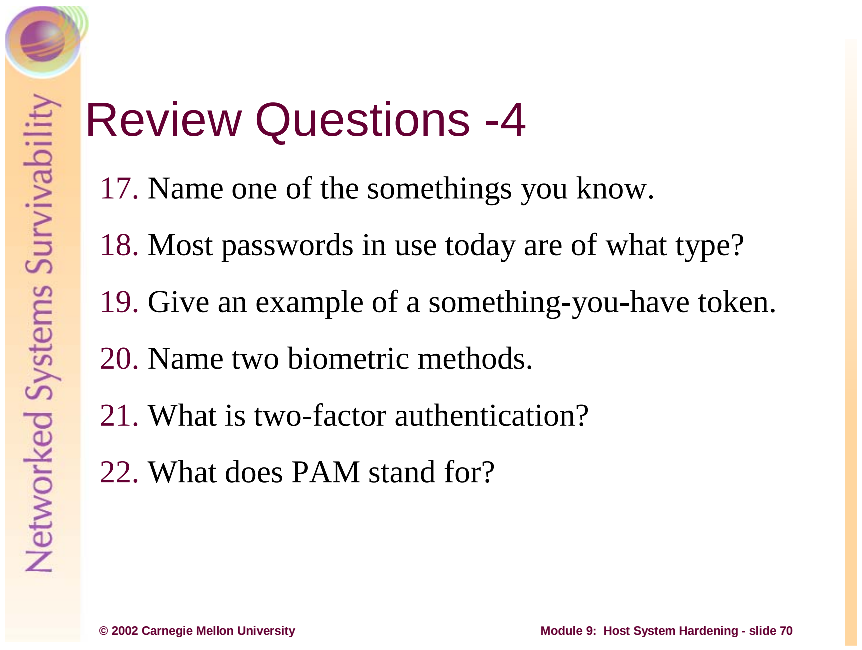
Review Questions -3

13. Name the DARPA sponsored LINUX encrypting file system, due on 9/20/02.
14. Name one of the file integrity hashing algorithms.
15. Name three of the seven host hardening areas.
16. What three areas make up the best protection for identification and authentication?

© 2002 Carnegie Mellon University

Module 9: Host System Hardening - slide 69

13. ReiserFS
14. MD5 and SHA-1
15. Identity and authentication; only needed software; patches; logging; backups, virus and malicious code protection; physical access
16. Something you know, something you have, something you are

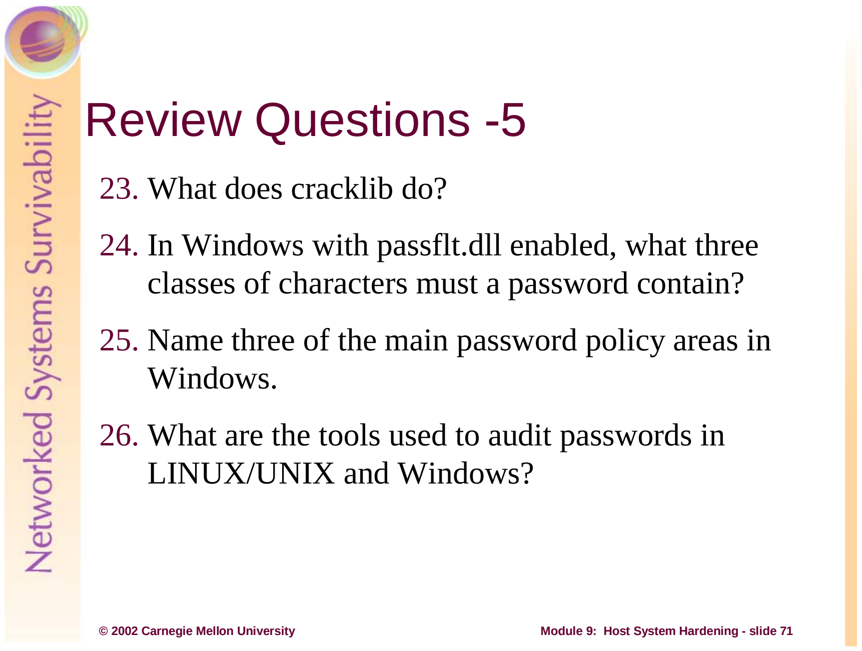

Networked Systems Survivability

Review Questions -4

17. Name one of the somethings you know.
18. Most passwords in use today are of what type?
19. Give an example of a something-you-have token.
20. Name two biometric methods.
21. What is two-factor authentication?
22. What does PAM stand for?

© 2002 Carnegie Mellon University Module 9: Host System Hardening - slide 70

17. Username, password, PIN, pasphrase
18. Reusable
19. S/Key, SecureID, CryptoCard
20. Fingerprints, thumbprint, retina patterns, Iris, hand geometry, vein patterns, voice, signature, keyboard typing skills
21. Two-factor authentication uses any two of something you know, have, and are
22. PAM – Pluggable Authentication Modules



Networked Systems Survivability

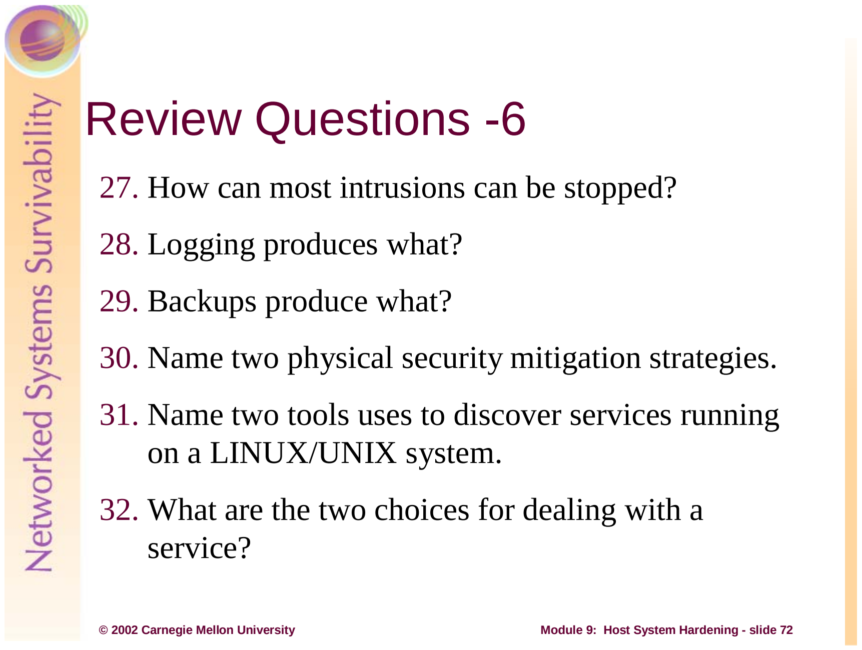
Review Questions -5

23. What does cracklib do?
24. In Windows with passflt.dll enabled, what three classes of characters must a password contain?
25. Name three of the main password policy areas in Windows.
26. What are the tools used to audit passwords in LINUX/UNIX and Windows?

© 2002 Carnegie Mellon University

Module 9: Host System Hardening - slide 71

23. Cracklib helps to insure that passwords cannot be discovered with the **crack** program
24. Letters, numbers, and punctuation
25. History, age, length, complexity
26. **Crack** and **LC3** (aka L0pht Crack)



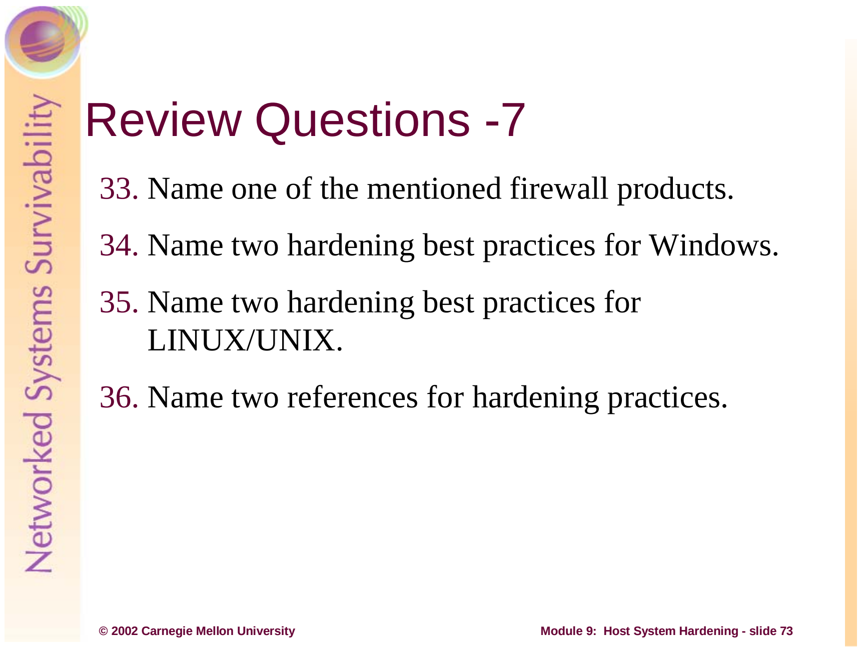
Review Questions -6

27. How can most intrusions can be stopped?
28. Logging produces what?
29. Backups produce what?
30. Name two physical security mitigation strategies.
31. Name two tools uses to discover services running on a LINUX/UNIX system.
32. What are the two choices for dealing with a service?

Networked Systems Survivability

Module 9: Host System Hardening - slide 72

27. Most intrusions can be stopped by patching systems against known vulnerabilities
28. Information
29. Information
30. Assigning BIOS passwords; changing boot device order; locking hardware; removing unnecessary devices; locating systems in a secure facility
31. **lsof**, **netstat**, and **nmap**
32. Disable or configure/constrain

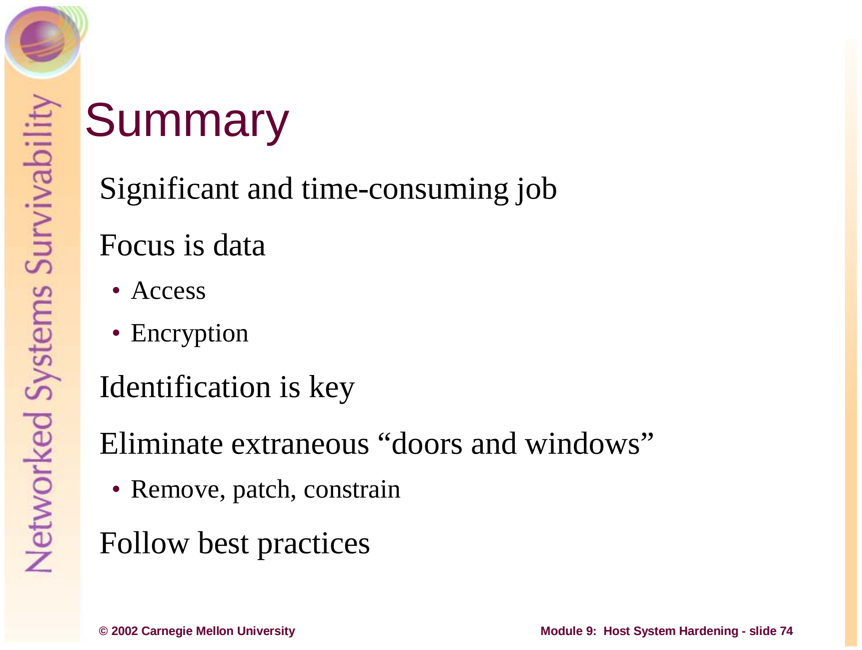


Review Questions -7

33. Name one of the mentioned firewall products.
34. Name two hardening best practices for Windows.
35. Name two hardening best practices for LINUX/UNIX.
36. Name two references for hardening practices.

© 2002 Carnegie Mellon University   Module 9: Host System Hardening - slide 73

33. Zone Alarm, Tiny Personal Firewall, Blackice Defender, IP Chains, IP Tables, IP Filter.
34. Install service packs and hotfixes; format drives as NTFS; restrict anonymous enumeration; restrict access from the network; restrict remote registry access; harden default accounts; disable unnecessary services; hide last logged in user; use available hardening/security tools
35. Poor passwords security; TFTP enabled; vulnerabilities in **sendmail**; misconfigured anonymous FTP; inappropriate network configuration file entries; inappropriate secure settings in */etc/ttys* or */etc/ttytab*; inappropriate entries in */etc/aliases*; inappropriate file and directory protections; old versions of system software; use of setuid shell scripts; inappropriate export settings; vulnerable protocols and settings.
36. See the references.



Summary

Significant and time-consuming job

Focus is data

- Access
- Encryption

Identification is key

Eliminate extraneous “doors and windows”

- Remove, patch, constrain

Follow best practices

Networked Systems Survivability
© 2002 Carnegie Mellon University
Module 9: Host System Hardening - slide 74

The first step in the SKiP Method is the task of hardening a host computer system. It is a significant and time-consuming job. Its purpose is to improve the security of that computer system – or any other entity to which it is applied – to address the presently known threats and vulnerabilities. Because this list continues to grow, the host-hardening task is also dynamic.

Information assets are the focal point for achieving the goals of host hardening. By applying the principles of restricting access and concealing content through encryption to those assets, the process can be simplified.

Assets can be accessed in many ways, from the traditional file system service and the applications that operate it, through network services and the access they provide. By using the principles of restricting access through identification and authorization, supplemented with encryption where necessary, an administrator can apply a consistent strategy when hardening a computer system.

Key to discriminating access to anything – information assets included – is the reliable identification of those who should have access. The traditional login and password scheme is the most prevalent, but the economics of today’s technology makes more esoteric methods affordable. Biometric devices provide stronger a means of identification that address many conventional identification problems.

Access to assets is also restricted through services, be they network services or the more traditional applications running on a computer system. By eliminating unnecessary and potentially vulnerable access points, intruders have fewer “doors and windows” to use to gain that access. This means installing only minimal software, patching what is installed, and constraining what remains.

Finally, there is a large and growing collection of best practices that should be applied when hardening a host. This module has detailed some of them and listed others by reference. An administrator needs to review these to develop the list of practices best suited for their environments, and then continue to review those sources for new steps to be applied.

Host hardening is but one step in the SKiP Method whose goals are improving the survivability of a computer system, network, or sub-infrastructure.

References

[Ackerman] *strace - trace system calls and signals*. Available at: http://www.linux.com/develop/man/1/strace

[Allen 01] **The CERT Guide to System and Network Security Practices,** Addison-Wesley. ISBN 0-201-73723-X. http://cseng.aw.com/book/0,,020173723X,00.html

[AUSCERT 01] *UNIX Security Checklist v2.0*. Available at: http://www.auscert.org.au/Information/Auscert_info/Papers/usc20.html

[Bastille 01] *Bastille-Linux, Version 1.2.0.* Available at: http://www.bastille-linux.org

[BIOSCentral] *BIOS Basics.* Available at: http://www.bioscentral.com/misc/biosbasics.htm

[Bird 01] *Paranoia is a Good Thing.* Drew Bird, September 12, 2001. Available at: http://www.networkstorageforum.com/outsourcing/features/article/0,,10562_882931,00.html.

[CERT 97] *UNIX Configuration Guidelines.* Available at: http://www.cert.org/tech_tips/unix_configuration_guidelines.html

[CERT 98-03] *Password Cracking Activity.* Available at: http://www.cert.org/incident_notes/IN-98.03.html

[CERT 00] *Installing, configuring, and using swatch 2.2 to analyze log messages on systems running Solaris 2.x.* Available at: http://www.cert.org/security-improvement/implementations/i042.01.html

[CERT 00a] *Installing, configuring and using lsof 4.50 to list open files on systems running Solaris 2.x.* Available at: http://www.cert.org/security-improvement/implementations/i042.05.html

[CERT 00b] *Installing, configuring, and using tcp wrapper to log unauthorized connection attempts on systems running Solaris 2.x.* Available at: http://www.cert.org/security-improvement/implementations/i041.07.html

[CERT 01] *CERT Security Improvement Modules*, December 2001. Available at: http://www.cert.org/security-improvement/#unix

[CERT 01a] *Securing Public Web Servers.* April 25, 2001. Available at: http://www.cert.org/security-improvement/modules/m11.html.

[CERT 01b] *Installing, Configuring, and Using Logsurfer 1.5 To Analyze Log Messages on Systems Running Solaris 2.x* Available at: http://www.cert.org/security-improvement/implementations/i042.02.html

[CMU 01] *Secure Continuous Biometric-Enhanced Authentication* October, 2001. Available at: http://www.pdl.cmu.edu/Secure/biometric_auth.html

[CNET 00] *Microsoft eyes new security for Windows*, Stephanie Miles, May 20, 2000. Available at: http://news.com.com/2100-1040-239960.html?legacy=cnet

[Computer 01] *A Practical Guide to Biometric Security Technology*. IEEE Computer Society. Available at: http://www.computer.org/itpro/homepage/jan_feb01/security3c.htm

[CTSSN 01] *Lesson Six: File Permissions* Available at: http://www.ctssn.com/linux/lesson6.html

[EFF 99] *EFF "Intel v. Schwartz" Archive* Available at: http://www.eff.org/Cases/Intel_v_Schwartz.

[Flipchip] *VXA External Tape Drive and Dantz Retrospect Backup Software Dual Review* Available at: http://www.flipchip.net/accy_reviews/vxatape_dantzretro_pg1.htm

[Geosoft 02] *Computer Time Synchronization; A Beginner's Guide to Network Time Protocol (NTP).* Available at: http://geodsoft.com/howto/timesync/wininstall.htm

[IANA] *Assigned Port Numbers* Available at: http://www.iana.org/assignments/port-numbers

[Hernberg 00] *User Authentication HOWTO* Peter Hernberg, May, 2000. Available at: http://www.linuxdoc.org/HOWTO/User-Authentication-HOWTO/

[IETF 81] *Internet Control Message Protocol.* Available at: http://www.ietf.org/rfc/rfc0792.txt

[IETF 92] *Network Time Protocol (Version 3) Specification, Implementation and Analysis*. Available at: http://www.ietf.org/rfc/rfc1305.txt

[Insecure] *Nmap Free Security Scanner.* Available at: http://www.insecure.org/nmap

[Jecto 01] *Bestcrypt*. Available at: http://www.jetico.com

[Laubenbacher] "The Caesar Cipher", part of *The Complete Resource for the Science of Cryptography*, Reinhard Laubenbacher. Available at: http://www.math.nmsu.edu/~crypto/Caesar.html

[LinuxJournal 98] *Issue 51: Encrypted File Systems.* July 1, 1998. Available at: http://www2.linuxjournal.com/lj-issues/issue51/2590.html

[LinuxSec] *Firewalls.* Available at: http://www.linux-sec.net/Firewall

[Lucy 00] *How to find and use the Unix truss command*. Available at: http://www.pugcentral.org/howto/truss.htm

[Microsoft 97] *How to Enable Strong Password Functionality in Windows NT (Q161990)*. Available at: http://support.microsoft.com/default.aspx?scid=kb;en-us;Q161990.

[Microsoft 00] *Step-By-Step to Guide to Encrypting File System (EFS)*, March 7, 2000. Available at: http://www.microsoft.com/windows2000/techinfo/planning/security/efssteps.asp

[Microsoft 00a] *Enabling Strong Password Functionality in Windows 2000 (Q225230).* January, 2000. Available at: http://support.microsoft.com/default.aspx?scid=kb;EN-US;q225230

[Microsoft 01a] *Driver Signing / File Protection.* Available at: http://www.microsoft.com/hwdev/driver/drvsign.asp

[Microsoft 01b] *CryptoAPI, Version 2.0.* Available at: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/security/portalapi_3351.asp

[Mills 02] *Time Synchronization Server.* Available at http://www.ntp.org

[MGM 02] *Windtalkers* Available at: http://www.mgm.com/windtalkers/html/frm_main.html

[Morgan 01] *LINUX-PAM*. May, 2001. Available at: http://www.kernel.org/pub/linux/libs/pam/

[NIST 01] *AES – Advanced Encryption Standard*. December, 2001. Available at: http://csrc.nist.gov/encryption/aes/

[NSA 01] National Security Agency; *60 Minute Network Security Guide*, Version: 1.0 October 16, 2001. Available at: http://nsa1.www.conxion.com/support/guides/sd-7.pdf

[NSA 01a] *Windows 2000 Network Architecture Guide* Version 1.0. April 19, 2001. Available at: http://nsa2.www.conxion.com/win2k/guides/w2k-1.pdf

[NSA 01b] *Guide to Securing Windows 2000 Group Policy* Version 1.1. September 21, 2001. Available at: http://nsa2.www.conxion.com/win2k/guides/w2k-2.pdf

[NSA 01c] *Guide to Securing Microsoft Windows 2000 Group Policy: Security Configuration Tool Set* May 17, 2001. Version 1.0. Available at: http://nsa2.www.conxion.com/win2k/guides/w2k-3.pdf

[NSA 01d] *Group Policy Reference.* Version 1.0.8. March 2, 2001. Available at: http://nsa2.www.conxion.com/win2k/guides/w2k-4.pdf

[NSA 01e] *Guide to Securing Microsoft Windows 2000 Active Directory* Version 1.0. December, 2000. Available at: http://nsa2.www.conxion.com/win2k/guides/w2k-5.pdf

[NSA01f] *Guide to Securing Microsoft Windows 2000 DNS* Version 1.0, April 9, 2001. Available at: http://nsa2.www.conxion.com/win2k/guides/w2k-6.pdf

[NSA01g] *Guide to Securing Microsoft Windows 2000 Encrypting File System* Version 1.0. January, 2001. Available at: http://nsa2.www.conxion.com/win2k/guides/w2k-7.pdf

[NSA 01h] *Guide to Securing Microsoft Windows 2000 File and Disk Resources* Version 1.0, April 19, 2001. Available at: http://nsa2.www.conxion.com/win2k/guides/w2k-8.pdf

[NSA 01i] *Guide to Securing Windows 2000 Schema* Version 1.0. March 6, 2001. Available at: http://nsa2.www.conxion.com/win2k/guides/w2k-9.pdf

[NSA 01j] *Guide to Securing Windows NT/9X Clients in a Windows 2000 Network* Version 1.0.2. January 23, 2001. Available at: http://nsa2.www.conxion.com/win2k/guides/w2k-10.pdf

[NSA 01k] *Guide to the Secure Configuration and Administration of Microsoft ISA Server 2000* Version 1.4. July 18, 2001. Available at: http://nsa2.www.conxion.com/win2k/guides/w2k-11.pdf

[NSA01l] *Guide to the Secure Configuration and Administration of Microsoft Windows 2000 Certificate Services* Version 2.1. July 26, 2001. Available at http://nsa2.www.conxion.com/win2k/guides/w2k-12.pdf

[NSA 01m] *Guide to the Secure Configuration and Administration of Microsoft Windows 2000 Certificate Services, Checklist Format* Version 2.0.1. July 3, 2001. Available at: http://nsa2.www.conxion.com/win2k/guides/w2k-13.pdf

[NSA 01n] *Guide to the Secure Configuration and Administration of Microsoft Internet Information Services 5.0* Version 1.2. August 20, 2001. Available at: http://nsa2.www.conxion.com/win2k/guides/w2k-14.pdf

[NSA 01o] *Guide to Using DoD PKI Certificates in Outlook 2000* Version 1.0 April 6, 2001. Available at: http://nsa2.www.conxion.com/win2k/guides/w2k-15.pdf

[NSA 01p] *Guide to Windows 2000 Kerberos Settings* Version 1.1. June 27, 2001. Available at: http://nsa2.www.conxion.com/win2k/guides/w2k-16.pdf

[NSA 01q] *Microsoft Windows 2000 Router Configuration Guide* Version 1.02. May 1, 2001. Available at: http://nsa2.www.conxion.com/win2k/guides/w2k-17.pdf

[NSA 01r]*Guide to Securing Microsoft Windows 2000 DHCP* Version 1.2. June 25, 2001. Available at: http://nsa2.www.conxion.com/win2k/guides/w2k-18.pdf

[NSA 01s] *Guide to Securing Microsoft Windows 2000 Terminal Services* Version 1.0. July 2, 2001. Available at: http://nsa2.www.conxion.com/win2k/guides/w2k-19.pdf

Used Throughout] Microsoft Security and Technet. Available at: http://www.microsoft.com/security

[NST 94] National Security Telecommunications and Information Systems Security Committee. *NSTISSI 4011: National Training Standard for Information Systems Security Professionals*. Available at: http://www.nstissc.gov/Assets/pdf/4011.pdf

[RAND 01] *Can Biometrics Help the Army Solve An Identity Crisis?* Available at: http://www.rand.org/publications/RB/RB3024

[Reiser] Available at: http://www.reiserfs.org

[SANS 99] *Solaris Security Step by Step* Version 1.0. Available at: http://www.sansstore.org

[SANS 01a] *Windows 2000 Security Step by Step*, Version 1.0c, May 20, 2001. Available at: http://www.sansstore.org

[SANS 01b] *Securing Linux: Step-by-Step:* Version 1.1. Available at: http://www.sansstore.org

[SANS 01c] *TCP WRAPPERS—What are they??* Available at: http://rr.sans.org/unix/TCP_wrappers2.php

[SmartLine] *Freeware For Windows.* Available at: http://www.ntutility.com/freeware.html

[SUN 01] *FixModes Script.* Available at: http://www.sun.com/blueprints/tools

[SUN 02] *Pluggable Authentication Modules.*Available at: http://www.sun.com/solaris/pam/

[Sysinternals 02] *Utilities for Windows NT and Windows 2000*. Available at: http://www.sysinternals.com/ntw2k/utilities.shtml

[Tripwire] *How Tripwire for Servers Works.* Available at: http://www.tripwire.com/products/servers/how_works.cfm

[TropSoft 01] *Des Encryption.* Tropical Software. Available at: http://www.tropsoft.com/strongenc/des.html

[Tzeck 99] *Encrypting your Disks with Linux*. Doobee R. Tzeck. October 25, 1999. Available at: http://koeln.ccc.de/~drt/crypto/linux-disk.html

[Tzeck 99a] *pppd: Practical Privacy Disk Driver*. Doobee R. Tzeck. October 25, 1999. *Available* at: http://koeln.ccc.de/~drt/crypto/linux-disk.html#ppdd

[WinNTMag 99] *Inside Encrypting File System, Part 1.* Available at: http://www.win2000mag.com/Articles/Index.cfm?ArticleID=5387&pg=1

[WinNTMag 00] *Windows 2000 EFS.* Available at: http://www.winntmag.com/Articles/Index.cfm?ArticleID=7977

[XINETD] *XINETD* Available at: http://www.xinetd.org

[Yuriev 96] *Cryptographic File System under Linux HOW-TO*. Alexander O. Yuriev , March 14, 1996. Available at: http://www.ibiblio.org/pub/Linux/docs/faqs/security/Cryptographic-File-System

[ZDNET 00] *Enable Strong Passwords in Windows 2000.* ZDNet staff. September 13, 2000. Available at: http://www.zdnetindia.com/help/howto/stories/1131.html