

# Windows NT Security Guidelines

*Considerations & Guidelines for Securely Configuring  
Windows NT in Multiple Environments*

*A study for*  
**NSA Research**

*by*



**Trusted Systems Services**

*ntguide@trustedsystems.com*

*http://www.trustedsystems.com*

*217-344-0996*

3 June 1999

*Author:*        **Steve Sutton**        *Trusted Systems Services*

*Sponsor:*      **Scott Cothrell**      *National Security Agency*

©1998 Trusted Systems Services, Inc. All rights reserved. The U.S. Government has unlimited usage license under 1995 DFARS 252.227-7013. This document was prepared wholly under contract to the National Security Agency (MDA904-97-C-0336) and has been approved for public release.

**UNCLASSIFIED**

## Table of Contents

New in this Release (3 June 1999) .....	vii
<b>1. Introduction .....</b>	<b>1</b>
Scope & Intent .....	1
Level 1 & Level 2 .....	2
Structure .....	2
Notes & Terminology .....	3
AdvancedChecker Software .....	3
Acknowledgments .....	4
<b>2. Overview of the Guidelines .....</b>	<b>5</b>
<b>3. Installation .....</b>	<b>9</b>
Guidelines .....	9
Disable Unused Hardware .....	9
Physical Protection .....	9
Using Other Operating Systems to Install Windows NT .....	9
Booting from Alternative Media .....	9
Installing Alternative Operating Systems .....	9
NTFS File System Format .....	10
Removing the POSIX and OS/2 Subsystems .....	10
Do Not “Copy Install” .....	10
Notes .....	11
Booting Alternative Operating Systems .....	11
Physical Protection .....	11
Multiple Copies of Windows NT on One Computer .....	12
<b>4. Domains &amp; Basic Access Restrictions .....</b>	<b>13</b>
Guidelines .....	13
Notes .....	14
Domains, Trusts & the Scope of Accounts .....	14
Accounts & Network Authentication .....	15
Domain Models .....	16
Logon Rights in Multidomain Environments .....	16
<b>5. Administrative Structure .....</b>	<b>19</b>
Guidelines .....	19
The “Administrator” Account .....	19
Full Administrators .....	20
Domain Operators & Power Users .....	21
Administrative Practices .....	22
Notes .....	23
Shared Administrative Accounts .....	23
The PASSPROP Utility .....	23
Renaming the Administrator Account .....	23
<b>6. General Policies .....</b>	<b>25</b>
Guidelines & Notes .....	25
Raw Devices & Non-NTFS Volume ACLs .....	25
Restricting Access to Floppies and CDROMs .....	25
Unauthenticated (“Anonymous”) Remote Access to the Registry .....	26

Restricting Who can Access the Registry Remotely .....	26
Enabling the Registry Editors .....	27
ProtectionMode .....	28
Unauthenticated Event Log Viewing.....	29
Print Driver Installation .....	29
Screen Saver Locking.....	29
Protecting Hashed Passwords & SYSKEY.....	30
Password Notification Feature.....	31
User & Share Names Available to Unauthenticated Users.....	32
Hiding the Last User Logon .....	32
Shutting Down the System .....	32
Miscellaneous Hot-Fixes .....	33
The C2CONFIG Tool.....	33
<b>7. File System &amp; Registry ACL Settings .....</b>	<b>35</b>
Guidelines .....	36
Notes.....	36
File System ACL Settings .....	36
Registry ACL Settings.....	43
Installing & Testing New Applications .....	47
A Modest Change to Tighten WINNT ACL's .....	47
Limiting Installers in the WINNT & Registry ACL Settings .....	48
Tightening Service Pack ACL's .....	49
Software Installation that "Moves" New Items into Protected Directories .....	49
<b>8. Application &amp; User Home Directories .....</b>	<b>51</b>
Guidelines .....	51
Application Directories.....	51
User Home Directories .....	52
<b>9. User Accounts &amp; Groups.....</b>	<b>55</b>
Guidelines .....	55
User Accounts.....	55
User Groups.....	56
Notes.....	56
<b>10. Passwords.....</b>	<b>59</b>
Guidelines .....	59
Password Complexity and Lifetime.....	59
Password Locking.....	59
Guidelines for Users who Define their own Passwords.....	59
Administratively Defined Passwords.....	60
Password Filtering .....	60
Password Warning Time.....	60
Notes.....	61
Logon Attempt Attacks.....	61
Captured Password Attacks .....	62
Example Policy A.....	64
Example Policy B .....	64
A Caveat on Network Password Exposure .....	64
Passwords for Local, Matching Accounts .....	65
Password Filtering .....	65
Summary.....	65

<b>11. System Policy Files .....</b>	<b>67</b>
Guidelines.....	67
Notes.....	68
Recommended Default User Policies for non-Administrative Users .....	68
Recommended Default Computer Policies.....	69
Compelling Use of Policy Files.....	69
Automatic versus Manual Update Mode .....	70
User Application Restrictions.....	70
Protected User Policies.....	71
Custom Policies.....	71
<b>12. User Rights.....</b>	<b>73</b>
Guidelines.....	73
Notes.....	74
Common Rights.....	74
<b>13. Auditing Policy &amp; the Security Log .....</b>	<b>77</b>
Guidelines.....	77
Notes.....	78
Object Auditing Always Records SAM Objects .....	78
Rights not Audited.....	78
Auditing “Base Objects” .....	79
Crashing when the Security Log Fills .....	79
Alternative Locations for the Security Log .....	79
Right to Manage the Audit Trail.....	79
Object Auditing .....	80
<b>14. System Services.....</b>	<b>81</b>
Guidelines.....	81
Minimizing Services & Their Capabilities .....	81
Restricting Operator Control of Services .....	82
Notes.....	83
Unprivileged Service Account.....	83
Minimizing Services & Stolen Service Password .....	83
<b>15. Network Sharing.....</b>	<b>85</b>
Guidelines.....	85
Network Share Directories .....	85
Printer Access.....	85
Notes.....	86
Summary of Sharing Mechanisms.....	86
Hidden, Administrative Shares.....	86
<b>16. Networking.....</b>	<b>89</b>
Guidelines.....	89
Notes.....	91
Unencrypted Passwords on the Network.....	91
SMB Signing .....	91
LANMAN Passwords.....	92
Service Attacks.....	92
Network Eavesdropping & Interception.....	93
Apply Cryptography to All Network Traffic.....	94
Isolating Native Windows NT Service from an Intranet .....	94
IP Spoofing.....	95
TCP/IP Port Limitations .....	95

	The Security of Windows NT's Protocols .....	95
<b>17.</b>	<b>Remote Access Service (RAS) .....</b>	<b>97</b>
	Guidelines .....	97
	Notes .....	98
	General Discussion .....	98
	Strong User Passwords .....	99
	RAS Sentry .....	99
<b>18.</b>	<b>Spoofing .....</b>	<b>101</b>
	Guidelines & Notes .....	101
	Logon Separation .....	101
	Trusted Path ("Security Window," "Secure Attention Sequence") .....	101
	System-Wide "PATH" and Other Environment Variables .....	102
	The "." Issue .....	103
	Data Files that Hold Hidden Programs .....	103
	CDROM Auto-run Programs .....	104
	Shortcut Spoofing .....	104
	Protecting Standard Extensions .....	104
	Defining Standard Extensions .....	105
	Removing "R" from Program Files .....	105
	Internet Browsers .....	105
	DLL Spoofing .....	105
<b>19.</b>	<b>User Responsibilities &amp; Practices .....</b>	<b>109</b>
<b>20.</b>	<b>References .....</b>	<b>111</b>
<b>21.</b>	<b>Reference Material .....</b>	<b>113</b>

## New in this Release (3 June 1999)

It's been over a year since we first released these Guidelines. They have proven enormously popular and have been downloaded by over 40,000 Windows NT administrators. Surprisingly, not much has happened in that year that warranted any significant change in the Guidelines. We've made number of minor corrections to the text (too small to be even noted), and added the "Addenda" that heretofore was on our Web site. All such significant additions are marked by the double revision bars in the right margin, like on this paragraph.

Of course there are the usual "bug-of-the-month" revelations, but our advice is still to follow the advice on the Microsoft security page, which does a good job of notifying the world of potentially dangerous bugs. These are ephemeral and as before we see no point in including them in the Guidelines.

Two aspects on the Windows NT security front have impressed us over the last year. First, more and more administrators are becomming Windows NT security literate. For example, we've not seen a single security magazine article in the last year that, in our humble opinion, was not aimed below most Windows NT administrators. There just aren't enough new topics! Second, enterprise administrators don't so much need advice as they need power tools, preferable large-scale and programmable, which leads us to...

Perhaps the biggest news from Trusted Systems is that it has now released AdvancedChecker, a complete rework and major expansion of the "Checker Prototype" funded by the same NSA grant that sponsored the Guidelines. **AdvancedChecker** is a full-fledged scripting language that can set and check, network-wide, virtually all Windows NT security parameters. It is a powerful tool for professional, enterprise Windows NT administrators. You can read more about AdvancedChecker a few pages below, or check our Web site:

<http://www.TrustedSystems.com>



---

**AdvancedChecker**





# 1. Introduction

---

This research into securing the Microsoft's Windows NT™ operating system was conducted by Trusted Systems Services, Inc., under contract to the National Security Agency (MDA904-97-C-0336). The goal was to capture the state-of-the art in securely configuring Windows NT Server and Workstation 4.0 based on extensive review of published works, and to offer guidance for both government and commercial users. This contract also included the development of a programmable software tool code-named "Checker" to check and enforce specific security policies.<sup>1</sup> See "Checker Software," below for a detailed description of this software prototype.

## Scope & Intent

These guidelines describe practices that counter common, known attacks on Windows NT network installations that expose or modify user data maliciously. The goal is to make Windows NT as secure as it can *reasonably and practically* be configured. We believe that these guidelines reduce security risks to a level on par with the most aggressive current efforts. The same set of threats exists in governmental and commercial environments and the techniques for containing them are the same. Hence, the guidelines are applicable to almost any Windows NT environment. The guidelines are a result of extensive review of current published works on securing Windows NT and are therefore consistent with or at least cognizant of several previous, major efforts, most notably [Sutt96], [Maye96], [Micr97], [TFM], and [Navy97]. (The items in "[...]" brackets refer to documents defined in the *References* section at the end of the document.)

Windows NT has many controls for tightening its security. However, even in the most secure mode that these guidelines address, they do not blindly recommend the tightest settings for all controls. Implicit in the guidelines is the understanding that its recommendations must be both effective against certain threats and also practical. Some controls impede operational capability and their use must be carefully balanced against the security they offer.

Security against active penetrations is a weak link phenomenon. One philosophy is that there is little point protecting against minor security risks when other, much larger risks remain. Under this philosophy, one brings all risks to roughly the same level by tightening the larger risks, leaving the minor ones unchanged. Another philosophy reduces *all* risks to their lowest values. Attempting to balance these two extremes, the guidelines prescribe controls that have the most dramatic effect on reducing overall system risk and leave the closing of smaller risks as optional. Ultimately the controls you implement depend on the risks present at your site, and you should implement even minor controls if they counter a legitimate threat.

Some guidelines are straightforward, while others require considerable judgment by their implementers, and for the latter we present a brief discussion of the salient criteria involved. Although this document includes some introductory topics, it is not a tutorial or administrative manual. We assume the administrators that implement these guidelines are familiar with the administrative manuals that accompany Windows NT and proficient in managing its security features. The *References* section at the end includes some other tutorial readings.

---

<sup>1</sup> Check the Trusted Systems Services Web site for announcements about the availability of Checker (<http://www.TrustedSystems.com>).

Windows NT has been rated as C2 under the U.S. *Trusted Computer Systems Evaluation Criteria* (TCSEC, or “Orange Book”) and comparably rated under the similar European guideline called the ITSEC. These ratings lend confidence that Windows NT’s basic architecture is sound and its features responsibly implemented. Unfortunately these criteria do not address configuring and using it securely.

Finally, these guidelines do not constitute NSA policy. They are presented as state-of-the-art advice on configuring Windows NT securely, and are open to interpretation and modification to fit the threats and policies of a particular site. These guidelines present an attempt at documenting “best commercial practice” for configuring Windows NT securely.

We view these guidelines as an active document, and readily welcome discussion and feedback from its readers. There are undoubtedly many areas that can be improved through this process. Send comments to [ntguide@trustedsystems.com](mailto:ntguide@trustedsystems.com).

## Level 1 & Level 2

We define two levels of security, Levels 1 and 2, where Level 2 is more secure than Level 1:

**Level 1** is a modest enhancement over a standard Windows NT installation. Virtually all sites that deem security important would likely want to implement Level 1.

**Level 2** applies to sites with a considerable commitment to security – those who wish to maximize the protection that Windows NT affords.

Implementing Level 2, with all its recommendations and options, places you among the most aggressive efforts to configure Windows NT securely. However, Level 2 requires *considerably* more effort to install and maintain than Level 1, and should not be undertaken lightly.

☞ Practices prescribed for a given level are simply stated, or noted as “**prescribed.**” Other practices are “**recommended,**” and it is implicit that they are *highly recommended* at Level 2.

In practice, few sites will be strictly Level 1 or 2, but will intermix practices to suit their situation. Our goal is **not** to establish a rating criteria. Levels 1 and 2 are working designations that we intend no one to officially bestow. For example, if you omit a few recommended Level 2 protections while implementing the rest, we do not intend that your system be labeled by someone as “not Level 2.”

## Structure

Each guideline holds one or more *Guideline* sections that succinctly list the recommended actions. A *Notes* section that usually follows includes rationale, background, and more detailed descriptions of techniques. A reference section at the end of each chapter notes related guidelines and other references.

Implicit in all guidelines is that they should be regularly reviewed for general conformance. *Periodic Review* notes (where present) inside the *Guidelines* list only special or detailed suggestions for these reviews. While we do not prescribe specific review periods, Level 1 sites would typically review every 3-6 months, and Level 2 every 1-2 months.

## Notes & Terminology

Most of the guidelines can be fulfilled with tools delivered with Windows NT. Some suggested practices require 3<sup>rd</sup> party tools, although we mention only a few. Where we do note these tools, it does not constitute an endorsement, nor do we imply that they are the only such tools for a particular purpose or even the best. Rather, our general theory is that if our readers know the name of one tool they can search the ‘net to find similar ones.

The guidelines do not address “denial of service” issues. Whether or not one considers denial-of-service as a “security” issue, its solutions better come from sciences other than security. We also don’t prescribe regular backup and restore procedures, although they can be critical in recovering from a penetration.

The guidelines cover Windows NT 4.0 through and including Service Pack 5, although a few of the items we note are fixed in the later service packs.

We use the uncapitalized term “administrator” generically.

## AdvancedChecker Software

This contract also included the development of prototype software tool code-named “Checker” to check and enforce Windows NT security policies. In the year since the original “Checker Prototype” produced on the project was finished, Trusted Systems has completely revamped and enormously expanded the prototype into its “Advanced Checker” software line.



**AdvancedChecker**

AdvancedChecker is a scripting language that lets one program scripts to check, set, and enforce network-wide Windows NT security and intrusion detection policies. It provides power and flexibility in a comprehensive, easy-to-use package. AdvancedChecker scripts can be quite simple. For example, the script that checks that auditing is enabled (or the Guest account disabled) for all computers in a specified domain – and sends E-mail announcing which ones are not – is just a few lines. Because AdvancedChecker is a flexible and powerful language with user-defined variables, parameterized functions, and a host of flexible control statements, administrators can readily create advanced tools that do precisely what they need them to do – customized to their own unique security and auditing requirements. AdvancedChecker can be programmed to implement all the recommendations in these Guidelines, and many are included in its sample scripting library.

AdvancedChecker’s security strength is its stable of over 270 built-in functions that work as easily over an entire Windows NT network as on a single host. Its security capabilities include:

- Check and set Audit, Rights, and Accounts Policies, and User Accounts and Groups.
- Fetch, set, and check ACL’s and ownership of files, directories, the Registry, and share directories – as well as Audit SACL’s.
- Read, process, and control Windows NT’s Security Log.
- Read, write, and manage files, directories, and Registry keys.
- Run any command line program and capture its output.
- Create cryptographic digital signatures to assure data has not been tampered.

- Manage Windows NT Services.
- Notify your administrators by E-mail.
- Traverse entire file system and Registry trees (“tree walking”) checking its items’ ACL’s, integrity, and so forth.

AdvancedChecker includes “Checker Editor,” an advanced text editor that lets you create Checker scripts and run them from inside the Editor. The Editor includes: color coded language elements; syntax-check and runs from within the editor; “project” support; file comparison with color-coded differences; multi-file find; auto-indent; highlighting of matching brackets; user-defined, auto-record editing macros.

AdvancedChecker uses Windows NT native security services and remote authentication. This means Checker programs are controlled and audited as strongly as Windows NT’s own network management functions. One need not install AdvancedChecker on each computer it manages – one copy installed on a central host can manage all the computers on a network. You can read more about AdvancedChecker at Web site:

<http://www.TrustedSystems.com>

## Acknowledgments

Many thanks to our NSA reviewers: Scott Cothrell, Chris Shutters, Tom Goss, and Jack Lehman. Thanks also to Alan Ramsbottom of ALS International Ltd. and Paul Ashton of Eigen Solutions for their thorough and insightful review of this document, and their many suggestions.

## 2. Overview of the Guidelines

---

Following is an overall summary of the guideline chapters:

### 3. Installation

This short but important guideline addresses hardware issues and the basic installation process.

### 4. Domains & Basic Access Restrictions

This general guideline addresses how to use the three fundamental features that determine which users can log onto and remotely access other computers on the network:

- **Domains & Trusts:** The domain definition and trust relationships fundamentally determine network user access. “Matching” accounts can be used to keep the trust relationships simple.
- **Logon Rights:** The Right to log on locally and the Right to logon from remote locations in each computer’s Rights policy protect that computer.
- **Per-account Restrictions:** Each account can have a list of computers that limits where that user can locally log on, although it does not limit remote access.

Designing how these features work in combination is fundamental in securing a Windows NT network.

### 5. Administrative Structure

This chapter address the configuration and use of administrative and operator accounts. It recommends no major reconfiguration of the standard Windows NT accounts, but offers guidelines on the scope and assignment of accounts to administrative personnel.

- **The “Administrator” Account:** Each Windows NT computer has a built-in account initially named “Administrator” that is all-powerful and cannot be deleted. Upon installation, it is the only such administrative account. The guidelines advocate using this account as a maintenance account of last resort, relying instead on domain-wide administrative accounts (following) when full administrative capabilities are required.
- **Full Administrators:** This guideline addresses other all-powerful, “full” administrative accounts – accounts that are members of the local “Administrators” group, and the domain-wide “Domain Admins” group. This guideline generally follows standard Windows NT practice. Level 2 separates workstation administrators from those who administer more critical domain controllers and major network servers. This section also presents security guidelines for using administrative accounts.
- **Domain Operators & Power Users:** This guideline advocates extensive use of standard Windows NT domain controller “operator” groups (including the workstation Power User group) to relieve day-to-day use of full administrative accounts. It does not recommend any significant changes to the capabilities of these groups. It presents options for a Domain Power User group for domain-wide workstation administration, and for separating the critical backup and restore roles.

## 6. General Policies

This guideline holds a lengthy collection of miscellaneous but important Windows NT security controls. While somewhat obscure and detailed, they pose important decisions you should make early in your security implementation.

## 7. File System & Registry ACL Settings

This guideline presents a strategy for tightening the Access Control Lists (ACLs) on critical system objects, mainly those in the system root directory (usually C:\WINNT) which holds most of the sensitive files in Window NT, and the Windows NT Registry. As a concession to ease-of-use and software compatibility, by default these ACLs are not as tight as they could be.

In general, several areas that allow all users the ability to create new objects or modify as-delivered objects are replaced with entries that allow them only read access. A new "App Installers" group is instead given the ability to add or modify these objects. Members of the App Installers are those users entrusted to approve and install new applications on the system, which is the major reason that users need write access to these areas. Although the ACLs are identical for Levels 1 and 2, the trust criteria for membership in App Users is substantially higher at Level 2.

## 8. Application & User Home Directories

This guideline presents a standard technique for setting up common application (program) directories to make these critical system components resistant to attacks like viruses. It also advocates removing access to such programs from critical administrators unless the programs are thoroughly trustworthy. This guideline also describes a common technique for setting ACLs on user home directories, including directories to be shared by several users or groups.

## 9. User Accounts & Groups

This guideline addresses the few considerations for the account parameters other than passwords. For example, at Level 2 this guideline advocates aggressive use of the account restrictions as to which computers the account can locally log on. This guideline also covers user groups other than administrative groups and common groups like Users and Domain Users.

## 10. Passwords

This guideline prescribes the full and aggressive use of account locking and other password parameters in the Account Policy. It also presents several common password schemes and classifies them according to the probability of successful attack based on various Windows NT password control parameters. Ultimately each site must tailor its password policy to its own risks. This guideline offers recommendations toward this end.

## 11. System Policy Files

System Policy Files are a Windows NT (and Windows 95) feature that lets administrators centrally control the basic appearance of the user's desktop environment. This includes various aspects of their start menu, the items on their desktop, and whether or not the system presents a "for official use only" window during logon. Administrators can set up central policies that apply to different groups of users and workstations. At Level 2, the guidelines recommend setting up a simple, basic policy file, even though relatively few system security policies are of significant security strength.

## 12. User Rights

Each Windows NT computer has an administratively controlled Rights policy that assigns combinations of about 30 “Rights” to various users and groups who access that computer. For example, the ability to set the system time and date is a Right. As installed, the Windows NT Rights policy is prudent. This guideline recommends a few small changes to enhance security.

## 13. Auditing Policy & the Security Log

The Windows NT security log can collect a variety of detailed, security relevant events into managed collection files, and administrators have considerable latitude over which events are saved. This guideline suggests which basic categories to record at each Level and how to manage Windows NT’s security log. This is a point-of-departure open to considerable site interpretation.

## 14. System Services

Windows NT system services are important components. Services are (usually powerful) programs that run largely unseen performing various services for user programs or remote elements on the network. This guideline presents cautions for eliminating unnecessary services as well as suggestions for running services under accounts safer than the customary, all-powerful SYSTEM account.

## 15. Network Sharing

This brief guideline contains recommendations for creating network share directories and printers, including comments on “hidden administrative shares.”

## 16. Networking

Many of these guidelines address security issues in Windows NT’s domain-base networking environment. This section gives basic advice on minimizing network services, removing hostile elements from a Windows NT network, isolating Windows NT’s native sharing services from an intranet, and general guidelines on where firewalls and encryption may be needed.

## 17. Remote Access Service (RAS)

RAS is a native Windows NT service that lets computers log onto remote networks through a Windows NT RAS server. This access is via a telephone line or, (using the companion PPTP protocols) an intranet. This guideline addresses setting the relatively few RAS security parameters for sites whose policy allows remote access.

## 18. Spoofing

Spoofing is where a malicious user attempts to lure an unsuspecting user into running a malicious program that the first user created. If successful, the malicious program runs with the full capabilities of the duped user and can cause widespread damage if that user is an administrator. Spoofing is perhaps the most pernicious threat in operating systems. Unfortunately it is also the most difficult to combat because countermeasures tend to be non-specific. This guideline presents several spoofing threats and counsels on how to minimize them.

## 19. User Responsibilities & Practices

This guideline presents basic practices that all regular users should understand and use. It recommends that administrators develop a site policy of such practices and impart the policy to their system users.

**UNCLASSIFIED**



## 3. Installation

---

These guidelines apply mainly to the initial installation of a Windows NT system.

### Guidelines

#### Levels 1 & 2:

#### *Disable Unused Hardware*

- ❑ Remove hardware components that you view a security risk (including COM or LPT ports), or disable them from the computers BIOS (provided you can also assign a BIOS password). (See also “Restricting Access to Floppies and CDRoms” in the *General Policies* guideline.)

#### *Physical Protection*

- ❑ It is difficult for these guidelines to recommend precise physical protections for the computer hardware. To the extent practical, you should implement the practices in “Physical Protection,” below.
- ❑ We *recommend* that removable drives with the NTFS file format be physically locked so that untrusted personnel cannot remove them. (See “Physical Protection” in the Notes that follow.)

#### *Using Other Operating Systems to Install Windows NT*

- ❑ We *suggest* that you do not use other operating systems like DOS or Windows 95 to install Windows NT because they can create files on the system not normally intended for the more secure Windows NT environment (although the risk is small). Assure that all other operating systems are purged from the computer before installing Windows NT, or reformat the hard drives during installation. If this is not possible, we *recommend* you remove all files and directories of the non-NT operating system not installed by Windows NT.

#### *Booting from Alternative Media*

- ❑ Disable all means that can be used to boot alternative operating systems. See “Booting Alternative Operating Systems” in the Notes that follow. Unfortunately, you may already have hardware that does not allow you these protections, and if so, you should assess the risks carefully. The ease with which floppy boot schemes can be perpetrated makes them an enormous risk at both Levels 1 and 2.

#### *Installing Alternative Operating Systems*

- ❑ Do not install any other operating system (like DOS, Windows 95, or Linux) on a Windows NT computer.<sup>2</sup> See “Booting Alternative Operating Systems” in the Notes that follow.
- ❑ Do not install more than one copy of Windows NT on a computer unless operationally necessary.<sup>3</sup> Some administrators install a second, emergency copy of Windows NT

---

<sup>2</sup> There certainly are cases where alternative systems can be installed and used without compromising Windows NT, but you must carefully assess the particular situation.

Workstation that has no users except the local Administrator with a suitably secure password. This practice is acceptable at both Levels 1 and 2, providing you take the precautions in “Multiple Copies of Windows NT on one Computer” in the notes following.

### ***NTFS File System Format***

- ❑ We *recommend* you format all NTFS-capable volumes as NTFS – not FAT.<sup>4</sup> FAT file systems have no ACLs and inherently afford lesser protection of data. (The “read-only” bit on FAT files can be reset by anyone.) You might use FAT volumes so long as they do not hold the system root directory or any files or folders that would not be granted Full Control to Everyone if the file system had ACLs. Even so, use FAT only if there is a compelling operational reason to do so, and on Windows NT there seldom is. Note that Windows NT can reformat partitions during installation from floppy disks – you don’t have to format to NTFS beforehand.

### ***Removing the POSIX and OS/2 Subsystems***

- ❑ There is little available information on the trustworthiness of the OS/2 and POSIX subsystems, although there is no reason to suspect they pose an inordinate security risk.<sup>5</sup> We *recommend* you disable these subsystems unless you need them. Disable them by removing “Os2” and “Posix” from the multistring value named “Optional” in the Registry entry:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\
  Session Manager\Subsystems
```

### ***Do Not “Copy Install”***

- ❑ Contrary to recommended practice, some administrators install Windows NT by copying the entire system root directory and a few other files from one computer to another. Do not do this. Each installation of Windows NT receives a special system ID unique among all other such ID’s on the network, which makes its accounts and groups ID’s also unique. Copied installations do not have this uniqueness and this can negate certain security protections. (See [KBase] Q162001.) Note that there are commercial “copy install” programs that solve this problem.<sup>6</sup>

<sup>3</sup> Development systems often have a legitimate need to have more than one copy of NT installed, and development personnel often require enhanced access and privileges that you would not normally give to ordinary users. These systems should be segregated if possible.

<sup>4</sup> FAT refers to the Windows NT media format that does not allow ACLs on its files and directories, although there are cases where an entire FAT device can have an ACL.

<sup>5</sup> WOW (Windows on Win32) is the Windows NT subsystem that supports older, 16-bit Windows applications. In his review of this document, Paul Ashton points out that there are several security concerns with WOW. It would therefore be prudent to also disable WOW if you do not need it (that is, if you run only the newer, 32-bit applications). However, as with POSIX and OS/2, it is difficult to assess the degree of risk WOW presents.

<sup>6</sup> See <http://www.ntinternals.com/ntsid.htm> for a list of several.

## Notes

### *Booting Alternative Operating Systems*

Most computers, especially the Intel-based computers popular in the Windows NT community, readily boot an operating system (like DOS or Windows 95) from floppy disks, even when the computer has Windows NT installed. This can happen when the firmware in these computers checks floppy drives for a bootable media before the hard drive from whence Windows NT boots. You should prevent this at both Levels 1 and 2 because programs can be run on these other operating systems that read and write NTFS partitions, completely circumventing their ACL protection.

The setup firmware on most of these computers (commonly called the “BIOS” and commonly activated by special keys during startup) have options for either preventing booting from the floppy, or booting from the floppy only if no hard-disk boot image is available. Set one of these options (preferably the former) if your computer allows.

However, this in itself does not prevent anyone from resetting the option. Most computers also have a BIOS (firmware) password that guards access to the BIOS program and you should define such a password. Because guessing such a password requires entering it manually, this password need not be as strong as your Windows NT password. However, you should still not use simplistic passwords. On most such computers, you can bypass the password by making certain changes to the internal circuit boards of the computer, and some sites may elect to physically prevent users such access with, for example, cabinet locks.

Alternatively, you can remove floppy hardware or buy floppy padlocks that prevent their use. Some computers may allow booting from other forms of removable media, like CDROM, and the same precautions apply. Note that techniques for assigning an ACL to floppy drives (see “Restricting Access to Floppies and CDROMs” in *General Policies*) do not prevent them from being used as boot devices.

Some computers can boot from remote computers. Unless cryptographically protected, these are vulnerable to network threats on unsecured networks. (See *Networking*.)

### *Physical Protection*

There are several reasons why you may need to secure the computer physically to prevent unauthorized users access inside the cabinet: the BIOS password can often be circumvented by accessing the computer’s circuit boards, hardware removed for security reasons can be reinstalled, the hard drives can be removed and mounted in other systems where their data can be read or modified, and although a bit remote, a penetrator could insert certain malicious hardware inside the cabinet.

There is little you can do to prevent abuse of personal, removable media like floppy drives or writable CDROMs short of removing the drive, physically locking them, or placing ACLs on their internal device object. (See “Raw Devices & Non-NTFS Volume ACLs” in *General Policies*). Shared, removable drives, especially those formatted under NTFS, should be physically locked, since otherwise they can be mounted on other operating systems and their data read or changed, bypassing the NTFS ACL controls. This is usually not so important for removable devices formatted under FAT because there is no expectation of privacy on these drives. They present no more risk than floppy disks except that they usually hold more data. However, if you protect an entire FAT device with an ACL (see “Restricting Access to Floppies and CDROMs” in *General Policies*), there becomes an expectation by its user of privacy and unauthorized removal becomes a larger issue.

### *Multiple Copies of Windows NT on One Computer*

You must take certain precautions if you install more than one copy of Windows NT on a computer. (Note that a boot option and its corresponding “VGA Mode” option refer to a single copy.) Windows NT opens certain files exclusively when it starts up as a way to prevent untrusted programs from accessing them, most notably the files in WINNT\System32\CONFIG that store the Registry. These files may have ACLs that would otherwise allow such access, although these guidelines recommend protecting them more tightly. Such files are exposed on copies of Windows NT other than the one currently active.

There are cases where ACLs created by one of the copies are not protected properly when another copy is active. For example, suppose two co-resident copies of Windows NT recognize a domain account named “JJones.” The first copy has a local group named GroupX that includes JJones, and an ACL that denies access to GroupX, and hence JJones. This ACL does not deny JJones access when the second copy of Windows NT is active because GroupX appears as an “Unknown” group to the second copy.

However, the following “built-in” users and groups have the same identity on all Windows NT systems: Everyone, INTERACTIVE, NETWORK, SYSTEM, Administrators, Users, Guests, Power Users, Replicator, Server Operators, Print Operators, Backup Operators, Account Operators, CREATOR/OWNER, and Authorized Users. ACLs that hold only these identities are always properly enforced by all co-resident copies of Windows NT.

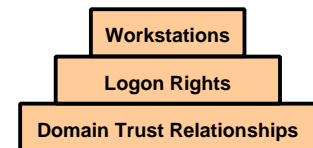
## 4. Domains & Basic Access Restrictions

---

The most fundamental aspect of designing a Windows NT network is to determine which user accounts can access which computers. The Windows NT security restrictions on this access is strong, fundamental, and not overridden by other Windows NT access policies. This aspect of network design is therefore one of the most important, in part because it is also one of the hardest to change later on.

By “access” we mean logon locally or remotely. A **local logon** is where the user sits at a computer and logs on to establish an interactive session. A **remote logon** is invisible to the user and occurs when they first access a network resource on another computer, where the secondary logon takes place. Before you install your Windows NT network, you must design a domain structure that optimally precludes user accounts from accessing computers to which they are not authorized. There are three fundamental aspects of this design:

- The basic **domain structure** of the network (domain membership and trust relationships), including the use of local, “matching” accounts to help keep the domain structure simple by accommodating special situations. Domains fundamentally and strongly determine which user accounts have access to which computers across the Windows NT network. Domains also isolate administrative control of the network. Administrators in one domain do not have administrative control of another domain without the express action of the administrator in the other domain. Hence, domains are a means by which an organization can hold ultimate control over its own computers.
- The Rights to **log on locally** and **remotely** in the Rights policy on each computer govern who can log onto that workstation. Because you must maintain this on each computer, it’s important that you develop your policy up-front (see *User Rights*).
- The **workstation logon restrictions** assigned to each account which govern local logon sites. Each domain account may have a list of workstations onto which the user is allowed primary logon. This can be usefully deployed, although it governs only local logon and is limited to 8 workstations. (See also *User Accounts & Groups*.)



There can be no specific guidelines for designing this structure. You must understand the principles of domains and matching accounts, and apply them to your installation. See the Notes below that expand each of these three topics, as well as the general references at the end of this chapter.

### Guidelines

#### Levels 1 & 2:

Carefully design your network-wide strategy for these three basic determinants of which users can use which computers, as discussed in the Notes that follow. While Level 1 involves a degree of planning, Level 2 warrants extensive site analysis. At Level 1, the focus is keeping untrusted users from sensitive information and central resources, like major domain controllers and major servers. At Level 2, all network data repositories should be of prime consideration. It is important to develop this design thoroughly *before* you install your network.

**Periodic Review:**

Review the domain structure and the use of the other mechanisms for restricting primary accounts. The audit should assess how well that these mechanisms enforce the following:

- ❑ No account should have access to a computer, that by site policy they are prohibited from accessing. (Levels 1 and 2)
- ❑ For level 2, you should reevaluate the account policies if more than roughly 10% of network-wide accounts have access to computers not directly required by the account's operational mission.

**Notes**

To underscore the importance of planning domains and basic account structure, we present several lengthy descriptions of these basic features. You can also consult the general references at the end of this guideline.

***Domains, Trusts & the Scope of Accounts***

Windows NT networks of any appreciable size are best subdivided into domains. A domain is a collection of member computers of one of two types: computers on which the product named "Windows NT Workstation" is installed in its domain member configuration, and computers on which the product named "Windows NT Server" has been installed in its "server" configuration. We refer to both as "workstations" even though the latter may function as a workgroup server. A workstation can only be a member of one domain, and typically remains in the same domain. Administrators control workstation domain membership.

Each domain has a **primary domain controller** that serves as the centralized user authentication controller of its domain and can establish "trust" relationships with the domain controllers in other domains. Each domain can have one or more **backup domain controllers** that share the primary domain controller's security authentication load, and one is promoted to primary when the primary fails. All domain controllers in a domain regularly communicate to assure each has a duplicate copy of essential authentication databases. We largely omit the backup domain controllers from our discussion because they add nothing to the security considerations for structuring Windows NT domains.

A domain controller holds a set of **domain accounts**. Each such domain account can be used to establish local or remote logon sessions on each computer in the domain (unless another control prevents it). Each computer in a domain, including domain controllers can also hold **local accounts**. A local account can be used to establish local or remote logon sessions only on the computer on which it is held.

The administrators from two domains can establish a trust relationship from one of the domains to the other. If a domain A trusts a domain B, then all of B's domain accounts may be used for local or remote logon to computers in A. However, A's accounts may not be used in B unless administrators establish a separate trust relationship where B trusts A. The phrase "A trusts B" means "the domain controller in A allows (trusts) B's account users to access, either remotely or locally, all computers in A (unless a secondary control prevents)." Note that if A trusts B, and B trusts C, it is *not* true that A trusts C. For this to occur, the administrators of A and C must set A to trust C directly.

By convention, most accounts on the networking domain environment are domain accounts. Local accounts are most frequently used to accommodate server-side, remote logon sessions for users whose client-side accounts are not in the server's scope.

The manner in which you subdivide your network into domains is the most fundamental control, and in many ways the strongest criteria for which users can use which computers, whether by local or remote logon. While there are many examples of domains in the popular literature, your overall goal is to use the domain structure to restrict users from all computers except those to which they are authorized to access.

There are two important techniques that help you keep your network domain trust relationships simple and "tight." The first uses local, **matching accounts**. Suppose you need to give a user in domain A access to a computer (whether workstation or domain controller) in domain D, but D does not trust A. Need you configure D to trust A to serve this one user? No. Instead establish a local account on the computer in D, giving it the same account name and a password known to that user. Note that this matching occurs only on the computer that stores the matching account, D in this case. (See "Passwords for Local, Matching Accounts" in *Passwords*.)

The second technique restricts an account from a computer that the domain structure would otherwise allow, and involves withholding the Rights to "log on locally" and/or "access this computer from the network" from that account on that computer. (See the "Logon Rights in Multidomain Environments" below.) There are also "Logon To" workstation restrictions in each domain account, but they govern only local (primary) logon and are limited to 8 workstations. (See *User Accounts & Groups*.)

### ***Accounts & Network Authentication***<sup>7</sup>

User identity, as represented by their user account, is the basis for access to all the capabilities and services of Windows NT based networks and their computers. With very few exceptions, Windows NT considers no request for basic operations or services unless the request is represented by a bonafide user account that can be used on the computer processing the request. (A number of controls limit the computer on which an account can be used.) Users are never represented by an account unless they know its password (except perhaps in certain "safe" instances controlled by the system).

**Local accounts** can be stored on any NT computer and their scope is on that computer alone. **Domain accounts** are stored on a domain controller (and their respective backup domain controllers), and their scope includes all computers that are members of that domain (including domain controllers) and all the computers of all domains that trust the domain that stores the account.

When a user physically logs onto an NT computer, they must present an account in the scope of that computer and its password. Until they log out, all their local activities are represented by that account. We call this a **local logon** (or **primary logon**).

When a locally logged on user attempts to use a remote Windows NT networking service, like connecting to a remote share directory or printer, or using a remote administrative service, the remote computer establishes a **remote logon** session that governs requests to the

---

<sup>7</sup> This section describes authentication mechanisms used by the native Windows NT network sharing mechanisms. However, one can install services (like FTP) that don't use these mechanisms, and controls like the Right to log on remotely do not constrain such services. One must evaluate the security of such services on a case-by-case basis.

remote server. We use the terms “client” and “server” to describe the local and remote computers. While the details vary as to which account is used, in all cases it must be an account that can be used on the server to which the user demonstrates they know the password.

If the user’s client account is a domain account within the scope of the remote computer by virtue of the domain configuration, that server uses the same account. Otherwise the server attempts to find an account of the same name. If found but the password is different, the user must enter the password. In some cases the user can instead designate an arbitrary account in the server’s scope and its password. (This procedure varies among different remote access scenarios.)

Once a client user establishes a remote logon session on a server, all additional requests to that server (like connecting to other shares on the server) are associated with the initial server session account. Remote server sessions terminate only when the user logs out locally.

Each Windows NT computer has a built-in account named **Guest** that can be disabled (like all accounts) but cannot be deleted. If the Guest account is enabled and has a blank password, most servers represent a remote client by that account when the client user cannot demonstrate the name and password of an account in the server’s scope. Hence, this is an unauthenticated remote logon session and is antithetical to most site security policies. Virtually all Windows NT security guidelines recommend you disable all Guest accounts unless you have an acceptable policy for their use.

To summarize, a user cannot obtain the services of a computer until the computer establishes a logon session associated with an enabled account, whether the session is a local session for a user directly logging on, or a remote session from a remote client computer. The user must in some manner demonstrate they know the name of an account and its password whose scope includes the server computer. A notable exception is that in many cases the server’s Guest account, if enabled, can be used to establish a remote, server-side session without the requesting user demonstrating such knowledge.

In only a very few, isolated cases can a user gain services from a Windows NT computer without a successful secondary logon. These cases are often called **anonymous logons** and we deal with them in later sections.

### ***Domain Models***

Various publications discuss different domain models – schemes for trust relationships. (See for example the references below.) We don’t repeat these here, but do mention a common model for small to medium installations called the Resource Domain model. A single “master” domain holds all the accounts for the site. A set of “resource” domains all trust the master domain and define no appreciable accounts of their own. The resource domains segregate site-wide administration, where each resource domain is the basis for defining administrative groups. Control of global user accounts and groups remains with the administrators of the master domain, whereas administrators of each resource domain retain perhaps exclusive control of their own domain resources.

### ***Logon Rights in Multidomain Environments***

Many sites will use the local Users group and global Domain User groups from visible domains to control basic local and remote logons. This involves assigning the Rights to “log on locally” and “access this computer from the network” to the appropriate, perhaps different combination of these groups on each computer as follows:



- ❑ Include the Domain Users group from the computer's native domain in the local Users group. (By default it is already there.)
- ❑ Decide whether the local Users group is to include Domain Users groups from other domains. (By default it does not.) A key criteria is what the local users understand "Users" to include, for they are the ones that might add Users to their ACLs.

**Note:** This decision should be consistent through a domain, and, preferably, your entire network of domains.

- ❑ Assign the Right "log on locally" to the combination of Users and Domain Users groups that befits your site policy for that computer. Do the same for the Right to "access this computer from the network," which of course may have a different combination of these groups.

The following examples show different combinations of membership in the local Users group, and the groups assigned to the local and remote ("network") Rights. The first is a permissive policy on a NT Workstation that lets all defined users from all visible domains log on locally and remotely:

Users includes:	Domain Users from <i>all visible domains</i>
Log on locally:	Users
Access from network:	Users

Note however, that simply assigning an account to the Right to access the computer from the network does not mean they can do so unless a share directory ACL or other such control allows it.

An example of two equivalent, tighter policies that allow remote sharing from selected, remote domains but not local logon follows:

Users includes:	Domain Users from <i>native domain only</i>
Log on locally:	Users
Access from network:	Users, Domain Users from <i>selected non-native domains</i>

Users includes:	Domain Users from <i>native &amp; selected, non-native domains</i>
Log on locally:	Domain Users from <i>native domain</i>
Access from network:	Users

The following is an example of a permissive policy on a Domain Server or Domain Controller that tightly restricts local logons (which is recommended for these systems) but allows broad sharing:

Users includes:	Domain Users from <i>native and all trusting domains</i>
Log on locally:	<i>selected administrators (the default Windows NT configuration)</i>
Access from Network:	Users

If you wish to more tightly control local and remote logons, add selected users and/or groups to these Rights. (Use groups instead of accounts when possible to simplify maintenance.) For example, the following combination (where "JJones" is the single, regular user for the computer) limits local logon but allows broad file sharing:

Users includes:	Domain Users from <i>native &amp; all visible domains</i>
Log on locally:	JJones, Domain Power Users, & Administrators
Access from Network:	Users

It is difficult to preclude local or remote logons to selected accounts because they are usually a member of the Users group. You must instead replace Users with the allowed accounts in the Rights policy. The Local and/or Remote Logons groups we just described might be useful here.

There are many security reasons for restricting local and/or remote logons more tightly than the domain structure might allow. For example, it is common in multidomain environments to allow a domain A to trust a domain B so that B's accounts can remotely access network services from servers in domain A. However, this implies that domain B accounts can locally log onto workstations in A, which may not have been intended. It is possible but less likely that the situation is reversed, that the trust is created to allow local but not remote logons.

**Related Guidelines:**

*User Rights*

*User Accounts & Groups*

*Administrative Structure*

**References:**

- [Sutt96] Chapter 6, *Planning Domains*. Overall domain planning and "domain models."
- [ConPln] Chapter 1, *Managing Windows NT Server Domains*. Overall domain planning and "domain models."
- [KBase] Q102716 and Q122422 give technical descriptions of authentication, although they are quite low-level.

## 5. Administrative Structure

---

This chapter address the configuration and use of administrative and operator accounts. It recommends no major reconfiguration of the standard Windows NT accounts, but offers guidelines on the scope and assignment of accounts to administrative personnel. Note that there are several alternative techniques that are just as strong.

### Guidelines

#### *The “Administrator” Account*

Each Windows NT computer has an Administrator account that can be renamed, but cannot be deleted. (This is a global account on domain controllers and a local account otherwise.) You define the password to this account during the Windows NT installation process and it serves as the only initial full administrative account. The Administrator account cannot lock due to failed, repeated logon attempts (as can other accounts) and thus benefits from a more complex password.

These guidelines *recommend* that full system administration be done through other administrative accounts (below) and this local Administrator account be used only as a maintenance account of last resort.

#### **Level 1:**

The following techniques are *recommended*:

- ❑ For each domain, define a 14-character password composed of random, printable keyboard symbols, intermixing uppercase and lower. Write this password down and store it in a physically secure location to which only full administrative personnel have access. Your network security is only as secure as this storage.
- ❑ Assign this password to the local Administrator account on each computer in the domain.
- ❑ Change these passwords if an administrator who knows them leaves your institution, or if you suspect the passwords have been compromised.

The guidelines do not recommend these passwords be changed routinely, although they do not preclude this practice.

#### **Level 2:**

Passwords of this complexity are not susceptible to realistic brute force attacks known to work against Windows NT.<sup>8</sup> However, because there is always some, albeit minimal risk relating to the storage of passwords on a computer, at Level 2 it seems prudent to not use the same password on user workstations as on critical network computers, like domain controllers and major data servers. A “major server” is any computer that holds a large amount of critical data from a variety of sources. This technique is also a useful precaution against forgotten administrative passwords. In addition to the procedures from Level 1:

- ❑ Define and store a second password of the same structure and complexity. Assign it to the Administrator account on critical network computers: (1) domain controllers (which

---

<sup>8</sup> Note that the older LANMAN network authentication format can render even these long passwords vulnerable on the network. See “LANMAN Passwords” in the chapter *Networking* for disabling this format.

includes their backup controllers because the account databases are automatically replicated), (2) major servers (see following), and (3) any computer that serves as a gateway to an intranet whose security is less than fully secured.

- ❑ Optionally, you can define even more such administrative passwords to the point where each computer has its own. However, as long as your scheme supports remote administration by other, full administrators, their passwords are likely to be the weak link, and this level of granularity may be excessive.

**Periodic Review:**

- ❑ Review the security of the locked-up, written passwords, including the conditions under which they have been removed for use.
- ❑ Review whether these passwords should be changed due to a change in staffing.

***Full Administrators***

Any account that is a member of the built-in local Administrators group has virtually unlimited power on that computer and, if a domain controller, throughout the domain. We call these **full administrators**. The Administrator account is by default a member of this group. A major goal is to use these accounts rarely, relying instead on accounts of lesser power for day-to-day activities. The procedures for configuring and using these accounts are practices that must be faithfully observed by full administrators themselves.

**Level 1:**

The following techniques are *recommended*:

- ❑ Follow the standard Windows NT metaphor of assigning full administrators domain accounts to be used only for full administrative tasks. Assign these accounts to the domain's global Domain Admins group. In each computer in the domain, assure that the domain's Domain Admins group is a member of the local Administrator's group, along with the local Administrator user (the default configuration).
- ❑ Create one such administrative account and distribute its password to the few individuals entrusted with full administration. See "Shared Administrative Accounts" in the Notes that follow. Name this account whatever you like. Some prefer an obscure name, see "Renaming the Administrator Account," below.
- ❑ If one domain trusts a second domain, the Domains Admins group from the second domain may be included in all the local Administrators groups on computers in the first. This allows full administrators from the second to fully administer the first, and is subject to your site strategy.

**Level 2:**

On larger networks, these guidelines further *recommend* you separate full administration of workstations from sensitive systems (domain controllers and sensitive servers).

- ❑ Define a "Workstation Admins" global group on the domain controller and include it in the local Administrators group of all workstations in the domain.
- ❑ Create a "Workstation Admins" domain account in each domain and include it in the Workstation Admins group. Distribute its password to those who are to be full administrators over workstations, presumably a smaller group than the other full administrators.

- ❑ These guidelines further recommend that each computer's local Administrator's group remove Domain Admins groups from other domains, if present. Workstation Admins may be used cross-domain if deemed secure.

**Note:** While you can view Workstation Admins as of lesser trust than Domain Admins, all full administrators (not just Master Admins) must be fully trusted and must guard their account with equal diligence.

**Periodic Review:**

- ❑ At Level 1, a network where roughly more than 2% of the users are full administrators should cause a reevaluation of the full administrative assignment. (1% for Level 2)
- ❑ For a medium size network (100-1000 computers), the usage of administrative accounts for more than a few brief periods a week should cause a reevaluation of the administrative structure of the network.

***Domain Operators & Power Users***

Domain controller operators (Server, Print, Account, and Backup) and Power Users and Backup Operators on workstations (collectively, "operator" accounts) are designed for most day-to-day administrative duties. These roles should be fully utilized for their intended purpose.

**Levels 1 & 2:**

Operator roles should be fully utilized for all possible day-to-day activities to eliminate the need for full administrative logons. It is acceptable for an account to hold more than one of these roles if the person is trusted with their combined responsibilities.

- ❑ Operator accounts should not be shared among users. We *recommend* a personal account, intended for use only during the course of his or her operator duties, for each operator and include it in the appropriate operator groups. If that person must perform non-operational duties, give them a separate, "personal" account that is not a member of the operator groups.
- ❑ Backup Operators on both domain controllers and workstations have the Backup and Restore Rights that override ACLs. Any malicious program they might run can completely subvert system security and, if a domain controller, domain security (see *Spoofing*). Accounts assigned as Backup Operators must absolutely never be used for any purpose other than performing backups and restores.
- ❑ As a precaution, remove the two Rights to backup and restore files and directories from the Backup Operators group on computers where backup procedures do not require them. For example, where network backup programs do not require membership in these groups.
- ❑ The domain controller operators are local groups and cannot be used outside the domain controller, although some operations can be done remotely when the operator is logged on to another computer. Especially at Level 2, we *recommend* you limit the computers at which operator accounts may log on. The primary danger is that they might be spoofed when logged onto computers that are not tightly controlled. (See *Spoofing*.)

The guidelines *recommend* the following technique that may be suitable for larger sites:

- ❑ **Domain Power Users:** Create a “Domain Power Users” group and include it in the local Power Users group of each workstation, except for critical systems. Place domain accounts in Domain Power Users that are entrusted to perform Power User duties for all workstations in the domain. Allow cross-domain use if appropriate by including the Domain Power Users groups from other domains in a workstation’s local Power Users groups.

Accounts who are members of both the Account and Server Operators might readily be placed in Domain Power Users and should then be allowed to log onto those workstations.

#### **Periodic Review:**

- ❑ Assure that users are only assigned to groups whose duties they are operationally required to perform.
- ❑ Assess the degree to which operators are using their accounts for other than operational duties. This should be minimized.
- ❑ Assure that Backup Operator accounts perform no system commands or operations other than backup and restores.

#### ***Administrative Practices***

- ❑ Administrators should observe the following practices. At Level 1, these should be strictly observed by full administrators, and at Level 2 by all administrators (except perhaps Print Operators, whose capabilities are minimal, but including Power Users).
  - ◆ Administrators should log on only when and where necessary.
  - ◆ The bulk of administrators’ attention should be toward minimizing the use of this account. If they routinely find themselves doing the same task that does not require full administrative capabilities, they should devise a way to handle the task under an account with less power (the less the better).
  - ◆ Administrators should never perform day-to-day, personal tasks (reading E-mail, browsing the Web, writing weekly reports) when logged into a full administrator account. (One of the techniques in *Application & User Home Directories* helps keep administrators from inadvertently running such programs.)
  - ◆ Administrators should always use locking screen savers. (See “Screen Saver Locking” in *General Policies*.)
  - ◆ All users, especially administrators, should be fully versed in various “spoofing” attacks. (See *Spoofing*.)
  - ◆ For maximum security and wherever practical, administrators should work on computers dedicated to administration, administering the network remotely. Generally, these computers should be secured as tightly as possible. They should include only the necessary administrative utilities, and should not hold general applications (word processors, Web browsers, and the like). Access to these computers should be tightly controlled by (1) granting the Right to log on locally only to authorized administrators, and (2), if possible, grant no one the Right to log on remotely and support no file shares. Service should be minimized. To the extent practical, administrators who use these dedicated computers should be prohibited from logging on locally to other computers by their Rights policies.

## Notes

### *Shared Administrative Accounts*

It is common and sound advice that full administrative users not share accounts, and that each should be given their own personal administrative account. However, these guidelines do not require this practice at Level 1 and small Level 2 sites (although they do not recommend against it) for the following reasons:

- ❑ One full administrator has no protection against another malicious one. All objects created by full administrators are owned by the local Administrators group, hence it requires special techniques for one administrator to attempt to protect their work from even benign ones.
- ❑ In a properly configured network, full administrative accounts are rarely used.
- ❑ Sharing a single account means that all administrators have to agree on the password, and this works to prevent one such account from becoming a weak link with a poorly chosen password.

The major difference between shared and personalized administrative accounts is that the audit and other system logs can distinguish between the actions of personalized administrative accounts. (Note that full administrators can alter all such logs so that personalized accounts are not protection against a determined, malicious administrator.) However, the guidelines appreciate the value of personalized administrative accounts, especially at larger sites, and leave them as an option. Note that full administrative accounts other than the local, built-in Administrator account are fully subject to locking.

### *The PASSPROP Utility*

The utility named “PASSPROP” from Windows NT 4.0 Resource Kit has a locking feature that subjects the local Administrator account to the current locking policy but only locks remote (not local) Administrator logons. While a fine feature, the chance of a penetrator guessing the large, random passwords we recommend is infinitesimal and PASSPROP is not strictly necessary.

### *Renaming the Administrator Account*

Some sites rename administrative accounts to something obscure to make it harder to guess administrative logons, although others claim this is classic “security by obscurity.” The goal of this practice is to increase the number of possible combinations of user name and password (the “space”) to a particular target space size. However, it is always better to increase the password complexity to achieve the target space. Windows NT strictly keeps the password private while user names are considered public information on a Windows NT network. Renamed administrative accounts do not stay “secret” very long. (See “User & Share Names Available to Unauthenticated Users” for notes that Windows NT does allow unauthenticated access to some, relatively benign information, like user and share names) The Windows NT password size can accommodate even the most demanding space requirements so there is no need to obfuscate an account name because the maximum password is too small.

One advantage to obfuscating administrative account names is for a degree of protection when an administrator mistakenly removes their password. On balance, the practice of renaming the administrator account does not provide sufficient protection to warrant its inclusion in the guidelines. Administrators who choose to obfuscate them are cautioned not to

decrease their password complexity correspondingly. Always use a password whose space meets your security requirements as though you had not renamed the account.

**Related Guidelines:**

*Spoofing*, on the dangers of operators running malicious programs.

*User Accounts & Groups* (general comments on password complexity)

**References:**

[Sutt96] “Full Administrators” in Chapter 7, *Managing Groups and Accounts*, p. 185.

[Sutt96] “Operators” and “Power Users” in Chapter 7, *Managing Groups and Accounts*, p. 188, 190.

[ConPln] Chapter 2, *Working with User and Group Accounts*.



## 6. General Policies

---

This chapter holds a collection of miscellaneous but important guidelines for general system administration. While these may seem somewhat obscure and detailed, they pose important decisions you must make early in your security implementation. Please note that the guidelines in this chapter and in *Spoofing* are interspersed within the Guidelines & Notes. The guidelines listed correspond to the preceding paragraphs of notes.

### Guidelines & Notes

#### *Raw Devices & Non-NTFS Volume ACLs*

Certain objects in the internal Windows NT object hierarchy allow direct reading and writing of storage media, like hard drives, bypassing the embedded ACLs of the NTFS format. These objects are accessible only to full administrators and require no additional protection.

All logical volumes (anything with a drive letter, like “D:”) are anchored by an object in the hierarchy and this anchor’s ACL governs access to the volume in addition to any embedded controls, like the ACLs of NTFS. Setting this ACL on an NTFS volume anchor is not typically necessary because the ACLs in the NTFS format protect its objects. However, some sites may wish to apply an ACL to the anchor to prevent access to the volume as a whole. Also, some sites may wish to prevent access to certain physical devices, like removable drives, using the anchor’s ACL. See [KBase] Q150101. There are no tools on Windows NT to set or see an anchor’s ACL, although there are 3<sup>rd</sup> party tools that do so.<sup>9</sup>

#### *Restricting Access to Floppies and CDROMs*

The Registry key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT
  \CurrentVersion\Winlogon
```

may have REG\_DWORD values named “AllocateFloppies” and “AllocateCDRoms.” If their values are 1, access to local floppy and CDROM drives, respectively, is restricted to the user currently logged onto the computer. (Note that floppies and CDROMs are among the devices that do not have embedded ACLs.) These entries do not protect other devices like tape drives or removable media. (See also the topic “Raw Devices & Non-NTFS Volume ACLs,” above.)

### Guidelines

#### Levels 1 & 2:

- ❑ Protect floppies and CDROMs (as just described) by setting AllocateFloppies and AllocateCDRoms to 1. You may forego protecting CDROMs on computers whose CDROMs are intended for network sharing. Physically label all such drives “NOT PRIVATE” to warn users of potential dangers.
- ❑ We *recommend* you seek third party applications that apply ACLs to the floppy and floppy-like devices and configure them according to your site policy.<sup>10</sup>

<sup>9</sup> For example, “WinObj” from <http://www.ntinternals.com>.

<sup>10</sup> There is a popular freeware utility called “Flock” by Konstantin Sobolev (sob@cmp.phys.msu.su) of Moscow State University that lets you apply ACLs to floppy and CDROM drives. There is also a

### ***Unauthenticated (“Anonymous”) Remote Access to the Registry***

Windows NT allows its Registry to be edited from a remote computer under standard, secondary logon authentication and under the full control of the Registry’s ACLs. Service Patch 3 (SP3) to Windows NT version 4.0 fixed an oversight whereby unauthenticated remote users could gain access to a computer’s Registry under the “Everyone” group. (The Everyone pseudo-group customarily refers only to authenticated accounts.) This is also called “Anonymous” access. Anonymous access can be restored by modifying two values in the key:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\
  LanmanServer\Parameters
```

First, define a REG\_DWORD value named **RestrictNullSessionAccess** and give it a value of 1. Second, add the string “WINREG” to the multistring value named **NullSessionPipes**. Defining the RestrictNullSessionAccess value as 1 instructs Windows NT to allow anonymous access to the items listed in NullSessionPipes and NullSessionShares. By default, RestrictNullSessionAccess is undefined.

Unauthenticated users are included in the Everyone group, but not other groups. In particular, the **Authenticated Users** group added in SP3 excludes anonymous access. If you must allow anonymous access to the Registry, you can therefore further control such access by using Authenticate Users in places where you might otherwise use Everyone.

Some applications depend on unauthenticated access (most notably, some backup applications) and may not work correctly when anonymous access is removed from the Registry.

See also “Restricting Who can Access the Registry Remotely,” following.

Refer to [KBase] xQ143138, Q126645, and Q143474.

#### **Guidelines**

##### **Level 1:**

- Avoid allowing anonymous access if feasible. If you must allow it to accommodate certain applications, minimize its effects by using Authenticated Users instead of Everyone in the portions of the Registry where anonymous access is not required.

##### **Level 2:**

- Do not allow anonymous access to portions of the Registry except for those areas that you can determine that anonymous access poses no threat.

### ***Restricting Who can Access the Registry Remotely***

If the following Registry key exists:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
  SecurePipeServers\WINREG
```

then only users listed in its ACL, or who belong to groups listed in its ACL, can access the Registry remotely.

---

commercial produce named “SmartSecurity” from Insight Software Solutions, Inc. at <http://smartcode.com/iss> that applies ACLs to a variety of Windows NT devices. Be wary of interactions between “AllocateFloppies” and “AllocateCDRoms,” and any such third party applications. Test the protections thoroughly.

**Note:** The permissions assigned to these ACL entries have no effect. That is, adding a group with Read permission does not allow them only Read access to the Registry. Their specific permissions to access Registry keys remotely is determined solely by the ACL's on the keys that remote users attempt to access.

However, the Registry keys each listed as a REG\_MULTI\_SZ value in the key:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SecurePipeServers\Winreg\AllowedPaths\Machine
```

define Registry keys in HKEY\_LOCAL\_MACHINE that are exempt from WINREG's ACL. These keys and their entire subtrees are exempt from WINREG control, and their access governed solely by their keys' ACL's. (These allowed key names are with respect to HKEY\_LOCAL\_MACHINE.)

The anonymous access allowed by NullSessionPipes is only under the Everyone group, and as such is restricted by the ACL on the WINREG key. That is, unless the Everyone group appears in WINREG's ACL, or unless the accessed key is in WINREG's AllowedPaths, anonymous access is not allowed, even when NullSessionPipes would otherwise allow it.

Refer to [KBase] Q153183, Q155363, and Q157474.

## Guidelines

### Levels 1 & 2:

- Create the WINREG key and add the following groups to the ACL (with any permissions):
  - Authenticated Users
- At Level 2, we *recommend* you remove the Authenticated Users entry and add back users or groups only where operationally necessary (like Administrators or Domain Power Users, if implemented). The ability to read a remote Registry may offer some collateral attack information, and it seems unwise on general principle to allow this capability without a compelling operational reason.
- Initially, assure that the "Machine" key has no values. You may add values but only where necessary and after assuring that the keys in these paths have suitable ACLs.

### *Enabling the Registry Editors*

If the Registry key:

```
HKEY_CURRENT_USER\Software\Microsoft\Windows
\CurrentVersion\Policies\System
```

has a value named "DisableRegistryTools" with a REG\_DWORD value of 1, the standard Registry editing tools do not run. You can start them, but they exit with a brief security message. You can set this to disallow the use of these tools on a computer, or integrate it into a System Policy File scheme. One can gain the same effect by setting the ACL on the executable files for the Registry editors to allow only certain users or groups the permission ("X") to run the program, and withhold read ("R") access from other users (which prevents them from copying the program and running the copy).

However, this protection may not prevent users from accessing the Registry using other commonly available tools, including ones that programmers can easily create. This protection is no barrier to even an unsophisticated, mischievous user. The only reliable protection for the Registry is its ACLs.

## Guidelines

### Levels 1 & 2:

- ❑ Although disabling the Registry editors offers little protection, disable them on computers where they are not needed. The Security Policy files present one convenient technique, where you can enable them for selected users, usually administrators (see *System Policy Files*). Alternately, set the ACL on the Registry editor program files (REGEDT32.EXE and REGEDIT.EXE) as described above.

### *ProtectionMode*

Unseen to most users, the Windows NT operating system has an internal hierarchy that holds many data objects accessible to user programs. Many programs create objects in this hierarchy as a part of their normal activities, and these objects can have ACLs. However, by default many of these objects have loosely set or no ACLs. They are vaguely referred to as “base objects.” This investigation could find no clear, public definition of these objects, although they are rumored to include shared memory segments and interprocess synchronization objects (describe in the Win32 API), communication ports, and drive letter assignments.<sup>11</sup>

Setting the REG\_DWORD value named “ProtectionMode” to 1 in the Registry key:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control
  \SessionManager
```

is said to place some sort of ACL protection on these objects that limits access to their creators and administrators. While the sufficiency of these protections was approved in Windows NT’s original C2 evaluation, public descriptions of what this protection means are vague.<sup>12</sup> The current consensus is that this value should be set on the more security conscious Windows NT installations.

Enabling ProtectionMode in this manner may cause users to be denied operations that may not be security relevant. If these problems develop, find a work-around or disable ProtectionMode. Unfortunately, there’s no common way to add auditing information to these objects, so using the auditing technique (as described in “Installing & Testing New Applications” in *File System & Registry ACL Settings*) is not feasible. See “Tight Security for Shared Objects” in [RKitW].

## Guidelines

### Level 2:

- ❑ The guidelines *recommend* you enable ProtectionMode (described preceding) unless it imposes unacceptable operational constraints. Be **forewarned** that ProtectionMode may disrupt normal system activity and its effects may be obscure. (If the protection and effects of ProtectionMode were better understood, we would probably recommend it at Level 1 also.)

<sup>11</sup> This is likely the control used by the C2CONFIG tool for restricting user redefinition of “drive letters and printers.”

<sup>12</sup> We make no comment on the state of affairs when a potentially important component of Windows NT security appears to be essentially undocumented.

### ***Unauthenticated Event Log Viewing***

By default, guests and unauthenticated users can read the System and Application event logs (but not the Security log). You can prevent this by creating a value named “RestrictGuestAccess” with a REG\_DWORD value of 1 in the Registry keys:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services
  \EventLog\Application
```

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services
  \EventLog\System
```

Apply this protection under the theory that penetrators may be able to glean useful information from these logs. There should be no adverse impact on the system’s overall operation. (Reference: “Secure Event Log Viewing” in [Micr97].)

#### **Guidelines**

##### **Levels 1 & 2:**

- Implement the aforementioned protection.

### ***Print Driver Installation***

You can restrict the ability to add printer drivers to administrators, Print Operators, and Power Users by creating a value named “AddPrinterDrivers” with a REG\_DWORD value of 1 in the Registry key:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control
  \Print\Providers\LanMan Print Services\Servers
```

Untrusted print drivers can maliciously divert user data. (See “Secure Print Driver Installation” in [Micr97].)

#### **Guidelines**

##### **Levels 1 & 2:**

- Implement the preceding protection.

### ***Screen Saver Locking***

Automatic locking of an unattended computer is an important security protection. (Automatic “locking” produces the same result as manually locking the computer from the Security Window.) Several of the screen savers delivered on Windows NT perform this function. Although you can use a Policy File policy to remove the Screen Saver panel from the Display Control, users can use other, readily available tools to change the Registry entries that hold the screen saver name and parameters. The only effective means to enforce screen saver use is to change the ACL on the relevant keys in user’s stored profile image of HKEY\_LOCAL\_USER.

#### **Guidelines**

##### **Levels 1 & 2:**

- As discussed preceding, institute a site practice that requires users to enable screen saver locking at all times. *Recommended* at Level 1 and *prescribed* at Level 2.
- The guidelines suggest 20 minutes-to-lock at Level 1, and 5 minutes at Level 2, although this is considerably site dependent.

### *Protecting Hashed Passwords & SYSKEY*

Windows NT does not store copies of users' passwords, however, it does store a "hash" value computed from the password (often called the "OWF"). You can think of a hashed password as an encryption of the password that nobody can decrypt. Although a person who knows another account's hashed password cannot directly use it to log on, there are some relatively effective attacks available to the holder of hashed passwords. Windows NT's main hashed password storage in the Registry protects them from public view. However, the hashed passwords can also appear in the following, other locations.

The SYSKEY command lets sites increase the protection of locally stored passwords. By default, Windows NT stores a one-way hash of user passwords in the SAM Registry to which only full administrators have access. However, the /s option of the RDISK command and some backup programs may save copies of the hashed passwords under lesser protection. While the hash cannot be used for direct logon, it can be used by malicious (albeit sophisticated) programs for network logon, and can be used in brute force password attacks. This may constitute a significant security risk.

The SYSKEY command lets you configure the system so that user password hashes are encrypted with a 128-bit encryption algorithm for extra protection. Barring a flaw in the algorithm, the algorithm is not breakable. You can use SYSKEY in one of three modes:

1. **Auto Boot:** The system generates an obfuscated, internal encryption key and stores it on the system. This allows for unattended startups. However, if your SAM without SYSKEY protection is vulnerable, then so is this encryption key. This mode is convenient but its security is tenuous.
2. **Floppy Boot:** The system generates a random, complex key and stores it on a floppy that you must insert to boot the system. The key is not stored on the system. This mode is safer, but if you lose the floppy (or a penetrator copies it), you're in trouble.
3. **Password Boot:** You as administrator choose a password that's used as the basis for the encryption key, and is required to boot the system. Again, if you forget the password or it's discovered, you're in trouble.

The description of how to use SYSKEY is involved and we refer the reader to [KBase] Q143475 for details. Option (3) uses an administrator chosen password. This password's space determines the strength of the encryption's resistance to brute force attack. A random, 14-character alphanumeric password<sup>13</sup> produces a key space of about 82-bits which should be more than adequate for all but the most secure sites. (By comparison, 56-bit DES encryption is still quite strong. While it has been broken by brute force, it required the coordinated computing resources of thousands of computers on the Internet. 82-bit encryption is almost 16 million times more difficult than 56-bit.) The risk, of course, is that such a password must be written down and referenced each time it's entered – at least for a while. Hence, physically protecting its written image is equivalent to protecting the floppy used in option (2).

Note that simply learning the SYSKEY encryption key or administrator password, or copying the floppy, does not let a penetrator directly discover your unhashed passwords. They must still break the other protections you apply under the the following guidelines. Assuming you apply them properly, they need to gain full administrative access to the system, and if they

---

<sup>13</sup> Consisting of digits, and upper and lower case alphabetic characters.

get that, security is lost in any case. However, using SYSKEY makes it less important to implement these other protections.

## Guidelines

### Levels 1 & 2:

- As just described, protect WINNT\CONFIG and WINNT\REPAIR directories as specified in *File System & Registry ACL Settings*.
- Restrict the following physical media to full administrators and Backup Operators:
  - Emergency Repair disks created with the “/s” option to the RDISK command,
  - files made by various programs that save the user accounts portion of the Registry (commonly called the “SAM”) to files,
  - backup tapes that have copies of Registry, copies of its active files or manually-made copies of those files, or copies of the files in WINNT\CONFIG.
- At Level 1, we *recommend* using any of the three SYSKEY modes, but not using option (1) on primary and backup domain controllers, or major servers. Effectively, use the SYSKEY mode (1) only on personal workstations. At Level 2, we *prescribe* the use of SYSKEY in modes (2) or (3).

### *Password Notification Feature*

Full administrators can install DLL's that Windows NT activates each time a password is changed, conveying the new password to the DLL. Some such DLL's use this to enforce password restrictions and others use it to synchronize non-Windows NT elements with the Windows NT password. The Registry key:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa
```

can have a REG\_MULTI\_SZ value named “Notification Packages” whose value is a list of DLL names (less the “.DLL” suffix) that reside in the SYSTEM32 directory. It is essential that this registry entry only name trusted DLLs that exist in SYSTEM32 and are read-only to other than full administrators. See also [KBase] Q151082 and Q161990.

## Guidelines

### Levels 1 & 2:

- As described preceding, assure that Notification Packages is empty unless the DLL's it names are installed and used on the system, and protected with a proper ACL.<sup>14</sup> Note that such DLL's are quite specialized and not required for most Windows NT installations.

<sup>14</sup> Microsoft shipped Windows NT 4.0 with this Registry entry set to “FPNWCLNT” but without the corresponding DLL. This means that anyone who could create files in SYSTEM32, which was everyone by default, could install a malicious DLL by simply naming it “FPNWCLNT.DLL.” Subsequent service patches fix this problem.

### *User & Share Names Available to Unauthenticated Users*

Windows NT allows users who, by virtue of the trust relationships, have no access to certain domains to nonetheless see user account names, as well as network and printer share names on computers in those domains. To prevent this anonymous viewing of names, one can add a value named “RestrictAnonymous” with a REG\_DWORD value of 1 to the key:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa
```

(This feature was implemented in 4.0 SP3.) However, this restriction prevents certain windows from listing such names where it might be useful for both administrators and regular users. Further, before a Windows NT user can see names from domains to which they have no access, they must log onto and therefore be authenticated by Windows NT. Hence, even though they have no access to some domains, they are nonetheless authenticated in at least one domain or workstation. Furthermore, even though common interfaces no longer list such names, users can easily write unprivileged programs that query such names even when the above restriction is in place. Allowing these names to be visible is a modest risk at most. For these reasons, the Guidelines do not prescribe this restriction.

### *Hiding the Last User Logon*

By default, Windows NT displays the previous account name on the logon window. You can prevent this by creating a value named “DontDisplayLastUserName” with a REG\_SZ value of “1” in the Registry key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\
Current Version\Winlogon
```

In most cases this is an almost inconsequential protection. Windows NT and many user communities do not consider user names to be secret. They are often abbreviations of a user’s name and used for E-mail. Windows NT’s general philosophy is to not hold them secret across the network. Access to a user’s account must be guarded by their password, which by contrast is a steadfast secret. Computers in “publicly accessible” locations should exercise this feature on general principle, but do not rely on hiding the last logon name for any significant degree of security.

## **Guidelines**

### **Levels 1 & 2:**

- We *recommend* you set this parameter on computers readily accessible to untrusted people who do not have logons to the network.

### *Shutting Down the System*

If the Registry key:

```
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\
CurrentVersion\Winlogon
```

has a value named “ShutdownWithoutLogon” with a REG\_SZ value of “1,” then a “Shutdown” button appears on the logon window that allows anyone to shut the system down without logging on. These guidelines do not treat shutting the system down (a “denial of service”) as a security issue, although it may be an important operational issue. We have seen no examples of how system shutdown can be used to compromise system security apart from denial of service. Note that ShutdownWithoutLogon is broader than allowing Everyone the Right to “shut down the system” because the Right applies only to valid user accounts that



have logged on, whereas ShutdownWithoutLogon lets unauthenticated users shut the system down.

Bear in mind that most workstations have an accessible power switch. A user determined to turn off the computer may well use this switch if they have no other means, and it is possible that auditing data could be lost in this case, which is a security issue. Therefore, you should disable the power switch (most workstations have a physical key lock to do so). But then again there is the power cord, circuit breaker panels, and so forth.

### **Guidelines**

#### **Levels 1 & 2:**

- ❑ While using this Registry key to help prevent shutdown on shared computers is usually a valuable operational technique, the more important security issue is the shutting off the computer's power which has some chance of losing auditing data. If auditing is important to your site policy, take whatever protections you can.

### ***Miscellaneous Hot-Fixes***

Microsoft maintains a list of recently fixed security problems at:

<http://www.microsoft.com/security>

and more detailed list of post-SP3 "hot-fixes" at:

<ftp://ftp.microsoft.com/bussys/winnt/winnt-public/fixes/usa/NT40/hotfixes-postSP3>

Microsoft has promptly issued hot-fixes when legitimate security flaws are discovered in Windows NT. This document accounts for all issues up to 1 February 1998.

### **Guidelines**

#### **Levels 1 & 2:**

Download and apply the following hot-fixes:

- ❑ Hot-fixes are available for a few denial-of-service attacks that can cause the TCP/IP protocol to hang or slow down. See [KBBase] Q179129, Q165005, and Q154174.
- ❑ A "GetAdmin" bug in Windows NT allowed users to become members of the local Administrators group and therefore all-powerful. See [KBBase] Q165005.

#### **Periodic Review:**

- ❑ Regularly check the latest Microsoft hot-fixes and apply them where appropriate.

### ***The C2CONFIG Tool***

The Windows NT Resource Kit [RKitS] includes a tool called C2CONFIG that purports to tell you whether your system is configured as "C2." Unfortunately, the C2 rating is not an operational criteria. For example, although C2 requires a security auditing mechanism, it does not attempt to force end user administrators to enable it. One cannot say that a system with auditing turned off is "not C2" – it's simply a C2 system whose administrators have chosen not to use auditing. C2 is a product stamp of approval that helps assure the buyer that an operating system includes certain security features and has been responsibly implemented. It does not indicate the system is "secure," although it elevates our confidence in a system's inherent security. Unfortunately, the meaning of C2 and DoD evaluation is terribly misunderstood in today's literature.

**UNCLASSIFIED**

The C2CONFIG tool can usefully indicate when you have enabled features that were not reviewed during C2 evaluation effort, and check a few security relevant settings. However, you should not assume that because you pass the C2CONFIG tool your system is “secure” enough for your situation, nor that if you fail any aspect that your system is not so. These guidelines address and subsume all significant security aspects of the C2CONFIG tool.

## 7. File System & Registry ACL Settings

---

Windows NT initializes the ACLs on its root NTFS file system (mainly the WINNT directory) to maximize compatibility to Windows applications which represents a modest tradeoff against security. While its ACL settings are not insecure, there is considerable room for tightening them. The Registry's ACLs protect it reasonably well by default. However, for compatibility reasons some of its keys are left more permissive than prudent. The changes below tighten both the NTFS and Registry ACLs and apply to both Levels 1 and 2. The only difference is that at Level 1 the trust criteria for including users in the App Installers group is lower than Level 2.

Upon installation, the "Everyone" group (which we call the "Public" in this section because it can at times also apply to other widely inclusive groups, like "Users") has "write" access to certain portions of the Registry and the WINNT directory tree. While not blatantly insecure, the guidelines put these portions under modest administrative control at Level 1, and reserved them to a trusted administrative function at Level 2.

Our general strategy is to remove all but Public read (RX) access from the items in the trees to be restricted, giving all permissions removed to a new "App Installers" local group. This group therefore has all the capabilities of the Public after a standard NT install. We use the term "installer" because almost all of the requirements to create new items in these trees come during the installation of new application software or hardware, or its maintenance.

At Level 1, App Installers can include regular users installing applications that need relatively benign write access to WINNT and the Registry. While it is acceptable to place all users in this group, proper practice is not to do so unless that user's applications currently require such access. At Level 2, App Installers becomes a trusted, administrative group, although with less than full administrative capabilities.

At all levels, we also allow App Installers to add new applications to official application directories. These are directories advertised as holding safe, approved software. See "Application Directories" in *Application & User Home Directories*. At Level 2, administrative diligence helps assure that all applications reside in these directories. (One could create two such groups: one for WINNT and the second for application directories, although we don't further discuss this option.)

App Installers do not gain complete control to application directories, and these Registry and WINNT trees. Typically, they have "Add & Read" access to directories which lets them add new objects, but at most Read access to preexisting objects. They of course have full control of objects they create. The difference between Levels 1 and 2 is the level of trust placed in the App Installers who must enforce such access.

The recommended settings also tighten a number of other items in WINNT and the Registry at both levels.

**Note:** The inevitable result in tightening these ACLs is that some applications may fail to install or function properly, and you must resolve any conflicts on a case-by-case basis. (See "Installing & Testing New Applications" in *File System & Registry ACL Settings*.) You may also encounter other operational problems which are a trade-off against security. However, as a part of this study, we tested many main-line applications against these settings with relatively few conflicts.

## Guidelines

### Level 1:

- ❑ We *recommend* you create a local “App Installers” group on each computer and a “Domain App Install” group in each domain. Include the latter in the former as per the standard domain-wide administrative group strategy. Populate these groups with accounts that you trust to install software. (It is not necessary to include full administrators in App Installers because they are given full access to all objects on the system.)

**Note:** The App Installers group is not strictly necessary, especially on small installations. You can instead rely on administrators to install software. If you choose to do so, omit App Installers from the ACL settings below.

- ❑ Set permissions on the Windows NT file system as defined in the following sections “File System ACL Settings,” and on the Registry as defined in the section “Registry ACL Settings.”

### Level 2:

In addition to the procedures at Level 1:

- ❑ Assure that a minimal number of trusted personnel populate the App Installers group. As a rule of thumb, there should be roughly the same number of App Installers as full administrators.
- ❑ Do not include regular, day-to-day accounts in App Installers. Instead, create accounts for use only during installation and include only these in App Installers. You can include Server Operator accounts in App Installers.

### Periodic Review:

- ❑ Assure that prescribed Registry and WINNT protections are not relaxed without reason.
- ❑ Assure that membership in the App Installers group is commensurate with guidelines above.
- ❑ We *recommend* you seek automated tools to check the correctness of these ACLs, such as the *Checker* software developed on this contract.<sup>15</sup> Otherwise, review them in your regular periodic reviews.

## Notes

### *File System ACL Settings*

The Windows NT file system holds many critical security files, most of which are in the system root directory (usually but not necessarily named “WINNT,” the name we use). While reasonably protected by default, for compatibility reasons many files and directories are left more permissive than prudent for more secure sites. The changes below apply to both Levels 1 and 2. The only difference is that at Level 1 the trust criteria for including users in the App Installers group is lower than Level 2. The table that follows lists recommended ACLs for standard Windows NT files and directories:

- ❑ The first line of each group below is the directory, and the rest are files (or where noted subdirectories) within the directory.

<sup>15</sup> Check the Trusted Systems Services Web site for announcements about the availability of Checker (<http://www.TrustedSystems.com>).

- The following shorthand names used in the table indicate the following local groups:

**Public** = Everyone  
**Installers** = App Installers  
**SrvOps** = Server Operators  
**PwrUsers** = Power Users

As a matter of site policy, you might want to use the potentially more narrow groups, like the local Users or Authenticate Users<sup>16</sup> groups, as the Public group.

- The permission set names (like “Read” and “Add & Read”) are those used by the ACL window.
- The asterisk (\*) after a set of directory permissions indicates that, if possible, the directory permissions (the “left-hand” set) should not propagate to newly created directories. While this can occur on Windows NT, there is no Windows NT interface to control where it happens. While not necessary, this is a recommended practice, although you must seek 3<sup>rd</sup> party tools<sup>17</sup> for this purpose.
- All these ACLs are to include the following entries, which is the standard default for WINNT:

**Common ACEs in all ACLs within WINNT**

CREATOR/OWNER Administrators SYSTEM	Full Control
---	--------------

- The “(none)” notation in the tables means no ACE’s except these Common ACE’s.
- For the directories listed, change only the directory’s ACL (not its subdirectories) unless indicated. Some of these file and directories do not appear on both the Windows NT Server and Workstation products. Note again that the system root may be named other than “WINNT.”
- We assume you have installed Windows NT on the C: volume. If otherwise, make the adjustment in the text that follows.
- For Windows NT Workstation, ignore Server Operators and Print Operators. For Windows NT Server, ignore Power Users.
- These settings seldom change the default Windows NT permissions for the various Operators, Power Users, or other built-in groups. Our general assumption is that Server Operators are generally trustworthy and they are given broad access to sensitive directories and files. They can maliciously use these capabilities to gain full administrative access through various spoofing techniques. Some sites may wish to temper this policy.

<sup>16</sup> The only difference between Everyone and Authenticated Users involves a few Windows NT services that provide services to unauthenticated users (who are a member of Everyone but not Authenticated Users). However, there are no publicized, unauthenticated ways to access Windows NT file systems, and the difference is a moot point. Some sites may wish to use Authenticated Users as the Public entry as a sensible precaution. The Users group can further restrict access, since all Users are authenticated but can also exclude some authenticated users. We see no compelling reason to use the Users group for the items in this table.

<sup>17</sup> Super CACLS ( <http://www.TrustedSystems.com> ) is commercial tool that can do this. Although there may be others, none are delivered with Windows NT or its resource kits.

UNCLASSIFIED

	Guidelines	Std NT Install	Comments
C:\	Installers: Change Public: Read SrvOps: Change PwrUsers: Add*	Public: Change SrvOps: Change	Note non-propagating ACE.
files	Installers: Change Public: Read SrvOps: Change	Public: Read SrvOps: Change	
IO.SYS MSDOS.SYS	Installers: Change Public: Read SrvOps: Change	Public: Change	16-bit compatibility only. Delete if possible.
BOOT.INI, NTDETECT.COM NTLDR	(none)	Public: Read	
AUTOEXEC.BAT, CONFIG.SYS	Installers: Change Public: Read SrvOps: Change	Public: Change	16-bit compatibility only. Delete if possible.
C:\TEMP	Public: (RWXD)*(NotSpec)	Public: Full	Note non-propagating ACE. See note [7].
C:\WINNT\	Installers: Change Public: Read SrvOps: Change	Public: Change SrvOps: Change	See note [16] & [19].
files	Public: Read SrvOps: Change	Public: Read SrvOps: Change	
win.ini	Installers: Change Public: Read SrvOps: Change	Public: Full	16-bit compatibility file.
control.ini	Installers: Change Public: Read SrvOps: Change	N/A	
netlogon.chg	(none)	N/A	DC file open-locked by OS. Created after install.
\WINNT\config\	Installers: Change Public: Read SrvOps: Change	Public: Change SrvOps: Change	All items in tree.
files	"	Public: Read SrvOps: Change	
\WINNT\ursors\ \WINNT\fonts\	Installers: Change Public: Add&Rd SrvOps: Change PwrUsers: Change	Public: Change SrvOps: Change	All items in tree. See note [4].
files	"	Public: Read SrvOps: Change	
\WINNT\help\	Installers: Change Public: Read SrvOps: Change PwrUsers: Change	Public: Change SrvOps: Change	All items in tree, except as noted following. Only *.HLP and *.CNT exist on install.
files	"	Public: Read SrvOps: Change	See note [2].
*.GID *.FTG *.FTS	Public: Change	"	See note [1]. These files not present at installation.
\WINNT\inf\	Installers: Change Public: Read SrvOps: ??	Public: Change SrvOps: Change	All items in tree. Only *.ADM, *.INI, & *.PNF installed.
files	"	Public: Read SrvOps: Change	See note [3].
*.ADM	Public: Read	"	See note [17].
*.PNF	Installers: Change Public: Read SrvOps: Change	Public: Change	

**UNCLASSIFIED**

<b>WINNT\media\</b>	Installers: Change Public: Read SrvOps: Change PwrUsers: Change	Public: Change SrvOps: Change	All items in tree. See note [18]. Only *.RMI, *.MID, & *.WAV present on install.
<i>files</i>	"	Public: Read SrvOps: Change	
*.RMI	Public: Change	N/A	See note [18].
<b>WINNT\profiles\</b>	Installers: Add&Read Public: (RWX)*(NotSpec)	Public: Change	See note [5].
Dir: (user name)	(same as NT default)	User: Full	
Dir: All Users Dir: Default	Installers: Change Public: Read	Public: Read	
<b>WINNT\repair\</b>	(none)	Public: Read SrvOps: Full PwrUsers: Change	All items in tree. See note [6]. Only hive files and two *.NT files present on install.
<i>files</i>	(none)	N/A	Probably dynamically managed by NT.
<b>WINNT\system\</b>	Installers: Change Public: Read SrvOps: Change	Public: Change SrvOps: Change	All items in tree. See note [20]. Mostly *.DLL and *.DRV upon installation.
<i>files</i>	"	Public: Read SrvOps: Change	
<b>WINNT\System32\</b>	Installers: Change Public: Read SrvOps: Change BckOps: Change	Public: Change SrvOps: Change	Same as WINNT except for Backup Operators. See note [8] & [19]. See note [9].
<i>files</i>	Public: Read SrvOps: Change	Public: Read SrvOps: Change	
\$winnt\$.inf	Installers: Change Public: Read SrvOps: Change	N/A	See note [10].
AUTOEXEC.NT CONFIG.NT	Installers: Change Public: Read SrvOps: Change	Public: Change SrvOps: Change	Used to init the NT-DOS environment. See note [10].
cmos.ram midimap.cfg	Public: Change	Public: Change SrvOps: Change	
localmon.dll decpsmon.* hpmon.*	Installers: Change Public: Read SrvOps: Change PrintOps: Change PwrUsers: Change	Installers: Change Public: Read SrvOps: Full PrintOps: Full PwrUsers: Change	
<b>WINNT\System32\config\</b>	Public: List	Public: Read SrvOps: Change	All items in tree. See note [6].
<i>files</i>	(none)	Public: Read SrvOps: Change	
default, software, system, userdiff	(none)	N/A	
*.EVT	(none)	(none)	
<b>WINNT\System32\DHCP\</b>	Public: Read SrvOps: Change	Public: Read SrvOps: Full	All items in tree. See note [21]. Empty on install.
<i>files</i>	"	Public: Read SrvOps: Change	
<b>WINNT\System32\drivers\ (including \etc subdir)</b>	Public: Read	Public: Read SrvOps: Full	All items in tree. See note [11].
<i>files</i>	"	Public: Read SrvOps: Change	
<b>WINNT\System32\LLS</b>	Installers: Change Public: Read SrvOps: Change	Public: Change SrvOps: Change	All items in tree. "License Logging Service."
<i>files</i>	"	"	

UNCLASSIFIED

<b>WINNT\System32\IOS2</b> (incl \DLL subdir)	Public: Read SrvOps: Change	Public: Change SrvOps: Change	All items in tree. See note [12].
<i>files</i>	"	Public: Read SrvOps: Change	
<b>WINNT\System32\IRAS</b>	Public: Read SrvOps: Change	Public: Change SrvOps: Full PwrUsers: Change	All items in tree. See notes [13] and [22].
<i>files</i>	"	Public: Read SrvOps: Change	
<b>WINNT\System32\Repl</b>	Public: Read SrvOps: Full	Public: Read SrvOps: Full	All items in tree. See note [14].
<i>files</i>	"	(none in std delivery)	
<b>WINNT\System32\Repl\</b> <b>\import, export</b> ... and "scripts" subdirs	Public: Read SrvOps: Full Replicator: Change	Public: Read SrvOps: Change Replicator: Change	
<i>files</i>	"	(none in std delivery)	
<b>WINNT\System32\spool</b>	Installers: Change Public: Read SrvOps: Full PrintOps: Change PwrUsers: Change	Public: Read SrvOps: Full PrintOps: Full PwrUsers: Change	See note [26].
<i>files</i>	"	Public: Read SrvOps: Change	
<b>\drivers\</b> <b>\drivers\w32x86\2\</b> <b>\prtprocs\</b> <b>\prtprocs\w32x86\</b> <b>\drivers\w32x86\</b>	Installers: Change Public: Read SrvOps: Full PrintOps: Change PwrUsers: Change	Public: Read SrvOps: Full PrintOps: Full PwrUsers: Change	
<i>files</i>	"	Public: Read SrvOps: Full PrintOps: Full PwrUsers: Change	
<i>files in \drivers\w32x86\</i>	"	Public: Read SrvOps: Change	
<b>\printers\</b> <b>\tmp\</b>	Installers: Change Public: (RWX)(NotSec) SrvOps: Full PrintOps: Change PwrUsers: (RWXD)(WXD)	Public: Read  SrvOps: Full PrintOps: Full	See note [27].
<i>files</i>	"	Public: Read SrvOps: Full PrintOps: Full	
<b>WINNT\System32\viewers</b>	Public: Read SrvOps: Change	Public: Change SrvOps: Change	All items in tree. See note [15].
<i>files</i>	"	Public: Read SrvOps: Change	
<b>WINNT\System32\wins</b>	Public: Read SrvOps: Change	Public: Change SrvOps: Change	All items in tree. See note [23].
<i>files</i>	"	Public: Read SrvOps: Change	
<b>C:\...\*.EXE</b>	Public: X	N/A	See note [24].
<b>C:\...\</b> <b>*.BAT</b> <b>*.COM</b> <b>*.CMD</b> <b>*.DLL</b>	Public: Read	N/A	See note [24].
<b>C:\...\*.INI</b> except boot.ini	N/A	N/A	
<b>WINNT\system32\</b> four BSD r* commands	(none)	N/A	See note [25].



UNCLASSIFIED

- [1] History files like .GID must be publicly writable to be used by more than one user. There is little security risk in doing so.
- [2] Help files can contain executable code. To prevent spoofing, these files should not be writable by regular users.
- [3] Both .INI and .PNF files control installation and actions of applications, and should not be publicly writable to guard against spoofing. We're a little nervous about letting less than full administrator install scripts that install hardware, but this seems okay for now.
- [4] Letting the public install new cursors and fonts seems safe under the assumptions that cursor and font files do not contain executable, arbitrary code.
- [5] Each subdirectory of PROFILES holds a user's profile. The system properly and automatically manages this tree and no changes are necessary.
- [6] Remove Everyone and Server Operator entries from these trees. Files in REPAIR can contain hashed passwords which should be accessible only to full administrators. See "Protecting Hashed Passwords & SYSKEY" in *General Policies*.
- [7] We do not allow Server Operators or Power Users access to files in TEMP because it allows them to read and write any other user's files, which seems unnecessarily lax. However, it also does not allow them to clean up the junk that inevitably accumulates in /TEMP, and it seems unwise to require a full administrative logon to clean up the junk. One might allow specific, trusted users (or a group) to delete (but not read and write) all files in /TEMP.
- [8] SYSTEM32 holds many critical security files. Unfortunately, it is also the great dumping ground of many applications that create files here when they don't know where else to put them. The system searches this directory for DLL names, and for this spoofing reason alone the public should not be allowed to add new files. You can expect restricting this directory to be a major source applications compatibility issue. Newly added files may need custom ACLs.
- [9] Windows NT backup programs customarily create temporary files here. Backup Operator's need Change access to some files by default.
- [10] These present potential spoofing opportunities and must be guarded. At Level 2, we recommend you remove the App Installers and Server Operator entries, leaving write access only to full administrators. Note that these are set with public Change permissions by the C2CONFIG tool.
- [11] By default Server Operators have Change or Full Control to these trees, which gives them the ability to subvert system security. Therefore we preclude their access. While this could be relaxed for Server Operators, App Installers should never be allowed to install new Drivers.
- [12] The guidelines do not address the OS/2 subsystem, and without this analysis, the prudent course is to set the files in this tree Public: Read.
- [13] We continue the default practice of granting Power Users Change access to the elements of this tree. However, this may well be contrary to some site policies.
- [14] Some files installed in this tree (like NTCONFIG.POL, see *System Policy Files*) may need tighter ACLs. With the exception of the NETLOGON share, this tree is used mainly by the Replicator service and the ACLs of replicated files and directories are at the discretion of administrators. It seems reasonable to give Server Operators Full permission (rather than Change) which lets them fully manage Replication.

## UNCLASSIFIED

- [15] Holds the QUICKVIEW add-in to the File Explorer's File menu that prints attributes of various file types using the DLL's in this directory. This directory must therefore be protected as holding an administrative utility, so we don't let App Installers manage it. We do allow Server Operators access, although Level 2 systems concerned with spoofing might want to curtail this since it's probably seldom used.
- [16] A few notes about files in WINNT.
- CONTROL.INI is not used by NT.
  - NETLOGON.CHG is on domain controllers only and is open-locked by the system.
  - SETUP.OLD and SETUP.TXT are of uncertain use.
- [17] These are used by administrative interfaces and should only be changed by administrators.
- [18] Of uncertain use in Windows NT. This directory and its files may need to have public Change access.
- [19] It's a little dangerous to allow Installers to add new executable files into directories in full administrator's DOS search path, like WINNT and WINNT\SYSTEM32. A script that continually denied administrators access to \*.EXE files not owned by the Administrators local group would be valuable.
- [20] Remove this directory from the administrator's search path.
- [21] DHCP network administration is beyond the duties of Installers, so they gain no access here. While it would do little harm to allow Power Users access, it seems pointless.
- [22] RAS network administration is beyond the duties of Installers, so they gain no access here. Power Users do not gain access on the theory that this lets them open up their systems to dial-in networking.
- [23] This appears to be a networking directory, and access is correspondingly denied to App Installers.
- [24] Set this only for standard NT delivery files. Note that only full administrators can modify or replace these files. One should only relax this for programs that are used by full administrators. For untrusted programs later added, preclude administrative access by removing the "X" permission.
- [25] These commands are for UNIX-like "remote logon" services. They should be removed from the system, and added back only in the context of which general network server/clients are allowed on the computer.
- [26] The general philosophy is to allow Server Operators full control of this tree, whereas Print Operators and Power Users only get Change.
- [27] This is the default, installed spool folder for Windows NT. However, its location can be changed by modifying the appropriate Registry entry, and this ACL should be applied to alternative spooling directories. This setting does not let Public users read each other's spool files. The TMP directory might also be used for temporarily printing information that might contain user data and its files should therefore be protected. The ACL setting for Power Users does not allow them to read other user's spool files, although they can delete them. While there may be other means by which Power Users can snoop on the users they allow onto the system, these settings are a common sense protection against snooping on spool jobs.

**Registry ACL Settings**

The permission set names used below are:

Read = QENR  
 Add = QCENR  
 Change = QSCENDR

For reference, individual permissions are:

Q = query value (read a key's values)  
 S = set value (write a key's values)  
 C = create subkey  
 E = enumerate subkeys (read names of subkeys)  
 N = receive notification when key changes  
 D = delete the key  
 R = read key's ACL

The shorthand names below indicate the following local groups:

**Public** = Everyone  
**Installers** = App Installers  
**SrvOps** = Server Operators  
**PwrUsers** = Power Users


The techniques that prevent unauthenticated access to the Registry remove the need to use Authenticated Users as the Public entry. Similarly, it's not clear whether there's any significant value in using Users as the Public entry, although it is potentially tighter than Everyone and Authenticated Users. If Users causes no operational problems, it could easily be used.

All the ACLs below are to include the following entries, which is the standard default for the Registry:

**Common ACEs in all ACLs within Registry**

CREATOR/OWNER Administrators SYSTEM	Full Control
---	--------------

All other Registry keys appear adequately protected as installed for both Levels 1 and 2.

 Modify the Registry keys that follow by removing the Everyone entry from these keys (if it exists) and replacing it with the entries indicated. Do not change the ACLs of the key's subkeys unless the "Entire tree" comment indicated you do so, and in this case change the ACLs of all keys in the base key's tree. Except as noted, do not change the permissions for the various Operators, Power Users, or other built-in groups.

**UNCLASSIFIED**

**HKEY\_LOCAL\_MACHINE**

\Software	Installers: Change Public: Read	See note [1].
\Software\Classes	Installers: Add Public: Read	Tree needs special treatment. See note [13].
\Software\Microsoft\Windows \CurrentVersion \App Paths	Installers: Change Public: Read	Entire tree. See note [2].
\Software\Microsoft\Windows \CurrentVersion \Explorer	Public: Read	Entire tree. See note [3].
\Software\Microsoft\Windows \CurrentVersion \Embedding	Installers: Change Public: Read	Entire tree.
\Software\Microsoft\Windows \CurrentVersion \Run, RunOnce, and Uninstall	Public: Read	Three keys. See note [4].
\Software\Microsoft \Windows NT\CurrentVersion \AeDebug	Public: Read	Entire tree. See note [5].
\Software\Microsoft \Windows NT\CurrentVersion \Compatibility	Installers: Change Public: Read	Entire tree.
\Software\Microsoft \Windows NT\CurrentVersion \Font*, GRE_Initialize	Installers: Change Public: Add	Keys that begin with "Font," except FontDrivers, and GRE-Initialize. See note [10].
\Software\Microsoft \Windows NT\CurrentVersion \Type 1 Installer\Type 1 Fonts	Installers: Change Public: Add	
\Software\Microsoft \Windows NT\CurrentVersion \Font Drivers	Public: Read	See note [11].
\Software\Microsoft \Windows NT\CurrentVersion \Drivers, Drivers.desc	Public: Read	Entire tree. See note [9].
\Software\Microsoft \Windows NT\CurrentVersion \MCI, MCI Extensions	Installers: Change Public: Read	Entire tree.
\Software\Microsoft \Windows NT\CurrentVersion \Ports	INTERACTIVE: Change Public: Read	Entire tree. See note [12].
\Software\Microsoft \Windows NT\CurrentVersion \WOW	Public: Read	Entire tree. See note [6].
\Software\Microsoft \Windows NT\CurrentVersion \Profile List	Public: Add*	Install as non-propagating ACE if possible. See note [7].
\Software \Windows 3.1 Migration Status	Public: Read	Entire tree.
\System\CurrentControlSet \Services\LanmanServer\Shares	Public: Read	Entire tree. See note [14].
\System\CurrentControlSet \Services	Public: Read	Entire tree. See note [8].

**HKEY\_USERS \.DEFAULT**

\Software\Microsoft \Windows NT\CurrentVersion \Program Manager \Common Groups	Installers: Change Public: Read	
---	------------------------------------	--

- [1] As entry point into this tree, the Software key should allow only App Installers to create new subkeys.
- [2] Empty on install. To contain the potential spoofing threat, withhold Public:Write access from all newly added keys.
- [3] Appears to be unused.
- [4] The command named in the Run key runs at logon for all users (including administrators) and must therefore be protected against spoofing. It should only be writable by full administrators. Similarly protect RunOnce and Uninstall. (Refer to [KBase Q126713], which postdates the discovery of this problem during this study.)
- [5] Parameters for the system debugger that users can run when a program crashes (like “Dr. Watson”). Access is restricted to prevent the potential spoofing opportunity.
- [6] Holds parameters for the DOS environment. Although it is not clear how serious a spoofing threat exists, it seems wise to prevent Public modification.
- [7] Each subkey in Profiles holds parameters for a profile created in WINNT\Profiles. To prevent spoofing, a new subkey should not be publicly writable. Unfortunately, there’s no standard Registry ACL tool that allows the public to create keys that then have no public access, although “Add” permission is secure as long as the subkeys don’t themselves have meaningful subkeys, which is the case in Profiles. Third party tools that can install ACL entries that don’t propagate to subkeys<sup>18</sup> are important here because they produce the desired protection.
- [8] Because only full administrators can install applications in the Service manager, installing their parameters here might seem benign. However, because services commonly run under the all-powerful SYSTEM account it seems advisable to only allow full administrators to modify service parameters. See *System Services*.
- [9] Drivers32 is the principal storage control location for Windows NT drivers, and is strongly protected. The function of the Driver key is unclear, but we protect it anyway.
- [10] This is consistent with the protection afforded fonts in the WINNT directory, above. Some sites may wish to restrict Public user access to Read to prevent them from adding fonts.
- [11] The function of items in FontDrivers is unclear. However, it may define the locations of executable code that runs within the operating system kernel, which makes it extremely sensitive code. We highly protect this key for safety.
- [12] Parameters for COM, LPT, and other ports. We allow INTERACTIVE users to modify these because there seems little security risk, although some sites may wish to tighten these ACLs. Note that [Micr97] advocates tightening these keys to Public: Read.

<sup>18</sup> Super CACLS ( <http://www.TrustedSystems.com> ) is the only one we found on our investigation.

- [13] Upon installing Windows NT, set the ACLs on the entire Classes tree to Public: Read (plus the Common ACEs), then set the ACL on Classes key as noted. (This removes the INTERACTIVE entry from these ACLs.) This Registry tree holds various properties associated with applications, like the correlation between the file name extension and the application defined to handle it. To contain potential spoofing threats, it seems prudent to limit these keys, although it may impact some applications.
- [14] The values in this key and its Security subkey holds critical information about directory and printer shares. These values are adequately protected by default. However, Public can add new subkeys to these keys, and guidelines like [Micr97] advocate tightening these as indicated above.

The following notes describe the general registry hives and are for general information only. The ACLs in the **HKEY\_LOCAL\_MACHINE** (HKLM) hive vary considerably. HKLM holds the trees: HARDWARE, SAM, SECURITY, SOFTWARE, and SYSTEM. The HARDWARE subtree is completely recreated at each boot to reflect the system's hardware detected during the boot process. The SAM subtree holds user accounts and is protected against all access. However, full administrators may change the ACL and therefore have access to this tree. The accounts are encrypted and such access is pointless. SECURITY\SAM and SAM\SAM are the same object. (One is linked to the other.) The SAM tree is stored in the CONFIG\SAM file inside WINNT\SYSTEM32. The SECURITY tree stores various policies (like Rights and audit policies), and is stored in the CONFIG\SECURITY file inside WINNT. It is also encrypted. SOFTWARE holds many user-independent application parameters and is stored in CONFIG\SOFTWARE. SYSTEM holds non-volatile system configuration information and is stored in CONFIG\SYSTEM, and CONFIG\SYSTEM.ALT is a fail-safe copy of this file.

The **HKEY\_CURRENT\_USER** (HKCU) hive holds the parameters that apply to the user logged on locally. In its normal profile mode, the system saves this hive at logout and restores it at logon to the local profile file ("###" is a number chosen by the system to make the profile unique, although it may not be present):

```
WINNT\Profiles\NAME### \NTUSER.DAT
```

The public does not have access to HKCU – only its user and administrators. The following HKCU subtree:

```
HKCU\Software\Microsoft\Windows\CurrentVersion\Policies
```

allows the current user only Read access. The system uses this tree for parameters intended to restrict the user in a manner that the user cannot relax.

The **HKEY\_USERS** hive includes a .DEFAULT subtree that the system uses for programs that need its parameters but that don't have logon sessions. It also holds a tree for each user with an active local session, including the logged on user (which is linked to HKCU), and, for example, users running from the Scheduler service.

Other hives:

HKEY\_CURRENT\_CONFIG is a symbolic link to:

```
HKLM\System\CurrentControlSet\Hardware Profiles\Current
```

HKEY\_CLASSES\_ROOT is a symbolic link to:

```
HKLM\SOFTWARE\Classes
```

### ***Installing & Testing New Applications***

The guidelines prescribe “tightening” ACLs on various files, directories, and Registry keys from the standard Windows installation. The most likely adverse effect is that application programs fail because they can’t modify these objects. While the guidelines only tighten areas that applications should not in principle be modifying, there are no strong standards, and applications not specifically written for Windows NT cause the most problems. It is therefore important that you test newly installed software for such errors. Installing such applications (as opposed to their day-to-day use) can also bring about these problems, and the security log technique below is useful in both cases.

Applications properly written for Windows NT keep user specific data in a directory and Registry keys dedicated to that user, for example their home directory or directories they can designate within the application (usually on its “Options” window). When used by regular (non-administrative) users, such applications do not, in general, create or modify files in their installation directories or the HKEY\_LOCAL\_MACHINE Registry hive as a result of use by regular (non-administrative) users.

However, there are exceptions. When you install an application, test it thoroughly by running it as a regular user. Specifically, don’t use an account that is a member of the groups granted more than “Everyone” access to the application’s installation directory. (See “Application Directories” in *Application & User Home Directories*.) You may encounter errors if the application’s programs cannot write a file, directory, or Registry hive. These errors usually do not occur when you run the programs from an account that is a member of the App Installers group.

If you wish general users to use the features prevented by such errors, you must give “W” and perhaps other permissions to the ACLs on the offending objects. However, a word of caution: this may allow one user to affect or compromise the work of another. Although it may be difficult to detect these situations, you should attempt to do so. You may be able to give expanded permissions to only the users you trust to not abuse them.

You can use the security log to determine which files or Registry keys are producing these errors. Set the auditing information on the file system and Registry trees to record “write” modification failures to the account that’s producing the errors. (You need the Right to “manage auditing & security log” to do so.) Make sure the audit policy enables failures in the “File and Object Access” category, although only full administrators can modify this policy. Examine the security log for the failures. There are three trees where you are most likely to find objects that cause these errors:

- The application’s main installation directory tree.
- The system root directory tree, typically named WINNT.
- The tree emanating from a Registry key inside HKEY\_LOCAL\_MACHINE\Software and named for the company or application, which Windows NT applications often create to hold user-independent data.

### ***A Modest Change to Tighten WINNT ACL's***

Windows NT’s strategy for protecting its critical system root directory (usually C:\WINNT) and its SYSTEM32 and other subdirectories is to give Everyone “Change” (read, write, delete, and execute) access to these directories but only Read access to most of their files. This lets everyone create new files and subdirectories, but not write or replace files installed by Windows NT.

Unfortunately, when users (most notably system administrators) create new files, Everyone now has Change access to the new items. The most critical case is where administrators install new applications, particularly administrative applications. It's common for these applications to place executable programs (\*.EXE) or libraries (\*.DLL) files in these directories, and these can now be compromised by any malicious user with local or remote access to the computer.

The heart of the problem is the Change access given to Everyone on directories. While the recommendations in earlier in this chapter cure this problem, a simpler change to the Everyone entry in these directory ACL's can quickly alleviate this problem with few potential application conflicts:

```
Everyone: *RWXD/RX
```

CREATOR OWNER, which should have Full Control, and other entries need not change. This change alone protects newly created files. The "\*" indicates that you should use a tool that lets you place directory permissions that do not propagate to newly created subdirectories, like Super CACLS. This protects newly created directories.

User applications who create items in these directories can fully access those items on behalf of the creating user, but all other non-administrative users gain only read access. The only case where you might see compatibility problems is where one user installs programs that must be altered by other users of the application, but this should be discouraged on secure sites anyway.

If you have an existing WINNT directory that may have been corrupted, you should first search for files in the WINNT directory tree owned by the Administrators group where Everyone had more than RX permissions. Remove all but the RX permissions. For example, if you use Super CACLS to remove these:

```
MODACL c:\winnt /s /f /om Administrators /rp Everyone:WDPO
```

You can then search for files and directories owned by other than Administrators where Everyone has access greater than RX, but you'll have to fix them on a case by case basis. Again, you can search using the Super CACLS command:

```
PRACL c:\winnt /s /v /xom Administrators /am "Everyone:>RX"
```

Note that some sites routinely replace Everyone with Authenticated Users (although they are equivalent for the file system). If you do so, simply read "Authenticated Users" in place of "Everyone" in the above discussion.

### *Limiting Installers in the WINNT & Registry ACL Settings*

The NTFS and Registry ACL settings recommended in Chapter 7 result in the "Installers" group having Change access to newly installed items in WINNT and other sensitive directories and the Registry. They can therefore modify sensitive elements installed by Administrators and, if industrious, likely make themselves administrators. While installers are trusted to a degree, it does not seem prudent to allow them to make themselves full administrators by subterfuge.

The **first option** is to change the Installer's permissions from "Change" to:

```
Installers: *RWXD/notspec
```

CREATOR OWNER, which should have Full Control, and other entries need not change. "Notspec" means this ACL entry does not propagate to files and is a standard feature of the Windows NT window ACL manager. This change alone protects newly created files. The "\*"



indicates that you should use a tool that lets you place directory permissions that do not propagate to newly created subdirectories, like Super CACLS, and protects newly created directories. These new setting means that installation accounts cannot modify or replace items installed by other installation accounts, which is a drawback and the reason we did not recommend this in the main tables.

The **second option** for Administrators to adjust items they add to remove the Installers group. Again using Super CACLS:

```
modacl c:\WINNT /s /om Administrators /de Installers
```

and similarly for the Registry. This is probably a more workable solution because it lets Installers modify other Installers' files.

### ***Tightening Service Pack ACL's***

Windows NT 4.0 sets up the all-important system root directory (usually C:\WINNT) and many of its subdirectories (like SYSTEM32) to allow Everyone "Change" (RWXD) access. This lets programs run by all users freely create files and new subdirectories, which maximizes application compatibility. By contrast, the files (not directories) that Windows NT installs in these directories usually allow only Read permission for Everyone. This properly prevents users altering or replacing the files with untrusted or malicious code.

When you install a Service Pack, it creates many new files in these directories (mostly \*.EXE and \*.DLL) that accept the default ACL (Everyone "Change"), which means they can be altered or replaced by all users. This appears contrary to 4.0's otherwise prudent policy of protecting such files from user alteration. (Similarly for \*.INF files in WINNT\INF.)

The solution is to search for \*.EXE and \*.DLL files in the WINNT directory tree owned by the Administrators group. Remove the Write and Delete permissions from the Everyone group. Similarly, change \*.INI files in WINNT\INI. You can do this by hand, or using a tool like Super CACLS:

```
modacl C:\WINNT /s /f /nm *.EXE *.DLL /om Administrators
/rp Everyone:WD
```

### ***Software Installation that "Moves" New Items into Protected Directories***

If you've taken prudent steps to set the ACL's on the various directories into which new applications install files and directories, you can help assure that the installed items have proper ACL's when the installation itself does not explicitly set these ACL's (and few do). However, some installation scripts can thwart your efforts. If the installation first places the items in a directory on the same logical drive as their final resting place, then "moves" them into the final directories (whose ACL's are strategically set to give the newly created items a proper ACL), the items carry the ACL of the original directory with them. This is because when you "move" a file or directory to another location on the same logical drive, it's ACL does not change – it is not given the ACL it would get if it were simply created in the new location.

The only protection is to scan the ACL's of the final directory trees looking for ACL's that are not proper. If you don't know which directories to scan, try the system root, usually C:\WINNT, and the application directories specified during the installation. You can focus your search using any technique at your disposal to determine the list of installed files and directories. For example, you can set per-object auditing on these trees to audit creations by the installing user, then check the Security Log. If you are not already using per-object

auditing on these trees, the standard window ACL manager can easily apply and remove the ACL's. Otherwise, you'll need a more selective tool.

**Related Guidelines:**

*Application & User Home Directories*

*Spoofing, AUTOEXEC.BAT use in "System-Wide 'PATH' and Other Environment Variables"*

**References:**

A companion document to this study summarizes these ACL settings against several other popular configurations.

[Sutt96] Chapter 4, *ACLs*, p. 159.

[Navy97] [Micr97]

These both present ACL configurations based upon and quite close to the C2CONFIG tool.

[RKitS] Chapter 24, *Registry Editor and Registry Administration*. A general overview of managing the Registry.

## 8. Application & User Home Directories

---

Because malicious applications can cause considerable damage, it is important that you tightly control their introduction onto the system, protect their various executable and data files from unwarranted change, and attempt to assure that users with sensitive capabilities don't run malicious applications. (An application that administrators trust to not be malicious is aptly called a "trusted" application.) To further these goals, store major, third party applications into "official" application directories under the control of trusted users. This affords a more consistent use of ACLs.

Directories that hold collections of miscellaneous, unrelated programs (\*.EXE files) are often called "BIN" directories. Consider a BIN directory as an application directory. The main difference is that you seldom create per-program subdirectories within BIN directories.

The App Installers group is given special access to these directories so they can add new applications. This is related to their abilities to modify installation items in the WINNT directory and Registry.

A new "App Users" group helps prevent sensitive users, like full administrators, from running applications unless and until those applications are deemed trustworthy. App Users includes all users allowed to use local applications *except* sensitive administrative accounts.

See the discussion of the App Installers group in the overview for *File System & Registry ACL Settings*.

### Guidelines

#### *Application Directories*

##### **Levels 1 & 2:**

The following practices are *recommended*. (Note that the App Installer group is also *recommended*, not prescribed.)

- ❑ Create one or perhaps a few official application directories, for example, a single one named "C:\Program Files," which is the standard default upon installation of Windows NT. Minimize the number of application directories. You typically install each application into its own subdirectory of one of these application directories, although some application suites share a single subdirectory.
- ❑ Give these directories the following ACL (including the files and directories in their entire tree, if already installed):

##### **C:\Program Files** ...and/or other program directories

App Users	Read [1]
App Installers	Change
Administrators	Full Control [2]
SYSTEM	Full Control [2]
Server Operators	Full Control [2]

[1] "App Users" refers to the group whose members are to use the applications. Because we also preclude administrators from executing \*.EXE files, we cannot use any group of which administrators are a member, like Everyone or Users.

[2] Use Server Operators on domain controllers only. Do not replace this with Power Users on NT Workstations.

- ❑ For any pre-installed application trees, change all \*.EXE files to remove “R” permission for App Users. Application installers should do this after installing each application. See “Removing ‘R’ from Program Files” in *Spoofing*.
- ❑ For any pre-installed application trees, change all \*.EXE files to remove “X” permission for App Installers, Administrators, Server Operators, and SYSTEM. Application installers should do this after installing and testing each application. Any \*.EXE files that you deem “trusted” can be exempted if it is important that these administrative groups use the programs.

You need not otherwise change any of the ACLs in a newly installed application directory tree unless the application requires it. See “Installing & Testing New Applications” in *File System & Registry ACL Settings*.

### *User Home Directories*

Each user who logs onto a computer and who requires local file system workspace under their personal control should usually be given a home directory on that computer. The general public should not have access to a home directory before its owner has had an opportunity to personalize its ACL structure.

We also address a home directory shared among users. The essence of sharing is that by default each such user has limited expectation of protection from the others who share the directory. So long as these users understand this, shared home directories are not precluded even at Level 2.

#### **Levels 1 & 2:**

The following practices are *recommended*.

- ❑ Create a directory named USERS on any logical drive to hold the home directories with the same ACL as recommended for the drive’s root directory (see *File System & Registry ACL Settings*). C:\USERS is a common location.
- ❑ Create a directory for each user with a directory name based on the user’s logon name. (Note that there can be users from different domains with the same account name, and you may need to distinguish between these, typically by prefixing the domain name where conflicts exist.) Apply the following ACL, where “*user name*” is the user account name.

```
C:\Users\user name    ... and/or other user home directories
  user name           Full Control
  CREATOR/OWNER      Full Control
  User Data Managers (opt) Full Control
  Administrators      Full Control
  SYSTEM              Full Control
```

User Data Managers are optional and represent any group that by policy are given by default full access to user’s data. You can also use groups like Server Operators. Users can subsequently exclude this group. Do not give such groups the Right to take ownership because it would allow them to change all files on the system.

- ❑ If a home directory is to be shared by users, replace the first ACL entry above by an entry of the following form for each user.

```
user name           Special Access (RWXD)*(Not Specified)
```

This allows each such user to create new items that by default grant no access to the other users. (The "\*" indicates that this entry should be installed as a non-propagating entry if you have the tools to do so.)

- ❑ If the directory is to be shared by a group, use this same entry substituting the group as the *user\_account* with the following permissions:

*group name*                      Special Access (RWXD)\*(Not Specified)

This allows each user in the group to create new items that by default grant no access to the other users.

- ❑ Install the home directory in the user's User Manager account.
- ❑ As an option, create a USERS\DEFAULT directory with the Public Directory ACL for users who have no home directory on this computer. This directory is much the same as the C:\TEMP except that users have a greater expectation of permanence.
- ❑ Alternately, you can create a home directory on a remote share directory. Use these same ACL techniques.

The guidelines promulgate the standard practice that allows the Administrators and SYSTEM groups full control to all files and directories inside user home directories. Users can remove these groups from their own ACLs, but it may tend to make operations more difficult by forcing administrators to take ownership under exceptional cases where they need to sort out problems in user directory trees.

#### **Periodic Review:**

The reviewer should formulate a site policy for user home directories, then review all user home directories looking for unsafe practices commensurate with that policy. Following are examples that might be included in such a policy:

- ❑ Personal files that contain "secret" information, like passwords to various applications, must be properly protected, or, preferably, removed. Under no circumstances should users store any Windows NT logon password in a file.
- ❑ Enforce the site policy as to whether non-administrative users may import programs onto the system. If they are allowed, the executable files and their DLL's must be protected from modification.
- ❑ Check for cases where individuals are not properly protecting their home directory trees.
- ❑ Check for cases where individuals are sharing a home directory but that one uses it to store data for which the others are not authorized by site policy.

#### **Related Guidelines:**

*File System & Registry ACL Setting*, including "Installing & Testing New Applications"

*Spoofing*, the damage that malicious applications can do

*User Rights*, the Right to Bypass Traverse Checking

UNCLASSIFIED

## 9. User Accounts & Groups

---

This guideline includes recommendations for the various settings in the user accounts, and a few general recommendations for creating user groups.

### Guidelines

#### *User Accounts*

This guideline includes various policies relating to user accounts defined in the various buttons along the bottom of the User Manager account window, except the policies covered by the Passwords guideline. The guidelines prescribe no security considerations for the Groups or Profile buttons. See *Remote Access Service* for a discussion of the “Dialin” button.

#### Level 1:

- ❑ Apply the following guidelines to the window buttons of each new account you create:
  - Profile:** Not prescribed for security. However, logon scripts and profile information should be accessible only by their user and administrators.
  - Hours:** Not prescribed. (See notes below.)
  - Logon To:** We *recommend* you set these reasonably tight, unless you have controlled logon using the Right to “log on locally.” (See notes below.)
  - Account:** The guidelines prescribe no specific lifetime for security reasons, but accounts with an intended lifetime limit should be set accordingly.
- ❑ Disable the Guest account (select “Account Disabled” on the main account window). For extra assurance, give it a long, random password that you do not retain, set its logon hours to none, allow it to log onto no workstations, and set its expiration date to a date past.

#### Level 2:

- ❑ Apply the account guidelines from Level 1, except:
  - Hours:** We *recommend* you set the allowed logon hours sensibly tight. However, even at Level 2, there is little reason to inconvenience your users significantly with unreasonably restrictive times.
  - Logon To:** Set these as tightly as possible, unless you have controlled logon using the Right to “log on locally.” (See “Logon Rights in Multidomain Environments” in *Domains & Basic Access Restrictions*).

#### Periodic Review:

- ❑ Check for enabled accounts that are no longer used.
- ❑ Assure that “Logon to” limitations are sufficiently tight.

## *User Groups*

This short guideline covers user groups other than administrative groups and common groups like Users and Domain Users (see Logon Rights in Multidomain Environments” in *Domains & Basic Access Restrictions*).

### **Levels 1 & 2:**

- ❑ Do not add users or global groups to the Replicator group on domain controllers.
- ❑ The “Guest” group does not enable unauthenticated logon as does the Guest account. This group therefore requires no special attention. Like any general group, whether and how you use it depends only upon who your users (who might apply it to ACLs) assume its members are. Define and distribute your site policy for using the Guest group.
- ❑ Note that members of the Users local group can create their own groups. If you wish to limit those who can create groups, remove them from Users (although they may then have difficulty accessing certain objects). Creating groups does expand one’s access capabilities, and the guidelines do not prescribe how you limit its use.
- ❑ You can use groups to select large collections of users for security controls, for example logon hours, Rights, and to forcibly renew passwords. While these guidelines do not prescribe this technique, you may find it improves your ability to manage account security parameters.

### **Periodic Review:**

- ❑ Continually monitor the members of various groups for members that system users would not expect to be in those groups. The problem is that users grant groups access to objects via their ACLs based on their understanding of who the group members are or might later include. For example, including users in a group named “BulldogProject” who were unconnected to the project is misleading.

## **Notes**

It is often more convenient to limit local logons through the “Logon to” button than the Rights policy because domain accounts are centrally managed on domain controllers, whereas the Rights policy must be maintained on each computer. (See *User Rights*.)

Limiting logon hours can serve to prevent users of minimal trust from using the system during times where their activities cannot be monitored, but it is only useful when any such monitoring varies in its effectiveness from some times to others. While it can also attempt to prevent the use of a stolen password, we do not judge this to be of significant security value. All things considered, at most sites we do not feel logon hours are an effective protection against attack, and suggest that you readily forego logon hours in favor of operational considerations, even at Level 2.

One might seek to limit account lifetime on the chance that administrators don’t properly manage or periodically review the accounts database. For example, administrators might forget to remove a user that’s left the site. However, any reasonable limitation could still allow considerable time for an attack on such an account, and there are better remedies for lax review practices.



**Related Guidelines:**

*Passwords*

*User Rights*, for local and remote logon Rights

“Logon Rights in Multidomain Environments” in *Domains & Basic Access Restrictions*  
for general use of logon Rights

The “Administrator” Account in *Administrative Structure*

**References:**

[Sutt96] “Accounts” in Chapter 7, *Managing Groups and Accounts*, p. 161-168.

**UNCLASSIFIED**

## 10. Passwords

---

This guideline recommends various parameters that govern the use of user and administrative passwords. Account locking is prescribed at both Levels 1 and 2. This chapter describes example password schemes and quantifies the probability of guessing these schemes. However, site requirements on password parameters of complexity and lifetime are too diverse to attempt an explicit recommendation. Rather, each site must formulate its own policies based on a variety of factors, most notably whether its network is exposed.

### Guidelines

#### Levels 1 & 2:

##### *Password Complexity and Lifetime*

Each site must formulate its own criteria for password complexity (“Minimum Password Length” in the Account Policy) and lifetime (“Maximum Password Age”). You should determine these against an estimated probability of password compromise as exemplified in “Login Attempt Attacks” and “Captured Password Attacks,” below. The important parameter is the probability that a password can be guessed during its lifetime – as opposed to simple password length.

##### *Password Locking*

Set password locking in the Account Policy. Passwords whose complexity are designed for Login Attempt attacks (below) should have the locking parameters from those tables. We *recommend* passwords whose complexity are designed for Captured Password attack should have the following values:

Lockout after	5	bad logon attempts
Reset count after	30	minutes
Lockout duration	30	minutes

Passwords designed for Captured Password attack are far more complex than those designed for Logon Attempt attacks, and the lock is not significant in defeating attacks. It is therefore less important to set its parameters tightly. However, there is considerable discretion in these values.

##### *Guidelines for Users who Define their own Passwords*

Set the following parameters for users who select their own passwords:

- Set “Password Uniqueness” in the Account Policy to remember the maximum value (currently 24).
- Set Minimum Password Age in the Account Policy to 2 days. This discourages users from cycling through a list of passwords to reach an old favorite. (There is considerable flexibility in the value you choose, but 2 days means that 48 days must pass before a user can return to a favorite password, and even then only after working through 23 other passwords.)
- When you create a new account for such a user, select “User Must Change Password at Next Logon” (but note the point that follows).

- ❑ Use the “User must log on in order to change password” to prevent users from getting a new password once their old one expires. However, this also prevents them from changing their password on first logon as per the previous point.
- ❑ We *recommend* you set “Forcibly disconnect remote users from server when logon hours expire” options. This prevents users from establishing new connections outside their logon hours, although it does not terminate connections existing when their logon hours expire.)
- ❑ Formulate your policy for matching account passwords as described in “Passwords for Local, Matching Accounts,” below.
- ❑ We *recommend* you run a selection of the Windows NT “password guessing” programs regularly that attempt to detect weak users.<sup>19</sup> (There are no such programs in the delivered version of Windows NT, although 3<sup>rd</sup> party vendors may provide them.<sup>20</sup>)

### ***Administratively Defined Passwords***

You can choose to let administrators select passwords for all or certain users. Careless users can choose weak passwords no matter what you do, and the only sound way to keep their passwords strong is for you to assign them. You can decide the password together with the user to assure that it is both easily remembered and judicious. This is *recommended* at Level 2 unless it contradicts site policy. Select the “User Cannot Change Password” option in the main account window for each such user.

Some sites have a policy that administrators should not know users’ passwords. These guidelines do not dispute such a policy. However, users have almost no protection against a full administrator, and little against an Account Operator.

### ***Password Filtering***

The guidelines *recommend* you implement a custom password filter that aggressively enforces your local password policy. See *Password Filtering* below. Unfortunately, you must now either write your own or acquire one from a trusted third party. We neither recommend nor discourage the Windows NT filter named “PASSFILT.”

### ***Password Warning Time***

By default, Windows NT begins warning a user 14 days before their password expires. You can change this value by adding a REG\_DWORD value named “PasswordExpiryWarning,” whose value specifies the number of days before expiration, to the following Registry key:

```
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\
CurrentVersion\Winlogon
```

### **Periodic Review:**

The most important review is to assess the randomness of user-chosen passwords.

<sup>19</sup> One potential disadvantage of using these password checkers is that if a penetrator knows you are using one, they can eliminate all of its disallowed passwords from their brute force attacks, which may significantly increase their chance of success.

<sup>20</sup> A tool called “L0phtcrack” is quite popular. You can find a lot about it on the Web.

## Notes

### *Logon Attempt Attacks*

The probability that a penetrator can guess a password depends upon how quickly they can attempt guesses on the password and its length. The guessing rate depends on the kind of attack and we include the two more common attacks: logon attempts, and captured password.

In a logon attempts attack, the attacker attempts to guess a password by attempting a legitimate logon, either primary logon from the keyboard, or secondary logon from a network source. Our assumption is that account lockout is enforced. Hence the lockout parameters are the dominant factor in the rate of successfully guessing a password and therefore the probability that it can be guessed.

The Logon Attempt Password Class table below shows the probability of a penetrator guessing a password under this scenario over its lifetime at three successively stronger locking levels (the three right columns) for various kinds and lengths of passwords (the lower rows). The table includes five different kinds of passwords:

**Random, lower case alphabetic:** Randomly selected, lower case characters taken from the alphabet (“a” through “z”), such as “hqisgt” or “oxdye.”

**Random alphanumeric:** Similar but randomly drawn from the digits (0-9) and upper and lower case alphabetic characters, for example “Hs6tY8.”

**C-V-C-V-C:** “Pronounceable” passwords of the form consonant-vowel-consonant-vowel-consonant, from random, lower case alphabetic characters, like “misoq” or “paxun.”

**C-V-C-C-V-C:** Similar but with an extra consonant in the middle, for example “mistoq” or “paxlun.”

**Phrase Abbreviations:** Lower case passwords constructed by concatenating the first 1 or 2 characters from a nonsense phrase, like “afeyc” from the first characters of the words in phrase “allow fish eat yellow car.” (People may find it easier to remember a phrase that they can visualize.) These probabilities are rough because some alphabetic characters are more readily chosen than others. The table divides the theoretical space by two (admittedly a rough guess) to account for non-randomness.

These guidelines do not intend to limit you to these schemes. However, if you choose an alternative scheme you should calculate its probabilities and subject them to the guidelines that follow.

We assume the highest rate at which passwords can be guessed by logon is approximately one less than the locking threshold (the “Reset count after” value) divided by the reset time. This is accomplished by quickly guessing one less than the threshold (to avoid triggering the lock) then waiting for the count reset period. Lockout duration has no effect in this case because the penetrator never triggers the lock. From the standpoint of limiting the logon rates, you can select the smallest lockout duration allowed which is equal to the threshold.

The table illustrates the chances of guessing various lengths (the “Len” column) of these passwords with a locking threshold (“Lockout after”) of 7 and reset times (“Reset count after”) of 5, 30, and 60 minutes. A “probability” value such as 2,947 means the chances are “one in 2,946” that a penetrator will guess the password by attempting to logon locally or remotely at the maximum rate allowed by the locking parameters over the lifetime of the password, in this case one month. The “Space” column shows the total number of possible

combinations for the password scheme. You can adjust these to different password lifetimes by dividing the probability by the lifetime in months.

**Logon Attempt Password Class Table**

Password lifetime, months			1	1	1
Threshold ("Lockout after")			7	7	7
Lock duration ("Reset count after), min			5	30	60
	<b>Len</b>	<b>Space</b>	<b>Probability</b>	<b>Probability</b>	<b>Probability</b>
Random, lower alphabetic	4	5.E+05		57	113
	5	1.E+07	246	1,473	2,947
	6	3.E+08	6,385	38,308	76,616
	7	8.E+09	166,001	996,008	1,992,016
C-V-C-V-C		4.E+05		54	109
C-V-C-C-V-C		1.E+07	236	1,417	2,833
Random alphanumeric	4	1.E+07	305	1,832	3,665
	5	9.E+08	18,935	113,608	227,215
	6	6.E+10	1,173,947	7,043,680	14,087,360
	7	4.E+12	72,784,693	436,708,160	873,416,321
	10	8.E+17	2.E+13	1.E+14	2.E+14
	14	1.E+25	3.E+20	2.E+21	3.E+21
"fish ate blue house"	4	2.E+05		28	57
	5	6.E+06	123	737	1,473
	6	2.E+08	3,192	19,154	38,308
	7	4.E+09	83,001	498,004	996,008

These password classes are based on the assumption that attack occurs through Windows NT's normal local or remote logon process that triggers locking. The local Administrator account does not lock and the guidelines deal with it specially in "The 'Administrator' Account" in *Administrative Structure*. There is also a way to attack passwords through their local storage that can be dealt with by other means, although more complex passwords help (see "Protecting Hashed Passwords & SYSKEY" in *General Policies*).

### ***Captured Password Attacks***

Windows NT stores a hashed version of the password, and if one knows the Windows NT hashing algorithm (and it appears rather widely known) one can attempt a brute force attack. Captured password attacks have been shown to be effective for the case where the Windows NT hashed password is captured from local storage. Fortunately, this case can be protected. (See "Protecting Hashed Passwords & SYSKEY" in *General Policies*.)

However, there are cases where active or passive taps on a network can view network authentication traffic and apply brute-force or dictionary attacks on a hashed password. For example, suppose a challenge-response scheme sends the challenge and response as cleartext, and the response calculation is a known function of the user's password. Although the password is never passed across the network, a penetrator can carry out a brute force attack

knowing the challenge and its response. These attacks run at the speed of available computer power and can be quite high. There has been considerable public discussion about capturing passwords by eavesdropping on its authentication traffic.

A “password exposed” network is one prone to the “captured password” attack above. The first condition for a password-exposed network is that passwords are passed in a way that allows this type of attack, for example the challenge-response example cited above. The second is that malicious programs on the network can read raw packet traffic from the network. This happens when there are malicious programs on (1) operating systems that are not kernel protected, like DOS, Windows (except Windows NT) and the Macintosh OS, or (2) kernel-protected operating systems that are mis-administered or maliciously administered. These cover the case where an attacker can attach a computer of their choice to the network. The global Internet certainly qualifies as a password-exposed network.

The following table shows the probabilities that a captured password can be guessed during its lifetime. It uses a subset of the password types above, omitting those that are ineffective. It assumes a penetrator can carry out 1,000 guesses per second and a lifetime of 1 month. (Although some reviewers have notes that this rate could be much higher.) As before, a “probability” value such as 2,947 means the chances are “one in 2,946” that a penetrator will guess the password during the password’s lifetime. (A value of 0 means that the password will be successfully determined.) You can adjust these to different password lifetimes by dividing the probability by the lifetime in months. The guessing rate of 1,000 is an arbitrary and rough estimate, but one cannot even empirically estimate this rate unless the specific algorithm is known.

### Captured Password Class Table

(Password Lifetime: 1 month)

	Len	Space	Probability
Random, lower alphabetic	7	8.E+09	3
	9	5.E+12	2,095
	10	1.E+14	54,463
	11	4.E+15	1,416,028
	12	1.E+17	4.E+07
	14	6.E+19	2.5E+10
Random alphanumeric	7	4.E+12	1,359
	8	2.E+14	84,236
	9	1.E+16	5,222,641
	14	1.E+25	4.8E+15
“fish ate blue house”	9	3.E+12	1,047
	10	7.E+13	27,231
	11	2.E+15	708,014

Note that the LANMAN authentication protocol can greatly compromise password space. These tables assume this danger has been removed. See “LANMAN Passwords” in *Networking*.

### ***Example Policy A***

We now present the first of two simple password policies. A Level 2 site has an unsafe network situation and decides upon an aggressive policy for all passwords. It assumes the worst case as represented by the Captured Password Class Table, above, and requires that the probability of exposure of 1 in 1,000 or less, and a password lifetime of 3 months. This means that, for example, a user who selects random, lower case alphabetic passwords could use a length of 11. (This probability is  $2,832/3$  or only 944, but close enough to 1,000.) In recognition that passwords of this length cannot be easily remembered, it also develops guidelines for safely storing written down copies of the password.

### ***Example Policy B***

In this example a “safe” network site decides that administrative passwords should be more strongly protected than those of regular users. It divides the Logon Attempt Password Class Table (above) in 3 classes of passwords based on this probability:

- Class A** 100 through 1,000
- Class B** 1,000 through 100,000
- Class C** 100,000 or larger

Probabilities under 100 are regarded as too small, and over roughly 1,000,000 as excessively large. (That is, probabilities over 1,000,000 yield the risk of password guessing considerably smaller than other risk levels.)

The site configures its account policy and user password parameters according to the following password class requirements. This table shows the password class requirements that the site might impose were it a Level 1 or Level 2 site. Users may use any password scheme that meets the following minimum specifications. Note that strengths vary within a scheme and, all other considerations being equal, users should choose the strongest practical scheme within a given class.

	<b>Level 1</b>	<b>Level 2</b>
Regular Users	Class A	Class B
Moderate Administrators [1]	Class B	Class C
Full Administrators & Backup Operators [2]	Class B	Class C

[1] Includes Power Users, and Server, Account, and Print Operators.

[2] Except the local Administrator account.

The site sets the account locking Lockout Duration (not shown in the table) equal to the count reset time, the minimum the window allows.

### ***A Caveat on Network Password Exposure***

Windows NT has come under criticism for its network password exposure. Windows NT does not itself encrypt networking traffic. While it provides modest network protection for its passwords, its techniques are vulnerable to modern attacks. Because Windows NT passwords are susceptible to brute force attacks from the network, their only protection is their password space. If your network is exposed to malicious elements, and you cannot assure that your users will select long, random passwords, you should encrypt *all* Windows NT network



traffic. This can certainly solve the password exposure problem. And after all, if your network has hostile elements would you not want to protect all networked data cryptographically? (You'll need to seek third party products. See *Networking*.)

### ***Passwords for Local, Matching Accounts***

A user who works in their "normal" account may need access (for example, connecting to a share directory) to a remote computer whose scope does not include their normal account. The most common case is when the remote resource is in a different domain without an allowing trust relationship. The most transparent way to accommodate this user is to create a local account on the remote computer with the same account name as the user's normal account with the same password. (See "Domains, Trusts & the Scope of Accounts" in *Domains & Basic Access Restrictions*.)

At issue is whether it is safe to use the same password for both their regular and this "matching" account. If the password is compromised on the remote system, it can be used to access any computers onto which the normal user account can locally or remotely log on. Consider this safe unless the remote computer has inadequate physical protection so that it poses an inordinate risk of password exposure as described by "Protecting Hashed Passwords & SYSKEY" in *General Policies*.

Note that if users can change their own passwords, they must maintain any policy upon which you decide. You can of course prevent them from changing their password in either their normal or the local, matching account.

### ***Password Filtering***

Administrators can install special programs that reject a user's new password based on defined criteria. These programs must be cast as DLL's. Microsoft provides one such DLL named "PASSFILT" that requires passwords at least 6 characters long with restrictions on the characters in the password. These restrictions make PASSFILT's password space (the total number of possible passwords) less than a scheme that uses randomly chosen characters, and PASSFILT does not prevent simple passwords like "Frog00." These guidelines do not recommend (but do not discourage) PASSFILT because it causes conscientious users (who would otherwise choose random passwords) to use slightly longer password lengths to afford the same protection and does not prevent simplistic passwords. In this manner, PASSFILT falls victim to the misconception that restricting password characters increases the resistance to being guessed, when in fact it may do the opposite.

You must activate these DLL's by adding their DLL name (without the ".DLL" suffix) to the multistring value named "Notification Packages" in the Registry key:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa
```

which defines all such DLL's. Install such DLL's in the system root directory and assure that at most, administrators and the SYSTEM user can modify or replace it. Note that the DLL is given the user's unencrypted password and must handle it carefully to prevent accidental exposure.

"Notification Packages" is also addressed by "Password Notification Feature" in *General Policies*.

### ***Summary***

In summary, there are three components to protecting Windows NT passwords:

- ❑ Attacks through Windows NT normal authentication mechanisms can be effectively countered by reasonable password complexity and account locking.
- ❑ Attacks that work against the hashed passwords stored in the SAM database (or copies thereof). There are effective ways to combat these. See “Protecting Hashed Passwords & SYSKEY” in *General Policies*.
- ❑ Attacks that eavesdrop on Windows NT’s authentication mechanisms. Your most complete protection is to seek third party tools that encrypt all Windows NT networking traffic. See *Networking*.

Password policy is not a precise science, and there are many ways you may improve upon these guidelines based on your site and its users. One essential consideration is that long, complex passwords are harder to guess, but they are more likely to be written down which may be an unwarranted exposure. Password complexity is therefore a trade-off.

If you require higher probabilities in the Logon Attempt Password Class Table, we recommend you increase the locking reset time rather than requiring your users to remember longer passwords.

Requiring a user to select their first password “at next logon” is not a strong security control. Conscientious users will do so anyway and negligent users choose weak passwords no matter what you do. A user’s password is no protection against full administrators and Account Operators, and little protection against others. However, you can use this feature to force users to change their passwords if you suspect they may have been compromised.

Some password schemes limit a password’s space (the total number of possible combinations) by restricting the password’s characters, for example, by requiring two digits (0-9) in the password. These modestly penalize conscientious users by making them choose longer passwords to achieve the same space, yet do nothing to enforce proper passwords on users bent on circumventing the intent. For example, a policy that requires two upper case, two lower case, and two digits is satisfied by “AAaa00” – hardly a strong password. However, these schemes may emphasize to middle-of-the-road users the importance of choosing random passwords. The choice to use such schemes is subjective, but you should calculate the reduction in password space before making your decision.

#### **Related Guidelines:**

“Protecting Hashed Passwords & SYSKEY” in *General Policies*

#### **References:**

- [Sutt96] “Choosing Passwords” in Chapter 3, *Your Working Environment*, p. 47.
- [Sutt96] “Accounts” in Chapter 7, *Managing Groups and Accounts*, p. 161. Especially topics “The Account Policy,” p. 168, and “Your Passwords,” p. 172.

# 11. System Policy Files

---

Consult your system documentation, the Notes below, and chapter 10, *System Policy Editor*, in [Sutt96] for an overall description of system policies. Setting up System Policy Files is an involved process and you need to consult the references carefully.

These guidelines recommend a simple, straightforward domain-based policy file strategy at Level 2. You may of course expand upon this simple scheme, but this guideline does not prescribe it. (See “Compelling uses of Policy Files” below.) The two basic modes of using the policy file are called “Automatic Update Mode” and “Manual Update Mode.” (The section “Automatic versus Manual Update Mode” below describes the criteria for choosing between the two.)

## Guidelines

### Levels 1 & 2:

The guidelines *recommend* installing a System Policy File with the following settings. Many of these system policies represent modest security controls in most environments, and most can be imposed by other means.

- ❑ If you choose to use Automatic Update Mode, create and install an NTCONFIG.POL file in the NETLOGON share on each domain controller with the settings below. For Manual Update Mode create and install the file at the chosen location.
- ❑ Refer to the section “Recommended Default User Policies for non-Administrative Users,” following. Set the policies marked “✓.” Those marked with a “?” need further consideration but are not prescribed by this guideline (although other guidelines might). The guidelines recommend against those marked “X.”
- ❑ Policy files should be owned by Administrators with the following ACL:
 

Everyone	Read
Administrators	Full Control
SYSTEM	Full Control
- ❑ Unless your site demands otherwise, in Automatic Update Mode assure that the same set of computer policies is defined within each domain family. (See “Automatic versus Manual Update Mode,” below).
- ❑ Confirm that the “Remote update” policy is set in each computer’s Registry. This defaults to Automatic Update Mode upon installation. Use the System Policy Editor to set the “Remote update” policy, or any Registry editor on the key:

HKEY\_LOCAL\_MACHINE\System\CurrentControlSet\Control\Update

setting its “UpdateMode” value to:

UpdateMode: REG_DWORD: 0x1	for Automatic Update Mode
UpdateMode: REG_DWORD: 0x0	for Manual Update Mode

### Periodic Review:

- ❑ Assure that policy files reflect current site policy.
- ❑ Assure that policies are enforced throughout their intended scope. (Specifically, that “Remote update” is enabled on each computer under policy control.)

- ❑ Assure that the NETLOGON share and its policy files on domain controllers are properly protected against public access.
- ❑ Justify the case where computer policies in different domains do not define the same set of policies.

## Notes

### *Recommended Default User Policies for non-Administrative Users*

#### Display

Restrict “Display” Control Panel

##### ? Hide Screen Saver Page

This is a useful way to prevent users from changing a preinstalled locking screen saver. However, this does not protect the Registry entries that define the screen saver itself. Therefore, hiding this panel does not prevent a user from setting their screen saver using other tools or editing the Registry directly.

#### Shell

Restrictions

##### ? Remove Run Command

There are many ways for users to run arbitrary commands that removing this one makes little sense. Also, the Run Command can be a useful counter to certain spoofs and keeping it on the Start Menu encourages people to use it. (See “The ‘.’ Issue” in *Spoofing*.) Unless you have prevented all other means of running commands (which is unlikely), there is little point in enforcing this policy.

##### ✓ Disable ShutDown Command

Assures that shutdown is done using the security window (“trusted path”) – a safer alternative.

#### Windows NT Shell

Restrictions

##### ✓ Only use approved shell extensions

This policy reduces potential spoofing attacks and is particularly important for administrators.

#### System

Restrictions

##### ✓ Disable Registry editing tools

While this prevents the use of specific tools, it does not prevent one from accessing the Registry using other tools or custom programs. While prudent in some cases, this affords little protection against a determined attack. See “Enabling the Registry Editors” in *General Policies*.

##### ? Only run allowed Windows applications

Easily circumvented. See “User Application Restrictions” below.

## ***Recommended Default Computer Policies***

### **Network**

System policies update

- ✓ Remote update

Set the Automatic or Manual mode as per your strategy. Select Display Error Messages, which prints a message during logon if the policy file is not available on the network. There's actually little need to define this policy because it must properly be set on a given computer before that computer can even load a policy file.

### **Windows NT Networking**

Sharing

- ? Create hidden drive shares (workstation)
- ? Create hidden drive shares (server)

See "Hidden, Administrative Shares" in *Network Sharing*.

### **Windows NT System**

- ✓ Logon Banner

This presents a window during user logon that you can use for one of two purposes depending on your site policy: (1) an official warning (a "legal notice") about site policy regarding the use of this computer, or (2) a security note to users upon which they can rely. Set the "Caption," the window title, and the "Text," the text displayed within the window. As a security feature, the first use may serve as a minor deterrent against some potential penetrators. In addition, most sites regard this as evidence in legal recourse against unauthorized use.

### **Windows NT Remote Access**

- X Max number of authentication retries
- X Max time limit for authentication
- X Auto Disconnect

See *Remote Access Service*. Setting these are a normal part of installing RAS on a computer and don't change once installed. These have minor security importance.

You may wish to set some of the policies not checked but for only certain users. For example, "Disable Registry editing tools" might be set for everyone except certain administrators. Or you might want to install certain "wallpaper" (screen background) for administrators. This requires that you create a system policy group for administrators so they can be treated specially. (Note that many system policies, like wallpaper settings, are not enforced and the user can freely change them once logged on. See "Protected User Policies" below.) See the general references, below.

## ***Compelling Use of Policy Files***

All of the "computer" policies can be enforced (for a given computer) independent of the logon account by setting them in the Registry using a Registry editor or a specific administrative program. The only absolute reason to define these in a policy file is when these computer policies must vary among logon accounts. Also, it can be somewhat easier to define computer policies using a policy file because these policies are automatically set at

each user logon. Hence, when you modify a computer policy in a policy file, it will sooner or later propagate to all the computers as users logon to them.

“User” policies can also be enforced without policy files for policies that apply to all users. However, they are difficult to apply without policy files because the user policies reside in the user profile, which are difficult for the administrator to change. When different policies are to apply to different users, the policy file is the most practical option.

### ***Automatic versus Manual Update Mode***

In **Automatic Update Mode** (as set in the Update Mode policy), computer as well as user policies are taken from the domain controller of the user’s domain account. This means that different computer policies may apply to different users at the same computer. While not inherently wrong, administrators new to policy files might think that the same computer policies always apply to a given computer. Also, logons to local accounts are not subject to the policy files under Automatic Update Mode.

If it is important that the same computer and user policies apply to *all* users of a given computer, regardless of their account’s domain, you can use the **Manual Update Mode** (as set in the Update Mode policy) on that computer. In Manual Update Mode, you specify the location of the policy file, either on the computer itself or on a network share directory. (The domain controller’s NETLOGON share is one convenient location.) Manual Update Mode applies policies to all logons, including local accounts. Neither Manual nor Automatic Update Mode is inherently less secure than the other, but one may be more appropriate than the other for a specific computer.

### ***User Application Restrictions***

The policy entitled “Only run allowed Windows applications” lets the administrator define a set of applications (like “EXPLORER.EXE” or “WINWORD.EXE”) that are the only set a user, to which the policy applies, can run using the desktop manager and its companion Explorer. (Indeed, even Explorer must be on the list before it can be run.) It also prevents them from being run as Startup apps and from the Security Window’s Task Manager. However, there are several cautions:

- ❑ Many applications have “run” options that let users start other programs from inside the application. These applications do not customarily enforce these restrictions. In particular, the DOS command window does not enforce the list. However, you can prevent users from running applications with run options by excluding them from the list (like CMD.EXE for the DOS command window).
- ❑ If a user can read a disallowed, executable file, they can make a copy in another directory then rename the file to an allowed name and run it under the new name. You can prevent this by removing “R” permission from these files, leaving “X.” (See “Removing ‘R’ from Program Files” in *Spoofing*.)
- ❑ A user can import any program onto the system (most notably one that lets them run other programs of their choice), name the program with an allowed name, then run it.

As a security control, this feature is weak at best and the guidelines do not therefore prescribe its use. Setting the ACLs on application program files is more effective. However, you may find this useful in “discouraging” users from running programs they should not.

Note that a user’s powers come from their account and their logon session – not from the programs they run. For example, prohibiting a user from using a Registry editor does not

prevent them from accessing the Registry – it just means they must do so using some other program, perhaps one they created.

### ***Protected User Policies***

The Registry key:

```
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies
```

is read-only to the current user. Policies enforced by items in this key are the only ones that the user cannot change themselves.<sup>21</sup> All the policies recommended in “Recommended Default User Policies for non-Administrative Users,” above, are in this key.

### ***Custom Policies***

One can add policies to the System Policy editor and its files by creating custom \*.ADM files. We know of no product documentation that describes their format, although you can learn it by studying the installed files. You must install these in the System Policy Editor as Policy Template options.

### **References:**

- [Sutt96] The section “System Policy Editor” in Chapter 10, *Subsystems and Other Security Features*, p. 261.
- [ConPln] Section “System Policy” in Chapter 3, *Managing User Work Environments*.

---

<sup>21</sup> However, this portion of the Registry can be loaded from the \*.DAT files in the user’s WINNT profile directory to which the user can gain write access, and can potentially change. (This is a bit complicated because the system holds the current user’s \*.DAT file open for exclusive access, so a user must change it when logged in from another account.) Note that Security Policies (from Security Policy Files) are written after loading the user’s profile, and its policies would therefore overwrite any such changes by the user.

**UNCLASSIFIED**



## 12. User Rights

---

Each Windows NT computer has a local “Rights Policy” that associates the various groups and accounts usable on that computer with each of about 30 Rights. Rights are not a property of the user or group account, and an account may have different Rights on different computers. Full administrators manage a computer’s Rights policy using the User Manager.

When Windows NT creates a local (primary) or remote (secondary) logon session, it attaches the Rights associated with the user’s account and all the user’s groups to the logon session. These Rights are inherited by programs that run within the session and to many servers from which those programs request services. Rights allow programs special capabilities. A logon session’s Rights are unaffected by changes to the Rights policy during the life of the session. Together, a user account’s identity, group memberships, and Rights determine their capabilities on the system.

The system default Rights policy is sensibly secure and the guidelines prescribe few modifications. Most Rights are of use only to the operating system itself and assigned to few accounts.

### Guidelines

#### Levels 1 & 2:

The system default Rights policy is sensibly secure and we list few modifications.

- ❑ We *recommend* you replace “Everyone” (if it exists) with “Users” in the Rights to “**Log on locally**” and “**Access this computer from the network.**” Your site policy may warrant further restricting these Rights as described in “Logon Rights in Multidomain Environments” in *Domains & Basic Access Restrictions*. Everyone and Users typically imply almost the same accounts. However, allowing Everyone logon or remote access seems too broad on general principle. Note that this precludes remote access by the Guest account because by default it is not a member of Users.
- ❑ Remove all users and groups from the Right named “**Bypass traverse checking.**” There is nothing inherently unsecure about assigning users the Right to “bypass traverse checking,” so long as all system users understand that closing a directory’s ACL to others does not necessarily deny access to its file and subdirectories. Under the opinion that users tend to believe otherwise, and that bypass traverse checking is seldom used or needed,<sup>22</sup> the guidelines recommend you assign it to no one. You could safely assign this Right to administrative users who likely have broad ACL Rights anyway.
- ❑ We *recommend* you add Server Operators and Power Users to the Rights to “**Increase scheduling priority.**” This is a benign security capability and granting it more widely helps decrease use of full administrative accounts.
- ❑ Remove the “**backup**” and “**restore**” Rights from Server Operators. These extremely sensitive Rights are best restricted to accounts used *only* for backup and restore, normally Backup Operators.

---

<sup>22</sup> There is at least one report that removing this Right from certain accounts causes the operating system to crash (the infamous “blue screen of death”). We could not confirm this at this time.

Except as described in these guidelines, do not expand the allocation of Rights unless you fully understand the ramifications. Many Rights allow their user sessions unlimited access to the system.

## Notes

### *Common Rights*

The following summarizes several Rights mentioned in the guidelines:

#### **Log on locally**

##### **Access this computer from the network**

An account must have the Right to “log on locally” before the system grants it local (primary) logon, and to “access computer from the network” before the system creates a remote (secondary) logon session for the account. These two accounts together are an effective way to control who can use which computers.

Note that some 3<sup>rd</sup> party services might not rely upon Windows NT’s native remote authentication mechanisms, and may provide services to remote users who do not have the Right to access the computer from the network. One would hope such a service has a user-based control equally as strong. Deal with these services on a case-by-case basis.

#### **Backup files and directories**

##### **Restore files and directories**

These are among the most powerful Rights and allow programs to override ACLs for reading and writing, respectively. With standard Windows NT tools, it is difficult for programs holding these privileges to directly read and modify files that ACLs would preclude, but creating special programs to do so are within the capabilities of most administrators and penetrators.

#### **Take ownership of files and directories**

This powerful Right lets you make your account the owner of another user’s objects (like files, directories, and Registry keys). Once the owner, you can change the ACL to give yourself unlimited access. However, this Right does not let you revert ownership of the object to the previous owner.

#### **Debug programs**

This Rights lets one user affect the programs being run by another user. This is a sensitive capability because the “other user” could be an administrator, which allows a clever debugger to assume the administrator’s capabilities.

#### **Bypass traverse checking**

Suspends the ACL check the system makes when a program uses a directory within a pathname. Without this Right, if you do not have permission to enter a directory you can never gain access to any object in its tree, even if you are allowed access to that object. By analogy, if the door to a building is locked, you can’t get into its offices even if they are unlocked. This all changes when you give a user Bypass Traverse Checking. While you still cannot read or write the directory, you can pass through it to an object in its tree to which you have access (providing you know its pathname). By default, Everyone has this Right.

### **Shut down the system**

These guidelines do not treat shutting the system down (a “denial of service”) a security issue, although it may be an important operational issue. See also “Shutting Down the System” in *General Policies*.

### **Related Guidelines:**

*Domains & Basic Access Restrictions*, Rights to log on locally and remotely

*System Services*, Rights that apply to the SYSTEM account.

*User Accounts & Groups*, Right to log on locally versus “Logon To” parameters in an account.

### **References:**

[Sutt96] “User Rights” in Chapter 7, *Managing Groups and Accounts*, p. 182.

[NetGd] “User Rights” in Chapter 2, *Network Security and Domain Planning*.

UNCLASSIFIED

## 13. Auditing Policy & the Security Log

---

Windows NT can record a prestigious number of security events in its security log, each with considerable detail. However, many audit records lack essential information, and many contain no information of use to manual analysis. Windows NT's audit analysis is limited to a convenient but simple viewer. Windows NT fully accommodates 3<sup>rd</sup> party audit analysis tools, and in view of the large amount of detailed information the system can generate, Windows NT auditing is limited without such tools. See the references listed at the end of this chapter for a general overview of the Windows NT security log (auditing) features.

### Guidelines

#### Levels 1 & 2:

It is not the intent of these guidelines to limit the amount or kind of data that site administrators choose to record. However, because the guidelines' philosophy is to include only practical security procedures, they include minimal auditing. The difficulty in prescribing any degree of auditing is that auditing ultimately depends on how much time and energy administrators are willing to put into its maintenance.

☞ Therefore, consider these auditing guidelines as a "recommended point of departure."

In the table that follows "Major Servers" are Windows NT computers that perform significant workgroup sharing functions, whether or not they are a domain controller. "Workstations" are all computers that are neither major servers or domain controllers. (The term "Domain Controllers" includes backup domain controllers, to which the Audit Policy is replicated automatically.)

- ❑ Configure the Audit Policies using the User Manager according to the Audit Event Table, below.
- ❑ Using the Event Viewer, set the security log settings as follows:
  - ◆ Set the maximum log size to the amount of storage you wish to dedicate to saving the audit trail. This has a direct correlation to the frequency with which you save the log to backup storage. There is no specific value, although we recommend you dedicate at least 10 megabytes of storage with which to experiment.
  - ◆ Set log wrapping to "Overwrite Events as Needed." (See the rationale that follows.)
- ❑ Regularly monitor the size of the audit trail, save it to long-term storage (like magnetic tape), and then clear it.
- ❑ **Optional:** Create a logical partition on the hard drive reserved for the security log. Place the security log file on this drive (see "Alternative Locations for the Security Log," below), and set the Maximum Log Size to the space reserved for the security log on this logical drive. This assures that disk space does not prevent the log from reaching its maximum specified size. (You can also place fixed-size information on this drive.)

#### Periodic Review:

- ❑ Assure that saved security logs are securely stored and accessible when needed.

Audit Event Table

	Workstations	Major Servers & Domain Controllers
Logon and Logoff	Levels 1 & 2	1 & 2
Startup, Shutdown & System	1 & 2	1 & 2
Security Policy Changes [1]	1 & 2	1 & 2
User & Group Management [1]	1 & 2	1 & 2
Use of User Rights [2]	—	—
File & Object Access [2] [3]	—	—
Process Tracking [2]	—	—

- [1] These are often rare events outside domain controllers, which makes them more readily saved. They are optional at Level 1.
- [2] These categories produce many events of little use without analysis tools.
- [3] The guidelines do not generally prescribe any object auditing. However, it may be appropriate for both levels on a case-by-case basis.

## Notes

These notes include a few of the lesser-known aspects about the Windows NT security log. The guidelines do not prescribe the use of these features, although some site policies may require them.

### *Object Auditing Always Records SAM Objects*

If you enable the “File and Object Access” category in the auditing policy, most low-level accesses to the Registry’s user account database (the “SAM”) are audited as object access events, even though audit control information is not explicitly assigned to these Registry keys. Like most object access, the security log only records the access permissions requested by the program when it opens the keys, and these permissions are often far broader than the program actually uses. Hence, without more sophisticated audit analysis tools, the information you can glean is limited. These guidelines do not prescribe auditing these objects. See [KBase] Q149401.

### *Rights not Audited*

The auditing policy “Use of User Rights” always ignores the use of the following Rights: bypass traverse checking, generate security audits, create a token object, debug programs, and replace a process level token.

Auditing also does not record the use of the two Rights to backup and restore files and directories unless the Registry key:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Lsa
```

has a REG\_DWORD value named “FullPrivilegeAuditing” whose value is 1. Recording these will probably flood the security log with largely extraneous events. These guidelines do not prescribe auditing of the use of these rights.

***Auditing “Base Objects”***

“Base objects” are internal Windows NT objects not in the file system or Registry. Users do not usually see or directly manipulate base objects although their programs may access them. Windows NT does not audit access to these objects unless the Registry key:

HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa

has a REG\_DWORD value named “AuditBaseObjects” whose value is 1. (This value is 0 by default.) The audit category “File and Object Access” controls overall issuance of these events. Object Access events for base objects are similar to file system access events, but with different object and access permission names. Auditing these events tends to flood the security log with events of little security relevance to most administrators. These guidelines do not prescribe auditing for base objects. See also “ProtectionMode” in *General Policies*.

***Crashing when the Security Log Fills***

If the Registry key:

HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa

has a REG\_DWORD value named “CrashOnAuditFail” whose value is 1, the operating system shuts down when the security log reaches its maximum specified size and the security log setting “Do Not Overwrite Events” is set. (This resets CrashOnAuditFail to 0 so an administrator can restart the system and fix the space problem).

Some sites require that the system shutdown in the event it does not have the resources to store audit events. This happens when the log reaches its maximum specified size, or the disk volume that holds the security log fills. The guidelines do not prescribe this drastic action even at Level 2. Rather, administrators should monitor the security log’s size and prevent it from filling.

***Alternative Locations for the Security Log***

The Registry key:

HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\  
EventLog\Security

has a value named “File” that holds the path name of the security event log file. You may change this to an alternative location that takes effect when the system is restarted. You can use this to place the security log on a disk volume dedicated to security log information (perhaps along with other information whose size does not change) so that you can guarantee a set amount of storage for the security log. Note that the security log files (which end in “.EVT”) should not be publicly accessible. However, the auditing system opens this file exclusively during system startup which precludes public access, even if the file’s ACL allows such access.

***Right to Manage the Audit Trail***

The Right to “Manage Security Log” allows an account to view and clear the security log, save the log to a file, and set auditing information on objects (like files and Registry keys). It does not allow one to change the security log settings (like the overwrite policy) or manage the Audit Policy through User Manager. This is a sensitive Right tightly protected by default.

### ***Object Auditing***

Per-object auditing (that is, placing audit information, SACL, on an object like a file or directory) is an open-ended issue that can only be resolved by site practice and policy. Object auditing is no substitute for ACL controls, and these events can easily overwhelm the security log. They are best used with audit log analyzers, none of which are included with Windows NT. Although these guidelines do not prescribe object auditing, they do not discourage its use.

#### **References:**

- [Sutt96] Chapter 8, *Security Auditing*, pp. 193-199.
- [RKitW] “Auditing Security Events” in Chapter 6, *Windows NT Security*. A few brief notes about the security log.



## 14. System Services

---

Windows NT services are programs the system starts on boot-up. Services usually continue to run in the background, servicing various requests from programs that the user runs as well as requests from the network. Full administrators manage services using the Services control panel.<sup>23</sup> Each service runs from an account specified by the Services control panel, and services commonly use the all-powerful, built-in SYSTEM account. Services often assume the identity of programs that request their services using a process called “impersonation.” Services can thus gain capabilities of the requestor (often called the “client”) which can include capabilities beyond the service’s own account.

Software you install may include services, although this requires the installer to have full administrative capabilities. The standard criteria for installing only trusted software suffices for services with the minor caveat that most services are all-powerful. However, even at Level 1 you may want to institute some of the Level 2 practices.

### Guidelines

#### *Minimizing Services & Their Capabilities*

##### Levels 1 & 2:

Although there are few explicit procedures to recommend, we *recommend* administrators should institute the following practices to the extent practical. A Level 2 site should maintain an aggressive program to implement these techniques.

- ❑ Strictly limit the services that run on a given computer. There are a large number preinstalled on Windows NT. Consult the system documentation for their function. When in doubt, disable a service and see if any operationally required functions fail. Be particularly wary of new software that includes services, although only full administrators can install services. Use the Services control panel or command line tools to list services before and after the install.
- ❑ Many services install into the all-powerful System account and can therefore completely subvert security. However, many services don’t need the following security-sensitive Rights, any one of which can completely subvert system security:

- Backup files and directories
- Restore files and directories
- Act as part of the operating system
- Create a token object
- Debug programs
- Local and unload device drivers
- Replace process level token
- Take ownership of files or other objects

Query the service’s documentation or vendor to establish which of these can be withheld from the service. If you have any doubt about the trustworthiness of the service, run it from a limited account like the “unprivileged service account” described in the following Notes.

---

<sup>23</sup> The NETSVC tool from the Windows NT Server Resource Kit lets you query and control Services remotely, and can be useful in determining which services are active across the network.

- ❑ Seek to separate the ways that services can interact with one another. Eliminate their interactions when it's not necessary for their function. The best way is to run each service under a unique account and to assure that any directories or Registry keys each service uses are not accessible to other services.
- ❑ On large networks where administrators cannot control the installation of services, use a port scanner to help detect unknown services. (See "Service Attacks" in *Networking*.) However, the inability to control the introduction of services is the root of this problem and the one that must ultimately be solved.

#### **Periodic Review:**

Reviewing the active services on each computer on the network (whether or not you use a port scanner to help find them) is an important periodic review activity.

### ***Restricting Operator Control of Services***

#### **Levels 1 & 2:**

The following two procedures block means by which Server Operators can easily expand their capabilities or install programs that run with full administrative capabilities, which is contrary even to standard Windows NT philosophy:

- ❑ The guidelines *recommend* that only full administrators install services by assuring that the ACL of the following Registry key and its tree can be modified only by members of the Administrators and SYSTEM groups:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services
```

It is a general intent of these guidelines that only full administrators should be able to install applications that run with full administrative capability (in this case services). Server Operators can modify these keys by default, and this is contrary to the intent that operators not be allowed to expand their capabilities. Although Server Operators have devious ways of usurping power,<sup>24</sup> we *recommend* that direct means like this should be removed even at Level 1.

- ❑ By default, the Scheduler service that performs jobs submitted through the AT command runs under the SYSTEM account, although you can set it otherwise. (This service is not started by default.) A Registry parameter allows Server Operators to manage Scheduler jobs, which gives them the ability to expand their capabilities to full administrators. In the Registry key:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Lsa
```

the value named "SubmitControl" with a REG\_DWORD value of 1 allows Server Operators this capability. If the Scheduler service runs under an account whose capabilities exceed Server Operators (especially System and full administrative accounts), we *recommend* you remove this value. (By default, it is not present.) See [KBBase] Q124859.

This constraint means that only full administrators can install Scheduler jobs, which may be too restrictive from an operational point of view. *Alternately*, you can run the

<sup>24</sup> While these guidelines attempt to prevent operators from expanding their capabilities, Server Operators have widespread access to objects and can probably perpetrate many spoofing schemes. Server Operators must be trusted not to abuse their status.

Scheduler service under an account to which Server Operators can safely be given access, for example, an pseudo-account that gains its capabilities by being a member of the Server Operators group. Server Operators can then be allowed to install Scheduler jobs.

## Notes

### *Unprivileged Service Account*

You may wish to establish an “Unprivileged Service” account. Create a local account named, for example, “Unprivileged Service” with a 14-character, random password. Deselect “User Must Change Password at Next Logon” and select “Password Never Expires.” You need set no other account parameters. Assign all but the above Rights to this account. Set selected services to run from Unprivileged Service rather than System, which gives them considerably fewer capabilities. You may wish to remove other Rights from this account, but the above list is the most sensitive. You may need to give this account membership in certain groups so that services that use the account can gain access to certain objects, typically one of the Windows NT operator accounts or Power Users. Note however that adding this account into some groups, like the local Administrators group, can restore the privileges you removed in the first place.

### *Minimizing Services & Stolen Service Password*

Service accounts (other than the built-in SYSTEM account) should be local accounts with long, random passwords that are not written down. (They never need to be remembered.) You should minimize the capabilities of the accounts by assigning these accounts minimum Rights necessary to perform the intended function of the Service, and either excluding them from the ACL's of (or minimizing their ACL permissions to) sensitive information.

A service account should not have the same password on different computers unless required by the service. Many networking services run on one computer but need to make requests to others, which can only succeed when the service's account name and password are the same on all the computers. One technique is to use a local account with the same name and password on all computers. Under this technique, you should only install this account on computers that are to be accessed by the service. A second technique is to run the service under a domain account. Under this technique, you should remove this account's Rights to log on locally and remotely from all computers that are not to be accessed by the service.

However, both of these techniques have a common security problem. When a penetrator illicitly gains Administrative capabilities on one computer, they can retrieve the clear-text version of the passwords for all locally installed Services (see the LSA-2 hotfix, below). They can use these passwords to gain access to other computers on the network under the capabilities of that Service account. However, if these accounts are local accounts and their password is unique to this computer, the penetrator cannot use these techniques to gain access to other computers.

Sites with high security requirements may wish to consider grouping their networking computers and running the service local to each group. The service account has the same password on each computer in a given group (whether a local or domain account) but the passwords would be different between any two groups. An administrative penetration of a computer could then extend no further than the other computers of that group. You would typically put your most sensitive and secure servers in one group, most vulnerable computers in another, with perhaps degrees in between.

Alternatively, some services that require the same password on more than one computer can be run under the internal System account, assuming they are thoroughly trustworthy. The System account is not prone to this password malady.

The LSA-2 hotfix to SP3 from Microsoft removes the penetrator's direct determination of Service account passwords. See, for example:

```
ftp://ftp.microsoft.com/bussys/winnt/winnt-public/fixes/usa  
/nt40/hotfixes-postSP3/lsa2-fix/Q184017.TXT
```

However, as a full administrator, the penetrator is limited only by their imagination and knowledge of Windows NT internals, and we must presume a sophisticated attack could still exploit this issue. Indeed, some have reported doing this, although their skill level is far from rudimentary.

**Related Guidelines:**

“Service Attacks” in *Networking*, for the importance of minimizing network services.

## 15. Network Sharing

---

This guideline covers the general policies for network share directories and printers. Note that Power Users, Server Operators, and Print Operators can create these shares.

### Guidelines

#### *Network Share Directories*

Network share directories are the primary Windows NT mechanism for sharing files between computers on a Windows NT network. Although there are no specific procedures for securing network share directories, carefully consider each new share under the following general guidelines. Note that you can share devices like floppy and CDROM drives. These may or may not pose an inordinate security risk.

#### **Levels 1 & 2:**

- ❑ Strictly minimize the number of shares and their ACL share permissions. The share ACL is stronger than file and directory ACLs because it cannot be relaxed by users of the share.
- ❑ Because a share name can be viewed by users without access to the share itself, and in some cases unauthenticated users (see “User & Share Names Available to Unauthenticated Users” in *General Policies*), the share name should not contain information that you wish to guard, for example, “Merger Plans with Ajax Corp” if a pending merger is a secret.
- ❑ Avoid sharing the system root directory (usually named WINNT), although this is not necessarily unsecure if its ACL is appropriately tight.
- ❑ Disable administrative shares if you don’t need them, especially at Level 2. See “Hidden, Administrative Shares” below.
- ❑ Windows NT Server’s NETLOGON directory can be instrumental in domain security policy and its share ACL and subdirectory ACLs should be set tightly.

#### **Periodic Review:**

It is particularly important to frequently review the existence and ACLs of network shares, since the number of shares and their ACLs tend to expand over time.

#### *Printer Access*

While there are many operational reasons for limiting the use of printers through their ACLs, there are relatively few security reasons. Physical security is the major concern for printers because it determines who can view or intercept other users’ printouts.

#### **Levels 1 & 2:**

- ❑ Limit access to printers whose output empowers a user. For example, a printer loaded with pre-signed, blank checks, or a printer used to convey operational orders to certain personnel.
- ❑ We *recommend* you limit user accounts that customarily handle extremely sensitive data to printers with the appropriate physical protection. This helps to prevent these users from

using printers in locations freely accessible to people who should not view such information. While you can alternately rely on the users to follow printer policy voluntarily, it is easy to send a document to the wrong printer.

## Notes

These notes present a few reminders about access control to share directories. Network printer sharing works the same, although we only refer to directory sharing explicitly.

### *Summary of Sharing Mechanisms*

- ❑ Any Windows NT computer can host network share directories. Each share name has an ACL (called its “permissions”). Share ACLs only restrict remote access – not access to a program on the same computer.
- ❑ Only full administrators, Server Operators, and Power Users can create or remove access to network share directories, and can arbitrarily set the share ACL. These users plus Print Operators can create printer shares.
- ❑ Remote access to a file or directory in a share directory must pass both the share ACL and the ACL on the file or directory.
- ❑ Allowing Everyone access to a share does not mean all users on the network can access it, only those that present a successful remote logon to that server. However, authorized client users can see all share names on a server computer, even those to which they have no access. There are cases where users unauthenticated to a server can see the share names, although not access them. See “User & Share Names Available to Unauthenticated Users” in *General Policies*.
- ❑ The “share name,” or share instance, is that to which a share ACL is attached. If one directory encloses another and both are shared, the only governing share ACL is the one the client referenced when it established the link to the share.

See “Accounts & Network Authentication” in *Domains & Basic Access Restrictions* that describes how users are authenticated for access to share directories.

### *Hidden, Administrative Shares*

The Registry key:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\
  LanManServer\Parameters
```

can hold a value named “AutoShareServer” on a domain controller (“AutoShareWks” on a workstation). If its value is 1, the system automatically creates “administrative shares” for its logical volumes, C:, D:, and so forth, which it names C\$, D\$, and so forth. Although browsing windows do not show these names to users, their existence is no secret and users can request to connect to them. On a domain controller, only members of the local Administrators, Server Operators, and Backup Operators groups (and on a workstation only Administrators and Backup Operators) can access these shares and their ACLs cannot be changed.

Although administrative shares do not constitute an unwarranted security risk even at Level 2, you can disable this automatic creation by setting this parameter to 0. This can also be set in the Policy Files as the policy “Create hidden drive shares.” (See *System Policy Files*).

Administrative shares were originally intended for system services to query to determine the amount of available disk space so the services could warn when the drives were in danger of becoming full. You can forego creating these shares if you don't need this service, although other services may also rely on them. See also [KBase] Q126309.

**Related Guidelines:**

*Networking*

*System Policy Files*, creating hidden drive shares

**References:**

[KBase] Q126309, Q158433, and Q100517.

[ConPIn] Chapter 4, *Managing Shared Resources and Resource Security*. A general treatment of establishing and managing network share directories.

**UNCLASSIFIED**



## 16. Networking

---

Networking is part and parcel of the Windows NT environment and many of the guidelines bear on its security. This guideline offers a few general considerations for using Windows NT in networking environments. Networking is a complex topic and it is difficult to give networking guidelines with the same certainty as other areas. The guidelines therefore allow administrators considerable latitude in this area, and the Notes review several relevant networking security topics.

We use the term “intranet” to mean any wide-area TCP/IP based network. An intranet is actually a managed assembly of TCP/IP-based local area networks (LAN’s), where each LAN connects to the intranet through a router (or Windows NT computer acting as a router). “Router” are special purpose computers whose job is to ferry packets between the intranet and LAN, often asserting certain controls and restrictions.

### Guidelines

None of these protections are theoretically necessary on a secured network where no malicious elements have direct, packet-level access to the network media. (See “Network Eavesdropping & Interception” in the Notes that follow for an expanded description.) For example, a small, properly administered, physically protected LAN with only Windows NT computers is reasonably safe. However, it is difficult to assure that larger networks are secure or remain so over time. One should therefore pursue the following guidelines for almost all networks of significant size or complexity, especially if your network is connected to a public network like the Internet.

#### Levels 1 & 2:

- ❑ Remove unnecessary networking services, where a “networking service” refers to any program that offers networking services whether or not it is managed under Server Manager. This is *recommended* for Level 1 and *prescribed* for Level 2.
- ❑ Formulate and enforce specific policies for the use of all networking client software, for example, whether or not users should enable Java or ActiveX in their Web browsers.
- ❑ Unless your Windows NT systems are using cryptographic techniques described below, remove computers from your network whose operating systems (see “Network Eavesdropping & Interception,” below) allow malicious programs direct, packet level access to the network unless you can assure that these computers run only trusted software. This is *recommended* for Level 1 and *strongly recommended* for Level 2. (This is unfortunately easier said than done on most networks. While you may be forced into concessions, recognize the degree to which these computers may compromise security.)
- ❑ Prevent Windows NT from passing cleartext (plaintext, unencrypted) passwords across the network as described below in “Unencrypted Passwords on the Network.”
- ❑ As described below in “LANMAN Passwords,” the LANMAN network authentication format has certain weaknesses compared to Windows NT’s native format. If you need not serve Windows systems other than Windows NT, disable the LANMAN format. If you must serve Windows 95 or older computers over a vulnerable network, set the value of “LMCompatibilityLevel” to 1 (which uses the LANMAN format only for those computers that request it), although this incurs a corresponding loss of security. At Level 2, we *recommend* you make no exceptions: disable the LANMAN format.

- ❑ Isolate the native Windows NT services (like file sharing and remote administration) from an unsecure intranet as described in “Isolating Native Windows NT Service from an Intranet” in the Notes that follow. This is *prescribed* for both Levels 1 and 2.
- ❑ Position a firewall between the LAN and an unsecure intranet. This is *recommended* for Level 1 and *prescribed* for Level 2. (The criterion for firewall selection is beyond the scope of these guidelines, although a companion study for this project presents general guidelines for configuring the Microsoft Proxy Server.) Some intranet routers have security features and we include these in the general term “firewall,” although router security is typically not as strong as firewall security.
- ❑ Several Windows NT services and applications offer cryptographic protection against networking threats. Utilize these protections fully.
- ❑ Require SMB signing on all Windows NT computers as described below in SMB Signing. At Level 1, servers that must serve computers that cannot perform SMB signing can enable but not require SMB signing. Otherwise, the only acceptable reason for not using SMB signing is if its performance degradation is unacceptable to the site’s mission.
- ❑ If your local network is connected to an unsecure intranet via IP router (and they usually are), you can effectively prevent local workstations from communicating with the external network by configuring them to run only protocols other than TCP/IP, like NETBEUI. However, routers and local servers that can proxy or translate protocols could subvert this protection by offering a pathway- to the external network.
- ❑ If you are using a Windows NT computer as the basis for a firewall to an intranet, be cognizant of the TCP/IP protocol feature called “IP forwarding” which directly transfers packets between the internal and external network. Enabling IP forwarding may subvert firewall or routing controls imposed elsewhere.
- ❑ Finally, and most important, pursue third party software that cryptographically protects all traffic (user as well as system data) to and from your Windows NT systems. This is *recommended* for Level 2, and may largely negate the need for several of the other protections in this list. See “Applying Cryptography to All Network Traffic” in the following Notes.

**Periodic Review:**

Networking threats are among the most worrisome and justify frequent auditing at Level 2.

- ❑ Implement a complete set of auditing guidelines for the features of any firewall or security-enhanced router.
- ❑ Use a port scanner to monitor for unauthorized services.

## Notes

We do not discuss Trojan Horse attacks here although they often arrive over the network. See instead *Spoofing*.

### *Unencrypted Passwords on the Network*

Windows NT has the ability to communicate with certain non-Windows NT systems that require sending user passwords unencrypted (“plaintext”) over the network. This feature is disabled by default and must be manually enabled by adding the value named “EnablePlainTextPassword” with a REG\_DWORD value of 1 to the Registry key:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RDR
  \Parameters
```

Prevent Windows NT from passing unencrypted passwords across the network by removing the EnablePlainTextPassword value from this Registry key. This feature was implemented in Windows NT 4.0 SP3. See also [KBase] Q166730.

### *SMB Signing*

Windows NT supports a cryptographic integrity mechanism with replay protection called “SMB Signing” for its native network sharing protocol, SMB. Windows NT uses SMB for access network share directories and printers, and several other native Windows NT services, like remote administration. SMB signing prevents active network taps from interjecting themselves into already established sharing sessions, usually called “session hijacking.” Without SMB signing, such penetrations can modify and view all the information on the server to which the client user has access. SMB signing prevents such penetrations. However SMB signing provides no encryption of user data, and therefore any network tap can view all the data the user transmits between client and server.

SMB signing currently provides “40-bit” protection (even though some of its interactions use stronger methods). However, it also uses the user’s password as the basis of the encryption key, and therefore the effective protection is the lesser of 40-bits and the key space of the password. To be 40-bit effective, users must conscientiously use a password scheme that provides at least a key space of 40 bits, or approximately  $10^{12}$ . (See “Logon Attempt Attacks” in *Passwords* for password space calculations for common password schemes. For example, this is equivalent to selecting a 7-character password with randomly selected numeric and upper and lower case alphabetic characters.) Future versions of SMB signing may support 128-bit encryption, although the effective protection is still limited by the password space.

Note that SMB signing allows a passive tap to brute-force attack the user’s password, although no more so than the standard, unsigned SMB protocol. Whether or not SMB signing is used, user password space must be chosen to prevent such attacks whenever the network has taps.

While an encouraging development in Windows NT network security, its lack of encryption and its reliance on properly chosen passwords leaves it a useful but partial solution to network security. Its net effect is to limit but no eliminate the damage active network taps.

Windows 95, earlier versions of Windows, and Windows NT without this feature installed cannot engage in SMB signing. If you configure your servers to require SMB signing, they cannot serve these systems. SMB signing reduces network response time (although it does not significantly reduce network bandwidth), but there are no published estimates.

Require SMB signing of server and/or client activities by creating REG\_DWORD values named “EnableSecuritySignature” and “RequireSecuritySignature” with a value of 1 in the Registry keys:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services
  \LanManServer\Parameters
```

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services
  \Rdr\Parameters
```

This feature was implemented in 4.0 SP3. See [KBase] Q161372, “How to Enable SMB Signing in Windows NT” for the details of implementing this feature. If you are interested in the details of this protocol or a detailed discussion of its security strengths, you can read the Microsoft articles on the “CIFS Authentication Protocols” available from the Microsoft Web site.

### ***LANMAN Passwords***

A tool that seeks to detect user passwords (L0phtcrack) has brought to light that the LANMAN format for passing authentication password information across a network is “considerably” more prone to attack from malicious, passive network elements than Windows NT’s native technique, although there are no precise figures for how much less secure. Roughly, this format negates the advantages of using distinct upper and lower case alphabetic characters, and effectively limits your password to 7 characters. (Fourteen character passwords are only twice as hard to guess as 7-character ones.)

To prevent Windows NT from using the LANMAN format, create and set the REG\_DWORD value named “LMCompatibilityLevel” in the Registry key:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\control\LSA
```

to 2, which prevents Windows NT from using the LANMAN format. Do this on all Windows NT computers. Note that eliminating the LANMAN password format means that Windows NT’s native services (like file and print sharing) cannot service Windows 95 and older Windows systems. Setting this value to 1 includes the LANMAN format only for computers who request it (like Windows 95).

This feature was implemented in 4.0 SP3. See [KBase] Q147706

### ***Service Attacks***

There are two principal networking threats. The first is where malicious clients attempt to gain services to which they should not be entitled. These attacks request services from a target computer in such a manner as to thwart normal security protections or exploit little-known, unprotected aspects of the service. For this reason, the services that a given computer runs encompass virtually its entire exposure to externally initiated attack. (“Externally initiated attack” includes all but Trojan Horse attacks.)

The first commonsense protection is to use the Services control panel to remove all services that are not necessary. Level 2 sites should take the reverse view of disabling all services then re-enabling only those that are operationally necessary. (See *System Services*.)

Even after a tight, initial Windows NT network installation that minimizes services, applications added to the system might implement their own network services, ones whose security protections are often weak at best. You can use a “port scanner” to find such

services.<sup>25</sup> Port scanners are software programs that attempt to connect to TCP and other protocol ports that service use to field client requests. While the scanner can't usually identify the service, it can at least report that some service was fielding requests on a particular port, which helps find unauthorized services on the LAN. Also, most common services use universally agreed-upon ports, which can help to identify the service involved.

The best overall protection for a LAN connected to an unsecured intranet is to interpose a firewall between the LAN and the intranet. Firewalls offer a host of services although these topics are beyond our current scope. A later study for this project will offer security guidelines for the Microsoft Proxy Server which can be considered a firewall.

### ***Network Eavesdropping & Interception***

The second principal networking threat is where malicious elements on a computer network can eavesdrop or intercept data between two legitimate communicants. Most networking communication media are prone to eavesdropping and interception, also called active and passive "taps." Computers on the network are capable of maliciously reading packets of information passed between two other computers, or even intercepting packets, changing them, then forwarding the false packets on to the intended receiver.

"Session hijacking" is a particular kind of active attack that is feasible on Windows NT. When a Windows NT client connects to a server's network share directory, the server authenticates the user without passing any passwords across the network. (The client and server use a classic "challenge/response" technique.<sup>26</sup>) Once the server authenticates the client user, the server passes an unprotected session identifier back to the client. The client subsequently presents the identifier for all subsequent requests to connect to a shared directory on the same server without further authentication.

Session hijacking is where an active attacker reads the identifier and subsequently presents requests to the server. The server will treat these as legitimate requests and fulfills them. This attack succeeds not because Windows NT has any particular security flaws, but only because networking traffic is not protected by cryptographic techniques. One can find variations of this attack on most networking operating systems that do not use such techniques. We mention this attack only to underscore the vulnerability of a network not cryptographically protected.

To perform such attacks, a malicious program needs direct access to the computer's networking hardware. Windows NT and systems like UNIX (all major varieties), Digital's VMS, and AS/400 are designed to deny unprivileged programs such access. Hence, when properly administered (and barring any security holes) these systems on your network pose no such threats. However, all programs on system like DOS, Windows (versions other than Windows NT), and the Macintosh have such access. Hence, to the extent they might run malicious programs, these latter systems pose the threat.

Cryptographic techniques provide the only strong protection against these and other network attacks. Such techniques can assure that eavesdropped packets cannot be deciphered and that packets maliciously modified can be reliably and immediately detected. Windows NT provides no such cryptographic protection itself. However, you may be able to acquire 3<sup>rd</sup>

<sup>25</sup> Port scanners typically work on IP protocol. The NETSVC tool from the Windows NT Server Resource Kit lets you query and control Services remotely, and can also be useful in determining which services are active across the network.

<sup>26</sup> Note, however, that a malicious passive network element can observe the challenge/response to mount a brute force attack on the password.

party products that do so for all NT-to-NT communications, or for specific applications like traffic between Web browsers and Web servers.

### ***Apply Cryptography to All Network Traffic***

Cryptographic techniques provide the only strong protection against active or passive network attacks (eavesdropping and interception, or “taps”). Some Windows NT networking services and applications provide such protection, but not all. The guidelines recommend any Level 2 Windows NT networking system prone to the following threats seek 3rd party network cryptographic protection<sup>27</sup> for all traffic (that is, user data traffic as well as Windows NT system-level traffic) in the following situations:

- ❑ Wide-area Windows NT networks that utilize untrusted intranets for communication between Windows NT enclaves. Note that non-encrypting firewalls do not themselves protect against these tap threats.
- ❑ Where the Windows NT network has computers whose operating systems cannot prevent untrusted programs direct, packet-level access to the network, including DOS, Windows (all versions except Windows NT), and Macintosh. When properly configured and administered, UNIX (all major varieties), Digital’s VMS, AS/400, and most “kernel-based” operating system claim to prevent such direct access. (However, “freeware” operating systems such as Linux and FreeBSD may be more exploitable due to the ready availability of kernel source code.)
- ❑ When any element of the communication transport mechanism is prone to eavesdropping or interception, such as utilizing a public carrier.

Cryptographic protection should include encryption (assuring that taps can’t view information) and integrity (assuring that all malicious modification of data in transit is detected). Also, consider protection against replay attacks, although replay threats are usually not as common as the others are.

### ***Isolating Native Windows NT Service from an Intranet***

There are many dangers when your Windows NT LAN is a part of an intranet, and attacks on Windows NT native sharing services are one such risk. The Windows NT native sharing services include file and print sharing, and mechanisms called “named pipes” and “RPC” that serve as the basis for many client-server services, for example most of Windows NT’s remote administration. There are a few techniques for making sure that sources on the intranet cannot interfere with or eavesdrop on these native services.

The first technique is to use only the NETBEUI protocol for these native services. Packets of the NETBEUI protocol cannot be routed onto or received from the intranet by most routers. Assuming you’ve already installed the NETBEUI protocol, use the Network control panel to assure that the following services are bound only to the NETBEUI protocol: NetBIOS Interface, Server, and Workstation. Do this on all the Windows NT computers on your LAN. (See “Isolating NT Services from an Intranet” in the chapter *The Internet and Intranets* in [Sutt96], pp. 255-257.)

A second common technique chokes these services at the router. These native services can also use the TCP/IP protocol which is in danger of being routed onto and in from the intranet. However, they use only the UDP and TCP ports 137 through 139, and by convention, communicants on TCP/IP networks use these ports for no other purpose. Many routers can

<sup>27</sup> See, for example, the “SnareNet” product from Snare Networks Corp. (<http://www.snare.com>).

identify and block designated TCP and UDP port numbers, and can therefore assure that elements on the intranet cannot interact with native Windows NT services on the LAN. Simply block both UDP and TCP ports numbered 137-139. The details obviously depend on the particular router. The advantage of this second technique is that you need manage only the router instead of all computers on the LAN. It also lets you use TCP/IP on the LAN for these services.

You can also “unbind” Windows NT services services from certain LAN cards (in the Network control panel), although of course they no longer serve any clients on those LAN’s.

Of course, this means computers on the LAN can no longer request or serve these native Windows NT services to computers on the intranet. If you need this capability, your best course is to seek 3<sup>rd</sup> party cryptographic packages that support it.

### ***IP Spoofing***

Many Windows NT security packages can filter packets to or from an intranet based on the source/destination network IP address. However, elements on an untrusted network can often insert any IP address they like into packets they send, and intercept packets regardless of the sending or receiving IP address. This is called “IP spoofing” and lessens the effects of IP filtering as an effective security protection. There are standardization efforts underway to counter this threat using cryptographic techniques. Also, 3<sup>rd</sup> party network encryption packages effectively counter IP spoofing. Level 2 Windows NT sites should not rely on simple IP filtering for security in hostile intranet environments.

### ***TCP/IP Port Limitations***

The advanced security options in Windows NT’s TCP/IP protocol (accessible through the Network control panel) can restrict TCP and UDP traffic to a specific list of ports. However, while services usually use fixed port addresses, clients can use a wide variety of system-selected ports. We recommend you explore using this option to tightly limit the ports available to unauthorized servers programs. This is an advanced, operational TCP/IP topic beyond our current scope, and these guidelines offer no specific guidance.

### ***The Security of Windows NT’s Protocols***

There is no significant, inherent security in any of the native Windows NT networking protocols: NETBEUI, TCP/IP, and IPX/SPX, and there is no appreciable security advantage in using one over the other, except as noted above.

### **Related Guidelines:**

*Guidelines for Security Microsoft Proxy Server*, an accompanying document to these Guidelines.

*System Services*, for minimizing system networking services.

Remote Access Service (RAS)

### **References:**

- [Kauf95] An excellent, albeit technical, presentation of networking security issues.
- [She197] Chapter 15, *Securing Private WANs and Virtual WANs*.
- [She197] Chapter 18, *Firewalls and Proxy Servers*.
- [Sutt96] Intranet topics in Chapter 3, *Your Working Environment*, p. 53-55.

[Sutt96] Chapter 9, *The Internet and Intranets*, p. 225, except the topic “Internet Information Server.”



## 17. Remote Access Service (RAS)

---

Readers new to the Windows NT RAS service might first read the references listed at the end of this guideline.

The Windows NT Remote Access Service (RAS) allows remote computers to connect to Windows NT RAS servers across a phone connection, or, using the PPTP protocol, over an intranet. Once connected, the client computer functions as if directly attached to the RAS server's LAN. For example, the client can connect to share directories and use printers on the LAN, subject to the normal Windows NT security controls. RAS can be limited to selected users, and has a "call-back" feature where the server calls back a predetermined or user-specified telephone number to complete the dial-in connection scenario. RAS security is tightly integrated into the Windows NT domain security architecture. RAS client software is available for other operating systems, like Windows 95.

**Note:** Remote access to your networks via telephone connection or across an intranet opens many opportunities for attack, due partly to the lack of physical protection of the remote computer. (See the Notes below.) Although the guidelines that follow counter many such attacks, you should carefully weigh the risks of remote access against any gain. Many sites, especially those that fit our Level 2 designation, categorically prohibit remote access.

### Guidelines

#### Level 1:

- ❑ Require "Microsoft encrypted authentication" on the RAS server and all clients. This assures that unencrypted passwords never pass over the communication media.
- ❑ If you are using RAS over telephone lines that may be tapped, or using PPTP over an unsecured intranet, require "data encryption" on both client and server. If your Windows NT software allows, choose 128-bit encryption rather than the default 40-bit. (See the notes below.)
- ❑ Grant remote access capabilities only to those users who require it. (Use the "Dialin" panel from the User Manager's main account window, or the RAS user administration window. They are equivalent.)
- ❑ Wherever feasible, use callback to a preset number (that is, a number that cannot be specified by the caller).
- ❑ Assure user passwords are of sufficient complexity. See "Strong User Passwords" in the notes that follow.
- ❑ Do not use RAS clients on operating systems other than Windows NT unless you deem that operating system secure enough to directly connect to your LAN. However, there is no way for the RAS server to enforce that a client be Windows NT. In any case, we recommend you only use Windows NT clients.
- ❑ Consider the "RAS Sentry" configuration described in the Notes.
- ❑ We *recommend* you investigate adding a secondary authentication mechanism to protect dial-in access. (See page 373 in [Shel97] for a list of such devices.)

**Level 2:**

The same as Level 1 with the following enhancements:

- ❑ Assure that the remote computer is physically protected. (See the Notes below.)
- ❑ Do not use remote RAS clients other than Windows NT.
- ❑ Do not rely on 40-bit data encryption.
- ❑ Use callback to a preset number unless there is strong evidence that risk of attack is acceptably low.

**Periodic Review:**

Because external threats are especially dangerous, the guidelines prescribe frequent review.

- ❑ Check for users that are allowed RAS access that are not actively using it.
- ❑ Assure that dial-back is used wherever possible.
- ❑ Review the physical security of the remote computers.

**Notes*****General Discussion***

Remote access is often from sites whose physical security is not as strong as if the computer were directly connected to the LAN, and this presents a significant risk that you must consider. There are many ways that physical access to a remote computer can completely compromise the system. While there are no definitive guidelines, you must attach considerable importance to remote security, especially at Level 2. (You might, for example, boot from a removable hard drive that can be locked away when not in use.)

United States export restrictions limit Windows NT to 40-bit encryption. However, U.S. users may be able to obtain 128-bit encryption directly from Microsoft or 3<sup>rd</sup> party suppliers. The 40-bit encryption provides only modest protection against a determined attack. The 128-bit encryption is impossible to break using brute force techniques—only by weaknesses in its algorithm. Current versions of Microsoft RAS use the RC4 algorithm from RSA, Inc., which has no publicized weaknesses.

You can limit remote access to the RAS server only, as opposed to the network to which the server is attached. (For example, under this limitation a remote user could access files on the RAS server but not network share directories.) This is a sensible restriction for users that don't need access to the network. It is also an effective layer of protection for the network. Note that individual accounts can effectively be restricted to the RAS server by making them local accounts on the server. On a Windows NT network, local accounts can only access the computer that holds the account, the RAS server in this case. The Registry value NetbiosGatewayEnabled in the key:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services
  \RemoteAccess\Parameters\NetbiosGateway
```

controls this access to the network in general. However, do not set this parameter except through the RAS setup program. These ideas are the basis for the RAS Sentry described below.

There are a number of RAS parameters that can be set by the client user, such as requiring encrypted communication. Legitimate RAS servers independently enforce the parameters

important to security and don't rely on the client settings. However, it is a prescribed practice to configure the client to "Accept only Microsoft encrypted authentication" and the same degree of data encryption as the host. Data encryption provides the only useful protection against a masquerading RAS server.

The client user has the option of letting the client system "save" passwords presented to the server for later, automatic presentation during RAS connection. These guidelines take the position that, on balance, there is no security advantage for saving versus not saving. While saving passwords does pose a burden on Windows NT to protect the password, entering the password during a connection scenario (required when they are not saved) is not always on the Trusted Path and as such is an exposure. The safest client practice is to use the option on the client logon window to "logon using dial-up networking" and then always electing to not save the password when asked. In this scenario, you always enter the password via the Trusted Path – the normal logon window. However, there's no way an administrator can enforce this.

There are a number of RAS control parameters in the Registry key:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services
  \RemoteAccess\Parameters
```

whose defaults are appropriate for both Levels 1 and 2. These include the parameter that instructs RAS to audit its transactions in the security log, which is enabled by default.

### ***Strong User Passwords***

Microsoft does not customarily publish the details of their cryptographic protocols and we were unable to locate an authoritative description for this study. However, there is strong anecdotal evidence that the user's password is the basis for the RAS encryption. If true, this not only exposes the user's password to communication path brute force attacks but effectively limits the RAS encryption to the strength of the user password space. It is therefore important to use random passwords whose space is at least as large as the cryptographic key length desired. See "SMB Signing" in *Networking* for a discussion of user password complexity when used as the basis for cryptographic keys.

Stated differently, a penetrator that seeks to determine the encryption key of a RAS connection by brute force would simply use a password brute force technique, generating each trial key from a trial password. (The RAS key is likely a hash of the user's actual password, but one must presume the penetrator knows how to construct a trial key from a given password.) And once they discover the encryption key, they also have the user's password!

### ***RAS Sentry***

A RAS sentry is a name we give to a RAS server designed to maximize insulation from the main network. To set up a RAS sentry:

- ❑ Install RAS server on a dedicated Windows NT domain controller according to the guidelines above. Restrict remote access to the RAS server itself – do not allow remote access to the server's network.
- ❑ Create a domain account on the RAS sentry for each user who needs remote access. (This is in addition to their day-to-day account they use for their on-site computer, but could use the same account name.) Create strong passwords based on "Captured Password Attacks" in *Passwords*. We recommend you assign such passwords to the user and not let them change these passwords. Assign the account RAS capabilities.

- ❑ Set up a home directory on the RAS server for each of these users and share it on the network so that the users can access their home directories from their normal, on-site workstations.
- ❑ Set up the RAS domain controller domain to trust other domains whose users must transfer data to or from the RAS sentry. Typically these are the domains that hold the on-site accounts for the users for whom you have set up accounts on the RAS sentry. Note that the remote RAS accounts are not recognized on any computer other than the RAS sentry, which minimizes the potential damage they can do on the network.
- ❑ Minimize the services and applications that run on the RAS sentry. It is also best to simplify the RAS sentry in general. For example, don't use it to provide any general data or print services to computers on the network.
- ❑ Enable as much auditing as you can usefully process and monitor.

Users do not have direct pass-through access from the remote site to any computer on the network. Typically, they use the RAS sentry by transferring data to and from the sentry from on-site workstations, then working on those files remotely.

**References:**

- [Sutt96] *Using the Remote Access Service*, p. 118, and *RAS and PPTP*, p. 252.
- [NetSp] Chapter 5-7 on Remote Access Service (RAS). Overview and general procedures for managing RAS.
- [TchNet] Article entitled "MS Windows NT Server 3.5 Remote Access Service." A basic overview of RAS's capabilities and technology.

## 18. Spoofing

---

Each program on Windows NT gains the full capabilities of the user that runs it. If a user without certain capabilities can create a malicious program and then arrange to have a second user with more capabilities run the program, the first user can effectively expand their own capabilities to that of the second user. This is commonly called “spoofing” and is particularly dangerous when the second user is an administrator. Such a malicious program is often introduced onto the system as a benign program. This is a “Trojan Horse.” These programs are also called “viruses” although the term has become quite generic.

Users with special security capabilities (which we normally call “administrators”) must be particularly vigilant and run only trusted programs whose ACL prevents them from being tampered or replaced in whole by users of lesser capabilities. Unfortunately, this can be difficult to assure. Many features of a general-purpose operating system like Windows NT are designed to make executing programs easy, transparent, and flexible, all of which enhance spoofing opportunities.

Spoofing is perhaps the greatest security risk you’ll encounter. Confining it requires a collection of a difficult, diverse, and nonspecific tasks. We present several of these in this section but there are undoubtedly many more.

One of the best ways to minimize the spoofing risk is to minimize the amount of work done under accounts with many capabilities, particularly full administrators. This is one of the primary goals of the guideline “Domain Operators & Power Users” in *Administrative Structure*. Another good way is to perform system administration remotely from limited, protected, minimally configured workstations whose use is dedicated to administration (see “The ‘Administrator’ Account” in *Administrative Structure*).

Spoofing of regular users can result in exposure or undetected modification of files under their or their groups’ control. While not as dangerous as administrators, this poses a significant threat for users who handle sensitive information and the same protections apply.

Please note that the guidelines in this chapter are interspersed within the Guidelines & Notes. The guidelines listed correspond to the preceding paragraphs of notes.

### Guidelines & Notes

#### *Logon Separation*

The surest way to prevent one user from spoofing a second at all is to assure that both cannot log on to the same computer (through either primary or secondary logon), or that the first has only read-only permissions to Registry and file system areas in which the second works. There are no specific guidelines we can offer here other than to consider the range of trustworthiness of the users who can access each computer.

#### *Trusted Path (“Security Window,” “Secure Attention Sequence”)*

When you use the CTL+ALT+DEL key combination to call up the Windows NT Security Window, you are invoking what is traditionally called the “Trusted Path,” “Secure Attention Sequence,” or “Security Window.” The Trusted Path is a physical action that any user can take such that the ensuing interaction with the computer (a window in this case) is guaranteed to be trustworthy and not a spoofed, “fake” window under the control of a malicious program. Sensitive interactions, like entering a user password, are traditionally on the Trusted Path and

Windows NT continues this tradition. Users should conscientiously use only the Trusted Path to logon, log off, change their password, and lock their workstation.

The Start Menu is not a Trusted Path. However, it is almost as strong on Windows NT. Because the desktop in Windows NT is inherently single-user, one user has little chance of presenting a fake Start Menu to another, or otherwise modifying a user's Start Menu. A malicious program would likely have to be running under the current user's session already in order to present a false Start Menu. Such a program already has full access to the user's data and environment, and spoofing the Start Menu gives the program little it doesn't already have. A malicious program in this situation can assure that some malicious element becomes a constant part of that user's environment promulgating into subsequent logon sessions, but there are many ways to do this besides altering the Start Menu.

Anti-spoofing practices must always seek to prevent malicious programs that could present a false Start Menu or any other desktop element in the first place. Once such a program runs, the battle is usually lost.

### ***System-Wide "PATH" and Other Environment Variables***

The "PATH" environment variable defines a list of directories where the DOS window and other system elements search for commands the user types. The system-wide PATH variable must contain not only directories whose ACL prevents untrusted users from adding or modifying files, especially executable files and DLL's, but also any data files trusted program may rely upon. Users can modify their own PATH variable but should be cautioned to follow this advice.

If enabled by the Registry, the system searches the AUTOEXEC.BAT file adding the paths it specifies to a user's PATH variable upon logon. The value named "ParseAutoexec" with a REG\_DWORD value of 1 in the key:

```
HKEY_CURRENT_USER\ Software\Microsoft\Windows NT\
  CurrentVersion\Winlogon
```

enables this, and by default it is enabled. Although any user is allowed to enable this key in their own environment, the key is accessible only to that user and is safe from attack. However, AUTOEXEC.BAT must be protected if any user enables this key. Note that the System Policy Editor can set the value of this key at user logon, although it does not prevent users from changing its value during their session.

Other user environmental variables can provide spoofing opportunities, but general users cannot modify other user's variables because they are stored in the user profile. However, administrators should exercise diligence when defining global system environment variables. (Use the Environment panel of the System control panel to view and define system and user environment variables.)

#### **Guidelines:**

##### **Levels 1 & 2:**

- Assure that the global PATH environment variable contains only directories maintained as application directories as per the guidelines "Application Directories" in *Application & User Home Directories*.
- Set the ACL on AUTOEXEC.BAT as in the guideline *File System & Registry ACL Settings*.

- ❑ Do not modify or add system environment variables unless you determine that their values do not pose a spoofing risk.
- ❑ Strictly minimize the search paths of all administrators and operators, confining them to protected directories that hold only trustworthy programs.

### ***The “.” Issue***

The DOS window implicitly searches for a command in the current working directory (referred to as “.”) before the directories in PATH, which allows a potent spoofing threat that’s difficult to confine. The Windows NT API’s that let programs start other programs (such as CreateProcess) search the current directory before PATH looking for the named program. (In both cases, the system can run programs other than those with the .EXE suffix, but only when their full names are specified.) These same general comments for .EXE files apply to batch files with the .BAT and .COM suffixes. There are many places in Windows NT and its applications that let users run other programs, and it is hard to tell where they look for programs. It is difficult to protect against this spoof. A few suggestions:

1. Consider third party command line programs that allow one to omit searching the working directory (“.”) for commands. (Because UNIX traditionally has this capability, UNIX command “shell” suites may be suitable.) These are particularly important for administrators who use command windows.
2. Monitor for the presence of executable files whose ACL indicates they were created by other than authorized users. Files whose names are the same as common system commands bear particular notice.
3. Avoid working in directories where users of lesser capabilities can create new files or modify existing files or their ACLs.
4. Where possible, use the Run item on the Start Menu, or the Task Manager accessed through the Security Window (Trusted Path). These search PATH but not the current working directory. Unfortunately, these are often impractical when you are actively working in a command window.
5. Monitor for the presence of files whose suffix is .EXE, .BAT, or .COM owned by other than users authorized to install applications (see “Application Directories” in *Application & User Home Directories*).

Note, however, that this threat is roughly equivalent to a penetrator simply placing an executable file in a directory that users might simply “open,” often by double-clicking.

### **Guidelines:**

#### **Levels 1 & 2:**

The preceding practices are *recommended*.

### ***Data Files that Hold Hidden Programs***

Many applications allow their “documents” to contain data that can run as a program in certain cases. An unsuspecting user who simply “opens” such a document may trigger these programs which often run unnoticed. This is a virulent spoofing opportunity.

The most common example is the “macro” viruses in programs like Microsoft Word. A user can attach programs written in a flexible programming language to these documents that the user activates via certain keystrokes or simply by opening or saving the document. As always, such macros run with the full capabilities of the unsuspecting user.

Examine a new application's documentation carefully for such traps before you install it. Sometimes you can unilaterally disable dangerous features. Otherwise, educate your users on the dangers and precautions when they use these programs. Unless absolutely necessary, withhold "X" access to these programs from administrators which helps to assure they don't run them accidentally.

### **Guidelines:**

#### **Levels 1 & 2:**

- ❑ Do not install an application on the system until you have assessed whether or not it presents this threat.

### ***CDROM Auto-run Programs***

CDROMs can have "auto-run" programs that run when you insert the CDROM or when you double-click ("open") the CDROM icon on your desktop. Fortunately, these do not activate when you log on to a computer with a CDROM present, which would be an easy and effective spoofing opportunity. As with other spoofs, malicious auto-run programs are likely to be invisible or appear benign. The Registry key:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Cdrom
```

can have a REG\_DWORD value named "Autorun" that controls this feature. Set its value to 0 to disable auto-run. Also, a user can depress the "SHIFT" key to forego auto-run where it would otherwise be triggered. (See [KBase] Q155217 and Q126309.)

### **Guidelines:**

#### **Levels 1 & 2:**

- ❑ Disable auto-run on all systems at Level 2, and on Domain Controllers and major servers at Level 1. While we *recommend* you disable it even at Level 1, you can leave it enabled if it provides useful site functionality, and if you educate users about the dangers of untrusted CDROMs and how to forego auto-run using the SHIFT key.

### ***Shortcut Spoofing***

If one user can redefine the properties of another's shortcut file, they can redirect the second user to a malicious, look-alike program. ACLs on shortcuts must prevent write access to users with fewer capabilities than the users that use the shortcuts. Note that shortcuts that appear on the Windows NT desktop and Start Menu are generally safe because they are stored inside the Profiles directory which by default is private. You can also caution users to create shortcuts only in their home directory tree and to disallow public write access to that tree.

### ***Protecting Standard Extensions***

Windows NT maintains a mapping of file name extensions (like ".TXT") and the programs that the system invokes when users perform certain actions (like opening or printing) an icon with that extension. This single mapping serves all users and should hold only trusted programs whose executable files are properly protected by ACLs.

By default, the INTERACTIVE pseudo-group, which includes all users who can locally log onto the computer, can extend or modify this mapping using the "View...Properties" menu on My Computer. This is a potent spoofing opportunity. Windows NT stores this mapping in the Registry under the key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Classes
```



which by default allows the INTERACTIVE group “Change” access. Replacing this group with one that holds only trusted users removes this threat.

Note that executables can be run from the command line regardless of their file extension (which usually is .EXE) so that attempting to remove access to an application by simply renaming it is futile. Rely instead on the file’s ACL.

*File System & Registry ACL Settings* prescribes the appropriate, tightened access to the Classes key.

### ***Defining Standard Extensions***

Even trustworthy applications can perform sensitive or unexpected actions and can be opportunities for spoofing. For example, \*.REG files hold scripts that cause REGEDIT to add entries to the Registry (subject to the Registry ACL permissions). The default “open” action for these files is to invoke the REGEDIT “import” function to automatically change the Registry. The user only sees a small window confirming that the Registry was updated.

#### **Guidelines:**

##### **Levels 1 & 2:**

- We *recommend* you disable actions for the .REG file extension, and any others that you deem to pose similar risks. At Level 2, we *recommend* you disable all extensions except those that are operationally necessary and that you assess to be trustworthy.

### ***Removing “R” from Program Files***

It is an excellent practice to remove the “R” permission from executable program files (although of course, not “X”). This prevents users from copying the program and using their own private copy. While a user cannot expand their capabilities by doing so, the copy may not be as strongly protected as the original and therefore represents a potential spoofing threat. Also, programs often consult the program file’s directory when searching for DLL’s and copying a program file to another directory may open this spoofing threat. (Note, however, that the Windows NT desktop manager cannot determine the icon for such files, and displays a default icon. This also produces many failed read attempts for sites that are monitoring file read failures, although they can disable the auditing of read failures on these files.)

#### **Guidelines:**

##### **Levels 1 & 2:**

- We *recommend* you remove “R” access from all executable files.

### ***Internet Browsers***

Spoofing can come from remote sources. The most potent popular threat is from WWW browsers that easily and transparently load programs from remote Web pages and execute them. Although beyond the scope of these current guidelines, such application threats are potent and dangerous.

### ***DLL Spoofing***

The DLL spoof is one of the most insidious spoofs in Windows NT. A Dynamic Link Library (DLL) is a software object module, or library, linked into a program while the program is running. DLL’s are a powerful feature that allow programs to share common code making them easier to develop and more efficient, and are used extensively in Windows NT.

The DLL code runs in the context of its host program and thus inherits the full capabilities of the program's user. A DLL spoof causes an otherwise trusted program (likely run by a conscientious administrator) to load a DLL with a Trojan Horse instead of the legitimate DLL. Once the Trojan Horse gains control, it can do anything the user can unbeknown to the user.

When programs load DLL's, they search a sequence of directories looking for the DLL. It is important that penetrators can't insert a "fake" DLL in one of these directories where the search finds it before a legitimate DLL of the same name. Confusing this issue is that the different methods search different sequences of directories.

DLL spoofing opportunities stem from the algorithm that the DLL linking algorithm uses to find the file that holds the DLL. (These files usually have the standard ".DLL" suffix.) There are a variety of places the linking algorithms search depending on the situation, but one can group them into three basic categories:

- ❑ **Program Directory:** This is the directory that holds the program's executable file.
- ❑ **System Directory:** One of the well-known and presumably protected system directories, like %SYSTEMROOT% (commonly "C:\WINNT"), or its SYSTEM32 subdirectory.
- ❑ **Working Directory:** This is the current working directory of the process, which may be the directory the user entered before running the program, or a directory into which the program placed itself. This is the only one of these three that cannot be readily protected because it is whatever directory the user has navigated into.

The problematic case is when the algorithm checks the Working Directory for the DLL file. To spoof a user, one must only insert a malicious DLL file in a directory that the user may be using as their Working Directory. The DLL file must have the same name as a legitimate DLL used by the otherwise trusted program. Instead of linking the real, trusted DLL, the algorithm links the fake DLL instead. It is a simple job for this malicious DLL to create a new process which runs under the administrator's full capabilities, then vector library requests to the real DLL which it can access in one way or another. (The details are a bit beyond our scope.)

Security is compounded by the fact that the user running a program may not know the Working Directory of the program, since the program itself may set the directory. Even a conscientious user seeking to avoid unprotected directories may have little way of knowing when their program is in one.

DLL's loaded from the Program Directory are not a serious concern because they should be as well protected as the program file itself. (That is, they should not be able to be written or replaced by untrusted users.) The System Directory is also presumed safe under the assumption that its ACLs are properly maintained as per these guidelines. The major caution in these two cases is the DLL files should be protected as strongly as the programs that reference them.

The system can also search the search path directories for DLL's, but only after the other options are exhausted.

The "KnownDLLs" Registry key:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\
  SessionManager\KnownDLLs
```

defines a set of DLL's the system loads at boot time. A sample listing of this key:

```
advapi32=advapi32.dll
comdlg32=comdlg32.dll
crt.dll=crt.dll
DllDirectory=Systemroot\System32
gdi32=gdi32.dll
```

The system consults the %SYSTEMROOT%\SYSTEM32 directory first (a System Directory) for KnownDLLs entries. The single exception is 32-bit DLL's loaded dynamically by a program through the LoadLibrary API, which ignores KnownDLLs. (Note the "DllDirectory" value in KnownDLLs. This is undocumented and we advise you not to change it unless you know how it is used.)

A DLL called by a DLL in KnownDLLs is also treated like a known DLL. Therefore, you only need to include the "roots" of the DLL calling trees in KnownDLLs. Unfortunately, it is hard for an administrator to know this calling hierarchy.

Choosing safe installation locations for DLL's is complicated because there are different algorithms for 16-bit and 32-bit DLL's, which also differ in their treatment of the WellKnownDLLs Registry entry. The following cases are safe locations for DLL's:

- ❑ 32-bit DLL's securely installed in the Program Directory.
- ❑ DLL's named in KnownDLLs stored in the SYSTEM32 directory, except when they are 32-bit DLL's loaded through the LoadLibrary API.
- ❑ Any securely installed DLL loaded through the LoadLibrary API by explicit, full pathname. These are usually ones in the system root directory tree.

Given the LoadLibrary problem, there are no absolute solutions to DLL spoofing. The following list is at least a basis for DLL safety. However, it can also cause considerable disruption and should not be undertaken lightly.

- ❑ All application DLL's loaded in the same directory as their application are safe as long as their ACLs protect them from change or replacement.
- ❑ Place all DLL files in the SYSTEM32 directory into KnownDLLs.
- ❑ Move all other DLL's into the SYSTEM32 directory and add them to KnownDLLs. Unfortunately, this may cause some programs that use these DLL's to malfunction, although you'll usually get a message to the effect that a DLL of a certain name could not be found.
- ❑ Constantly scan the system for files with the .DLL extension outside the SYSTEM32 directory or a recognized application directory.

Unfortunately, DLL's loaded through LoadLibrary that don't use absolute pathnames are the weakness that a penetrator would most likely target, and this case is not protected by these precautions.

For more information, see the [KBase] Q164501, "INFO: Windows NT Uses KnownDLLs Registry Entry to Find DLLs," and Win32 "LoadLibrary" API.

**Guidelines:**

**Levels 1 & 2:**

At this point no practices for containing DLL spoofing seem sufficiently complete to recommend. However, Level 2 sites should consider the precautions above as a basis for their own confinement policy for DLL spoofing.

## 19. User Responsibilities & Practices

---

General users are in part responsible for guarding their own environment and that of the groups in which they are a member. You should formulate site policies practiced by all users. Consider the following points for inclusion in the local user policy. This is by no means a comprehensive list, but includes important user practices with respect to these guidelines.

- ❑ Users must faithfully implement site practices for choosing new passwords, especially directions for choosing “random” combinations of password characters.
- ❑ They should keep their password private and not write it down. If users are likely to write down long passwords no matter what you counsel, it may be better to specify the manner in which written passwords are protected.
- ❑ They should never use their Windows NT logon password for any other purpose, like application-specific passwords or passwords on other operations system, especially DOS, Windows and Windows 95, and Macintosh.
- ❑ Users should only enter their password in a scenario that they initiate with the CTRL+ALT+DEL sequence, specifically: logon and logout, changing their password, and locking the workstation. The single exception is to connect to remote shares under another account which requires entering the account’s password.
- ❑ Users should not store passwords in files on the system, like batch files or startup scripts, especially the Windows NT logon password. While in theory these files can be kept private to the user, there are many risks and this is a dangerous practice.
- ❑ A user should lock their terminal through the CTRL+ALT+DEL sequence when they leave their workstation unattended if there is any chance others might use the workstation maliciously.
- ❑ Users should assure that any automatically locking screen savers the administrator installs remain enabled. (See the technique for hiding the screen saver page in the Control Panel in *System Policy Files*.)
- ❑ Most applications are not specifically programmed for Windows NT’s ACLs. To avoid potential problems, “documents” should retain the ACL that they would receive if newly created in the same directory. If you need to “tighten” the ACL on an individual file, either tighten the ACL on its directory also, or move it to another directory whose file defaults have the tighter ACL. (See the topic “Applications and ACLs” in [Sutt96], pages 102-104.)
- ❑ Users should understand that when they move or copy files and directories, the new copies can in some cases retain their original ACLs and other cases be re-protected according to the directories into which they are copied or moved. When in doubt, they should confirm the new ACLs. (See the topic “Copying and Moving” in [Sutt96], pages 98-100.)
- ❑ General users must not import unapproved, untrusted programs onto the system. Users allowed to do so should understand the practices for keeping these programs safe, mainly assuring that the general public cannot modify the program executable files, and DLL’s. They should also protect data files not intended to be changed by the general public.

- There are several anti-spoofing techniques that regular users can employ (see *Spoofing*). While it is asking a lot of most users to fastidiously follow them, a few bear particular user attention:
  - Users should never double-click an icon unless they are confident that it opens the expected application. Perhaps the more useful advice is “when in doubt, don’t run, open, or double click an icon.”
  - Users should keep personal programs and batch files in directories under their control where other users cannot add, modify, or replace these files.
  - Similarly, they should only include directories in your PATH environment variable where other users cannot add, modify, or replace files.
  - Auto-run programs on CDROMs can trigger when the CDROM is inserted or its drive icon opened. Users should avoid using untrustworthy CDROMs, and should depress the SHIFT key to bypass auto-run, when enabled, if there is any doubt as to a CDROMs trustworthiness. (See “CDROM Auto-run Programs” in *Spoofing*.)

## 20. References

---

The following are the major guidelines for securing Windows NT drawn upon or at least taken into account for the preparation of these guidelines (in order of influence):

- [Sutt96] *Windows NT Security Guide*, by Stephen A. Sutton, Addison-Wesley, ISBN 0-201-41969-6. Written by the author of this study, and the basis for many of its techniques. It is a tutorial on Windows NT security concepts and gives extensive instructions for securing Windows NT.
- [Mayer96] *Less Well-Known Considerations for Configuring a Secure Windows NT System*, Frank L. Mayer, SAIC, March 29, 1996. No longer available. This is a seminal paper by one of the C2 evaluators which formed the basis for several subsequent studies. Although this paper is no longer available (or up-to-date), its ideas are subsumed by these guidelines.
- [Micr97] *Securing Windows NT Installation*, Microsoft Corporation, April 10, 1997. Available on their Web site: <http://www.microsoft.com>
- [TFM] *Windows NT C2 Security Administrator's Guide*. This is the Trusted Facility Manual required for C2 evaluation, but to our knowledge is not generally available to the public. We reviewed a undated, final draft copy.
- [Navy97] *Secure Windows NT Installation Guide*, Department of the Navy, Space and Naval Warfare Systems Command, Naval Information Systems Security Office, PMW 161, November, 1997. Newly released and generally available to the public. Although not designed for general use, it contains specific directions for installing Windows NT for the U.S. Navy.

---

The following Windows NT product documentation was helpful in formulating the guidelines:

- [ConPln] *Concepts and Planning Guide*, Windows NT Server Manuals, version 4.0. Available on the Windows NT distribution CDROM.
- [KBase] Microsoft's *Knowledge Base*, available in several forms, including [TchNet], [RKitW], [RKitS], and at <http://www.microsoft.com>. References to Knowledge Base articles are given by their unique access code number, for example "Q164501," that you can readily search for using the respective tool's search capabilities.
- [NetGd] *Networking Guide* in [RKitS].
- [NetSp] *Networking Supplement Manual*, Windows NT Server Manuals, version 4.0. Available on the Windows NT distribution CDROM.
- [RKitW] Microsoft's *Resource Kit* for Windows NT Workstation 4.0. This is a commercial product distributed on CD-ROM that includes documentation and software tools. It is typically included in [TchNet].
- [RKitS] Microsoft's *Resource Kit* for Windows NT Server 4.0. This is a commercial product distributed on CD-ROM that includes documentation and software tools. It is typically included in [TchNet].

[TchNet] Microsoft's *Technet* information service, available by subscription and distributed monthly on CD-ROM's. References to items in Technet are given by their title that you can easily search for using Technet's search capabilities.

---

We reviewed several commercial books on Windows NT security in the preparation of these guidelines:

- [Micr94] *Windows NT 3.5 Guidelines for Security, Audit, and Control*, Microsoft Press, ISBN-1-55615-814-9. An early and general introduction to Windows NT security, but has not been updated for later versions.
- [Shel97] *Windows NT Security Handbook*, by Tom Sheldon, Osborne McGraw-Hill, ISBN 0-07-882240-8. This book summarizes the Microsoft and other documentation on Windows NT, and is a good general reference to Windows NT security.
- [Sutt96] *Windows NT Security Guide*. (See reference above.)

The following textbook is an excellent (albeit technical) presentation of networking security issues:

- [Kauf95] *Network Security: Private Communications in a Public World*, by Kaufman, Perlman, and Speciner, Prentice Hall, ISBN 0-13-061466-11.

---

We scoured innumerable Web sites and documents and would like to note the following:

- ❑ *NT Security – Frequently Asked Questions*, an **excellent** collection of Windows NT security resources on the Web. Most of these sites were reviewed during this study but are not specifically named. Available at:  
<http://www.it.kth.se/~rom/ntsec.html>
- ❑ *Windows NT Security Issues*, at:  
<http://www.somarsoft.com>.
- ❑ *Known NT Exploits*, a database of known Windows NT “hacks,” available at:  
<http://www.secnet.com/ntinfo/index.html>
- ❑ Product documentation on software tools from ISS, Inc., available at:  
<http://www.iss.net/eval/manual/nt/index.html>
- ❑ *Final Evaluation Report, Microsoft, Inc., Windows NT Workstation and Server*. This is the final report (“FER”) from the C2 evaluation. While it gives no configuration advice, it is an excellent treatise on the security structure of Windows NT. This document *may* be available from the National Computer Security Center, Ft. Meade, MD.
- ❑ *Department of Defense Trusted Computer System Evaluation Criteria*, CSC-STD-001-83, December 1985. Commonly called the “Orange Book,” this document defines the criteria for C2 evaluation. However, it is not general reading and not useful in understanding how to configure Windows NT securely. Please note that C2 is not an operational guideline – it does not address how a computer system is to be configured in any particular environment.



## 21. Reference Material

---

Setting file system SCL's from the tables in Chapter 7 is admittedly a bit tedious. The following script, from the AdvancedChecker library (written in the Checker language), sets common file system ACL's, including the all-important WINNT directory in the spirit of the tables in Chapter 7. (It's does not implement the tables exactly, taking a few additional liberties.)

```

/*=====
                                NTFS Drive ACL's

Checks or sets the ACL's on main drive folders like "C:" and
their close descendents, like the files and C:\TEMP. Uses the
advice in the "NSA Windows NT Security Guidelines" in included
in the "doc" directory of Advanced Checker.

=====*/
include_file "Utils_H.ckr"

slist %Verbose = true // prints more!!!

/*=====

%NTFSDriveACLs Sets or checks these files for a specified drive.

Automatically detects whether target host is NT Server or Workstation.
On errors (like a file missing or no access) the function prints a
message, but continues on.

*/
func %NTFSDriveACLs
( %Drive,          // root name, e.g., "C:\WINNT\"
  %Action          // "check" or "set"
)
  //-- Internal worker that does the actual set/check

  func %SetOrCheckACL (%Obj, %ACL)
    if %CheckOnly
      if not CheckACL (%Obj, %ACL, limit)
        print { "*** ACL too broad:", %Obj } // Opt: Enhance!
      elseif %Verbose
        print { "--- ACL okay:      ", %Obj }
      end_if
    else
      call SetACL (%Obj, %ACL)
      if %Verbose
        call %PrintACL_1 ({"Setting: ", %Obj}, %ACL)
      end_if
    end_if
  end_func

  //-- Check for valid parameters

  if %Action $= "check"
    slist %CheckOnly = true
  elseif %Action $= "set"
    slist %CheckOnly = false
  else
    return exception "Invalid 'action' option"
  end_if

  //-- Save and adjust options

  slist %OptsSave = GetOptions ()
  options on multiSID
  options on readpriv error end_error // ignore failures
  options on writepriv error end_error

```

## UNCLASSIFIED

```
//-- Make sure %Drive ends in "\" -- adjust it if not
if Last (Expand (%Drive)) $!= "\"
    set %Drive = Compress ({Expand (%Drive), "\"})
end_if

//-- Determine whether target host is NT Server or Workstation

if %NTProductType () $= "NTServer"
    slist %IsServer = true
elseif %NTProductType () $= "NTWorkstation"
    slist %IsServer = false
else
    return exception "host is not a NT Server or Workstation"
end_if

//-- Define basic permission components

slist %Public = "Everyone" // NOTE: Authenticated Users popular,
// but makes no difference

slist %PwrUsr = "Power Users"
slist %SrvOps = "Server Operators"
slist %Installers = "Installers" // OPTS: %PwrUsr on NTWorkstation

slist %PubRead = %Public >< ":read"
slist %PubChange = %Public >< ":change"
slist %Admin =
    {"Administrators:full", "SYSTEM:full", "CREATOR OWNER:full"}
// OPT: add "Server Operators:full"
slist %Standard = {%PubRead, %Admin}

slist %InstChange = %Installers >< ":change"
slist %BckOpsChange = "Backup Operators:change"
slist %PrtOpsChange = Pick (%IsServer, "Print Operators: change", {})
slist %ReplFull = Pick (%IsServer, "Replicator:full", {})

slist %SrvOpsChange = Pick (%IsServer, %SrvOps >< ":change", {})
slist %SrvOpsFull = Pick (%IsServer, %SrvOps >< ":full", {})

slist %PwrUsrChange = Pick (%IsServer, {}, %PwrUsr >< ":change")
slist %PwrUsrFull = Pick (%IsServer, {}, %PwrUsr >< ":full")
slist %PwrUsrSpec1 = Pick (%IsServer, {}, %PwrUsr >< ":RWXD/WXD") // special

//
// NOTE: We carefully avoided setting items then setting them later to their
// final, correct value, because this "set twice" scheme means we can't call
// CheckACL in place of %SetOrCheckACL.
//
// Opt: In almost all cases, %Standard could probably include %SrvOpsChange

scope
//-- Error handler for using during actual traversal
// NOTE: SCOPE used primarily to limit %DefaultErrorHandler to
// these statements

func %DefaultErrorHandler
    if not ((ErrorNumber () = 2) or (ErrorNumber () = 3))
        // NOTE: Ignore "not found" errors
        print { "*** Error on: ", ErrorObject (), ErrorNumberText () }
    end_if
    // OPT: Could abort on any error, for example:
    // return exception AllErrors ()
end_func

call %SetOrCheckACL (%Drive, {%Standard, %InstChange,
    %SrvOpsChange, %PwrUsrChange})
walk %Obj %Drive walkopts {nobase, immediate, files}
    skip {"boot.ini", "ntdetect.com", "ntldr",
        "ntbootdd.com", "pagefile.sys"}
    call %SetOrCheckACL (%Obj, {%Standard, %InstChange, %SrvOpsChange})
end_walk
```

**UNCLASSIFIED**

```
for %Obj in {"boot.ini", "ntdetect.com", "ntldr", "ntbootdd.com"}
    call %SetOrCheckACL (%Drive >< %Obj,
        {%Standard, %InstChange, %SrvOpsChange})
end_for

call %SetOrCheckACL (%Drive >< "Temp",
    {%Standard, %Public >< ":RWX//", %SrvOpsFull, %PwrUsrFull})

    call SetOptions (%OptsSave)
end_scope
end_func

//=====
//
// Sample caller

host "Actg-Serv" // *** use one of your hosts here; C$ must be available!
    call %NTFSDriveACLs (RemotePath ("C:"), "check")
end_host
```

**☺☺ The End ☺☺**