Patrick Grace

GSEC Practical Assignment Version 1.4 Option 1

Title: Steps Toward a Secure Windows XP Stand Alone Workstation

Abstract

These pages constitute a “how to” guide for configuring public access computers to protect them from user alterations. Past and present schemes to accomplish this goal are mentioned and expanded upon. Specific steps and recommendations are laid out toward creating a Secure Windows XP Stand Alone Workstation. All tools used are of no cost beyond the effort to implement them. An introduction is given to the use of the Microsoft Management Console as it relates to policies and profiles. Basic security concepts are subtly intermixed with the procedures. In the end a shortcut to replicating the configuration to other computers is explained.

Steps Toward a Secure Windows XP Stand Alone Workstation

Purpose

One element useful in security is secrecy. Certainly it helps to hide a key to prevent access to a locked asset. Yet presenting the concepts used to design the lock to public review is necessary. The knowledge and experience of others can help design a complicated lock and bring to light the vulnerabilities of the current design. This work details steps taken to provide security for a very specific environment. It is hoped that publication of these ideas and methods will lead to an effective security model backed by discussion. That dialog will build a more effective security procedure.

The Task

The task is to provide a computer or multiple computers for use by the public. By itself, this is a simple task. In response to a transfer of sufficient funds any computer manufacturer will deliver a complete working computer system with any and all options. Simply unpack the components, follow the colorful setup map, and push the "on" button. The task becomes interesting when other goals are added. Consider meeting the following additional goals:

1. The computers must present the same look and function day after day.
2. Disk drives and file structure must be available to the user.
3. The computer must run needed but not pirated or damaging programs.
4. The computer is a stand-alone system without the benefit of a server.
5. Windows XP is the operating environment.

1. In a classroom it is helpful if the instructor sees the same display the students see. The "second icon on the left" should be same icon on every computer in the room. In contrast, one of the selling points for any desktop Graphic User Interface (GUI) is its ability to be customized. Users have become accustomed to having their desktops arranged their way. The concept that a public access computer is shared seems foreign to most users. Users see nothing wrong with changing a classroom, computer lab or library computer to accommodate their tastes. This is without regard to the overall usefulness of the system to the next person seated in their place. Some "customizations" are done innocently and some are done with malice. In view of the goal the intent of the user is irrelevant. What is relevant is that the computer is usable for the stated function. On the other hand, users expect computer professionals will have the tools and skills to clean up any mess they can create through negligence or simply as a byproduct of productivity.

2. Users have a reasonable expectation they will be able to save the work they have done. For speed and long-term projects the fixed disk is the storage medium of choice. For portability, removable media like floppies and CDs are

needed. This means that the Windows "My Computer" must be available. Can any computer's file system be left open to browsing and still remain secure? Multi-user Unix systems have long allowed extensive file browsing. This is especially true when users are allowed to make publicly readable web pages available. Windows with NT Files System (NTFS) with carefully set permissions can compete with the Unix Network File System (NFS). [1]

3. The computer industry has made restricting a workstation environment nearly impossible. The ongoing aim has been to make computers usable by individuals with no computer skills. Although unattainable in the absolute, only minimal skills are required to employ the complex power computers offer. It is an incredibly simple task to install and run applications. Programs can be introduced to a workstation via a CD or as a download over the Internet by anyone at any skill level. These programs can include utilities aimed at circumventing security efforts. Although we may not be able to stop the installation, we may be able to stop the execution of rogue applications. Windows provides a facility to rub only specific applications.

4. The task of providing a secure desktop environment is not new to administrators that work in a server environment. One of the roles of a Domain Controller is to provide a user profile that is protected from alteration. However, in a small scale installation the maintenance of a server environment is not cost effective. The server hardware and software come with a large price tag. Dealing with the security of a server is often a larger task than providing security to multiple desktops if a method to successfully duplicate a single configuration is devised.

5. The Windows XP operating environment has been designed with security as an integral part. It should be clear Microsoft is a large and capable corporation. They make large mistakes and are capable of large solutions. One such mistake was to ignore the importance of security. Microsoft security has been compared to jumbo shrimp as adequate definition of an oxymoron. However, Microsoft has recently turned a collective eye toward security. The undeniable capability of Microsoft predicts the perception of security flaws in Windows will linger longer than the actual vectors for system compromise. [2]

Security Considerations

Security maintenance can be stated in other words as providing protection. Protection is insured by sufficient defense. Before determining what is a sufficient defense, it should be determined what needs protection. Our first priority includes the phrase "day after day". We must protect the workstations so they will work predictably everyday. The integrity of the system configuration must be maintained. The user may not be allowed to customize the system by changing the look or by adding or removing functions. Although daunting, the task is made a bit easier by also examining what does not need protection. The computers under discussion are public access workstations. They are not servers. Therefore they will not contain confidential information. There are no secrets to hide. We can make all the files on the system readable to the public. In fact, no unique information is stored on these computers. Routine backups are not required to protect newly created files. The installation is essentially static.

Although no unique information is stored on these computers, a backup plan is needed to provide consistent availability of the systems. If a computer goes down it must be restored quickly so its functions are available to the users. If the solutions provided later were applied to a single computer, then a backup in the form of a disk image would be essential. However, typically these solutions will be applied to several identical systems. In that case the restoration of a lost system due to a component failure is simply a matter of taking a disk image from one of the remaining systems. In essence you have as many backups as you have similar systems. Therefore, complete backups, which are the foundation of single system availability assurance, are provided automatically by multiple identical configurations.

Method

History

In July of 1996 an article appeared in PC Magazine that offered security for a stand-alone public access Windows 95 system.[3] The scheme employed a clever use of policies and user accounts. Poledit.exe is a GUI for changing registry entries that pertain to policies. The Policy Editor tool (poledit.exe) was used to create three user accounts: Administrator, Guest and Dummy. The administrator account was created without restrictions. The guest account was created with settings that were dependent on what the guest user was allowed to do. Windows NT 4.0 included 72 policy settings that were used to define the guest account. The dummy account was created with the most restrictive settings that resulted in no access to system resources or programs. The dummy account was meant to provide for an interesting aspect of Windows 95 login and policies.

With Windows 95 a user could add a new user by simply typing a new name in the username dialog box. Further, a user could bypass the login dialog altogether by simply hitting cancel. In either of the previous two examples the result would be a default profile with no restrictions. The dummy account provided for those situations by creating a default profile with no privileges. Windows XP does not allow for either of these unfortunate aspects of Windows 95 by requiring administrator created user accounts and does not offer a cancel option to the login process.

The procedure outlined in 1996 was a bit obscure probably caused by the length constraints imposed by a magazine article. The core concept of the method was clear. A profile was created for each user as might be done in an environment containing an NT server. The only difference between server based profile security and stand-alone profile security was the location where the profile was stored. Normally a user would log in and the profile would exist on the NT server. The policy editor allowed for a local update of the profile. The profile was stored in a file called config.pol and the registry was set to obtain the profile from that file stored on the local fixed disk instead of from the server. Applying that facility to specify the location of the config.pol file and locating it on the local hard drive gave birth to security of a stand-alone system.

A doubt lingers as to if the procedure outlined in 1996 for Windows 95 might be applied to Windows XP. In principal all the same elements appear to exist. It appears the location of policies for individual users can be read from the local hard drive. The stumbling blocks to exploring this pathway to solving the current puzzle are the .adm files and poledit. The .adm files for Windows 2000 and XP are Unicode files with new options not present in the NT .adm files. These changes have made the .adm files unreadable by the readily available poledit.exe program included on the Windows 95 installation disk. There exists a version of poledit that will read the current .adm files but it is not readily available. Microsoft elected to only include the new poledit.exe on the Windows 2000 server CD. Purchasing Windows 2000 Server to obtain that tool would partially go against the goal to cut cost by not using a server environment. It is hoped a reader with the current policy-editing tool will have cause to explore this possibility. For now we must deal with group policies instead of individual account policies.

Present

Windows XP has brought some new items to the topics of profiles and policies that both help and hinder the current project. Microsoft shifted support from user policies to group policies beginning with Windows 2000. The GUI configuration tool is now the Microsoft Management Console (MMC) with the Group Policy plug-in instead of Policy Editor. Both MMC and Poledit use Administrative Template files that use the .adm extension. The adm files included with Windows 2000 and XP provide for over 600 settings as compared to the 72 settings provided with Windows NT. The additional settings make the resultant configuration more flexible and powerful. It is also now possible to restrict accesses to items that were not controlled before. The down side to MMC is the inability to define local user profiles. Policies set for a Windows XP professional desktop apply to all users of the workstation. That includes the administrator. Obviously the administrator will have a need to control some aspects of the workstation that should not be altered by a typical user. Earl Grylls nicely circumvented this problem in his article Building a Kiosk in Win XP.[4]

Heart of the Matter

As might be expected group policy settings are stored in a file. Upon login Windows reads the group policy file in the name of the user and applies the included settings[5]. Windows XP with NTFS allows permissions to be applied to files. Permissions control who may perform operations such as reading or writing specific files. The essence of this entire procedure is this: if the administrator is denied read permission of the group policy file then the group policy settings are not read and therefore are not applied to the administrator upon login! This procedure might rightly appear as a "work-around". Microsoft suggests this method may be valid[5]. It must be remembered there are instances when the administrator must not be denied read and write permission for the group policy file. Switching the permission of the group policy file appropriately and consistently introduces some danger.

Caution

So, the administrator must use the MMC with the group Policy plug-in to configure a secure environment. The administrator must also maintain the ability to change the permissions of the group policy file so they may be set to "read and write deny". Those permissions provide for an unrestricted environment for the administrator upon login. Using MMC requires "allow read and write" permission of the group policy file. Once the changes are made with MMC the permission must be reset to read and write deny. It is possible for the administrator to forget to change the permissions before logging out. The result would be the policy settings applied to the administrator account upon the next login. That will disallow the administrator access to MMC. Another danger exists.

It appears that the group policy is reread and applied as soon as the MMC is closed. This means all the restrictions of the group policy are immediately applied to the administrators account. As a result the administrator may need to set the correct deny permission for the group policy file, log out and then log back in to obtain an environment created without the restrictions imposed by the policy of the group. Earl Grylls cautions: go slow when developing group policies. He believes the administrator could be locked out of the MMC by the group policy. On occasion this has happened during development of this paper. If this happens, first try to take ownership of the group policy file. Then change the read and write permissions to deny and log out and then back in.

Step-by-step

The following is the step-by-step process of implementing a group policy for a standalone XP workstation. Explanations have been included so you might understand why steps are taken. A simple list of steps is included as an appendix.

Let’s assume a fresh installation of Windows XP Professional has been performed. Log in as the administrator by simultaneously depressing the Ctrl, Alt and Shift keys. Enter the username administrator without a password since one has not been established. Go to the control panel and assign a password for the default user account and the administrator account. Windows XP requires a default account be established during installation. Although the default account is part of the administrators group, it is not the administrator account. Create a user account. In this example the account will be called student. Set the account to limited. Assigning a password is optional but the password will need to be publicly available to allow users to log in. There is little point in setting a password. When creating the user account you might be tempted to use “guest” as the account name. You will not be allowed to do that because “guest” is a predefined account with specific properties. Thankfully it is set to inactive by default and is best left that way.

The MMC is an extensible GUI interface meant to consolidate many maintenance tools. Details of using the MMC can be found at the Microsoft Web Site[6]. For our purpose you need to add the Group Policy snap-in. This plug-in allows for registry changes to be made via the MMC GUI. Which registry items may be changed is determined by the contents of the .adm files[7]. The .adm files can be found in the Group Policy ADM subfolder. The registry items are arranged in a tree like structure seen in the registry itself. Let’s change some minimum settings.

Browse through the Group Policy Template starting with Local Computer Policy - User Configuration - Administrative templates. The next settings will be under Windows Components. Browse down to “StartMenu and Taskbar” and enable “Remove Programs and Settings Menu”. This will prohibit changing the appearance of the StartMenu. Still under Windows Components, browse to Control Panel. Here enable “Prohibit access to the Control Panel”. This step is taken to prohibit the student from changing the password for the student account. Continuing under Windows Components open “System” and enable “Prevent Access to the Command Prompt”. Finally, also under “System” enable “Run Only Allowed Windows Applications” This is a very powerful setting.

Once “Run Only allowed Windows Applications” is enabled you will be required to enter the name of executables file. For example, to include notepad you must enter notepad.exe. The path to the executable file is not required. The list of executables you include should be chosen to be complete with respect to the user’s needs. A side benefit of this setting is realized if you do not include setup.exe or install.exe. Further, although virus protection software should be installed along with standard Windows hardening procedures, many viruses are installed when executables are loaded. A virus called wipeout.exe, for example, would not be included in the “allowed” list and therefore would not run.

You may change other items but beware. As stated previously, all settings are applied to the administrator’s account upon closing MMC. You might restrict your own access to the gpt.ini file. Without that access you will not be able to set the permission to read and write deny and never be able to disable the restrictions to the administrators account. Should you find yourself in this situation there may be a way out.

There is nothing quite like command line computer operation. Please be introduced to the cacls.exe program. “Cacles.exe can be used to display or modify access control lists (ACLS) for one or more files at a time.”[8] With cacles you can boot to a command prompt, change the permissions on gpt.ini to deny read access for the administrator account. Thus once again you gain control of the desktop.

Continuing on, save the MMC settings with the default name “console1” and exit. Beyond configuring a restricted environment for all local computer users, the past step created the c:\windows\system32\GroupPolicy folder and its contents. The group policy file (gpt.ini) has been created. Among other things within that folder is the Adm folder. It contains the four default .adm files used by the Group Policy snap-in and gpt.ini file.

You will remember the permission settings of the gpt.ini file are what make this procedure work. Since Windows Explorer does not normally display the gpt.ini file, we need to perform the next three steps.

1. From “My Computer”, browse to c:\windows\system32.
2. Using the “tool” pull down menus select “folder options” and then “view”.
3. Check “Show Hidden Files” and uncheck “Use simple file sharing”.

The GroupPolicy folder will now appear within the c:\windows\system32 folder. Browse into the GroupPolicy folder and right click on the gpt.ini file. Select properties, security and then advanced. Uncheck “Inherit”, answer copy to the warning and click OK. These past steps insure that the permissions we set for the gpt.ini file will not be unexpectedly changed by permissions of folders above it in the file structure.

Log out as administrator, log in as student, and log out as student. These steps allow Windows to complete the student account creation. It is likely registry changes are made. During the research behind this document, registry changes were not monitored. However, after several steps a search was done for files that had been created during the current day to determine what files were involved in each step. Several files, folders and subfolders are created for the student account within the “Documents and Settings” folder.

Closing doors

At this point many of the key security objectives have been obtained. We have specified only those programs we will allow and certain aspects of the user's desktop. Now the real challenge begins. What other user powers can be brought to bear against our goals? As stated earlier, exposing these methods to public discussion will go a long way towards finding open doors into this stand-alone security scheme.

Any user can delete an icon from the desktop by highlighting the icon and depressing the delete key. Likewise icons could be added by right clicking on the desktop and selecting "new shortcut". To prevent alteration of the desktop, browse to the c:\documents and settings folder. First browse to the "All Users" folder. Right click on the "Desktop" folder and select "properties". Select the security tab and then the advanced tab. Check "Deny" for the following items:

- Create Files / Write data
- Create Folders / Append data
- Delete Subfolders and Files
- Delete
- Change Permission
- Take Ownership

Browse to the student desktop folder and repeat the settings.

Replicating

The procedure above can take a bit of time to perform. Certainly you would not want to repeat the process for a room full of workstations. Happily, you can take a short cut. The first steps of creating the accounts, adding passwords and running MMC with the group policy plug-in must be performed at each workstation. Once those steps have been completed you do not need to navigate through the group policy settings and reenter the names of all the applications you wish to authorize. From you initial installation copy the following items:

- Administrator Desktop folder
- All User Desktop folder
- Group Policy/ADM folder
- Group Policy/Usr folder
- gpt.ini file

Log into the workstation to receive the replication and copy those items into the appropriate locations. Change the permissions on the All User Desktop folder, the Student Desktop folder and finally the permission to read and write deny for the gpt.ini file for the administrator.

Summary

These pages have highlighted and defined a need for security of a stand-alone Windows XP workstation. Two previous works first developed for Windows 95 and then Windows XP were built upon. The aim has been to explain those works and suggest extensions to them. It is hoped readers will examine and use the methods presented here. Finally the users will prove them robust or make them so by discovering security weaknesses and suggesting improvements.

Appendix

Log off
Log on as administrator
Assign password to default account
Create student account, set to limited and assign password
Run MMC
Add group Policy snap-in
Open local computer policy- user configuration- administrative templates and click on system
Set programs to run
Close and save MMC as console1
Open c:\windows\system32
Pull down tool- folder options- view.
Check “Show Hidden Files”.
Uncheck “Use simple file sharing”.
Right click c:\windows\system32\GroupPolicy\gpt
Select properties- security- advanced
Uncheck “Inherit”- copy- OK
Check deny read- OK- yes
Log off as administrator
Log on as student
Log off as student

To move to another computer

Log on as administrator
Copy Administrator Desktop
Copy All user Desktop
Change All User Desktop Folder Permission to Read Only Do not allow Delete
Copy Group Policy ADM Folder
Copy Group Policy USR Folder
Copy GPT file
Change GPT property to deny read by administrator

References

[1] Caldera OpenLinux vs. Windows
Tom Henderson
http://www.cnn.com/TECH/computing/9905/19/ntvlinux.ent.idg/

[2] Report: Linux hack attacks on the rise
Matthew Broersma ZDNet (UK) July 15, 2002
http://techupdate.zdnet.com/techupdate/stories/main/0,14179,2874408,00.html

[3] Create Secure User Profiles with Windows 95's Policy Editor
Neil J. Rubenking
PC Magazine July 1996
http://www.etwebtools.org/policy_editor.htm

[4] Building a kiosk in Win XP
Earl Grylls Feb 2002
http://searchsystemsmanagement.techtarget.com/tip/1,289483,sid20_gci804962,00.html

[5] Group Policy Storage
© 2002 Microsoft Corporation. All rights reserved.
http://www.microsoft.com/WINDOWS2000/techinfo/reskit/samplechapters/dsec/dsec_pol_cxxv.asp

[6] Introduction to Configuration and Management
Local Group Policy Objects
© 1985-2001 Microsoft Corporation. All rights reserved.
http://www.microsoft.com/WINDOWS2000/techinfo/reskit/en/ProRK/prda_dcm_mavn.htm

[7] Group Policy Registry Table
© 2002 Microsoft Corporation. All rights reserved.
http://msdn.microsoft.com/library/default.asp?url=/library/en-us/gp/GPRef.asp

[8] Microsoft Windows XP Professional
Resource Kit Documentation
Page 714.