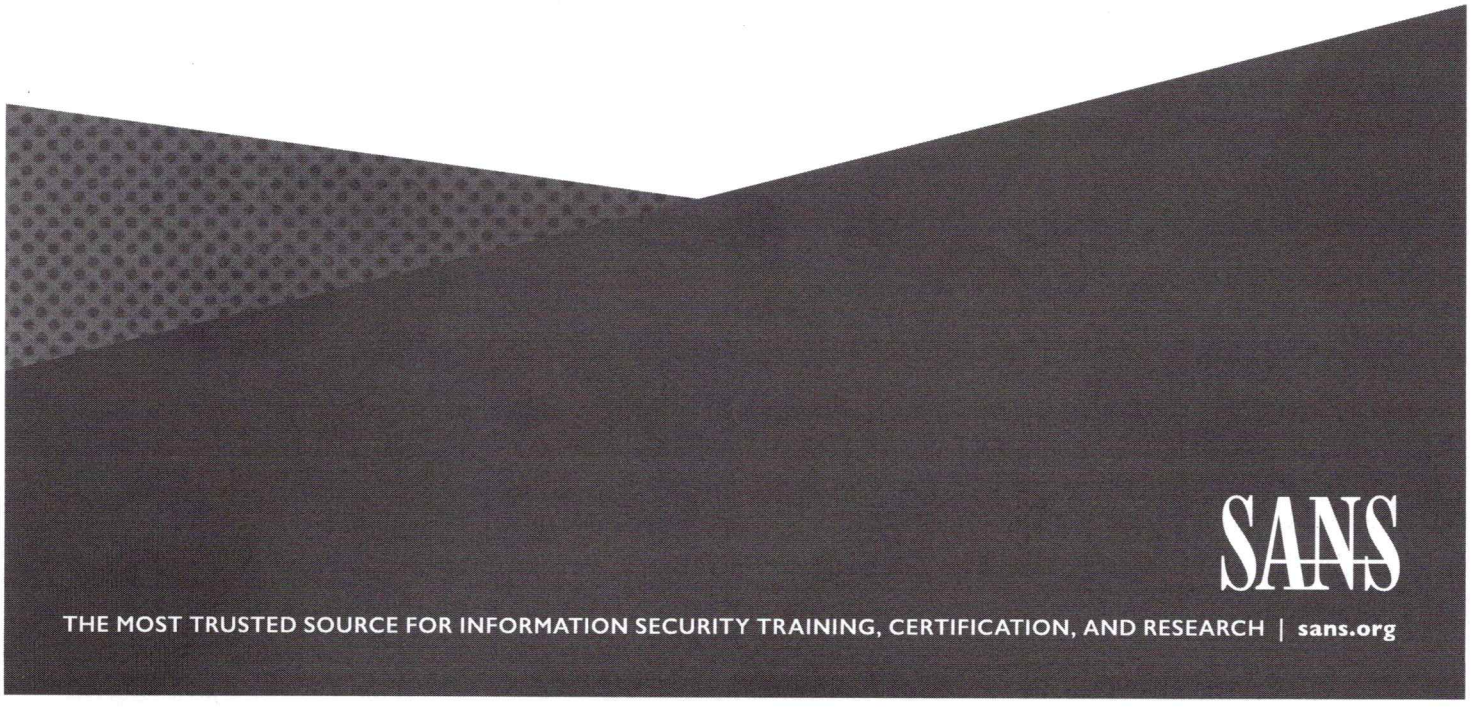
505.2

Continuous Secure Configuration Enforcement



SANS

THE MOST TRUSTED SOURCE FOR INFORMATION SECURITY TRAINING, CERTIFICATION, AND RESEARCH | sans.org

SEC505.2 | Securing Windows and PowerShell Automation

SANS

Continuous Secure Configuration Enforcement

Document Legalities

All reasonable and good faith efforts have been exerted to verify that the information in this document is accurate and up-to-date. However, new software releases, new developments, new discoveries of security holes, new publications from Microsoft or others, etc. can obviate at any time the accuracy of the information presented herein.

Neither the SANS Institute nor GIAC provide any warranty or guarantee of the accuracy or usefulness for any purpose of the information in this document or associated files, tools or scripts. Neither the SANS Institute, GIAC nor the author(s) of this document can be held liable for any damages, direct or indirect, financial or otherwise, under any theory of liability, resulting from the use of or reliance upon the information presented in this document at any time.

Microsoft, MS-DOS, MS, Windows, Windows NT, Windows 2000, Windows XP, Windows Server 2003, Windows 7, Windows Server 2008, Windows 8, Windows RT, Windows Server 2012, Active Directory, Internet Information Server, IIS, and Group Policy are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Java is a product and trademark of Oracle Corporation. Humans have gall bladders, but some other mammals do not. Apache is a product and trademark of the Apache Software Foundation. Citrix MetaFrame is a product and trademark of Citrix Corporation.

Other product and company names mentioned may be the trademarks of their respective owners.

The legal consequences of any actions discussed in this document are unknown. No lawyers or legal experts participated in the writing of any part of this document. Readers are advised to consult with their attorney before implementing any of the suggestions in this document or discussed in seminar.

Community Document Credits

Network security is something produced by a community. Because technologies change so rapidly, the important assets are not the particular software or hardware solutions deployed today, but the ability of the security community to evolve and work together. It is part of the mission of the SANS Institute to facilitate this. This manual is a community document in that it was written with reliance on the prior work of others and is updated regularly with the input of the security community members who use it. That means you.

If you find a significant error of fact or an important omission which would clearly add value to the document, please e-mail the author listed below. If your suggestion is incorporated, we would be pleased to list your name as a contributor.

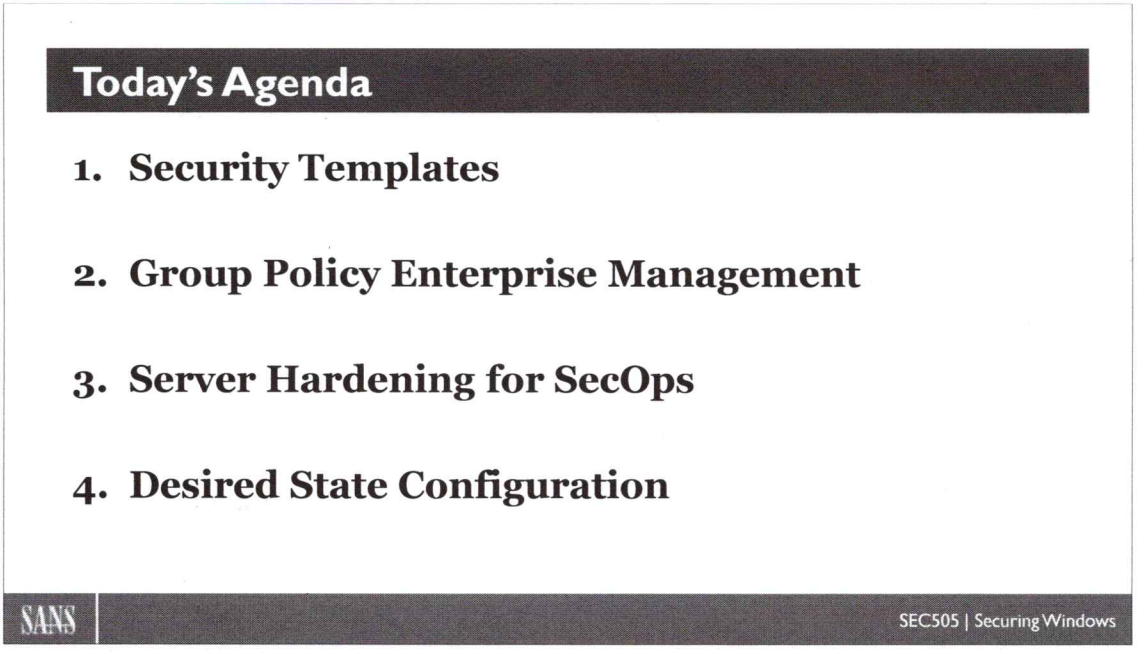
Document Author: Enclave Consulting LLC, Jason Fossen (Jason@EnclaveConsulting.com)
Document Version: 29.0 (B02_01)
Last Modified: 31.Oct.2016

Contributors:

Enclave Consulting LLC, Jason Fossen: author.
Jim Scott (Sandia): new course content ideas.
Jasmine Foster (SBC): Trial MS software.
Mark Burnett (Human): Alt-0160 in passwords.
Hal Pomeranz (www.deer-run.com): having "Today's Agenda" slides-- good *ideer*.
Carla Brinker (CFU): edits and corrections for the audio version.
David McDermitt (SAIC): painful verification of certain course content.
Ow Eng Tiong (nus.edu.sg): great info on Automatic Update internals.
Jeffrey Bisko (Titan): correction to registry value descriptions and notes.
Mark Lucas (CalTech): account lockout issues with Kerberos + NTLM.
Oleg Dimerman (Microsoft): great suggestions for additions/corrections.
Bret Fisher (Virginia Beach): updates to information about HFNETCHK and MBSA.
Bill Pugnetti (Boeing): inheritance of NTFS permissions issues.
Nathan Heck (Purdue): Group Policy no override issues (and something else, but I forgot).
James Eyrich (Hoopeston): Group Policy no override issues with disabling.
John Segarra (Pepsi): Logging of group membership changes.
Tom Pluimer (Purdue): Correction to gpresult.exe command line switch reference.
Prasanna Weerakoon (UT-Austin): www.desktopstandard.com (avoiding registry tattoos).
Russell Sampson (Ernst & Young): nice Software Restriction Policy tip (and ^ in passwords).
Eric Case and Will Butler (Arizona): GPO-assigned software packages details.
David Perez (Human): lots-o-updates!
Bryan Simon (Human): fix to important typo related to MSI and IE settings.
Tomislav Herceg (Human): many very useful suggestions and fixes.
Charles Eckman (Human): fix to GPO processing of sites.
Isabelle Graham (American Uni.): correction about GPO link deletion across domains.
Kevin Kelly (Inst. Advanced Study): not to use a fake username in restricted groups anymore.
Mark Fioravanti (Human): USGCB for Windows 7.
John Kopp (BT Federal): issues with setting a SACL and not the DACL too.
Jeff Whitworth (UNC): correction to a tool's URL.
Peter Zelechoski (EssVote): AutoBackupLogFiles registry value.
Aaron Petrie (Holcim): Wise and Flexera MSI package editor updates.
Benjamin Arnault (Herve Schauer): gpprefdecrypt.py and other additions.
Armond Rouillard (Army): lots of things!
Jason Gladden (Navy): DISA STIGs for Windows and EMET.
Jon Zeolla (Human): Java Deployment Rule Sets.
Torsten Juul-Jensen (Human): GPO Preferences Update.
Rick Moffatt (Human): clarification of AppLocker deny rules.

Brett Slaughter (Human): MS14-025 update for GPO-assigned local user passwords.
Stefan Hazenbroek (Dutch): Group Managed Service Accounts for scheduled tasks.
Ginny Munroe (DeadlineDriven.com): lots of typo-snookies -- like this line!

Table of Contents

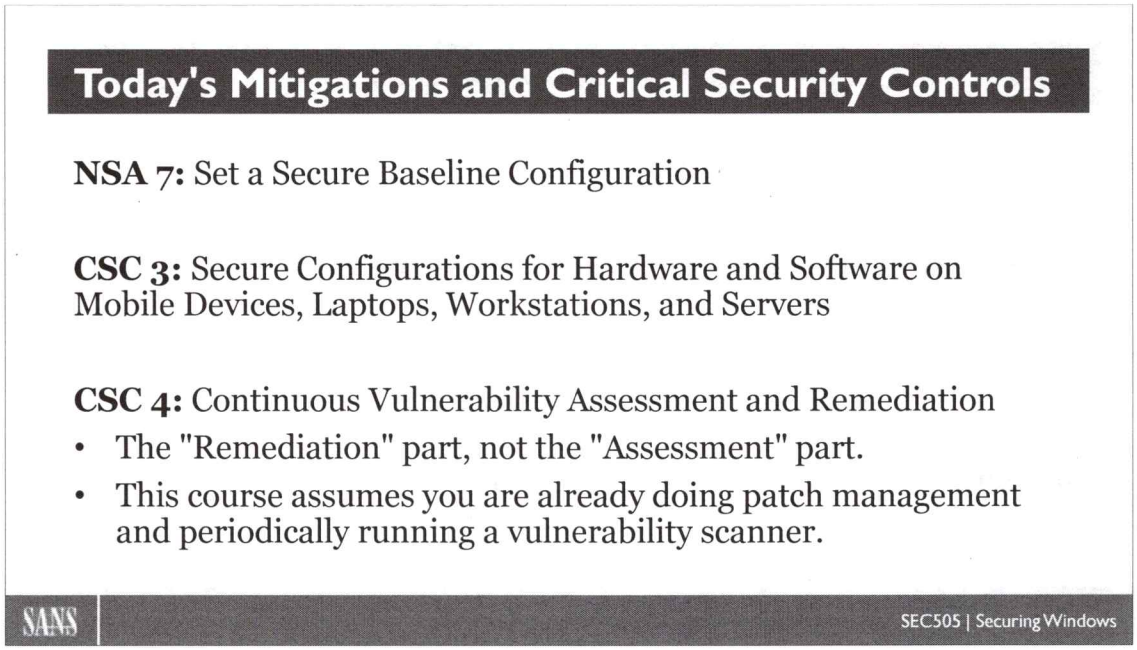


Today's Agenda

Today's course is about applying the Critical Security Controls to automate the enforcement of secure configurations for Windows clients and servers. Without automation, we cannot quickly deploy new servers, clients, applications or security configuration changes. Automation is also needed for scaling up to handle networks with thousands of Windows hosts. Our main tools are Group Policy, PowerShell, INF security templates, and security templates defined as Desired State Configuration (DSC) files. Security Operations (SecOps) teams are expected to use these types of tools to quickly adapt to the ever-changing threat landscape. Security needs to be baked in from the beginning of the design process, but it also needs to continuously reapplied to ensure compliance.

By the end of this course, you will be able to:

- Apply security templates to automate security configuration changes.
- Use Group Policy for scalability and configuration management.
- Use Server Manager and PowerShell to manage roles and features.
- Use PowerShell Desired State Configuration (DSC) for security.



Today's Mitigations and Critical Security Controls

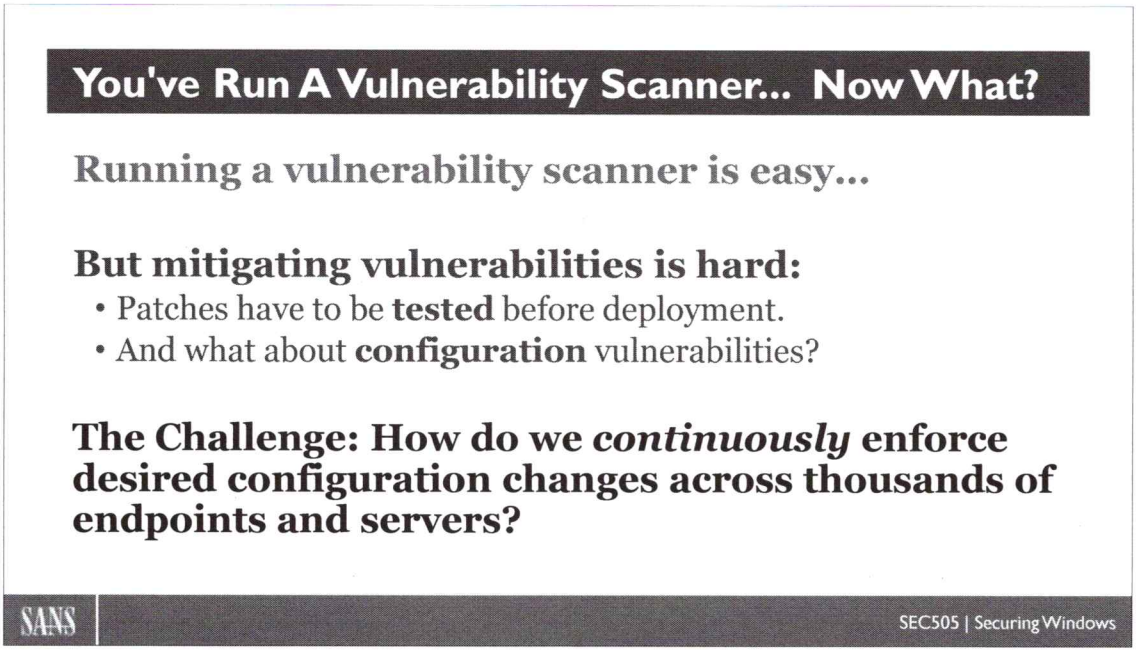
One mission of the National Security Agency (NSA) is to offer network security guidance through its Information Assurance Directorate (IAD). The NSA/IAD list of Top 10 Information Assurance Mitigation Strategies can be downloaded from the IAD web site (www.iad.gov).

The Critical Security Controls (CSC) project aims to describe the 20 most important tasks and activities for network security. You can download the latest version of the CSC from the web site of the Center for Internet Security (www.cisecurity.org).

Today's material is especially relevant for implementing the following NSA Top 10 Mitigations and CIS Critical Security Controls:

- NSA 7: Set a Secure Baseline Configuration
- CSC 3: Secure Configurations for Hardware and Software on Mobile Devices, Laptops, Workstations, and Servers
- CSC 4: Continuous Vulnerability Assessment and Remediation

For CSC 4 above, today's emphasis is on the "Remediation" part, not the "Assessment" part of vulnerability management. This course assumes you are already doing patch management and periodically running a vulnerability scanner.



You've Run A Vulnerability Scanner, Now What?

Running a vulnerability scanner is easy. Mitigating vulnerabilities is hard. Continuously enforcing these mitigations using automation tools and scripting is even more difficult. How can we make it easier? How can we do it for free using only built-in tools?

You Already Know How To Apply Patches

Most vulnerabilities are fixed by applying patches. This course does not cover patch management. You've probably been doing patch management for years already. But there is something hard about patch management: the testing of patches before release is hard to do. Testing must be automated somehow or else it's just too tedious and boring. This is another reason why this course emphasizes PowerShell so much. PowerShell is ideal for automating the patch testing process on Windows.

What About The Other Vulnerabilities Not Related To Patches?

But what about the other vulnerabilities, the ones not fixed by applying patches? These vulnerabilities are, by definition, mitigated by configuration changes (or they are unfixable). How do we automate configuration changes across thousands of endpoints and servers? Even more difficult, how do we continuously re-enforce these configuration changes to make sure they "stick", to make sure the desired configuration is automatically reapplied again if a host somehow drifts away from what we want?

Group Policy And PowerShell Are Designed For Automation

This course emphasizes Group Policy and PowerShell for continuous secure configuration enforcement because they are specifically designed for this purpose! Group Policy and PowerShell are built into Windows and Active Directory already. In

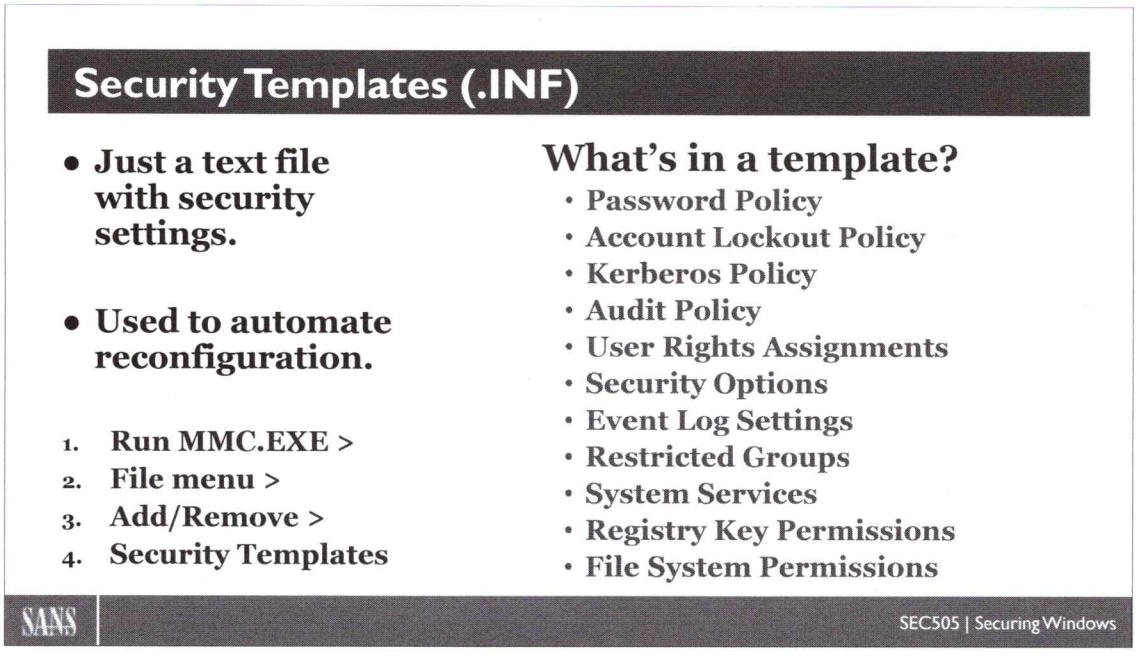
that sense, they are free, you don't have to purchase anything else. Don't let third-party security vendors sell you something you don't need.

Group Policy Templates And PowerShell Desired State Configuration

Group Policy Objects (GPOs) are like configuration templates which can be applied and re-applied automatically. PowerShell Desired State Configuration (DSC) is designed to do what the name implies: enforce a desired configuration state. A PowerShell DSC configuration is also a type of template that can be re-applied by hand or automatically.

Group Policy and PowerShell team up in other ways too; for example, we can use Group Policy to push out PowerShell scripts to hundreds of thousands of hosts and have the scripts executed hands-free, even if no one is logged on. Group Policy can also be used to manage scheduled jobs, and these jobs can run PowerShell scripts repeatedly.

In this course, we will see how to automate security configuration changes through Group Policy and PowerShell. One of the many things we can do with Group Policy is to apply .INF security templates to client devices and servers. What are .INF security templates?



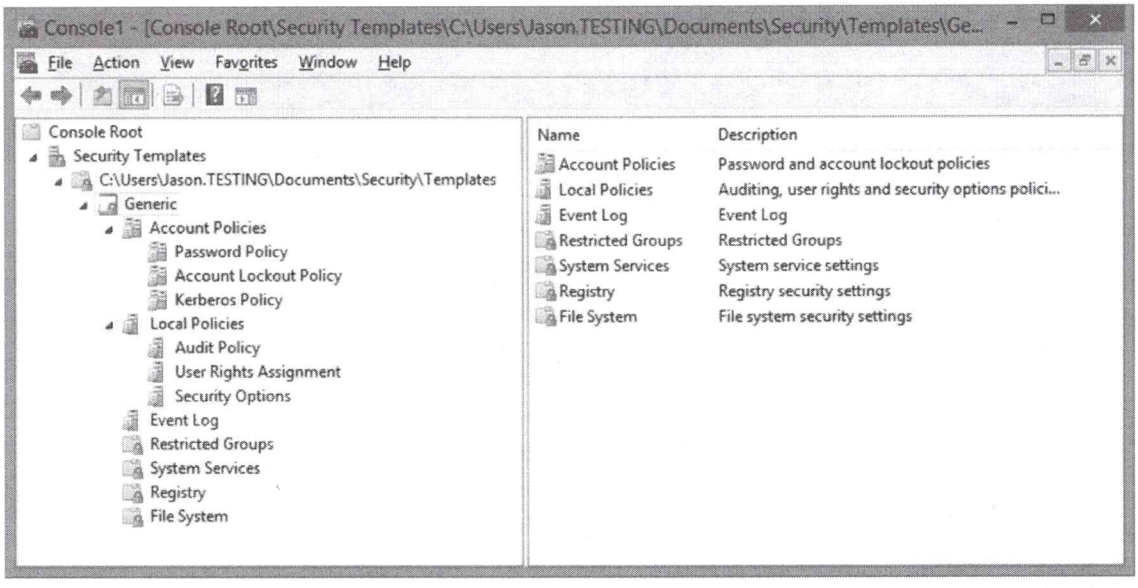
Security Templates (.INF)

The purpose of a security template is to store a large number of security settings in a single file. This file can then be "applied" to a machine using a special tool (discussed in next section) that will reconfigure the machine's settings in one shot to match the template. The template can be used on multiple boxes to quickly and easily enforce a consistent set of security options across them.

Templates are plain text and end with the .INF extension. By default, they are found in C:\Users\%UserName%\Documents\Security\Templates, but they can be kept anywhere.

Templates can be edited with any text editor, but there is an MMC snap-in named "Security Templates" which is designed to make creating and editing templates easier by representing the configuration options graphically.

> ***Try It Now!***
>
> In PowerShell, execute "mmc.exe" > pull down the File menu > Add/Remove Snap-In > select Security Templates > Add > OK. In the snap-in, expand down the subcontainers > right-click on the yellow templates folder > New Template > enter "Generic" as the name > OK. This creates a Generic.inf file.



A template stores the following security settings, which will all later be configured on a machine when the template is "applied" to it:

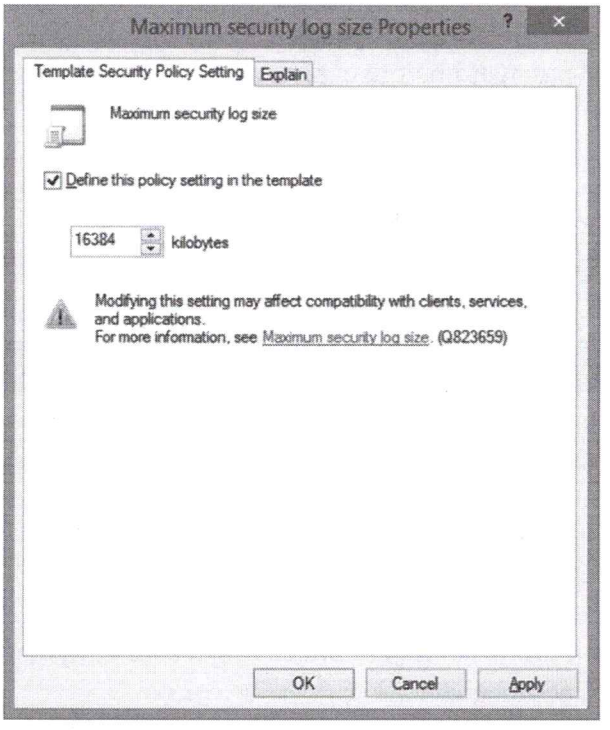
- Password policies.
- Account lockout policies.
- Kerberos policies.
- Audit policies.
- Custom user rights assignments.
- Security options, e.g., authentication protocols.
- Event log sizes and wrapping options.
- Custom memberships in important groups.
- Startup options and permissions on services.
- Registry key permissions and audit settings.
- NTFS permissions and audit settings.

Note: What these settings *should* be will be discussed later. Let's just discuss the mechanics of using templates for the time being.

To make a change to the Generic.inf template, open a container on the left-hand side, then double-click an icon on the right to bring up its dialog box. Don't forget to save the template after making the change.

Try It Now!

To make and save a change, double-click the Generic template > Event Log > double-click "Maximum Security Log Size" > check the "Define this policy setting in the template" checkbox > enter 16,384 kilobytes > OK > right-click on the Generic template again > Save.



Security Options

The Local Policies > Security Options section includes numerous registry options for well-known security changes. We will discuss many of these today:

- Additional restrictions for anonymous connections
- Allow server operators to schedule tasks (domain controllers only)
- Allow system to be shut down without having to log on
- Allowed to eject removable NTFS media
- Amount of idle time required before disconnecting session
- Audit the access of global system objects
- Audit use of Backup and Restore privilege
- Automatically log off SMB users when logon time expires
- Clear virtual memory pagefile when system shuts down
- Digitally sign client communication (always)
- Digitally sign client communication (when possible)
- Digitally sign server communication (always)
- Digitally sign server communication (when possible)
- Disable CTRL+ALT+DEL requirement for logon
- Do not display last user name in logon screen
- LAN Manager Authentication Level
- Message text for users attempting to log on (logon banner)
- Message title for users attempting to log on
- Number of previous logons to cache (in case domain controller is not available)
- Prevent system maintenance of computer account password

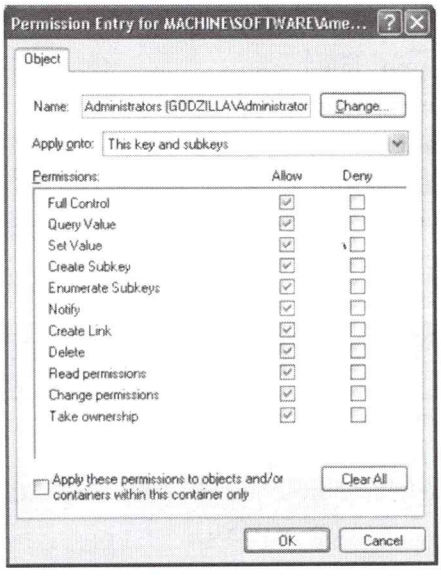
- Prevent users from installing printer drivers
- Prompt user to change password before expiration
- Recovery Console: Allow automatic administrative logon
- Recovery Console: Allow floppy copy and access to all drives and all folders
- Rename administrator account
- Rename guest account
- Restrict CD-ROM access to locally logged-on user only
- Restrict floppy access to locally logged-on user only
- Secure channel: Digitally encrypt or sign secure channel data (always)
- Secure channel: Digitally encrypt secure channel data (when possible)
- Secure channel: Digitally sign secure channel data (when possible)
- Secure channel: Require strong (Windows 2000 or later) session key
- Secure system partition (for RISC platforms only)
- Send unencrypted password to connect to third-party SMB servers
- Shut down system immediately if unable to log security audits
- Smart card removal behavior
- Strengthen default permissions of global system objects (e.g. Symbolic Links)
- Unsigned driver installation behavior
- Unsigned non-driver installation behavior

Registry Key Permissions

The Registry container is not for setting particular registry values, but for assigning permissions and audit settings to registry keys. You can edit these ACLs manually using REGEDIT.EXE.

Try It Now!

To add a registry key to the template, right-click on the Registry container > Add Key > browse to the desired key > OK > click Advanced > edit the Permissions and Auditing tabs as desired > OK > OK > choose how you want the ACLs to be inherited > OK.



NTFS File System Permissions

The File System container is for setting NTFS permissions and audit settings. Environmental variables are automatically entered when defining folder paths so that the template will be as portable as possible. You can add hundreds of folders and individual files to the template, if desired, and customize every single one of them.

Try It Now!

To add a file *or folder*, right-click on the File System container > Add File > browse to select the file *or folder* whose ACLs you wish to edit > OK > click Advanced > edit the Permissions and Auditing tabs as desired > OK > OK > choose how you want the ACLs to be inherited > OK. You can always double-click on the yellow file/folder icon you added to edit its settings, or right-click on it to delete it.

Service Startup and Permissions

You can also set a service's startup status to automatic, manual or disabled. It is also possible to change the permissions and audit settings on the services themselves, such as perhaps to delegate authority over the World Wide Web Publishing service to your organization's WebMasters group.

> **Warning!** When using a security template to set audit settings on a service, you will overwrite the existing audit settings and the existing permissions. Since the permissions shown in the template's dialog box are not necessarily the factory defaults, you must take care not to accidentally change the permissions when you only intend to change the audit settings. When editing NTFS or registry key permissions, the dialog boxes give you more control over the inheritance options.

What About The Other Settings In The Template?

The other settings and containers in the template will be discussed throughout the rest of the week. The previous examples were just to have something concrete to discuss.

INF Template Syntax Reference (If You Edit The Template By Hand)

When would you ever have to edit an INF template with Notepad by hand? This would be necessary, for example, in order to set arbitrary registry values not exposed in the SCA graphical tool.

It's also wise to examine any template obtained from a source not trusted 100%. There are also some tricks with template editing that can save you a fair amount of time and repetitive mouse-clicking.

A security template is divided into sections with a section name in square brackets, e.g., [System Access] demarcates everything that follows it as part of the System Access section until you get to another section name in square brackets.

Many of the values are fairly self-explanatory and won't be discussed here. For example, a decimal one usually means "Enabled", a zero, "Disabled", and the meaning of these lines is pretty obvious when compared against their GUI dialog boxes in the Security Templates snap-in:

```
[System Access]
MinimumPasswordAge = 1
MaximumPasswordAge = 90
MinimumPasswordLength = 8
```

On the other hand, some of the sections contain encoded information that's hard to understand. Below you will find tips to help edit the not-too-straightforward sections of the template. If you have a computer with you now, open one of the larger templates in Notepad.

[Event Audit] Section

The [Event Audit] section configures audit policy. The option names are pretty straightforward, but the code numbers have the following meanings:

0 = Define the policy, but audit neither Success nor Failed events.
1 = Audit only Success events.
2 = Audit only Failed events.
3 = Log both Success and Failed events.

For example:

```
[Event Audit]
AuditSystemEvents = 3
AuditLogonEvents = 3
AuditObjectAccess = 2
AuditPrivilegeUse = 2
AuditPolicyChange = 2
AuditAccountManage = 3
AuditAccountLogon = 3
```

[Service General Setting] Section

The [Service General Setting] section holds the startup and ACL settings in the System Services container in the template. But how can you add a service to the list which is not already there when you edit the template with the Security Templates snap-in? There are two options.

One, you can edit the template on a machine which *does* have the desired service installed, then it will appear in the template even when you copy it to other systems. But this is a bit inconvenient. Two, configure some other service just the way you want the desired service to be configured, then edit the template to change the name of the service configured.

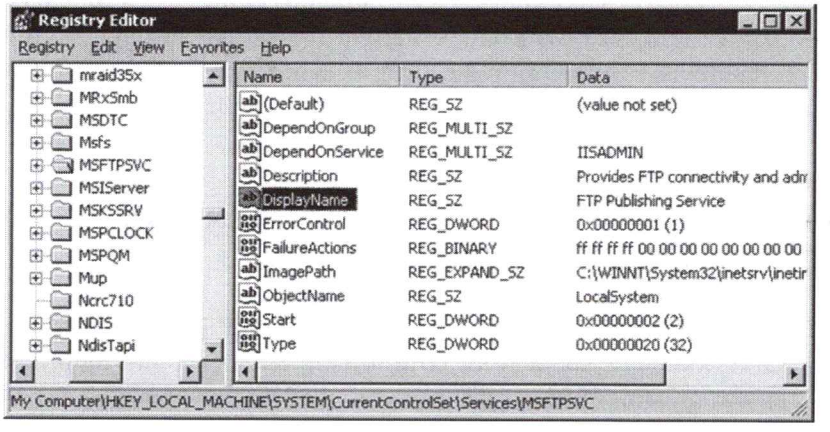
For example, I configure the Browser service in the template to be disabled by default and I grant the Everyone group the "Start, Stop and Pause" permission on it. The template will look like this (the Browser... line is wrapped for three lines):

```
[Service General Setting]
Browser,4,"D:AR(A;;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;BA)(A;;RPWPDTRC;;;
WD)(A;;CCLCSWLOCRRC;;;IU)(A;;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;SY)S:(A
U;FA;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;WD)"
```

What's all that garbage after "Browser,..."? *Who cares!* We know that it contains the ACLs we want so we don't touch it. (You can confirm this afterwards in the snap-in after making the changes below.)

Now, change the name of the service to the desired service, e.g., if you want to disable the IIS FTP Publishing Service, then change "Browser" to "MSFTPSVC". How do we know that that's the correct string?

The name of the service in the template must match the name of the registry key under HKLM\System\CurrentControlSet\Services\ for that service. In the screenshot below we see the MSFTPSVC key for the FTP Publishing Service. We know it is the correct key because of the DisplayName value in that key which we found by using the Find feature in REGEDIT.EXE.



Tip: Don't forget that you can access the registry on remote systems with REGEDIT.EXE to get the right service key names.

Tip: These are also the service names --the names of the keys-- which always work when using NET.EXE to start or stop services from the command line.

The edited lines for the FTP Publishing Service will look like this:

```
[Service General Setting]
MSFTPSVC,4,"D:AR(A;;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;BA)(A;;RPWPDTR
C;;;WD)(A;;CCLCSWLOCRRC;;;IU)(A;;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;SY)
S:(AU;FA;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;WD)"
```

Now that we have this trick down, we can simply copy-and-paste the ACL portion into multiple service lines in the template. This saves the time and endless mouseclicks of doing it in the GUI for each service we want identically configured. It is also more reliable because all the mouseclicking is prone to errors.

[Registry Keys] And [File Security] Sections

The same trick used to copy ACLs in the [Service General Setting] section above can be used for registry key and NTFS settings as well. It's not worth the effort to decipher exactly what the encoded ACLs mean or their syntax. Here are two examples, and notice that the key or folder/file to be modified is always the first parameter:

```
[Registry Keys]
"MACHINE\SYSTEM\",2,"D:PAR(A;CI;KA;;;BA)(A;CIIO;KA;;;CO)(A;CI;KA;;;SY)"

[File Security]
"%SystemRoot%\system32\secedit.exe",2,"D:PAR(A;OICI;FA;;;BA)(A;OICI;FA;;;SY)"
```

Instead of trying to decipher the ACL syntax, configure one key or file exactly the way you want it, then copy-and-paste the ACL information to the other keys or files. You can confirm that it worked by saving the template, refreshing it in the Security Templates snap-in, and seeing what settings are configured in the GUI.

> **Tip:** In the [File Security] section, use environmental variables as much as possible to make the template work on as many different partition configurations as possible. Execute "set" in a command-prompt window to see the live variables on your machine, e.g., %SystemRoot%, %WinDir%, %SystemDrive%, etc.

[Registry Values] Section

There is a [Registry Keys] section, but that is for configuring registry key ACLs, not values. Look under [Registry Values] to edit registry data.

You can modify any registry value you wish when you apply your template, but you must edit the template by hand first. The snap-in for editing templates does not include a feature for this, hence, you must use a text editor.

> **Note**: If you want to add your own items to the Local Policies > Security Options container, you'll need to edit the SCEREGVL.INF template in the %WinDir%\Inf folder. Afterwards, execute "REGSVR32.EXE scecli.dll" at the Run line.

In the [Registry Values] section, the full path to the registry value is given, then two numbers separated by a comma. The first number determines the value type, the second, the value data to be set (KB214752).

Registry value types are mapped to the following code numbers:

1 = REG_SZ
2 = REG_EXPAND_SZ
3 = REG_BINARY
4 = REG_DWORD
5 = REG_DWORD_LITTLE_ENDIAN
6 = REG_LINK
7 = REG_MULTI_SZ
8 = REG_RESOURCE_LIST
9 = REG_FULL_RESOURCE_DESCRIPTOR

For example, the following would set the REG_DWORD SynAttackProtect value to 2. REG_DWORD values have a type code of 4, and the value is 2; hence, the line ends with "=4,2".

[Registry Values]
MACHINE\System\CurrentControlSet\Services\Tcpip\Parameters\SynAttackProtect=**4,2**

Notice that the pathname does not start with "HKEY_LOCAL_MACHINE" or "HKLM". In templates, "MACHINE" is taken as a keyword for HKEY_LOCAL_MACHINE.

Not only must you edit the template manually to set these registry values, but the values do not show up in the Security Templates snap-in under the \Local Policies\Security Options\ container in the GUI version of the template (unlike the values set by Microsoft, which do appear in the snap-in). The only way to know that these values are being changed is to examine the file, hence, always check templates from sources not trusted 100% before applying them.

> **Tip:** Don't get too attached to the exact ordering of the sections in your template. The SCA reorders the sections as it sees fit and ignores associated comments. In general, put all of your comments (semicolons) at the very top of the template.

Well-Known SIDs In [Privilege Rights] And Elsewhere
A "well-known" Security ID number (SID) is a SID that is always valid no matter what the name or instance of the domain/computer is. Scripts and templates which refer to these well-known SIDs will be portable across different domains/computers.

The [Privilege Rights] section is for the assignment of custom user rights. Here, and in other sections, you will commonly see the well-known SIDs of various users and groups. For example, the following rights are assigned only to the local Administrators group because that group has a well-known SID of "S-1-5-32-544":

```
[Privilege Rights]
seloaddriverprivilege = *S-1-5-32-544
seprofilesingleprocessprivilege = *S-1-5-32-544
seremoteshutdownprivilege = *S-1-5-32-544
sesystemtimeprivilege = *S-1-5-32-544
setakeownershipprivilege = *S-1-5-32-544
```

List of Well-Known SIDs

Here is a list of the well-known SIDs as a reference (KB243330):

S-1-0-0	Null group, i.e., an empty or unknown group.
S-1-1-0	Everyone.
S-1-2-0	Locally logged-on users.
S-1-3-0	Creator Owner group.
S-1-5-1	Dialup Users group.
S-1-5-2	Network Users group.
S-1-5-3	Batch Users group.
S-1-5-4	Interactive Users group.
S-1-5-6	Service Accounts group.
S-1-5-7	Anonymous Logon account.
S-1-5-9	Enterprise Controllers/Domain Controllers group.
S-1-5-10	Self.
S-1-5-11	Authenticated Users group.
S-1-5-12	Restricted code.
S-1-5-13	Remote Desktop Services Users.
S-1-5-18	Local System.
S-1-5-32-544	Local Administrators group.
S-1-5-32-545	Local Users group.
S-1-5-32-546	Local Guests group.
S-1-5-32-547	Local Power Users group.
S-1-5-32-548	Domain local Account Operators group.
S-1-5-32-549	Domain local Server Operators group.
S-1-5-32-550	Domain global Print Operators group.
S-1-5-32-551	Domain global Backup Operators group.

There are a few well-known SIDs for special accounts and groups. The *XXX* portion of the SID will be different for each installation. Nonetheless, it is useful to be able to recognize these numbers in logs, scripts, templates, source code, etc. (They all start with "Domain" to indicate that they do not refer to any local accounts or groups by the same name.)

S-1-5-*XXX*-500	Domain Administrator account.
S-1-5-*XXX*-501	Domain Guest account.
S-1-5-*XXX*-502	Domain krbtgt account.
S-1-5-*XXX*-512	Domain Domain Admins group.
S-1-5-*XXX*-513	Domain Domain Users group.
S-1-5-*XXX*-514	Domain Domain Guests group.

S-1-5-*XXX*-515	Domain Domain Computers group.
S-1-5-*XXX*-516	Domain Domain Controllers group.
S-1-5-*XXX*-517	Domain Cert Publishers group.
S-1-5-*XXX*-518	Domain Schema Admins group.
S-1-5-*XXX*-519	Domain Enterprise Admins group.
S-1-5-*XXX*-520	Domain Group Policy Creators Owners group.
S-1-5-*XXX*-553	Domain RAS and IAS Servers group.

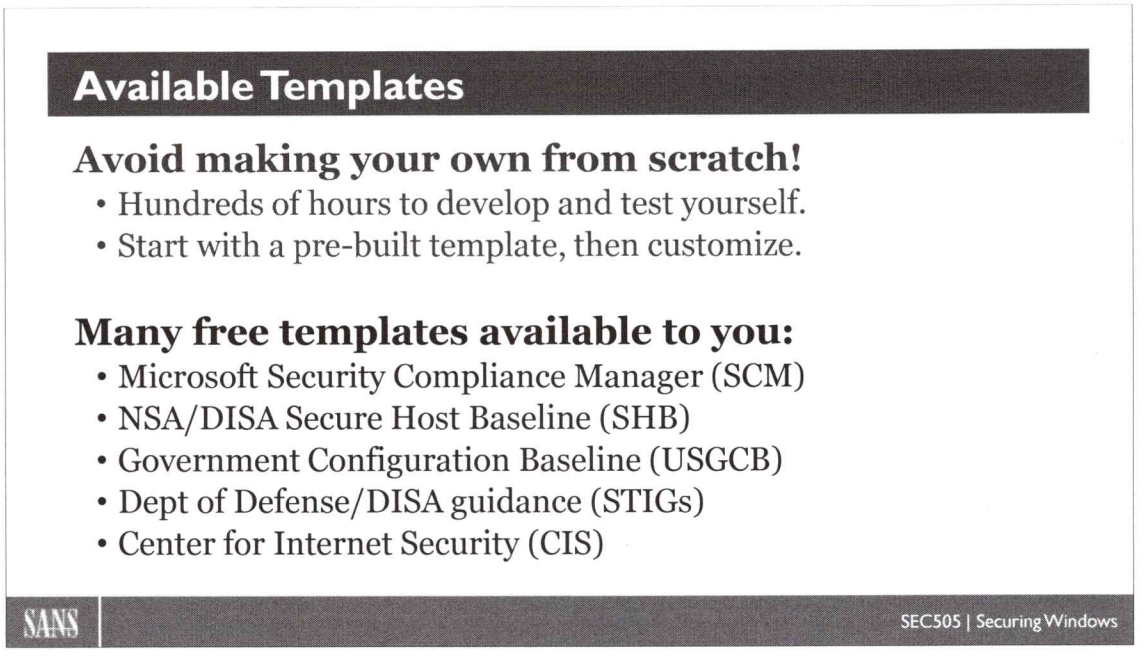
Use Incremental Templates To Your Advantage

A template does not have to include all the possible sections. The only text a template *must* have is the following, all the other sections are optional:

```
[Unicode]
Unicode=yes
[Version]
signature="$CHICAGO$"
Revision=1
```

Because templates can be imported incrementally into a SCA database, consider separating out your templates based on the section(s) contained in them. This way you can mix and match your custom templates to build up just the right settings in the database you want.

For example, there is probably a core set of NTFS permissions that you want configured on every workstation and server. Put these ACLs into one template. Then, for each different type of machine, create additional templates to augment the core NTFS permissions. If you need to change your core permissions, you only need to do it once and not in every platform-specific template you have.



Available Templates

Templates are condensed knowledge and expertise, ready for automated (re)deployment. You don't have to create your own templates from scratch. Microsoft, DISA, NIST, NSA, CIS, and other players in the Windows security arena have customized templates which are free to download.

And it is highly recommended that you begin with someone else's templates instead of starting from scratch. The reason for this is that security is bad for usability. In general, the more security options you configure, the more applications you are likely to break. Templates from Microsoft, NIST and others have been debugged and tested in order to improve security as much as possible while breaking as little as possible.

Consider, out of the thousands of registry key permissions and NTFS permissions which could be changed, which ones *should* be changed? Which changes will break your favorite applications? Templates represent the condensed knowledge of experts who have suffered for dozens or hundreds of hours to fine-tune them. Hence, start with a template created by a group you trust, then customize that template to meet your needs.

What Security Templates Are Available From Microsoft?

Microsoft has a set of security templates and best practices for various applications and versions of Windows. These templates provide an excellent starting point for your internal testing. They should not be applied without compatibility testing first.

To download the templates and their associated documentation, download the free Microsoft Security Compliance Manager (SCM) tool:

- **Microsoft Security Compliance Manager**
  http://technet.microsoft.com/en-us/library/cc677002.aspx

Most of the security settings in the baseline templates in the SCM tool can be read on Microsoft's web site too, just do a search on the terms "security guide" or "security baseline" along with your chosen operating system name. Microsoft's security guidance blog is a good place to do the search (https://blogs.technet.microsoft.com/secguide/).

What Templates Are Available From NIST And The US Government?

[NSA DISA STIG & SHB] United States Department of Defense (DoD) Directive 8500.1 requires that all DoD computers be configured using security configuration guidelines developed by the Defense Information Systems Agency (DISA) and the National Security Agency (NSA). These guidelines come in the form of Security Technical Implementation Guides (STIGs) which include security templates, checklists, scripts, SCAP XML specifications, and other documents. In particular, this includes a Secure Host Baseline (SHB) for Windows 10 and later systems.

These DISA STIGs and the SHB are available to the public:

- http://iase.disa.mil/stigs/
- https://github.com/iadgov/Secure-Host-Baseline

[USGCB] The US Department of Defense and NIST have updated the older FDCC standards and renamed the project to the "United States Government Configuration Baseline (USGCB)", though this is being replaced by the Secure Host Baseline (SHB).

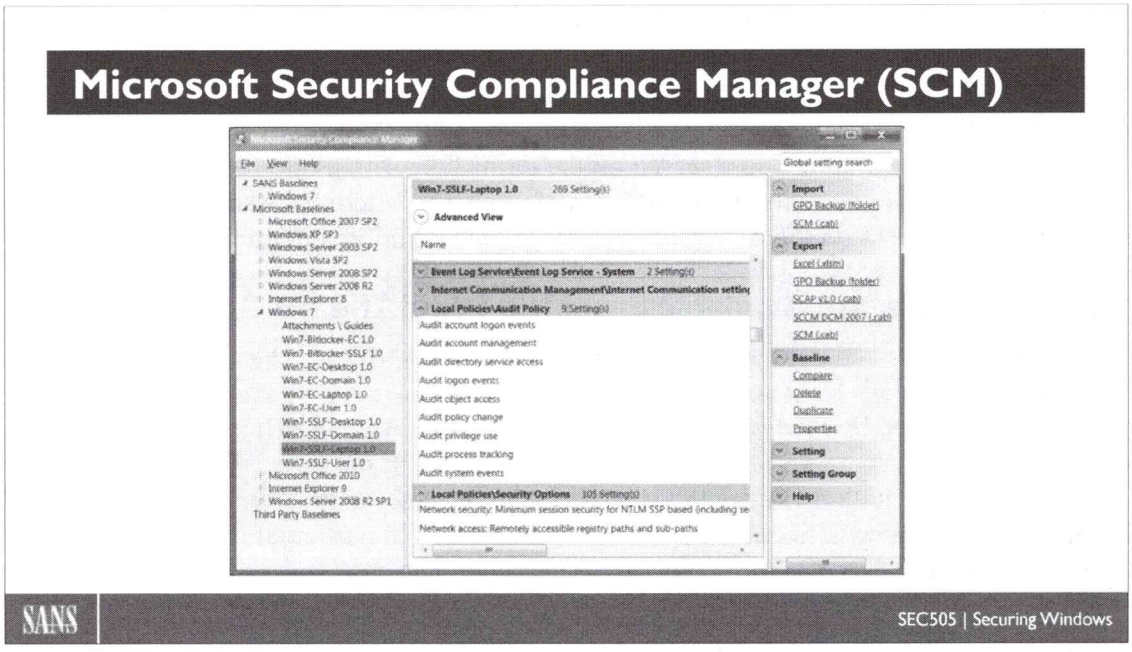
- http://usgcb.nist.gov

[CIS] The Center for Internet Security (CIS) not only has security templates available, but also configuration guides and audit tools to go with them. Government representatives participated in the creation of many of the CIS templates. Get the latest guidance and tools from the CIS site:

- http://www.cisecurity.org

[FDCC] The federal NIST version of Microsoft's templates were incorporated into the Federal Desktop Core Configuration (FDCC) standards, but FDCC was replaced by USGCB and SHB above:

- http://nvd.nist.gov

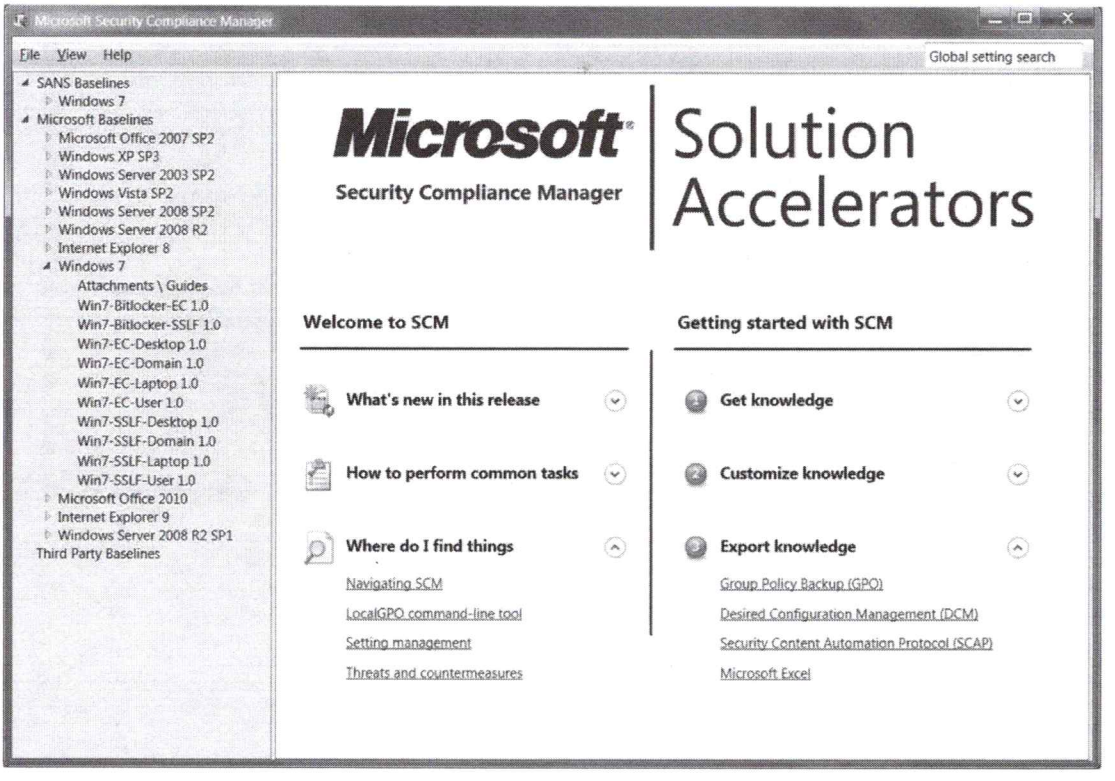


Microsoft Security Compliance Manager (SCM)

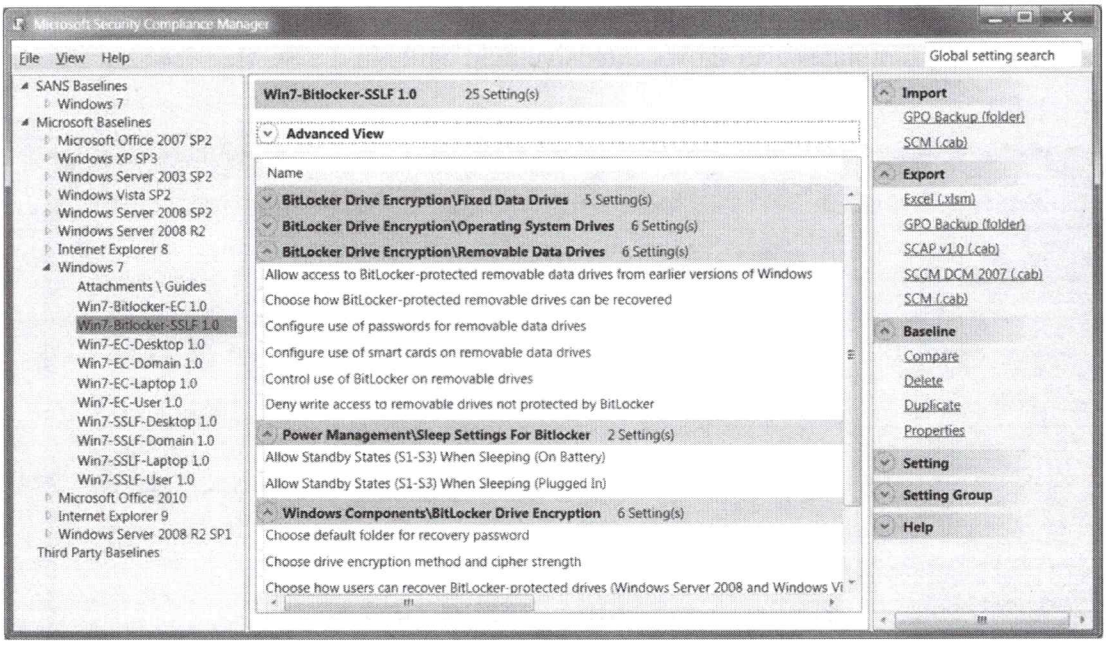
The Microsoft Security Compliance Manager (SCM) is more than just a set of templates (http://technet.microsoft.com/en-us/library/cc677002.aspx). The SCM is a tool which performs a variety of template management tasks, such as:

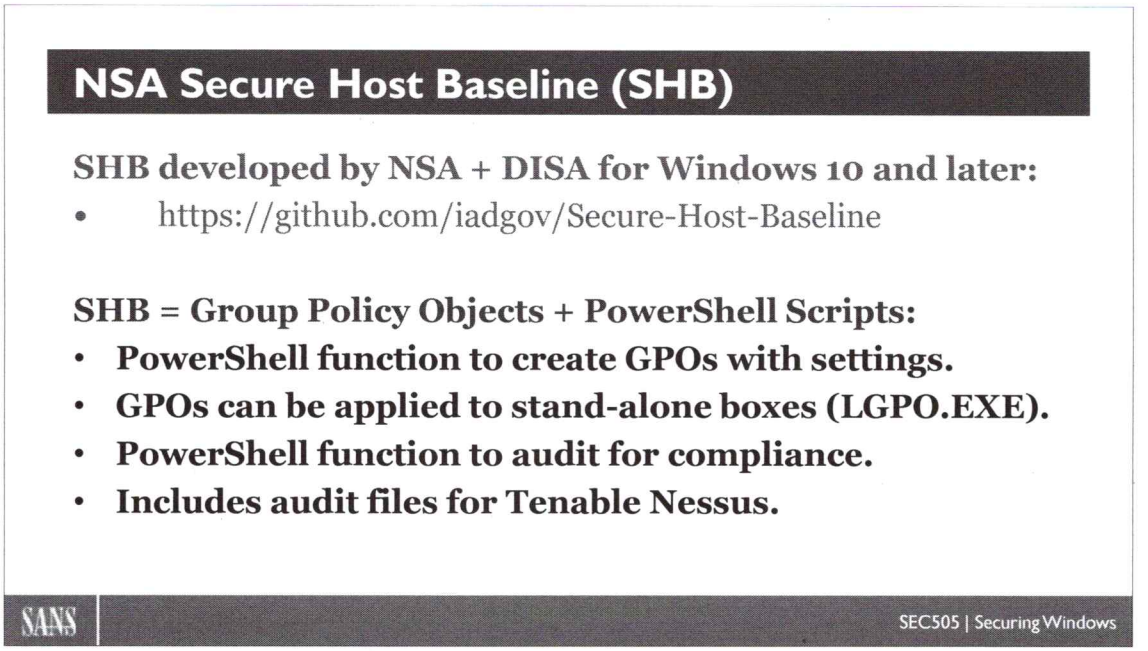
- Manage security settings beyond just those found in INF security templates; the SCM also manages ADM/ADMX registry value changes that can be made through a Group Policy Object (GPO).
- Import from a GPO which has been backed up to disk, including the local GPO.
- Export security settings to a spreadsheet, GPO, System Center Configuration Manager (SCCM) Desired Configuration Management (DCM) file, or Security Content Automation Protocol (SCAP) file.
- Includes a command-line script (localgpo.wsf) to export the local GPO, import a GPO to override the local GPO, and restore the local GPO to its factory defaults (but does not restore NTFS and registry permissions to their factory defaults).
- Manage version histories for different templates in a SQL Server database, including automatic updates to the official templates as they are published.
- Report the differences between templates to identify changes.
- Merge templates together, with control over conflict resolution.

- Maintains documentation for most built-in templates in Word document format. Browsing templates also shows extensive help while in the SCM.



After selecting a template, notice on the right-hand side the options to import, export, compare and duplicate. You must duplicate a read-only template before editing it.





NSA Secure Host Baseline (SHB)

In November 2015, the US Department of Defense (DoD) ordered all DoD agencies and commands to begin rapidly upgrading to Windows 10, preferably within one year. A mass upgrade of this scale was unprecedented in the history of the DoD and reflects the urgency of the cyber threat to US government and military information systems.

In support of this mass upgrade, the National Security Agency (NSA) and the Defense Information Systems Agency (DISA) were directed to develop a Secure Host Baseline (SHB) for Windows 10 and later operating systems to maximize security and consistency across all DoD commands.

Most of the Secure Host Baseline (SHB) is available to the public:

- https://github.com/iadgov/Secure-Host-Baseline
- http://iase.disa.mil

SHB = Group Policy Objects + PowerShell Scripts

The SHB is mostly composed of Group Policy Objects and PowerShell scripts. It is not an accident that this SANS course (SEC505) emphasizes PowerShell and Group Policy. The SHB is designed for scalable, repeatable, updateable security configuration enforcement, and so is this course.

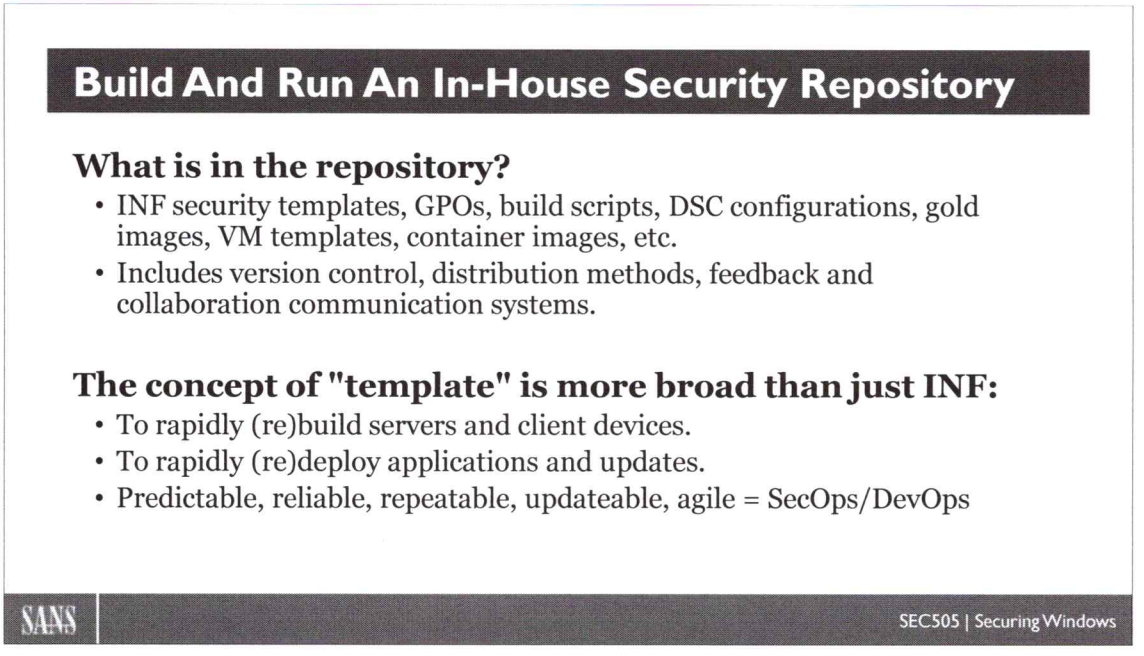
The SHB hardens settings for Adobe Reader, AppLocker, BitLocker, Google Chrome, Microsoft EMET, Windows Firewall, Microsoft Office, and other applications.

The SHB includes a PowerShell function, Invoke-ApplySecureHostBaseline, to create GPOs which contain the SHB configuration settings (see the GitHub link above for instructions on how to download and apply the SHB).

The SHB is also compatible with the free LGPO.EXE tool from Microsoft. This means an SHB GPO can be created on the local hard drive and directly applied with LGPO.EXE, even on stand-alone computers inside an air gap.

Compliance Checking

For checking compliance, the SHB includes audit definition files for Tenable Nessus Professional (www.tenable.com), which is a commercial vulnerability scanning tool. But the SHB also includes a PowerShell function (Test-Compliance) which can scan a single system with these same audit files, but without the necessity of purchasing Nessus. Through PowerShell remoting, the function could be run on many remote systems.



Build And Run An In-House Security Repository

Over time, you'll develop an in-house repository of security templates, Group Policy Objects (GPOs), baseline configurations for Desire State Configuration (DSC), PowerShell build scripts, template VMs for Virtual Desktop Infrastructure (VDI), "gold images" for hard drives, container images, firewall rule sets, AppLocker XML policies, etc. The repository will contain much more than just INF security template files.

Indeed, for this course, the concept of "template" is much more broad. The repository contains *whatever* is necessary to automate the rapid (re)deployment and (re)audit of servers, applications and client devices, including their security settings.

Centralized Version Control and Distribution

Just like for source code, we need version control and a centralized distribution point. A distribution point might be an SMB shared folder, a SharePoint collection, a third-party content management system, or another type of server. You might have a separate distribution point for each major type of item, such VMs or GPOs, which is fine, as long as there is a single, authoritative source for each type of item in the organization. For version control, it could be as simple as having a numbered subdirectory in a shared folder for each version. It doesn't have to be overly fancy or expensive --you don't need to reinvent GitHub inside your LAN-- it just needs to make sense and be easy to use.

Working with your help desk, developers, DBAs, and other IT staff, agree on 1) how many of these repositories there should be, 2) where they will be located, 3) the distribution methods, such as SMB shared folders or SharePoint sites, 4) how versions will be numbered, incremented and controlled, and 5) how feedback, bug submission, discussions, change requests and other communications will be facilitated among the

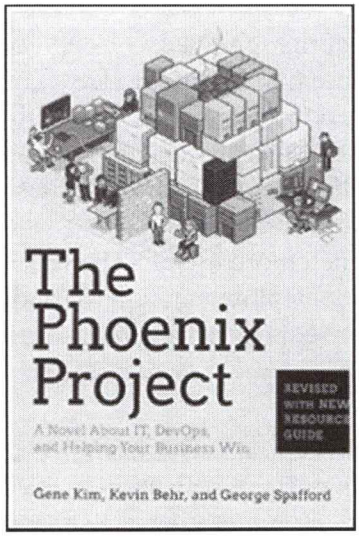
various teams. For example, you will have at least two repositories: one for testing and another for publishing ready-to-go files into production.

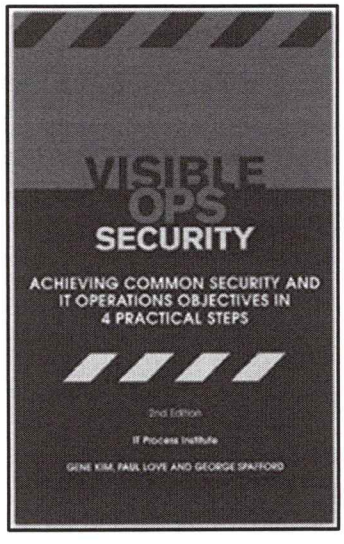
Team Communication and Feedback

A very important part of managing the repository is facilitating communication and collaboration among the various IT groups, such as auditors, help desk, compliance, developers, perimeter security, forensics, legal, and so on. This can be done through internal forums, blogs, news feeds, feedback/bug ticketing systems, chatrooms, mailing lists, and face-to-face meetings. Continual feedback and suggestions would be a part of a larger organizational culture change which promotes this type of teamwork and collaboration in a healthy way which avoids blame.

Recommended Reading

Do these recommendations sound familiar? They are part of the larger "DevOps" trend in IT, made famous by books like *The Phoenix Project* (Kim, Behr) and *Visible Ops Security* (Kim, Love, Spafford). Both of these books are definitely on the SEC505 Recommended Reading list. In fact, this course presumes (or at least hopes) that attendees are working for organizations that follow the guidance and lessons of books like these.

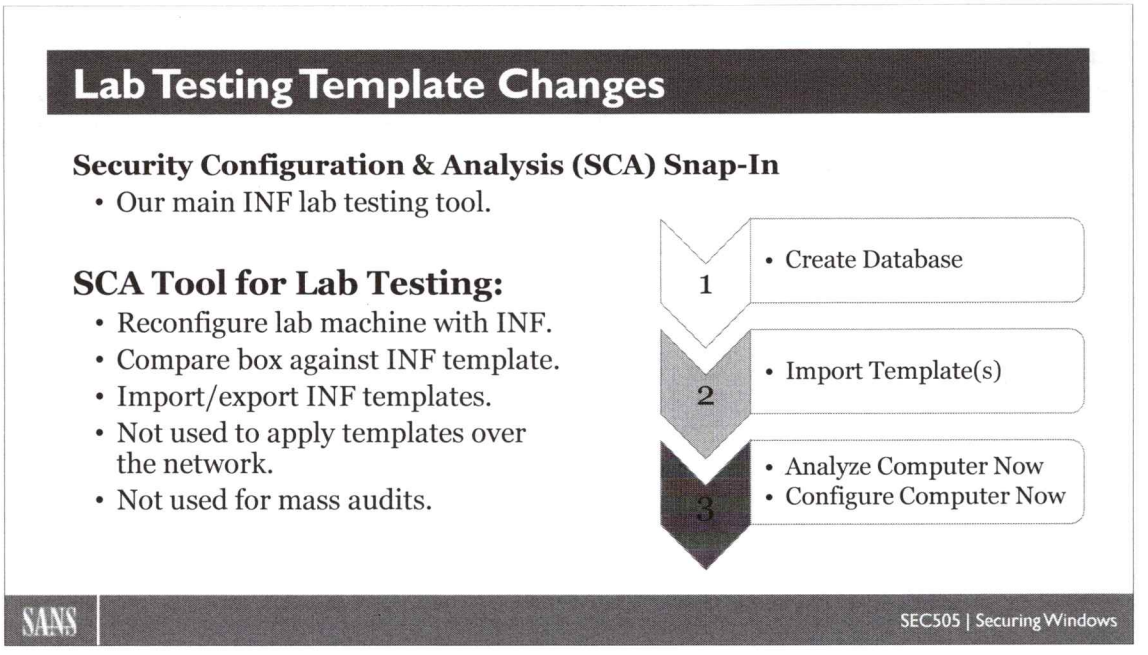




Why is this so important? Many organizations try to cope with security problems in isolation, when really the fundamental problem is a lack of good change management.

Organizations that can't handle a high volume of IT changes per year often resort to whack-a-mole "incident response" as their sole security practice: not because whack-a-mole is so effective at reducing harm or risk, but because that's the best they can do. Hackers, malware, threats and vulnerabilities are constantly changing, so organizations that have poor change management are always behind the curve, always rushing from one emergency to the next, and never getting ahead of the game.

Many of the recommendations in this course will seem "impractical" or even "impossible" to organizations with poor change management. But these are also the same organizations with the highest rates of malware infection or hacker intrusion. This is not an accidental coincidence. If this sounds like your organization (join the club!) then consider picking up copies of the two books above. Start with *The Phoenix Project*, it's written like a novel and is a good fun read (give it to your boss afterwards).



Lab Testing Template Changes

The Security Configuration and Analysis (SCA) snap-in is used to:

- Reconfigure a system to match the settings in a security template.
- Compare the settings on a system against a security template (auditing).
- Create a "database" of security template information.
- Import/export security template settings from these "databases".

Install the SCA in the same MMC console as the Security Templates snap-in, using the same procedure described above.

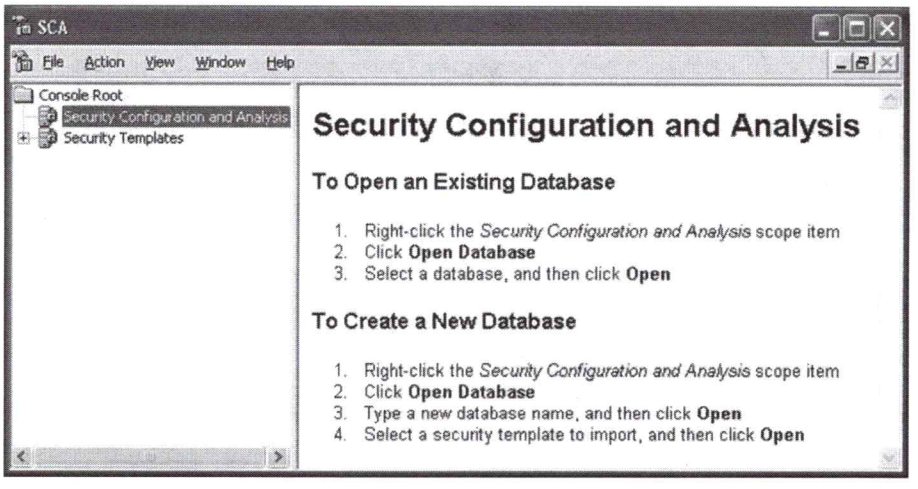
Before you can apply a template to a system, or compare a system against a template, you must store the settings in a SCA "database".

SCA Databases

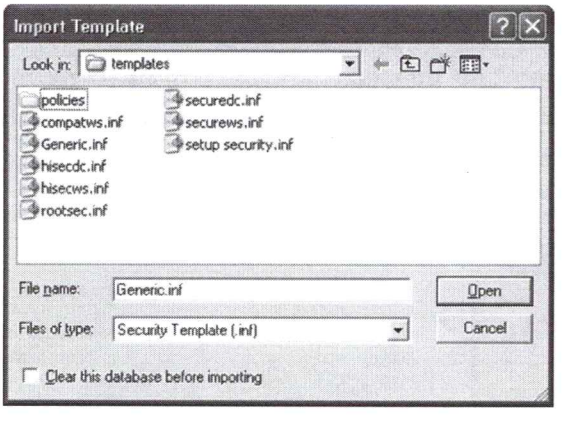
You do not need to install SQL Server, Microsoft Access, or any other database software to use SCA databases. An SCA database is much more like a semi-permanent temp file intended to store settings from one or more templates. The important files are the security templates, which are carefully edited and used on multiple machines; SCA databases, on the other hand, are disposable and machine-specific. Databases are not edited directly. Templates are edited and then imported into databases.

The idea is that you will import the settings from one *or more* security templates into a single SCA database. It is the database, not the template(s), which will be used by the SCA tool to reconfigure the machine or to perform an audit.

When you first click on the SCA snap-in, instructions for creating/opening databases appears on the right-hand side of the console. Follow the instructions to create a new database named "testing".



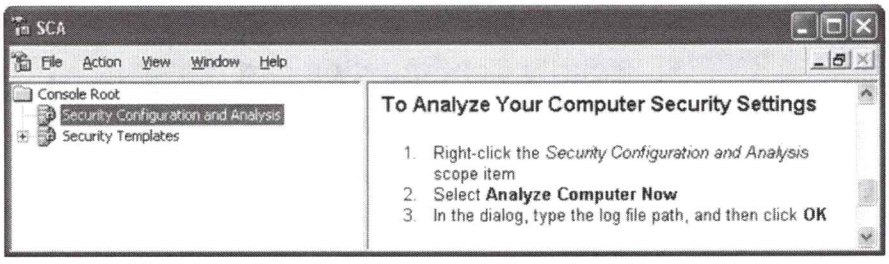
When creating a database, you will be prompted to import the settings from a security template. Select the Generic.inf template created earlier (or go back and create that now with the Security Templates snap-in) and click Open.



Again, the instructions for reconfiguring or auditing your machine will appear on the right-hand side. But before doing this, right-click on the SCA snap-in again and select Import Template. You will get the same dialog box as before. Notice the checkbox at the bottom: "Clear this database before importing". Importantly, you can *layer* the settings from multiple templates into one database if this box is left unchecked during each import. Whenever there is a conflict of settings, the template(s) applied later override the settings from the template(s) imported earlier (so the order of import is important). Many templates are designed to be layered with other templates; the fancy phrase for such templates is "incremental templates" (KB234926). Most of the templates from Microsoft are incremental templates.

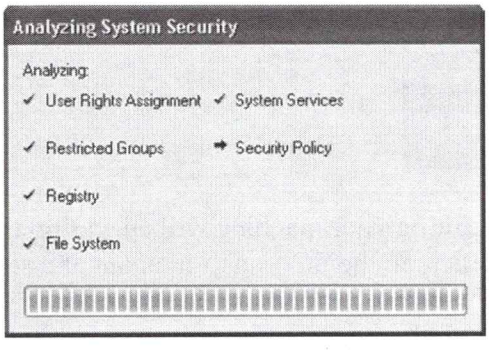
Analyze Your Computer (Audit Against The Template)

With an active SCA database assigned, you can compare your system against the settings in the database. This will not make any changes to your machine. The instructions for doing the audit appear on the right-hand side of the console. Follow them now.

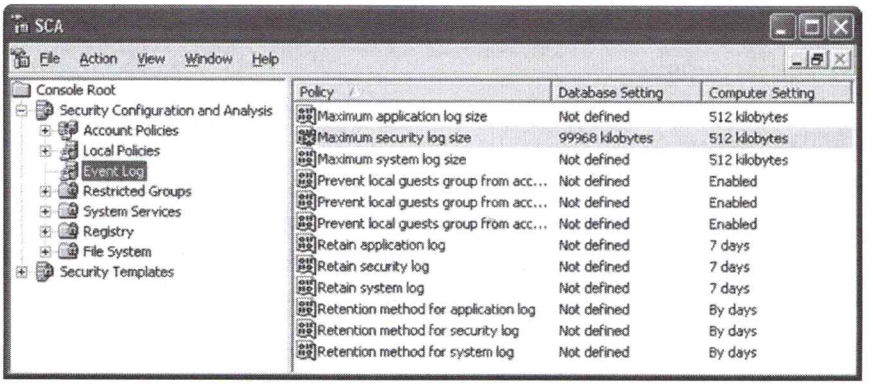


You will be prompted for the path to the logfile that will be created. Note what the path is because it will be different depending on who you are logged on as.

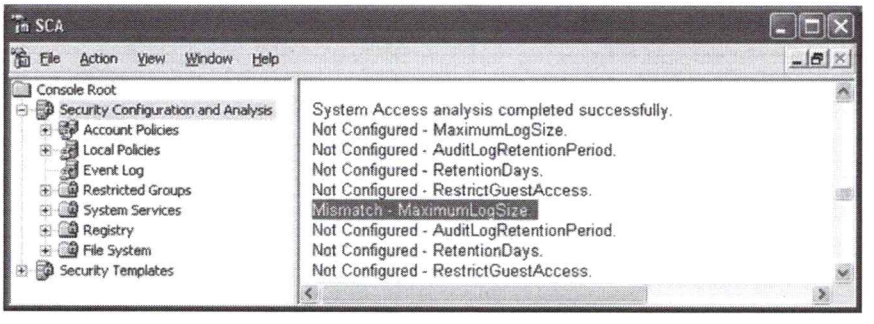
The SCA tool will then perform the audit. This may take anywhere from 10 seconds to 30 minutes, depending on the size of the database and the speed of the machine. (NTFS and registry key ACLs take the longest time to process of all the settings.) The Generic.inf template will take about 4 seconds.



Once completed, double-click the SCA snap-in to reveal its subcontainers. The containers are identical to the containers within the template. Navigate to the settings for the Event Logs. Each computer setting that does not match the setting in the database is marked with a red X. Matching items have a green checkmark. Items not defined in the database are plain blue icons. In the screenshot below, the "Maximum Security Log Size" specified in the database is 99,968 kilobytes, but the computer was configured with only a 512 kilobyte security log.



Now, right-click on the SCA snap-in and select View Log File if it is not already checked. Click once on the SCA to select it. The right-hand side of the console now shows the log that was produced. Scroll down about two-thirds of the way to the bottom and you'll see this line: "Mismatch - MaximumLogSize". Using FINDSTR.EXE, GREP.EXE or a script, you would search for all lines that said "mismatch" when automating your audits (see the SECEDIT.EXE tool below also). But notice that there are other "MaximumLogSize" entries as well. Unfortunately, the log format is not always explicit and sometimes the relevant information is not all on one line, which makes extracting the necessary data a chore.



Configure Computer (Apply The Template)

With the database loaded with the settings you want to enforce, simply right-click the SCA snap-in and select "Configure Computer Now". The process will be exactly the same as before, except that the machine will be made to match the template(s) in the database, not just compared.

> **Warning!** There is no "undo" feature in SCA. Test new security settings on a non-production system first, and make a backup of the production server you intend to reconfigure (including the "System State") before you apply the template. Also see the "SECEDIT.EXE /GenerateRollback" option discussed below.

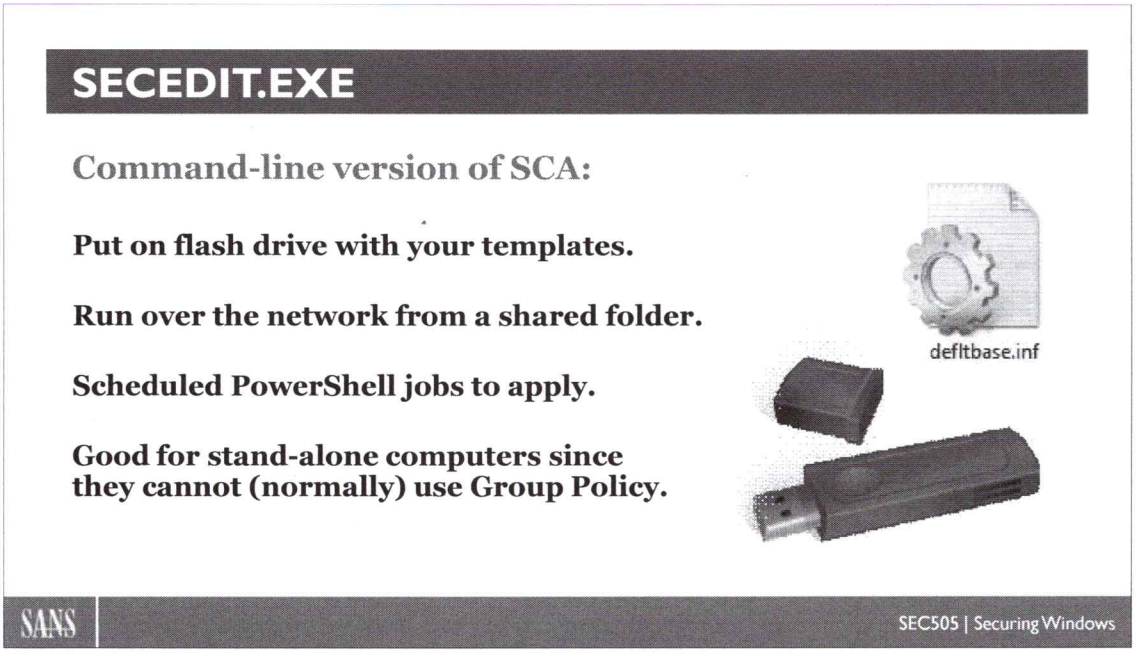
> **Tip:** You can make a snapshot of various parts of your current configuration without being compelled to make a full system backup. For example, branches of the registry can be exported with REGEDIT.EXE, and there are third-party tools

for taking snapshots of NTFS/registry permissions separately from the files/keys with these permissions.

What SCA Cannot Do

The SCA snap-in cannot be used to configure remote systems over the network. To run SCA, you must be sitting at the box or using Remote Desktop Services.

A batch file and SECEDIT.EXE, though, can help get around this problem.



SECEDIT.EXE

SECEDIT.EXE is a command-line version of the SCA snap-in. It can be used to build databases with templates, perform audits, or reconfigure a system. This is very useful for automating the work. Imagine creating a flash drive with SECEDIT.EXE and a simple batch file to run it-- you could field a platoon of IT staff with these floppies to quickly reconfigure hundreds of machines. Even regular users could figure it out (well, maybe).

Alternatively, the necessary files could be placed in a shared folder. Other computers would need only map a drive letter to the share and run SECEDIT.EXE from there. Or a scheduled batch file could reapply settings on a critical server every night at 3 AM.

```
E:\WINDOWS\System32\cmd.exe
E:\WINDOWS\security\templates>secedit /analyze /db dbase.sdb /cfg Generic.inf

Task is completed successfully.
See log %windir%\security\logs\scesrv.log for detail info.

E:\WINDOWS\security\templates>secedit /configure /db dbase.sdb

Task is completed successfully.
See log %windir%\security\logs\scesrv.log for detail info.

E:\WINDOWS\security\templates>
```

In the screenshot above, the Generic.inf template is used to create a database named "dbase.sdb", which didn't exist, and compare the current system against it. Another command-line switch, /log, could have been used to save the output log anywhere desired. The second command just uses the raw database file without specifying the template to reconfigure the machine. The database file was created, in this example, with

the first command. Alternatively, you can build the database with the SCA snap-in and then use SECEDIT.EXE to apply it.

Special Command-Line Switches

Run "SECEDIT /?" to see all the command-line switches, but a few should be noted here:

/AREAS *area1 area2 areaX ...* -- When used with the /export switch, allows you to specify which parts of the database should be exported to a security template. When used with the /configure switch, this allows you to apply the settings from the database to only the selected areas of the machine you desire. You cannot do this with the SCA snap-in. The areas include:

- SECURITYPOLICY: Account policies, audit policies, etc.
- GROUP_MGMT: Restricted groups settings.
- USER_RIGHTS: Logon and user rights settings.
- REGKEYS: Permissions and audit settings on registry keys.
- FILESTORE: Permissions and audit settings on NTFS folders and files.
- SERVICES: Start-up state and permissions on Windows services.

/GENERATEROLLBACK -- This creates a template which can be used to "roll back" the changes another template would make. Create a rollback template for a new experimental template *before* applying it!

/EXPORT -- When not used with the /db switch, this exports the current local Group Policy security settings.

/MERGEDPOLICY -- When used with /export, this collects the security template settings applied from Group Policy with the settings from the local Group Policy object and exports them into a merged template file.

/VALIDATE -- Checks a security template for errors.

/QUIET -- Suppresses all screen and log output.

Note that there is not a switch to specify a remote machine across the network. To apply security templates to many systems over the network, use Group Policy.

As an example, to compare the settings defined in a security template against the local computer, saving the output to a text log file, while generating a temporary database file:

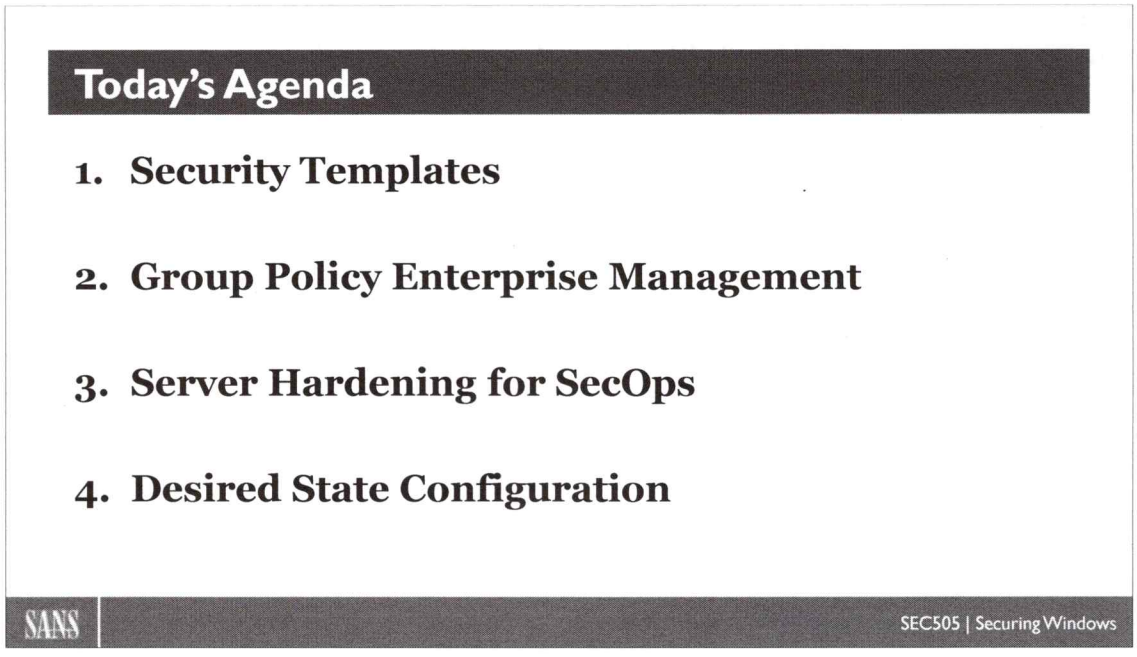
```
secedit.exe /analyze /db temp.sdb /cfg SecurityTemplate.inf /log
log.txt
```

Get the contents of the log file and search it with a regular expression pattern:

```
get-content log.txt | select-string -pattern 'mismatch'
```

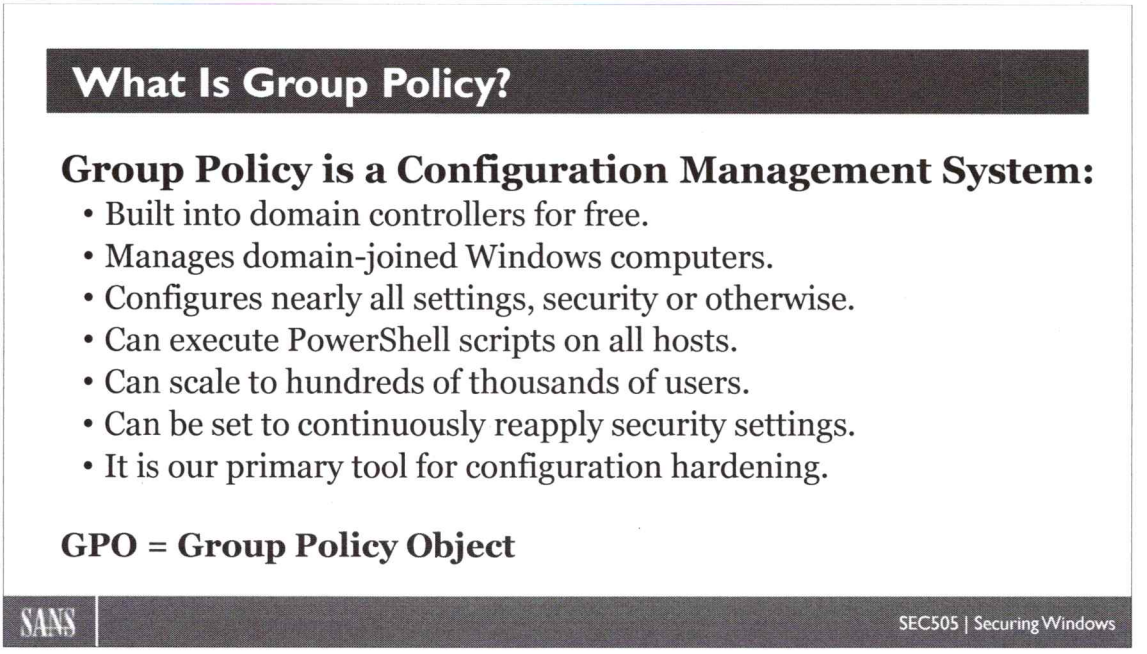
The mismatch lines in the log file show the names of the settings which are not configured the same in the security template and in the local computer.

Unfortunately, the textual log file is not formatted in a way to make it easy to extract all the information you might need to perform an audit. The log file is not XML or CSV or any other particular standardized format, it's just the original developers' debug log.



Today's Agenda

In the next section, we will learn how to use an enterprise configuration management system built into your Active Directory domain controllers: Group Policy. Don't be fooled by the name. Group Policy is immensely powerful and scalable. We can use Group Policy to manage almost every security configuration setting in Windows! Group Policy can also be used to remotely execute PowerShell scripts with Local System privileges as startup scripts or scheduled jobs. This opens the door to vast opportunities to automate our security work and to help the Hunt Team fight back.



What Is Group Policy?

Group Policy is your most important Windows hardening tool. If you manage a Windows network, the first question to ask for any configuration change that needs to be deployed is, "How do I make this change through Group Policy?"

Group Policy, despite the name, is not just for managing groups. Group Policy is an Enterprise Management System (EMS) similar to Microsoft System Center Configuration Manager (SCCM), except that, unlike SCCM, Group Policy is built into Windows for free. Once you have an Active Directory domain, you get Group Policy as a bonus. Any Windows computer joined to the domain can be managed through Group Policy.

There is no workstation or server configuration option that cannot be managed either directly or indirectly through Group Policy. And Group Policy scales to accommodate hundreds of thousands of machines because Active Directory can scale to hundreds of thousands of machines.

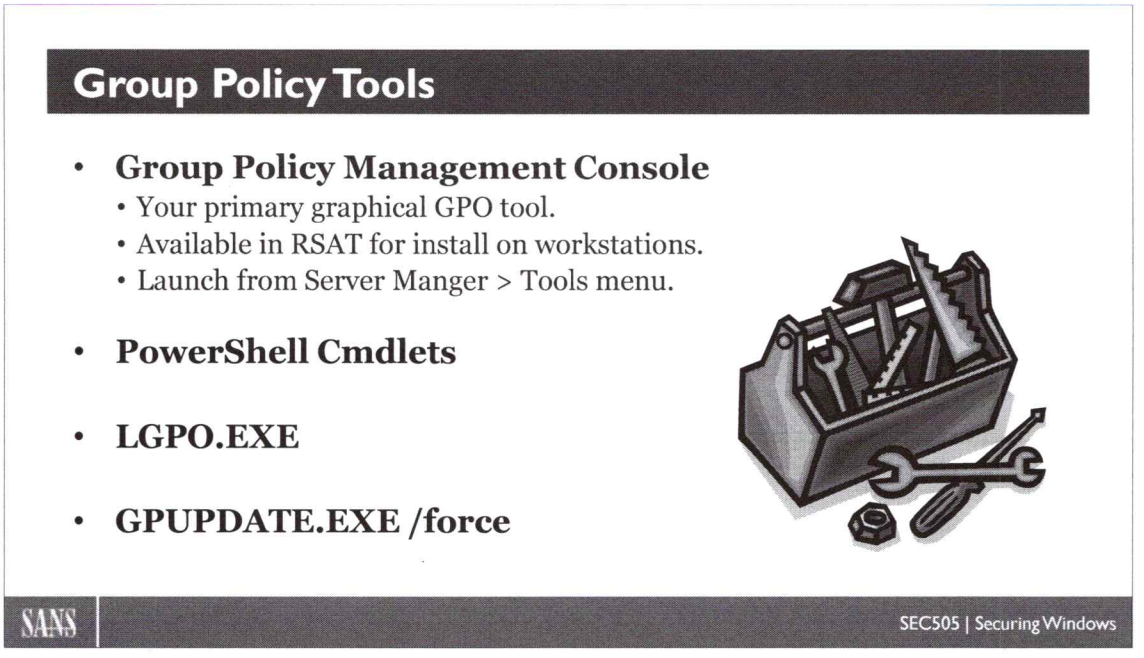
Here is a partial list of what can be done through Group Policy:

- Set any registry value or update any config file.
- Push out and apply .INF security templates hands-free.
- Assign scripts for startup, shutdown, logon and logoff.
- Configure IPSec, PKI, firewall, and wireless settings.
- Redirect users' profile folders to other servers.
- Control which programs users can run through AppLocker.
- Limit access to Control Panel applets and PC Settings.
- Configure virtually every option in Edge and Internet Explorer.

- Install applications (just not as well as SCCM).
- Manage PowerShell policies and scheduled tasks.
- And much more…

An acronym you'll often hear is "GPO", which stands for "Group Policy Object." A GPO is a set of configuration files. GPOs are created and stored on domain controllers, then they are later downloaded and self-applied by Windows clients and servers joined to the domain. Most Group Policy administration centers around the creation, editing and deployment of GPOs to organizational units.

Well, we have a domain controller now, so how do we use Group Policy?



Group Policy Tools

There are a variety of tools for managing and troubleshooting Group Policy, but by far the most important one is the Group Policy Management snap-in for MMC.EXE consoles. The main Microsoft web site for Group Policy, at least for the moment, is http://technet.microsoft.com/windowsserver/grouppolicy/

Group Policy Management Console (And RSAT)

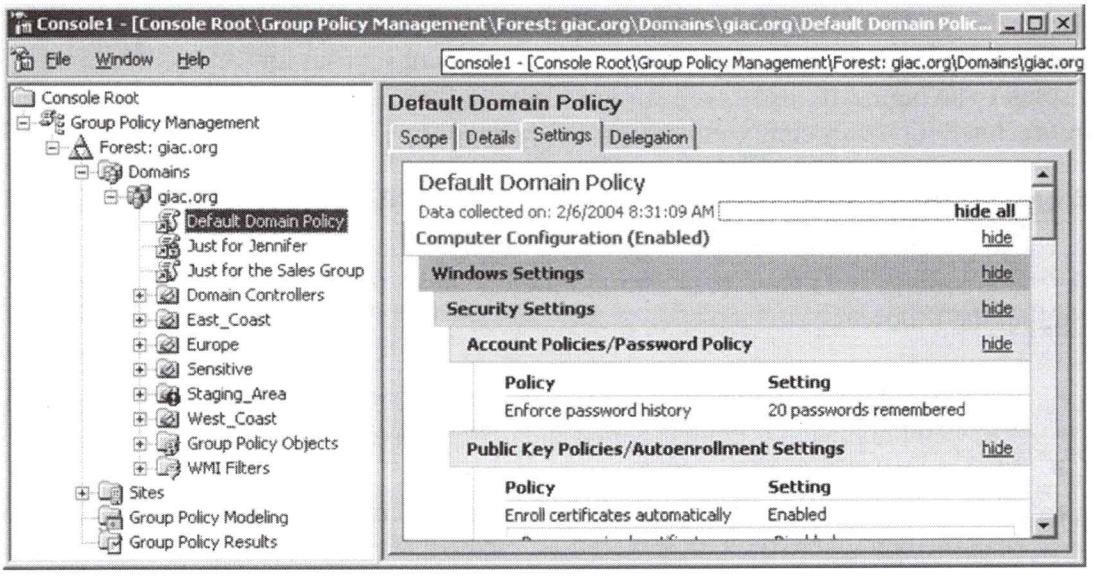
The GPMC is your primary tool for managing Group Policy Objects. It performs many tasks that administrators have been asking for ever since Active Directory first came out, including the following:

- Create, edit, link, enable, disable, search for, and delete GPOs.
- View and manage GPO inheritance and delegation (permissions).
- Manage WMI Filters for GPOs.
- Backup and restore GPOs, including the ability to restore GPOs into a forest other than the one where the GPOs were first created (see KB818736 for details).
- Create and save HTML or XML reports of the settings in particular GPOs.
- Create and save an HTML or XML report of all the final GPO settings enforced on a system or for a user, given that system's or user's location in AD, group memberships, permissions, etc. (Group Policy Results).
- Create and save an HTML or XML report of the final GPO settings a computer or user *would* have *if* they had the arbitrary location, bandwidth, group memberships, permissions, etc. selected by the user of the GPMC (Group Policy Modeling). This feature permits an administrator to ask "what if?" questions concerning Group Policy and to simulate various scenarios without being required to enact them on real systems in order to test new GPO configurations.

- Script most of the above actions so that the GUI is not required. Sample scripts are installed into the %ProgramFiles%\Gpmc\Scripts\ folder by default when the GPMC is installed on a computer (especially note the scripts for exporting and importing the structure of a forest for recreating it in a lab environment).

The GPMC only installs on Windows XP-SP1 or later. You also must install the tool on a member of an Active Directory domain, not a stand-alone, and that member server must have the Microsoft .NET Framework installed. If you have Vista-SP1 or later, you'll have to download and install the Remote Server Administration Tools (RSAT) in order to get the GPMC (KB941314), and then go to Control Panel > Programs and Features in order to install it. You'll want the latest version of the GPMC, so in real life you should use the latest OS, Service Pack and RSAT versions, but, at a minimum, it has to be Vista+SP1+RSAT (or, because Vista sucks, Windows 7 + RSAT).

Note that your domain does *not* have to be running Windows Server 2003/2008 domain controllers. Even a Windows 2000 mixed mode domain can have its GPOs managed through the GPMC tool, but this management still must be performed while sitting at a Windows XP-SP1 or later machine with the .NET Framework installed. And make sure your Windows 2000 domain controllers have at least Service Pack 3 installed (see KB325465 for issues).



Download the latest version of the GPMC from Microsoft's website by visiting http://technet.microsoft.com/windowsserver/grouppolicy/ or by Googling on "site:microsoft.com gpmc rsat". After installing the GPMC on your system, install the "Group Policy Management" snap-in in your favorite MMC console for administering Active Directory. Note that if you have Vista-SP1 or later, you'll have to download and install the Remote Server Administration Tools (RSAT) in order to get the GPMC back again after applying the Service Pack (KB941314).

Note that once you install the GPMC on a Server 2003 system, you will no longer see the standard property sheets on the Group Policy tab of a domain, site or OU in the usual AD-related console snap-ins.

GPMC Backup And Restore GPOs

One of the most useful features of the GPMC is its ability to backup and restore GPOs.

Try It Now!

To back up a GPO to a file, open the GPMC and expand your forest and domain containers, then highlight the Group Policy Objects container (it will look like an OU in the GPMC). Next, right-click on a GPO in the right-hand window pane > Back Up > enter a folder path > click the Back Up button. If you wish to save all GPOs in one shot, right-click the Group Policy Objects container itself > Back Up All. To restore or delete your GPO backups, right-click on the Group Policy Objects container itself > Manage Backups.

GPMC HTML Reports

A GPO contains hundreds of settings, but often only a few are configured. The GPMC can produce an easy-to-read HTML report of only those settings in a GPO which been specifically enabled or disabled. This is immensely useful when troubleshooting.

Try It Now!

To view an HTML report of the configured settings in a GPO, highlight any GPO in the left-hand window pane (that is, don't select a GPO in the right-hand pane when the Group Policy Objects container is selected) and click on the Details tab. A report will be generated automatically. Click the "Show All" link at the top of the report. To save the report as a file to the drive, right-click the GPO itself > Save Report.

Because multiple GPOs can be inherited, and because a user or computer can be a member of various groups each with its own GPO permissions, it can be difficult to say what the final or "effective" GPO settings will be for any particular system or person. The GPMC, however, can show you what these final settings are, and it can tell you from which (possibly inherited) GPO a setting came.

Try It Now!

To see the final GPO settings for a user or computer, right-click on the Group Policy Results container > Group Policy Results Wizard. Answer the Wizard's questions about which user and computer you wish to query and it will produce a familiar HTML report of the final GPO settings effective there. In particular, look at the "Winning GPO" column of the report on the Settings tab.

Testing new combinations of GPO settings on new combinations of users and computers can be extremely time-consuming. The GPMC's Modeling Wizard can drastically reduce this time by helping you to define "what if?" scenarios and then generating a report of what the final GPO settings would be if that scenario occurred in real life.

Try It Now!

To create a report of final GPO settings for a simulated combination of user, computer, OU, bandwidth, and other factors, right-click on the Group Policy Modeling container >

Group Policy Modeling Wizard. Answer the questions to define the scenario and a familiar HTML report will be created for you.

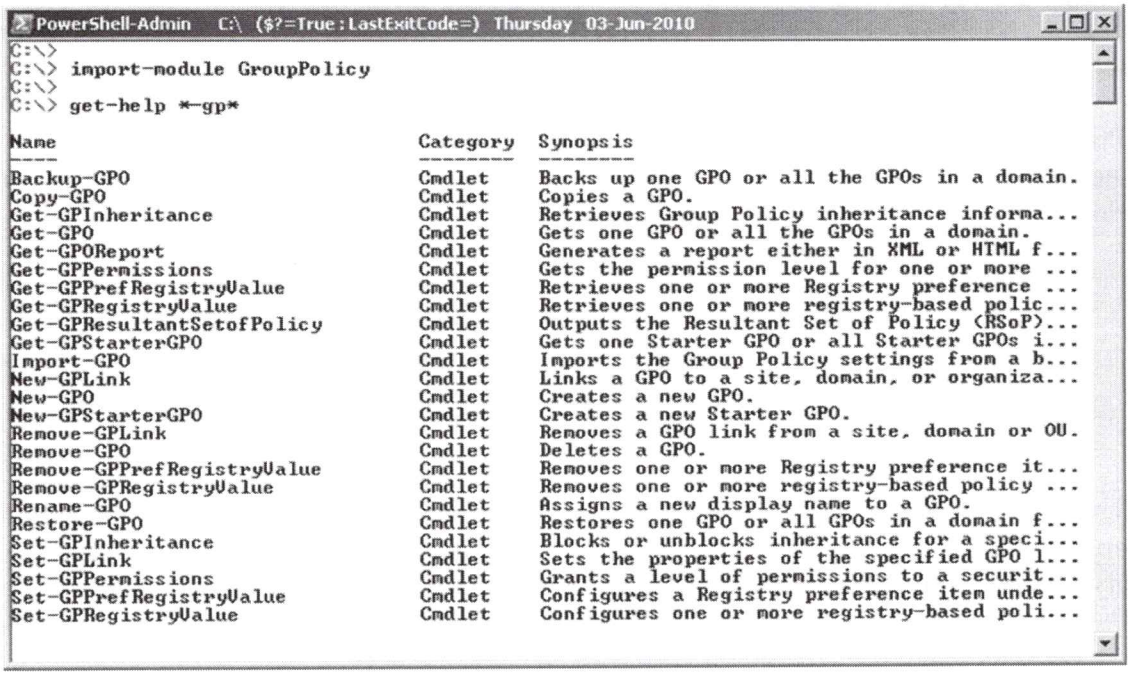
PowerShell Scripting for GPO Management

Most of the functionality of the GPMC has been exposed through scriptable COM objects that your PowerShell and Windows Script Host (WSH) scripts can access. On Server 2003, Microsoft provides sample scripts in the %ProgramFiles%\Gpmc\Script\ folder, but on later versions of Windows, you'll have to download the scripts from Microsoft's web site. Importantly, though, you cannot currently script the editing of individual settings inside of GPOs this way.

If you have Windows 7 or later with the RSAT tools installed, or if have a domain controller running Server 2008-R2 or later, or if you have a domain member running Server 2008-R2 or later with the GPMC installed, then you can use PowerShell cmdlets to perform some GPO management as well. In an elevated PowerShell, run the following commands:

```
import-module GroupPolicy

get-help *-GP*
```

The first imports the Group Policy module containing the GPO-related cmdlets, and the second lists these cmdlets. Explicitly importing the module isn't necessary anymore, but it's a good reminder.



Advanced Group Policy Management (AGPM)

If you have the Microsoft Desktop Optimization Pack (MDOP) for Software Assurance, then you also get Advanced Group Policy Management (AGPM) as a part of MDOP, hence, AGPM is not free because MDOP is not free. AGPM is very useful in medium- and large-sized environments because it provides workflow, auditing, reporting, delegation of authority, automatic backups, rollback, templates, and versioning for Group Policy. AGPM integrates into the GPMC tool.

GPUPDATE.EXE

On Windows XP and later, GPUPDATE.EXE is used to refresh Group Policy settings instead of SECEDIT.EXE as on Windows 2000. It supports more flexible refresh options as well, e.g., forced reboot and/or forced logoff after updating.

To force Group Policy to refresh, even if no GPOs have changed and even if some GPO settings would require a reboot or a relogon to be fully applied, run:

```
gpupdate.exe /force
```

A "synchronous refresh" of Group Policy means that, while GPOs are being processed by the OS during a reboot or logon event, do not allow the user to log on at the console until after all the GPO have been fully processed. This can delay the user getting to his or her desktop, but is also more secure.

To ensure that the next normal refresh of Group Policy at logon or reboot is synchronous:

```
gpupdate.exe /sync
```

Note that /sync does not imply /force, or vice versa. If the /sync switch is used at the same time as the /force switch, the /force switch is ignored.

LGPO.EXE

Command-line tool to manage and apply local GPOs (replaces the old LocalGPO tool that came with the Security Compliance Manager). Very useful on stand-alone machines. Download from http://blogs.technet.com (do a search on the tool name).

GPOTOOL.EXE

Checks GPO consistency and version numbers; compares GPOs on multiple DCs to check replication; displays detailed information about GPOs not available in the snap-ins; browse GPOs based on name or GUIDs. This is a good first check to run, similar to a ScanDisk for GPOs.

SECEDIT.EXE

SECEDIT.EXE is the command-line version of the "Security Configuration and Analysis" snap-in. It can analyze and/or configure a system with a GPO Security Settings INF template from a scheduled batch file. The /refreshpolicy switch can be used

to force the re-application of GPO Security Settings immediately on Windows 2000 (KB227302 and KB227448).

GPRESULT.EXE

Shows the last time Group Policy was applied and from which DC it was retrieved; lists all GPOs applied, the registry values they modified, folders redirected, applications published, disk quotas, IPSec settings, and scripts assigned by GPOs. This utility can produce a vast amount of information in "super-verbose mode" with the /z switch. (See KB258595 and KB250842.)

ADDIAG.EXE

Lists information about Windows Installer applications (.msi installed applications) obtained from Group Policy or otherwise. This is for low-level troubleshooting.

DCGPOFIX.EXE

Re-creates the Default Domain Controllers GPO and/or the Default Domain GPO.

ADMX Migrator

The ADMX templates unique to Vista and later are more difficult to edit, but you can create/edit an ADM template and convert it into an ADMX template with a free tool from FullArmor named the "ADMX Migrator" tool (Google on "site:microsoft.com admx migrator" since it's actually downloaded from Microsoft's site).

Third-Party Group Policy Enhancements

There are a number of GPO extensions available to fill in the gaps left unfulfilled by vanilla Group Policy from Microsoft. We can't discuss or demo them all here, but they are important enough to warrant their own section. Just like Group Policy Preferences was originally a third-party enhancement which Microsoft purchased (discussed later), hopefully Microsoft will purchase some of these companies and incorporate them into GPOs by default too. Unfortunately, none of them are free.

- **Specops (www.SpecopsSoft.com)**
  Specops Software has an add-on which can inventory the hardware and software on domain-joined machines in very large environments, and a software deployment add-on for installing and managing applications, including non-Microsoft applications.

- **PolicyPak (www.PolicyPak.com)**
  PolicyPak allows you to easily manage the configuration settings of Microsoft and non-Microsoft software without having to dig through the registry or ADMX templates yourself. It can also prevent users from changing the configuration settings of that software too, including software you've developed in-house.

- **Avecto Privilege Guard (www.Avecto.com)**
  Privilege Guard is specifically for getting users out of the local Administrators group while still permitting them to exercise administrative control over just the applications, tasks and settings which you define through Group Policy.

- **BeyondTrust PowerBroker Desktops (www.BeyondTrust.com)**
  PowerBroker is a GPO extensions which controls the execution of applications, software installs, ActiveX controls, and system tasks that require administrative rights.

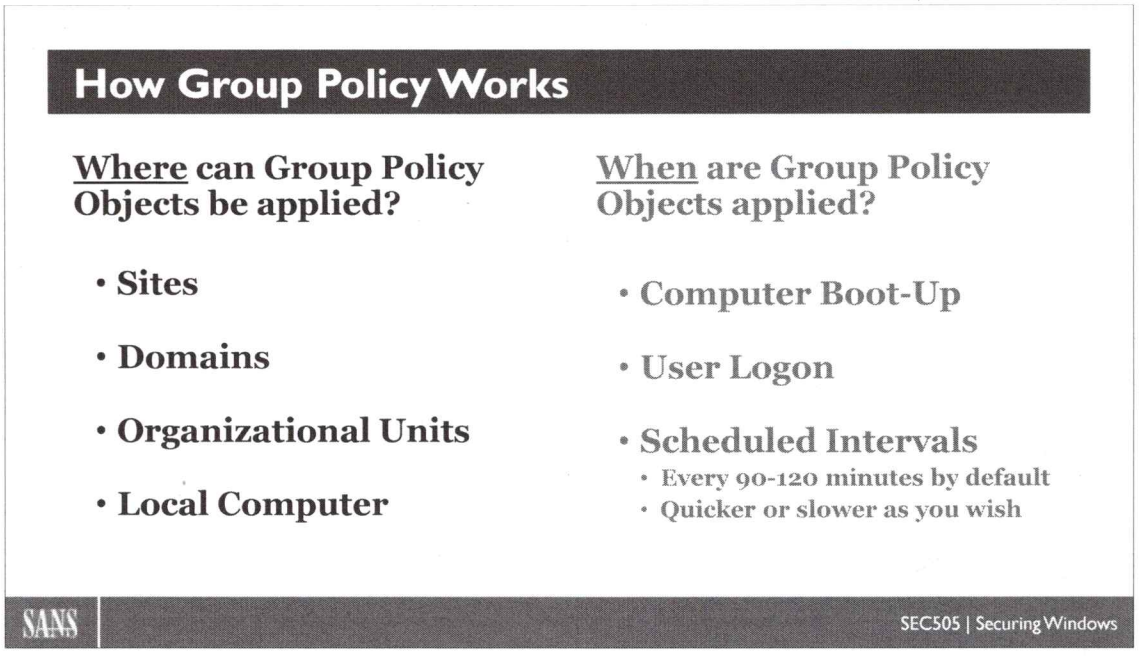
- **Centrify DirectControl (www.Centrify.com)**
  Centrify DirectControl permits the application of Group Policy to Mac and Linux boxes. Centrify also has a related product for Mac and Linux authentication against AD.

Dissatisfied With The GPMC?

As long as we're on the subject, if you're dissatisfied with the GPMC tool, there are replacements and enhancements for this as well. Most of these tools specialize in change control, versioning, automatic backups, auditing, alerting and reporting.

- Blue Lance LT Auditor (www.BlueLance.com)
- Microsoft Advanced Group Policy Management (www.Microsoft.com)
- NetIQ Change Guardian and Group Policy Administrator (www.NetIQ.com)
- NetWrix Change Reporter (www.NetWrix.com)
- New Boundary Policy Commander (www.NewBoundary.com)
- Quest GPOADmin and ChangeAuditor (www.Quest.com)
- SDM GPExpert Automation and Backup Manager (www.SDMSoftware.com)
- SysPro PolMan (www.SysProSoft.com)

If you already have the Microsoft Desktop Optimization Pack (MDOP), then the Microsoft Advanced Group Policy Management tool comes with it. MDOP is not free.



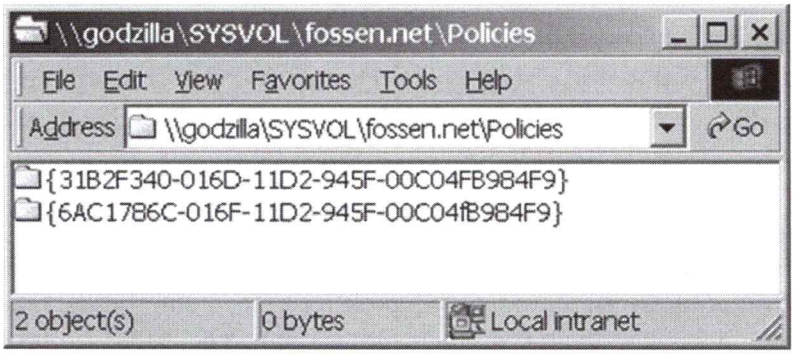
How Group Policy Works

Group Policies are applied to the computer at boot up, to the user at logon, and again at scheduled intervals. When Group Policies are linked to a container they apply to all the users and computers in that container. Group Policies can be linked to:

- Sites
- Domains
- Organizational Units

When the computer boots up, it will query AD to find out which Group Policy Objects (GPOs) apply to the computer and in what order of precedence. GPO information is stored in LDAP://cn=Policies,cn=System,dc=*domainname*. The Policies that are applied may direct the computer to download various scripts and files from a special SYSVOL folder associated with the GPO's Globally Unique ID (GUID) number. The same process occurs for the user when he or she logs on.

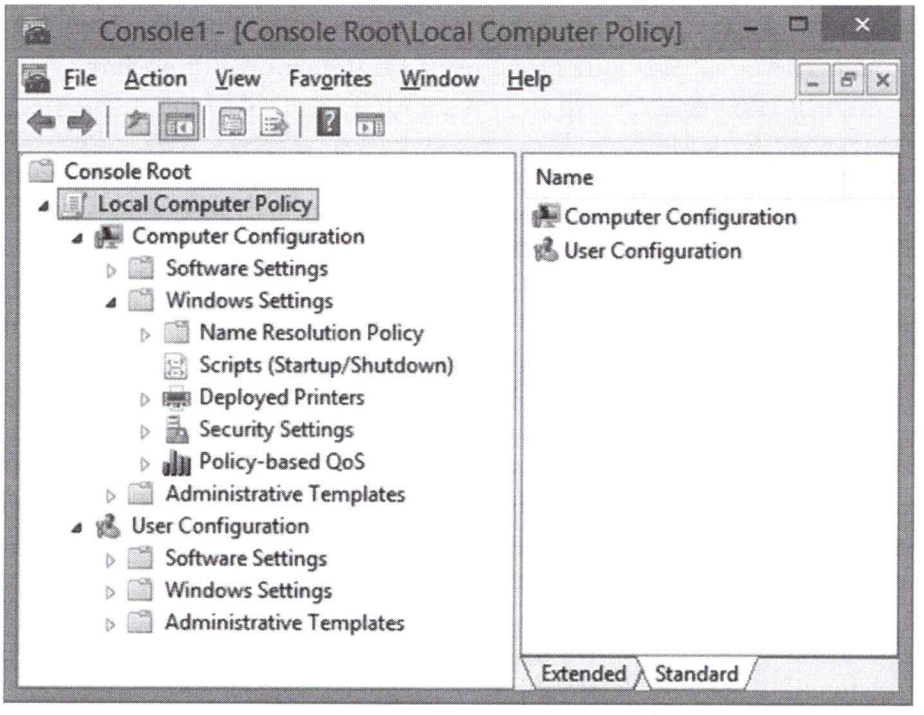
> ***Try It Now!***
>
> Go to the Run line and execute "*servername*" where *servername* is the name of your DC. Then open the \SYSVOL*domainname*\Policies folder. The subfolders are named with GUID numbers which correspond to the GPOs in AD. You can find the GUID of a GPO by going to that GPO's properties > Details tab.



Creating GPOs

Only sites, domains and OUs can have Group Policies linked to them. Though GPOs can be edited separately with the Group Policy MMC snap-in, it is easier to create, edit and delete GPOs at the containers where they are linked.

To create a GPO linked to a domain, site or OU, open the GPMC > right-click that container > Create a GPO in this domain and link it here > enter a GPO name > OK. To edit that GPO, right-click it > Edit.



Try It Now!

In the GPMC, double-click on a domain > select the Default Domain Policy > Edit. This launches the Group Policy snap-in with the Default Domain Policy GPO loaded for editing. Open Computer Configuration > Policies > Windows Settings > Security Settings > Local Policies > Security Options, and double-click the item named "Interactive logon: Do not display last user name in logon screen". Check the box to "Define This Policy Setting" and click Enabled > OK.

Group Policy "Object"

All Group Policy settings are stored in Group Policy Objects (GPOs). Even though the term "object" implies singularity and simplicity, GPOs are anything but. There are many discreet parts of GPOs collectively housing thousands of settings.

Computer Configuration vs. User Configuration

A GPO is divided into two top-level containers: Computer Configuration and User Configuration. The Computer Configuration settings apply to the computer no matter who is logged on at the machine or even if no one is logged on. The User Configuration settings are only applied to the user's desktop when a user logs on. Typically, all the user settings are scrubbed off the machine when the user logs off, i.e., most of these settings do not permanently "tattoo" the computer. The next user who logs on can then override the static settings if their GPOs configure these settings.

When there is a conflict between Computer and User settings, almost always the Computer setting wins. The few exceptions are best discovered by Googling the name of the setting (seriously, there is no one central master source for such information).

What If The User And The Computer Accounts Are In Different OUs?

Excellent question! A user's account might be in one OU, while the account for the computer at which he or she is sitting might be in a different OU (or different domain for that matter). Which GPO settings will be applied to the user's desktop in this case?

When a computer boots up or refreshes its own GPOs, it will only download the GPOs linked to its own domain, site and OUs, based on the location of its computer account in AD. It will then only process the Computer Configuration settings and completely ignore the User Configuration settings from those GPOs. The computer has no idea who will log onto it locally, so it only queries AD to find out where its own computer account is located in the OU hierarchy-- that's all it cares about at this point.

When a user does log on at that computer, the computer will query AD to find out where that user's account lives: which domain and which OU. On behalf of the user, the computer will download all the GPOs that apply to the user and process them. But the computer only processes the settings in the User Configuration container of the GPOs-- the Computer Configuration settings are completely ignored.

Hence, the critical question to ask is, *whose* GPOs are we interested in? The computer's or the user's? In either case, we next ask, *where* is that computer/user account located in AD? This determines which GPOs are downloaded (they don't have to be the same) and which top-level container of the GPOs are processed. Computers only process the Computer Configuration container settings, while users only have the User Configuration container settings applied to them.

It might have been more intuitive had there simply been two different types of GPOs -- Computer GPOs and User GPOs-- but instead there is only one type, broken into two top-

level pieces. Below we will see how to more-or-less create Computer GPOs and User GPOs, if desired, to make the deployment process more friendly.

Group Policy Links

The snap-ins and property sheets for managing GPOs can be confusing. The most important distinction to remember is the difference between GPOs themselves and their "links" to containers. Even though a GPO can be created and modified by going to the properties of a container, a GPO is a separate object in itself, not just a property of the container(s) to which it is linked.

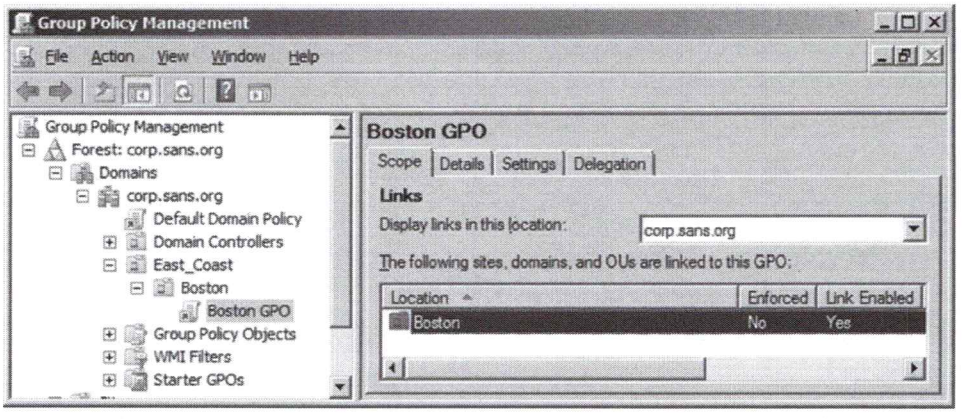
A GPO can be created in AD, but then simply not linked to any site, domain or OU. If it is not linked to any container, then a GPO will never be used even though it exists in AD.

Conversely, a single GPO can be created and linked to dozens of OUs. Change that GPO and all the OUs to which it's linked will receive those changes at the next update.

You can open the property sheet of a GPO and view a list of all the containers to which it is linked. Even if all these links were deleted, the GPO would still exist.

Try It Now!

To view a list of all containers to which a GPO is linked, click on the GPO in the GPMC and look on the Scope tab on the right. The links are shown at the top.



You can also link a pre-existing GPOs to a container without having to define a new GPO from scratch again.

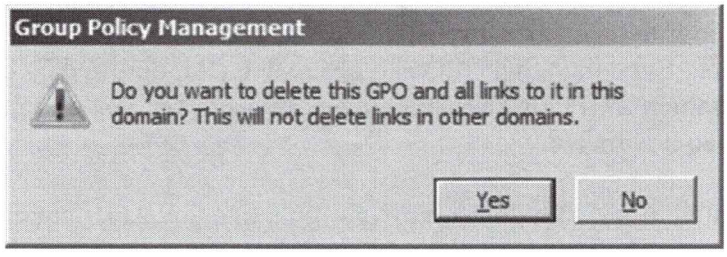
Try It Now!

To link a pre-existing GPO to a container, right-click that container in the GPMC > Link An Existing GPO > select the GPO from the list > OK.

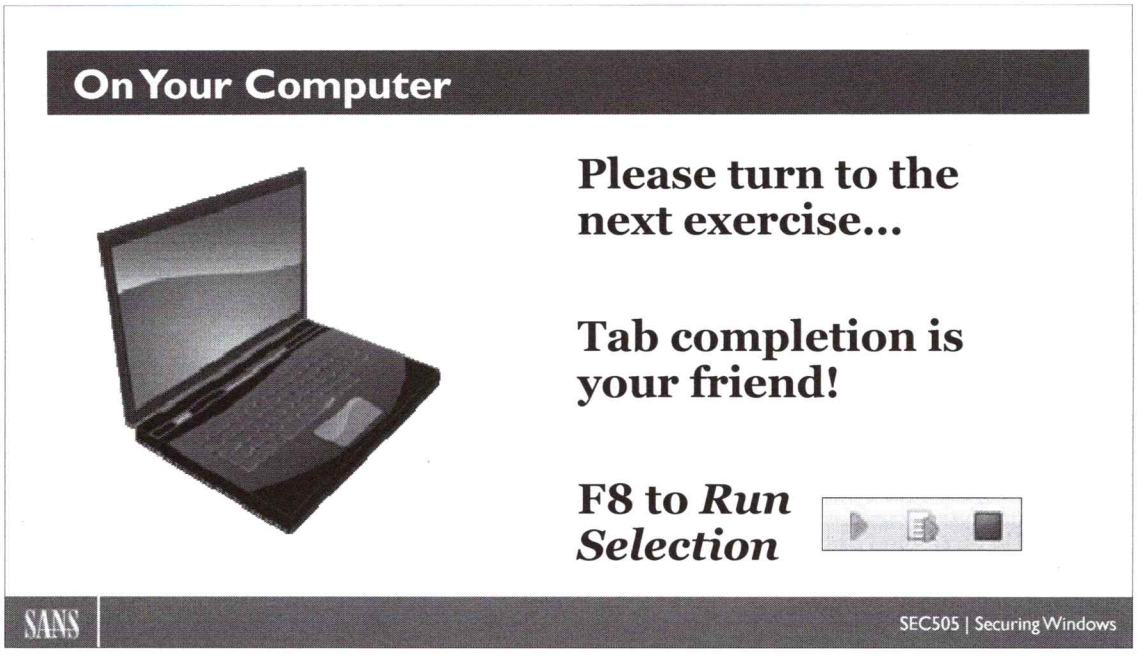
Furthermore, when you delete a GPO from the property sheet of a container, you must choose whether you want to 1) delete just the link or 2) delete the GPO and all of the links to it.

Try It Now!

To delete a GPO link to a container, right-click the GPO > Delete > OK. But to delete the GPO itself, go to the Group Policy Objects container in the GPMC > right-click a GPO > Delete > Yes.



Later, when we discuss delegation of authority over GPOs, we'll see that the link is a property of the site, domain or OU. Hence, separate permissions can be placed on links and GPOs because they are separate items.



On Your Computer

Using the Group Policy Management console, expand your list of forests, expand your list of domains, expand the testing.local domain, find the Default Domain Policy GPO linked to your domain, right-click the Default Domain Policy GPO and select Edit.

In the Default Domain Policy GPO, navigate down to User Configuration > Policies > Administrative Templates > System > Ctrl-Alt-Del Options. On the right-hand side, edit and enable the option named "Remove Task Manager".

In the Default Domain Policy GPO, navigate down to User Configuration > Policies > Administrative Templates > System > Scripts. On the right-hand side, edit and enable the option named "Run logon scripts synchronously".

Close the Group Policy Object Editor window showing the Default Domain Policy GPO. Your changes have already been saved to the GPO. There is no "Apply" button.

In PowerShell, import the Group Policy module:

```
import-module grouppolicy
```

List the available Group Policy cmdlets:

```
get-command -module grouppolicy
```

Create a new GPO named "Sales_Group_GPO":

```
new-gpo -name "Sales_Group_GPO"
```

> **Note:** You do not have to create GPOs in PowerShell. GPOs are normally created using the Group Policy Management snap-in. This is just for practice.

Link that new GPO to the testing.local domain:

```
new-gplink -name "Sales_Group_GPO" -target "dc=testing,dc=local"
```

Create an HTML or XML report of the settings inside the Default Domain Policy GPO:

```
get-gporeport -name "Default Domain Policy" -reporttype html
  -path c:\temp\report.html
```

View that HTML report in your browser (simply execute the name of the file):

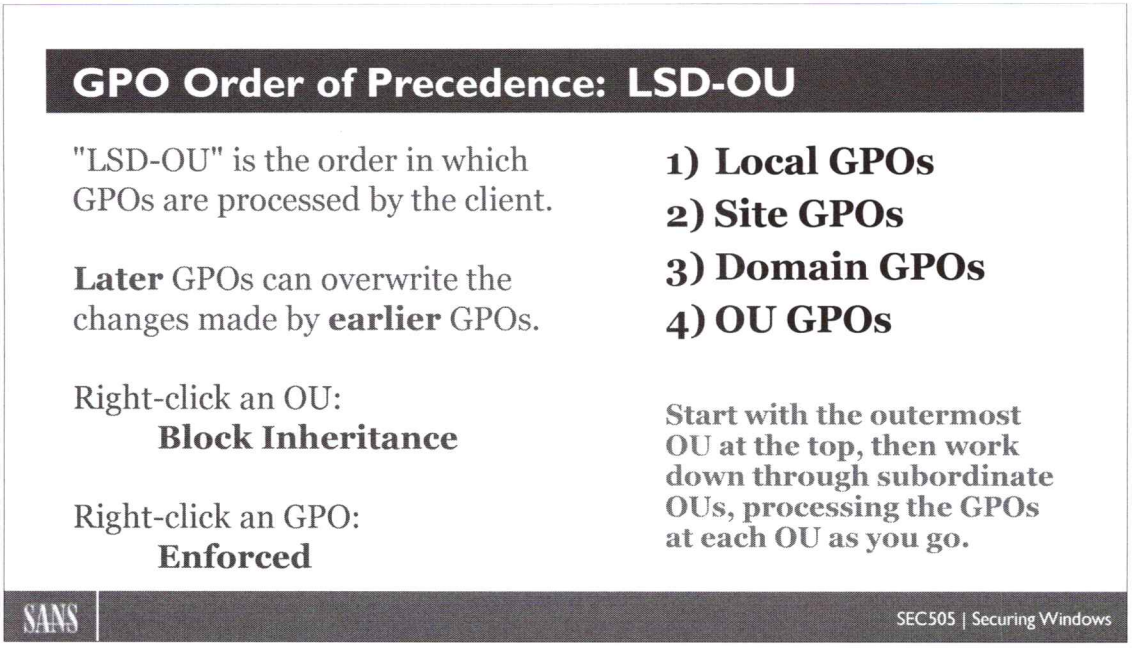
```
c:\temp\report.html
```

Close the browser when you are done.

In PowerShell, run this command to refresh group policy and log off:

```
gpupdate.exe /force ; logoff.exe
```

Log back into your VM and use the VM interface to send a Ctrl-Alt-Del keystroke sequence to your VM (it should be a menu option). Notice that Task Manager is no longer present as an option. This means Group Policy has hidden it. (The change we made for running logon scripts synchronously will be used later.)



GPO Order of Precedence: LSD-OU

Multiple GPOs can apply to a single computer or user. The settings in these GPOs may conflict with each other, hence, the order in which they are applied is important. In general, settings applied later override settings applied earlier when they conflict (exceptions are below).

GPOs are applied in the following order. Note that not all must be implemented, and later GPOs override earlier GPOs which are applied. The default order can be memorized with the acronym "LSD-OU":

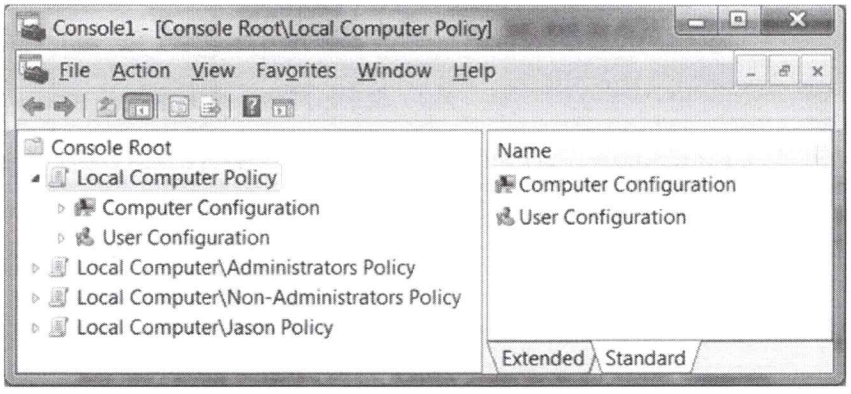
1) Local GPO(s) -- these are stored on the machine, not in AD.
2) Site GPOs
3) Domain GPOs
4) Organizational Unit GPOs in nested order (outermost to innermost)

OU GPOs are applied in nested order: largest container down to smallest (inside) container holding the user or computer. In this sense, the domain container is just the outermost container, hence, its GPOs are applied before any of the OU's GPOs.

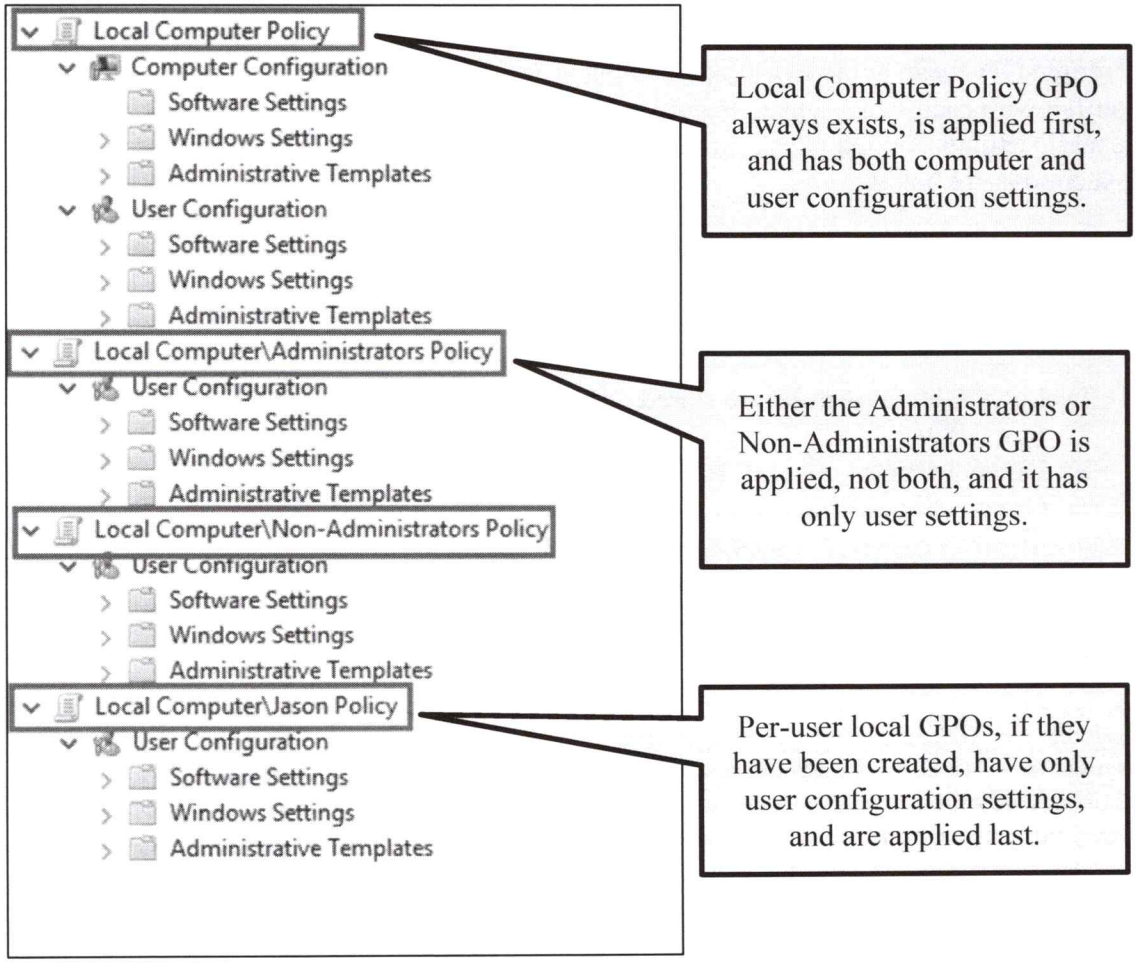
Multiple Local GPOs

Windows Vista and later systems can have multiple local GPOs. When you add the Group Policy Object Editor snap-in to an MMC console, click the Browse button, go the Users tab. There you can select Administrators, Non-Administrators, or a particular user who has ever logged on in the past. The computer's local GPO is always applied, then it's either the Administrators local GPO or the Non-Administrators local GPO (not both, it

depends on whether the user is a member of the Administrators group), and then finally the per-user local GPO, if a GPO exists for a particular user (none exist by default).



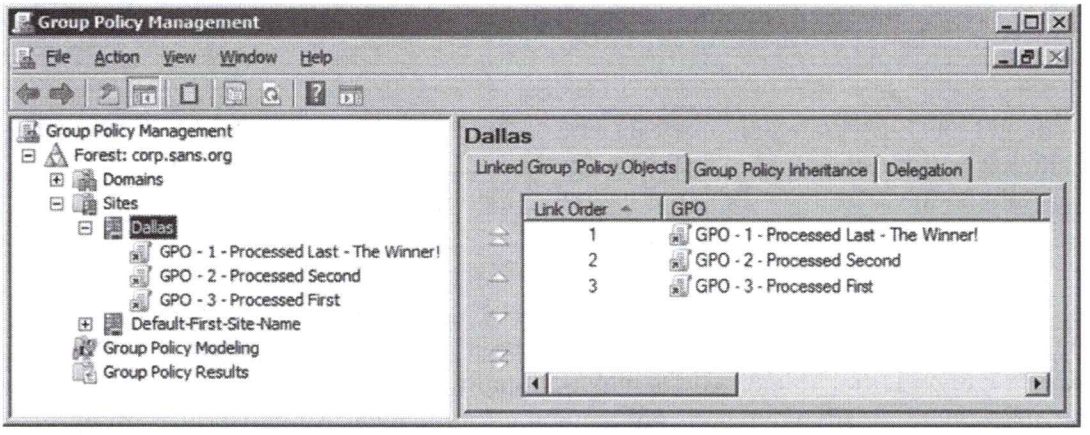
When you expand these local GPOs, notice that the default Local Computer Policy GPO has both computer and user configuration settings, while the Administrators, Non-Administrators, and per-user local GPOs (such as for Jason) only have user configuration settings.



Can Multiple GPOs Be Linked To A Single Container?

Yes. If multiple GPOs are linked to a single container, then they are applied in the *reverse* order shown under the Link Order column, i.e., they are processed from bottom to top. Remember that "last writer wins", so the last GPO to be processed for a container is the winner in any conflict with other GPOs linked to that container. The winning GPO is at the top of the list *because* it is processed last.

The screenshot below is of the Group Policy tab on the properties of an OU. The names of the GPOs indicate the order in which they will be applied.



Block Inheritance

You can prevent the users and computers in an OU from inheriting any GPOs from the domain or any higher-level OUs. This can simplify the GPO infrastructure and may be desired for political reasons.

> **Note:** If the inheritance of GPOs is blocked for political reasons, keep in mind that Kerberos policy, password policy and lockout policy will still be enforced for everyone because everyone shares the same domain controllers.

If you wish to block the inheritance of GPOs from any parent containers at an OU, then right-click that OU in the GPMC > Block Inheritance. Local GPOs are still processed, though, even if this box is checked.

Enforced (a.k.a., "No Override")

On the other hand, it is possible to force a parent container's GPO down onto all subcontainers. This is true at both the domain and OU level, i.e., one OU can force its GPOs down onto all the sub-OUs beneath it.

> ***Try It Now!***
>
> To force a GPO down onto subcontainers, overriding whatever conflicting settings those other GPOs may possess > right-click that GPO in the GPMC > Enforced.

> **Note:** This feature used to be called "No Override" in Windows 2000/2003.

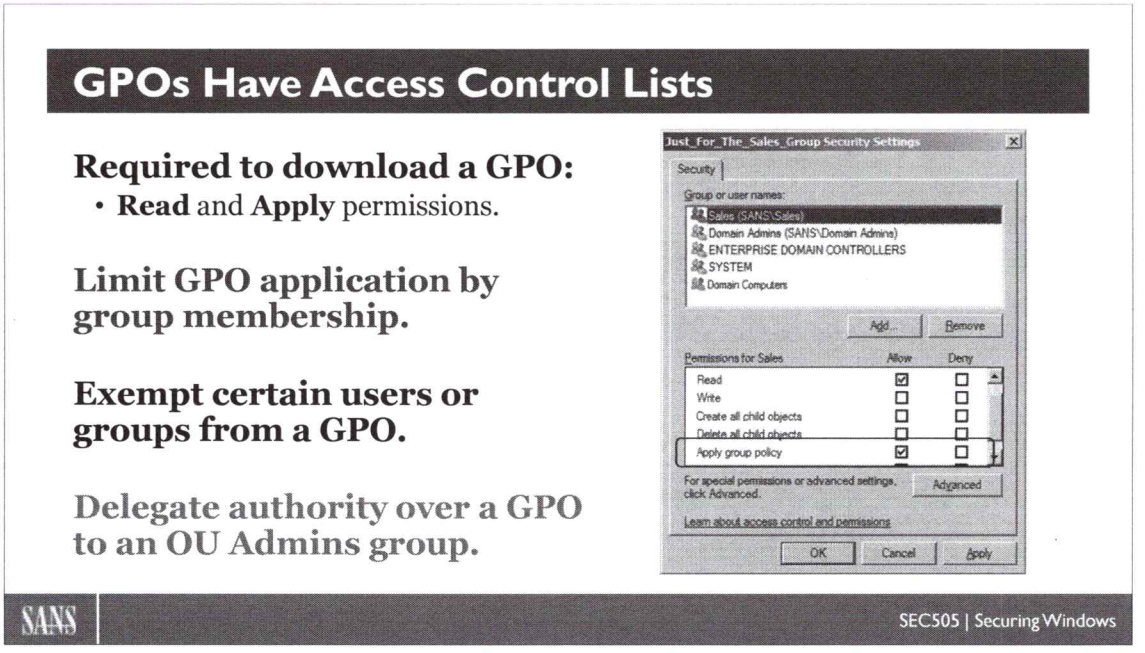
For example, there may be a domain-wide policy you may wish to enforce across all OUs, hence, you do not want to allow the OUs to override your policy settings. Enforced is configured on a per-GPO-link basis, so you will have precise control over what is compulsory and what is optional for the OUs. Because it is configured *per link* a single GPO that is linked to multiple containers might have Enforced set for some of the links to some of those containers while not having it set for other links to other containers. Politically, any domain-wide or compulsory GPO settings should be agreed upon in the IT Committee first in order to avoid secession from the AD union and bloodshed.

If a parent container GPO is set to Enforced, and one of its child container GPOs is set to Block Inheritance, the child container will still inherit anyway. Enforced overrides Block Inheritance.

If a container has Enforced, and one of its subcontainers also has a GPO set to Enforced, then whichever Enforced GPO is further *up* in the hierarchy (further outside in the nesting) will be the effective GPO. For example, a domain GPO with Enforced will override any OUs with Enforced GPOs.

Local GPOs are applied even when Block Inheritance is enabled on the container(s) holding one's computer and/or user account. There is always the built-in Local Computer Policy GPO, but additional local GPOs may optionally be added for members of the Administrators group, anyone who is not a member of the local Administrators group, and optionally for each local user account as desired. When there are multiple local GPOs, the built-in Local Computer Policy GPO is applied first, followed by either the Administrators local GPO or the Non-Administrators local GPO (not both), and then finally any per-user local GPOs. Of course, after the local GPO(s) come all the GPOs from the domain controllers.

GPOs with Enforced set override everything except Loopback Mode GPOs.



GPOs Have Access Control Lists

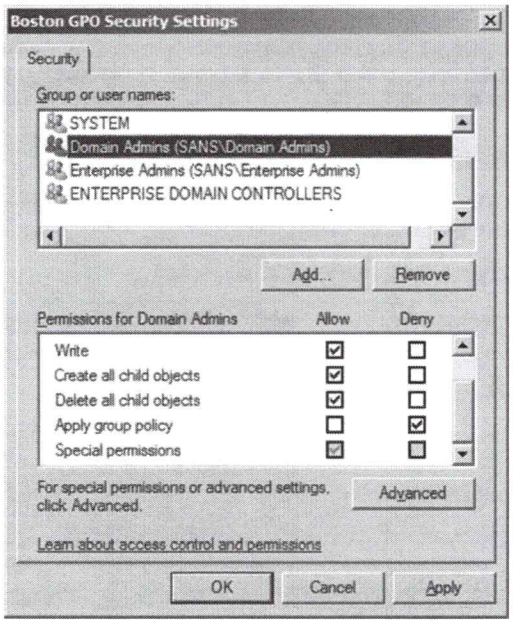
The critical parts of a GPO exist in Active Directory, hence, GPOs have access control lists! These permissions can be leveraged to fine-tune GPO distribution and to delegate authority over them.

A user must have at least the Read and Apply Group Policy permissions in order to have a GPO applied to his or her desktop. Hence, to apply a GPO only to a particular user or group, set the permissions on the GPO so only that user or group has Read and Apply Group Policy permissions.

Conversely, a user or group (or computer) can be made exempt from an existing GPO by assigning the Deny Read and Deny Apply Group Policy permissions on that GPO.

Try It Now!

To view the permissions on a GPO, highlight that GPO in the GPMC > Delegation tab > Advanced button. Note that Authenticated Users have Read and Apply Group Policy permissions. Click Add and add the Administrator account. Assign Deny Apply Group Policy to the Domain Admins group. This makes Domain Admins exempt from this GPO. Click OK.



Delegation of Control Over GPOs

To edit a GPO, a user requires at least Read and Write permissions on that GPO. In order to change the permissions on a GPO, a user must have Full Control of it.

A typical scenario is when a Domain Admin creates a few generic GPOs, keeps Full Control for him- or herself, but gives Write permission to an OU Admins group. That OU Admins group can now modify those GPOs as desired and link them to their own OU. That same group, however, cannot modify any other GPOs and cannot change which GPOs are linked to anybody else's OU.

Permissions on GPOs are different from permissions on OUs, even though it seems like GPOs are just properties of OUs (because that's where we typically create/edit the GPOs). And a GPO link is, strangely enough, not a property of the GPO but of the OU, site or domain to which it is linked! Hence, an OU Admins group would have control over which GPOs are linked to their OU, but they do not necessarily have (or need to have) the Full Control permission over any of those linked GPOs. Again, to modify a GPO one only requires the Read and Write permission to it.

> ***Try It Now!***
>
> To manage the ability to link a GPO to an OU, open AD Users And Computers > right-click on the OU > All Tasks > Delegate Control > Next > select a user or group > Next > check the box to Manage Group Policy Links > Next > Finish. Or you can set the ACLs manually.

When you delegate control to a user or group in order to manage Group Policy links on an OU or site, that user/group is granted the following permissions on the container:

- Read : gPLink
- Write : gPLink

- Read : gPOptions
- Write : gPOptions

You can assign these same permissions manually without the Delegation of Control Wizard. These permissions would already be possessed by any OU Admins group which had been granted Full Control to the entire OU.

MS16-072 Permissions Changes

In 2016, Microsoft released a security update (MS16-072) which changes how Windows downloads GPOs and enforces GPO permissions. Any computer with this update will download all GPOs under the context of the computer, even when that GPO is for the user. Previously, user GPOs were downloaded under the context of the user, not as the computer where the user is logging on.

If the permissions on a GPO grants Read access to the Authenticated Users group, then there is nothing to worry about. This permission is also the default. If a GPO has the default permissions, there is nothing to do and nothing will be broken. But if the Read permission for Authenticated Users group has been removed or denied on that GPO, then Read access to the GPO must be granted to the Domain Computers group or to some other group which includes the Active Directory computer accounts of the machines which will need to download the GPO. For more information, see KB3163622.

More Processing Options To Fine-Tune GPOs

In addition to GPO permissions, there are other options that can be configured to fine-tune how Group Policy operates. Unless stated otherwise, all of the following options are located under Policies > Administrative Templates > System > Group Policy. Most are under Computer Configuration, but some are User settings.

Group Policy Slow Link Detection

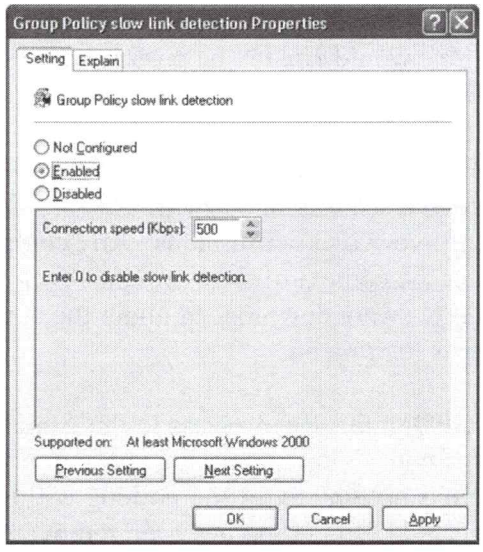
You may wish to prevent the application of certain GPO settings when the user is connecting over a dial-up, VPN or other relatively slow link.

Security settings, ADM registry value changes, EFS Recovery Policy, and IPSec Policies are *always* applied, no matter how slow the link is (despite what the GPO Explain tab says).

On the other hand, MSI/ZAP packages, scripts, IE configuration, disk quotas and folder redirection settings are *not* applied over slow links. But you can override these don't-apply defaults with the other policies in that container, e.g., "Scripts Policy Processing" can be configured to download GPO-assigned scripts even if the bandwidth is slow.

How slow is "slow"? By default, a slow link is 500 Kbps or less. The client pings the DC to determine bandwidth, or, if ICMP is being blocked, it uses the timing in SMB. You can adjust this number in the "Group Policy Slow Link Detection" option. When found under Computer Configuration, this option is for the settings in that container.

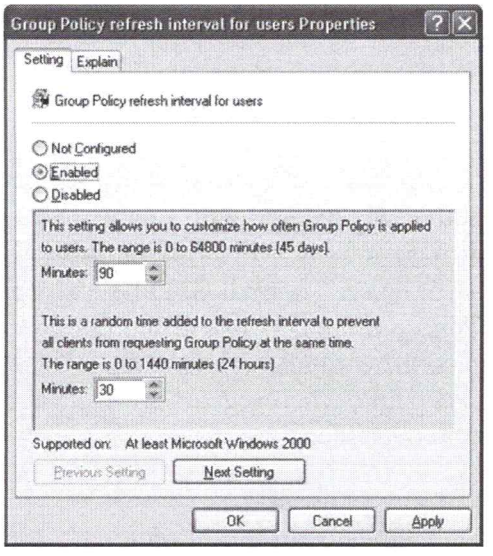
When found under User Configuration, this option is for the settings in the User Configuration container. Probably best to set them to the same bandwidth.



Note: There are similar slow-link options related to roaming user profiles. Look in the Computer Configuration > Policies > Administrative Templates > System > Logon container for these options. Except for software packages and long scripts, it usually takes longer to download one's profile than one's GPO settings.

Group Policy Refresh Interval

By default, GPOs are refreshed automatically in the background every 90 minutes, plus/minus a random number of minutes up to 30 (except on DCs). Under both Computer and User Configuration, you can change these numbers with the "Group Policy Refresh Interval" option. The range is 0 to 64,800 minutes, which equates to once every seven seconds to once every 45 days. Unfortunately, the randomizer can only be up to 1440 minutes (one day). Setting larger values is very useful when DCs are being overworked or a large software deployment is in the works.



You can also disable background updates entirely with the "Turn Off Background Refresh of Group Policy" option in Computer Configuration. In this case, GPOs are applied only after the user logs off.

Notice, however, that there is a separate refresh interval for domain controllers. For security reasons, DCs update their GPO settings every five minutes by default. In sites with a large number of DCs this can be increased to 10 minutes with a 5-minute randomizer to somewhat coincide with the 15-minute intrasite replication latency, but the default is usually just fine.

Loopback Processing Mode

There are times when you want a fixed set of User Configuration settings at certain machines, no matter what user sits down at it. For example, public kiosks and computer labs require locked-down desktops, and you don't want the lax GPOs of users to follow them there.

Normally, when a computer processes its own GPOs it will ignore the User Configuration settings in its GPOs. What we want is for the computer to not do this anymore. We want the computer to apply the User Configuration settings of *its own* GPOs to any user that logs on at it, and to ignore the User Configuration settings from the user's GPOs.

To achieve this, enable "User Group Policy Loopback Processing Mode" in at least one of the computer's GPOs. This option is located under Computer Configuration > Policies > Administrative Templates > System > Group Policy.

Try It Now!

In the GPMC, right-click the Default Domain Policy > Edit > Computer Configuration > Policies > Administrative Templates > System > Group Policy, then open the policy named "Configure User Group Policy Loopback Processing Mode". Click on the Extended tab and read > Cancel > Close > Cancel.

Replace vs. Merge
Loopback mode supports two options: Replace and Merge.

- Replace: Ignore all User Configuration settings from the other GPOs which would normally be applied to a user. In short, replace the normal settings with just the ones specified in the GPO with Loopback Replace enabled.

- Merge: Apply all of a user's normal GPO User Configuration settings, but then apply the User Configuration settings of the Loopback Merge GPO. If there are conflicting settings, the computer's loopback GPO settings will win. This option permits users to keep all of their regular preferences except for the ones you specifically want to override.

Loopback Overrides Enforced Option
Keep in mind that loopback mode overrides "Enforced". If one of a user's GPOs has the Enforced option set, that GPO's User Configuration settings are still overridden by the User Configuration settings in the computer's loopback mode GPO.

Loopback Enabled By Default For Cross-Forest Logons
Note that loopback mode with the replace option is enabled automatically when logging on at a computer in a different forest than the forest which contains one's user account. In this case, the user settings from the GPO for the computer at which one is sitting are applied to the user, while the user's normal user-related settings from their GPOs in their home forest are just ignored. If you don't want this unexpected behavior, then turn it off by going to Computer Configuration > Administrative Templates > System > Group Policy Allow Cross-Forest User Policy And Roaming User Profiles.

WMI Filtering

Windows Management Instrumentation (WMI) is a scriptable interface for querying and managing a very wide variety of settings in the operating system, Active Directory, IIS, and more. GPOs in Windows Server 2003 and later can leverage the power of WMI to fine-tune exactly to which users and computers a given GPO will apply, just as GPO permissions can be used to fine-tune the application of GPOs.

You use WMI filtering by writing a snippet of script code and pasting the code into a GPO. The script will access the WMI interface and perform a test to see whether the GPO should be applied. The test can be based on any criteria accessible through the WMI interface, and that is *a lot*! WMI can extract information about processes, drivers, users, shares, software versions installed, patches, Service Packs, printers, Event Logs, ports, network adapter card settings, IP configuration, registry values, and more. Any or all of this could be combined or scrutinized for the WMI filtering test of the GPO.

> ***Try It Now!***
> To configure a WMI filter on a GPO in Windows Server 2003 and later, write your WMI query and copy it to the clipboard. Next, right-click the WMI Filters container in the GPMC > New > enter a name for the filter > Add button > paste in your query > OK >

Save. To assign this filter to a GPO, highlight that GPO > go to the Scope tab on the right > pull down the list of WMI Filters at the bottom > select the one you want > Yes.

An example piece of code is the following; it would be used to determine whether or not the target computer that is about to process the GPO is running Windows Server 2012 Standard or not; if it is, then run the GPO, otherwise prevent the GPO from being processed.

```
SELECT * FROM Win32_OperatingSystem WHERE Caption = "Microsoft
Windows Server 2012 Standard"
```

If you think this looks like SQL, you're correct! It's actually "WMI Query Language (WQL)", but WQL was loosely modeled on SQL, so if you've got database management experience then you've got a good head start on understanding WQL. WMI and WQL will be discussed again in the "Scripting for Security" course.

Disabling GPOs and GPO Settings

GPOs can be disabled in whole or in part. It's a good practice to immediately disable a GPO after creating it, then re-enable it only after all of the edits you want have been made. This prevents users from receiving partial changes. Disabling GPOs and portions of GPOs is also very useful when troubleshooting because it helps to identify which GPO is misbehaving.

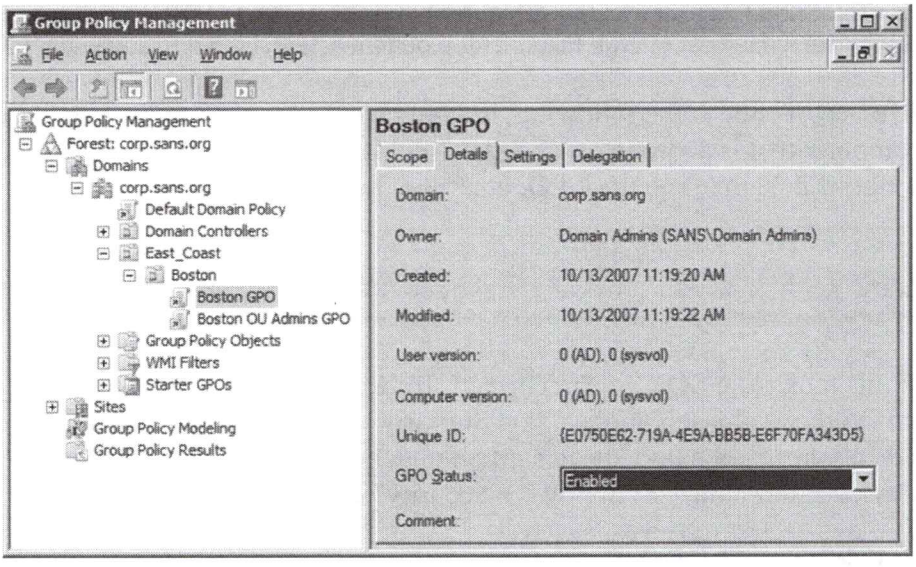
Also, when you want to delete a GPO, disable it first, let that change replicate to all DCs, let all computers and users update their settings (this could take a few hours), and then finally delete the GPO. If you simply delete the GPO its settings will not be gracefully reversed out of the systems which have applied it.

Disabling A GPO (Or A Portion Of A GPO)

When you disable an entire GPO, you will get the following warning because the change takes effect immediately. "Reversing out" changes simply means reapplying all GPOs minus the settings being disabled (except for ADM "preferences"--discussed later).

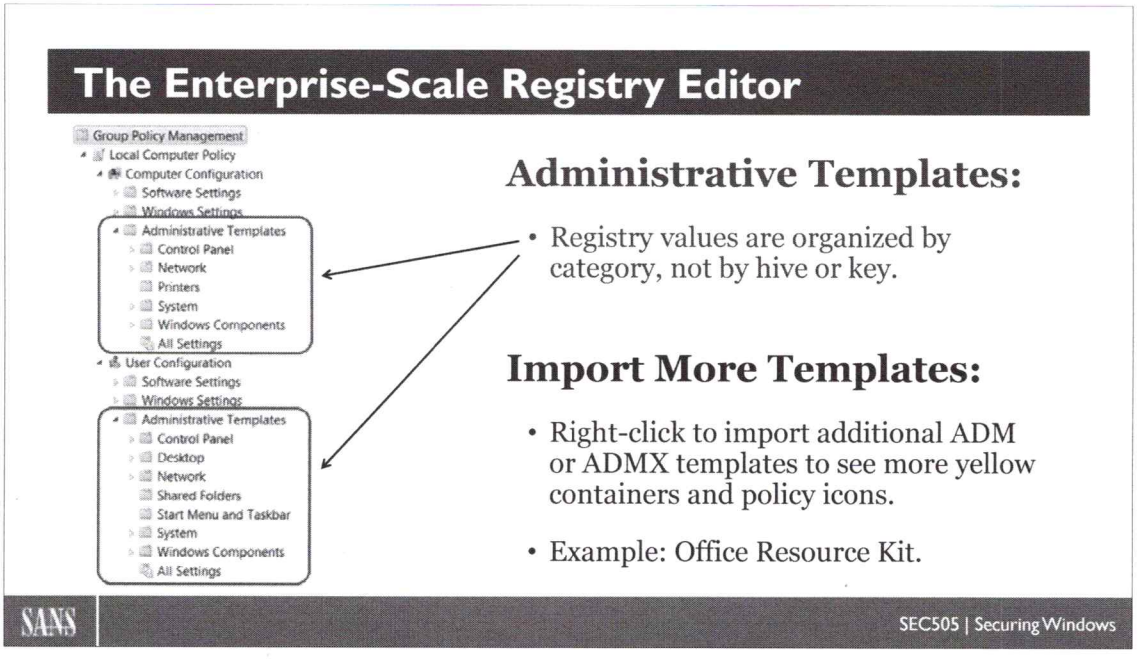
Try It Now!

To disable an entire GPO, highlight the GPO to be disabled > Details tab > select All settings disabled from the GPO Status list > OK.



Note that you can also disable just the user-related settings or just the computer-related settings. This is recommended because 1) you might create separate GPOs for user and computer settings, and 2) disabling the unused portion of the GPO improves logon performance.

You can also disable a single setting within a GPO. This is done rather commonly because a default setting will likely be enabled at the domain level or a high-level OU. But an OU which requires that setting to be disabled can do so without affecting the rest of the AD or the GPO from which the enabled setting was inherited.



The Enterprise-Scale Registry Editor

The Administrative Templates containers in GPOs change registry values on the machines which process them. Because the registry controls more-or-less *everything*, this is an extremely important capability. Many security hardening tasks are accomplished through registry edits. Now, through Administrative Templates, these edits can be made on thousands of machines quickly and easily.

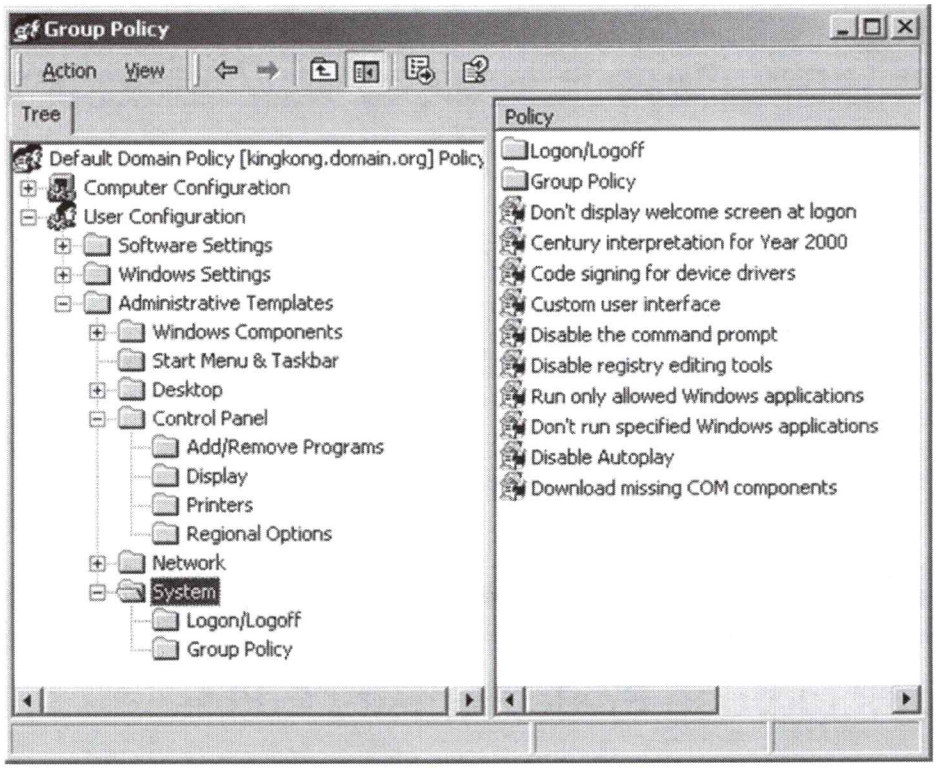
Administrative Templates can be found under both Computer Configuration and User Configuration. Whenever there is a conflict, the Computer setting almost always wins.

- Computer Configuration > Policies > Administrative Templates section of a GPO modifies the **HKEY_LOCAL_MACHINE** key of the user's registry.

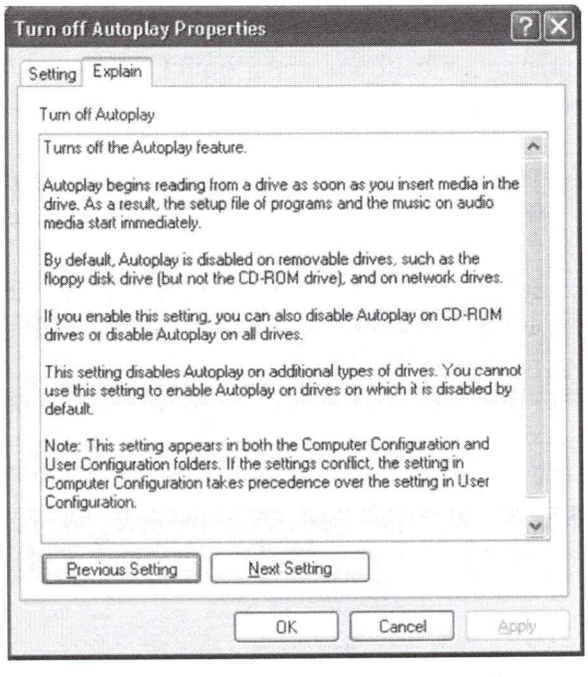
- User Configuration > Policies > Administrative Templates section of a GPO modifies the **HKEY_CURRENT_USER** key of the user's registry.

Browse The Templates And View The Explain Tabs

Browsing through the yellow subcontainers you'll find hundreds of configuration settings that can be enabled (far too many to list or discuss here). Spend some time going through the containers to get a feel for what's possible, especially under User Configuration.



Note that most option icons, when you double-click them, have an Explain tab to describe what the option is and any caveats. For a detailed explanation, see the Group Policy documentation that accompanies the Windows *Resource Kit*.



Why Do I See Just These Yellow Folders And Options?

Notice that these yellow folders and option icons do not represent the entire registry. Exactly which options you see in a GPO is determined by which ADM templates have been imported into that GPO. Multiple templates can be imported into a GPO where they are merged together and displayed as yellow containers and option icons.

Templates are ASCII or Unicode text files with .ADM extensions. They can be modified by hand if desired. ADM templates are typically stored in \%SystemRoot%\Inf, and many are included with the operating system.

Importing Templates

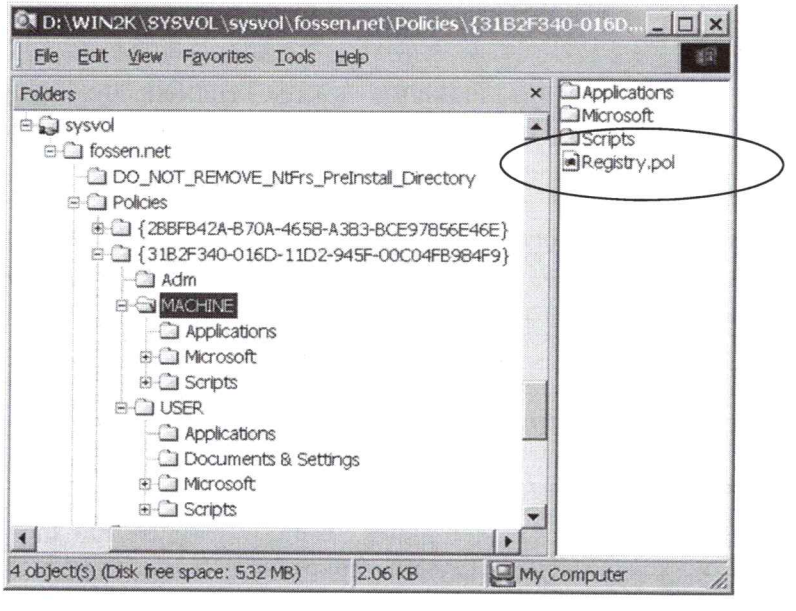
Before the registry settings in a template can be used, the template must be imported into a GPO. Once imported, additional containers and options will become available under the Administrative Templates sections.

Try It Now!

To add another ADM template to a GPO, open the Group Policy Management Console > right-click a GPO > Edit > Computer Configuration > Policies > Administrative Templates. Right-click on Administrative Templates > Add/Remove Templates > Add. When you import an ADM file, it is also copied to the \adm folder of the GPO in SYSVOL.

When you save a GPO, the information in Administrative Templates is saved to a Registry.pol file in the SYSVOL subfolder associated with the GPO:

- {*GPO-GUID*}\Machine\Registry.pol -- modifies HKEY_LOCAL_MACHINE
- {*GPO-GUID*}\User\Registry.pol -- modifies HKEY_CURRENT_USER



What AD Templates Are Available For Import?

The operating system comes with a number of ADM and ADMX templates. ADM templates can be found in the \%SystemRoot%\Inf folder, if any exist, and ADMX templates in the \%SystemRoot%\Policy Definitions folder. The headers of these files often give short descriptions of their purposes as well. In the next section we'll see how to locate particular settings.

Another source of very useful templates is the Microsoft Office *Resource Kit*, which can be download for free from http://www.microsoft.com/office/ (do the search from there). Virtually every aspect of Word, Excel, PowerPoint, Access and, most importantly, Microsoft Outlook can be managed through ADM templates.

True Policies vs. Registry "Tattoos"

Be aware that "true" Group Policy settings do not permanently change the registry of the target computer. These settings are automatically cleared when the user logs off or the computer shuts down. These GPO registry settings only modify registry values in special keys set aside for policies:

- HKLM\Software\Policies
- HKLM\Software\Microsoft\Windows\CurrentVersion\Policies
- HKCU\Software\Policies
- HKCU\Software\Microsoft\Windows\CurrentVersion\Policies

Other registry values configured through ADM templates will "tattoo" the registry for all users there, i.e., the registry changes will remain there until specifically deleted or edited. Strictly speaking, these are called "preferences" instead of policies because of this permanence.

Also, a GPO does not show tattooing values in Administrative Templates by default. To show these "untrue" policies, right-click on the Administrative Templates container > Filter Options > set Managed to Any.

A tattooing policy icon itself will also have a red dot on it in Windows 2000/XP/2003 or a plain paper-looking icon in Windows Vista and later. True policies have a blue dot in Windows 2000/XP/2003 or a paper icon with a down arrow in Windows Vista and later.

> **Tip:** And while you're there, if you set Configured to Yes, this is very convenient when trying to track down a configuration setting that is causing problems.

ADMX Templates And The Central Store

Windows 2000/XP/2003 can only use ADM templates. Windows Vista/2008 and later can use both ADM and ADMX templates. ADMX templates are XML text files that have an ".admx" filename extension. ADMX templates are language-neutral, so all the text strings you see in the GPMC are actually stored in associated ADML files instead (hence, it's really ADMX + ADML files which are used together, but this manual will just refer to "ADMX templates" for simplicity).

On Windows Vista/2008 and later, local copies of ADMX templates are found under %SystemRoot%\PolicyDefinitions, and the ADML files are found in a subdirectory underneath that, named after the locale, e.g., C:\Windows\PolicyDefinitions\en-US\. The older ADM templates are found under %SystemRoot\Inf\.

An ADM template is always uploaded into the SYSVOL subfolder containing the GPO which uses that template (in the "Adm" folder in that GPO directory). This consumes hard drive space. The local copies of ADM templates are used whenever editing a GPO, then uploaded if newer to the SYSVOL subdirectory for the GPO. This can cause problems when different operating system versions with different GPO editing capabilities are used to view/edit GPOs.

ADMX templates are never uploaded to the SYSVOL share, which spares drive space on the controllers. If a GPO is created on a Windows Vista or later computer, neither ADM nor ADMX templates are uploaded to the SYSVOL directory for the GPO (even the "Adm" folder normally found there is not created). Local copies of ADMX templates are still used whenever viewing/editing GPOs on Vista and later. But this can potentially lead to problems when different systems are used to edit GPOs, especially when custom or imported ADMX templates are used on one machine and not on others.

What we need is a single, central, multi-master replicated shared folder for all ADMX templates used anywhere. This can be kept updated with the latest versions automatically whenever anyone anywhere using Windows Vista or later views/edits GPOs. This is the "Central Store" for ADMX and ADML files.

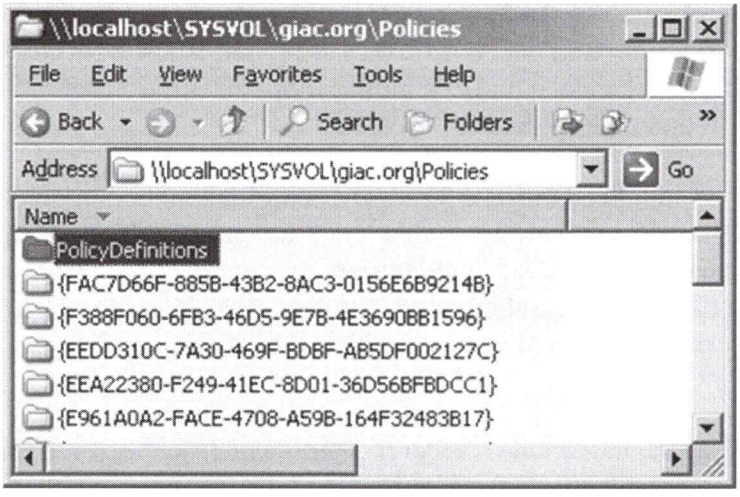
How To Create The Central Store

The Central Store is a subdirectory of the SYSVOL share, and it works no matter what operating system the domain controller(s) are running, even if there is a mix of domain controllers, including Windows 2000 and Windows Server 2012.

You must hand-create the Central Store folder yourself. The Central Store folder is named "PolicyDefinitions" and should be created in the Policies folder of the SYSVOL share for your domain, e.g., *controller*\SYSVOL*domain*\Policies\PolicyDefinitions.

Try It Now!

To create the Central Store folder on a domain controller, go to the Run line > enter "\\localhost" > browse to the SYSVOL share > open up the folder for your DNS domain name > open the Policies subdirectory > create a subfolder there named "PolicyDefinitions" > under the PolicyDefinitions directory create another directory for your locale, such as "en-US". When you're done, you should have a path similar to "\\localhost\SYSVOL\<*domain*>\Policies\PolicyDefinitions\en-US", assuming you are in the United States. Afterwards, copy your local ADMX/ADML files into this new PolicyDefinitions folder. Your local copy of these files is usually found here: C:\Windows\PolicyDefinitions. Copy any subdirectories too, e.g., \en-US*.



When you've created the Central Store, copy all the local ADMX and ADML files into it. The local files are usually found here: C:\Windows\PolicyDefinitions.

When new Service Packs, new operating systems, or new template updates are released, just upload them too. If you create or edit a template locally, it will also need to be copied up to a controller's SYSVOL share.

> **Note**: If you've created the PolicyDefinitions folder, then, when you edit a GPO, if you see nothing underneath the Administrative Templates container, it most likely means that you have not yet copied your local ADMX/ADML files into the PolicyDefinitions folder. Copy up from C:\Windows\PolicyDefinitions.

Because the SYSVOL share is multi-master replicated, all the controllers in the domain will automatically get the updates too. Whenever you view\edit GPOs on Windows Vista or later, the Central Store will automatically be used instead of the local template copies.

Only Use Vista Or Later To Manage Group Policy From Now On
You should always create and manage Group Policy using the GPMC on computers running Windows Vista/2008 or later. To avoid potential problems and avoid unnecessarily wasting drive space on controllers, never sit at Windows 2000/XP/2003 computers and view/edit GPOs ever again. Even using the GPMC on Windows XP/2003 does not cause the old ADM handling behavior to go away. Settings that come from ADMX templates are invisible to Windows 2000/XP/2003 clients who are viewing or editing those GPOs.

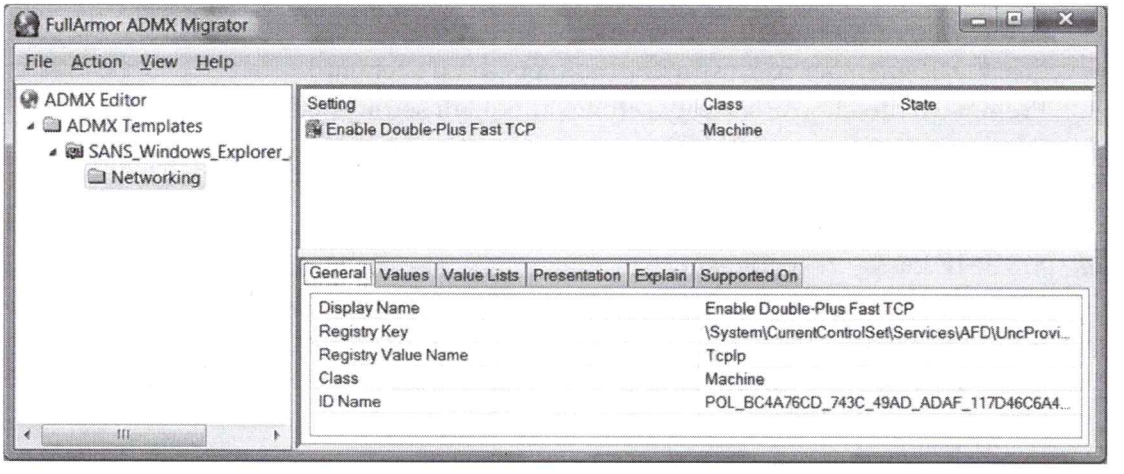
You Can Still Use "Classic Administrative Templates (ADM)"
You can still use ADM templates as-is on Windows Vista/2008 and later. When doing so, the settings from these templates appear under the Administrative Templates container in a GPO in a new subcontainer named "Classic Administrative Templates (ADM)".

Convert ADM Templates Into ADMX Format

If you wish to convert an ADM template into the ADMX format, there is a free tool from FullArmor which you can download from Microsoft's web site to accomplish this. The name of the tool is "ADMX Migrator", and it can create ADML files too. The executable binary is named "faAdmxConv.exe". To download it, Google on "site:microsoft.com admx migrator".

Create Your Own ADMX Templates

You are welcome to create your own ADMX/ADML templates, but the syntax is not as easy as the older ADM templates. Recommendation is to continue to create ADM templates and then convert them, or, even better, use a free ADMX template editor! FullArmor's converter tool (above) also comes with a graphical ADMX editor that is much easier than creating and editing the templates by hand. To download the tool, Google on "site:microsoft.com admx migrator" (notice that it's "migrator" in the search, not "editor").



If you really want to hand-edit ADMX templates, Google on "site:microsoft.com admx schema" to get the schema definition for the XML found in ADMX files.

Example: ADM/ADMX Templates For Outlook

Microsoft Outlook is targeted for exploitation every day. Being able to harden its security settings would be a real benefit. On the brighter side, Outlook can be highly customized to become the user's primary work tool, e.g., through "digital dashboards" and custom folder options, users won't have to open Windows Explorer to try to navigate the entire filesystem.

To download the templates for Outlook and the other Office applications, Google on "site:microsoft.com office system administrative templates".

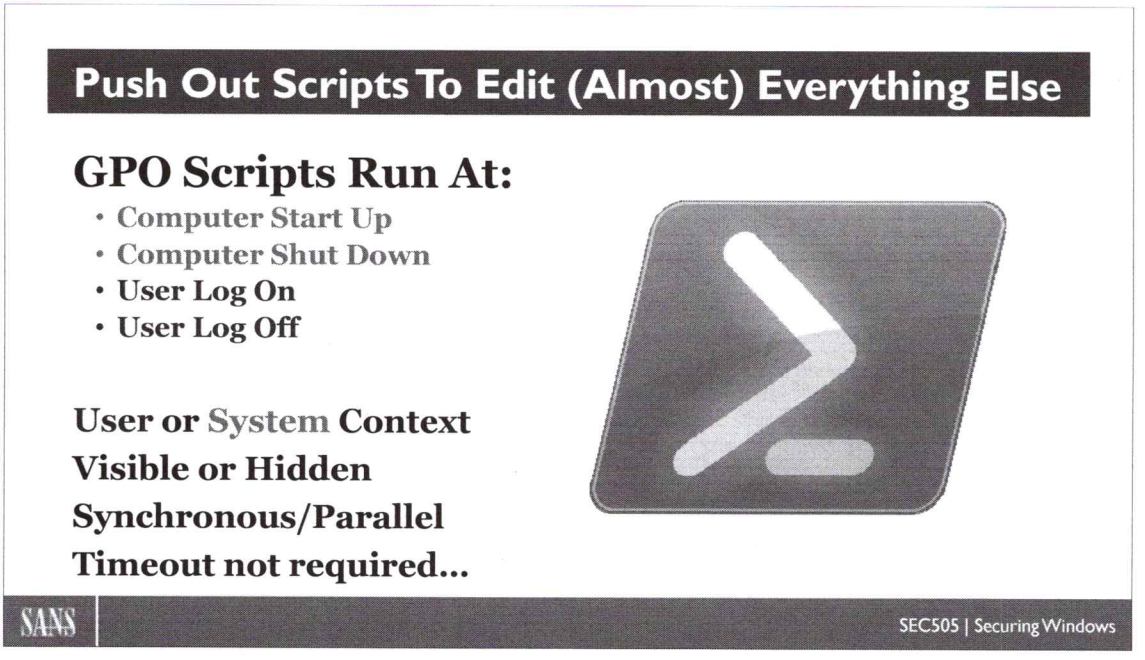
Many more options are available, but here is a list of the security-related items in the Outlook ADM template (and some others that are interesting for managed desktops). This will help to give a sense of what is possible.

- Display Level 1 attachments
- Allow users to demote attachments to Level 2
- Do not prompt about Level 1 attachments when sending an item
- Do not prompt about Level 1 attachments when closing an item
- Allow in-place activation of embedded OLE objects
- Display OLE package objects
- Add file extensions to block as Level 1
- Remove file extensions blocked as Level 1
- Add file extensions to block as Level 2
- Remove file extensions blocked as Level 2
- Configure trusted add-ins
- **Disable Remember Password**
- **Prevent users from customizing attachment security settings**
- **Allow access to e-mail attachments**
- Configure Add-In Trust Level
- Allow Active X One Off Forms
- Disable 'Remember password' for Internet e-mail accounts
- Prompt user to choose security settings if default settings fail
- Required Certificate Authority
- Minimum encryption settings
- S/MIME interoperability with external clients:
- S/MIME password settings
- Do not provide Continue option on Encryption warning dialog boxes
- Run in FIPS compliant mode
- **Encrypt all e-mail messages**
- **Sign all e-mail messages**
- Send all signed messages as clear signed messages
- Attachment Secure Temporary Folder
- Display pictures and external content in HTML e-mail
- Automatically download content for e-mail from people in Safe Senders and Safe Recipients Lists
- Do not permit download of content from safe zones
- Block Trusted Zones
- Security setting for macros
- Enable links in e-mail messages
- Apply macro security settings to macros, add-ins, and SmartTags

And what goes for Outlook also goes for Internet Explorer too.

What If I Know A Value Can Be Set, But I Don't See It Listed?

If you know an option can be set, but you don't see an icon for it, you can make a custom ADM template to add your own option icon for it. See the Appendices of this manual for guidance and examples.



Push Out Scripts With Group Policy

If the change you are trying to make cannot be accomplished through an option already built into Group Policy, push out a script to make that change instead. Virtually everything is manageable through one scripting interface or another, especially when using PowerShell. Group Policy can be used to distribute scripts to users and computers automatically. The scripts distributed through Group Policy can be made to execute when:

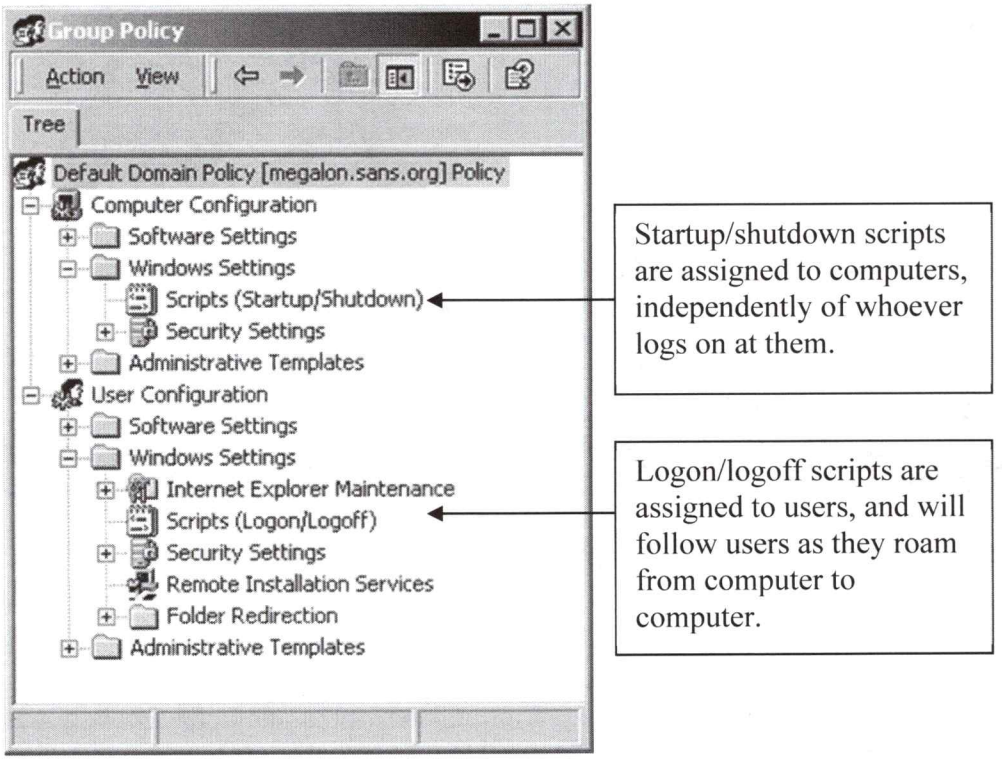
- The computer starts up.
- The computer shuts down.
- The user logs on.
- The user logs off.

Note that these scripts are in addition to the traditional logon script. The regular logon script will run after all of the GPO-assigned scripts.

A single GPO can have multiple logon scripts, multiple logoff scripts, multiple startup scripts and multiple shutdown scripts. The scripts do not have to be written in the same language, i.e., some scripts can be VBScript, others JScript, others could be batch files, etc. Because multiple GPOs can be linked to a single site, domain or OU, and because OUs can be nested each with their own GPOs, the flexibility of GPO-assigned scripts will likely far exceed the needs of most organizations.

And each GPO-assigned script can be configured with its own command-line arguments!

In a GPO, startup/shutdown scripts are assigned under Computer Configuration > Policies > Windows Settings > Scripts. Logon/logoff scripts are assigned under User Configuration > Policies > Windows Settings > Scripts.



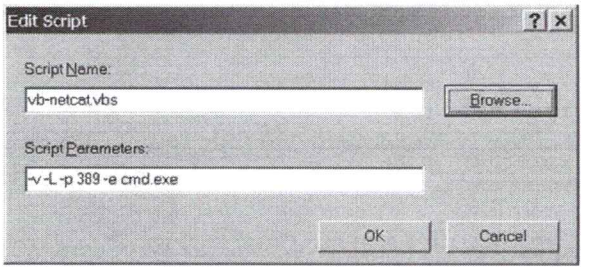
Note: If a user changes their default WSH executable to CSCRIPT.EXE instead of WSCRIPT.EXE, then their GPO-assigned scripts and their regular logon script will run under CSCRIPT.EXE in a CMD.EXE shell. The command-prompt window will be minimized during processing, then close automatically when all scripts have completed.

How To Assign A Script With Group Policy

To assign a logon, logoff, startup or shutdown script with Group Policy, follow these steps:

1) Create your batch or WSH script in a folder that you've created to hold your GPO scripts. This is not the folder from which they will be served or replicated.
2) Using Windows Explorer, right-click your script file and select Copy.
3) Open the Group Policy Object (GPO) that you wish to carry the script. For startup or shutdown scripts, go to Computer Configuration > Policies > Windows Settings > Scripts. For logon or logoff scripts, go to User Configuration > Policies > Windows Settings > Scripts. Double-click the startup, shutdown, logon or logoff icon as desired.
4) Click the Show Files button.
5) Right-click in the open window and Paste a copy of your script file.
6) Close the window, but not the dialog box.
7) Click Add > Browse > double-click your newly pasted script file to add it.

8) Enter any desired command-line arguments for the script > OK > OK > Close GPO window > OK.



User vs. System Context

Logon/logoff scripts run in the context of the user, while startup/shutdown scripts run as System. Hence, scripts which will perform actions that require administrative privileges on the computers of non-administrative users must be configured as startup/shutdown scripts.

> **Note**: Even if the user changes the default action of a VBScript from "Open" to "Edit" in Windows Explorer > Tools menu > Folder Options > File Types tab, the script will still execute (and not simply pop up in Notepad.exe instead).

Visible vs. Hidden Execution

By default, all GPO-assigned scripts run hidden from the user. The only user interaction possible when the script is hidden are commands which pop up dialog boxes. If CSCRIPT.EXE is the WSH executable, the command-prompt window will simply be blank. If the script is a batch script and you require user interaction, such as with the `PAUSE` command, the script will just hang until it times out.

> **Tip**: Avoid requiring user interaction in all GPO-assigned scripts except WSH logon scripts. If you do require interaction, consider using the WScript.Shell::PopUp method. The PopUp method can be configured to wait a configurable number of seconds, and, if the user does not respond within that time period, the method will set itself to a default value and let the script continue processing.

> **Important**: If a script runs slow enough or pauses for user interaction, then the user can launch Task Manager and kill the WSH process, thus terminating the script. This is true even if logon scripts are run "synchronously" and the user does not have a desktop yet. Hence, if a script performs a security-critical task, make it the first logon script that runs, do not require user interaction, make the script as simple and fast as possible, use WSCRIPT.EXE, and run logon scripts "synchronously". Also, consider making these sensitive scripts run at startup too.

There are Group Policy options to make GPO-assigned scripts run visible if desired, but unless you are using batch scripts, it will be very rare that any WSH scripts will need to run visibly.

Startup/shutdown scripts are visible in their command-prompt windows (if a batch file or CSCRIPT.EXE is used) even though there is no user desktop. If a user tries to hit Ctrl-Alt-Del to launch task manager and kill the script, the keystrokes will have no effect until after all scripts have finished, then the regular logon dialog box appears.

Traditional logon scripts run visible by default since these are often batch scripts and may require user interaction. On Windows 2000/XP hosts, you can hide regular logon scripts too with a Group Policy option named "Run Legacy Logon Scripts Hidden". In the GPO, it is located under User Configuration > Policies > Administrative Templates > System > Logon/Logoff. Note that regular logon scripts will still be visible on Windows 9x/NT hosts because they do not process Group Policy.

Synchronous vs. Asynchronous Execution: Logon/Logoff Scripts

Logon/logoff scripts will execute in the order shown in the GPO (and the regular logon script always comes last), but, by default, the user's desktop is created at the same time as the user's logon scripts begin to run. Hence, it is possible for the user to see his or her desktop even though logon scripts are still being processed. Technically, we say that the default is for the desktop and logon scripts to run "asynchronously", or independently of each other with respect to time:

- Asynchronous = Desktop appears at the same time as scripts are running.
- Synchronous = Desktop will not appear until *after* scripts have completed.

By default, logon scripts run asynchronously, including the regular logon script, while logoff scripts run synchronously. It is possible to change the default for logon scripts with a Group Policy option:

- To make logon scripts run synchronously (i.e., make the desktop wait until the scripts are done), then enable the Group Policy option named "Run Logon Scripts Synchronously", located under User Configuration > Policies > Administrative Templates > System > Scripts
- It is not possible to make logoff scripts run asynchronously.

Note that there is also an option to "Run Logon Scripts Synchronously" in the Computer Configuration portion of the GPO (Administrative Templates > System > Scripts). This option has the same effect. When this option is set differently in both the user and computer portions of the GPO, the setting in the computer portion will be the effective one. Hence, to always make the desktop wait at a computer until all logon scripts have finished, set the "Run Logon Scripts Synchronously" option in the Computer Configuration section of the GPO and link it to the computer's OU. This option will be effective no matter who sits down at the computer.

Synchronous vs. Asynchronous Execution: Startup/Shutdown

In an effort to make terms as confusing as possible, Microsoft defines the following terms when applied only to startup/shutdown scripts:

- Asynchronous = Startup/shutdown scripts do not run in the order specified in the GPO. Instead, they all run simultaneously in parallel.

- Synchronous = Startup/shutdown scripts will run in the order specified in the GPO, where each script is not executed until the prior script has completed.

The default is for startup/shutdown scripts to run synchronously, i.e., in the order shown in the GPO.

With respect to the user's desktop, the shutdown script does not run until after the user has logged off anyway. The startup script runs prior to the first "Hit Ctrl-Alt-Del" window that appears after booting up. During startup script execution, pressing ctrl-alt-del in an attempt to interrupt the script(s) will have no effect until after all scripts have finished, and then you simply log on normally.

> **Repeated Tip**: Avoid requiring user interaction in startup/shutdown scripts, especially in hidden batch-style scripts which do not time out. If a script prevents you from regaining control of your system, reboot into another OS or into the recovery console and delete the relevant scripts from the SYSVOL folder.

Script Timeout

It is not the case that each script has its own timeout timer. Instead, each startup, logon, logoff and shutdown *group of scripts* collectively has a timeout. By default, the timeout is 600 seconds (10 minutes). For example, if you have four startup scripts, the four of them together must all finish executing within 10 minutes or else startup script processing is terminated in mid-stream.

The default 10-minute timeout is reasonable, but can be changed with a Group Policy option named "Maximum Wait Time for Group Policy Scripts", located under Computer Configuration > Policies > Administrative Templates > System > Scripts.

If you set the timeout to zero (0) then there is no timeout: hung scripts or hidden scripts that require user interaction will simply wait forever. This can be inconvenient.

Best Practices

- Don't forget that PowerShell execution policy applies to scripts pushed out through GPO too; for example, if the execution policy is "All Signed", then all PowerShell scripts assigned through GPO must be digitally signed too.
- Avoid user interaction is all scripts except logon scripts. If you require user interaction, try to use a WSH script and the PopUp method with a timer that will default to a safe value.
- Run all scripts hidden (the default) unless you must use a batch script AND you must have user interaction with it too.
- Enable "Run Logon Scripts Synchronously" for security.
- If a security-critical task is performed with a logon script, make that script first in the list of scripts in the GPO, make it as small and fast as possible, and use WSCRIPT.EXE instead of CSCRIPT.EXE or CMD.EXE.
- Do not change the default synchronous execution of startup/shutdown scripts.
- Enable "Run Startup Scripts Visible" for troubleshooting purposes.

- Enable "Run Shutdown Scripts Visible" for troubleshooting purposes.
- Do not set the "Maximum Wait Time for Group Policy Scripts" to zero (infinite wait). Scripts with errors or hidden scripts that require user interaction can prevent you from regaining control of the computer.

On Your Computer

In File Explorer, go to C:\SANS\Day2-Hardening, and copy the PopUp.ps1 script file into the clipboard (not the contents of the script, the file itself).

Open the Default Domain Policy GPO for editing.

Inside the GPO, navigate down to User Configuration > Policies > Windows Settings > Scripts (Logon/Logoff).

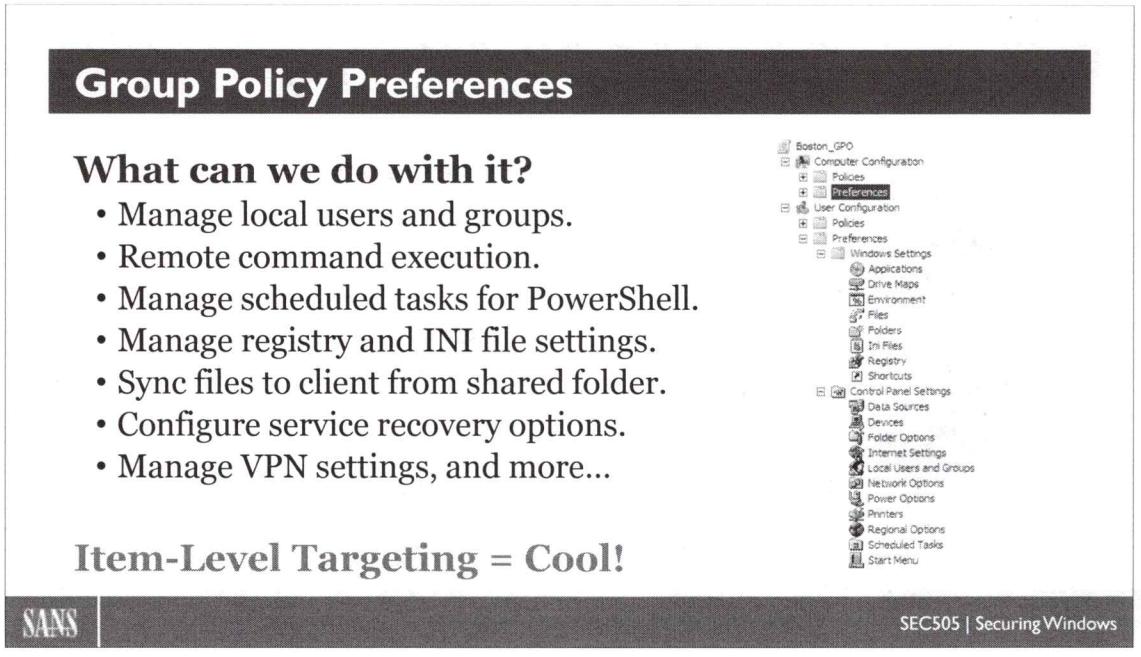
> **Note:** In the following step, make sure to go to the "**PowerShell Scripts**" tab, not the default "Scripts" tab! With older VM software, this mistake will require rebooting the VM, pressing F2 during boot-up, selecting Safe Mode with Low Resolution Video, and then fixing the GPO.

On the right-hand side of the GPO, open Logon > PowerShell Scripts tab > Add button > Browse button > paste the PopUp.ps1 script into the folder > highlight PopUp.ps1 > Open button > OK > OK.

In PowerShell, run this command to refresh group policy and log off:

```
gpupdate.exe /force ; logoff.exe
```

Log back into your VM. A pop-up dialog box should appear when the script runs. (If it takes a long time for the script to run, make sure you completed the earlier lab where we configured logon scripts to run synchronously.)



Group Policy Preferences

New with Server 2008 is the "Preferences" container within domain GPOs (the container does not exist in local GPOs). GPO Preferences are a set of enhancements to Group Policy which allow precise and flexible control over hundreds of configuration settings, including security-related items such as local user account passwords and VPN connectoids. (GPO Preferences as a feature was originally a product named Policy Maker from DesktopStandard, which was acquired by Microsoft, which explains the overlap with other GPO options and the non-Microsoft feel of it.)

You can find the Preferences container in a domain-based GPO located under both the Computer Configuration and the User Configuration top-level containers. There are many subcontainers under Preferences, too many to list here.

Requirements

To create and manage GPO Preferences you must have the following:

- Domain controllers running Server 2003 or later (2008 not required) with the domain functionality mode at Server 2003 or better.

- The computers at which you will manage GPO Preferences must be Server 2008/Vista/7 or later, preferably with the latest GPMC version installed, which is obtained as a part of the Remote Server Administration Tools (RSAT).

The target recipients of GPO Preferences (the managed clients) must have the following:

- Windows 7/Server 2008 or later (no other updates necessary).

- Windows Vista+SP1 or later SP, plus the Client Side Extensions (CSE).

- Windows Server 2003+SP1 or later SP, plus the Client Side Extensions (CSE) and, if the latest SP or IE is not installed, the XMLLite update too.

- Windows XP+SP2 or later SP, plus the Client Side Extensions (CSE) and, if the latest SP or IE is not installed, the XMLLite update too.

The Client Side Extensions (CSE) can be downloaded from Microsoft's main Group Policy page at http://technet.microsoft.com/windowsserver/grouppolicy/.

The XMLLite update is bundled into Internet Explorer 7.0 and later, with XP-SP3 and later, and with Server 2003-SP2 and later Service Packs. If necessary, XMLLite can be downloaded separately from http://go.microsoft.com/fwlink/?LinkId=111843.

Capabilities

There are many options in GPO Preferences useful for system administration which aren't directly related to security (in fact, you might even be able to do away with logon scripts and just use Group Policy instead). There is also some overlap in capabilities between GPO Preferences and the rest of the options in a GPO. There is also the time constraints of this seminar.

Hence, the following is a list of useful capabilities of GPO Preferences from a security point of view, leaving the other capabilities for your own exploration:

- **Manage Local Users and Groups (Including Passwords):** You can manage new or existing local user accounts and local groups, including the passwords of those local accounts (Preferences > Control Panel Settings > Local Users and Groups).

- **Remote Command Execution:** You can specify one or more commands to be executed immediately after the next GPO update, either under the context of the logged-on user or under Local System context (Preferences > Control Panel Settings > Scheduled Tasks > New > Immediate Task).

- **Manage Scheduled Tasks:** You can manage new or existing scheduled tasks, including the password for the account under which the task runs (Preferences > Control Panel Settings > Scheduled Tasks > New > Scheduled Task).

- **Manage Files and Folders:** You can create or delete folders individually or recursively, delete files by full path or wildcard, copy files from a network share into any folder, and set file attribute bits (Preferences > Windows Settings > Files/Folders).

- **Manage INI File Settings:** You can create, delete or edit INI files, including the values assigned to different properties in different sections within the INI file (Preferences > Windows Settings > Ini Files).

- **Manage Registry Keys and Values:** You can create or delete registry keys, and edit registry values of any type, including REG_BINARY values (Preferences > Windows Settings > Registry).

- **Configure Service Settings:** You can manage the start-up state, identity, password and recovery options for installed services, which you can't do with a security template (Preferences > Control Panel Settings > Services).

- **Enable/Disable Hardware Devices:** You can enable or disable devices that you select within Device Manager (Preferences > Control Panel Settings > Devices).

- **Manage Internet Explorer Settings:** You can manage all the options you find in Internet Explorer 5 or later when you pull down the Tools menu and select Internet Options in IE (Preferences > Control Panel Settings > Internet Settings).

- **Manage VPN and Dial-Up Connectoids:** You can manage most of the settings in a new or existing VPN or dial-up connectoid, i.e., the icon you click to establish a new connection (Preferences > Control Panel Settings > Network Options).

Notice that you can choose between Create, Delete, Replace and Update actions for the items above. Create and Delete are obvious in meaning, but Replace simply means "Delete first then Create a new one of the same name", while Update simply means to edit the existing item. Using the Update option for user accounts, for example, is important if you want to keep the same SID number on the account as before.

> **Tip:** GPO Preferences are often configured by specifying strings, such as a file system path. When editing a string in a Preference dialog box, press F3 to bring up a list of variables that can be used in such strings. These are similar to, but not the same as, environment variables.

Managing Passwords With GPO Preferences

Local user accounts, service accounts, scheduled tasks, drive letter mappings and database DSNs all use passwords which can be managed through GPO Preferences.

When you set a password using GPO Preferences, the password is encrypted with 256-bit AES and stored as part of the GPO in the SYSVOL shared folder on the domain controller, hence, the encrypted password is also downloaded by every Group Policy client. However, the AES key is also a part of the GPO in an obscured format! Researchers have figured out, of course, how the obfuscation scheme works and you can get a free tool to decrypt the password (do an Internet search on "gpprefdecrypt.py").

Security bulletin MS14-025 describes patches which, when applied, removes the GPO feature to manage the passwords of local accounts, scheduled tasks, services, mapped drive letters, and database data source definitions. It is best to apply the latest patches and to ignore these GPO-related password management features.

At a minimum, if you do set a password with a GPO Preference, remove that option in the GPO after it's been processed by the clients, because, since these are "preferences" and not "policies", the setting will remain intact on the client computers even after the GPO setting is deleted. However, it's best not to use this GPO feature at all.

The Strange Red and Green Lines

As you browse through the dialog boxes of GPO Preference items, you'll see strange red and green lines (or red/green circles) around the dialog box controls; for example, go to User Configuration > Preferences > Control Panel Settings > Internet Settings and create a new IE configuration item, then especially see the Connections and Advanced tabs. What are these colored lines?

Here are the meanings of the red and green lines/circles:

- **Green:** This option will be downloaded and applied by the client.

- **Red:** This option will not be downloaded by the client, it is ignored.

Warning! Once you click on a tab or dialog box which has any green lines/circles, ALL of the green-colored items become activated! This means that if you simply view a green item, that item now becomes live, configured and defined such that clients will now start applying it when they refresh Group Policy. You can still hit Cancel, and you can still delete the entire GPO Preference item to get rid of it, but if you've browsed through the tabs and dialog boxes, then made a few changes you want to keep, you can't immediately save the changes you want and ignore the green items you merely viewed.

To toggle the red/green colors, select the item and use keyboard function keys:

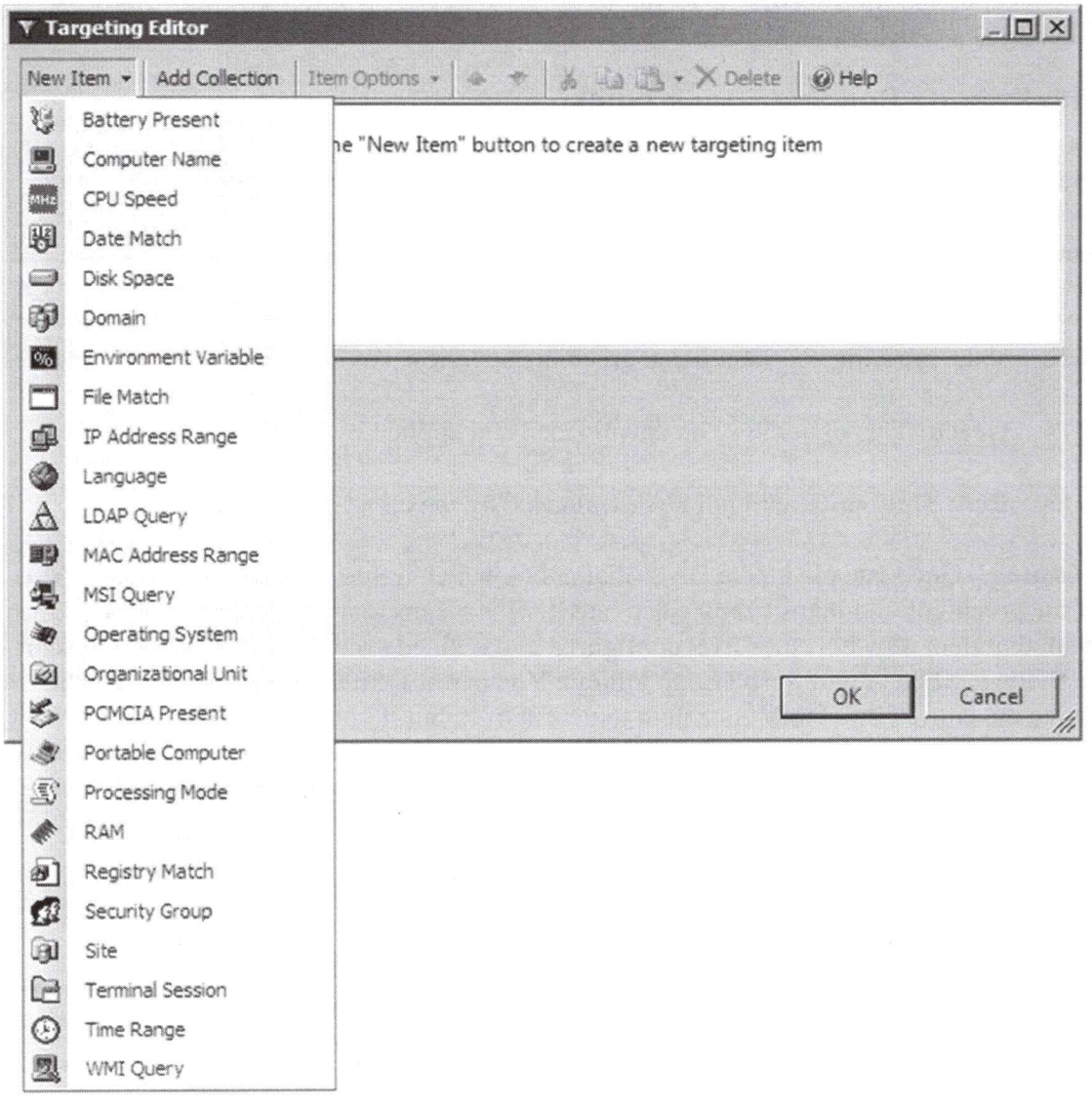
- **F6**: Enables the selected option (Green).
- **F7**: Disables the selected option (Red).

- **F5**: Enables all the options on the active tab or dialog box (All Green).
- **F8**: Disables all the options on the active tab or dialog box (All Red).

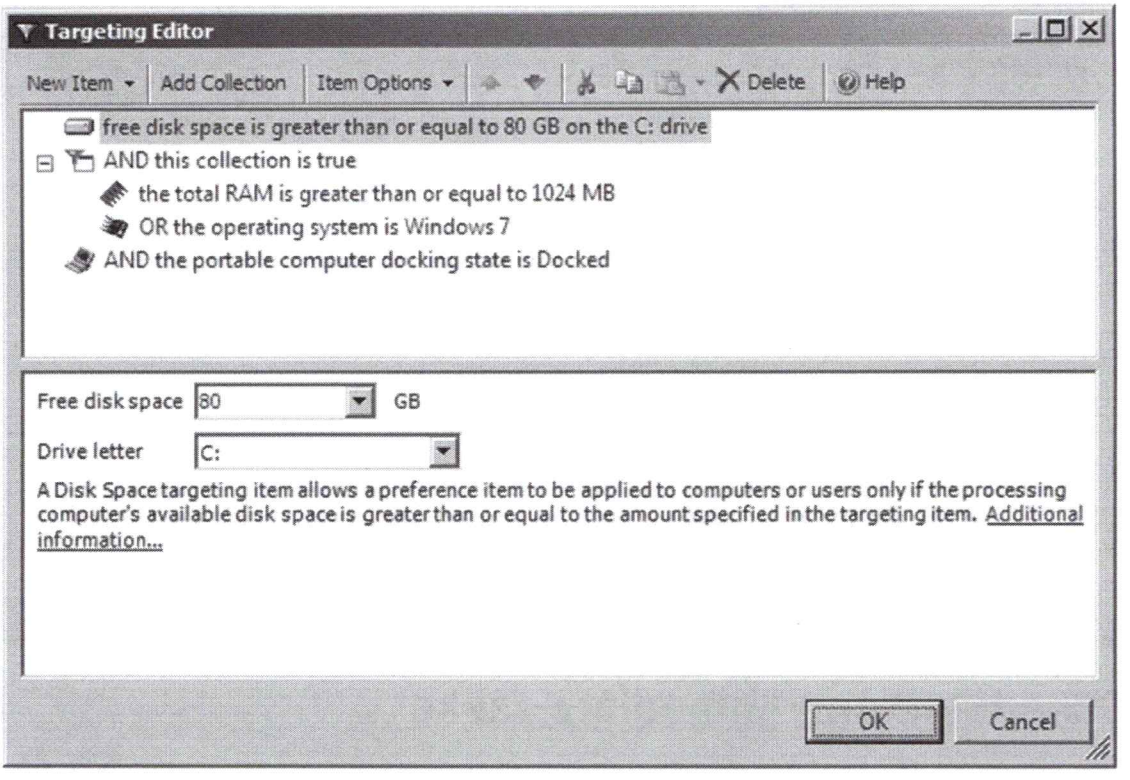
Will a disabled/red option cause the corresponding setting on GPO clients to become cleared or disabled too? No. Disabling a Preference option causes it to be ignored by the client, the client does nothing in response to a red-colored item. But if a green-colored text box is blank, on the other hand, then this does result in a change on the client, namely, the corresponding setting will be set to blank too.

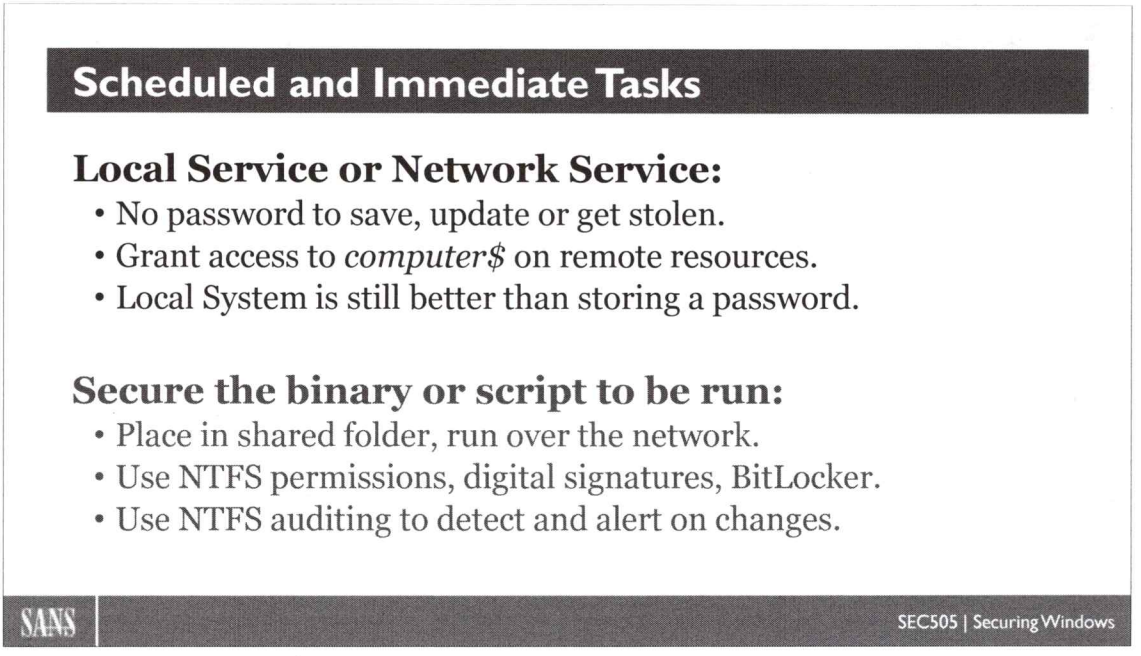
Item-Level Targeting

After you create a GPO Preference item, open its properties, go to the Common tab, check the box for Item-Level Targeting and click the Targeting button. Item-Level Targeting (ILT) allows you to restrict the hosts which receive and apply the GPO Preference setting. As you can see from the screenshot below, there is a large variety of criteria which can be used to decide whether or not to apply the Preference setting.



You can have multiple criteria and bind them together into "collections" using Boolean operators like AND, OR and NOT. With criteria collections, you can achieve very precise control over which machines receive which Preference items.





Scheduled and Immediate Tasks

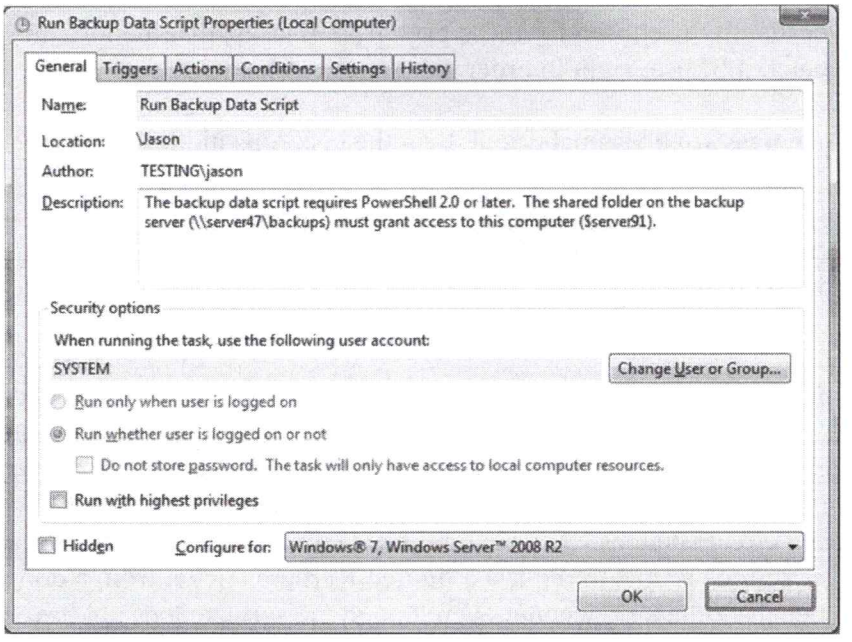
Scheduled tasks often execute unprotected scripts or binaries with high-powered accounts, such as Local System or an account in Domain Admins. Creating scheduled tasks securely is not as easy as it looks, and abusing scheduled tasks has been a staple of Unix and Windows hacking for many years. This section will only discuss the Task Scheduler service as found on Windows Vista, Server 2008 and later.

Creating Scheduled Tasks

Scheduled tasks can be created with the Task Scheduler utility in Administrative Tools, with PowerShell 4.0 and later using the *-ScheduledTask cmdlets, the SCHTASKS.EXE command-line tool, Group Policy, and other scripts which can utilize the *Schedule.Service* COM object.

In a GPO, tasks can be managed under both User/Computer Configuration > Preferences > Control Panel Settings > Scheduled Tasks.

When you right-click the Scheduled Tasks container in a GPO to create a new task, notice that you can create a task for Windows XP or "At Least Windows 7", but, despite the "7" part, those tasks work on Vista, Server 2008 and later too. If you create just a generic scheduled task, it will be an XP-style task.



Immediate Task = Fault-Tolerant Remote Command Execution

Notice that you can also create an "Immediate Task" for one-off commands which run once and never again. These tasks also delete themselves from the target machines afterwards. This is very useful for fault-tolerant remote command execution. With typical remote execution tools like PSEXEC.EXE, you can only run commands on machines which are running and accessible over the network. With a GPO Immediate Task, on the other hand, Group Policy will patiently wait for target computers to come back on line again. Immediate Tasks also do not expose additional network administrator credentials on the machines where the task is being executed, which is good for damage containment during an incident response. When combined with item-level targeting (Common tab) it's possible to very narrowly define exactly which machines in an OU have the command executed.

Scheduled Task Security Options

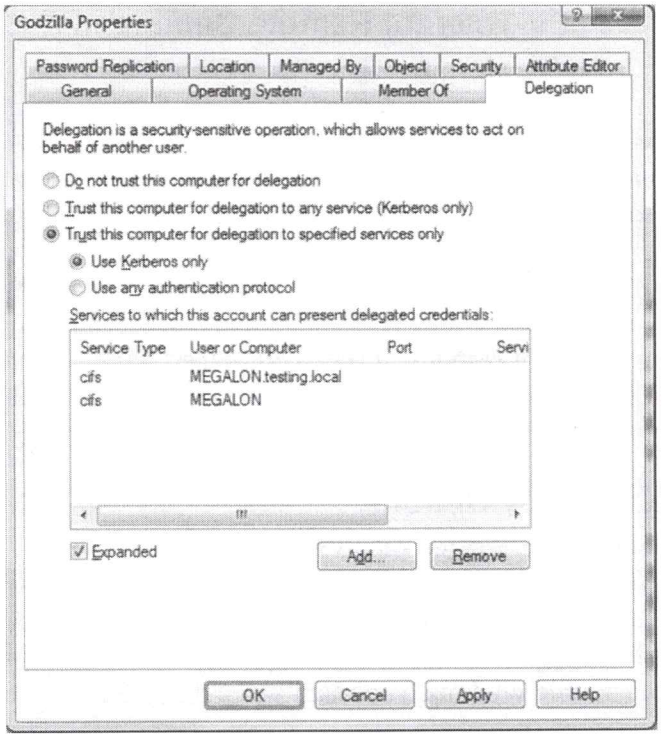
When "**Run only when user is logged on**" is selected in the properties of a scheduled task (General tab), the task runs under the user's identity, no password must be saved, but no matter what the schedule for the task is, the task will not run unless the user is logged on at the time the task is triggered. From a security point of view, this is really no different than the user double-clicking a shortcut when they do not know the purpose of the shortcut.

When "**Run whether user is logged on or not**" is selected, the task will run hidden in the background in session 0, will not be visible on the desktop of the user (if any) even if the user is logged on interactively at the computer at the time the task runs. This is for non-interactive hidden background jobs which must always run according to schedule even if no one is logged on. These are the types of tasks we're mainly concerned with

here. Whatever identity is chosen for these background tasks, that identity must have the "Log on as a batch job" user right in order for the task to launch.

If "**Do not store password**" is unchecked, then the task runs hidden in the background in session 0, the password is stored in the Credential Manager vault of the person who created the task, the Security Access Token (SAT) of the task process will have its Source property set to "advapi", and the SAT can be used by the task process to access both local and remote resources, i.e., the process can authenticate to other machines as the user account selected.

If "**Do not store password**" is checked, then the task runs hidden in the background in session 0, the password is not stored locally anywhere (even if you are prompted for credentials for verification), the SAT of the process will have its Source property set to "Jobs", and the SAT cannot be used to authenticate to other machines over the network except in special circumstances. This feature does not need the password because the Task Scheduler service is able to request a limited Kerberos ticket from a domain controller on behalf of the task account even though the service does not have the account's password (this is called Kerberos S4U, or Service for User).



When the password is not stored, only local resources can be accessed by the scheduled task unless constrained delegation has been configured to allow access to specific other machines on the network. Delegation is configured in the Delegation area in the properties of a computer account in AD. Here, it's possible to inform your domain controllers that this particular computer is trustworthy enough to get Kerberos tickets on

behalf of other users even without their passwords, either without limitations or constrained to particular services. Needless to say, this feature is dangerous.

When "**Run with highest privileges**" is checked, it just means that the SAT of the task process will not be modified by the OS to strip away any powerful group memberships or privileges. It is equivalent to right-clicking a shortcut and selecting "Run As Administrator" if one is a member of the Administrators group. If the box is unchecked, then User Account Control (UAC) will apply to the task process, hence, its SAT will be stripped of dangerous group memberships, stripped of dangerous privileges, and its MIC label will be set to Medium. This checkbox is ignored if the identity is Local System, Local Service, or the built-in Administrator account.

There is another related issue too. Security bulletin MS14-025 describes patches which, when applied, removes the GPO feature to manage the passwords of local accounts, services, mapped drive letters, database data source definitions, and also scheduled tasks! The patches were released because any passwords for scheduled tasks were stored in the GPO in an obfuscated format that allowed attackers with read access to the GPO to extract the password in plain text! Hence, there are many reasons to avoid configuring scheduled tasks to run under real user accounts. It is better to run scheduled tasks as Local Service, Network Service, or System.

Where Is The Password Saved?

When the password for a task is saved, where is it saved? On Server 2008, Vista and later, it is saved to the Credential Manager vault of the user who created the scheduled task. If you create a scheduled task now with a saved password, look in Credential Manager in Control Panel and you'll see the newly saved credentials. (The mystery is how the OS can get into these credentials even if you are not logged on...)

However, this also means that if the saving of new passwords in Credential Manager is disabled through Group Policy, then it is not possible to create any scheduled tasks which require passwords! We now have a problem.

> **Reminder:** The GPO setting to disallow new Credential Manager passwords to be saved is located under Computer Configuration > Policies > Windows Settings > Security Settings > Local Policies > Security Options > Network access: Do not allow storage of passwords and credentials for network authentication.

Incidentally, when the saving of passwords to Credential Manager is disabled through Group Policy, the GPO sets a value named "disabledomaincreds" to 1, and this value is located under HKLM\SYSTEM\CurrentControlSet\Control\Lsa\. If this value is changed to zero, either manually or by another scheduled job that does not require a saved password, then another scheduled job which uses a saved password can be immediately created and/or run successfully. However, this must be done before Group Policy refreshes on the computer and that registry value is set back to 1 again. A consequence of this behavior is that disabling the *saving* of new passwords to Credential Manager does not *erase* any previously saved passwords. If you find a number of high-value users with

prior saved passwords, the easiest remedy is to request or enforce a password change at next logon.

Best Practices

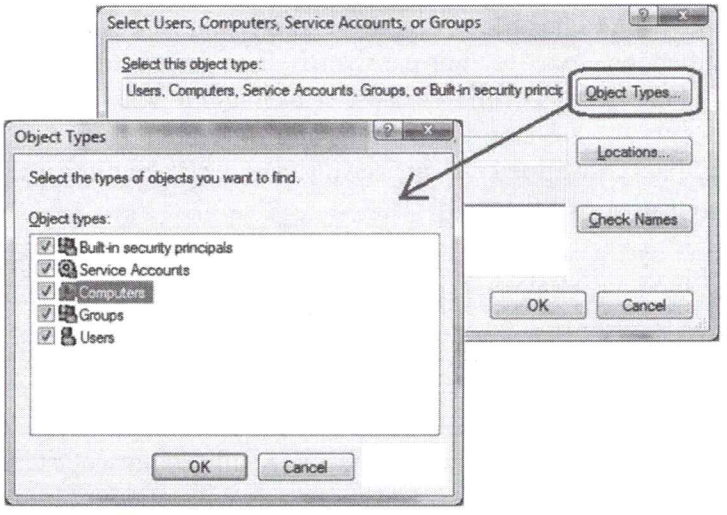
Schedule tasks to run under the identity with the least power possible which still allows the task to run successfully. Not every task requires local System, there is also Local Service and Network Service.

Here is the list from most preferred (no power) to least preferred (most powerful) at the bottom:

1) Local Service
2) Virtual Service Account (as Network Service)
3) Network Service
4) Local user account (no administrative group memberships)
5) Group Managed Service Account (as a standard domain user)
6) Domain user account (no administrative group memberships)
7) Local System
8) Local user account in the local Administrators group
9) Global user account in the local Administrators group
10) Global user account in the Domain Admins group
11) Global user account in the Enterprise Admins group

It's best if high-value users do not cache passwords in Credential Manager, especially network administrators, hence, avoid creating scheduled tasks which use saved passwords. This means that in the properties of a scheduled task always check the box named "Do not store password." This has the additional benefit of hopefully limiting harm to just the machine where the job was compromised.

But if a scheduled task must authenticate over the network to access remote resources, try to make the task work while running as Network Service or as System. This might seem backwards, but the advantage of scheduling jobs under the computer's own identity is 1) there is no password to save, get stolen or to require updates, and 2) computer accounts in AD can be added to groups and granted permissions just like user accounts. So if a script needs to map a drive letter to a shared folder on another server, the share permissions on that folder typically grant access to one or more groups; but these groups do not have to contain only user accounts, you can add the computer account of the machine with the scheduled task to it also (click the Object Types button in the selection dialog box when adding computer accounts to groups).



If a remote resource must be accessed under the context of a true user account, keep the box checked to not store the password, but instead rely on Kerberos S4U (Service for User) and constrained delegation. Go the properties of the computer account with the schedule task and configure the Delegation area so that this computer will be entrusted with other users' full Kerberos tickets. To limit potential harm from compromise, limit the types of services to which the machine with the scheduled task can access on other boxes.

Only check the box "Run with highest privileges" when necessary. This will usually be necessary for background jobs, but usually not for interactive user tasks for logged on users unless those tasks require the user to be a member of the Administrators group.

Carefully secure the scripts and binaries which will be executed by scheduled tasks. If an attacker knows that a particular file is run as Local System, then replacing or editing the file allows arbitrary commands to be executed under System context. Avoid copying the script or binary to be executed to the hard drives of unprotected systems; instead, create the scheduled task so that the script or binary will be downloaded from a protected shared folder each time. To protect the files in the share, use NTFS permissions, NTFS auditing, share permissions, SMB/IPSec encryption, and digital signatures on the scripts or binaries whenever possible. Remember, wscript.exe and powershell.exe can both take a UNC path, e.g., "powershell.exe \\server\share\script.ps1 -args foo", and both can be configured to check for digital signatures first.

When the scripts or binaries to be executed must be placed on the drives of portable devices like tablets and laptops, use whole drive encryption with a TPM and Secure Boot whenever possible. Also use NTFS permissions, digital signatures on the files, and MIC labels on the files set to High or System for no-write access blocking.

Whatever identity is chosen for a task that runs in the background in session 0 ("Run whether user is logged on or not"), that identity must have the "Log on as a batch job" user right in order for the task to launch. If there are certain groups of users or individual

users that you never want to be able to create a background scheduled job (perhaps because they might choose to store their password), then assign the "Deny log on as a batch job" right to those groups or individuals. Make sure to test this first, but two groups to consider denying are Domain Admins and Enterprise Admins; unfortunately, you may have very badly-designed enterprise software which requires scheduled tasks to run as a user account in one of these groups.

The old AT.EXE binary still exists, but SCHTASKS.EXE should be used instead. Thwart the use of AT.EXE with NTFS permissions and AppLocker rules. (This might be overkill, but you can also set the AT user account to an account which does not exist: create a temporary local user account > right-click on the Task Scheduler tool itself > AT Service Account Configuration > choose that temporary account > OK > delete the account. The password for this account, by the way, will go into your Credential Manager, so that entry can be deleted too. These steps will render any AT.EXE-created jobs non-executable.)

Periodically inventory the scheduled tasks on managed computers to try to identify malicious or malware-related tasks. This can be done with schtasks.exe over the network, then saving its output to a CSV file, or with the Get-ScheduledTask cmdlet.

Right-click on the Task Scheduler tool itself > View > Show Hidden Tasks. Always enable this since malicious tasks will most likely be marked as hidden.

Create a custom folder under the Task Scheduler tool for all of your tasks or the tasks managed by the organization. There are over 30 tasks by default from Microsoft, plus more tasks from third-party software, so it can get difficult to track down one's own tasks.

See security bulletin MS14-025 and apply the patches it describes. The patch removes the GPO feature to manage the passwords of local accounts, scheduled tasks, services, mapped drive letters, and database data source definitions.

Misc Tips

When a task is exported to an XML file, no password or password hash information is exported inside it. If a password is required for the task to run, it will have to be entered again in the Task Scheduler, using SCHTASKS.EXE, or Group Policy. When a large number of tasks need to be moved or copied from one machine to others, use SCHTASKS.EXE to export the settings (for example, "schtasks.exe /query /xml") and then import them on other computers across the network.

Scheduled tasks cannot use the older Managed Service Accounts (MSAs) for their identities, but, on Server 2012, Windows 8 and later, we can use Group Managed Service Accounts (GMSAs) for scheduled tasks.

To run a task under Local System identity, enter "system" then click Check Names; the identity will be set to "NT AUTHORITY\SYSTEM", but you can't type that full string in

and make it work. To run as "Network Service" or "Local Service", though, simply enter those names by themselves. There is no password to be entered for these identities.

Keep in mind that there are various built-in identities under which scheduled tasks can be run, and these identities have different levels of access to local or remote systems:

- **Local System** identity is the most powerful on Windows, and a process with this identity will authenticate over the network as the AD computer account of the machine (*domain\hostname*$).

- **Network Service** has the privileges of any authenticated user, is a member of the Local Users group (or Domain Users group on a controller), and will authenticate over the network as the AD computer account of the machine, and will be a member of the Authenticated Users group at the remote computer.

- **Local Service** has the privileges of any authenticated user, is a member of the Local Users group (or Domain Users group on a controller), but can only authenticate over the network anonymously as a "null session" user (not as *hostname*$) and only as a member of the Everyone group at the remote target.

Virtually nothing on Server 2008 or later can be accessed anonymously over the network anymore without first modifying default permissions, and loosening permissions for this purpose is generally not recommended.

In the Windows Firewall, there are built-in rules for remote access to the Scheduled Tasks service. They are disabled by default.

In Event Viewer, right-click an entry in a log > Attach Task To This Event. When events of that type are logged again, a task can be executed, perhaps to display a message. The event type filtering can be highly customized, including the use of XPath queries.

Scheduled tasks do not have to run on a scheduled basis, tasks can also be triggered by a variety of circumstances, which really expands the usefulness of the tasks for security:

- At log on (not a per-user logon script, a per-system logon script for any user)
- At startup (like autoexec.bat, but it runs after Windows is fully up)
- On idle (after X minutes of no mouse or keyboard input)
- On an event (in Event Viewer)
- At task creation/modification (to be alerted for hacker/malware changes)
- On connection to a user session (RDP)
- On disconnect from user session (RDP)
- On workstation lock (perhaps to trigger a privacy hygiene cleaner)
- On workstation unlock (perhaps to stop the hygiene cleaner)

Tasks can also be set to run every few minutes on condition that a given wireless or VPN connection is currently established. This can help deal with split tunneling and other issues related to insecure network usage.

You can create a scheduled task to run only once a few hundred years from now, i.e., never, but allow regular users to manually run that task, perhaps with a shortcut or script which runs schtasks.exe to execute the task. The task can run as System or as an administrative user, which is dangerous, but does allow emulating Linux *su* on Windows. Running a hidden background job this way is easy, but to launch a graphical application on the desktop of the local user as System or as another administrative user (dangerous!) you will have to use a tool like psexec.exe, which is a free download from Microsoft's web site; for example, to have a task launch Calculator as a visible GUI application on the desktop of the user with session ID number 1, running as the System identity:

```
psexec.exe -s -i 1 -d -accepteula calc.exe
```

Scheduled PowerShell Script Tips

If you create a scheduled task to run a PowerShell script and it unexpectedly fails, here are some troubleshooting tips:

- Enter the full path to powershell.exe, not just the binary name, for the command, e.g., C:\WINDOWS\System32\WindowsPowerShell\v1.0\powershell.exe.

- Use the -File parameter followed by the full path to the script to be run, and surround that path with double quotes (not single quotes) even if there are no space characters in that path.

- It's often easier to have the scheduled task run a wrapper script than to pass in a variety of command-line arguments through the Task Scheduler. The wrapper script simply executes your desired script with all of your desired arguments and comments.

- Make the last command of the wrapper script execute "return 0" (success) or "return -1" (failure) so that the Task Scheduler GUI will accurately display success or failure.

- Have the script write more detailed debugging info, if desired, to the Application Event Log or to your own textual log file.

- When running the script as Local System, also check the "Run with highest privileges" checkbox.

On Your Computer

Examine the contents of the Process-Reaper.ps1 script:

```
ise C:\SANS\Day1-PowerShell\Process-Reaper.ps1
```

Close the script tab.

Share your PowerShell scripts folder to the Everyone group as read-only:

```
New-SmbShare -Path C:\SANS\Day1-PowerShell -Name TaskScripts
```

Make a note of what your computer name is, which you'll need soon:

```
$env:computername
```

Pipe your computer name into the clipboard, so you can paste it later:

```
$env:computername | clip.exe
```

Launch the Microsoft Paint application and leave it running:

```
mspaint.exe
```

Edit the Default Domain Policy GPO using the Group Policy Management console.

In the Default Domain Policy GPO, navigate down to Computer Configuration > Preferences > Control Panel Settings > Scheduled Tasks.

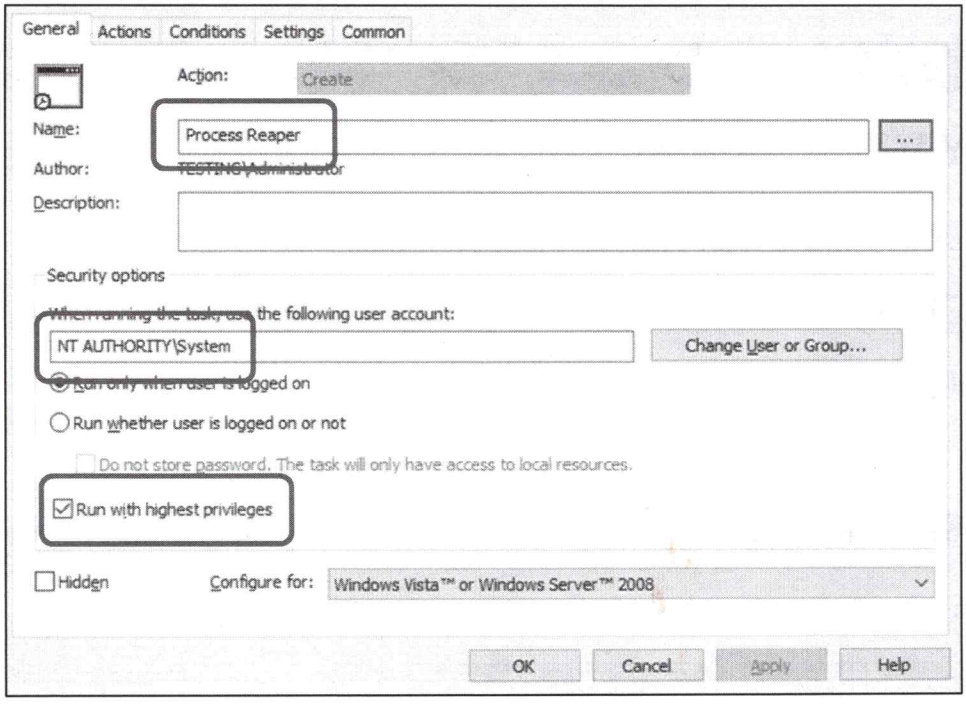
Right-click Scheduled Tasks > New > Immediate Task (At Least Windows 7) > use the following settings to define a new task:

General tab: Name = Process Reaper

General tab: Use the following user account = NT AUTHORITY\System

(Tip: For the user account, click the "Change User or Group" button, enter "system", click the "Check Names" button, choose SYSTEM, press OK twice.)

General tab: Check the box "Run with highest privileges"

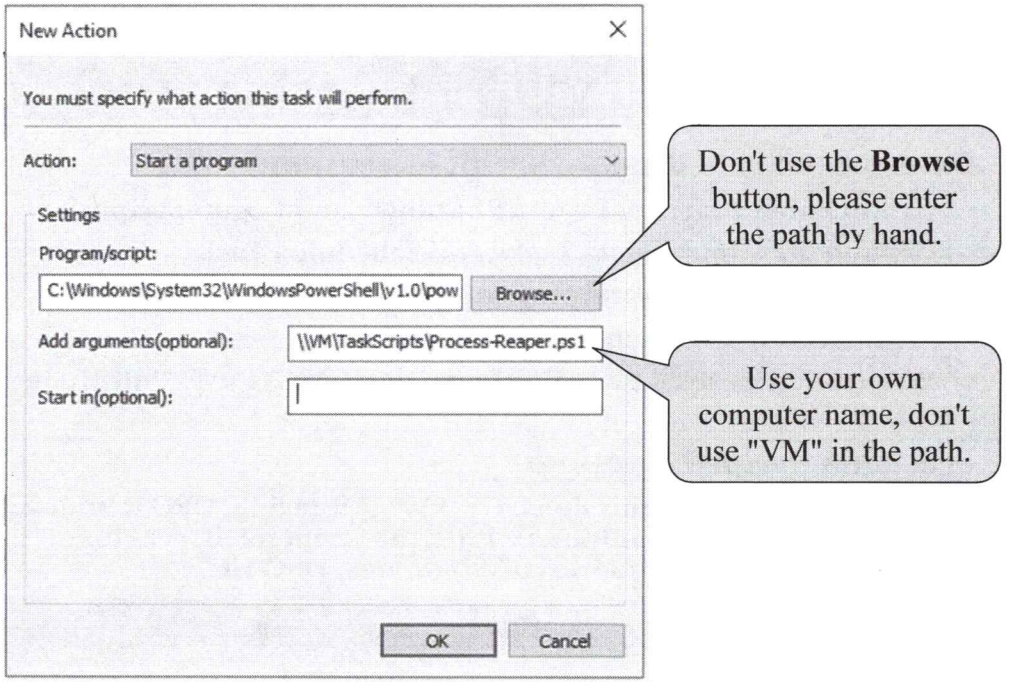


Actions tab > click the New button >

Action: Start a program

Program/Script: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe

Arguments: \\<*YourComputerName*>\TaskScripts\Process-Reaper.ps1



Note: In the screenshot above, replace "VM" in the UNC network path to the script in the shared folder with the name of your own virtual machine.

Close the editor window for the Default Domain Policy GPO (helps to commit changes).

Refresh Group Policy and the Microsoft Paint process will be terminated:

```
gpupdate.exe /force
```

Launch Microsoft Paint again and it will be terminated within 60 seconds:

```
mspaint.exe
```

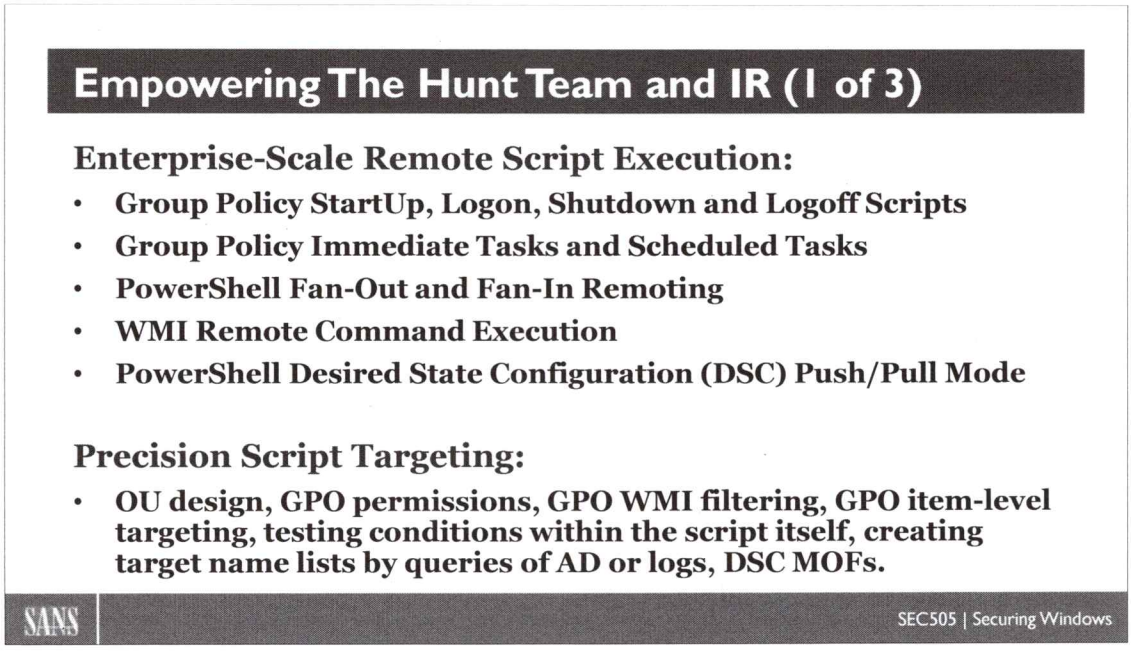
The script runs forever in the background, but it spends 99.9% of the time sleeping, hence, the script consumes only a trivial amount of CPU cycles.

Finished Already?

In PowerShell, how do you get scheduled task objects?

How do you get the "Process Reaper" task by its taskname and list all of its properties?

How do you stop just the "Process Reaper" scheduled task?



Empowering The Hunt Team And Incident Responders (1 of 3)

This is not a course on *what* the Hunt Team and Incident Responders should look for, this is a course on *how* they can look for it in a large enterprise.

When you only have two or three laptops sitting in front of you on a desk, it's easy to run your favorite forensics tools to gather the data you need. When you have "dead" memory and disk images to examine, the images can be repeatedly searched without disrupting any users at their desktops or interfering with any live servers.

But when you need to gather that data from ten thousand machines, it's a different kind of problem. This is more like a Big Data problem, except that the data is not in a database, it's distributed in the logs, file systems, registries, and memories of thousands of live workstations, laptops, tablets, phones and servers, some of which are inside the LAN, some are VMs in public clouds, and some roaming around the Internet.

How can the Hunt Team and Incident Responders gather the data they need to find indicators of compromise?

Enterprise-Scale Remote Script Execution

We need to be able to easily run our PowerShell scripts on thousands of remote endpoints and servers. We don't want to install an agent on each target, so the solution must rely on built-in features. The solution must be secure, or at least no less secure than the risks we already accept today. And we don't want to pay anything more for it. Possible?

We can do enterprise-scale remote script execution through:

- Group Policy Startup, Logon, Shutdown and Logoff Scripts
- Group Policy Immediate Tasks
- Group Policy Scheduled Tasks
- PowerShell Remoting
- PowerShell Desired State Configuration (DSC)
- WMI Remote Command Execution and Eventing

When we have many roaming or powered-off devices, remember that Group Policy will just patiently wait until these devices refresh Group Policy again. PowerShell remoting only works against targets that are running and accessible at the time the remoting connection is attempted, but this is not a limitation for Group Policy.

And for non-Windows targets, PowerShell supports Secure Shell (ssh) either natively or with free add-on modules.

And, of course, we may already have another Enterprise Management System, such as Microsoft System Center Configuration Manager (SCCM), Puppet, Chef, MobileIron, AirWatch, Altiris, etc. which may also have remote script execution capabilities. By all means, use these too!

Precision Targeting

It is possible to run our PowerShell scripts on every endpoint and server in the domain. But we will more likely need to execute our scripts on just a selected subset of our hosts, not on every host. Hence, we must have a targeting system for remote script execution. Sometimes, we will need to execute a script on just a few hundred boxes, or maybe just a dozen, or just one!

We can precisely target our remote script execution in various ways:

- Organizational Unit (OU) design and GPOs linked to these OUs.

- Global groups for computer accounts, with GPOs targeted to these groups with GPO permissions.

- GPOs targeted to specific computers based on WMI filtering, which allows GPO targeting based on any data the WMI service can access, i.e., almost anything.

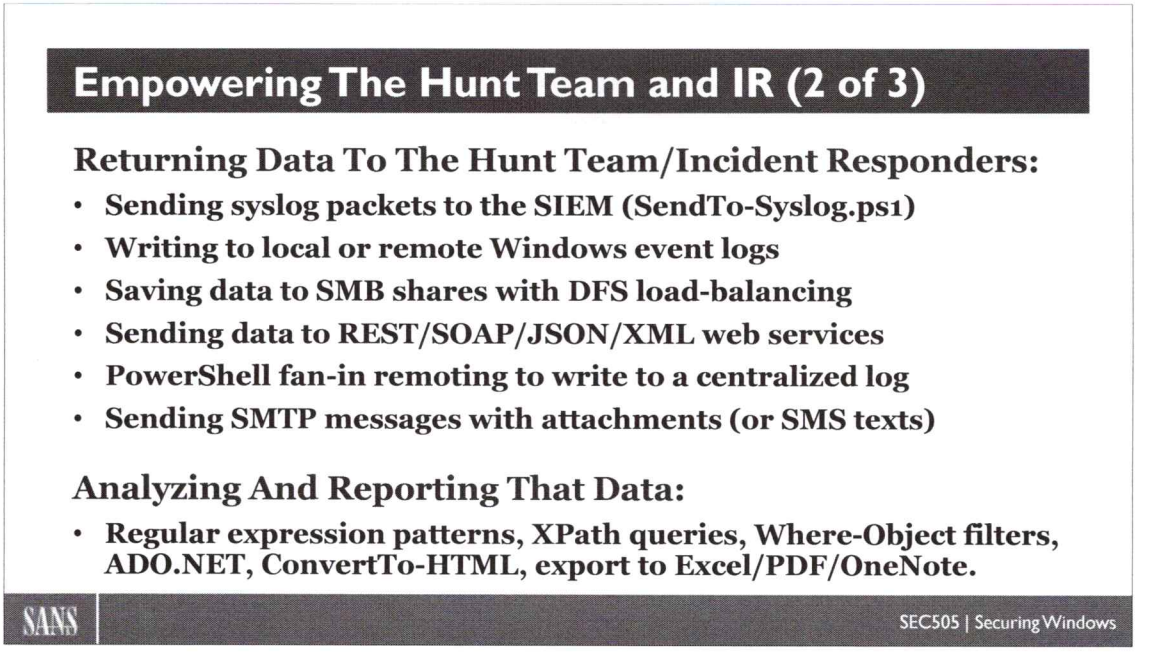
- Immediate and scheduled tasks which run PowerShell scripts are configured through GPO preferences; GPO preferences have item-level targeting filters that can key off almost any boolean-linked set of characteristics of the target computers, including anything queryable through WMI.

- GPO-assigned startup scripts do not always have to fully execute; a startup PowerShell script might first check for a set of conditions, such as certain registry values or installed services, then continue running if these conditions are met or immediately exit if the conditions are not met; any set of conditions we care to

code into our scripts can be used to decide whether to continue executing or to exit.

- With PowerShell remoting, ssh and WMI, we often first create an array of computer names, then feed that array as an argument into our scripts which then connect to those remote computers. Our targeting occurs when we select which computer names go into the array. We normally query for computer names from Active Directory, DNS, DHCP and our many various logs, including Windows event logs and the consolidated logs in our SIEM. With PowerShell and other tools, we can query Active Directory, log files, SIEM databases and other data sources with almost any conceivable set of search filters needed. It doesn't matter what tool is used to perform the query to extract the desired computer names: once the list of computer names is saved to a text file, that file can be fed into our PowerShell scripts as target data.

- With PowerShell Desired State Configuration (DSC) we "compile" configurations into MOF files, the MOF files are place in a shared folder or on a web server, then target hosts automatically download and apply their MOF files based on computer name or GUID number. By adding or removing MOF files, we can target the changes made through DSC.



Empowering The Hunt Team And Incident Responders (2 of 3)

When incident responders, forensics analysts and the Hunt Team members talk about "indicators of compromise", they are talking about data in the logs, file systems, registries and memories of compromised systems. For example, in the memory of a Windows laptop, there may be processes with malicious DLLs loaded, rootkit drivers, or back door services that have listening TCP ports.

Often, the data can queried on live systems without disrupting any users, but sometimes it will be necessary to reboot the machine to do a disk image capture or to freeze a machine to do a kernel-mode memory image capture. This is an important difference in the approaches of the Forensics Team and the Hunt Team: the Forensics Team often has no choice, the tools they need to run often require disrupting user activities, and they have management support to do so, but the Hunt Team usually is usually not permitted to disrupt user activities, so they are limited in the tools and techniques they are permitted to use. The Forensics Team are incident responders, they jump into action after the fact, but the Hunt Team is proactive and always busy doing something -- but they can't always be annoying and frustrating the users though.

For the live data the Hunt Team and Incident Responders need to acquire, and which they are *permitted* to acquire because it does not disrupt users, then PowerShell and WMI can often do it. And if there are other tools these teams want to use, then PowerShell and Group Policy can often be used to run them too, but in a scalable way.

Returning The Data

When PowerShell scripts are executed through remoting, Group Policy or some other method, and the scripts have discovered indicators of compromise or have gathered some other useful data, how do we get that data back into the hands of the Hunt Team? We need a way for our own PowerShell scripts to "phone home" and to interoperate with our SIEM or other monitoring systems.

There are many scalable ways PowerShell can return data back from target hosts:

- Send syslog packets to your SIEM with the SendTo-Syslog function (C:\SANS\Day1-PowerShell\SendTo-Syslog.ps1).

- Write to local or remote Windows event logs (Write-EventLog cmdlet or Write-ApplicationLog.ps1 script) and then collect that data with your SIEM.

- Redirect command output to a UNC shared folder path, saving data to a file named after the computer and a time stamp. The shared folder acts as a database table, the files are the records, and the file names allow us to select just the records desired. For fault tolerance and load balancing, use Distributed File System (DFS) shared folders that are replicated across a cluster of SMB servers.

- PowerShell can load classes from the .NET Framework (ADO.NET) to insert data directly into ODBC or SQL Server database tables over the network. Or, that data can be saved to a properly-formatted CSV or XML text file, then that file submitted to a database server over the network for import into a table.

- PowerShell can directly interact with web server applications designed around REST, SOAP, JSON or XML using cmdlets like Invoke-WebRequest and Invoke-RestMethod. Hence, if a logging system or SIEM supports such web protocols, PowerShell can directly write output data to it for analysis. When the SIEM or log analysis is performed by cloud providers as a service, this will mostly likely be the kinds of techniques used.

- PowerShell fan-in remoting allows many systems to remote inbound to one server and execute a command on that central server to write data to a file, event log or database on that server. PowerShell remoting is not just for *us* to connect *out to them*, but also for scripts running on *them* to connect *inbound to us* -- in this case, to connect inbound to run commands to save output data on a centralized server. For load balancing and fault tolerance, we would need multiple fan-in target remoting servers in a cluster (or just old-fashioned DNS round robin).

- PowerShell scripts can also send e-mail messages with large attachments (built-in Send-MailMessage cmdlet) or send SMS text messages (with third-party modules). In Outlook, inbox rules can be used to automatically move these messages into subfolders, highlight them by category or importance, and pop-up alerts as desired on the workstations of the Hunt Team. E-mail messages and

attachments can be encrypted before delivery with an S/MIME certificate or GnuPG (www.gnupg.org).

For example, to get a list of running processes on the local computer and export that list to a SMB shared folder on another machine, where the name of the CSV file is the name of the local computer:

```
Get-Process |
   Export-Csv -Path "\\RemoteMachine\Share\$env:ComputerName.csv"
```

Or to send a syslog packet with whatever payload, facility and severity you wish, use the function inside the SendTo-SysLog.ps1 script included with your courseware:

```
SendTo-Syslog -IP "syslogger.sans.org" -Facility "authpriv"
   -Severity "critical" -Content "Malicious service detected on
   workstation47: DComSrvcWin32"
```

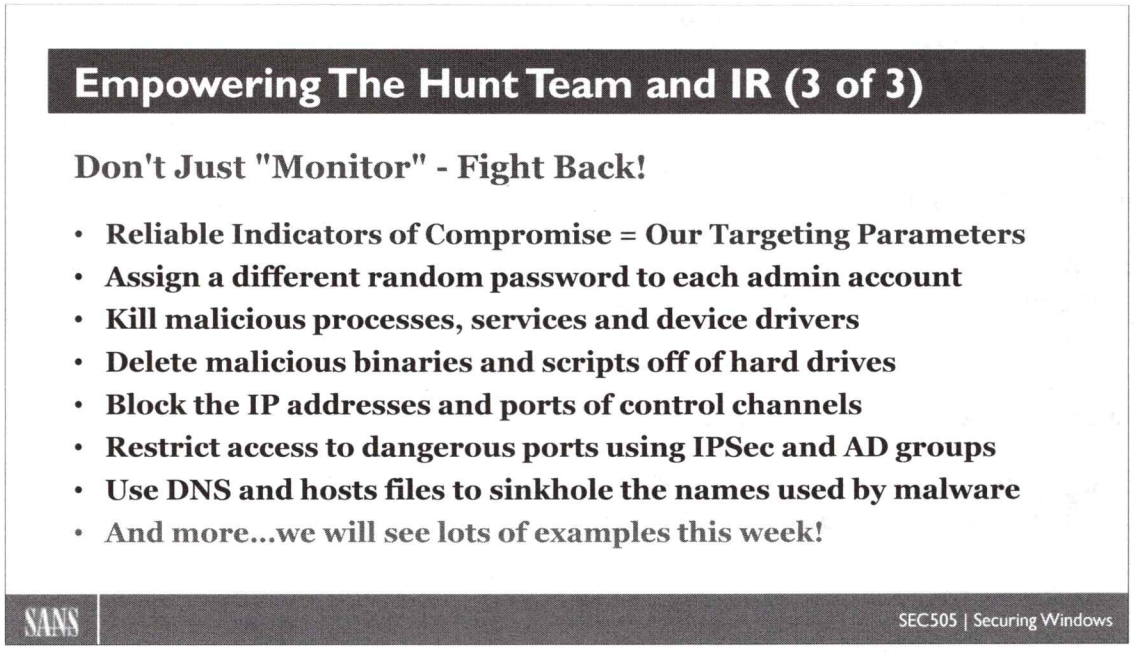
Analyzing and Reporting That Data

Real time log data is best analyzed by dedicated monitoring and IDS systems with machine learning intelligence. The old days of human, manual, hand-crafted log analysis are dead and gone. Except in the simplest of cases, human examination of raw monitoring data *in real time* does not scale, even when scripted with PowerShell. With machine learning and AI advancements in pattern recognition, we will hopefully detect totally novel threats for which there are no existing signatures.

But non-real-time forensics analysis can still be done by hand, and often must be done by hand when dealing with new threats. It's not real time, but it is still useful. Scripted analysis is still often used for generating custom reports too, like a nightly report which summarizes the output from multiple SIEMs and IDS systems that do not talk to each other directly.

For this type of non-real-time log analysis and reporting, PowerShell is great! PowerShell can parse and filter logs of various formats (XML, CSV, EVTX, etc.) and save the results in those formats too. There are even the various ConvertTo-* cmdlets for rendering to different formats (ConvertTo-HTML, ConvertTo-CSV and ConvertTo-JSON) and exporting that data. You can even "print" to PDF or OneNote pages.

PowerShell can use regular expressions, XPath and the Where-Object cmdlet for searching and filtering data. PowerShell can import data into SQL Server or Microsoft Excel, then invoke their built-in analysis functions instead of doing that heavy lifting itself. For number crunching, PowerShell can utilize the free Math.NET Numerics library (http://numerics.mathdotnet.com) and get hardware acceleration using the Intel Math Kernel Library (MKL) DLLs. These are the same kinds of techniques used by Python and NumPy.



Empowering The Hunt Team And Incident Responders (3 of 3)

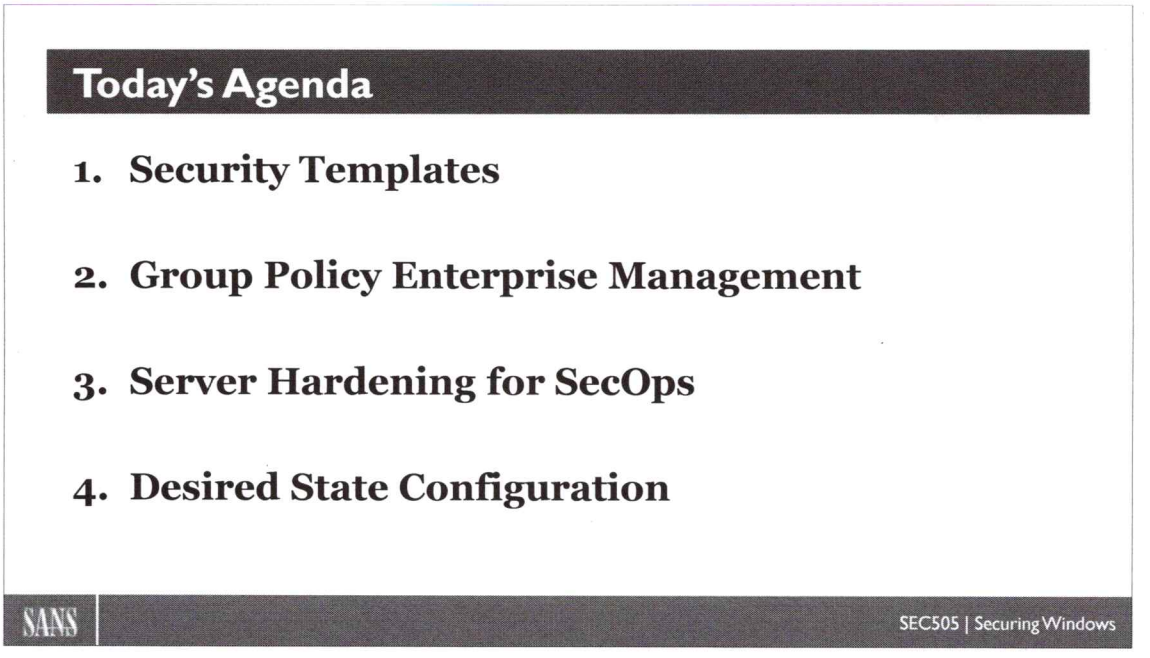
Don't just "monitor" -- *Fight Back!*

Once we have reliable indicators of compromise that we can query, we now have targeting criteria to launch a counter-attack with PowerShell and Group Policy.

As we receive alerts about hackers using stolen administrator account credentials to move laterally, reset those passwords! If hackers are using known IP addresses or port numbers for their control channels, block those packets! If we have the paths or hashes of the tools, services or drivers left behind by hackers to maintain control inside our LAN, delete them! If malware needs to resolve particular names through DNS, sinkhole those names on our DNS servers!

The point of doing monitoring is not to receive alerts, wring our hands in worry, and then do nothing. *Fight!* PowerShell and Group Policy give us additional SecOps weapons to fight these battles at enterprise scale, not just on the machines we can touch by hand.

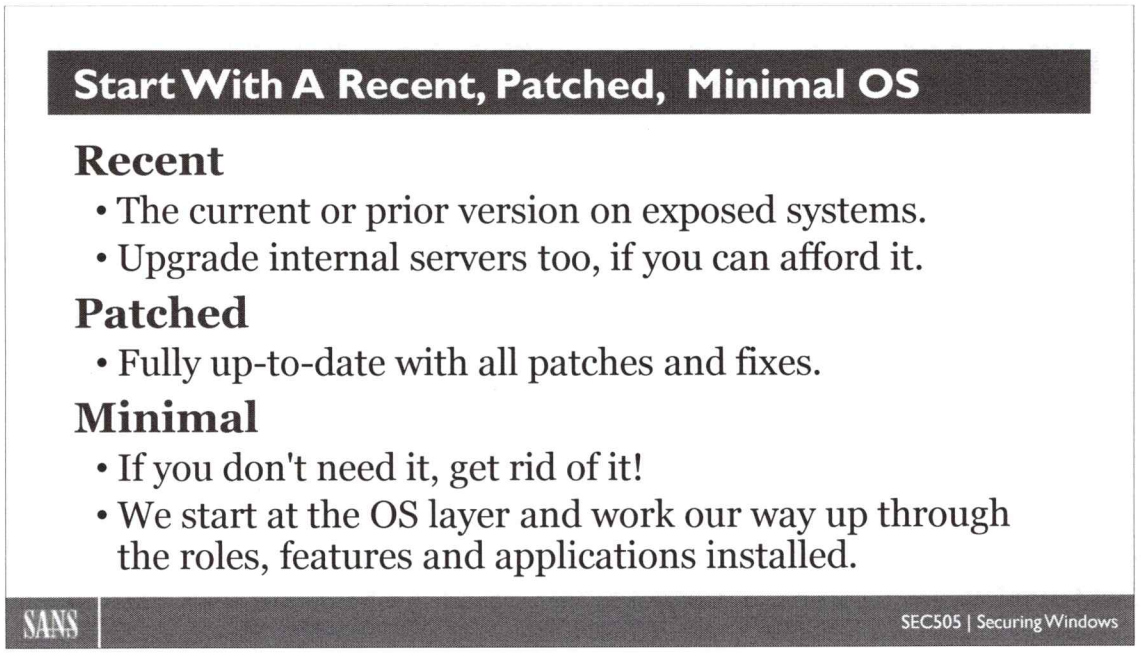
SecOps at scale is what this course is about. We will see more examples this week. We can't talk about everything in one day, so there is more to come!



Today's Agenda

"DevOps" is the term to describe how developers and operations staff need to work together as a team to deploy frequent updates to applications in a repeatable, scalable, secure way. Very similar to DevOps is SecOps. SecOps expresses the idea that the security and operations teams really should be just one team that bakes security into the design, deployment, maintenance, reconfiguration and retirement phases of IT projects and operations. SecOps, like DevOps, requires the use of templates and automation to achieve this goal.

The functionality of Windows Server is divided into roles and features. With Server Manager and PowerShell, we can inventory and change the roles and features installed on remote machines. With a bit more PowerShell, we can also automate other SecOps/DevOps tasks too.



Start With A Recent, Patched, Minimal OS

For server hardening, we want to start with a recent, patched and minimal operating system. This is the foundation, it's what we'll build on top of for the rest of the course.

Recent

This courseware assumes you have a recent operating system, where "recent" means either the latest version or the prior version. This is an expensive requirement, but remember that we are focusing on Internet-exposed servers, such as IIS web servers in the DMZ or on a cloud provider's network. In general, prefer the latest version you can afford, even for internal servers. And Windows 2000 and Server 2003 are dead operating systems and should be upgraded in all cases.

Use the Standard Edition server unless you need the expandability of Datacenter Edition. Note that in Server 2012 and later, there is no longer an Enterprise Edition.

Patched

After installing the latest OS version you can afford, apply the latest Service Pack and patches, and enroll the server in your patch management system to keep it updated. Quickly testing and applying new security patches is one of the most important duties in maintaining a hardened server, especially when the patch relates to a vulnerability on a listening TCP/UDP port which is exposed to the Internet.

Minimal

None of us, not even Microsoft or the hackers themselves, can predict what will be found to be vulnerable tomorrow. By definition, a new vulnerability is a *discovery*. What we

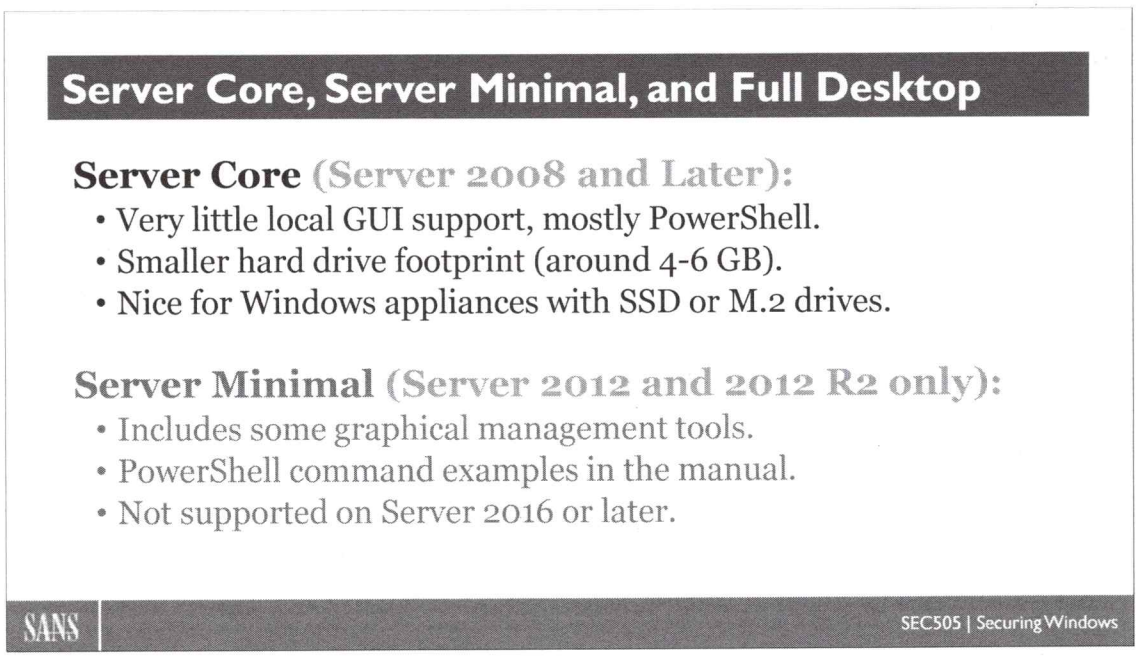
can do, though, is uninstall or disable the roles and features we don't need today in case they are discovered to be vulnerable tomorrow.

Throughout the course, the security recommendation is always the same:

If you don't need it, get rid of it!

There's no need to ask, "Is *X* vulnerable?", where *X* is an application, service, listening port, feature, bell or whistle. If it's running, it's vulnerable to some degree. The goal is to reduce risk by eliminating vulnerabilities while at the same time satisfying the needs of our users, managers and compliance auditors.

A minimal server runs the least amount of code necessary, starting at the OS layer and working our way up through roles, features and applications.

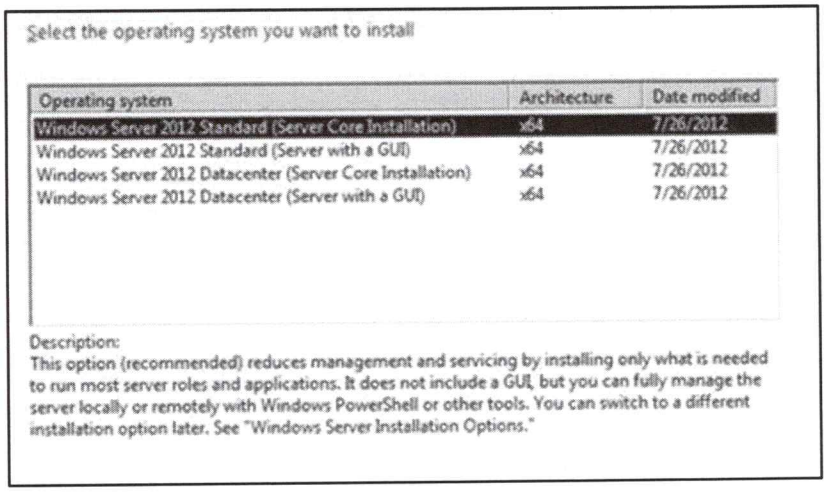


Server Core, Server Minimal, and Full Desktop

Windows Server 2008 and later includes an installation option named "Server Core" which strips away services, applications, and other features which are typically only necessary on users' client computers. Most of the graphical desktop and user applications are removed, such as File Explorer, Internet Explorer, Control Panel and the taskbar.

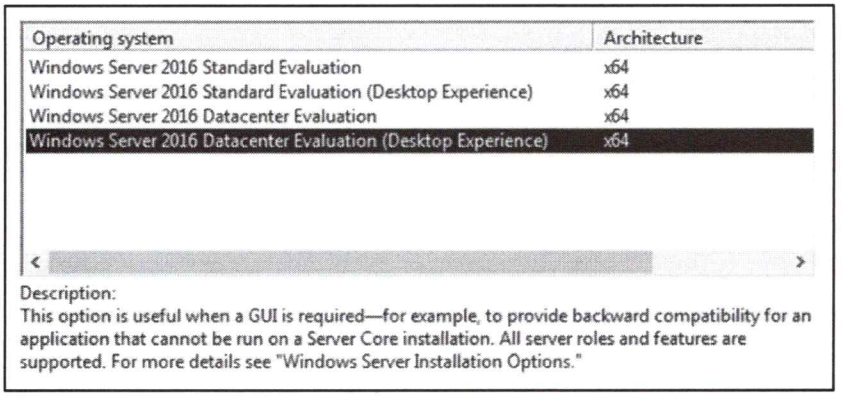
Server Core Installation

During the GUI install of any edition of Windows Server 2008 or Server 2012 you will see a screen similar to the one below where you can select the "Server Core Installation".



On Server 2016 and later, installing as Server Core is the default, so the word "Core" does not appear in the installation GUI; instead, if you do not want Core, if you want a

graphical interface, you choose the edition with the words "Desktop Experience" in the name. In the past, installing Windows Server with a graphical desktop was the default, now Server Core is the default and you must explicitly choose (against best practices) to install the unnecessary desktop. Microsoft is moving away from the idea of "Windows Server Core" being somehow different or special than just "Windows Server."



Again, Server Core is an *installation option* for Standard, Enterprise or Datacenter editions of Windows, it is not a separate edition of Windows Server itself. First you choose the edition (Standard or Datacenter), then you choose whether you want a graphical desktop (Core or Desktop Experience).

Switch Roles Without A Reinstall? (Server 2012 R1/R2 Only)

On Server 2008 and 2008 R2, there was no switching between the full installation and the Core installation without a full reinstall of the operating. This was a major barrier to the use of Core.

But on Server 2012 and Serve 2012 R2, when you're done with the GUI components, the graphical desktop can be uninstalled without reinstalling the entire OS (though you will need to reboot). Similarly, if you have Core and need the GUI desktop, you can switch from Core to Full Desktop mode without reinstalling the entire OS. Very handy!

However, with Server 2016 and later, Microsoft changed its mind again, and it is no longer possible to switch between Core mode and the full Desktop Experience mode without reinstalling (booo! a pox on Microsoft!). Once you install Windows Server 2016 or later you are stuck forever with having or not having the graphical desktop. You'll need to reinstall from scratch to switch.

Supported Roles and Features

The following roles are currently supported on Server Core:

- Active Directory Domain Services (AD DS)
- Active Directory Certificate Services (AD CS)
- Active Directory Rights Management Services (AD RMS)
- Active Directory Lightweight Directory Services (AD LDS)

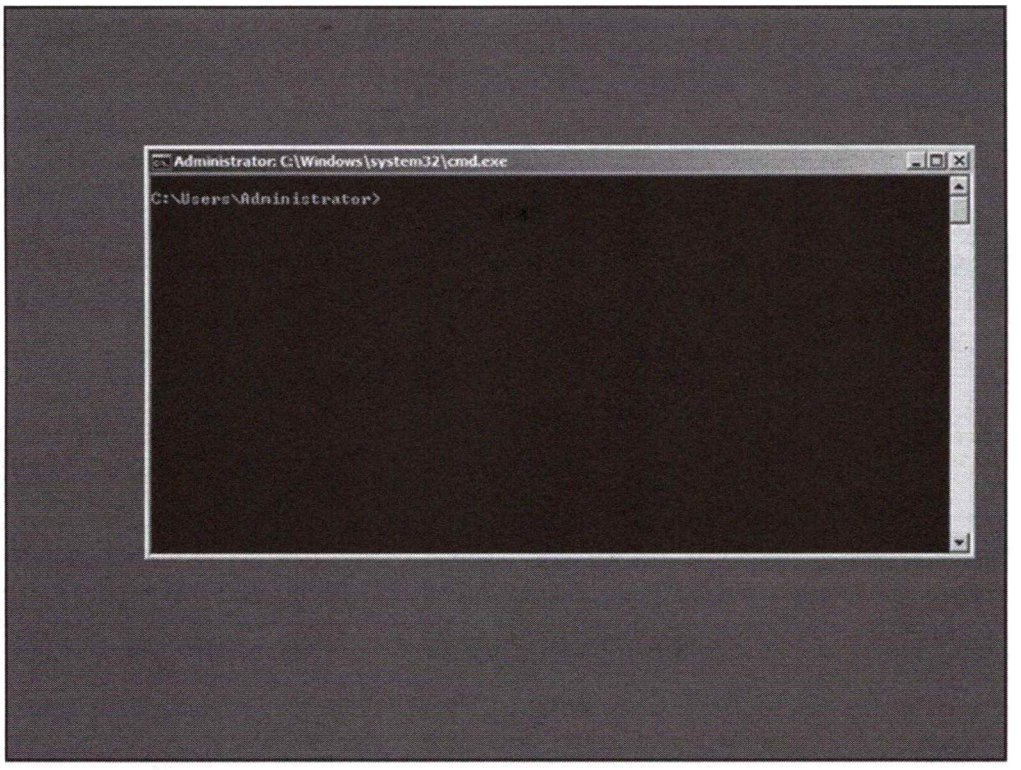
- DHCP Server
- DNS Server
- File Services
- Web Server (IIS)
- Hyper-V
- RRAS
- Print and Document Services
- Streaming Media Services
- SQL Server
- Windows Server Update Services (WSUS)

The following features can be installed on Server Core too:

- Failover Clustering (Enterprise Edition)
- Network Load Balancing
- Subsystem for UNIX-based applications
- Backup
- Multipath IO
- Removable Storage
- BitLocker Drive Encryption
- Simple Network Management Protocol (SNMP)
- Windows Internet Name Service (WINS)
- Telnet client
- WoW64 support for 32-bit applications (2008-R2 and later)
- PowerShell (2008-R2 and later)

No Graphical Interface Whatsoever?

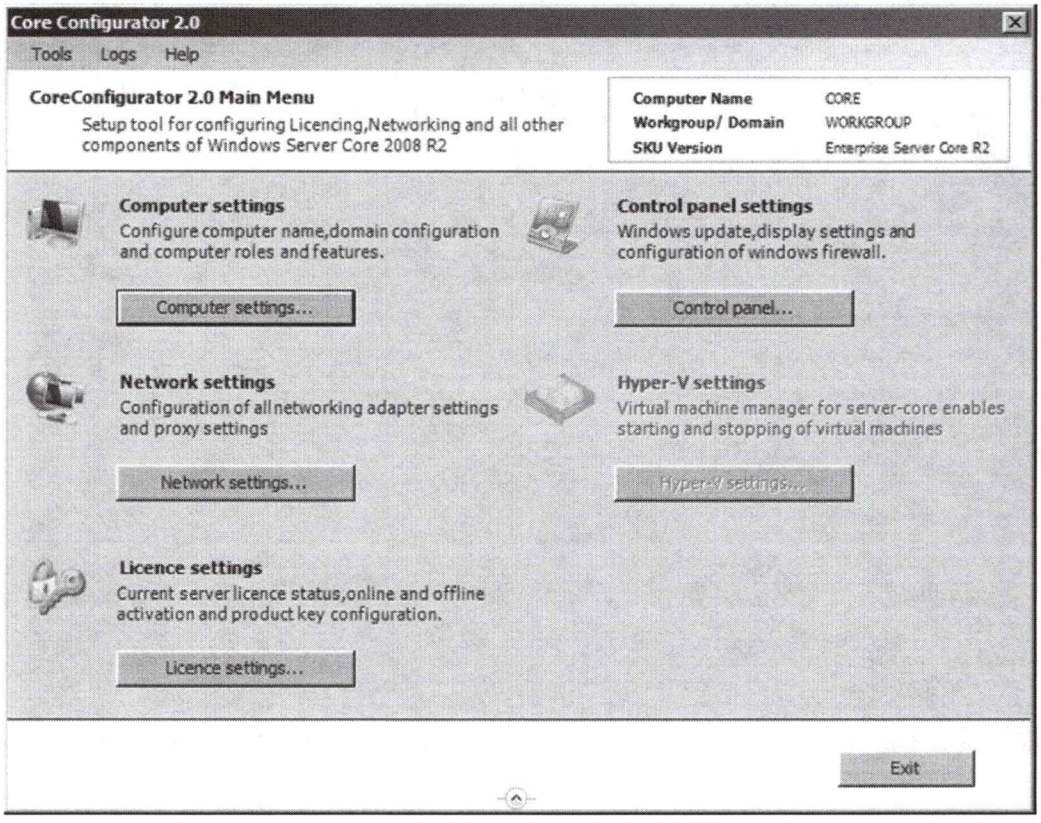
It is not entirely true that there is no graphical user interface with Server Core. There is a GUI desktop, but you can only run a CMD window, Task Manager, Notepad, and a few Control Panel applets by default (e.g., run "control.exe timedate.cpl" to change the time or "control.exe intl.cpl" for international settings). What you don't get is the taskbar, Start menu, Administrative Tools, MMC consoles, desktop shortcuts, Control Panel, etc.



When you log onto a Server Core box, either interactively or via RDP, you get an empty desktop with a CMD shell window. If you close the CMD shell, you'll have to hit Ctrl-Alt-Del to run Task Manager to relaunch CMD.EXE. From within CMD you can run powershell.exe.

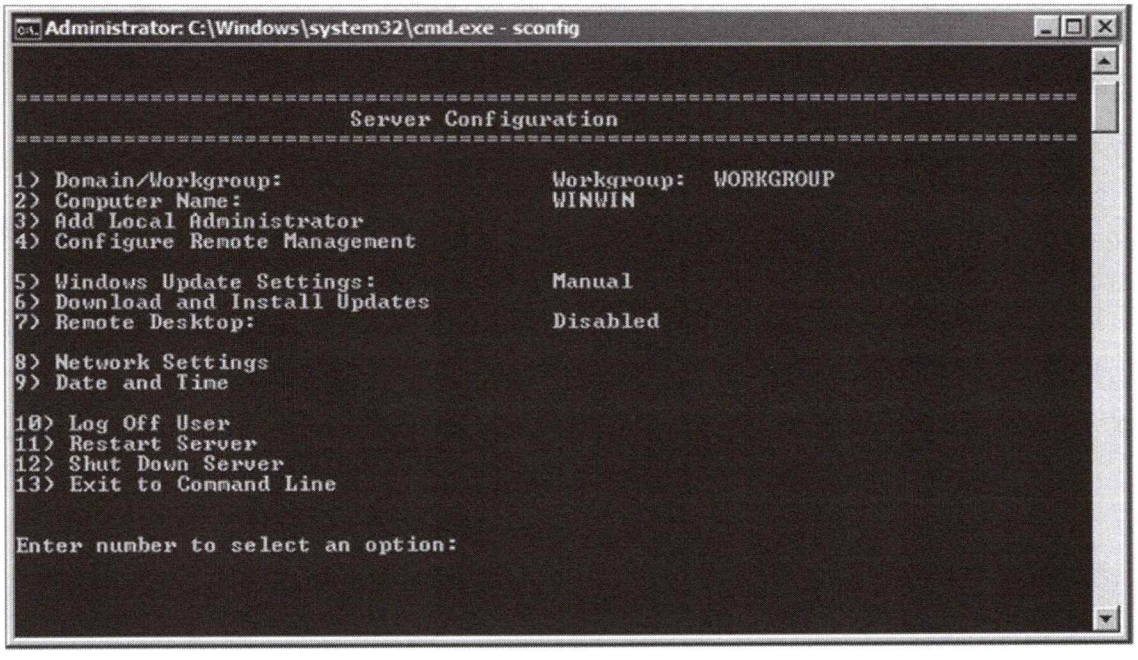
However, because there is limited support for GUI applications on Server Core, others have been developing GUI tools for managing the box, such as:

- Server 2008 R1 and R2: Core Configurator (http://coreconfig.codeplex.com)
- Server 2012 and Later: Corefig (http://corefig.codeplex.com/)



SCONFIG.CMD

On Server 2012 and later there is also a configuration script (sconfig.cmd) which will prompt the administrator and perform several initial configuration tasks.



Once configured with an IP address and joined to a domain, you can also connect to a Server Core box with most MMC console snap-ins. Hence, you can use many of your favorite graphical tools to manage Server Core, just not while you are sitting at the box or while controlling it remotely through RDP.

Server Minimal (Server 2012 and Server 2012 R2 only)

On Server 2012 and Server 2012 R2, there is a third option: Minimal Server. This is actually not an official installation option, but rather the ability add/remove certain graphical interface components on-the-fly without the necessity of reinstalling the OS (though you will need to reboot). You can start with a Full install, then demote down to Minimal or Core; or you can start with a Core install and add components to get back up to Minimal or Full.

Minimal Server includes support for locally running Server Manager, the MMC.EXE console and its snap-ins, and most Control Panel applets. Internet Explorer, the desktop, and File Explorer are not included in Minimal mode, but they can be added back if the Full mode is restored (specifically, if the "Server-Gui-Shell" component is added back).

What about Server 2016 and later? Officially, there is no such thing as "Server Minimal" for Server 2016 or later, but, on the other hand, it is not true that you cannot run any graphical tools on Server Core at all, and the list of GUI tools you can run on Server 2016 and later is growing as Microsoft releases updates. There is no way to know for sure which GUI tools can and cannot be run other than doing Internet searches and testing.

To reduce from Full mode to Minimal:

```
Remove-WindowsFeature Server-Gui-Shell
```

To reduce from Minimal to Core:

```
Remove-WindowsFeature Server-Gui-Mgmt-Infra
```

To reduce from Full GUI down to Core with a single command:

```
Remove-WindowsFeature Server-Gui-Shell,Server-Gui-Mgmt-Infra
```

To go from Core back up to Full is more difficult because you will likely need to provide the path to the install.wim file on the source DVD or ISO file.

If the following command shows "Removed" for the Install State, the binaries are not present on the local drive will need to be copied from the DVD:

```
Get-WindowsFeature Server-Gui*
```

If the binaries are not present on the drive, they'll have to be copied from the DVD or ISO file. You will need the index number of the correct image from the source DVD (drive

d:\). If your source is an ISO file, just double-click or execute the name of the file in PowerShell in order to mount that ISO file as a drive letter.

To list the index numbers of the images in the source WIM file (replace d:\ with yours):

```
Get-WindowsImage -ImagePath d:\sources\install.wim
```

To list the index numbers of the images in the source WIM file using dism.exe instead:

```
dism.exe /get-wiminfo /wimfile:d:\sources\install.wim
```

Here are the normal index numbers of the images in install.wim:

Index 1 = Server Core Standard
Index 3 = Server Core Datacenter
Index 2 = Full GUI Standard
Index 4 = Full GUI Datacenter

To go from Core back up to Full GUI Standard (Index 2):

```
Install-WindowsFeature server-gui-mgmt-infra,server-gui-shell
  -source:wim:d:\sources\install.wim:2
```

To go from Core back up to Full GUI Datacenter (Index 4):

```
Install-WindowsFeature server-gui-mgmt-infra,server-gui-shell
  -source:wim:d:\sources\install.wim:4
```

Server Nano

Server Nano (Server 2016 and Later):

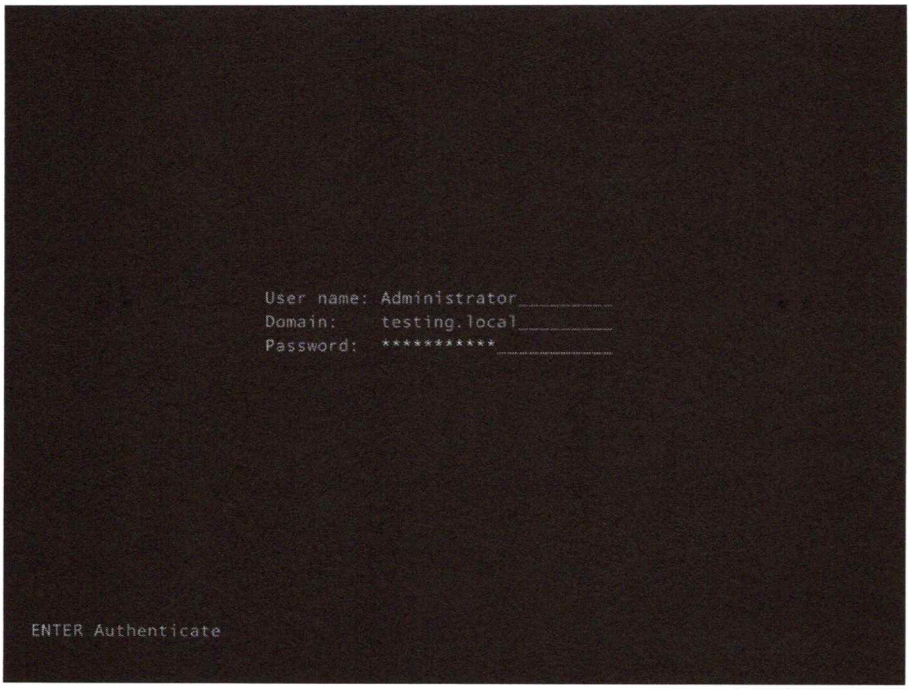
- Only about 410MB disk footprint!
- No graphical desktop whatsoever (runs "headless").
- Best for web servers, Hyper-V servers, and containers.
- Install image from DVD:\NanoServer (PowerShell).
- Requires a Software Assurance agreement with Microsoft.
- Only supported with current or prior update (no LTSB).

What about the security benefits?

SANS | SEC505 | Securing Windows

Server Nano

Windows Server 2016 introduced a new installation option: Server Nano. A basic installation of Nano requires only about 410MB of disk space! To achieve this radical reduction in footprint size, Nano has no GUI support whatsoever, i.e., it runs "headless" and can only be managed via PowerShell remoting, remote WMI, serial port, or similar.



Server Nano is mainly intended for roles like being an IIS web server, Hyper-V server, SMB file server, and DNS server. A single Hyper-V server running Nano, for example, could host thousands of VMs running Nano. These tiny VMs could be terminated and respawned quickly in response to changing demand, attacks, or service failures. Nano can be installed directly to disk or in a VM as either Standard or Datacenter edition.

Server Nano can also host Docker-style "containers" or run in a container itself. Microsoft has partnered with Docker (www.docker.com) to natively support Docker's management tools, but the container implementation is Microsoft's own.

For Nano-specific hardening, it's best to uninstall any unneeded packages; otherwise, Nano hardening techniques are the same as for Server Core.

Licensing and Support Headaches

Sound too good to be true? Be aware that there are some Nano headaches:

- You must have a Software Assurance agreement with Microsoft.
- There is no Long-Term Service Branch (LTSB) support, only the Current Branch, which means you must regularly apply updates to maintain Microsoft's support.
- Each update will be cumulative of all prior updates, i.e., each update "rolls up" all prior updates into one all-or-nothing package that must be applied as a whole.
- There will be approximately 2 to 3 updates per year, and Microsoft will only support you if you have either the latest or the prior update applied; hence, if your last-applied Nano update is more than two update releases behind, Microsoft will not support you or help troubleshoot any Nano problems!
- While Server Nano can be joined to an Active Directory domain, it currently does not support Group Policy! Nano is intended to be managed entirely through PowerShell and Desired State Configuration (DSC).
- In the past, Windows Server was licensed per-processor, but with Server 2016 and later, it is now per-core (which may be good or bad for you, it depends).
- Nano only supports 64-bit applications.

Clearly, Server Nano is installed for rapid DevOps-style management, especially for web applications, microservices, cloud-hosted VM appliances, and the like. Hence, use Nano if you can, but you'll often need to use Server Core instead, especially inside the LAN for on-premises servers.

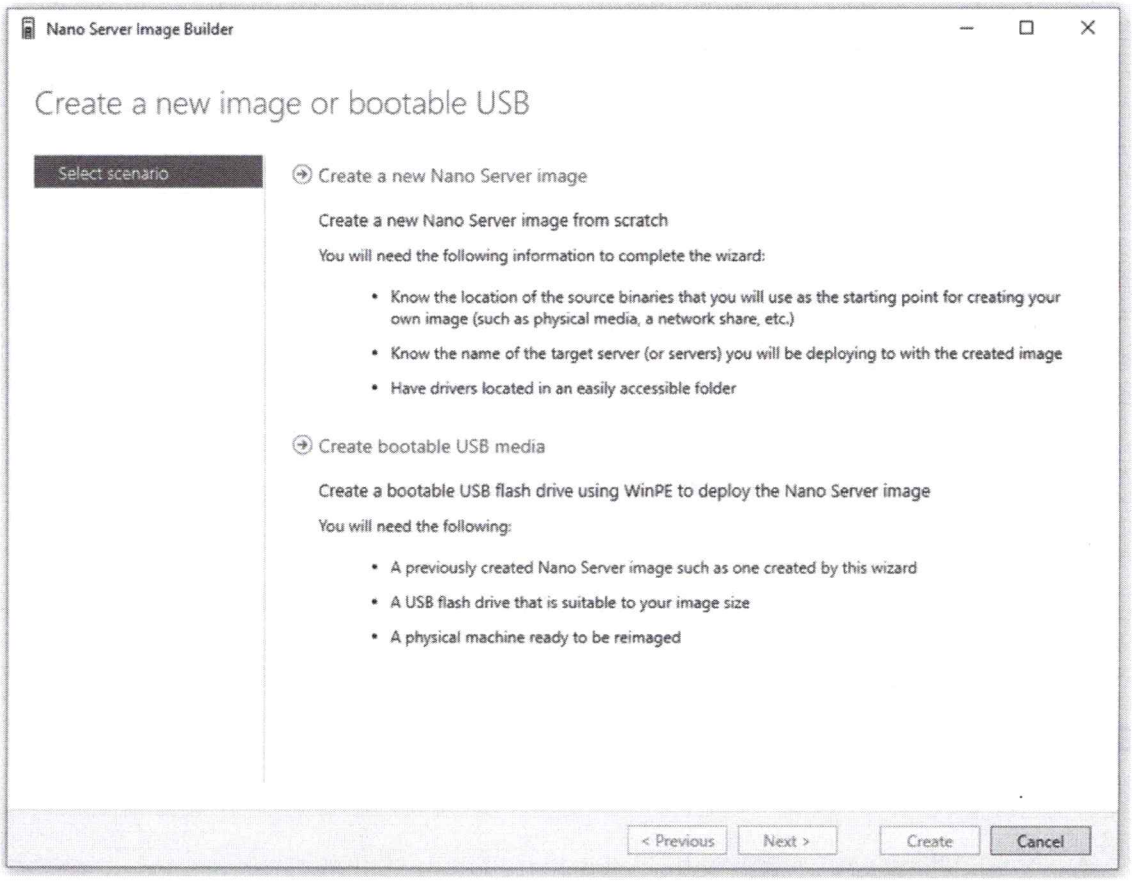
Nano Server Image Builder Tool (GUI)

Server Nano is not installed using the normal Windows Server GUI setup application, it must be custom-built as an installation image or VM, including any necessary third-party

device drivers you'll need. This is all done in PowerShell, but there are graphical wrappers available to make it almost point-and-click easy.

In particular, see Microsoft's free Nano Server Image Builder graphical tool, which can downloaded from http://aka.ms/NanoServerImageBuilder. Using this GUI tool, you can:

- Create bootable VM, ISO, or USB drive image.
- Select roles and features to install.
- Install servicing packages, such as for IIS.
- Install device drivers (with associated tool to automatically detect hardware).
- Join the Nano image to an Active Directory domain.
- Set the Administrator password.
- Configure the IP address or DHCP.
- Assign a custom script to run after setup is complete.



Also, on the Windows Server installation DVD/ISO, look in the \NanoServer folder and its subdirectories. There you will find a ReadMe.txt file with a link to Microsoft's web site and instructions on how to use the NanoServer.wim image file in that same folder and the PowerShell scripts in the \NanoServer\NanoServerImageGenerator folder too. We cannot cover all the steps here, it's best to read the latest instructions on Microsoft's web site.

To get a taste of what it's like to create a new Nano image by hand, start with the following commands (assuming Windows Server DVD/ISO is mounted on the D:\ drive):

```
cd D:\NanoServer\NanoServerImageGenerator

Import-Module .\NanoServerImageGenerator.psm1

Get-Command -Module NanoServerImageGenerator
```

The graphical Nano Server Image Builder tool (mentioned above) is just a GUI wrapper for running the New-NanoServerImage cmdlet from this module.

Server Core and Server Nano Benefits

These are the main benefits of using the Server Core or Nano installation option:

- Because a Server Core/Nano installation has fewer applications and services, it has fewer software components that can fail, lock up the system, waste memory, have bugs, listen on TCP/UDP ports, or otherwise be attacked and compromised, i.e., it has a smaller "attack surface". Stripping away unnecessary roles, features, components, bells and whistles is an essential part of making a hardened server.

- Fewer roles, features and components means there is less to manage on a Server Core/Nano installation. This means the box can be regarded more as an appliance than a full-featured server.

- Fewer roles, features and components means fewer updates and reboots per year.

- A Server Core/Nano installation requires less hard drive space. When combined with data deduplication, a large number of VMs can be crammed into one storage volume.

- On Server 2012 and later, you do not have to reinstall the OS to migrate from Core to Minimal to Full or back again (does not apply to Nano).

Server Core and Server Nano Drawbacks:

However, there are some drawbacks and important issues to consider:

- Server Nano has restrictive licensing and support requirements.

- When you uninstall GUI components, third-party software that is installed and requires these components may no longer function correctly. Microsoft roles and features may also be uninstalled, but at least there is a warning. Make sure to back up configuration settings for third-party and Microsoft software before removing GUI components, such as when returning to Core mode.

- When moving from Core up to Full, be aware that several gigabytes of binaries will need to be installed either over the Internet or from local media. Also, if updates are applied after moving to Full, several more gigabytes of patches will also need to be downloaded and applied.

- Most local administration is performed from the command line using PowerShell or tools like NETSH.EXE and SC.EXE, unless you install a third-party GUI tool on Server Core (no such option on Nano). Graphical tools can still be used remotely over the network, unless there are networking problems of course...

- If there are networking problems, you cannot use your favorite GUI tools to remotely manage Server Core/Nano. You are back to the console, if only to move from Core to Full temporarily.

- There are no graphical notifications of password expiration, new patch updates, or impending product activation deadlines when at the console.

- To install the .NET Framework on Server Core, you must have Server 2008-R2 or later. Without the .NET Framework, no ASP.NET and no PowerShell. Hence, Server 2008-R2 is the realistic minimum OS version.

- Only SQL Server 2012 and later is supported on Server Core, not earlier versions.

Is Server Core or Server Nano Really More Secure?

Server Core/Nano is nice for virtual machines and appliances, but the *security* benefits of Server Core/Nano is less important than other factors. Simply using Server Core or Server Nano will not be a magic security woes cure-all.

Here are some security implications to consider before deploying Server Core/Nano:

- A well-managed network is more secure than a poorly managed one. Windows administrators tend to be less proficient working at the command line than working with graphical tools (many hate the command line). Hence, as more Server Core/Nano boxes are added to the network, the more poorly the network as a whole will likely be managed in most environments.

    - Example: One reputed benefit of Server Core/Nano installations is that they are "like appliances, so you can deploy and forget them!" This is false, but the opinion will probably be very popular since Microsoft is one of the culprits spreading the notion. But these forgotten, unpatched, unmonitored and unaudited Server Core/Nano boxes will now become more attractive targets for hackers and safe havens for malware.

    - Example: In the confusion and panic of a malware outbreak or successful penetration, there is no time to research which command-line tools do the

same things as familiar graphical tools, hence, most environments will respond less quickly and effectively to emergency incidents as more Server Core/Nano boxes are added to these environments. Plus, what if the security tool you must run to save the day only comes in GUI form and cannot be run over the network?

- Server Core/Nano boxes must also be patched, even if the patching is less frequent than on full install boxes. But the hard part of patch management is the setup and maintenance of the patching infrastructure, not the addition of new servers or the installation of a few more patches. Installing less patches on some of the servers doesn't reduce the workload much, unless you are doing it all by hand to begin with.

- Like full installation machines, Server Core/Nano boxes must be hardened before deployment, even if there are fewer components on the system to harden. The assumption that you don't have to put roughly the same amount of hardening effort into the Core boxes than the regular servers is often false.

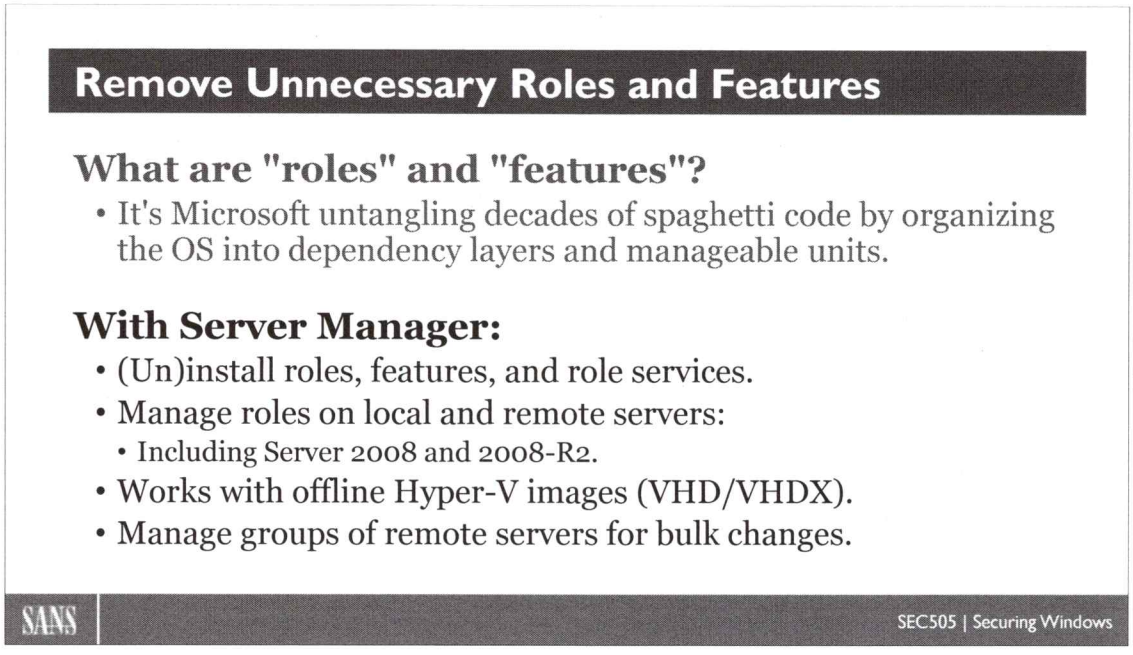
- Whether a box has the full or Core/Nano installation of Windows Server, the more roles and features you add, the more listening ports and protocols the box will have. Is a Server Core box with five roles more secure than a full installation with only one role? Or consider it a different way. Both a full installation and a Server Core installation of IIS might listen on TCP ports 80/443 (HTTP/HTTPS), 21/20 (FTP), 139 (SMB), 445 (SMB/CIFS), 135 (RPC), and 3389 (RDP/Remote Desktop Services), and all the same services are bound to these ports on both the full and Core installations. To a hacker who is port scanning, the full and Core installation boxes will look almost identical if they have the same roles and features installed. Granted, the Core box has fewer components overall, and this is good for security, but the *dangerous* components are the ones that get installed as roles or features, hence, it seems that reducing roles and features is much more important than using the Core option *per se*. Nano has fewer listening ports, but that's because it supports fewer features. For apples-to-apples comparisons, we need to compare systems running the same roles and features.

- You cannot install graphical browsers, e-mail programs, peer-to-peer file sharing applications, instant messaging clients, or other such malware magnets on Server Core/Nano, hence, the average infection rate from these vectors should be lower on Core/Nano boxes. But you don't install Core/Nano on workstations or laptops. And through corporate security policies, Group Policy and other security controls you can forbid, prevent and detect negligent administrators running such applications. It seems better to exercise organizational discipline to prevent negligent administrators from doing stupid things than to embrace Server Core/Nano just to achieve the same end result.

Finally, this really isn't a security issue, but Microsoft sometimes says that Server Core/Nano is great for remote branch offices that do not have local IT personnel since

you can always get remote command-line control of the box. But what if you can't? What if you have to get someone out there on the phone and walk them through the entire troubleshooting process *using only a command shell?* The horror…

In sum, then, the Server Core/Nano option has gotten lots of wonderful press, but the reality is that it is far more important to remove unnecessary roles and to quickly apply patches than it is to just run Server Core/Nano. Removing unnecessary roles and features can be done with Server Manager and PowerShell.



Remove Unnecessary Roles and Features

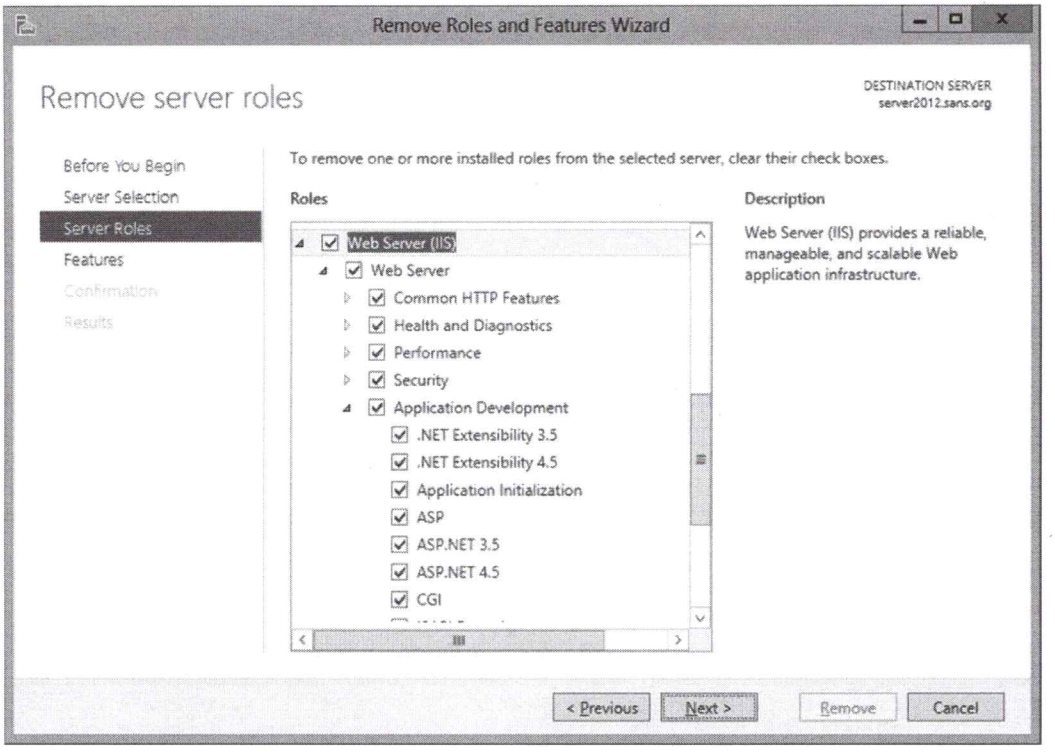
Windows Server 2008 and later versions are highly modular in the design, allowing very precise control over the features and components installed. In Server 2012 and later, the Server Manager tool can manage roles and features of remote machines over the network, including custom groups of servers simultaneously, and can install or uninstall roles and features from offline Hyper-V virtual machines which are not even running.

Server Manager can (un)install roles and features on remote Server 2008 and later operating systems, but there are special requirements for the older boxes. The older systems must have Server 2008 SP2, Server 2008-R2 SP1, .NET Framework 4.0 or later, PowerShell 3.0 or later, a special hotfix (KB2682011), and run a configuration script named Configure-SMRemoting.ps1 from Microsoft.

Remove Unnecessary Roles and Features

Many of the servers you will harden and expose to the Internet will be IIS servers hosting applications like SharePoint, Outlook Web Access, OCSP, WSUS, FTP, and a variety of SOAP/REST applications for your tablets and smart phones. You will also have DNS, VPN, DirectAccess, Exchange, RDS, SMB, Active Directory, RADIUS and other servers, but this manual will often focus on IIS because that's what many organizations are most worried about.

So, as an example, if you needed to install IIS, you would open Server Manager > Manage menu > Add Roles and Features > follow the guidance of the wizard. When you select the Web Server (IIS) role, it includes many components or "role services", but you will never need all of them at the same time on one server. As always, only install the components you need.



If IIS were already installed, but you wanted to remove unnecessary components, then go to Server Manager > Manage menu > Remove Roles and Features > select your server(s) > Web Server (IIS) section > uncheck the boxes for the unneeded components.

What Is Minimal?

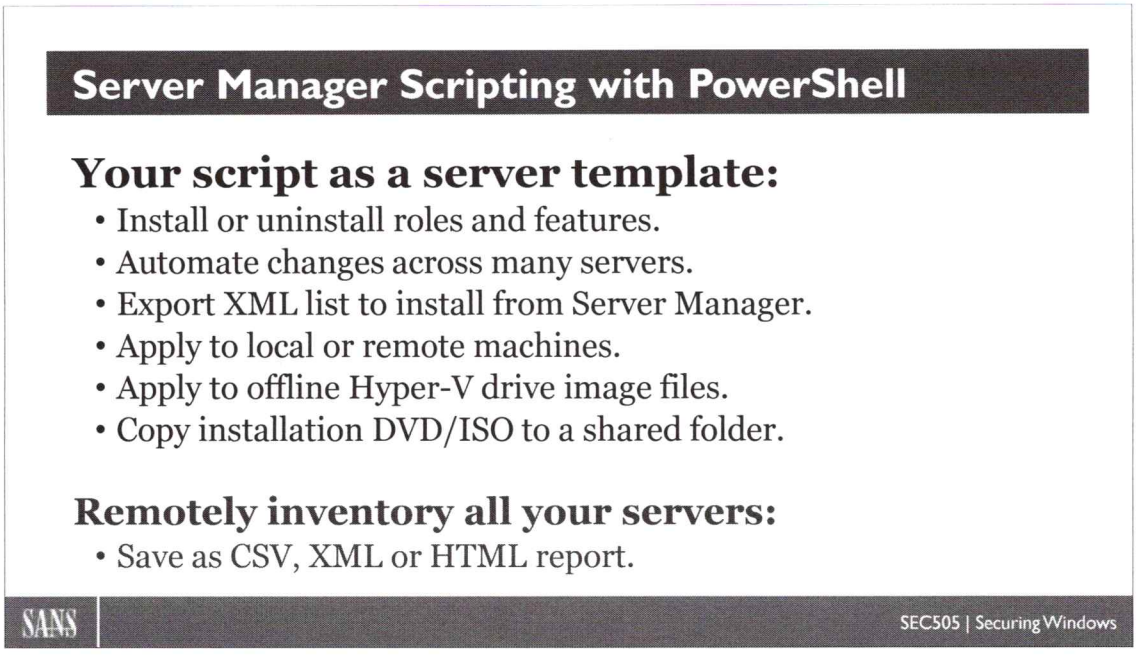
But what are the minimum roles, features and role services for any given server? That depends on what you want that server to be able to do. Each server will have different purposes, so each server will be different. It all depends on the applications, services and protocols needed.

On an IIS web server, for example, if you just want to serve up static HTML and graphics files, then the following would be a minimal component set:

- Common HTTP Features: Static Content
- Common HTTP Features: Default Document
- Management Tools: IIS Management Console

But notice that the above list does not include support for logging, non-anonymous user authentication, authorization rules, request filtering, compression, or ASP.NET. These are things you might need on this or another server. Each server will be different. Remember, though, that you can start with a minimal set of components and then add more as needed very easily using the graphical Server Manager or the PowerShell cmdlets for the same. You don't have to get the list exactly correct right from the beginning, you can always add more roles, features and role services later on.

When building the server, do it on a protected subnet not directly accessible from the Internet. Move the VM or physical server to the firewall's DMZ only after the server has been fully configured and hardened. It is not unusual for new servers to be scanned by automated scripts within minutes of going live on the Internet (see the reports of such tests at http://www.honeynet.org).



Server Manager Scripting with PowerShell

Instead of using the graphical Server Manager tool, on Server 2008 and later you can view, install and uninstall roles and features with PowerShell on both local and remote machines. This requires PowerShell 3.0 or later on both the local and remote computers.

To see your currently-installed roles and features:

```
# Next command only required on 2008-R2 and earlier:
Import-Module ServerManager


Get-WindowsFeature
```

To add or remove roles and features with PowerShell cmdlets, see the following:

```
Get-Help Install-WindowsFeature -Full

Get-Help Uninstall-WindowsFeature -Full
```

When you run Get-WindowsFeature, a great deal of text will scroll by, but the layout is similar to what you see in the graphical Server Manger. The hyphenated name on the right is what would be used to (un)install components or an entire category of components for a role.

For example, to install IIS with all optional components and tools:

```
Install-WindowsFeature -Name Web-Server -IncludeAllSubFeature `
  -IncludeManagementTools -Restart
```

Remember, these cmdlets work on offline Hyper-V image files and remote servers too:

```
Install-WindowsFeature -Name BitLocker -Vhd .\ImageFilePath.vhdx

Install-WindowsFeature -Name DNS -Computer Server47
```

When a role is uninstalled, the -Remove switch deletes the unneeded binaries from the drive or Hyper-V image file, which reduces storage consumption and is slightly more secure because, if reinstalled again, a path to a presumably-clean source can be given:

```
Uninstall-WindowsFeature -Name BranchCache -Remove
```

If the server installation DVD is copied to a shared folder which grants Read permission to Everyone (not Write!), then this UNC path can be given when installing roles:

```
Install-WindowsFeature -Name DNS -Source
   \\server\share\Sources\SxS
```

Server Manager XML Template

If there are many roles and features being installed at one time, you might prefer to use the graphical Server Manager to build a list of the additions, save the list to an XML file, then give this file as an argument to the Install-WindowsFeature cmdlet:

```
Install-WindowsFeature -ConfigurationFilePath
   .\DeploymentConfig.xml
```

To create this XML file with Server Manager, pull down the Manage menu, add roles and features like normal, but at the end of wizard's dialog boxes, don't click the Install button, click the "Export configuration settings" link at the bottom instead to save the XML file, then cancel the wizard. Note that this XML file can only be used to install features, not remove them.

Group Policy: Set Default UNC Path or Allow Windows Update

Instead of providing the UNC path to the installation files as an argument, you can also set the default path through Group Policy. You can even have servers download the necessary files over the Internet via Windows Update.

These options are controlled in the GPO setting named "Specify settings for optional component installation and component repair", which is located in a GPO under Computer Configuration > Policies > Administrative Templates > System.

On Your Computer

Display the currently-installed roles and features on a local or remote server:

```
Get-WindowsFeature -ComputerName $env:computername

Get-WindowsFeature | Out-GridView
```

Query the status of the DNS Server role and some of its properties:

```
$role = Get-WindowsFeature -Name dns

$role.installed

$role.additionalinfo

$role.postconfigurationneeded
```

> **Note:** A role may be installed, but not yet functional until the necessary post-configuration changes are completed, e.g., certificate services.

Install the File Server Resource Manager (FSRM) role, plus any necessary management tools for it, and, only if necessary, reboot the system:

```
Install-WindowsFeature -Name fs-resource-manager
  -includemanagementtools -restart
```

Install roles and features as defined in an XML template exported from Server Manager:

```
Install-WindowsFeature -configurationfilepath
  C:\SANS\Day2-Hardening\TelnetClient.xml
```

Create an XML inventory of server roles and features in your domain:

```
cd C:\SANS\Day2-Hardening

.\Get-FeaturesInventory.ps1 | export-clixml -path .\inventory.xml
```

Import the XML inventory and show some query examples:

```
$data = Import-CliXml -path .\inventory.xml

$data

$data[0]

$data[0].ComputerName

$data[0].Features

$data[0].Features.DNS

$data | foreach { if ($_.Features.DNS){ $_.ComputerName } }
```

Finished Already?

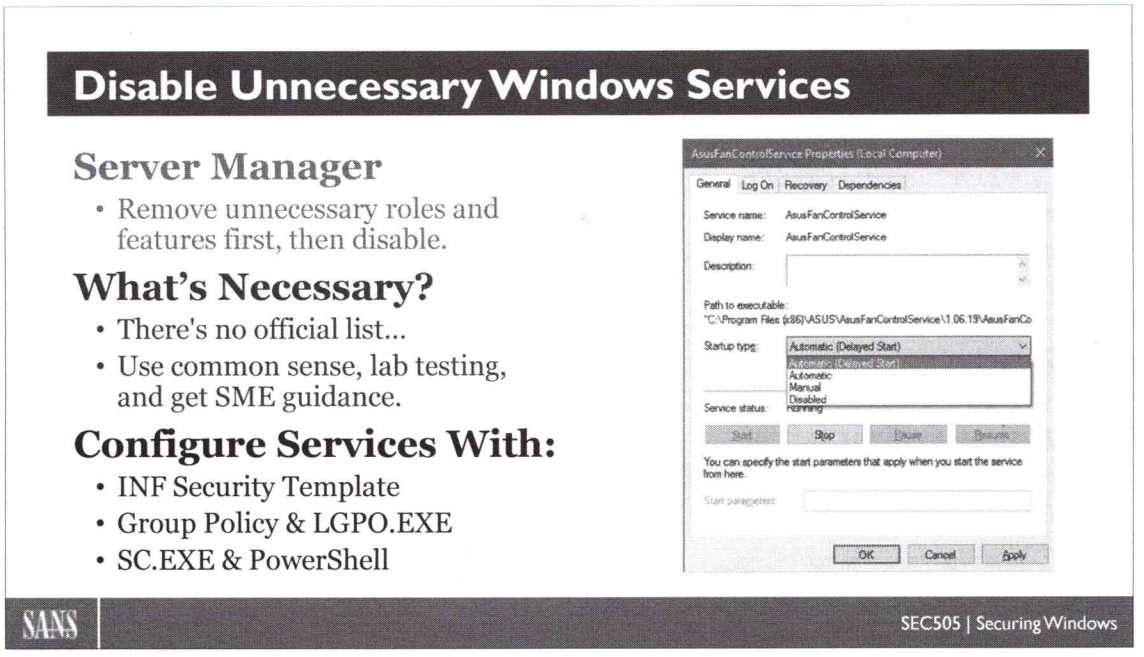
Look at the source code for the Get-FeaturesInventory.ps1 script used above. There are lines which run Get-AdDomain and Get-AdComputer. What are the properties and methods of the object(s) outputted by these cmdlets? To get started, try running:

```
$domain = Get-AdDomain

Get-AdComputer -Filter *
```

What is the difference between an array and a hashtable?

```
$array = @(1, 2, 3)

$hashtable = @{ one = 1; two = 2; three = 3 }
```

(Note that the array is defined with parentheses, while the hashtable uses curvy braces.)



Disable Unnecessary Windows Services

As a policy, if a service or feature is not used, it should be disabled or uninstalled. Disabling unnecessary services and features reduces the *potential* number of security holes (known and unknown) a hacker can exploit, and should improve system performance. Services may be disabled with the Services applet in the Administrative Tools folder, with an INF/XML security template, Group Policy, or the SC.EXE command-line tool. SC.EXE is the best command-line tool for managing services and device drivers on both local and remote systems; it is also very handy on stand-alone servers when Group Policy is not available.

Which Services Can Be Disabled?

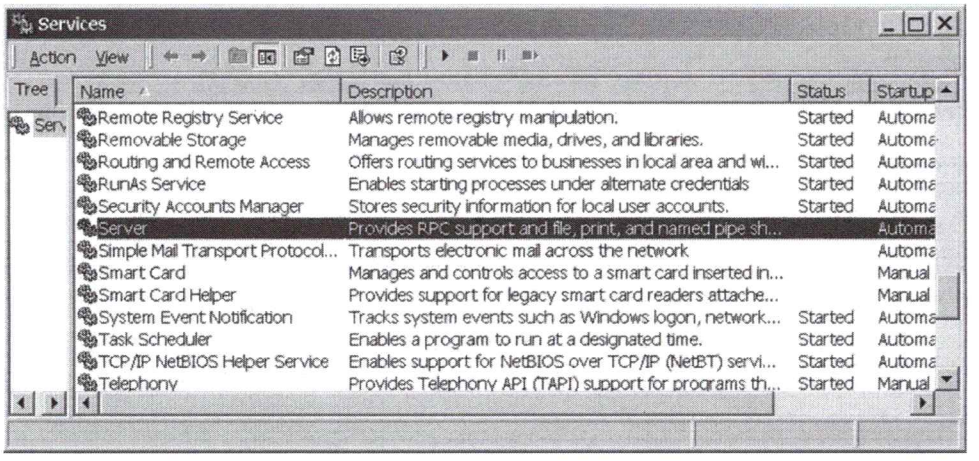
Most of the unnecessary services were uninstalled or disabled when Server Manager and the Security Configuration Wizard were used to eliminate unneeded roles and features. Hence, most of this work has already been done. Now it's time to mop up the stragglers.

However, there is no absolute or always-correct list of services which can be disabled. It depends on the type of server and the applications running on it. Hence, this part of the hardening process will require some testing in the lab. After a server is deployed for a few weeks, if it turns out that a disabled service was actually required, then that service can be easily started and the corresponding security template updated accordingly.

As an example, an IIS web server usually does not require the following services, but it will depend on the roles, features and applications installed (KB810866):

- Alerter
- ClipBook Server

- Computer Browser
- DHCP Client
- Distributed File System
- Distributed Link Tracking Client
- Distributed Link Tracking Server
- Distributed Transaction Coordinator (may be needed for database integration)
- DNS Client (be prepared to re-enable this except on simple installations)
- Fax Service
- File Replication
- FTP Publishing Service
- Indexing Service
- Internet Connection Sharing
- IPSec Policy Agent (unless IPSec is being used)
- Licensing Logging Service (may be required when using SSL)
- Messenger
- NetLogon (required on member servers)
- NetMeeting Remote Desktop Sharing
- Network DDE
- Network DDE DSDM
- Network Monitor Agent
- NNTP Service
- Print Spooler (set to manual, not disabled, so Service Packs can install)
- QoS RSVP
- Remote Access Auto Connection Manager
- Remote Access Connection Manager
- Remote Registry Service (some remote admin tools require this)
- Removable Storage
- RPC Locator (some RPC protocols require this)
- RunAs Service
- Server Service (some administrative tools require this)
- Simple TCP/IP Services
- Smart Card
- Smart Card Helper
- SMTP Service
- Task Scheduler
- TCP/IP NetBIOS Helper Service (Group Policy problems may arise without this)
- Telephony
- Telnet
- Remote Desktop Services
- Uninterruptible Power Supply
- Windows Management Instrumentation
- Windows Management Instrumentation Driver Extensions
- Windows Time
- Workstation Service (UNC virtual folders and some admin tools need this)

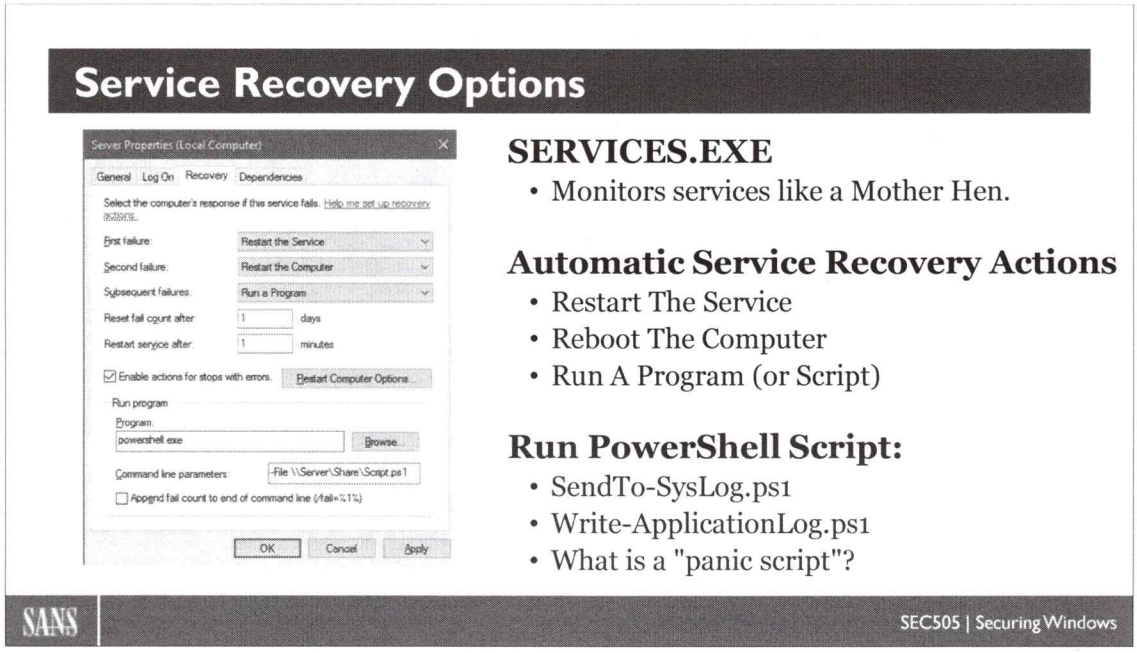


However, the following services usually are required on an IIS web server, but the exact list will be determined by the roles and features required (SCW can guide you):

- DCOM Server Process Launcher
- Event Log
- Protected Storage
- Remote Procedure Call (RPC) Service
- Windows Installer
- Windows NTLM Security Support Provider
- Windows Process Activation Service
- World Wide Web Publishing Service

It is a common misconception that an IIS web server requires the Server service (also known as the "File and Printer Sharing Service"). It does not. The Server service shares files over the SMB protocol, while IIS shares files over HTTP and FTP. However, this service is often needed for remote administration and over-the-network backups.

If a service is only needed temporarily, perhaps to run an administrative tool, then a service can be started and stopped on demand with a scheduled script.

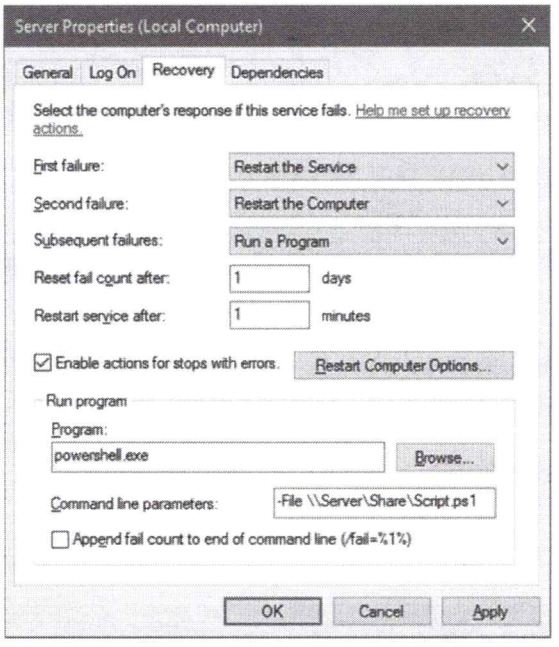


Service Recovery Options

Windows provides options for automatic recovery when a service fails or exits with a non-zero error code number. You define the number of days over which the count of service failures is tracked. During this period, you can specify up to three actions which will be undertaken by the computer when the configured service fails. You can attempt to restart the service, run an executable or script of your choice (including command-line arguments) or reboot the entire server. Each will be executed in turn as attempts to restart the service fail.

Try It Now!

To define recovery options for IIS, open the Services applet in Administrative Tools > double-click the World Wide Web Publishing Service > Recovery tab. At a minimum, set the action for all three failures to "Restart the Service". Consider writing a script that will undertake more intelligent recovery and notification actions. The option to "Reboot the Computer" should be the last action, if this option is used at all.



SC.EXE and Set-ServiceRecoveryOptions.ps1

You can set recovery options on many services more easily with the SC.EXE tool. For example, to reset the failure count after three days (259200 seconds) on the World Wide Web Publishing Service (w3svc), restart failed service after two minutes (120000 ms) twice in a row, then run a script named NOTIFY.PS1, execute the following:

```
sc.exe failure w3svc reset= 259200 actions=
   restart/120000/restart/120000/run/1000
   command= "powershell.exe \\server\share\notify.ps1"
```

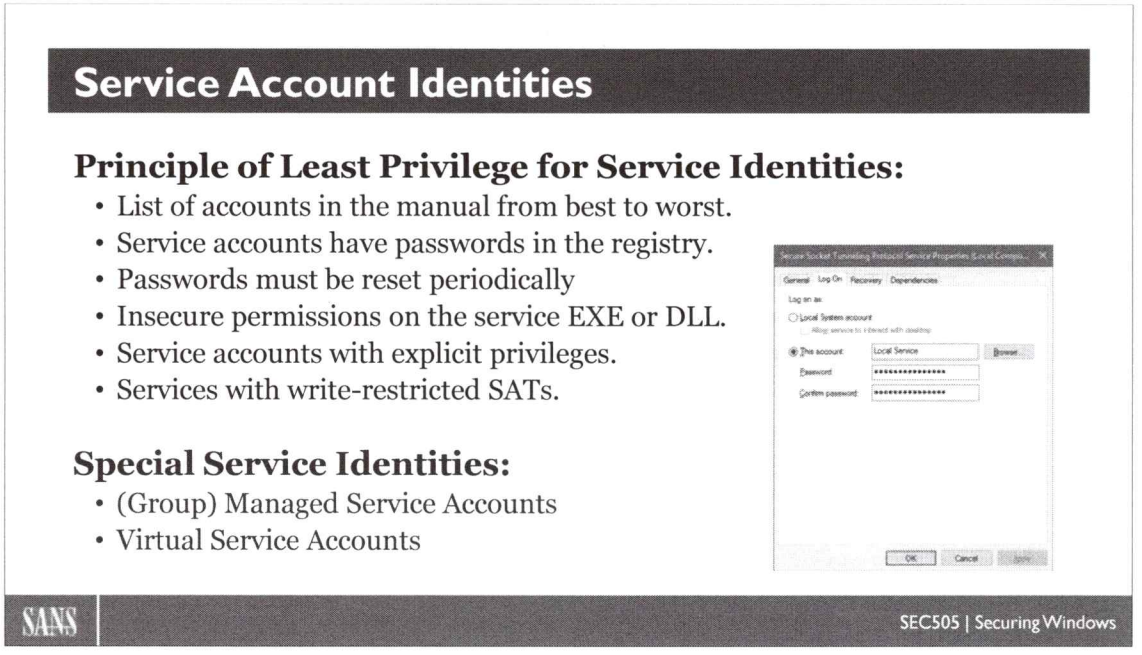
There is a script in your courseware files to manage these recovery options:

```
Get-Help -Full .\Set-ServiceRecoveryOptions.ps1
```

Group Policy for Service Recovery Options

If your server is a member of an AD domain, you can also use Group Policy Preferences to manage service recovery options. In a GPO, navigate to Computer Configuration > Preferences > Control Panel Settings > Services.

Server 2008-R2 and later support these GPO Preferences by default, but on Server 2003 and 2008 you must install a free add-on called the "Client Side Extensions" (KB943729).



Service Account Identities

A service process is like a scheduled task which runs at boot-up, on demand, or in response to a trigger event (trigger-start services were first introduced with Vista and Server 2008). The service process must have an identity, so just like with scheduled tasks, we have to worry about excessive power, especially when services have listening TCP/UDP ports exposed to the Internet.

Principle of Least Privilege for Service Accounts

When possible, choose the service identity with the least power which still allows the service to function normally. Here is the list from most preferred (no power) to least preferred (most powerful) at the bottom:

1) Local Service
2) Virtual Service Account (as Network Service)
3) Network Service
4) Local user account (no administrative group memberships)
5) Managed Service Account (as a standard domain user)
6) Domain user account (no administrative group memberships)
7) Local System
8) Local user account in the local Administrators group
9) Global user account in the local Administrators group
10) Global user account in the Domain Admins group
11) Global user account in the Enterprise Admins group

Often, you will have no choice for the identity of the service account, the product will have hard-coded requirements. But you may have a choice indirectly by upgrading to a

newer version of the product or choosing a competitor's product instead. In general, other things being equal, prefer products and product versions according to the list of preferences above. Services built for Server 2008-R2 and later are more likely to use less-powerful service identities as defaults.

To see the identity under which each of your current services run:

```
.\Get-ServiceIdentity.ps1
```

Or, if you wish to use the SC.EXE utility directly yourself:

```
Get-Service | foreach { sc.exe qc $_.name | select-string 'name'}
```

Service Account Passwords

We say that a service or scheduled task runs under an "identity" because sometimes that identity is not a standard user account. When a service or task runs as Local Service, Network Service or Local System, there is no password. But when a service runs as a local or global user account, there is a password to worry about. The service is configured with the password using the Services applet in Administrative Tools, the SC.EXE program, or other methods.

If the password configured for the service and the password actually assigned to the user account do not match, the service will fail to restart. This causes management headaches when, for example, a password is reset on a global service account in AD, but not updated on the thousands of machines with services that use that account. These problems make administrators very hesitant to ever change the password because it must be done at the same time in both places.

To change the password for the "*MyDaemon*" service on a remote machine (*Server47*):

```
sc.exe Server47 config MyDaemon password= TheNewLongPassphrase
```

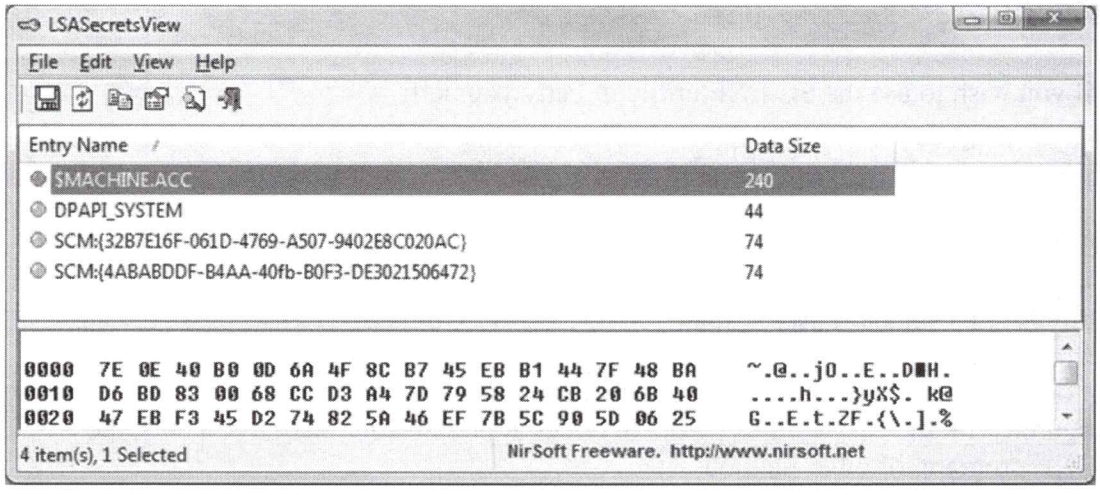
> **Tip:** When changing service passwords, it's sometimes easier to create a completely new user account for the service, update machines across the enterprise to use that new account without rushing, then delete the old service account once all instances of that service have been confirmed to be using the new account. Remember, SC.EXE is scriptable and can operate over the network.

Global service accounts are often added to the Domain Admins or Enterprise Admins group in AD, such as for some enterprise backup systems. Because these service accounts are so powerful, and because their passwords are so rarely changed, they are prime targets for hackers and malware.

When a service runs as a real user account, the password for this account is stored in a special part of the registry called the "LSA secrets" (HKLM\SECURITY\Policy\Secrets). Only a process running as Local System or which has the Debug Programs privilege can

see the LSA Secrets. There are many exploits which can meet these requirements, and there are free tools available to simply dump the LSA Secrets in plaintext afterwards.

For example, NirSoft's LSASecretsView tool can be used to easily view LSA Secrets (www.nirsoft.net) if the tool can be launched with the Debug Programs privilege. There is a command-line version of this tool as well for scripting.



So imagine a service installed on all/many systems that runs under the identity of a global user account in AD, and this user is a member of the Domain Admins group. If even a single machine in the LAN running the service is compromised, the account's password can be exposed. Now your adversaries have a Domain Admin password which is unlikely to be changed soon, if ever. Game Over.

There is another issue too. Security bulletin MS14-025 describes patches which, when applied, removes the GPO Preferences feature to manage the passwords of local accounts, scheduled tasks, mapped drive letters, database data source definitions, and also service accounts. This is because these passwords are stored in the GPO in an obfuscated form that allows attackers to extract the password in plaintext. It is best to apply the latest patches and to ignore these GPO-related password management features.

Insecure Service Binary Path

When a service runs, an EXE or DLL is typically executed or loaded. If the NTFS permissions on this EXE or DLL are not secure, it makes it easier for an adversary to replace or modify that binary in a malicious way.

To see the path to the EXE or DLL used by a service:

```
.\Get-ServiceIdentity.ps1 | Select Name,Path
```

The NTFS permissions on the C:\Windows, C:\Program Files, and C:\Program Files (x86) folders and their subdirectories are good. In general, service binaries should be

installed under one of these three folders, but any folder is acceptable as long as the permissions are as restrictive as on C:\Windows and its subfolders.

To list any service binaries not located under one of the three standard folders, run the Get-InsecureServiceBinaryPath.ps1 script in your courseware media for today.

```
.\Get-InsecureServiceBinaryPath.ps1
```

If there is no output, that is a good thing.

Service Account Privileges

Privileges are normally managed system-wide through Group Policy. On Vista, Server 2008 and later, it is also possible to define exactly which privileges each service process receives when it runs. This is independent of the system-wide global settings. The main tool for querying and reconfiguring these per-service privileges is SC.EXE.

Some privileges, like the Debug Programs privilege, are extremely powerful. It's not unusual for services to be granted these ultra-dangerous privileges, but any *changes* to service privileges is something to monitor.

To list the privileges for a service named "WinRM":

```
sc.exe qprivs winrm
```

You also have a PowerShell script, Get-ServicePrivileges.ps1, which acts as a wrapper for SC.EXE to list service privileges. It can list privileges for just one service, only the services whose names match a wildcard pattern (not regex), or for all services.

```
.\Get-ServicePrivileges.ps1 -ServiceName winrm

.\Get-ServicePrivileges.ps1 -ServiceName w*

.\Get-ServicePrivileges.ps1
```

To list the services which have been explicitly granted the Debug Programs privilege:

```
.\Get-ServicePrivileges.ps1 |
  Where { $_.Privileges -match 'SeDebugPrivilege' }
```

Write-Restricted Service Account SATs

Services can also be launched with write-restricted SATs, which means that any resource which the service attempts to access, the permissions on that resource must explicitly grant access to the service by name. This capability is one reason why Virtual Service Accounts (VSA) can be useful in partially "sandboxing" a service through permissions.

However, unless a service is designed for a write-restricted SAT by its original developers, it is very unlikely you will be able to change the restricted SID setting without breaking the service or causing error messages.

To see which services have restricted SATs:

```
Get-Service | foreach { sc.exe qsidtype $_.name }
```

In the output above, if the SERVICE_SID_TYPE is equal to "NONE" or "UNRESTRICTED", then the service runs with a normal, not-write-restricted SAT. If it is "RESTRICTED", then the service has a write-restricted SAT and that service's identity must be explicitly granted permissions to any resources, such as files, the service needs to access.

To list only the services which have write-restricted SATs:

```
.\Get-ServiceWithWriteRestrictedSAT.ps1
```

What Are Managed Service Accounts? (Server 2008-R2)

Managed Service Accounts (MSAs) are a replacement for global service accounts and are much easier to manage. MSA passwords are random, 120 characters long, and reset by default every 30 days using the same mechanism that computers use to update their own computer account passwords. In other words, you don't have to reset MSA passwords yourself, the reset is handled automatically. There are some issues, though, which might prevent you from using MSAs.

MSAs can only be used on Windows 7, Server 2008-R2 and later operating systems. In Active Directory, the schema must be upgraded to at least the 2008-R2 level, but there are no absolute domain functional level requirements.

MSAs are created and managed entirely with PowerShell cmdlets designed for MSAs (Get-Help *ADServiceAccount*). There are a fair number of steps required to create, configure and use MSAs with PowerShell, so the steps are not reprinted here. Please run "Get-Help -Full New-ADServiceAccount" to see the help for the cmdlet, but you'll also need to read the *Service Accounts Step-by-Step Guide* at http://technet.microsoft.com.

MSAs are stored by default in the AD container named "Managed Service Accounts", but they can be moved or created elsewhere with PowerShell if you wish.

An MSA is always associated with only one computer at a time on Server 2008-R2, it cannot be shared across multiple machines unless you have Server 2012 or later (see "What Are *Group* Managed Service Accounts?" below). But a single computer can host many services running under different MSAs (assuming the services support using an MSA, not all do).

MSAs can be configured for constrained delegation just like computer accounts, but the delegation settings are configured through PowerShell, not in the Delegation tab in the properties of the computer account to which the MSA has been linked.

MSAs can manage their own Service Principal Names (SPNs), but only if the domain functional level is Server 2008-R2 or better; at any lower functional levels, the SPNs must be managed by hand. An MSA's name is "*hostname\msaname*$", but MSAs have additional names too, like AD distinguished names and a FQDN for DNS.

An MSA password can be manually set by an administrator if necessary, but this will be rare and kind of defeats the purpose of having an MSA. MSA accounts cannot be locked out and cannot be used for interactive logons no matter what "Allow log on locally" rights have been configured. MSAs can be members of groups and these groups can be assigned permissions.

What Are *Group* Managed Service Accounts? (Server 2012 and Later)

On Server 2008-R2 it is not possible share a single MSA across multiple servers, such as on nodes in a cluster or on multiple IIS servers in a load-balanced farm. With Server 2012 and later, however, it is possible to create a *Group* Managed Service Account (GMSA) which does permit the use of the account across multiple servers.

To use a GMSA on a computer, that system must be running Server 2012, Windows 8 or later, and the computer must be joined to a domain which has at least one domain controller running Server 2012 or later. There are no particular domain or forest functionality requirements.

However, the steps required to implement a GMSA are more difficult than a regular MSA and cannot all be restated here; for example, you must first create a "root key" with the Add-KdsRootKey cmdlet and then wait at least 10 hours for replication to succeed before any domain controllers will allow the use of a GMSA. Please read the PowerShell help on the various *Kds* cmdlets (Get-Help *Kds*) and search for the *Getting Started with Group Managed Service Accounts* article on http://technet.microsoft.com.

Another benefit of GMSAs is that they may be used with scheduled tasks, which is not possible with the older MSAs.

To see a few of the PowerShell commands necessary to setup a GMSA:

```
ise .\Group-Managed-Service-Accounts.ps1
```

What Are Virtual Service Accounts?

Virtual Service Accounts (VSAs) have nothing to do with Managed Service Accounts (MSAs) or Group Managed Service Accounts (GMSAs). A VSA can replace the use of the Network Service identity in a way which allows the more precise assignment of permissions on local resources, just as with local service accounts, but a local service

account is not required when using a VSA, which saves the time and effort that might have gone into managing a local account's password.

VSAs can only be used on Windows 7, Server 2008-R2 and later. There are no schema or domain functionality requirements to use a VSA. VSAs do not exist in Active Directory or in any other accounts database for that matter. A new VSA does not exist anywhere in fact, it's really just a service configuration option.

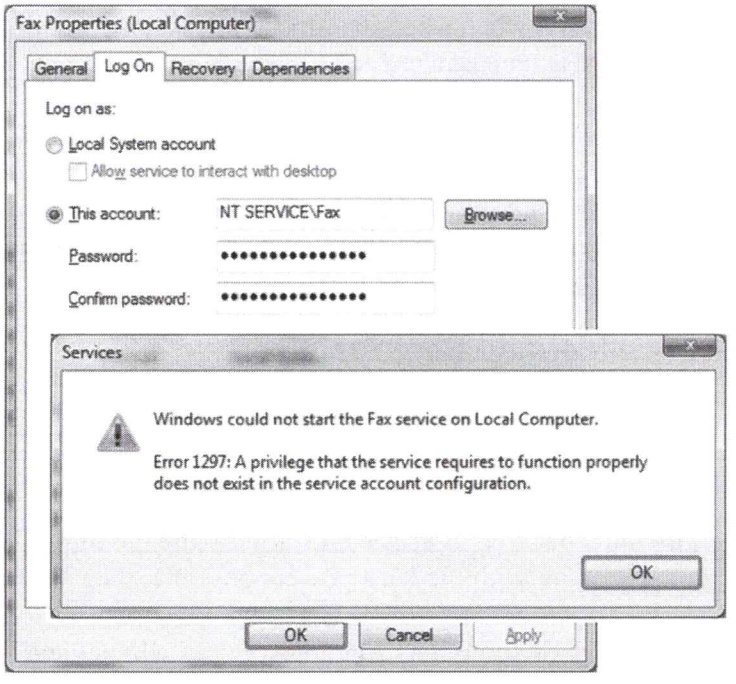
For example, SQL Server uses VSAs by default for many of its services and instances, such as "NT SERVICE\MSSQLSERVER" . The VSAs are created automatically for you during the installation process.

IIS on Server 2008-R2 and later uses something very similar to VSAs by default for each W3WP.EXE worker process launched for a web site's Application Pool. If the web site runs inside an Application Pool named "DefaultAppPool", then the W3WP.EXE worker processes launched to satisfy requests for that site will have a user name in Task Manager of "DefaultAppPool" as well. When you examine the advanced properties of an Application Pool, if the identity setting of the pool is "ApplicationPoolIdentity", then the name of the pool is used as the name of the VSA for those worker processes ("ApplicationPoolIdentity" is just a placeholder here, it just means "use whatever is the name of the pool as the VSA"). While other VSAs used with regular Windows services are identified as "NT SERVICE*ServiceName*", the IIS worker process identity is specified as "IIS AppPool\<*poolname*>", where <*poolname*> is the name of the Application Pool.

To list services which use a VSA:

```
.\Get-ServiceIdentity.ps1 |
   Where { $_.Identity -match 'NT SERVICE' }
```

To use a VSA for a service (and hopefully not break it), choose a service which runs in its own separate process, not a shared process, and which runs under the Network Service identity. Then use the SC.EXE command-line tool or the Services applet to configure that service to run as "NT SERVICE*ServiceName*", where *ServiceName* is the exact internal name of the service; for example, the Fax service runs as Network Service in its own separate process, so its VSA name would be "NT SERVICE\Fax". Keep in mind, though, that this could break the service (like it does for the Fax service) if there are local NTFS permissions it requires which have not been granted to the its VSA identity. In truth, you will rarely be able to convert a service to use a VSA if that service has not already been designed by its coders to run as a VSA. VSAs are much less useful to us than Managed Service Accounts, and even MSAs are sometimes a pain.



From a security perspective, a service running as a VSA is running as Network Service, but with a *locally* known name for the sake of assigning privileges and permissions to *local* resources only. When a service running as a VSA authenticates over the network to another computer, the service authenticates as "*domain\hostname*$", just like Network Service does, not "*domain\vsaname*$", like MSAs do. The VSA's unique service name is not transmitted to other computers and, even if it were transmitted, it wouldn't have any meaning to other computers since the VSA name is not like a well-known SID. An authenticated global SID or an AD attribute claim can be referenced across multiple domain-joined computers because these SIDs/claims exist in Active Directory for everyone, while a VSA, on the other hand, exists only on one machine for its own local use.

A VSA is not a replacement for Local Service, even though it would work, because this would elevate the powers of the service. By definition, if the service runs fine under the Local Service identity, then it would be unnecessary and bad for security to give this service the ability to authenticate over the network.

A VSA is not a replacement for Local System because it would likely prevent the service from running normally. A VSA would have far fewer privileges than Local System.

A VSA is like a mash-up between Network Service and a local user account. It can authenticate to remote systems under a "*domain\hostname*$" global identity, but the service can be kept distinct from other local services running as Network Service when it accesses local resources, like NTFS files. Permissions on local resources can grant one local service access and deny access to the other local services because the first service uses a VSA identity while all the others, let's assume, are all running uniformly as just

Network Service. (Incidentally, when a service runs as a VSA, there will be a key for it in the LSA Secrets, but there is no password.)

So what's the point of VSAs then? Like on IIS and SQL Server, you can assign permissions to VSA identities for securing access to *local* resources, and local resources only, such as files on an NTFS hard drive. Other than that, a VSA is little different than the Network Service identity, i.e., it's just a standard account lacking special privileges which can still authenticate over the network when necessary.

Choosing A Service Identity: Best Practices

Follow the top-down list of service identity preferences described earlier when you actually have a choice of identity, product, or product version. You often won't get a choice.

If your service identity must be a real user account, follow the list of preferences above to choose a local or global account with the least powers possible, then use these guidelines:

- Maintain an inventory of computers running services which have user account identities that are members of administrative groups, such as local Administrators and Domain Admins. The inventory should at least include the computer name, service name, user account name, and required group memberships.

- Choose a long, random passphrase (25+ characters).

- Change the passphrase following the same schedule you enforce for other similarly-privileged accounts, and for the same reasons. Using Group Policy, scheduled tasks with custom scripts, third-party service management systems, or Managed Service Accounts (MSAs) will help to automate the work. In fact, without automation no one will ever follow this best practice, so some form of automation is required.

- Review the groups to which the service account belongs and remove it from any which are not necessary. When a vendor says that its service account must be a Domain Admin, the account might only require membership in the Administrators group on just the machines where the service is installed or to which the service authenticates; but these group memberships can be managed on an OU basis through GPO, the account doesn't actually require Domain Admins *per se*.

- If possible, limit the number of machines where the service is installed, especially when the computers and the service(s) under consideration are exposed to the Internet through the firewall.

- For global service accounts, as much as possible restrict their "Access this computer from the network" and "Allow log on through Remote Desktop Services" rights using Group Policy on other machines. Deny service account

logon rights on all the computers where there is no need for the service account to ever authenticate, especially on the high-value computers.

A service's privileges can be constrained by special registry edits using SC.EXE without changing the system-wide privilege settings. So, one way or another, try to remove Debug Programs and the other dangerous privileges from services which do not require them. If a service already has a list of needed services (sc.exe qprivs *servicename*), then the list is probably not editable; but if that list is not configured, then copy the privileges from another similar service and see if this service can make do without the dangerous ones (the "dirty half-dozen privileges" discussed earlier).

Install, Update Or Remove Other Applications

Every binary or script for the operating system and the applications installed on top of the OS can be potentially exploited. Any file can potentially be a part of a discovered vulnerability. But we need these files and we cannot predict which one(s) will be found vulnerable tomorrow. What to do?

Install Necessary Applications and Keep Them Updated

On top of the roles you install, you will have other Microsoft and third-party applications installed too, such as web applications, RPC services, clustering agents, anti-virus software, etc. Install whatever is necessary for your users, naturally, since this is a business decision, not really a security decision. Keeping those products up to date with the latest security fixes, that of course is an important security issue (and one we have already discussed).

Don't Install Unnecessary Software In The First Place

In general, don't install applications that the server doesn't require in order to fulfill its role. If unnecessary software is discovered, then uninstall it.

Be especially wary of developers, webmasters, auditors and other admins logging on locally at the server or via RDP and launch locally-installed apps. For example, no one should be logging on locally at a server to browse the Internet or check e-mail, which should be too obvious to say, but it's often done in the real world. If extra tools must be installed on the server, at least apply restrictive NTFS permissions to them so that only Local System and the Administrators group have access.

Operating System Tools

Potentially dangerous files should either be deleted or have permissions set which restrict access to them. In general, it's better to apply restrictive NTFS permissions than to delete the files. When you next apply a patch, the files might be copied back again and forgotten. And the Windows Resource Protection feature (also called "Code Integrity") gets cranky about deleting these OS files too. If you are determined to delete the unneeded files, use a PowerShell script so that you can do so consistently and easily again in the future, and consider replacing the files with identically-named empty files. You'll need to run this script at least weekly, perhaps with a scheduled job. Expect it to be rough going.

A partial list of potentially dangerous executable files includes the following (not all are installed by default). Since you will need to use these to manage your servers, consider burning known-good copies onto an administrative DVD or flash drive (with a read-only switch) that you can carry with you from box-to-box when necessary. This same DVD could have your other favorite security, troubleshooting and forensics tools of course. Alternatively, copy the DVD to a shared folder to which only administrators have access and run the tools over the network; just take care to secure the folder, keep it updated, and regularly probe with an anti-virus scanner.

The most dangerous executable is CMD.EXE. In general, any tool which provides arbitrary command execution, remote administration, and network file transfer should be viewed suspiciously. A security template can lock the permissions down on these tools, but on Server 2008 and later, the default permissions should be adequate. Perhaps more importantly, the NTFS audit settings on these files should record deletions, modifications and execution of these files by the Everyone group.

- **APPCMD.EXE**
- ARP.EXE
- AT.EXE
- ATSVC.EXE
- ATTRIB.EXE
- CACLS.EXE
- CLIPSRV.EXE
- **CMD.EXE**
- **COMMAND.COM**
- **CSCRIPT.EXE**
- **DEBUG.EXE**
- DIALER.EXE
- EDIT.EXE
- EDLIN.EXE
- FINGER.EXE
- **FTP.EXE**
- HYPERTRM.EXE
- HTIMAGE.EXE
- IMAGEMAP.EXE
- IPCONFIG.EXE
- ISSYNC.EXE
- MSIEXEC.EXE
- NBTSTAT.EXE
- **NET.EXE**
- **NET1.EXE**
- **NETSH.EXE**
- NETSTAT.EXE
- NSLOOKUP.EXE
- PING.EXE
- POLEDIT.EXE
- POSIX.EXE
- **POWERSHELL.EXE**
- QBASIC.EXE
- QFECHECK.EXE
- RCP.EXE
- RDISK.EXE

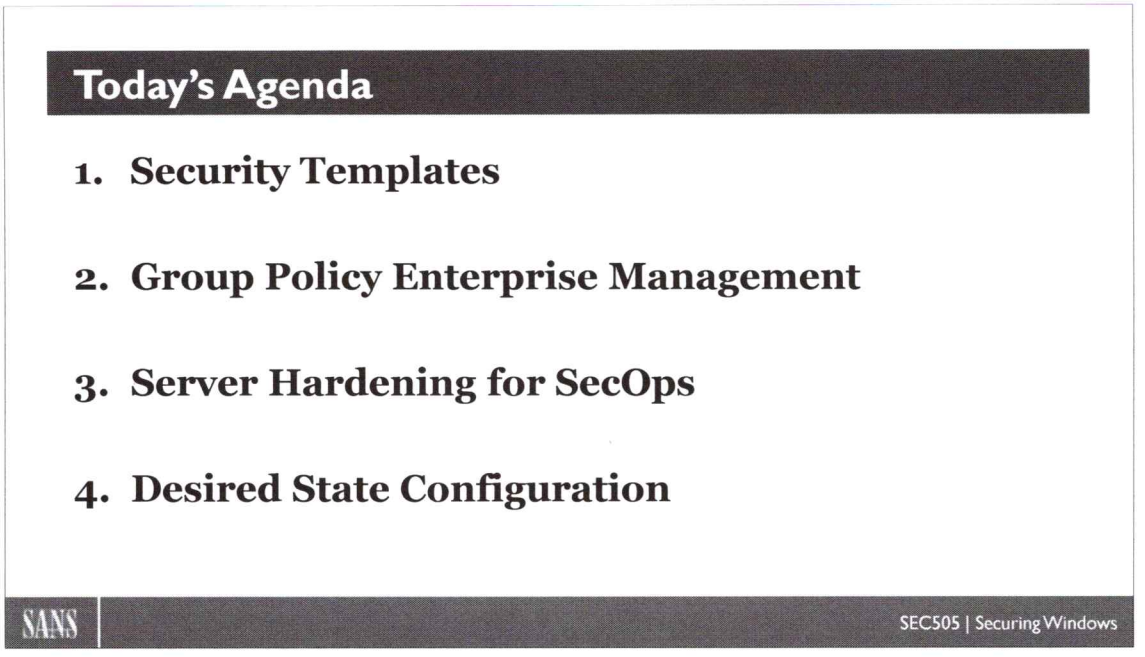
- REG.EXE
- REGEDIT.EXE
- REGEDT32.EXE
- REGINI.EXE
- REGSVR32.EXE
- REXEC.EXE
- ROUTE.EXE
- RSH.EXE
- RUNAS.EXE
- **RUNDLL32.EXE**
- RUNONCE.EXE
- **SC.EXE**
- SCHTASKS.EXE
- SECFIXUP.EXE
- SYSEDIT.EXE
- SYSKEY.EXE
- TELNET.EXE
- **TFTP.EXE**
- TRACERT.EXE
- TSKILL.EXE
- UNINST.EXE
- **WSCRIPT.EXE**
- **WMIC.EXE**
- XCOPY.EXE

Don't Install Any Resource Kits or SDKs

Resource Kits and Software Development Kits (SDKs) are loaded with tools and scripts which are too dangerous to leave installed on production servers. Your developers may, of course, install them on staging or QA servers, as long as they are not accessible from the Internet. Your webmasters and coders will fight you on this, but remain firm.

Favorite Admin Tools

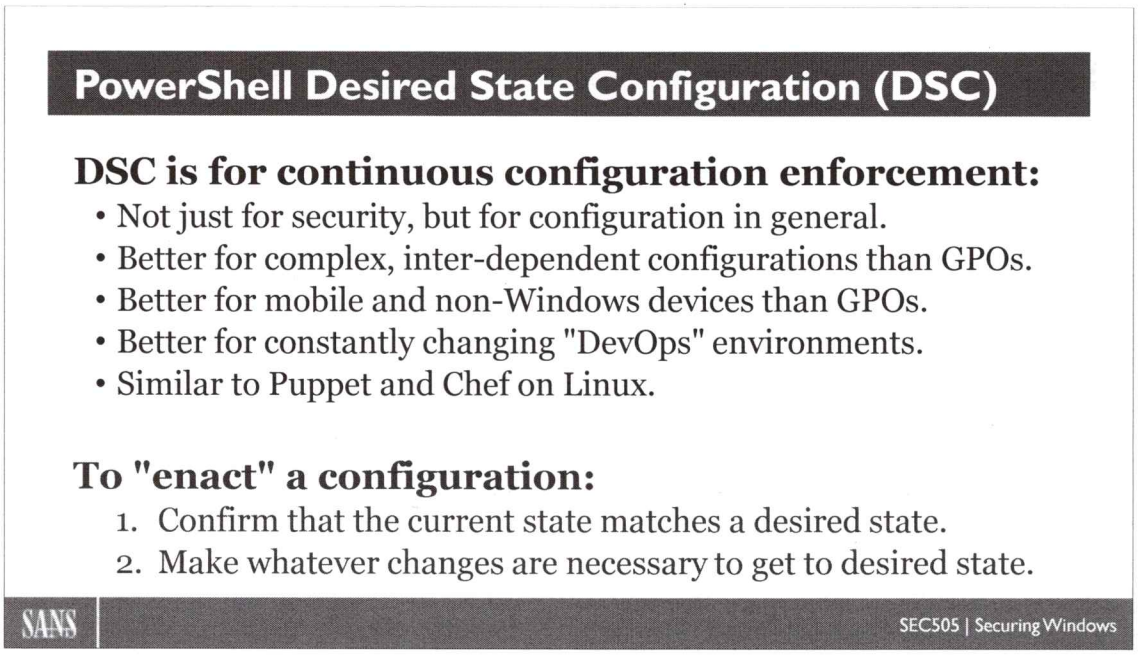
It's a strong temptation to simply copy one's favorite administration tools to all the servers, but you have to assume that hackers will find (and like) these tools too. Also, when you suspect that a machine has been compromised, you can't trust the binaries on its hard drives anymore. All in all, it's best to burn all of one's favorite tools (and known-good OS tools as well) to a CD/DVD that can be carted around with you. A USB drive will work too, as long as it has a read-only switch that is left in read-only mode when plugged into a suspected compromised machine. You can also purchase a forensics "bridge" device which prevents write access to a USB drive with the press of a button on the device. If the tools can run over the network from an SMB shared folder, then permissions can be set on that folder to protect it, along with SMB encryption, integrity monitoring, anti-virus scanning, and NTFS logging to keep track of access.



Today's Agenda

Using Server Manager and PowerShell to manage roles and features is great, but not good enough. We don't really want to have to write all of our own scripts. We want these automation tasks to scale to thousands of machines, not just a few hundred. And there is much more than just roles and features that need to be managed, there are all the configuration settings and application files too. We need something more comprehensive.

The future of security and application configuration management on Windows is PowerShell Desired State Configuration (DSC). If you are familiar with Puppet or Chef for Linux, then DSC will feel very familiar.



Desired State Configuration (DSC)

Desired State Configuration (DSC) is for the continuous enforcement and monitoring of device configuration. DSC goes beyond just registry values and OS settings. DSC can be used to sync folders over the network, (un)install server roles, configure server roles, manage group memberships, install virtual machines, and much more.

Why DSC?

Group Policy is great for managing domain-joined computers inside the LAN, but not good for managing stand-alone computers or for managing roaming devices out on the Internet (especially non-Windows devices). Group Policy is great for managing individual settings which are independent of each other on one computer, like registry values, but not ideal for managing complex combinations of settings that must be orchestrated as a consistent set across multiple interacting computers, such as a web application on a load-balanced farm of IIS and database servers.

In the past, OS settings were the responsibility of the admins, while application-layer settings were the responsibility of the developers. But this distinction is blurring. The term "DevOps" expresses the idea of how developers and operations people must work together because of their overlapping responsibilities. This is important for the trend towards cloud computing and virtualization too.

In the past, the security configuration of a computer was relatively static. A "gold image" was used to build the box, then maybe Group Policy was used to reinforce a limited number of security settings, but beyond that it was mostly a matter of "Configure and Hope" that nothing would change and that nothing would need to be changed to combat security threats. But modern threats requires both continuous enforcement and

continuous monitoring of security settings. This is especially hard to do when some of those security settings, like firewall and whitelisting rules, need to be changed on a weekly or daily basis. With "Windows as a Service", new application versions, roaming users, new zero day exploits, new malware variants, etc., the change is constant, and our configuration systems must be able to keep up.

For all the above reasons, Microsoft has developed DSC. And Microsoft is not alone.

Is This A PowerShell Version Of Puppet Or Chef?

If DSC sounds similar to Puppet (puppetlabs.com) or Chef (www.chef.io), this is no accident. Puppet and Chef are especially popular for managing Linux servers, but the *idea* of how Puppet and Chef work, the management *strategy* they represent, is the same as DSC and is the most important thing to grasp. So, why use PowerShell, why not just use a third-party product like Puppet or Chef?

For the sake of DSC, PowerShell gives us 1) a language for expressing and managing the configurations we want to enforce and monitor, 2) a remoting capability to securely manage DSC in a scalable way, 3) a scripting language with which millions of IT people are already familiar, and 4) PowerShell is already built in for free. Today, PowerShell is a mandatory skill for Windows administrators, especially when integrating with Azure and Office 365, so we don't really have a choice anyway. Some familiarity with PowerShell is required, even if only the basics.

Of course, you can use many management tools together at the same time, with different tools on different types of machines, such as on workstations and servers. So it's not an either/or choice between PowerShell and other products; they can be used together. We will also see that DSC is in fact independent of and larger than PowerShell. Puppet and Chef can use DSC too.

Terminology: What Does It Mean to "Enact" a Configuration?

A note about terminology is useful here. When a DSC configuration is used to define a configuration state, it declares what the final end result should look like, and nothing else. DSC is not a programming language. DSC does not define *how* to reconfigure a machine to match a desired state, DSC only defines *what* that desired state should be. By contrast, a PowerShell script would have to include a series of commands and progress checks while working towards some end result. If we are writing the PowerShell script, we have to do all the work. If we are writing a DSC configuration in PowerShell, on the other hand, we mostly just say what we want, which is much easier for us.

The DSC term for taking a defined configuration state and confirming that a system matches that state, or making any changes necessary to achieve that configuration state, is called "enactment." Hence, to "enact" a configuration means to 1) examine the current configuration, 2) compare the current state with the desired configuration, 3) log the (mis)matching status of each defined element of that configuration, 4) invoke the resources required to install, uninstall, delete, copy, create, set, or otherwise do whatever

is necessary for the current configuration to come to match the desired configuration, and then 5) log the success or failure of any of these changes.

DSC Commands

There are several PowerShell commands related to DSC.

To see a list of DSC-related PowerShell commands:

```
Get-Command -Module PsDesiredStateConfiguration
```

Just for reference, here are short descriptions for the important DSC commands. There are other commands related to DSC too, but these are the most common. Note that several of these commands require PowerShell 5.0 or later.

Command	Description
Find-DscResource	List available DSC resource modules from registered, online galleries, such as the PowerShell Gallery.
Get-DscConfiguration	Display the current DSC configuration, if any.
Get-DscConfigurationStatus	Display a history of DSC events.
Get-DscLocalConfigurationManager	Display LCM operational settings, like refresh.
Get-DscResource	List DSC resources and their properties.
Install-Module	Install a DSC resource module (or any module).
Remove-DscConfigurationDocument	Deletes MOF files, such as Current.mof.
Restore-DscConfiguration	Roll back to previous MOF configuration.
Start-DscConfiguration	Compile a DSC script to one or more MOF files.
Test-DscConfiguration	Check compliance with a MOF configuration.
Update-DscConfiguration	In pull mode only, refreshes the configuration.

Glossary of DSC Terms

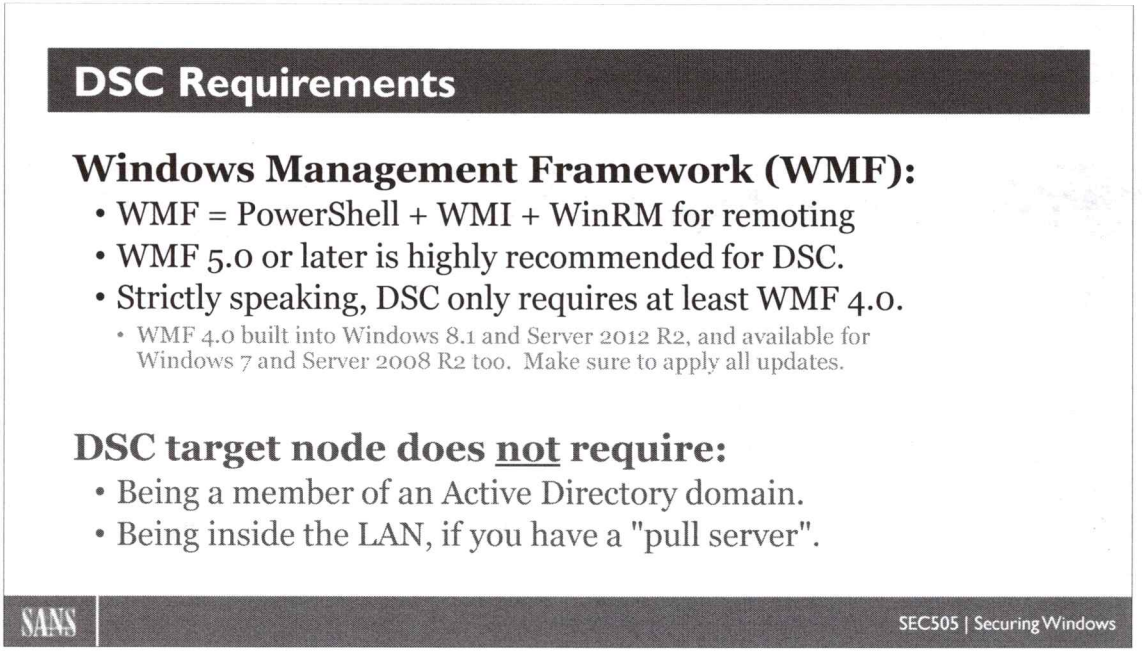
Just for reference, here is glossary of DSC terms.

Term	Definition
Config Document	A MOF file, such as Current.mof or Previous.mof.
Configuration	PowerShell code which declares the final, desired configuration state of a machine or application; similar to a function.
Local Configuration Manager (LCM)	Windows component which uses resources to confirm or enforce desired configuration settings. LCM is the main engine of DSC.
MOF File	Managed Object Format (MOF) file used by WMI to confirm or enforce DSC changes; a DSC script compiles to a MOF file, but a MOF file is not binary, it is a text file with special syntax.
Node	A computer, device, service or other DSC management target.
Pull Server	HTTPS, HTTP or SMB server with configuration MOF files.
Resource	Module which performs the work of enacting changes.
WMF	Windows Management Framework, which includes PowerShell.

DSC File Types

DSC uses several types of files, identified by their file name extensions. Virtually all of these files are just plain text files that can be edited in Notepad or ISE.

File Name Extension	File Description
.meta.mof	Configuration MOF for the Local Configuration Manager (LCM) itself.
.mof	A textual MOF file, "compiled" from a PowerShell script or other suitable language, containing resource commands to enact a particular configuration on a particular node.
.mof.checksum	Used in pull mode to detect MOF file changes.
.ps1	A PowerShell script.
.psd1	A PowerShell data file, such as for a DSC configuration data file.
.psm1	A PowerShell script module, used to create a custom resource.
.schema.mof	Schema for a custom DSC resource.



DSC Requirements

Support for Desired State Configuration (DSC) requires Windows Management Framework (WMF) 4.0 at a bare minimum, but really WMF 5.0 is the practical minimum for performance, stability and manageability reasons.

WMF is a set of related technologies for remote management, the most important of which is PowerShell itself, but WMF also includes the Windows Management Instrumentation (WMI) service and the Windows Remote Management protocol (WinRM). The terms "PowerShell" and "Windows Management Framework" are mostly synonymous in common usage.

To check your WMF version, see the PSVersion property of the $PSVersionTable object:

```
$PSVersionTable
```

WFF 4.0 comes with Windows 8.1 and Windows Server 2012 R2 already. Make sure to apply all the latest updates, or at least the rollup update number KB3000850 .

WMF 4.0 is available for Windows 7 SP1 (KB2819745), Server 2008 R2 SP1 (KB2819745), and Server 2012 (KB2799888). However, on Windows 7 and Server 2008 R2, you must install the .NET Framework version 4.5 before installing WMF 4.0.

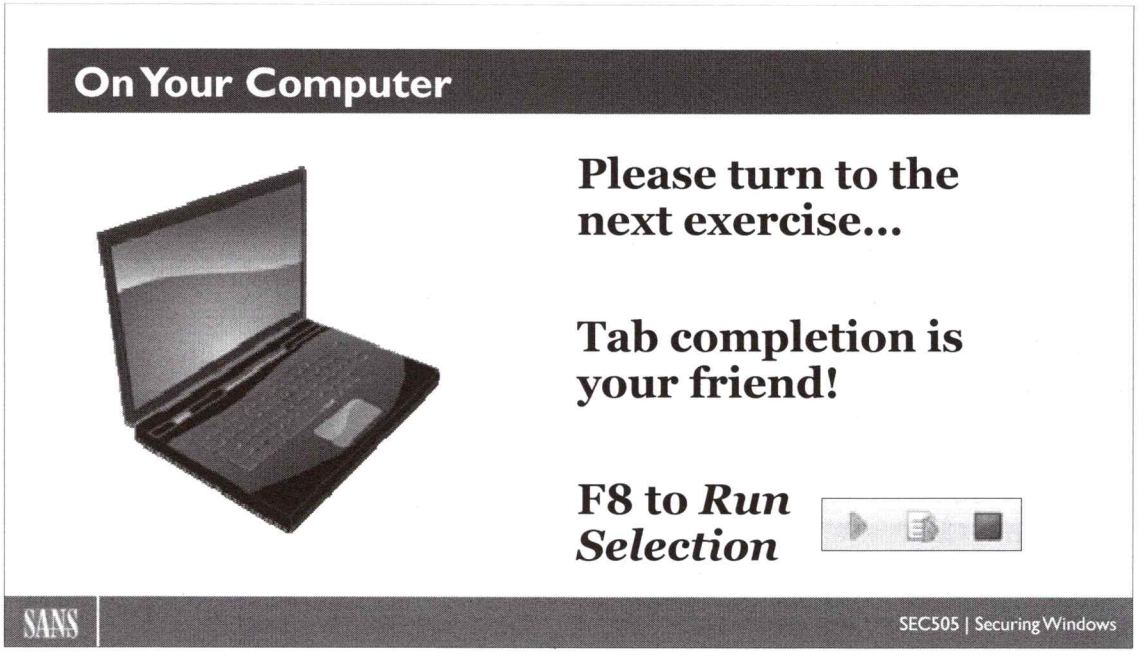
WMF 4.0 is not available for Windows 8.0; you must upgrade this unloved OS.

DSC support is built into WMF 5.0 on Windows 10 and Server 2016. WMF 5.0 can be downloaded and installed on Windows 7 SP1, Server 2012 and Server 2012 R2 too.

In general, it is best to upgrade to the latest version of PowerShell/WMF, but for DSC in particular it is best to have at least WMF 5.0 for performance, stability and the availability of very useful enhancements over WMF 4.0. Do not deploy DSC in production on machines that have anything less than WMF 5.0.

Note that DSC does not require Active Directory. DSC works on stand-alone computers. Strictly speaking, DSC does not require a standard PC or laptop computer either, and it even does not require a Windows operating system! (See MOF discussion below.)

A DSC configuration may also require particular helper modules to be available too, but let's discuss these items later when we talk about DSC resources.



On Your Computer

The purpose of the lab is to get a quick taste of DSC first before we dive into the details. Some of the steps will make no sense, but that's OK, the discussion will follow the lab.

Go to the C:\SANS\Day2-Hardening\DSC folder:

```
cd C:\SANS\Day2-Hardening\DSC
```

Launch the graphical REGEDIT.EXE application:

```
regedit.exe
```

In REGEDIT.EXE, navigate to HKEY_LOCAL_MACHINE\SOFTWARE. Notice that there is not a subkey named "AAANewKey" underneath SOFTWARE.

Briefly examine the contents of the RegistryTest.ps1 script (and keep the tab open):

```
ise .\RegistryTest.ps1
```

Note: Notice the period at the beginning of the next command for dot-sourcing.

Dot-source that script to create a new configuration function in the Function:\ drive:

```
. .\RegistryTest.ps1                                    #See the period?
```

Notice that you now have a new special type of function in the Function:\ drive:

```
dir function:\TestConfig
```

Note: Ignore any warning messages with the next command.

Run that new configuration function by calling its name (run the function, not the script):

```
TestConfig
```

When you ran the TestConfig function, the function created a new subdirectory under the folder you are in right now. The new subdirectory has a name which is identical to the name of the function. Both the function and subdirectory are named "TestConfig."

See a listing of subdirectories in the current working folder:

```
dir -directory
```

Running the TestConfig function also created a new text file in the \TestConfig subdirectory named "LocalHost.mof", hence, the full path to the MOF file created by the function is C:\SANS\Day2-Hardening\DSC\TestConfig\LocalHost.mof.

See a listing of files in the .\TestConfig subdirectory:

```
dir .\TestConfig
```

Briefly examine the contents of the LocalHost.mof file (it's like a security template):

```
ise .\TestConfig\LocalHost.mof
```

Close the tab in ISE showing the MOF file.

Apply that MOF file to your computer to reconfigure your computer to match it:

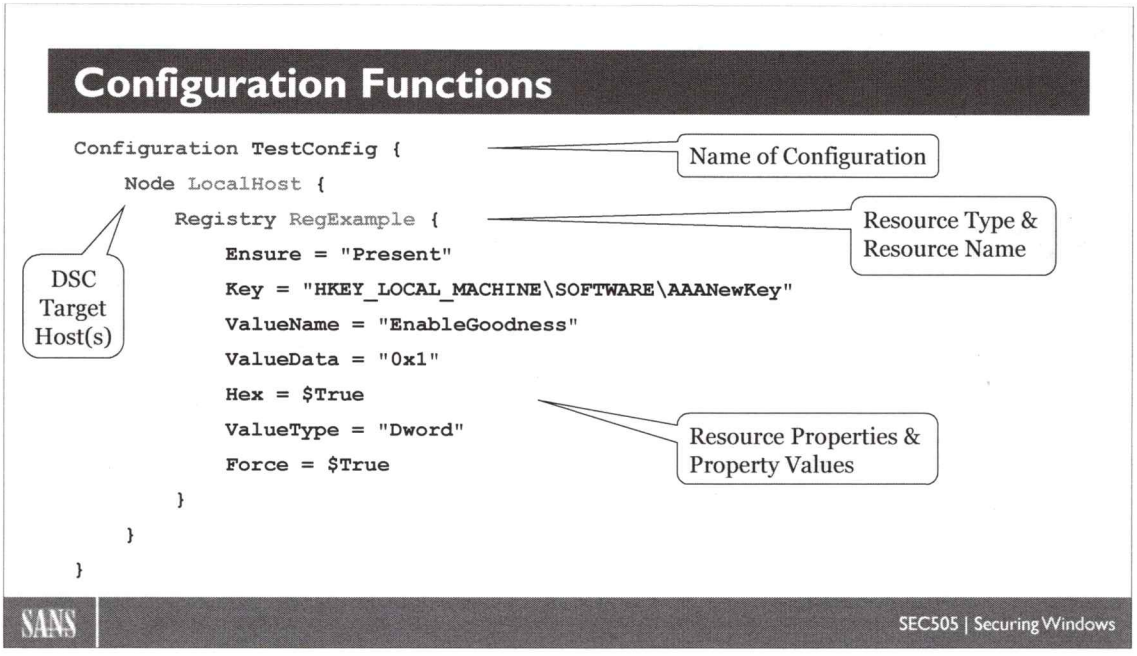
```
Start-DscConfiguration -Path .\TestConfig -ComputerName localhost
```

In REGEDIT.EXE, pull down the View menu and select Refresh (or just press F5). Notice that there is now a new key named "AAANewKey" under SOFTWARE:

HKEY_LOCAL_MACHINE\SOFTWARE**AAANewKey**

And inside that key there is a new value named "EnableGoodness", set by the MOF file.

But what does all this craziness mean? What's going on here! Let's talk about it…



Configuration Functions

Imagine you wish to use DSC to manage a particular registry value. Here is some PowerShell code to create a DSC configuration:

```
#.\DSC\RegistryTest.ps1

Configuration TestConfig
{
    Import-DscResource -ModuleName PSDesiredStateConfiguration

    Node LocalHost
    {
        Registry RegExample
        {
            Ensure = "Present"
            Key = "HKEY_LOCAL_MACHINE\SOFTWARE\AAANewKey"
            ValueName = "EnableGoodness"
            ValueData = "0x1"
            Hex = $True
            ValueType = "Dword"
            Force = $True
        }
    }
}
```

Executing the code above does not set any registry values, but it does create a new object in the Function:\ drive named "TestConfig" with a command type of "Configuration", which is very similar in concept to a function.

Importantly, the code is "declarative" not "imperative", i.e., the code says *what* we want (a registry value set in a particular way) but does not say *how* to get it (by including a bunch of commands to check to see if any changes are necessary, to modify the registry, handle any errors, write to a log file, etc.). Unlike normal PowerShell functions, DSC configuration functions don't do much, they just express *what* we want to have done.

Configuration

In the code above, "Configuration" is a keyword, just like "Function" is a keyword. It is a special command to PowerShell to create a new DSC configuration object in the Function:\ drive named "TestConfig." The name of the DSC configuration object, "TestConfig", is not a fixed keyword, it is an arbitrary name you invent, so it can be anything (it may include letters, numbers or the underscore character, and it's not case-sensitive). By analogy, think of a DSC configuration as a special type of function.

Node

In the code above, "Node" is also a keyword, and usually one or more nodes are defined inside a DSC configuration. A node is any target of a DSC enactment action, such as a computer, tablet, phone, or any other DSC-capable target. The name which follows the "Node" keyword uniquely identifies the target device, such as the hostname or NetBIOS name of a Windows computer. Hence, the node name is not totally arbitrary, it does have to identify some kind of identifiable target. The node name will usually not be hard-coded like in the example above; usually, the target node will be a variable so that the node name(s) may be fed in as an argument. We will see this later.

Resource Type (Registry)

In the code above, "Registry" is the name of a type of DSC resource. A DSC resource is a type of thing which may be managed by DSC. There are many DSC resource types, such as Registry, File, Log, Group, Package, User, Service, and so on. Hence, in the configuration example above, we happen to have the word "Registry", but it may have been the name of some other available resource type. What is arbitrary in the above example, though, is the name which follows the resource type, in this case, "RegExample", which is simply a name you choose to identify a set of configuration settings to enact. The resource name must be unique within the node definition.

> **Note:** None of these DSC keywords or names are case-sensitive.

Resource Properties (Ensure, Key, ValueName, Etc.)

In the code above, under the Registry resource, there is a list of properties: Ensure, Key, ValueName, ValueData, etc. These are called "Resource Properties", and the names of these properties come from the code which implements the resource in PowerShell, hence, different resource types will have different resource properties appropriate to that type. For a Registry resource type, it's appropriate to have properties for the path to the registry key, the name of the registry value, the hex DWORD data, etc. But for a User resource type, there would be different properties, such as user name, password, full name, description, etc. The only way to know what resource properties are available is to consult the documentation for the DSC resource.

> **Tip:** In the ISE script editor, place your blinking cursor inside of a DSC configuration and press Ctrl-Space. This will bring up a graphical list of keywords for that part of the configuration. (Requires WMF 5.0 or later.)

Resource Property: Ensure
Many resources have a property named "Ensure", and the Ensure property usually takes one of two legal values: Present or Absent. Choosing "Present" will normally install, create or define something, if that something is not already installed, existing or defined. Choosing "Absent" will normally uninstall, remove or clear something, if that something currently exists or is installed.

Resource Property: DependsOn
Though it's not in the example code above, it's important to mention another special resource property you will frequently see in other examples: DependsOn.

The "DependsOn" resource property is used to define dependency or prerequisite relationships between resources defined for a node. The chain of DependsOn relationships must not be circular, e.g., A depending on B and B depending on A, would be circular. The Local Configuration Manager (LCM) will enact the resource changes in the proper dependency order. The syntax for DependsOn is:

DependsOn = "[<*ResourceTypeName*>]<*ResourceName*>"

For example, if you are writing a configuration, you might define a DSC node with multiple resources (named "ResourceFirst" and "ResourceSec"), but where the second resource (ResourceSec) cannot be enacted until after the first resource (ResourceFirst) has been successfully applied. In this case, in the properties of the second resource (ResourceSec) you would have a property that would look something like this:

DependsOn = "[Registry]ResourceFirst"

Don't worry if you're not getting this all now, more examples are needed.

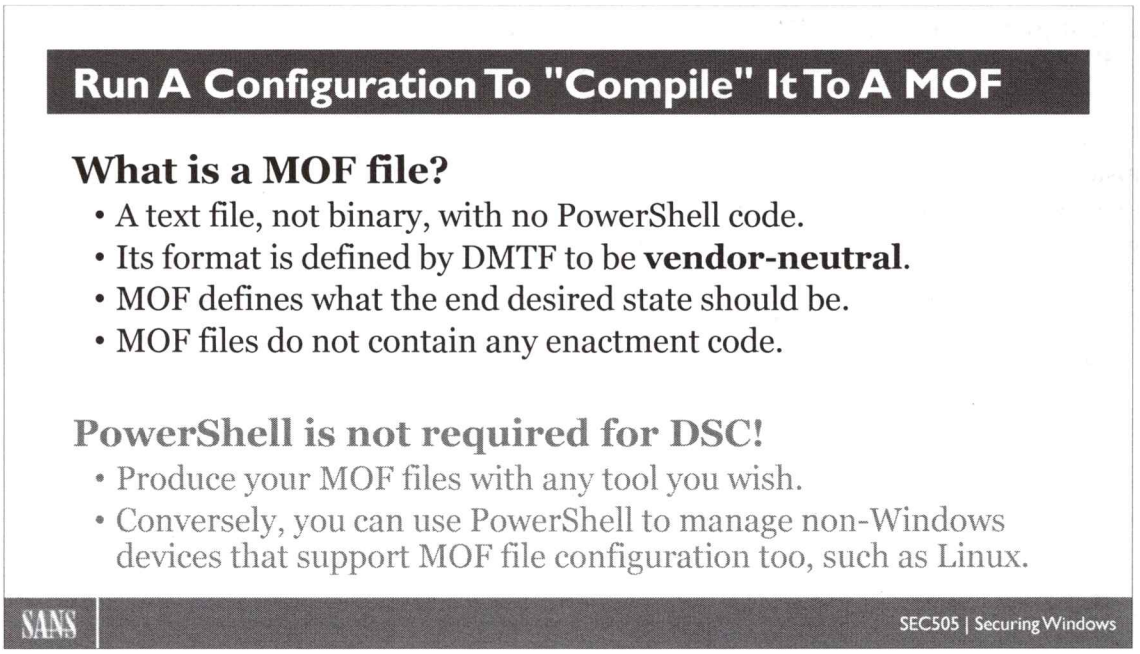
Is A DSC Configuration Really A Type Of Function?

Yes and no. A configuration behaves like a function in that 1) the name of the configuration must be invoked or called in order for it to execute, 2) a configuration may be called repeatedly, 3) a configuration defined in a script must be above/higher in the script than any calls to it, 4) the Param keyword may be used to define input parameters to the configuration, just like in an advanced function, 5) configurations stored in a module are imported and called just like functions from a module, and 6) once created in the current scope, a new configuration is visible in the Function:\ drive.

```
dir function:\ | where { $_.CommandType -eq 'Configuration' }
```

But, strictly speaking, a DSC configuration is not a function. Nonetheless, this manual will often refer to "configuration functions" to simplify and clarify.

In summary, a PowerShell script for DSC must have at least one configuration defined inside it, but a single script or module may define many configurations. A configuration may have zero, one or more nodes defined in it; for example, a multi-tier web application with multiple load-balanced servers might all be defined as nodes in a single DSC configuration because of their inter-dependencies. And within a node there will usually be many resource items defined, such as many registry values, files to copy, roles to install, etc.



Run A Configuration To "Compile" It To A MOF

Running a PowerShell script which only has a configuration defined inside of it (and nothing else) will create a configuration function in the Function:\ drive (and do nothing else). Just like a function, a DSC configuration is first created, then explicitly executed or "called" afterwards. This is why you might dot-source a script with a DSC configuration inside of it, then run the configuration; or, even more likely, you would import a module with multiple configurations defined inside it, then run one, some or all of those configurations. So, creating a DSC configuration results in something new in the Function:\ drive, but nothing more at this point.

After a DSC configuration is created, it can be run. Now, when a configuration function is run, something strange happens. The output of a configuration function is 1) a new folder on the hard drive and 2) the creation of one or more MOF files inside that folder.

The folder created by a configuration function is named after the configuration itself; hence, if the name of the configuration is "TestConfig", then there will be a new folder with the same name (.\TestConfig). Where is this folder created? When running the examples manually, as we are doing now, the folder is created in whatever present working directory ($pwd) you happen to be in inside of PowerShell at the moment. So make sure you are in the correct directory first before running any of these examples. (When you are later using DSC in real life, it won't really matter where this folder is located because you'll move or copy it to an HTTPS or SMB server afterwards for mass distribution to your target nodes. More on this later.)

Hence, after creating a DSC configuration function, we run that function, and we get one or more MOF files. What are MOF files? Why do we care?

What Is A MOF File?

A text file with an .mof extension is a MOF file. "MOF" stands for Managed Object Format. The MOF file format is defined by the Distributed Management Task Force (DMTF), not Microsoft. The DMTF also defines the Common Information Model (CIM) standard. The goal of the DMTF with both CIM and MOF is the same: define vendor-neutral standards for remote administration of any type of network-attached device from any other vendor's management application running on top of any operating system. Microsoft and Red Hat are both members of the DMTF. Hence, MOF is not a proprietary Microsoft invention, and MOF files are not just for Windows.

Note: You can read more about MOF at the DMTF's web site: www.dmtf.org.

A MOF file contains only text. You could create and edit a MOF file in Notepad, if desired, but the syntax is not very friendly. Fortunately, you don't need to know this syntax in order to use a MOF file. As a standard, MOF defines exactly *what* end result should be enacted by DSC, but not *how* the changes should be made. The thousands of lines of code necessary to make the changes are inside the DSC resource modules, not inside the MOF files. A MOF file is like a project management document: it just defines what end result should be achieved, but not how it should be done, when, or by whom.

Again, you do not need to understand the syntax of MOF files to use DSC!

"Compiling" MOF Files

Running a configuration function creates one or more MOF files. In PowerShell-speak, we say that the configuration was "compiled" to one or more MOF files, even though MOF files are not executable binaries. Why do we say "compile" then?

Strictly speaking, DSC on Windows does not require PowerShell. Any properly-formatted MOF file can be used with DSC. A PowerShell script can produce a MOF file, but so can other tools and other programming languages. You could create a MOF file using a text editor or script on Linux, copy the MOF to a Windows box, and it would work. Someday, another vendor might release a better graphical tool or a better scripting language for creating MOF files than anything Microsoft has to offer, and DSC would still happily consume these MOF files to manage Windows.

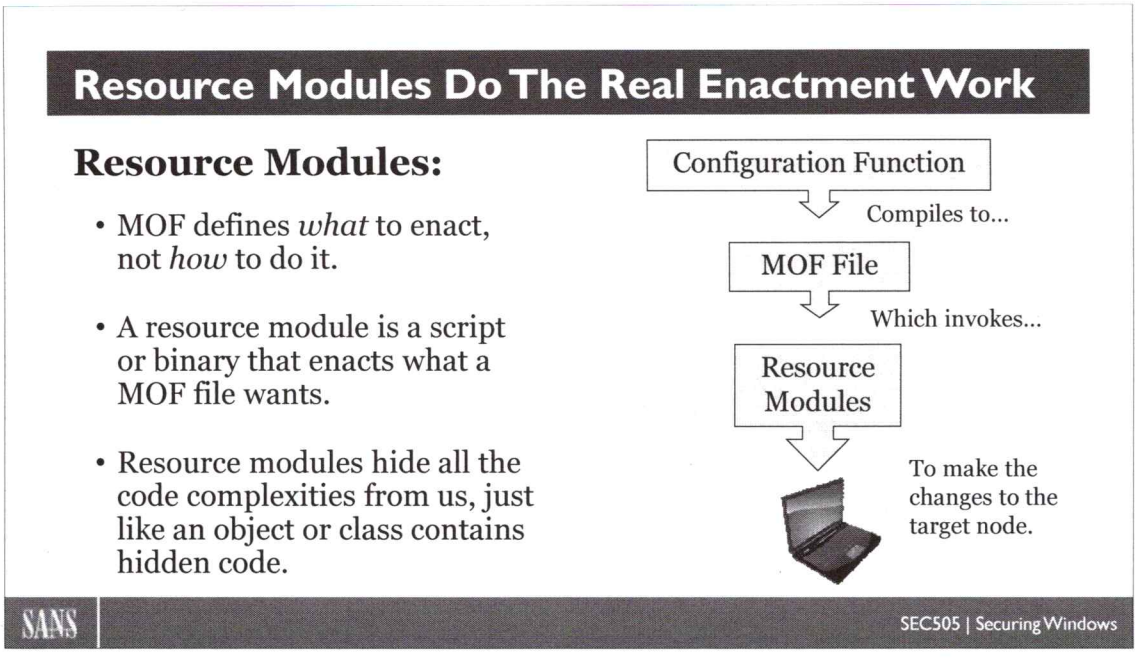
Hence, DSC is not just a PowerShell feature. DSC is bigger than and independent of PowerShell. DSC would still be a central part of configuration management in Windows even if PowerShell were to somehow magically disappear tomorrow.

PowerShell DSC To Manage Non-Windows Devices

Conversely, a PowerShell script can produce MOF files to manage non-Windows machines too. PowerShell configurations could be created on Windows using PowerShell for the sake of creating MOF files to reconfigure some Linux web servers or Cisco routers. It is up to the target vendor to implement support for processing MOF files, but the MOF file format itself is standardized by the DMTF to be cross-platform. That's the whole intention of MOF.

Hopefully, all the other vendors will implement these DMTF standards too so that we can get away from the proprietary and incompatible "Management Hell" we live in today. Today, you are likely to have a different configuration management system for each type of device or application you own, especially when these devices and applications come from different vendors.

And what about products like Puppet (puppetlabs.com) and Chef (www.chef.io), can they also use MOF files compiled from PowerShell scripts? Yes. The industry trend towards DevOps automation is pushing all these changes. From a IT management perspective, it's a whole new world.



Resource Modules Do The Real Enactment Work

MOF files define *what* to achieve, but not *how* to achieve it. But there is no magic, there must be something which does the real work behind the scenes. In order to make a device match what a MOF defines as the desired end state, many changes will have to be made, e.g., server roles must be (un)installed, registry values set, firewall rules added, software applications updated, files copied over the network, and so on. What is doing this work behind the scenes then?

Resource modules do the work of enacting the desired configurations expressed in MOF files. A "resource module" is a folder with one or more scripts and/or binary executables. These scripts or binaries contain the code which gets executed to make configuration changes to the machine. A DSC resource module is just a special type of PowerShell module, a module whose code is invoked through a MOF file instead of a regular PowerShell script.

By analogy, an executive at a company might write a policy document (the DSC configuration) stating what should be done at a high level of abstraction, i.e., the document is mostly a bunch of bullet points. The executive gives this document to the Chief Information Security Officer (CISO) and says, "Do whatever it takes to make it happen!" The executive doesn't really know or care how the CISO gets the work done, or even whether the work has already been done; after all, the change being ordered may have been completed last week, so it doesn't have to be done again, just confirmed. Next, the CISO creates a detailed project management plan (the compiled MOF file) and gives that detailed plan to the SecOps team (the resource modules). It's the SecOps team members who do the real, hands-on work of confirming compliance and making any

necessary changes. The SecOps team is where the rubber meets the road, the CISO drives the car, and the executive points where to go on the road map.

The overall DSC enactment process goes like this:

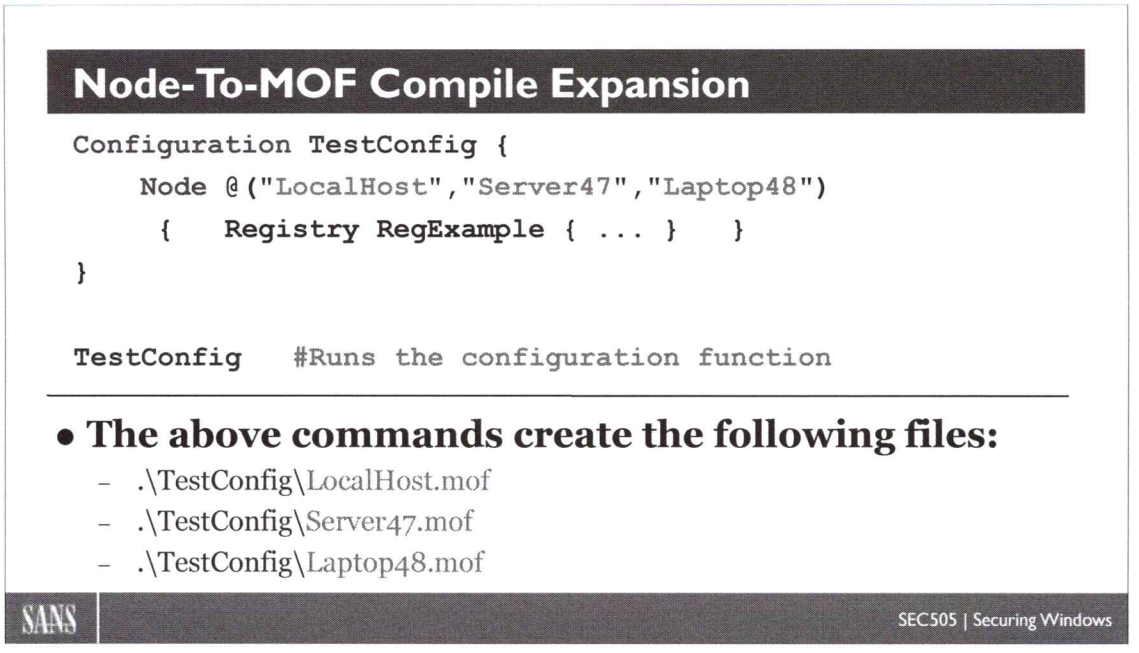
1) We download a DSC configuration script from a trusted source or write a new configuration script ourselves. The script contains one or more DSC configuration functions inside it.

2) The DSC configuration function in the script is "compiled" to one or more MOF files when that DSC function is executed. We now have one or more new MOF files on the hard drive ready to be applied or copied somewhere.

3) Using a special command we have not talked about yet, the MOF file is applied to a target device node to "enact" the configuration defined in the MOF. This is done by giving the MOF to the built-in Local Configuration Manager (LCM) service, which invokes the necessary DSC resource modules. You don't have to install the LCM as a server role, it's already part of PowerShell (strictly speaking, it's part of the Windows Management Framework that includes PowerShell).

4) With the MOF file in hand to guide it, the LCM calls the functions in whatever DSC resource modules are necessary to reconfigure the machine. The LCM does not make configuration changes itself, these changes are made by the DSC resource modules, but the LCM is what calls or invokes these modules.

Why is all this complexity a good thing? Think of what you are responsible for and how you would automate it. Your DSC configuration script might be 200 lines long, which is a lot, but the resource modules you are controlling might have tens of thousands of lines of code in them, written not just in PowerShell, but perhaps also in C# or C++. You don't have to write any resource modules yourself, you can just use them as-is. To automate your security work without DSC, you might have to write the equivalent code yourself. Or, what happens in real life, these tasks just don't get done at all, or they are done only once and never confirmed again, or a not-entirely-stable "quickie script" is written (and then rewritten, and rewritten, and rewritten as bugs are found and the environment changes over time). The DevOps approach to security requires continuous configuration enforcement and monitoring, which requires automation. In the long run, DSC makes our automation work easier, even if it's complex to learn right now.



Node-To-MOF Compile Expansion

A DSC configuration usually includes one or more node elements. Each node defined in a configuration will compile to a separate MOF file.

For example, when the following configuration is "compiled", three MOF files will be created for each of the three nodes:

```
#.\DSC\NodeToMofCompileExpansion.ps1

Configuration TestConfig
{
    Import-DscResource -ModuleName PSDesiredStateConfiguration

    Node @("LocalHost","Server47","Laptop48")
    {
        Registry RegExample
        {
            Ensure = "Present"
            Key = "HKEY_LOCAL_MACHINE\SOFTWARE\AAANewKey"
            ValueName = "EnableGoodness"
            ValueData = "0x2"
            Hex = $True
            ValueType = "Dword"
            Force = $True
        }
    }
}
```

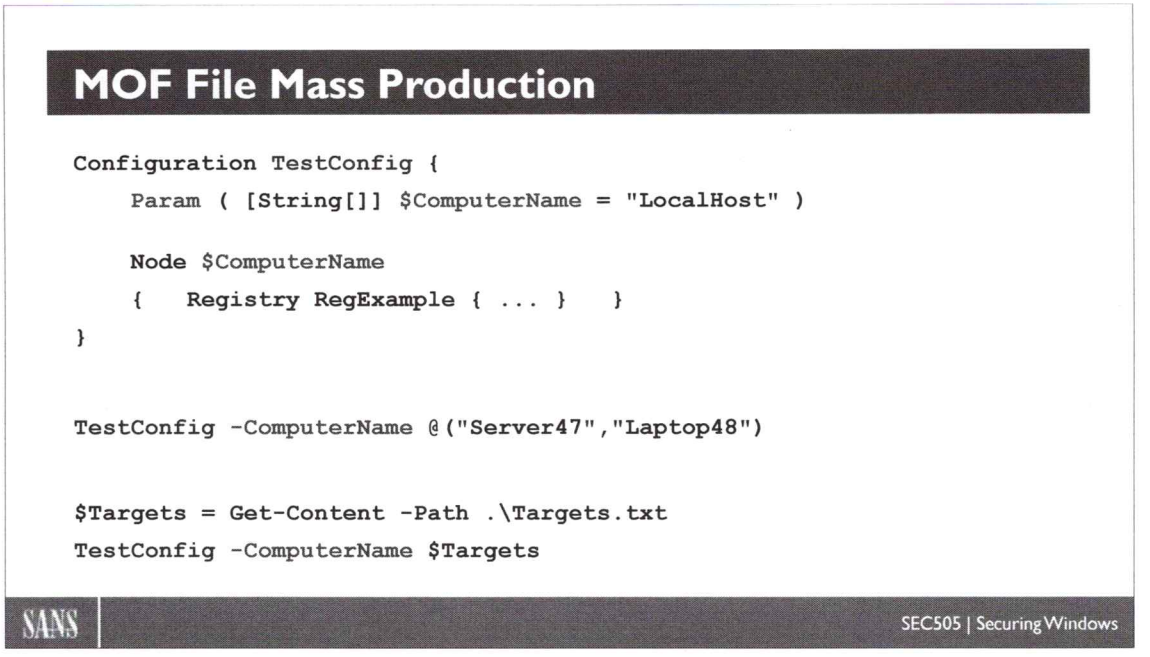
```
# Run the configuration function:
TestConfig
```

The code above will create the following files:

.\TestConfig\LocalHost.mof
.\TestConfig\Server47.mof
.\TestConfig\Laptop48.mof

If the TestConfig folder already exists, it will be used without an error. If any MOF files with identical names already exist in the folder, they will be overwritten without warning and without any error messages. Indeed, MOF files are supposed to be frequently overwritten as configuration requirements change, so this is normal and expected.

But are we going to hard-code the names of our machines into all our DSC scripts? Thankfully not!



MOF File Mass Production

Hard-coding computer names into DSC configuration scripts does not scale very well. We need a way to automate the creation of MOF files for thousands of machines.

Fortunately, a DSC configuration is like a function, hence, it supports the Param(…) keyword to pass in an array of node names as an argument to the DSC script. Once we have an array of computer names, perhaps after querying Active Directory, we can feed these names as nodes into a configuration function to mass-produce MOF files.

Here is the same sample code as before, but with the Param keyword added:

```
#.\DSC\MofFileMassProduction.ps1

Configuration TestConfig
{
    Param ( [String[]] $ComputerName = "LocalHost" )

    Import-DscResource -ModuleName PSDesiredStateConfiguration

    Node $ComputerName
    {
        Registry RegExample
        {
            Ensure = "Present"
            Key = "HKEY_LOCAL_MACHINE\SOFTWARE\AAANewKey"
            ValueName = "EnableGoodness"
            ValueData = "0x3"
```

```
            Hex = $True
            ValueType = "Dword"
            Force = $True
        }
    }
}
```

Notice that "Node" is now followed by a variable, $ComputerName, and not a hard-coded string or list. $ComputerName is an array of strings ([String[]]) declared after the Param keyword at the top and given a default value of "LocalHost." The Param keyword in a configuration is used just like in a script or function, i.e., you can define multiple input parameters to the configuration function and assign default arguments to these parameters too.

To call the above configuration function and pass in an array of node names:

```
TestConfig -ComputerName @("LocalHost","Server47","Server48")
```

The output of this call will be 1) a new folder named "\TestConfig" and 2) multiple MOF files created in that folder, one for each of the computer names passed into the function as an argument.

Imagine you have a list of computer names in a text file (Targets.txt) and you want to generate a MOF for each one of them. The list might have been typed up by hand, exported from a spreadsheet, saved after querying Active Directory with PowerShell or a third-party GUI tool, etc. Here is how easy it is to generate MOFs for all of them:

```
$Targets = Get-Content -Path .\Targets.txt

TestConfig -ComputerName $Targets
```

Imagine you wish to generate MOFs for every computer in an Active Directory organizational unit (OU) in your domain. Here is an example:

```
$Targets = Get-ADComputer -Filter * -SearchBase
   "OU=HVT,DC=testing,DC=local" | Select -ExpandProperty Name

TestConfig -ComputerName $Targets
```

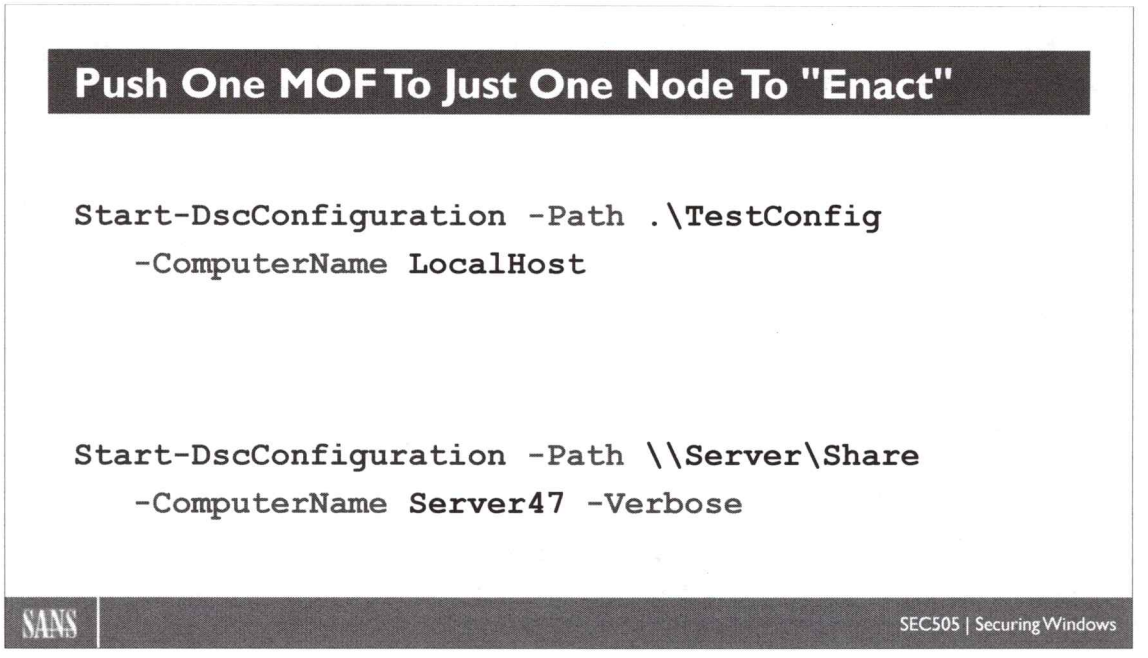
When passing in arguments to the parameters of a script, that PowerShell script must have the Param keyword on its first executable line (after any comments or blank lines). Hence, when executing a script, arguments can be passed into the script, and these arguments can be given to a DSC configuration function inside that script as arguments too.

Here is the same code as above, but placed inside a PS1 script:

```
#.\DSC\MofFileMassProduction.ps1

Param ( [String[]] $ArgsToScript = "LocalHost" )

Configuration TestConfig
{
    Param ( [String[]] $ComputerName = "LocalHost" )

    Import-DscResource -ModuleName PSDesiredStateConfiguration

    Node $ComputerName
    {
        Registry RegExample
        {
            Ensure = "Present"
            Key = "HKEY_LOCAL_MACHINE\SOFTWARE\AAANewKey"
            ValueName = "EnableGoodness"
            ValueData = "0x3"
            Hex = $True
            ValueType = "Dword"
            Force = $True
        }
    }
}


# Create MOF files using computer names passed into the script:
TestConfig -ComputerName $ArgsToScript
```

Notice that the Param keyword is used twice when we're looking at the code of a script. The first time Param is used, it's the first executable line of the script as a whole, and so this Param is processing any arguments passed into the script. These to-the-script arguments go into the $ArgsToScript array. The second time Param is mentioned, it is inside the configuration function, and this second Param processes the arguments passed into the function.

(Incidentally, if you'd rather pipe your target computer names into the function instead of using arguments to a parameter, you could write your script using the Process(…) keyword instead of Param(...).)



Push One MOF To Just One Node To "Enact"

Once we have MOF files, they can be applied or "pushed" to local or remote machines using PowerShell remoting. This is when the Local Configuration Manager (LCM) uses the MOF file to invoke resource modules to reconfigure the computer. This is when changes are actually made to the target node(s).

Requirements

To push a MOF to a remote target node, the target computer must have PowerShell remoting enabled, be accessible over the network, have WMF 4.0 or later installed, and the user pushing the MOF file must be a member of the Administrators group at the target. To apply a MOF to the local computer only, PowerShell remoting does not need to be enabled.

To push MOF files to local or remote systems, you must know the path to the folder containing the MOF files. This can be a local file system path or a UNC network path to a shared folder. Only read permission is required to push from that folder, but write permission is required to create, overwrite or delete those MOF files. For fault tolerance and scalability, use Distributed File System (DFS) shared folders when using UNC paths.

Automatic MOF Selection

MOF files are named after their target nodes, e.g., LocalHost.mof, Server47.mof and Laptop48.mof. When given the path to a folder with multiple MOF files, a target node will automatically select and enact the MOF named after itself. This is why we need only give the path to the folder itself, not the full path to any particular MOF in that folder.

Note: When applying a MOF to the local computer, DSC will prefer a MOF named LocalHost.mof over any other MOF in the source folder, even if there is a MOF in that folder which matches the NetBIOS name of the local computer.

To apply or "push" a MOF in the .\TestConfig folder onto the local computer only:

```
Start-DscConfiguration -Path .\TestConfig -ComputerName localhost
```

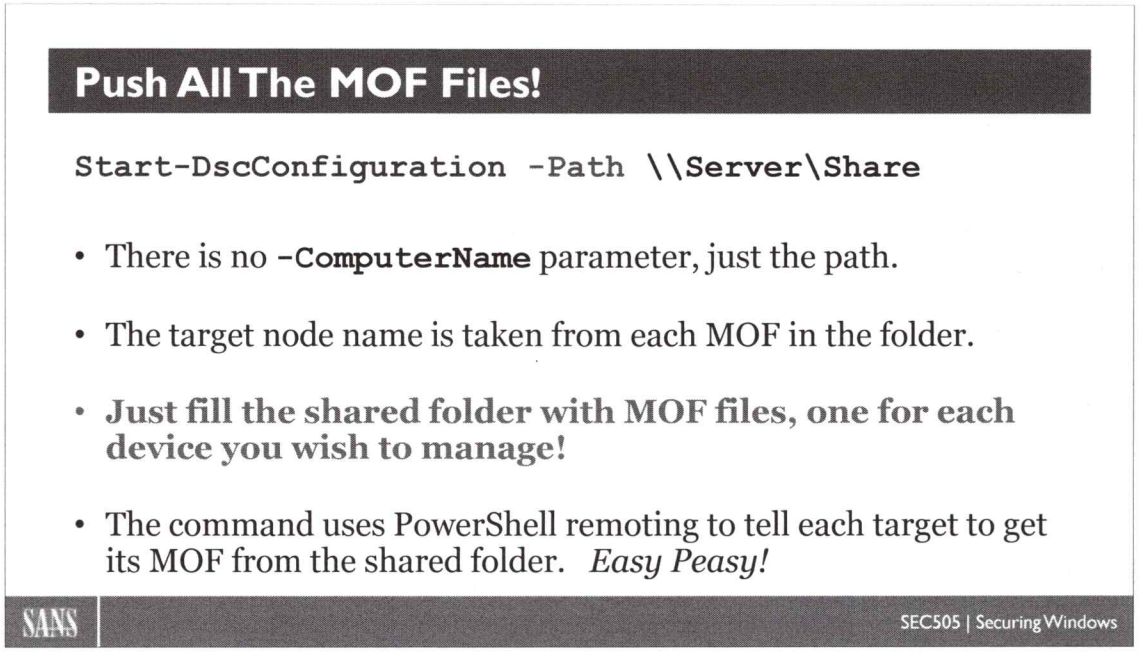
To apply/push a MOF from a shared folder onto the local host computer:

```
Start-DscConfiguration -Path \\Server\Share
   -ComputerName localhost
```

For the sake of troubleshooting, always use the -Verbose switch when pushing MOF files over the network or when dealing with remote machines. Status information will be displayed during the enactment process when you use the -Verbose switch.

To apply/push a MOF from a shared folder onto a remote computer named Server47:

```
Start-DscConfiguration -Path \\Server\Share
   -ComputerName Server47 -Verbose
```



Push All The MOF Files!

Remember, the target node will automatically look for a MOF in the shared folder named after the node itself, which, in this case, is the NetBIOS name or host name of the Windows box. We are using PowerShell remoting to connect to the target machine, telling it to grab its MOF from the shared folder and enact it.

Now here is something very important!

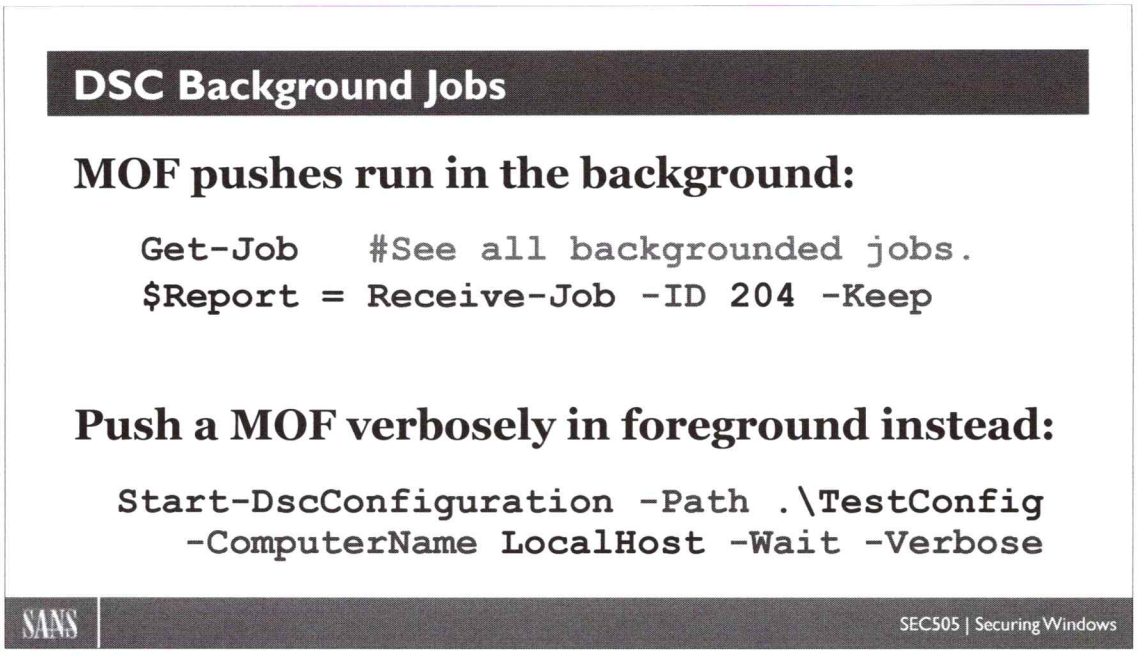
What if you want every MOF in the shared folder to be downloaded and enacted on every matching target node in your environment? If the shared folder contained ten thousand MOF files, with every MOF named after one of the computers in your environment, what is the command to tell those ten thousand machines to connect to the share, download its appropriate machine-named MOF, and then enact that MOF?

To push *every* MOF in a shared folder to every matching node on the network:

```
Start-DscConfiguration -Path \\Server\Share
```

Yow! Look how easy it is! The command reads the shared folder, get the target node names from the MOF files, then uses PowerShell remoting to instruct each target node to connect to that shared folder and enact its MOF. *Easy peasy!*

But how do we know if it actually worked? Can this really scale to a 1000 nodes?



DSC Background Jobs

By default, when a MOF is applied, DSC will do the work in the background as a hidden job. Each background job has a unique ID number. The job ID number is shown when Start-DscConfiguration is run to push MOFs, and this ID number can be captured to a variable like this:

```
$DscJob = Start-DscConfiguration -Path \\Server\Share
```

To see the status of a background job with ID number 204, for example:

```
Get-Job -ID 204 | Format-List *
```

To see a list of all backgrounded jobs, not just the DSC jobs:

```
Get-Job
```

To see the captured output of a background job with ID number 204:

```
Receive-Job -ID 204 -Keep
```

> **Tip:** Use the -Keep switch when viewing captured job output. Without -Keep, the captured output will be displayed and then deleted from the drive. If you forgot to put that data in an array when you received it, that data is now gone after it scrolls by in your command shell window.

By reviewing the DSC job, you can monitor the progress of a long-running MOF push. After all, pushing MOFs to ten thousand nodes may require several hours. By default, when there are multiple PowerShell remoting targets, PowerShell will only connect to 32 of them at a time. When a target node does not respond, PowerShell gives up on it and moves on to the next target in the list.

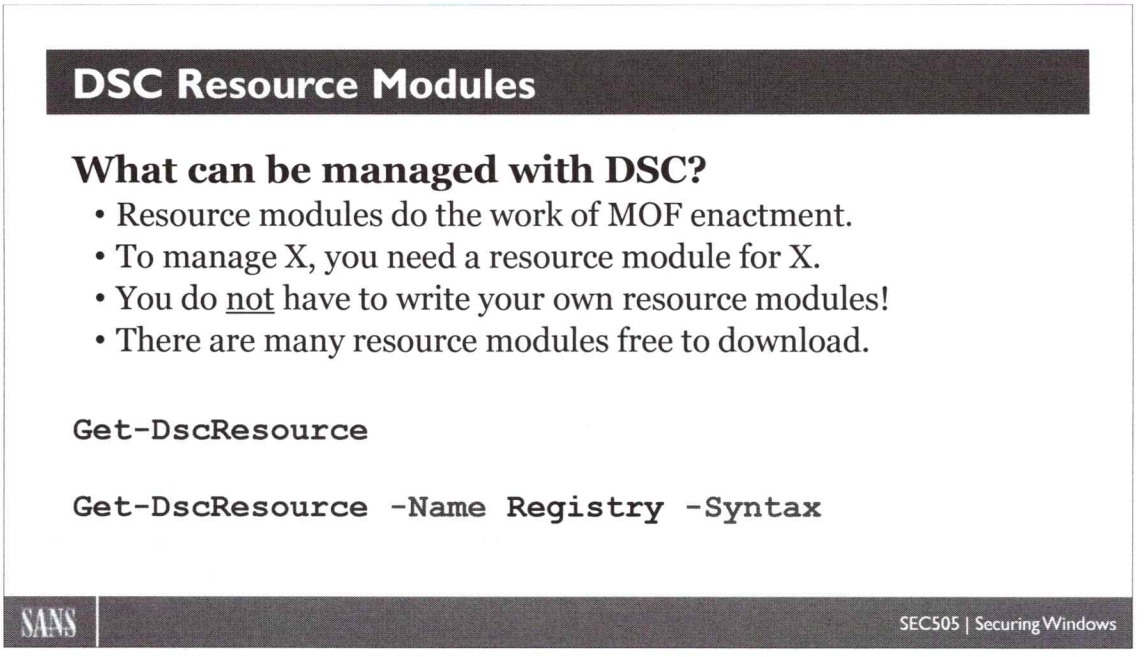
Pushing In The Foreground With Verbosity

When troubleshooting, it can be helpful to push MOFs in the foreground instead of as a backgrounded job. This is done with the -Wait switch. You won't be able to use that shell window for other commands while it's running, but at least you will see errors and other output in real time.

When pushing MOFs in the foreground, you may also want to display status information verbosely with the -Verbose switch. This can result in many pages of output. The -Wait and -Verbose switches can be used together or separately.

To push MOFs as a foreground command, not a background job, with the -Wait switch:

```
Start-DscConfiguration -Path \\Server\Share -Wait -Verbose
```



DSC Resource Modules

DSC can be used for more than just setting registry values. DSC can be used to manage user accounts, groups, files, networking settings, firewall rules, BitLocker, IIS web sites, Active Directory, SQL Server, Exchange, SharePoint, Hyper-V, Azure virtual machines, Windows Server roles and features, and more. Indeed, the intention is to eventually be able to manage *every* Microsoft product and setting through DSC. Someday, a "security template" will just be a DSC configuration that happens to only have security-related settings inside it. DSC configurations replace the old INF security template files.

What Can Be Managed Through DSC?

A DSC "resource" is a PowerShell module which contains the code necessary to enact a configuration. The Local Configuration Manager (LCM) uses resource modules to confirm and enforce the changes defined in a MOF file. A DSC configuration defines *what* end result is desired, but not *how* it should be accomplished. A resource module is the "*how*" part, it contains the code to actually get the job done.

Most importantly, you do not have to write your own DSC resource modules!

There are several built-in resource modules in the Windows Management Framework (WMF) you already have installed, and many more resource modules can be downloaded from Microsoft, from third-party vendors, or from open source project web sites.

To list the available DSC resource modules currently on your local computer:

```
Get-DscResource
```

Note that a third-party resource module might be on the hard drive, though, but not visible using the Get-DscResource cmdlet. If a DSC module is not installed in one of the normal PowerShell module folders, the Get-DscResource cmdlet can't find it (you'll have to remember where it was installed or ask the vendor).

DSC resource modules are (or should be) stored in one of the normal module locations:

```
$env:PSModulePath -split ';'
```

Custom and third-party DSC resource modules will normally be stored under here:

```
$env:ProgramFiles\WindowsPowerShell\Modules
```

A single module, by the way, may contain more than one DSC resource. If the module is stored in one of the normal $env:PsModulePath directories, the Get-DscResource cmdlet can discover the names of all the resources inside that module. (If the module folder is stored somewhere else outside of these normal locations, then Import-Module must be used to load the resources from the module before those resources can be used.)

Here are some of the built-in DSC resources you can see with Get-DscResource:

DSC Resource Name	Management Target or Uses
Registry	Windows registry keys, values and data.
File	Files and folders in a storage volume.
Archive	Extract files and folders from a compressed .zip archive file.
Service	Startup type, run-as identity, and running status of services.
User	Local user accounts, including password and enabled status.
Group	Local groups, including members to include or exclude.
Script	Your own custom PowerShell script code.

> **Tip:** In the ISE editor, press Ctrl-Space when inside of a Configuration or Node script block to pop up a list of DSC resources or other DSC keywords.

The Script resource can be used to assign your own custom block of PowerShell script code to run when there are no other resources available to manage what you want. This is what you might use to fill in the DSC management gaps left behind by the other resource modules available from Microsoft or third-parties.

Resource Properties

PowerShell resources are "declarative", hence, each resource will have a set of properties to define what final configuration state each management target should have after a configuration is applied; for example, the Registry resource includes properties to define the path to a registry value, its data, and value type. Some properties have a limited set of acceptable arguments; for example, with the Registry resource, the type of a registry value can only be a DWORD, MultiString, ExpandString, etc.

To see the properties of a DSC resource, such as the Registry resource:

```
Get-DscResource -Name Registry | Select -Expand Properties
```

To see the syntax of the code to use a particular resource type, such as for the Registry:

```
Get-DscResource -Name Registry -Syntax
```

For example, here is the (slightly edited) output of the prior command to get the syntax:

```
Registry [String] #ResourceName
{
    Key = [string]
    ValueName = [string]
    [DependsOn = [string[]]]
    [Ensure = [string]{ Absent | Present }]
    [Force = [bool]]
    [Hex = [bool]]
    [PsDscRunAsCredential = [PSCredential]]
    [ValueData = [string[]]]
    [ValueType = [string]{Binary | Dword | MultiString | String}]
}
```

Above, when the name of a resource property is not in square brackets (like Key or ValueName), it means that assigning a value to that property is mandatory. Property names which are in square brackets (like Force or Hex) are optional; you don't always have to assign values to them.

When there is a limited set of valid arguments to a property, the above syntax will show those arguments in curvy braces after the name of the property (like how ValueType can only be set to Binary, Dword, MultiString, or String).

Many resources have a property named "Ensure", and the Ensure property usually takes one of only two legal values: Present or Absent. Choosing "Present" will normally install, create or define something, if that something is not already installed, existing or defined. Choosing "Absent" will normally uninstall, remove or clear something, if that something currently exists or is installed.

Each resource type will have its own properties and syntax.

The PowerShell Gallery

The main public repository of PowerShell modules, scripts and DSC resources is the PowerShell Gallery (www.PowerShellGallery.com). The PowerShell Gallery includes code from authors at Microsoft, third-party companies, and individuals in the community who volunteer their time. This site is sponsored and funded by Microsoft.

Many items in the PowerShell Gallery are open source projects on GitHub, especially the larger projects (https://github.com/PowerShell). You can track these projects' version updates on GitHub and see discussions about known issues/bugs. If you want to live on the cutting edge or see what's coming over the horizon, then browse the GitHub PowerShell projects, but normally you'll just need to use the PowerShell Gallery.

Community and Experimental DSC Resources

Be aware that many of the DSC resources in the PSGallery are not officially blessed or supported by Microsoft. If the name of a resource begins with an "x", such as "xDisk", then Microsoft considers that resource to be **x**perimental, i.e., not supported by Microsoft. If the name of a resource begins with a "c", then that resource is **c**ommunity-written by volunteers, i.e., also not supported by Microsoft.

It is possible that a script, module or DSC resource on a public repository contains malware not yet discovered by the original authors or the repository maintainers. This includes the PSGallery too. As always, exercise caution when downloading and running code from the Internet, especially when not using the PSGallery or NuGet.org.

Finding DSC Resources

The easiest way to find DSC resource modules is to browse the PowerShell Gallery web site (www.PowerShellGallery.com) and filter using the "DSC" search keyword.

You can also search the PowerShell Gallery from within PowerShell itself, and this might be faster anyway once you get the hang of it (especially if you are already familiar with running commands like "apt-cache search *keyword*" on Linux).

To list all DSC resource packages available from the PowerShell Gallery (PSGallery):

```
Find-DscResource
```

To list the DSC resources whose names include a keyword from the PSGallery:

```
Find-DscResource | Where { $_.Name -Like "*disk*" }
```

To list the DSC resources in a particular module from the PowerShell Gallery:

```
Find-DscResource -ModuleName xSharePoint
```

Installing, Updating and Uninstalling DSC Resources

Once a DSC resource module is discovered, it can be installed on the local computer. When any PowerShell module is installed, it can be installed just for the current user or installed for all users who log onto the computer.

Installing a module only for oneself does not require membership in the Administrators local group because the module folder is installed underneath here:

$env:UserProfile\Documents\WindowsPowerShell\Modules\

Installing a module for entire computer and all users does require membership in the Administrators local group because the module folder is installed under:

$env:ProgramFiles\WindowsPowerShell\Modules\

Remember, when installing a DSC resource module, you install by using the name of the module, not the name of the resource. Also, always use the -Verbose switch when installing, updating or uninstalling modules to assist with troubleshooting. When installing a module from an untrusted repository, there will be a pop-up message asking you to confirm the installation. When installing a package, if that package has any dependencies, then those dependency packages will be installed by default too.

To install a DSC resource module, such as xNetworking, only for oneself:

```
Install-Module -Scope CurrentUser -Name xNetworking -Verbose
```

To install a DSC resource module, such as xWindowsUpdate, for all users on the computer, and to suppress the pop-up confirmation dialog box (-Force switch):

```
Install-Module -Scope AllUsers -Name xWindowsUpdate -Force
```

When -Scope is not specified, the module is installed for all users.

To update a module to the latest version, just run the Install-Module cmdlet again like when the module was first installed. If the latest version is already installed, it will not be downloaded and installed again (unless this is what you want, in which case you can use the -Force switch).

To confirm that a DSC module has been installed and is visible to our configuration scripts, list all locally-installed DSC resources and look for the newly-installed ones:

```
Get-DscResource
```

To uninstall a module, use the Uninstall-Module cmdlet, optionally with -Verbose again:

```
Uninstall-Module -Name xWindowsUpdate -Verbose
```

Other PowerShell Gallery Modules and Scripts

By the way, the PowerShell Gallery contains more than just DSC modules and resources. The PSGallery includes many other modules and scripts too not related to DSC.

To list all the available modules and scripts from the PSGallery (requires Internet access):

```
Find-Module

Find-Script
```

To download and save a module or script (including a DSC module):

```
Save-Module -Name SomeModule -Path C:\SomeLocalFolder

Save-Script -Name SomeScript -Path C:\SomeLocalFolder
```

To download and save a **module** for your own **personal** use into $env:USERPROFILE\Documents\WindowsPowerShell\Modules, which does not require Administrators membership:

```
Install-Module -Scope CurrentUser -Name SomeModule
```

To download and save a **script** for your own **personal** use into $env:USERPROFILE\Documents\WindowsPowerShell\scripts, which does not require Administrators membership:

```
Install-Script -Scope CurrentUser -Name SomeScript
```

To download and save a **module** for **machine-wide** use into $env:ProgramFiles\WindowsPowerShell\Modules, which does require Administrators membership:

```
Install-Module -Scope AllUsers -Name SomeModule

Install-Module -Name SomeModule
```

To download and save a **script** for **machine-wide** use into $env:ProgramFiles\WindowsPowerShell\Scripts, which does require Administrators membership:

```
Install-Script -Scope AllUsers -Name SomeScript

Install-Script -Name SomeScript
```

To list all modules and scripts installed from the PowerShell Gallery using either the Install-Module or Install-Script command:

```
Get-InstalledModule

Get-InstalledScript
```

To list just DSC modules and their DSC resources (as opposed to all modules):

```
Get-DscResource
```

To update all local modules and scripts that were previously installed from the PowerShell Gallery by using Install-Module or Install-Script in the past:

```
Update-Module

Update-Script
```

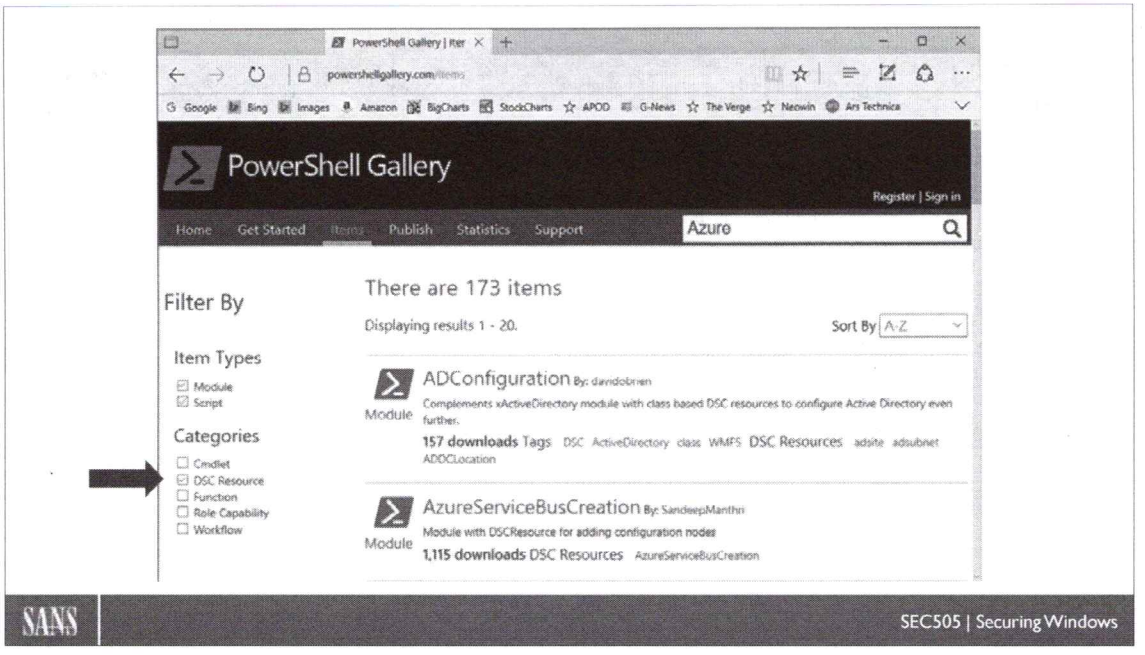
To update a specific module or script:

```
Update-Module -Name SomeModule

Update-Script -Name SomeScript
```

To uninstall a module or script installed from the PSGallery:

```
Uninstall-Module -Name SomeModule

Uninstall-Script -Name SomeScript
```

Over time, the PowerShell Gallery will become more and more important to Microsoft and to you. It will pay to spend some time playing with the code you find there. Besides, browsing the PSGallery and playing with the modules and scripts you find is fun.



PowerShell Gallery Web Site

The PowerShell Gallery (www.PowerShellGallery.com) can be searched using a browser or from within PowerShell itself. All the scripts, modules and DSC resources are free.

Notice the checkboxes on the left in the browser to limit the search.

Instead of using your browser, you can also search the PSGallery within PowerShell:

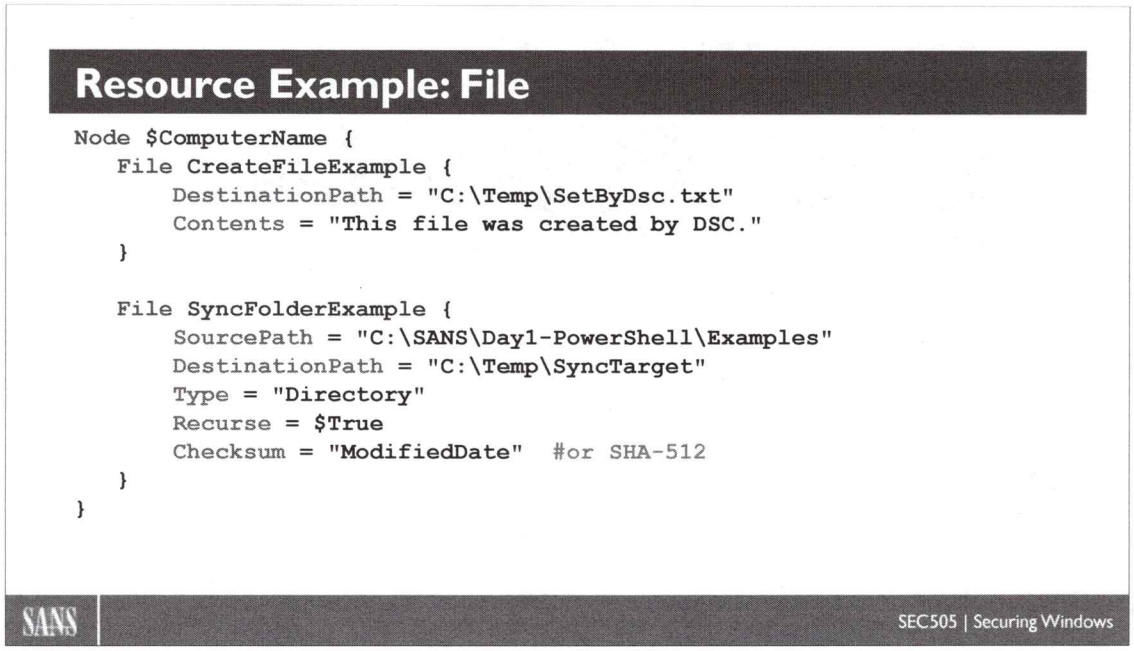
```
Find-DscResource
```

To list the DSC resources whose names include a keyword, such as "disk":

```
Find-DscResource | Where { $_.Name -Like "*disk*" }
```

To list the DSC resources in a particular module from the PowerShell Gallery:

```
Find-DscResource -ModuleName xSharePoint
```



Resource Example: File

The File resource module is used to create, copy and/or delete files and folders. It can also be used to sync two folders, either local or over the network via SMB. It is built in by default, but there might be better modules for file management on the PSGallery.

In the following code, the File resource module is invoked twice:

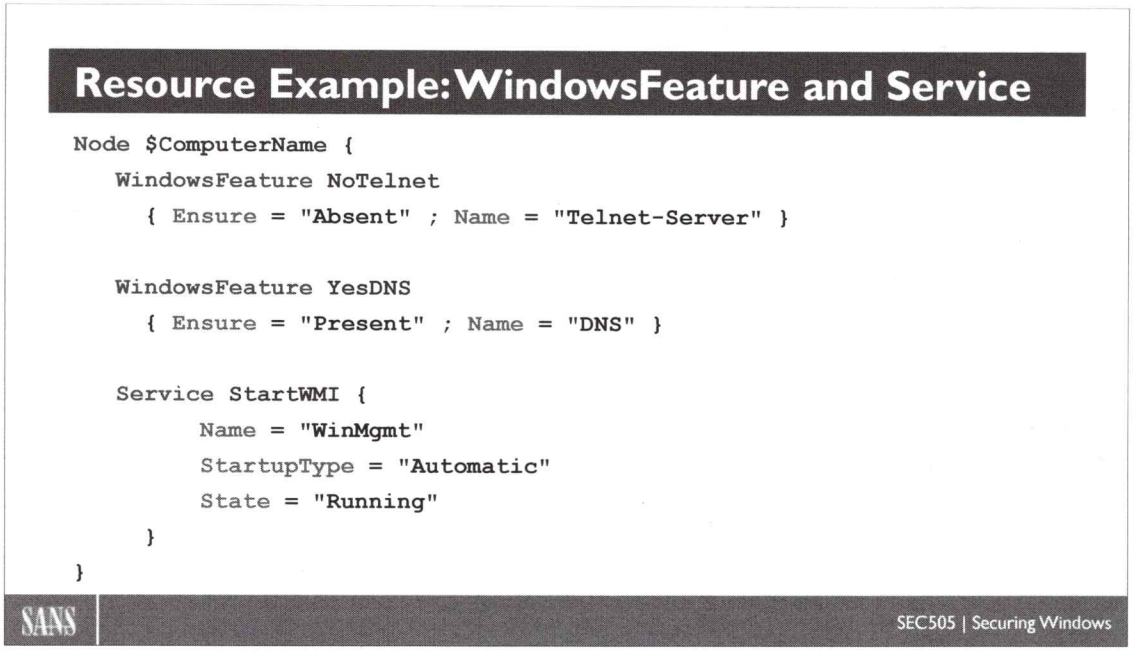
```
#.\DSC\ResourceExample-File.ps1

Configuration TestConfig
{
    Param ([String[]] $ComputerName = "LocalHost")

    Import-DscResource -ModuleName PSDesiredStateConfiguration

    Node $ComputerName
    {
        File CreateFileExample
        {
            DestinationPath = "C:\Temp\SetByDsc.txt"
            Contents = "This file was created by DSC."
        }

        File SyncFolderExample
        {
            SourcePath = "C:\SANS\Day1-PowerShell\Examples"
            DestinationPath = "C:\Temp\SyncTarget"
            Type = "Directory"
```

```
            Recurse = $True
            Checksum = "ModifiedDate"
        }
    }
}


# Create MOF file(s):
TestConfig -ComputerName "LocalHost"


# Enact MOF for localhost, run in foreground with verbose output:
Start-DscConfiguration -Path .\TestConfig -ComputerName
   "LocalHost" -Verbose -Wait
```

The first use (CreateFileExample) creates a file with some text content. If you delete the file, then the file is re-created when the configuration is enacted again. If the file is modified, its contents are overwritten the next time the configuration is enacted. If the file has not been modified, it is not overwritten again.

The second use (SyncFolderExample) recursively syncs a source folder to a destination folder using the last modified timestamp on each file and subfolder to determine whether that folder or file has changed since the last sync. Instead of using the last modified timestamp, it's also possible to use the creation timestamp or SHA-512 hash.

The -Verbose and -Wait switches are optionally used with Start-DscConfiguration in order to watch the progress of the sync operation in the command shell. Using these switches is not required.



Resource Example: WindowsFeature and Service

Server Manager can be used to install and uninstall roles and features on Windows Server boxes. Roles, features and services can be scripted with cmdlets like Install-WindowsFeature, Remove-WindowsFeature and Set-Service (or SC.EXE) to create one's own hardening script. These tasks are ideal for management through DSC instead.

There are multiple DSC resources related to roles, features and services:

WindowsFeature	(WMF 4.0 and later)
Service	(WMF 4.0 and later)
WindowsFeatureSet	(WMF 5.0 and later)
WindowsOptionalFeatureSet	(WMF 5.0 and later)
ServiceSet	(WMF 5.0 and later)

The resources above which require at least WMF 5.0 are better because their resource properties provide more control.

And, as always, there may be community or third-party resources on the PSGallery which might be even better for particular roles or for products from third-party companies that would have been roles had they been owned by Microsoft.

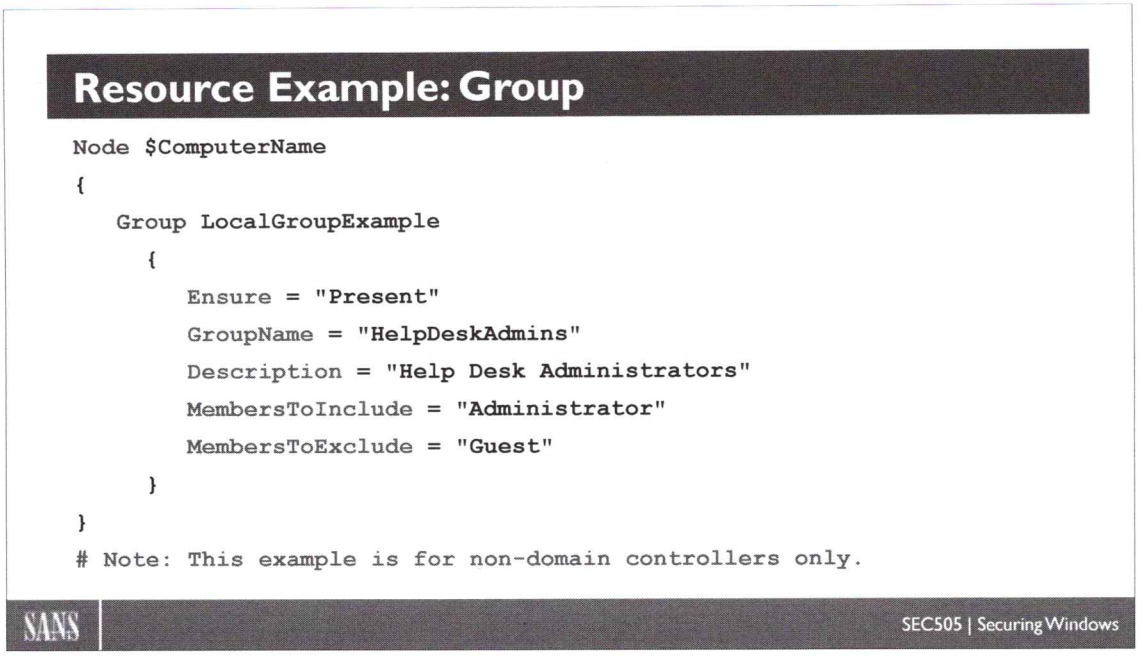
In the following configuration script, the telnet server role is uninstalled, the DNS server is installed, and the WMI service is set to start automatically:

```
#.\DSC\ResourceExample-WindowsFeature.ps1

Configuration TestConfig
{
    Param ([String[]] $ComputerName = "LocalHost")

    Import-DscResource -ModuleName PSDesiredStateConfiguration

    Node $ComputerName
    {
        WindowsFeature NoTelnet
        {
            Ensure = "Absent"
            Name = "Telnet-Server"
        }

        WindowsFeature YesDNS
        {
            Ensure = "Present"
            Name = "DNS"
        }

        Service StartWMI
        {
            Name = "WinMgmt"
            StartupType = "Automatic"
            State = "Running"
        }
    }
}


# Create MOF file(s):
TestConfig -ComputerName "LocalHost"


# Enact MOF for localhost only:
Start-DscConfiguration -Path .\TestConfig
   -ComputerName "LocalHost"
```

The names of roles and features can be seen in the output of Get-WindowsFeature. The internal names of services can be seen with Get-Service.

```
Get-WindowsFeature | Format-Table Name,DisplayName -AutoSize

Get-Service | Format-Table Name,DisplayName -AutoSize
```

Keep in mind that a role often needs to be configured after it is installed. In the configuration above, we are simply installing a role, but more lines in the DSC configuration might be necessary to configure additional settings for that role.



Resource Example: Group

The Group resource is for managing local groups only. It is not for global, universal, or domain local groups in Active Directory. Active Directory groups would be managed with different DSC resources, but these are not built-in with WMF 4.0.

> **Note:** This means the following example is not to be run on your training domain controller. It is an example for workstations and member servers only.

The Group resource can be used to create or remove local groups, set the group description, add members, remove members, or empty a group of all members.

In the following configuration, a new local group named "HelpDeskAdmins" is created, if necessary, the Administrator account is added, and the Guest account is removed:

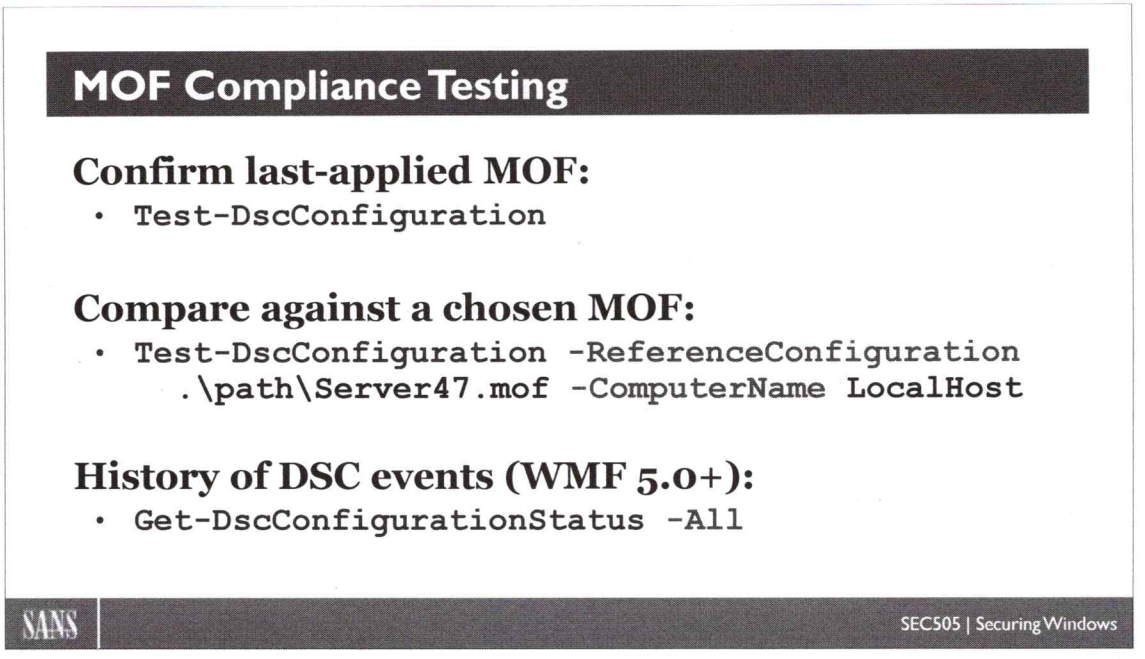
```
#.\DSC\ResourceExample-Group.ps1

Configuration TestConfig
{
    Param ([String[]] $ComputerName = "LocalHost")

    Import-DscResource -ModuleName PSDesiredStateConfiguration

    Node $ComputerName
    {
        Group LocalGroupExample
        {
            Ensure = "Present"
```

```
            GroupName = "HelpDeskAdmins"
            Description = "Help Desk Administrators"
            MembersToInclude = "Administrator"
            MembersToExclude = "Guest"
        }
    }
}


# Create MOF file(s):
TestConfig -ComputerName "LocalHost"


# Enact MOF for localhost only:
Start-DscConfiguration -Path .\TestConfig
   -ComputerName "LocalHost"
```

Even though the above resource is just for local groups, the syntax and resource properties for managing global groups in Active Directory would be nearly identical. After all, every group has a name, members to include, members to exclude, etc.

To find resources in the PowerShell Gallery related to group management:

```
Find-DscResource | Where { $_.Name -like "*group*" }
```



MOF Compliance Testing

How do we know that DSC is really working? How can we confirm that a MOF has actually been enacted on remote nodes?

The Test-DscConfiguration cmdlet confirms that the current state of a node matches the desired state for that node (WMF 4.0+). The cmdlet outputs $True if both the current and desired states match each other, $False otherwise.

```
Test-DscConfiguration
```

With WMF 5.0 and later, there is also a -Detailed switch to see more information:

```
Test-DscConfiguration -Detailed
```

By default, the Test-DscConfiguration cmdlet evaluates the local computer, but one or more remote computers can be evaluated as well with the -ComputerName parameter.

```
Test-DscConfiguration -Detailed -ComputerName server47

$List = @("server47","server48","server49")

Test-DscConfiguration -Detailed -ComputerName $List
```

But what is a node evaluated against? Which MOF? By default, a node is tested against its last-applied configuration MOF. The Local Configuration Manager (LCM) remembers and stores the MOF which was applied most recently.

However, when testing a node for compliance, the path to a particular MOF can supplied instead. This is very useful for debugging and judging the suitability of different configurations. In this case, the MOF can be thought of as an auditing template, perhaps even used as part of a vulnerability scanner.

When the path to a MOF file is given to the Test-DscConfiguration cmdlet, the node specified in the MOF is used to determine which node is the target of the test. It does not default to "LocalHost" when you do not specify a computer name, the target computer's name is extracted from the MOF. If you wish to compare a computer against a MOF when the node defined in the MOF is not the same as the name of the target computer, you must use the -ComputerName parameter. This is required even when the MOF is being compared against the local computer.

Normally, the name of the MOF file itself will match the target node named defined inside the MOF file; for example, the contents of the Server47.mof file will most likely include "Server47" as the name of the target node.

To evaluate a computer against a particular MOF file when that computer is the target node defined inside the MOF:

```
Test-DscConfiguration -ReferenceConfiguration .\path\Server47.mof
```

To evaluate a computer against a MOF when the name of the computer does not match the name of the node defined in the MOF, you must specify -ComputerName:

```
Test-DscConfiguration -ReferenceConfiguration .\path\Server47.mof
  -ComputerName LocalHost
```

History of DSC Events (WMF 5.0 and Later)

After a DSC configuration is initially applied, the Local Configuration Manager (LCM) will periodically monitor whether the computer is consistent with that configuration (by default, it is every 15 minutes). This requires WMF 5.0 or later.

To see a history of all DSC events on the local computer:

```
Get-DscConfigurationStatus -All
```

To see only the most recent DSC event and show all of its details:

```
Get-DscConfigurationStatus | Format-List *
```

Remove or Restore DSC Configurations

The Local Configuration Manager (LCM) stores various MOF files for DSC under the C:\Windows\System32\Configuration directory. The most important MOF files are:

- Current.mof -- The most-recently-applied configuration (current desired state).

- Previous.mof -- The previously-applied configuration (just prior to current).
- Pending.mof -- The about-to-applied configuration (a temporary file).
- MetaConfig.mof -- Any non-default settings for the LCM (might not exist).

When a new MOF is applied to a computer, that MOF is renamed to Pending.mof and enacted. Afterward enactment, the Current.mof file is renamed to Previous.mof, and the Pending.mof file is renamed to Current.mof.

The MetaConfig.mof file does not exist by default. When any LCM setting is changed from the factory default, this MOF file is created to store the LCM's settings. Deleting the MetaConfig.mof file will restore the LCM settings to their factory defaults again.

> **Note:** A MOF file is also sometimes called a "configuration document".

To restore the previous DSC configuration:

```
Restore-DscConfiguration
```

Restoring the previous DSC configuration will make a copy of Previous.mof, rename that copy to Current.mof, and overwrite any existing Current.mof file. If the Previous.mof file does not exist, an error will be produced and no other changes will occur.

To remove the current DSC configuration (use -Verbose for troubleshooting):

```
Remove-DscConfigurationDocument -Verbose -Stage Current
```

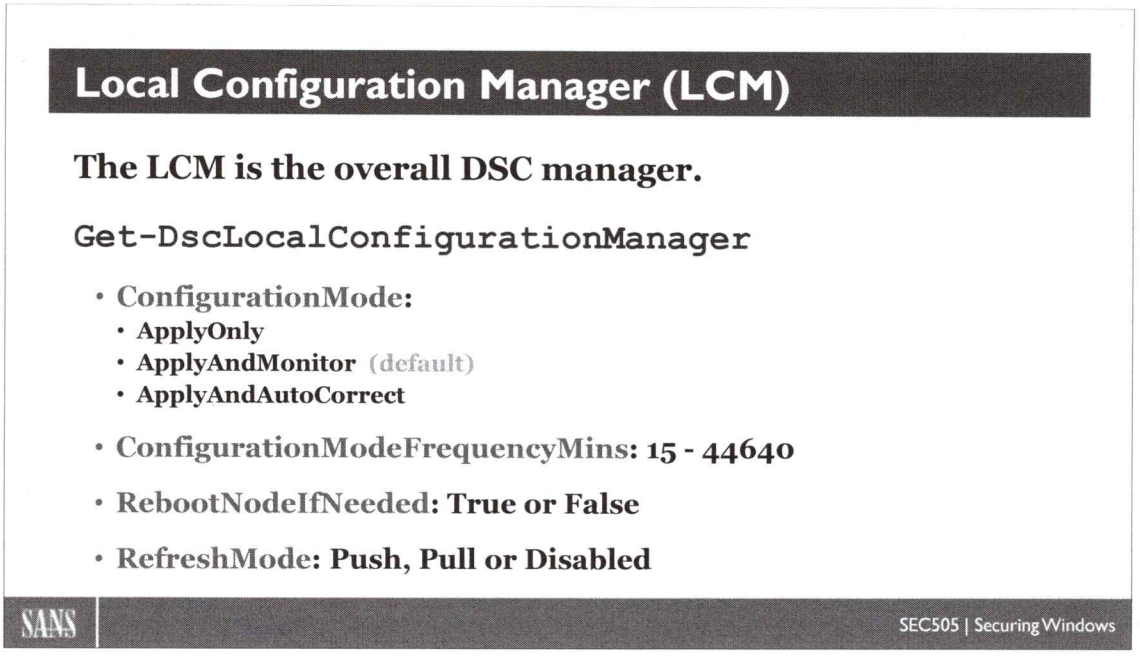
After removing the current DSC configuration, running the Get-DscConfiguration cmdlet will report that the current configuration does not exist, hence, it cannot be applied.

The -Stage parameter accepts one or more of three legal arguments: Current, Pending, and/or Previous. These values correspond to the Current.mof, Pending.mof and Previous.mof files. All three arguments can be passed in at once to delete all three files.

```
Remove-DscConfigurationDocument -Stage Current,Pending,Previous
```

> **Note:** The Current.mof, Pending.mof and Previous.mof files are encrypted with DPAPI, which only the local computer can decrypt, so they cannot be copied to another computer and used there. Nor can hackers read these files on disk.

Wouldn't it be better if it weren't just an audit check, but actually a reapplication of the MOF, if necessary, if the machine had drifted out of compliance? After all, continuous configuration enforcement requires *enforcement*, not just monitoring. Can we do it? Of course!



Local Configuration Manager (LCM)

The Local Configuration Manager (LCM) runs on each DSC-managed device. The LCM is the nexus for all DSC-related activities. The LCM is the enactment orchestrator or manager. It is what confirms and enforces the desired end state defined in a MOF by invoking the necessary resources. MOF files do not trigger themselves, this is the job of the LCM.

The LCM has its own configuration settings. By changing these LCM settings, you change how often the last-applied MOF is refreshed, where MOF files should be downloaded from, whether to run in audit-only mode, and so on. You do not have to modify the default LCM settings to use DSC, but let's see how to do it anyway.

To see the current configuration of the LCM on the local computer:

```
Get-DscLocalConfigurationManager
```

To see the LCM configuration on a remote computer, open a CIM session to it first:

```
$CimSess = New-CimSession -ComputerName <RemoteComputerName>

Get-DscConfiguration -CimSession $CimSess

Remove-CimSession -CimSession $CimSess
```

Unfortunately, the syntax used to manage the LCM is slightly different between WMF 4.0 and WMF 5.0 or later. To simplify things, and to improve DSC performance and

stability, it's best to upgrade to at least WMF 5.0 on all hosts. This manual assumes you have at least MWF 5.0, but many examples work on WMF 4.0 too (just not these).

The LCM has several settings. Here are the four most important ones:

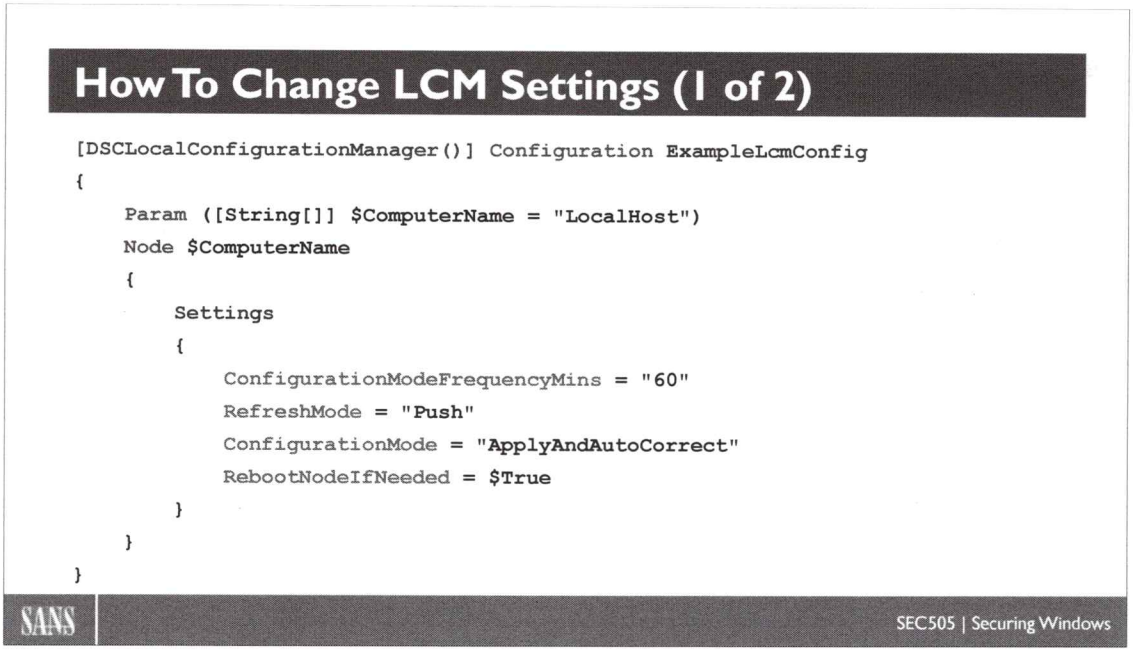
- **RefreshMode**: Push, Pull, or Disabled (defaults to Push). Determines how the LCM obtains its MOF files, or whether DSC is disabled entirely. In pull mode, the LCM downloads its MOFs from an HTTPS, HTTP or SMB server.

- **ConfigurationMode**: ApplyOnly, ApplyAndMonitor, or ApplyAndAutoCorrect (defaults to ApplyAndMonitor). Determines whether the LCM simply applies a new MOF only once and then sleeps (ApplyOnly), or applies a new MOF and periodically logs deviations from that configuration (ApplyAndMonitor), or applies a new MOF and then automatically reapplies the MOF when deviations are detected (ApplyAndAutoCorrect). The last one is the best for security.

- **ConfigurationModeFrequencyMins**: The number of minutes between each ConfigurationMode refresh (between 15 and 44640 minutes; defaults to 15).

- **RebootNodeIfNeeded**: $True or $False (defaults to $False). Determines whether unattended reboots occur when required by the application of a MOF.

How are these LCM settings changed?



How To Change LCM Settings (1 of 2)

The LCM has its own configuration settings, and its own special DSC configuration function format. Unfortunately, this format has changed from WMF 4.0 to version 5.0 and later. This manual only discusses the WMF 5.0 and later format because the assumption is that you will upgrade PowerShell to at least WMF 5.0.

The LCM itself is managed through a DSC configuration function. The syntax is a bit different than what we have seen so far, but the concepts are the same.

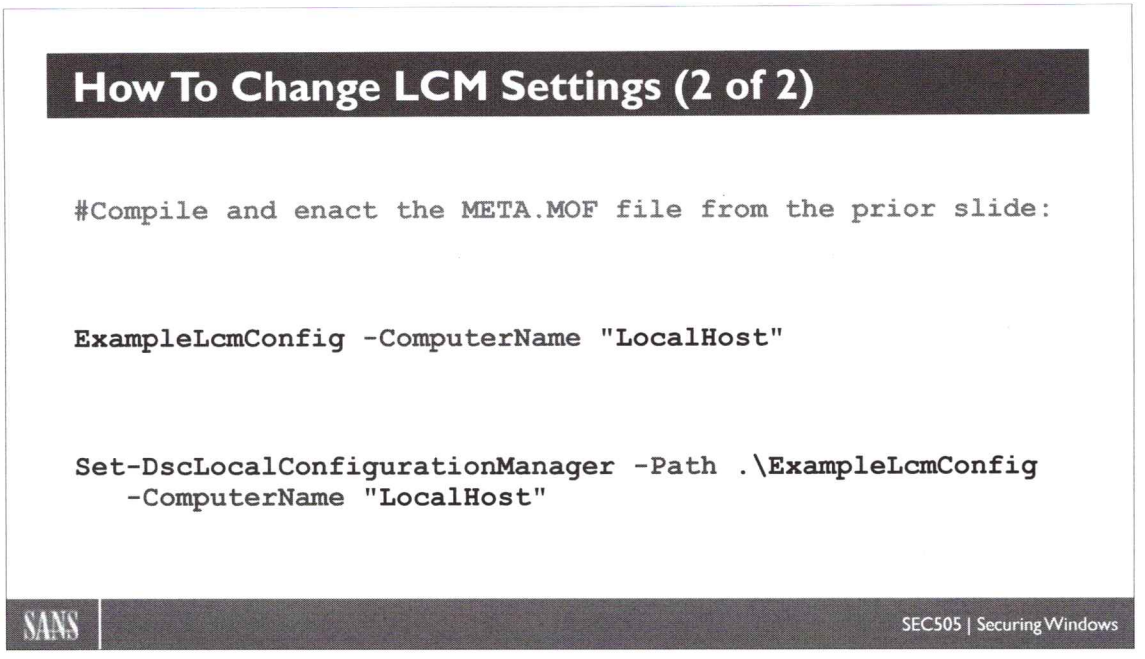
Here are the differences when managing LCM settings through a DSC configuration:

- The "Configuration" keyword is preceded by a string which informs PowerShell that this is an LCM configuration: "[DSCLocalConfigurationManager()]".

- There is a resource type named "Settings" which is just for the LCM, and, because you can only have one set of LCM settings, there is no resource identifier name after the "Settings" keyword in the configuration.

- When the LCM configuration function is run to compile a MOF, that file will end with the ".meta.mof" extension, not just ".mof". This is a META.MOF file.

- The META.MOF file is enacted with Set-DscLocalConfigurationManager, not the normal Start-DscConfiguration cmdlet (same idea, different cmdlet name).



How To Change LCM Settings (2 of 2)

Here is an example LCM configuration for WMF 5.0 or later:

```
#.\DSC\HowToChangeLcmSettings.ps1
# This is only for WMF 5.0 and later.

[DSCLocalConfigurationManager()]
Configuration ExampleLcmConfig
{
    Param ([String[]] $ComputerName = "LocalHost")

    Node $ComputerName
    {
        Settings
        {
            ConfigurationModeFrequencyMins = "15"
            RefreshMode = "Push"
            ConfigurationMode = "ApplyAndMonitor"
            RebootNodeIfNeeded = $False
        }
    }
}
```

The name of this configuration function, "ExampleLcmConfig", is not special, it can be anything you wish. Once the above configuration function is run (or dot-sourced from a script) to create the function object in the Function:\ drive, then that function can be run to compile and save a new META.MOF file for a target computer node:

```
# Create the META.MOF file:

ExampleLcmConfig -ComputerName "LocalHost"
```

This META.MOF file will go into a new subdirectory named after the configuration itself. To apply or enact that META.MOF file, we give the path to the folder:

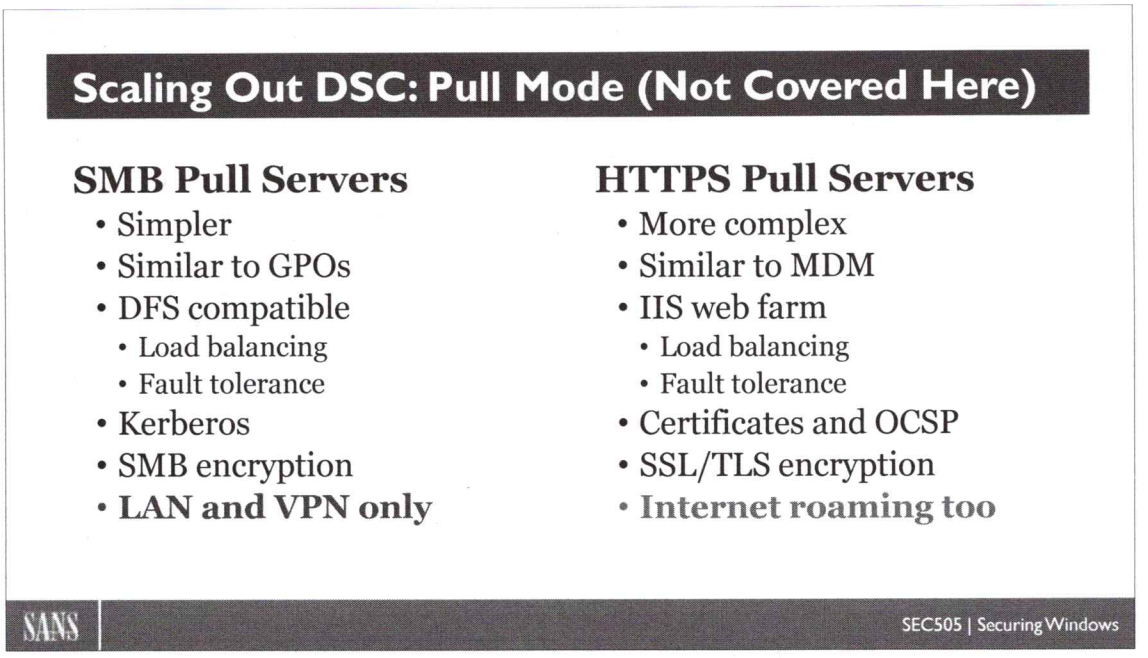
```
# Enact the META.MOF file on the localhost only:

Set-DscLocalConfigurationManager -Path .\ExampleLcmConfig
   -ComputerName "LocalHost"
```

Reset Back To Factory Default LCM Settings

Enacting a META.MOF file on a node changes the LCM settings on that node. The new LCM settings are stored in a new file named "MetaConfig.mof", which can be deleted to reset back to the original, factory default LCM settings.

To reset back to the factory default LCM settings:

```
del C:\Windows\System32\Configuration\MetaConfig.mof
```



Scaling Out DSC: Pull Mode

In the examples so far, we have been "pushing" DSC configurations to local and remote computers. This works fine in small- to medium-sized environments, perhaps up to a few thousand nodes, but push mode becomes difficult to scale as the number of nodes grows. Another problem is how to use DSC with mobile devices that are rarely, if ever, inside the LAN or accessible through a VPN.

In a large environment that includes roaming mobile devices, it would be better to have DSC clients download their MOF files on a regular basis from a load-balanced farm of HTTPS servers. The farm of servers would need to be accessible both from within the LAN and over the Internet. Can we do it? Yes!

Instead of pushing MOFs, the LCM on DSC clients can be configured to download their MOF files from a "pull server" using SMB, HTTPS or HTTP.

SMB Pull Servers

An SMB pull server is the easiest to configure, but it can only be used inside the LAN, not over the Internet, unless the DSC client maintains a VPN connection into the LAN. SMB supports native encryption, digital signatures, Kerberos authentication, and fault tolerance and load balancing when using Distributed File System (DFS) shared folders. Conceptually, DSC with a SMB pull server is the most similar to Group Policy.

Configuring the SMB shared folder is very simple: just grant Full Control access to Administrators and Read access to the Everyone group (just like the permissions for the SYSVOL shared folder on domain controllers for GPOs). To configure file replication,

fault tolerance and load balancing with SMB shares, please see the documentation for the Distributed File System (DFS) tools that come with Windows Server.

HTTPS Pull Servers

An HTTPS pull server is more difficult to configure, but a major advantage of HTTPS is that it can be used with DSC clients both inside the LAN and roaming out on the Internet. Setting up a fault-tolerant and load-balanced farm of HTTPS servers is a task most organizations are already familiar with; in fact, most likely you already have such a farm for the sake of other web applications, so this existing farm could be used for distributing DSC files too (hence, no new servers required to deploy DSC).

HTTPS requires an SSL/TLS digital certificate from a trusted Certification Authority (CA). DSC clients sometimes require their own certificates too, such as for the encryption of passwords. It's best to use your own Public Key Infrastructure (PKI) to obtain certificates. PKI is discussed elsewhere in this course, not in this manual.

Use DSC To Help Create An HTTPS Pull Server

The easiest way to set up an HTTPS pull server for DSC is to use DSC!

From the PowerShell Gallery site (www.PowerShellGallery.com) you can read about the xPSDesiredStateConfiguration module, which is a part of the DSC Resource Kit maintained by Microsoft. One of the DSC resources in this module is xDscWebService, which can be used to configure an HTTPS/HTTP pull server on top of IIS.

To install the xPSDesiredStateConfiguration module from the PowerShell Gallery:

```
Install-Module -Name xPSDesiredStateConfiguration -Verbose
```

Unfortunately, We Can't Talk About Everything...

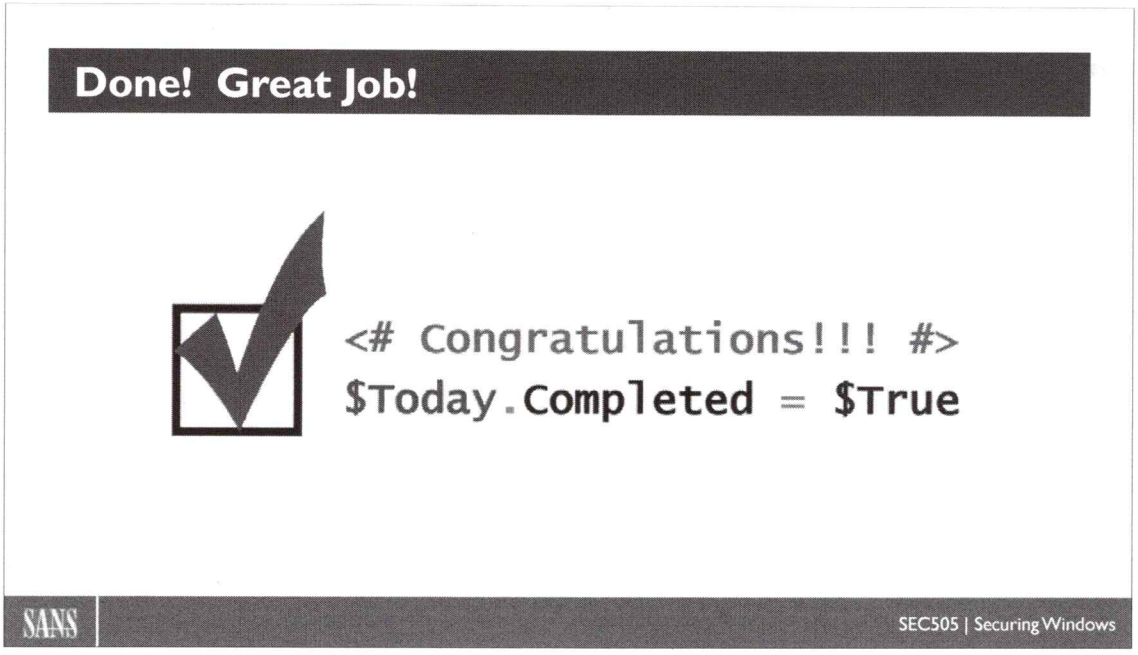
Unfortunately, due to time constraints, we can't talk about the remaining steps to configure pull servers and DSC clients. The xDscWebService resource helps, but we would still need to perform several other tasks to deploy a scalable, fault-tolerant, large-scale enterprise solution, including the following tasks:

- Install SSL/TLS certificates on IIS servers from our PKI.
- Generate GUID numbers or Agent IDs for clients.
- Configure client LCM settings, such as reporting vs. resource pull servers.
- Generate checksum hashes for the MOF files.
- Compress resources into zip archives for distribution.

You can also mix pull and push modes together with partial configurations, but there are some complexities with partial configurations that we cannot cover here.

There are several guides on the Internet and from Microsoft on how to configure DSC pull mode. The goal for this manual is to see how to do DSC using push mode to get a

taste of DSC in general, then you might choose to deploy pull mode later in your environment. Unless we spend an entire day on nothing but DSC, there just isn't time to also cover pull mode here today.



Congratulations!

You have finished the course!

Thank you for attending this seminar, and please complete the evaluation form. Your feedback plays an important role in the development of future courses and the editing of this current one.

Thank You!

Appendix A: Custom ADM/ADMX Templates

By editing ADM/ADMX templates yourself, you can create your own yellow folders and values underneath the Administrative Templates container in GPOs. Hence, you can set virtually any registry value you wish through Group Policy. This is immensely useful for security and network administration.

Fortunately, the syntax of ADM templates is only about as complex as simple HTML files. And there are shareware and commercial ADM file editors too, including:

- Policy Template Editor (http://www.tools4ever.com/software/additional-software/policytemplateeditor/)
- Reg2Adm (http://www.rct.net)

Many of the editor sites also have special-purpose ADM templates that can be downloaded for free. Make sure to examine them in Notepad first though.

The ADMX templates unique to Vista and later are more difficult to edit, but you can create/edit an ADM template and convert it into an ADMX template with a free tool from FullArmor named the "ADMX Migrator" tool (Google on "site:microsoft.com admx migrator" since it's actually downloaded from Microsoft's site).

Example ADM Template File

Let's look at a few simple templates. We can't discuss every possible ADM keyword, but the essentials can be covered quickly enough. (For a more complete discussion see the Recommended Reading list.)

Example 1: Simple ADM

The text below is the contents of a file which ends with the ".adm" extension.

```
CLASS USER

CATEGORY "SANS Windows Explorer Annoyances"
   POLICY "Show Filename Extensions"
      KEYNAME "Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced"
      EXPLAIN "If enabled, all filenames will show their last extension."
      VALUENAME "HideFileExt"
      VALUEON NUMERIC "0"
      VALUEOFF NUMERIC "1"
   END POLICY
END CATEGORY
```

CLASS USER
The first line, "CLASS USER", indicates that all of the following registry paths will be found under HKEY_CURRENT_USER only. This assumption continues until a line with "CLASS MACHINE" is encountered. Thereafter, all registry paths will be found under HKEY_LOCAL_MACHINE.

CATEGORY and END CATEGORY
Notice how the "CATEGORY" and "END CATEGORY" tags work together to demarcate a block of lines very similarly to HTML tags. The category is the yellow folder seen when you open the Administrative Templates container in a GPO.

POLICY and END POLICY
The POLICY tag is the name of the icon on the right-hand side. It's inside the yellow category folder. These tags demarcate information about a registry path under HKCU or HKLM, the name of a value, and the data to be assigned to that value.

KEYNAME
The KEYNAME is the path to the registry value under HKLM or HKCU, not including the name of the value itself.

EXPLAIN
When you open a policy icon there will be a Setting tab and an Explain tab. The EXPLAIN line gives the text on the Explain tab. To indicate a new line, put "\n\n" into the string (without the double quotes).

VALUENAME
This is the name of the registry value found under the KEYNAME path.

VALUEON and VALUEOFF
When the policy is set to "Enabled" on the Setting tab, the VALUENAME is added --if it doesn't already exist-- as a REG_DWORD value with the decimal equivalent of the number following the VALUEON tag. If the policy is set to "Disabled", it is the number after the VALUEOFF tag. The number does not have to be either 0 or 1. "NUMERIC" indicates that the data should be entered with a REG_DWORD value; if "NUMERIC" were omitted, the value would have been a REG_SZ.

Example 2: Using [Strings]

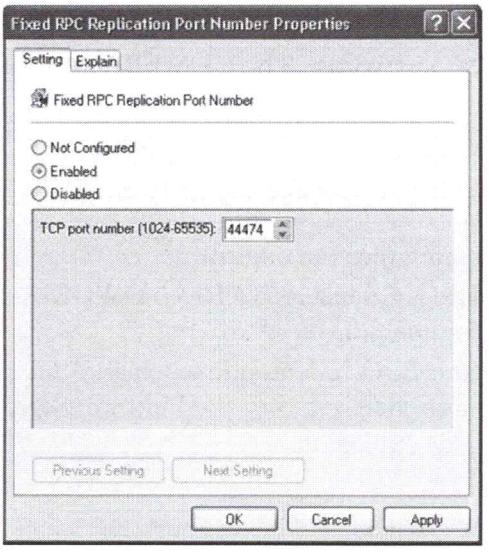
Here is the same template as above, but using string variables instead. String variables are extensively used in Microsoft's built-in templates. String variables always start with "!!" and correspond to the variable names under the "[Strings]" line. Except for long EXPLAIN strings, using variables can be annoying.

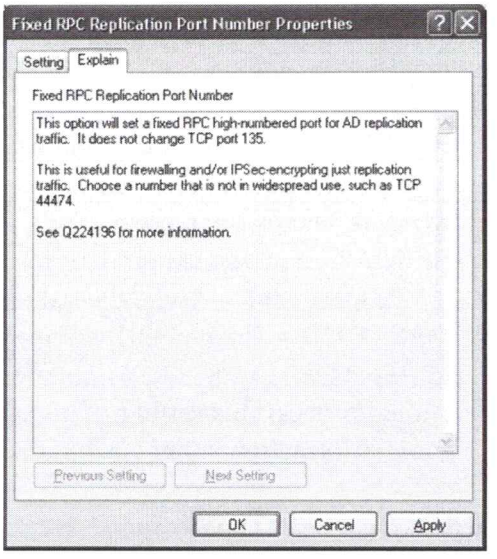
```
CLASS USER

CATEGORY !!strCategoryText
  POLICY !!strPolicyText
    KEYNAME !!strKeynameText
```

```
        EXPLAIN !!strExplainText
        VALUENAME !!strValueText
        VALUEON NUMERIC "0"
        VALUEOFF NUMERIC "1"
    END POLICY
END CATEGORY

[Strings]
strCategoryText  = "SANS Windows Explorer Annoyances"
strPolicyText    = "Show Filename Extensions"
strKeynameText = "Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced"
strExplainText   = "If enabled, all filenames will show their last extension."
strValueText     = "HideFileExt"
```

Example 3: PART and END PART

The PART tag is used when you want to put text, checkboxes, pull-down lists, spin controls, or anything else into the body of the Setting tab.






```
CLASS MACHINE

CATEGORY "SANS Active Directory"
    POLICY "Fixed RPC Replication Port Number"
        KEYNAME "SYSTEM\CurrentControlSet\Services\NTDS\Parameters"
        EXPLAIN !!strExplanationVariable1

        PART "TCP port number (1024-65535):"
            NUMERIC REQUIRED  MIN "1024"  MAX "65535"  DEFAULT "44474"
            VALUENAME "TCP/IP Port"
        END PART

    END POLICY
END CATEGORY

[Strings]
```

```
strExplanationVariable1="This option will set a fixed RPC high-numbered port for AD
replication traffic.  It does not change TCP port 135. \n\n This is useful for firewalling
and/or IPSec-encrypting just replication traffic.  Choose a number that is not in
widespread use, such as TCP 44474. \n\n See KB224196 for more information."
```

CLASS MACHINE
The first line, "CLASS MACHINE", indicates that all of the following registry paths will be found under HKEY_LOCAL_MACHINE only. This assumption continues until a line with "CLASS USER" is encountered. Thereafter, all registry paths will be found under HKEY_CURRENT_USER.

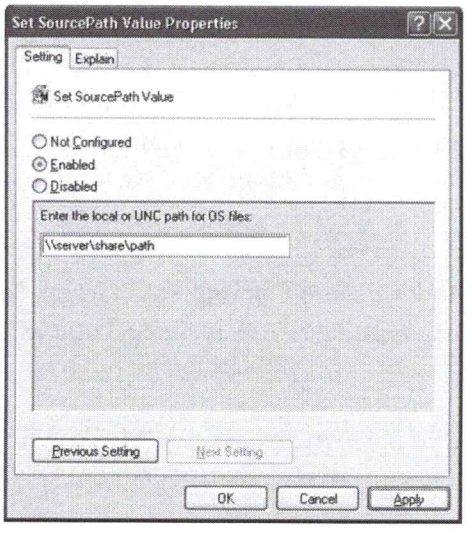
PART "TCP port number (1024-65535):"
The PART and END PART tags demarcate a block of options that affect just one value, the value in VALUENAME inside the PART block. The text which follows PART is displayed in the Setting tab.

NUMERIC ...
NUMERIC has another use. It is also a keyword which, when it occurs in a PART block, causes a spin control to be displayed for the selection of a number. NUMERIC can be further qualified by adding one or more "helper" keywords. The NUMERIC helper switches are:

- REQUIRED -- This value must be configured and cannot be blank.
- MIN -- The minimum value.
- MAX -- The maximum value.
- SPIN -- The increment amount for adjusting the spin control.
- TXTCONVERT -- Write the value as REG_SZ instead of REG_DWORD.
- DEFAULT -- The default setting in the spin control.
- CLIENTEXT -- The GUID number of the GPO "client-side extension" DLL that should process this setting. This will rarely be used. See the Platform SDK for more information about custom client-side extensions.

Example 4: EDITTEXT

The EDITTEXT keyword creates a box for entering text and creating a REG_SZ and REG_EXPAND_SZ values.




```
CLASS MACHINE

CATEGORY "SANS Active Directory"
   POLICY "Set SourcePath Value"
      KEYNAME "SOFTWARE\Microsoft\Windows\CurrentVersion\Setup"
      EXPLAIN !!strExplanationVariable2

      PART "Enter the local or UNC path for OS files:"
         EDITTEXT REQUIRED DEFAULT "" MAXLEN "254"
         VALUENAME "SourcePath"
      END PART

   END POLICY
END CATEGORY

[Strings]
strExplanationVariable2="The SourcePath registry value determines where the OS looks
first when it needs to copy fresh OS files.\n\nThe path entered can be a local drive path
or a network UNC pathname.\n\nThe advantage is that users are not prompted for the
CD-ROM and you can keep the files at the UNC path always updated."
```

EDITTEXT

EDITTEXT displays a simple text box for typing in data. EDITTEXT supports a few switches to customize it:

- EXPANDABLETEXT -- Indicates that a REG_EXPAND_SZ value should be created instead of a REG_SZ, which is the default.
- REQUIRED -- Something must be entered, it cannot be left blank.
- DEFAULT -- Data that initially appears in field. May be overwritten.
- MAXLEN -- The maximum number of characters that can be entered.
- OEMCONVERT -- Automatically converts data from Windows ASCII to OEM and back to ASCII again. This ensures proper conversion on clients.
- CLIENTEXT -- The GUID number of the GPO "client-side extension" DLL that should process this setting. This will rarely be used. See the Platform SDK for more information about custom client-side extensions.

Other ADM Keywords

There are about a dozen more ADM keywords and keyword-switches. For example, it is possible to have checkboxes, comboboxes, pull-down lists, #If…#EndIf statements, etc. After going through the examples above, the best way to learn is to examine the ADM templates with Microsoft and browse material from the Recommended Reading list.

Recommended Reading

In the Windows Help documentation, navigate to Users and Computers > Group Policy > Concepts > Using Group Policy > Administrative Templates > Advanced Topic: Creating Custom .ADM Files.

Group Policy: Management, Troubleshooting, and Security, by Jeremy Moskowitz (Sybex). Also, check out Jeremy's site, it's very useful: www.gpanswers.com.