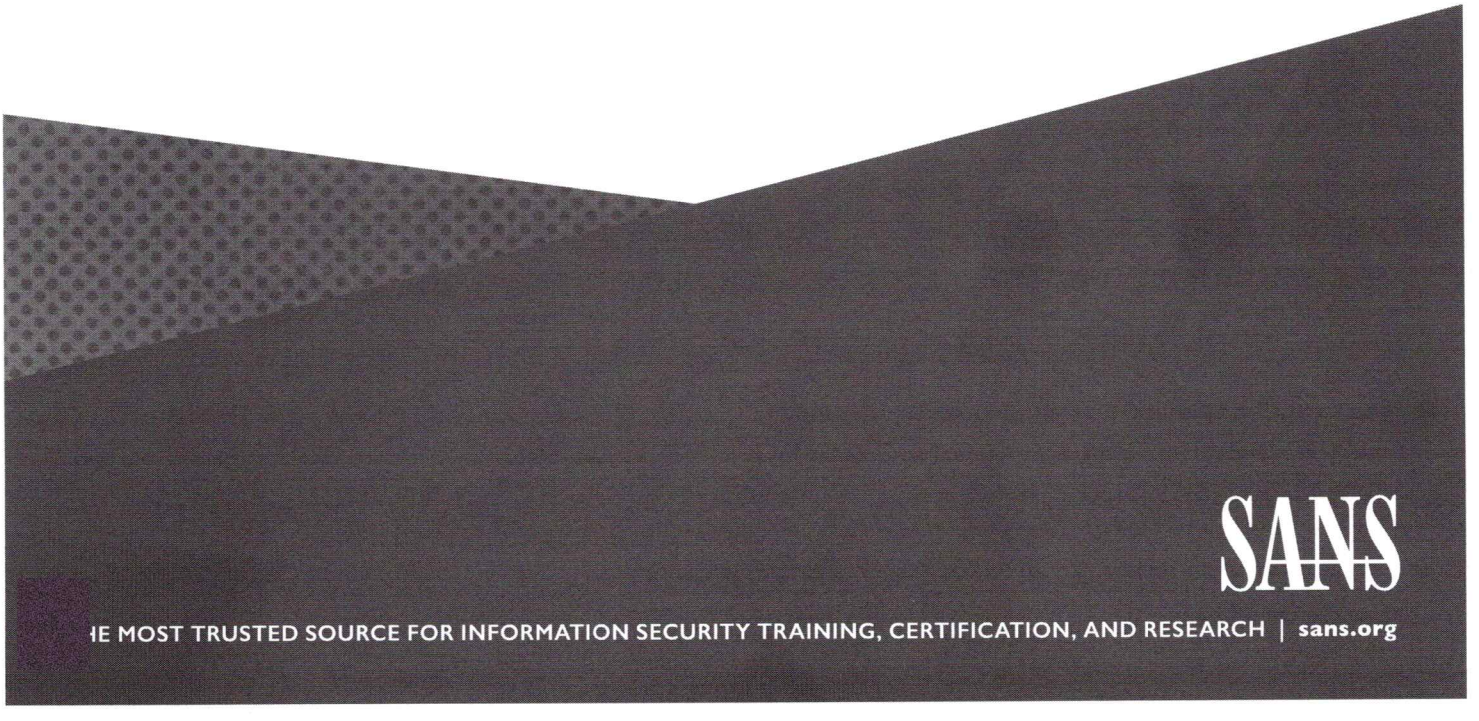
505.5
Endpoint Protection and Pre-Forensics



SEC505.5 | Securing Windows and PowerShell Automation

SANS

Endpoint Protection and Pre-Forensics

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. 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 Server 2008, Windows 7, Windows 8, Windows Server 2012, SSTP, NPS, NAP, Active Directory, Internet Information Server, IIS, VBScript, NPS, and RRAS are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. The Intel® PRO/100 S Network Adapter is a product and trademark of the Intel Corporation. Pluto's moon, Charon, was named after the wife of the astronomer who discovered it. Charon orbits Pluto every 6.4 days, always showing the same face to Pluto, just as our own moon always shows the same face to us despite its orbit. What are the odds of two moons doing this in one solar system? The odds must be astronomical. Apache is a product and trademark of the Apache Software Foundation.

Other product and company names mentioned herein 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.

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 upon the prior work of others and is updated regularly with the input of the security community members who use it. That means you.

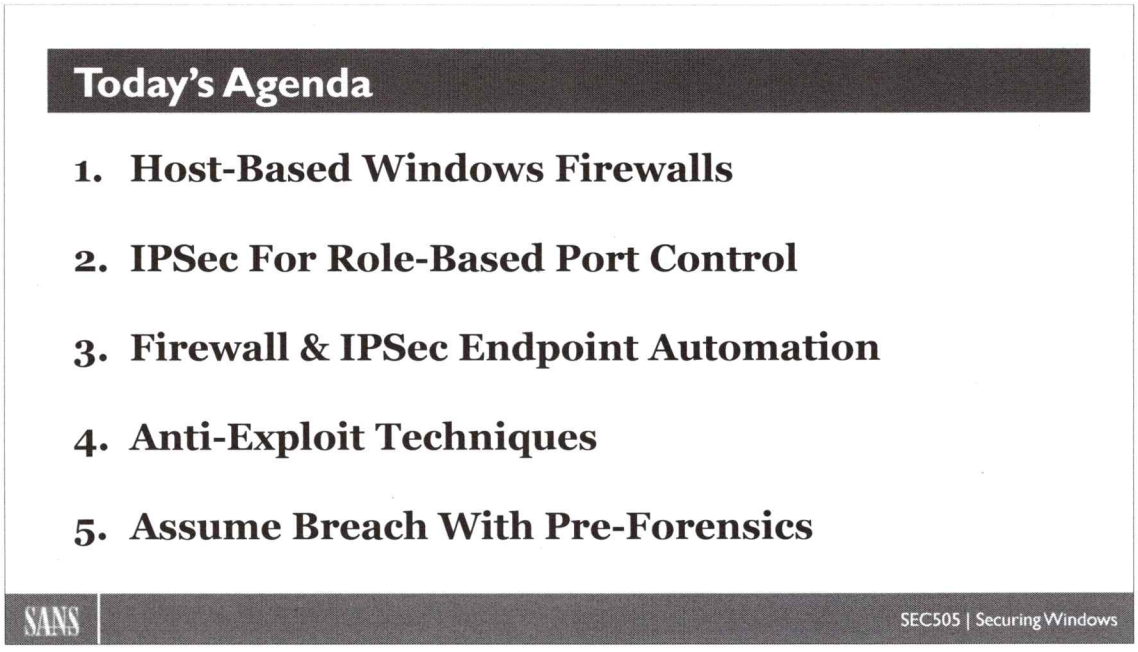
If you find a significant error of fact or an important omission which would clearly add value to the document, please e-mail the contact listed below. If your suggestion is incorporated, we would be pleased to list your name as a contributor.

Document Author: Enclave Consulting LLC, Jason Fossen (Jason@EnclaveConsulting.com)
Document Version: 33.0 (B02_01)
Last Modified: 31.Oct.2016

Contributors:

Enclave Consulting LLC, Jason Fossen: author.
William Dixon (www.microsoft.com): generous sharing of IPSec implementation details.
Anonymous (www.rsasecurity.com): whoever wrote RSA's Cryptography FAQ -- thank you!
P. Reuter (CSC): IPSec interoperability information.
Jeff Nieusma (StorageWay): IPSec and NAT clarifications.
Tina Bird (www.counterpane.com): technical review and NAT clarifications for IKE's UDP port.
Carla Wendt (www.sans.org): RFC number correction and other factoidal doo-dads.
Jeffrey Basista (SEI): ESP transport mode vs. NAT, and Windows 2003 IPSec issues.
Bryce Alexander (Vanguard): 802.1X authentication of wireless devices with IAS.
Chris Weber (FoundStone): interesting factoids and good surfing tips.
Dustin Decker (Dude): corrections to document references.
Sean McDermott (Talbots): correct location of ipseccmd.exe for XP.
David Perez (Human): lots-o-updates!
Bryan Simon (Human): MD5 command line options and other updates.
Tim Legge (Human): correction to IKE statement.
Robert Smith (CMU): nine hosts per line in the hosts file.
Bruce Meyer (ISAC): netsh.exe correct and other goodies.
Armond Rouillard (Army): many fixes and recommendations - Go Army!
Tom O'Reilly (CMS Energy): IPv6 ADMX template.
Greg Hall (tamu.edu): many typo and spelling fixes.
Charlie Martin (Human): accidental repeated slides.
Adam Haynes (Human): script fix.
George Allen (Army): null encapsulation description fix.
Lisa Peterson (Progressive): typos and fixes in this and other manuals.
Mike Pilkington (Human): DNS diagnostics logging.
Rick Moffatt (Human): auditpol.exe and advanced audit policy issues.
Dimitris Margaritis (CERT-EU): AppLocker variables and event ID codes.
Ginny Munroe (DeadlineDriven.com): tunns of tyepohs -- like this line!

Table of Contents



Today's Agenda

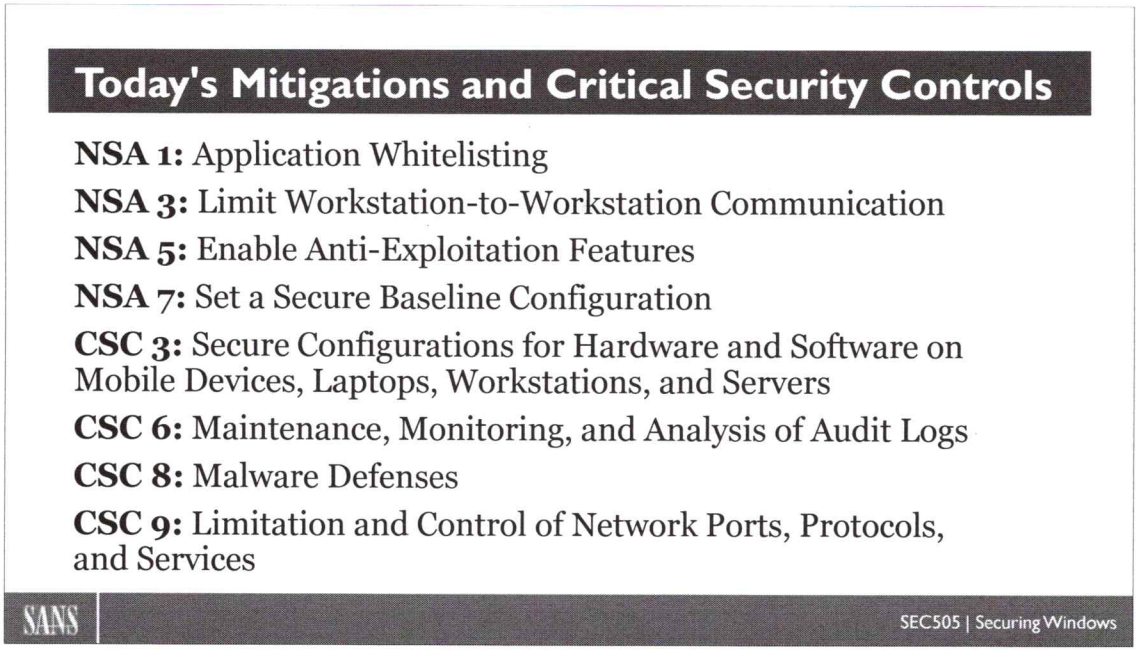
This seminar is about endpoint protection, but many of the techniques can be applied to servers as well. IPSec is not just for VPNs! This manual will explain how to use the Windows Firewall and IPSec for secure communications and port permissions. We will see how to automate the deployment of these rules using Group Policy and PowerShell.

This course assumes your endpoints are fully patched and users are out of the Administrators group, but what can we do beyond these steps to prevent malware exploitation? We will talk about anti-exploitation techniques, then, like application whitelisting and a benevolent rootkit from Microsoft named EMET.

Finally, although we try our best to prevent compromise, we have to assume that we'll eventually fail. To prepare for a future forensics incident response and to help the Hunt Team find active threats, we need to do pre-forensics. If we wait until after a compromise, it's too late. So what should be done now?

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

- Understand and configure the Windows Firewall.
- Use IPSec to control access to ports based on role.
- Deploy IPSec and firewall rules through Group Policy and PowerShell.
- Deploy AppLocker application whitelisting.
- Deploy Microsoft EMET for anti-exploitation.
- Enable Windows audit policies for SIEM consumption.
- Capture system snapshots to help the Hunt Team.



Today's Mitigations and Critical Security Controls

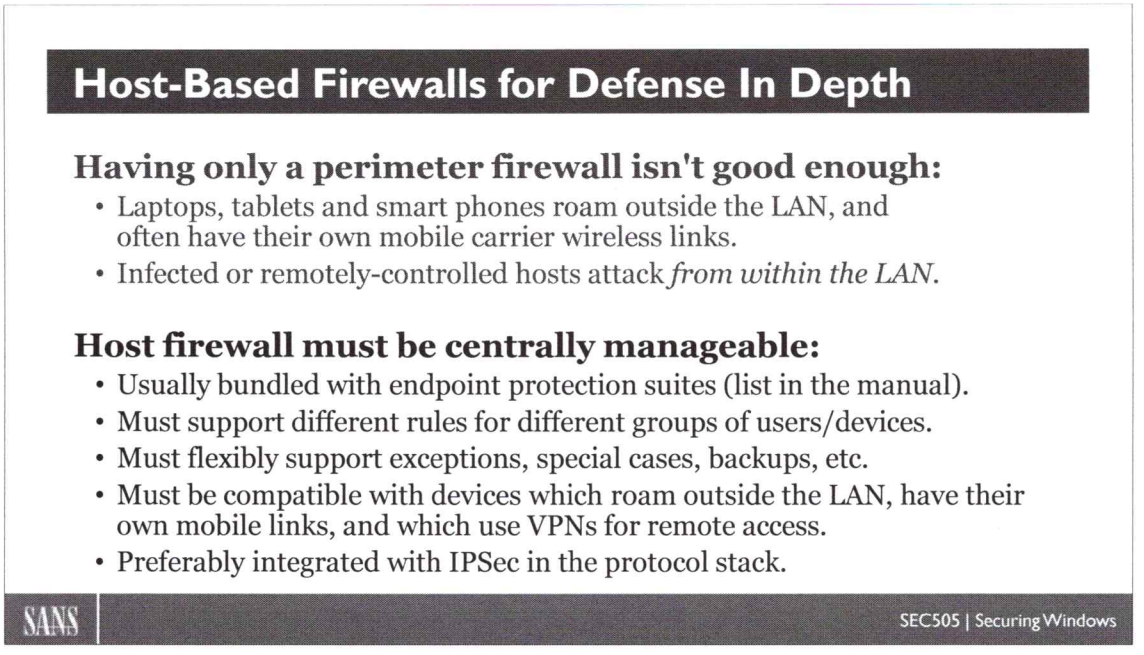
One mission of the National Security Agency (NSA) is to offer network security guidance through its Information Assurance Directorate (IAD). The NSA/IAD list of Top 10 Information Assurance Mitigation Strategies can be downloaded from the IAD web site (www.iad.gov).

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 1: Application Whitelisting
- NSA 3: Limit Workstation-to-Workstation Communication
- NSA 5: Enable Anti-Exploitation Features
- NSA 7: Set a Secure Baseline Configuration
- CSC 3: Secure Configurations for Hardware and Software on Mobile Devices, Laptops, Workstations, and Servers
- CSC 6: Maintenance, Monitoring, and Analysis of Audit Logs

- CSC 8: Malware Defenses
- CSC 9: Limitation and Control of Network Ports, Protocols, and Services



Host-Based Firewalls for Defense In Depth

An "enclave" is a protected network within a larger network. This is sometimes called "internal segmentation", but the issue is broader than that concept. Your corporate LAN is an enclave from the Internet, but within your LAN you might have a datacenter with critical servers, an R&D lab with sensitive data, a "dark" network for IDS and security operations, or special subnets just for the workstations of network administrators, HR, attorneys or corporate executives.

In general, not all machines are equally valuable or vulnerable, hence, it doesn't make sense to lump them all together when securing the network; instead, internal enclaves can be created to protect high-value assets not only from the Internet but also from the rest of your own internal network.

Enclave Implementation Methods

An enclave network can be implemented in many ways. A low-security enclave could just be a VLAN. Or, using Software Defined Networking (SDN), there could be multiple enclaves using segmentation rules for sets of virtual machines running on a cluster of Hyper-V or VMware servers. Or an internal firewall could separate a collection of subnets from the rest of the LAN, perhaps with reverse proxy servers to regulate access to the resources within the enclave. Or an internal VPN gateway might have to be traversed in order to access the subnets behind it inside the enclave. Or traffic restrictions could be enforced by the host-based firewalls and IPSec policies of the protected machines themselves, but the machines might still be directly connected to the rest of the regular LAN (or they could additionally have their own VLAN, firewall, etc.). For the most secure enclave, the enclave might be physically separated (air-gapped) from the rest of

the LAN, such as for a malware analysis lab, work area for classified information, or the control systems at a utility company.

Enclaves with IPSec and Host-Based Firewalls?

At a larger scale, even average-value systems might be separated into networks with well-defined boundaries, similar to how WAN links define the connections between branch offices and the main office. In this case, sections of the local LAN would be treated somewhat like "branch offices" in that there would be choke points defined by routers or firewalls to enforce traffic restrictions and IDS/IPS sensors to observe what is permitted through those choke points. Because of the mixture of different types of systems in these internal "branches", though, the firewall and VLAN rules at the choke points will have to be rather permissive. And because the roles that computers and users play are constantly changing, not to mention their IP addresses and physical locations, it will be difficult to keep the policy rules up to date.

What we need is something like per-group firewall rules for different groups of users and computers based on their various *roles in the organization*, not based on their IP addresses or segment locations. These rules would have to be centrally manageable and easily changed as needed.

Host-Based Firewalls To The Rescue

Defense in depth requires host-based firewalls on almost all systems. Simply having a perimeter firewall isn't good enough anymore. Using host-based firewalls is also a flexible and (relatively) inexpensive way to implement internal enclave networks. In a large enterprise, the most important feature of the host firewall is its ability to be centrally managed. Different sets of computers will require different firewall rules, and these rules will need to be regularly edited, especially when changes are made to how these systems are backed up, managed and remotely troubleshot. If a host-based firewall cannot be easily managed from a central management console using custom groups of computers and/or users, then it is not very useful for enclave networks or defense in depth.

For example, host firewalls can help to enforce network security policies, such as compulsory proxy server usage for Internet access and prohibiting the installation of HTTP/FTP/Telnet servers on unauthorized systems. Malware can spread over the network by means other than e-mail too; Conficker, for example, attacked the Server service using SMB on TCP/445. Malware can open new listening ports for backdoors, connect outbound to command-and-control servers, upload stolen files, and anything else for which the designer wishes to use the network. By enforcing good host firewall rules, we can try to thwart malware's lifecycle, especially on the not-yet-infected systems. But because users, computers, segments, applications and organizational roles are constantly changing, we need host-based firewalls which can be quickly changed.

Since we cannot use host-based firewalls that do not support centralized rule management, you'll find that most firewall applications worth considering come bundled as part of a package with other endpoint security software. When you are selecting an anti-virus or patch management vendor, you will likely also be choosing a host-based

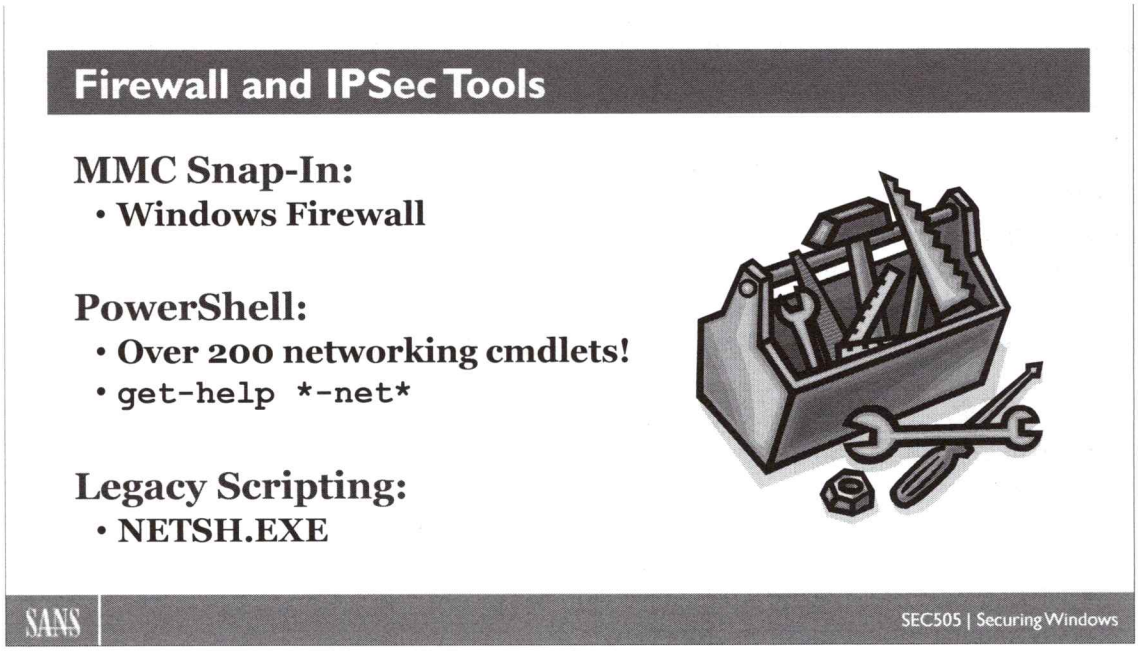
firewall vendor at the same time, hence, include the features of its host-based firewall too in your evaluation criteria. Specifically, it's important to find out if 1) computers and users can be categorized into custom groups or roles to which custom sets of firewall rules can be assigned, 2) how quickly changes to the firewall rules can be pushed out to large numbers of systems, 3) how easy it is to define exceptions or special cases in the firewall rules to allow remote administration, backups and troubleshooting, and 4) how well the firewall behaves for users with laptops who frequently move in and out of the corporate LAN or who often work remotely via VPN.

Here is a partial list of the host-based firewalls available which can be centrally managed:

- Blink Professional (www.eeye.com)
- Checkpoint Endpoint Security (www.checkpoint.com)
- F-Secure Business Suite (www.f-secure.com)
- Kaspersky Internet Security (www.kaspersky.com)
- McAfee Total Protection for Endpoint (www.mcafee.com)
- Panda Security for Enterprise (www.pandasecurity.com)
- Symantec Endpoint Protection (www.symantec.com)
- Trend Micro Enterprise Security for Endpoints (www.trendmicro.com)

Most of these firewalls come as a part of the vendor's integrated suite of endpoint protection products, hence, the firewall is just part of a packaged bundle including AV, patch management, application blacklisting and inventory components.

Without launching a debate about which is best, let's look at one example you probably already own. Let's look at the host-based firewall which comes bundled with Server 2008, Windows Vista and later.



Firewall And IPSec Tools

There are a variety of tools for managing the Windows Firewall and IPSec.

Windows Firewall with Advanced Security (Vista and Later)

Starting with Windows Vista, the primary tool for managing IPSec is also the MMC snap-in for managing the firewall: Windows Firewall with Advanced Security. Specifically, it is the Connection Security Rules container in that snap-in, as well as the IPSec Settings tab in the properties of the snap-in itself.

PowerShell 3.0 and Later

PowerShell 3.0 and later include over 200 cmdlets for managing IPSec, firewall rules, and networking settings in general. It's best to assume that if a networking setting can be managed with a graphical tool, it can be managed from the command line as well.

PowerShell 3.0 is included by default on Server 2012 and Windows 8, but you can always download the latest version from http://www.microsoft.com/powershell (though some features may require a recent operating system version).

To see a listing of the cmdlets for IPSec, firewall rules, network interfaces, TCP/IP, etc.:

```
Get-Help *-net*

# Multiple wildcards can be used to narrow the output list.

Get-Help *-net*ipsec*
Get-Help *-net*firewall*
Get-Help *-net*ip*

# Get the full help for a cmdlet with the -Full switch.

Get-Help Set-Netipaddress -full
```

NETSH.EXE

NETSH.EXE is a command-line utility for managing network adapters, protocols and some network services such as WINS, DHCP, and, on Windows Server 2003 and later, IPSec too. NETSH.EXE is one of the most important new command-line tools from Microsoft. Search on the tool's name in Windows XP/2003 Help for more information, and note that it does support the "-r" switch to operate against remote systems.

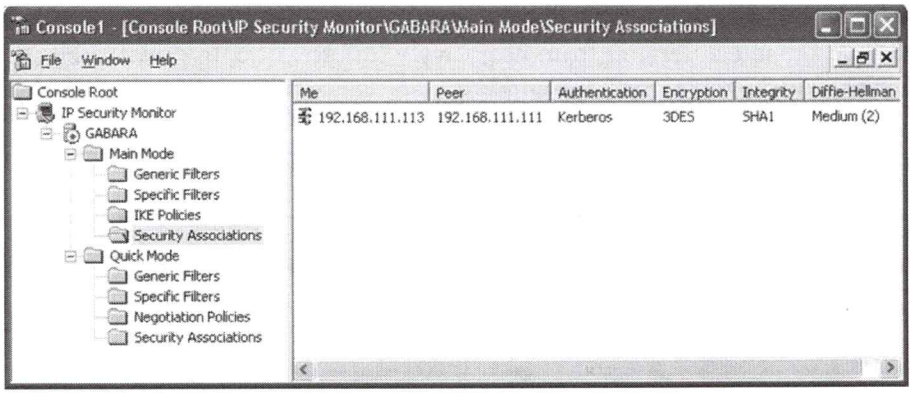
For a taste of what NETSH.EXE can do for you, try these commands one at a time:

```
netsh.exe int ip show address

netsh.exe int ip set ?

netsh.exe int ip show offload
```

The last command above will show the IPSec-offload characteristics (if any) of your network adapter card. These network interface cards have on-board processors for handling some of the CPU work of doing IPSec. On Windows Server 2003 you can also manage your IPSec settings with NETSH.EXE. This will be discussed later in the manual along with IPSECPOL.EXE (2000) and IPSECCMD.EXE (XP).

IP Security Monitor MMC Snap-In (XP and Later)

IP Security Monitor is an MMC snap-in which can be used to view IPSec statistical data and session information on local or remote Windows XP or later systems (not 2000). It can display hostnames instead of IP addresses, if desired, and supports a query feature that helps to determine which IPSec rule would govern certain types of traffic. It is a GUI version of the data that can be acquired with "ipseccmd.exe show all".



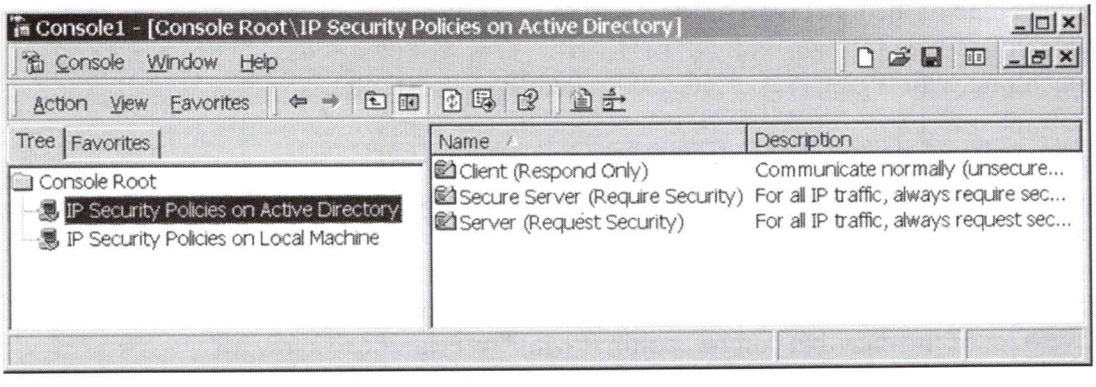
In Windows Vista and later, some of this snap-in's functionality has been added to the Monitoring > Security Associations container in the Windows Firewall with Advanced Security snap-in too.

IP Security Management MMC Snap-In (2000/XP/2003)

The primary IPSec configuration tool for Windows 2000/XP/2003 is the "IP Security Policy Management" MMC snap-in. It is available on newer operating systems as well even though it is no longer primary for them. This utility can be used to do the following:

- Configure local IPSec Policies in a computer's registry.
- Assign local IPSec Policies from a computer's registry.
- Configure IPSec Policies in Active Directory.
- Create, edit and delete IPSec Filters, Filter Actions, and other IPSec options.
- Check IPSec Policy integrity.
- Restore the built-in Policies to their default settings.
- Import/export Policies from/to files.

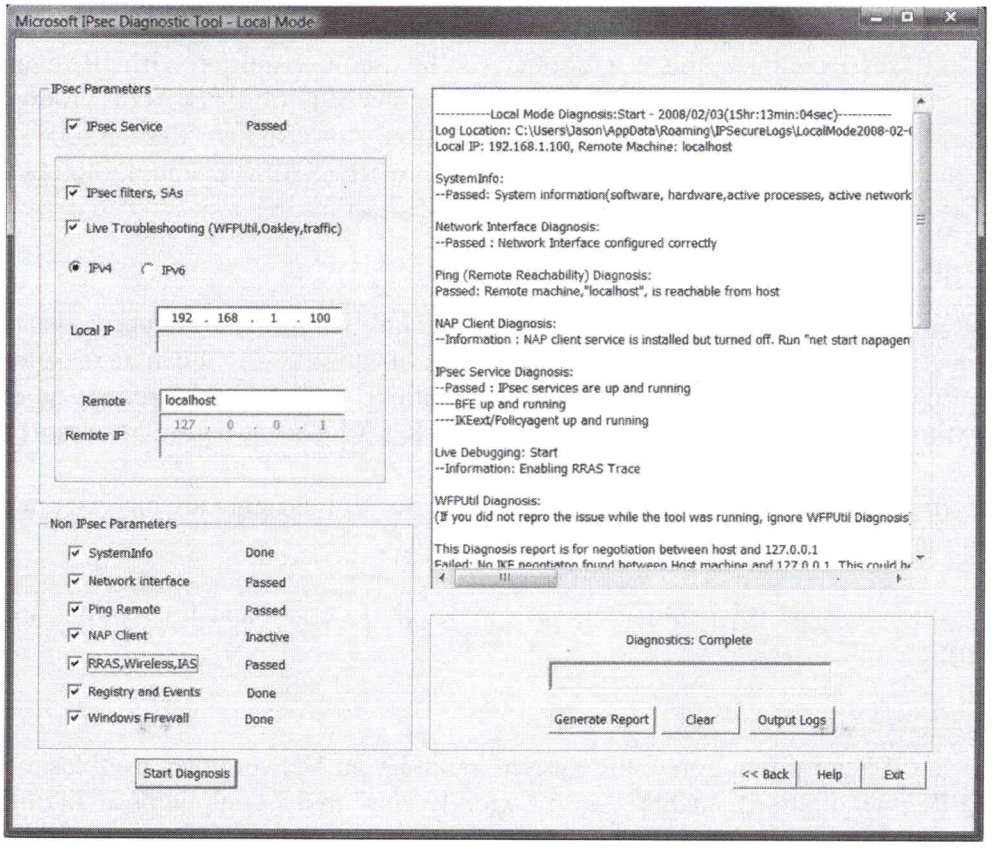
The IP Security Policies snap-in must be installed by hand. It is not in the Administrative Tools folder. When installed, the snap-in can be used to manage IPSec Policies stored either in Active Directory or on the local machine.



Checking Policy integrity will verify that the Filters and Filter Actions linked to the Policy are valid, that the locally cached Policy matches the Policy stored with Group Policy, and that the internal structure of the IPSec Policy is consistent. You can also export/import Policies for backup purposes. For either, right-click on the "IP Security Policies" snap-in and select All Tasks.

IPSec Diagnostic Tool

The IPSec Diagnostic Tool is a free Microsoft download which can examine the IPSec service, driver, current associations, event log entries, and other IPSec-related settings to help troubleshoot IPSec problems. When run, it examines these settings and produces a report to help guide the administrator in the troubleshooting process. The tool works on Windows XP/2003/Vista/2008 and later. To find the download, do a Google search on "site:microsoft.com ipsec diagnostic tool download".



NETDIAG.EXE (2000/XP/2003)

NETDIAG.EXE is one of the Support Tools that is found on the Windows CD-ROM in the \Support\Tools folder or downloaded from Microsoft's website.

The NETDIAG.EXE command-line utility is a comprehensive networking diagnostic tool. It performs a series of in-depth tests against protocols, adapters and some network

services to help troubleshoot networking problems. One of the tests it can perform is for IPSec.

The following commands will run tests against IPSec on the local Windows 2000 computer and output operational statistics:

```
netdiag.exe /test:ipsec /v

netdiag.exe /test:ipsec /debug
```

The /v switch only shows Phase I offers, while /debug will show both Phase I and Phase II offers. The /v switch shows packet statistics, while /debug does not.

IPSECCMD.EXE (XP)

IPSECCMD.EXE only works on Windows XP and installs with the XP Support Tools. Similar to IPSECPOL.EXE, this is a tool that can be used to configure virtually every IPSec option from the command line. It is also used instead of NETDIAG.EXE on XP when viewing IPSec statistics and session information. On Windows Server 2003, though, the features of this tool have been moved into NETSH.EXE, and it's likely that future XP Service Packs will do the same to NETSH.EXE on XP too.

IPSECPOL.EXE (2000)

IPSECPOL.EXE is a Windows *Resource Kit* command-line utility which provides most of the functionality of the IP Security Management MMC snap-in. It can be used to manage Filters, Filter Actions, IKE Phase I negotiations, and other features. (You can also download the tool for free as a part of IISLOCK.EXE package from Microsoft.)

Importantly, IPSECPOL.EXE can set temporary IPSec Policies that are cleared after rebooting or restarting the IPSec Policy Agent service.

A later section will discuss IPSECPOL.EXE in detail. Its command-line options are somewhat complex.

Event Viewer

You can log IPSec information to the System log in Event Viewer when troubleshooting. To audit IKE negotiations, enable "Audit Logon Events" in the computer's audit policy. To audit IPSec Policy changes, enable "Audit Policy Changes" as well. IPSec driver event data can be audited once per hour by setting the DiagnosticMode value to 1 under HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\IPSEC\.

Performance Monitor

There are a large number of Performance Monitor counters related to IPSec, such as total number of security associations, authentication failures, bytes encrypted, etc. These can all be logged and graphed over time for troubleshooting.

The Good:

- **Built-In (Free)**
- **Enabled by Default**
- **Integrated with IPSec**
- **Stateful Filtering:**
  - Dynamic RPC Ports
  - Application/Service-Aware
  - IPv4 and IPv6
  - Ingress and Egress Filtering
- **Centralized Management:**
  - Group Policy & PowerShell
- **W3C Extended Logging**

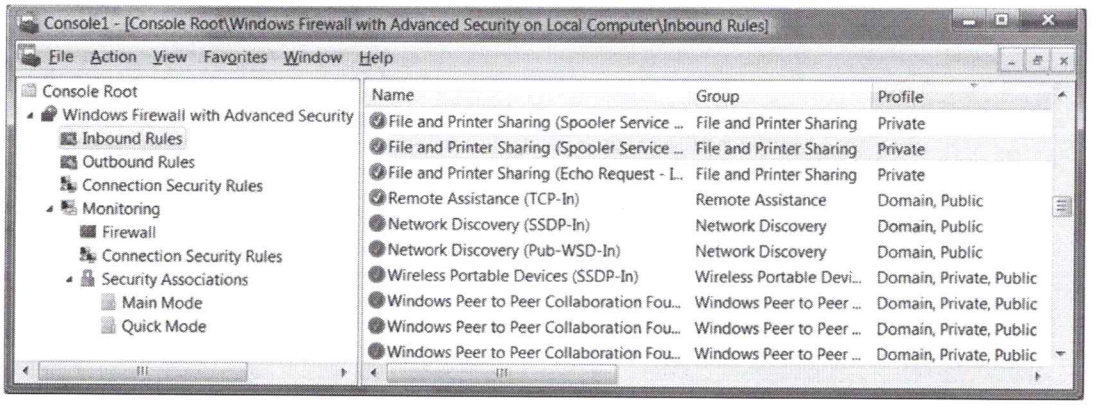
The Bad:

- **No network IDS features.**
- **No host IDS features.**
- **No user behavioral monitoring.**
- **No centralized logging.**
- **IPSec adds complexity.**
- **Roaming, multi-homed network profiles also add complexity, but this is probably unavoidable.**

SEC505 | Securing Windows

Windows Firewall

The original version of Windows XP had a built-in firewall called the Internet Connection Firewall (ICF). Service Pack 2 for XP drastically improved the ICF and it was renamed to the Windows Firewall (WF). Starting with Windows Vista, the firewall was again enhanced and this time not-so-eloquently renamed to Windows Firewall *with Advanced Security* (WFAS). Importantly, the IPSec driver in Vista and later is tightly integrated with WFAS, and IPSec is primarily managed through the WFAS snap-in.



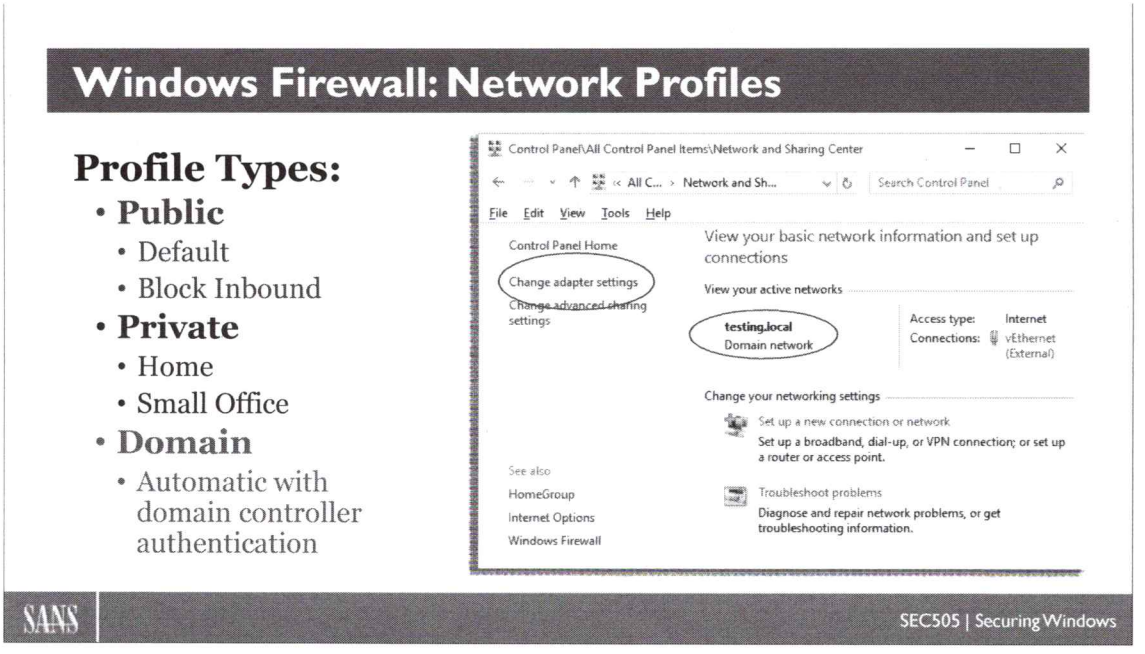
WFAS is built into the operating system, stateful, easy to configure, supported by Microsoft, can be managed through Group Policy or custom scripts (NETSH.EXE), and works with all types of interfaces (LAN cards, 802.11 wireless, dial-up connections, VPN tunnels, etc.). WFAS also is compatible with the Internet Connection Sharing service, provides good ASCII text logging in W3C Extended format, and can easily be configured to permit access to services on the host itself or to another machine behind it on the LAN,

such as to an HTTP or FTP server. The WF in XP-SP2 had only meager egress filtering, but WFAS filters both inbound and outbound traffic with equal facility.

On the bad side, though, WFAS lacks the sophisticated intrusion detection capabilities of some other popular personal firewalls, doesn't work on Windows 2000/XP/2003, and there's no built-in support for automatic upload of log data to a central server. The flexibility of WFAS also comes with an administrative price: complexity. A large part of that complexity comes from having different rules for different network profiles and the IPSec integration.

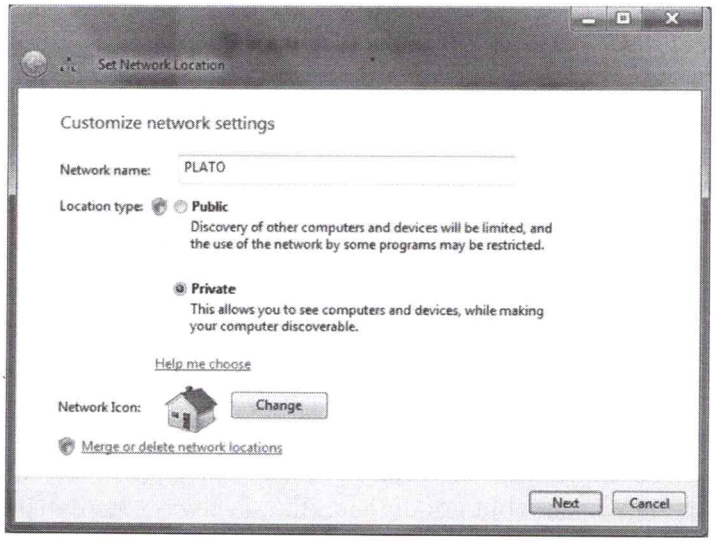
Defense In Depth

It's important to understand that WFAS is not just for laptop and home use. The firewall is supposed to be enabled on all machines, both workstations and servers, whether they are "safe" inside the corporate LAN or outside on the Internet. Defense in depth means you no longer have just a perimeter firewall, you are performing filtering on every single computer under your control. To make this possible, however, you must be able to 1) manage the firewall across the enterprise and 2) provide filtering exceptions for remote administration, troubleshooting, backups, monitoring and other necessary communications (while blocking the rest). IPSec actually makes this *easier* because you can make firewalling exceptions but only for traffic secured with IPSec.



Windows Firewall: Network Profiles

When a Windows Vista or later computer is connected to a network, that network will be categorized as a Public, Private or Domain network. (Microsoft also calls these categories "network location types" or "network profiles"). A domain network can be used to access domain controllers for the computer's Active Directory. A private network is a trusted network that does not have domain controllers, e.g., a home or small office network. A public network is everything else, such as airports and coffee shops with Internet access.



If a computer is a domain member, and if a domain controller can be contacted through the network via LDAP, then the domain category will be automatically assigned. The domain category cannot be manually assigned, this category is automatically assigned only after a domain controller is contacted. LDAP also requires DNS and Kerberos.

On Windows 7, Server 2008-R2 and later, categories are assigned on a per-interface basis, hence, if a controller can be reached through only one interface on a multi-homed computer, then that one interface will be categorized as a domain interface even though its other interfaces may be categorized as public and/or private.

However, on Vista and Server 2008, the domain category will only be assigned to an interface if a controller can be contacted through *every* network interface on that computer, including its wireless and VPN interfaces; hence, only if *all* interfaces can be categorized as domain will any interface be categorized as domain. This is generally not a problem with internal LAN servers, but on roaming laptops running Vista it created big problems. This design flaw is yet another reason to avoid using Windows Vista. The flaw was fixed in Windows 7 and Server 2008-R2, but by then the damage had been done.

Different Firewall Rules for Different Network Profile Types

The Windows Firewall can have different sets of rules for each network category and will switch the rulesets automatically as the computer is disconnected and reconnected to different networks. If you have multiple network interfaces with different profile assignments, then different sets of firewall rules will be applied simultaneously according to each interface's profile. Generally speaking, rules for the public network are the most strict, private networks are somewhat strict, while rules for the domain network are the least strict, i.e., they allow inbound access to the most ports.

Manage Network Profiles

You can see and edit the category of the network to which you are currently attached by going to Control Panel > Network and Sharing Center > Change Adapter Settings. The computer will remember your settings the next time you connect to that network.

Unfortunately, another design flaw that still exists today is that a network administrator cannot override the decision of the NLA service to *not* categorize an interface as domain; for example, on a server exposed to the Internet, if the Internet-exposed interface is categorized by NLA as domain (correctly or incorrectly), then it's not possible to override the NLA decision and permanently assign a different profile type. It doesn't mean that other changes can't be made to secure the interface, but it is very annoying.

On Windows 8 and later, you can run Get-NetConnectionProfile to see the current network category of each interface, then run Set-NetConnectionProfile to assign either public or private to an interface (but not domain, that is always set dynamically).

To see the current network category for each live interface:

```
Get-NetConnectionProfile
```

The service named "Network Location Awareness" (NLA) is what receives notifications of network configuration changes, is what attempts to contact domain controllers when determining network category type, and is what notifies the Windows Firewall of changes to interface category types as well. Search the Internet on the name of this service to read more about it and its troubleshooting steps. Often, just restarting the service will fix any associated problems.

To restart the Network Location Awareness (NLA) service and its dependents:

```
Restart-Service -DisplayName "Network Location Awareness" -Force
```

> **Tip:** If you change your networking settings and believe that an interface should be re-categorized as a domain interface, but it is not being re-categorized automatically or quickly enough, then restart the NLA service.

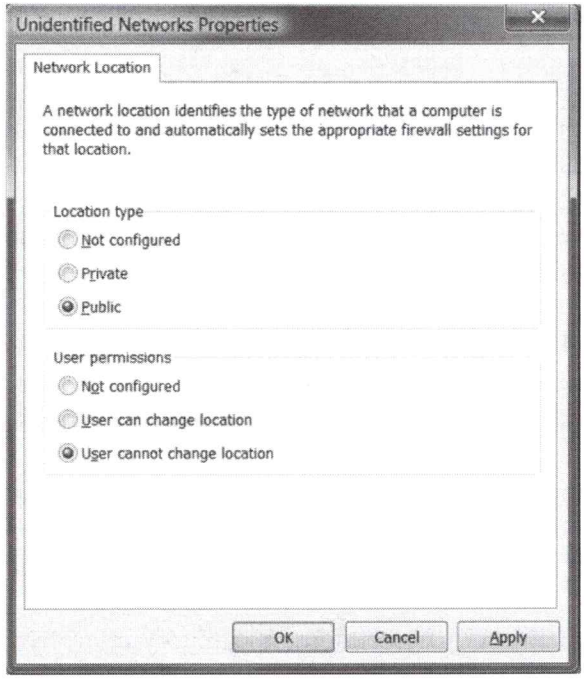
To ensure that NLA can correctly detect domain category interfaces, confirm that no perimeter or internal firewalls are blocking UDP/53 (DNS), TCP/88 (Kerberos) or UDP/TCP/389 (LDAP) between the Windows device and internal DNS servers and domain controllers.

In particular, if the Public Profile tab in the properties of the Windows Firewall snap-in is set to Block for outbound traffic, then the following are the minimum outbound Windows Firewall rules necessary on the endpoint to allow any public-categorized interfaces to be recategorized as domain interfaces:

- Allow outbound UDP 53 for the DNSCACHE service by name
- Allow outbound TCP 88 for %SystemRoot%\System32\lsass.exe
- Allow outbound UDP 389 for %SystemRoot%\System32\lsass.exe
- Allow outbound TCP 389 for the NLASVC service by name

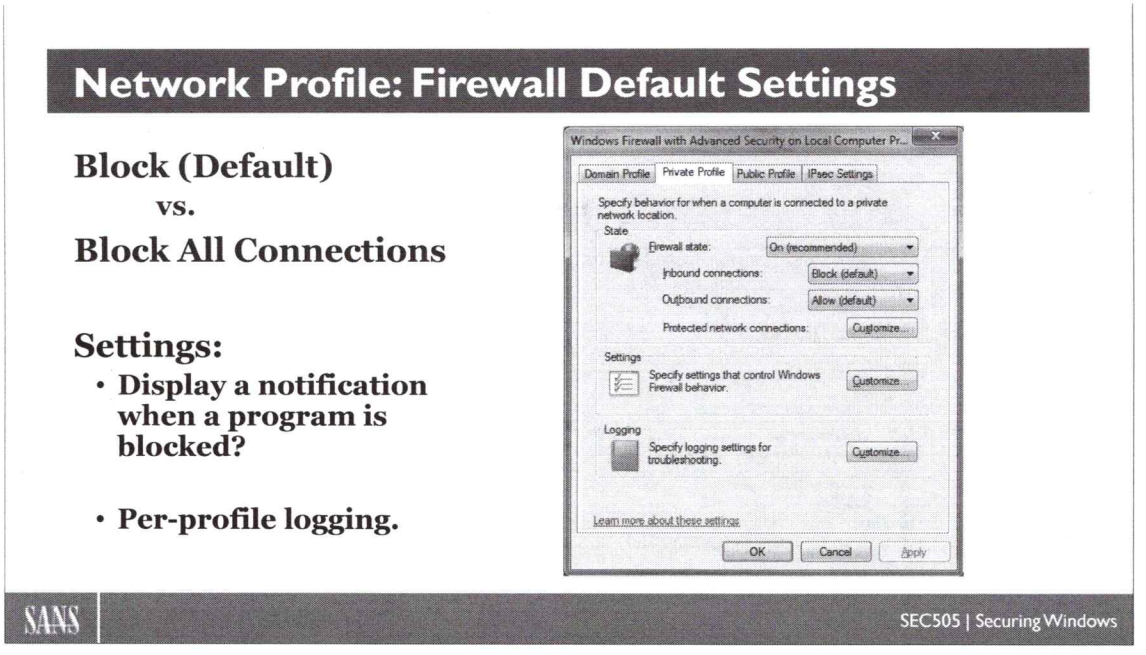
Group Policy Control Over Profiles

Note that you have Group Policy control over network profiles and their defaults. Inside a GPO, navigate to Computer Configuration > Policies > Security Settings > Network List Manager Policies, and here you can determine whether users can change the profile type for a given network and what the default profile should be for new networks.



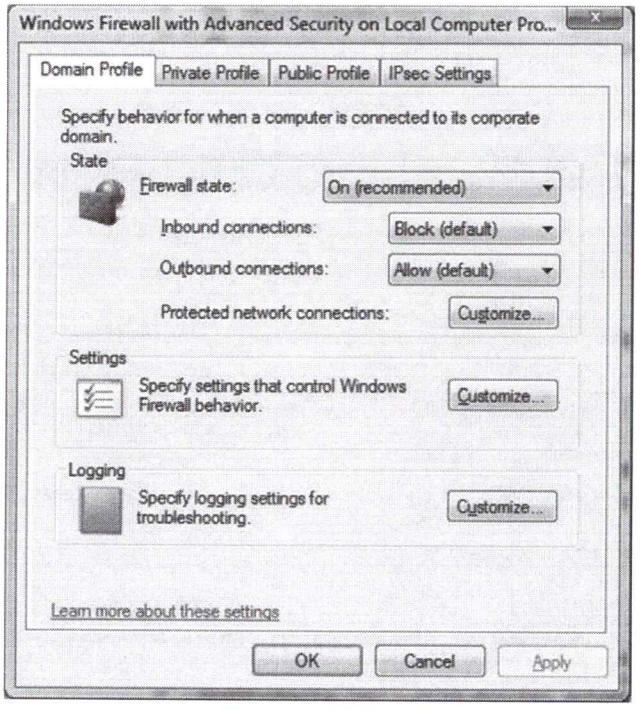
Sharing And Discovery Section

In the Network and Sharing Center applet there is a link to "Change advanced sharing settings". On Windows 8 and later, when connecting to a new network for the first time, you'll see a similar prompt about "sharing". These sharing options are actually for managing the firewall in a user-friendly way. When, for example, file sharing is enabled or disabled, certain firewall rules are automatically modified to allow or block the sharing. File sharing might be enabled for private profile networks, but disabled for public networks. Exactly what these firewall rule changes are we'll see in a moment.



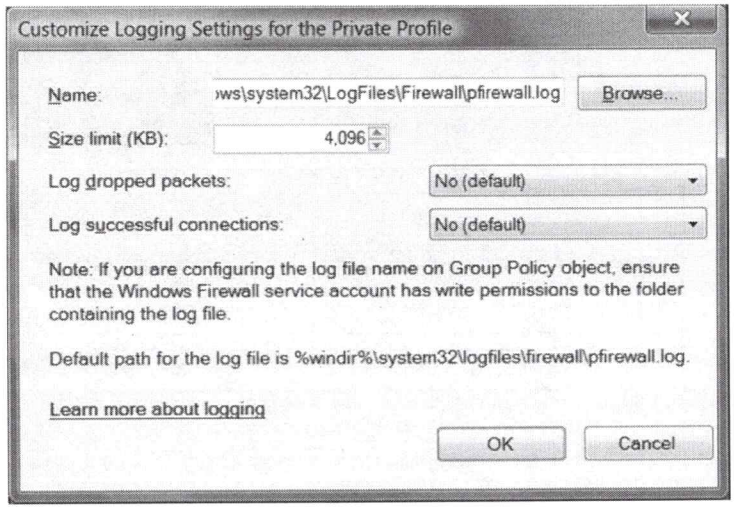
Network Profile: Firewall Default Settings

There are different default settings for the different network location types. To edit these per-network defaults, right-click the WFAS snap-in > Properties > choose the appropriate tab: Domain, Private or Public.



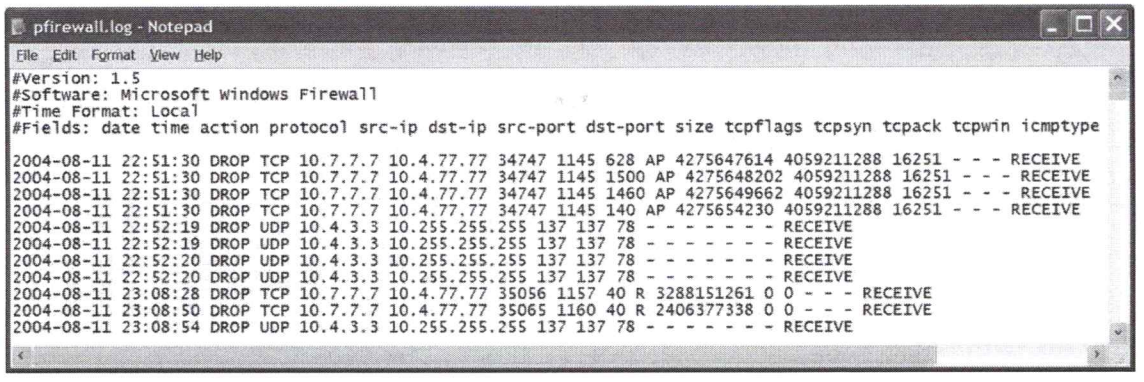
For each profile, you can enable/disable the firewall, block/allow inbound or outbound connections, configure logging options, and specify whether the user is notified when a program is prevented from receiving inbound connections (giving administrative users the opportunity to allow them in for that program).

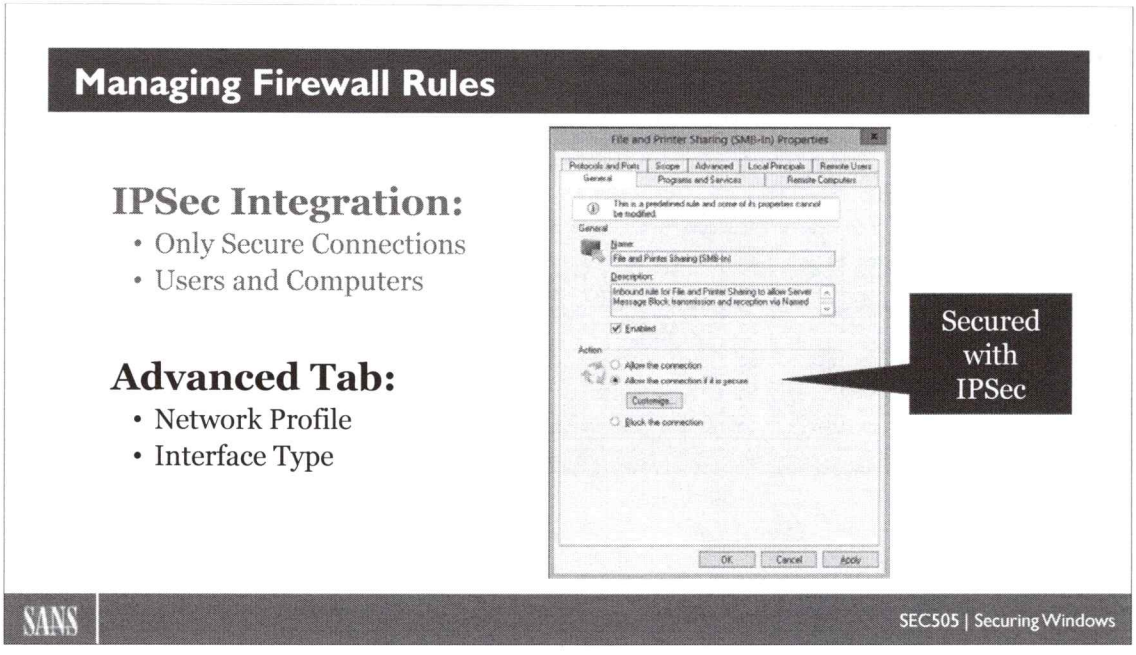
For inbound connections, if the option is set to "Block All Connections" then all inbound connections are blocked even if there is a rule which would normally allow it, while the "Block (Default)" option blocks only those inbound connections for which there isn't a rule to allow them.



Logging is written to an ASCII text file (`pfirewall.log`) in W3C Extended format. Note that all dropped packets are logged, if those packets are logged at all, but only the

initial packet in a successful connection is recorded in order to avoid killing performance by logging all the permitted packets that follow. The maximum log size is 32MB.





Managing Firewall Rules

WFAS performs both ingress and egress filtering. Connections are regulated by the rules in the Inbound Rules and Outbound Rules containers in the WFAS snap-in. To create a new rule, right-click the Inbound/Outbound Rules container > New Rule. To edit a rule, simply double-click it.

The property sheet of each rule has the follow tabs and options:

- **General**: Allow or block, or allow only if secured with IPSec. The Customize button is for customizing the IPSec options, such as encryption required, signature-only permissible, null encapsulation, and block rule override.

- **Programs and Services**: The exact program binary, background service(s), or Metro APPX software package(s) to which the rule applies.

- **Remote Users**: If only IPSec-secured connections are allowed (General tab) and if the IPSec authentication protocol can identify the exact user account of the client (such as with Kerberos), then user accounts and groups can be selected from Active Directory (not the local accounts database or the workgroup) in order to limit connections to/from just those particular users. This is how a "share permission" on a TCP or UDP port can be implemented with IPSec.

- **Remote Computers**: If only IPSec-secured connections are allowed (General tab) and if the IPSec authentication protocol can identify the exact computer account of the client (such as with Kerberos), then computer accounts and groups can be selected from Active Directory (not the local accounts database or the workgroup)

in order to limit connections to/from just those particular computers. This is how a "share permission" on a TCP or UDP port can be implemented with IPSec.

- **Local Principals:** Unlike the Remote Users/Computers tabs which must specify accounts or groups from AD, the Local Principals tab allows the selection of a local SAM database user or well-known identity (on a domain controller, only global accounts can be selected anyway). The tab also allows the selection of WinRT application capabilities such that any Metro app which has the selected capability will have its network traffic filtered by this firewall rule.

- **Protocols and Ports**: Select any protocol, including IPv6, or any port(s), including dynamically-assigned RPC and NAT ("Edge Traversal") ports, or any ICMP protocol, including custom ICMP type and code numbers, to which the rule applies.

- **Scope**: Limit the source and/or destination IP address(es) to which the rule applies, including DHCP-assigned WINS, DNS, DHCP and default gateway addresses.

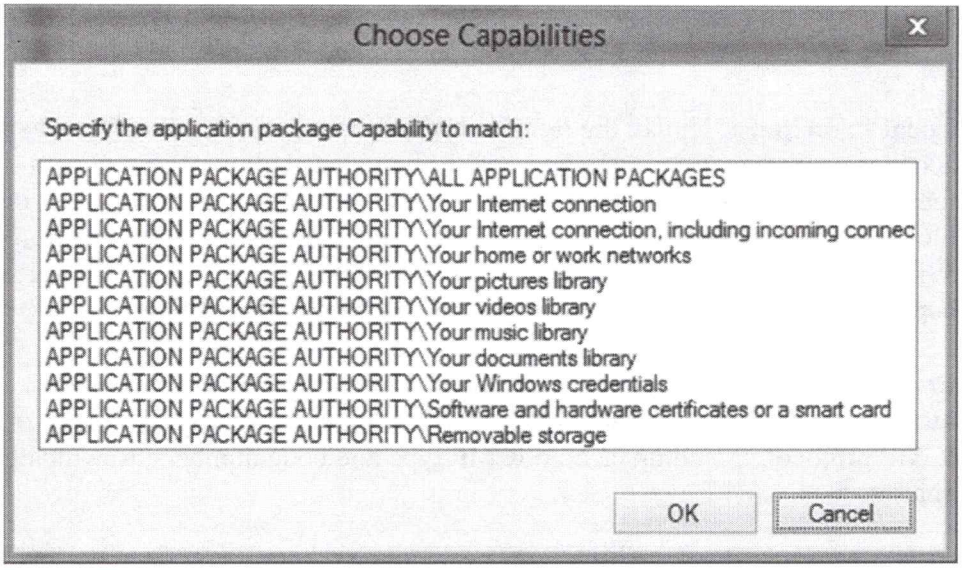
- **Advanced**: Specify the network location profile(s) and the types of network interfaces (wireless, VPN/dial-up, physical NIC) to which the rule applies, as well as whether an Internet-accessible IP address should try to be obtained (inbound rules only) for the sake of publishing a service to the Internet without the aid of a NAT-ing device in front of the computer.

Filtering the Display of Rules

With a large number of firewall rules to sift through, it can be difficult to focus on just the rules which are relevant to you. Notice that you can right-click the Inbound Rules or Outbound Rules containers and filter by profile, enabled state, or grouping. This is very useful when, for example, you want to see only the enabled rules for the public profile.

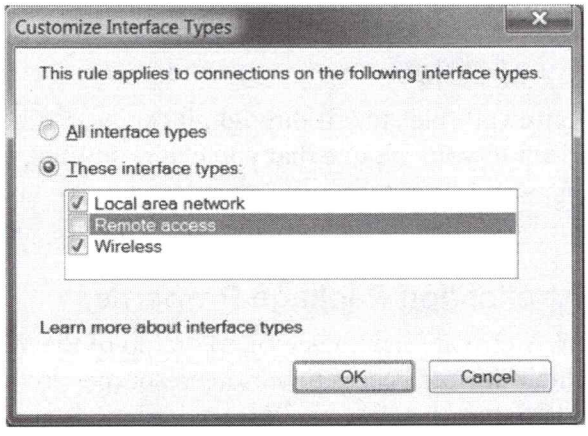
Local Principals: Application Package Properties

On Windows 8 and later, Metro applications written on top of the Windows Runtime API (WinRT) must declare their desired access to various resources so that the user can approve the access first and then the OS can enforce restrictions against any access attempts to the other unapproved and undeclared resource types. Access to these resources are called WinRT "capabilities" and the Windows Firewall is capabilities-aware. You can see the list of available capabilities in the screenshot below, which is what you will see when configuring a local principal in a firewall rule. This feature could be used, for example, to block all Internet access to any Metro app which has been granted the "Your documents library" capability or the "Removable storage" capability. In the WinRT API, a Metro app is a principal in the sense that it has a SID number.



Advanced Tab: Profile and Interface Types

Each interface is assigned a network profile type when it connects to a live network. You can have different rules for different profiles, hence, different rules for different interfaces on a multi-homed machine simultaneously. But notice on the Advanced tab that you can also apply different rules to different types of interfaces based on their media: LAN, wireless, and remote access (VPN and dial-up).



General Tab: "Allow Only Secure Connections"

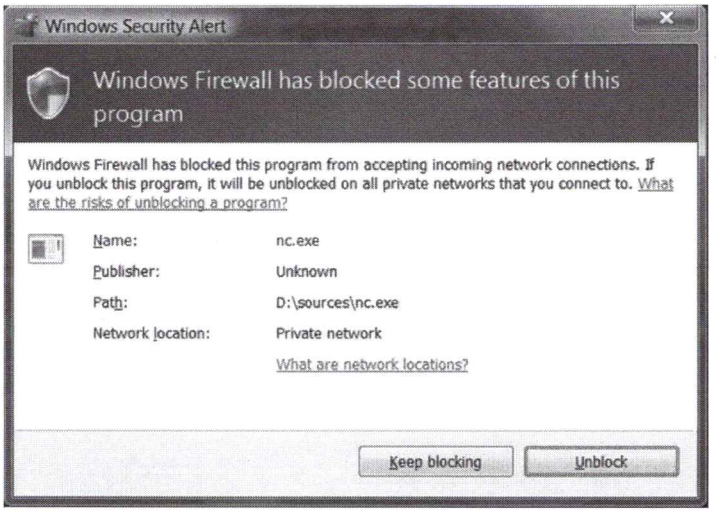
Note that a "secure connection" is traffic signed using IPSec Authentication Header (AH) or Encapsulating Security Payload (ESP) with the encryption disabled, while a "secure connection with encryption required" is traffic both signed and encrypted using IPSec ESP with the encryption enabled.

If an IPSec-secured connection is only for particular users or computers (as defined on the Users and Computers tab), then the IPSec channel must be authenticated with either Kerberos or a certificate, and those users and computers must exist in Active Directory.

The IPSec settings are configured using the Connection Security Rules container in the WFAS snap-in. If an appropriate IPSec Connection Security Rule is not created for a firewall rule that requires it, the firewall rule will not allow the traffic. The default IPSec settings used can be seen by right-clicking the WFAS snap-in > Properties > IPSec Settings tab.

Defining Program Exceptions

Another way for a user with administrative rights to add an inbound rule is to simply launch a program which attempts to listen on a new port number. This action causes a dialog box to appear which alerts the user to the attempted port binding. In the screenshot below, netcat was run to make it listen on TCP port 7890 and connect an incoming session to a new instance of CMD.EXE ("nc.exe –L –p 7890 –e cmd.exe"). Maybe the user did this knowingly and deliberately, or maybe the system has been compromised and a back door is being opened.



The dialog box gives the user two choices:

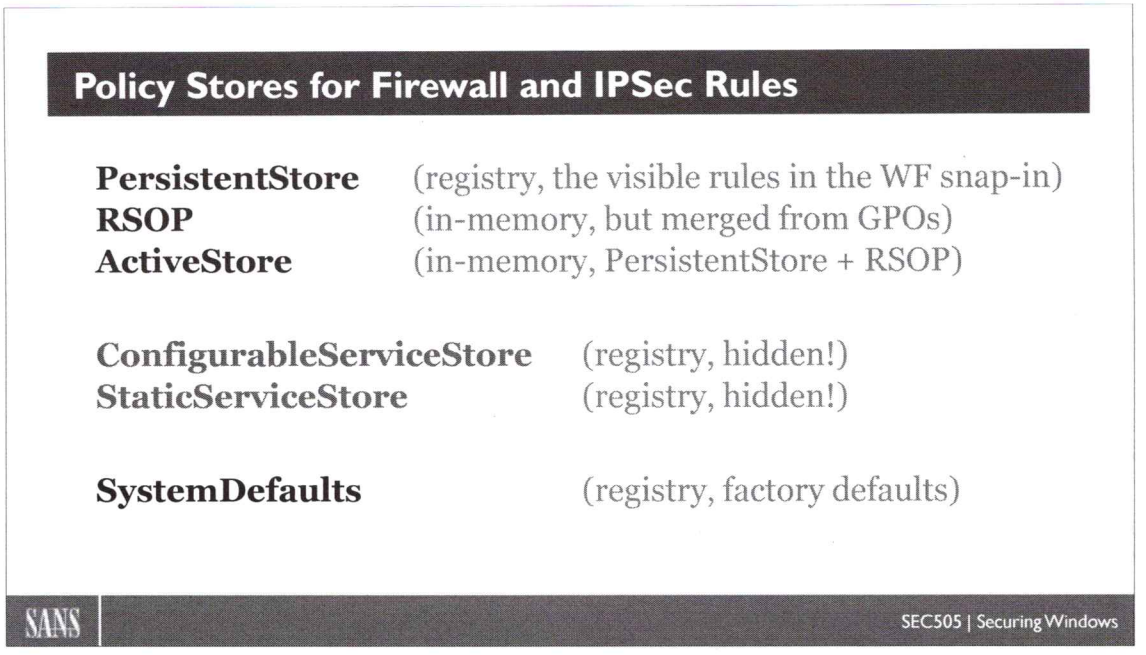
- Keep Blocking: Don't allow the program to acquire a listening port. Train your users to choose this option when there is any doubt.

- Unblock: Create inbound rules for this program to permit it to listen on the port it is currently requesting and on any other port it may request in the future.

If you don't want this dialog box to ever appear, right-click the WFAS snap-in > select the tab for the relevant network profile > Customize button for settings > select No to display a notification (default is Yes). For non-technical users, this is perhaps the best thing to do.

Keep in mind that you should limit the IP addresses of the other machines which you allow to connect to you as much as possible. This is done on the Scope tab in the

properties of the inbound rule. At a minimum, don't set the remote IP addresses on the Scope tab to "Any IP Address".



Policy Stores for Firewall and IPSec Rules

There are multiple "stores" of firewall and IPSec rules. Each store is a set of zero or more rules for either firewall rules or IPSec rules. Firewall and IPSec rules are not mixed together in one store; there are separate stores for rules of each type. Each store has a name which can be used as an argument to the -PolicyStore parameter in various firewall and IPSec cmdlets (it's why there are no space characters in their names).

The names of the stores for firewall and IPSec rules are:

- PersistentStore (also known as the Static store)
- ConfigurableServiceStore
- StaticServiceStore
- RSOP

All these stores are in the registry except for the RSOP store, which holds the rules merged in from local and/or domain Group Policy Objects (GPOs). GPOs themselves can be considered stores, but they only exist to be read into memory into the RSOP store.

There are also two other special-purpose stores:

- ActiveStore
- SystemDefaults

The ActiveStore store exists only in memory. It is the live store of rules being used and enforced by the Windows Firewall and IPSec drivers. Windows reads rules from

multiple stores and merges them into the ActiveStore, hence, the ActiveStore is more like an in-memory cache of rules read from the "real" stores in the registry and in GPOs.

Officially, the ActiveStore is supposed be equal to PersistentStore + StaticServiceStore + ConfigurableServiceStore + RSOP, but in fact it only includes the PersistentStore + RSOP store. This has been a known issue or bug for many years.

The SystemDefaults store is in the registry and is only used for resetting firewall rules back to Microsoft's factory defaults. The default rules are located under HKLM\SYSTEM\CurrentControlSet\Services\SharedAccess\Defaults\FirewallPolicy\FirewallRules.

To list the rules in the ActiveStore and SystemDefaults stores:

```
Get-NetFirewallRule -PolicyStore ActiveStore

Get-NetFirewallRule -PolicyStore SystemDefaults
```

IPSec rules are also stored in the registry, they are local under HKLM\SYSTEM\CurrentControlSet\Services\SharedAccess\Parameters\FirewallPolicy\ConSecRules.

To list IPSec rules in the PersistentStore and RSOP stores:

```
Get-NetIPsecRule -PolicyStore PersistentStore

Get-NetIPsecRule -PolicyStore RSOP

Get-NetIPsecRule -PolicyStore ActiveStore #Persistent + RSOP
```

There are no IPSec rules by default in the ConfigurableServiceStore, StaticServiceStore or SystemDefaults stores. These three stores only contain firewall rules by default.

PersistentStore and RSOP Store

The rules visible in the Windows Firewall snap-in are composed of PersistentStore and RSOP rules that have been merged together. In the Windows Firewall snap-in, if you add the Rule Source column, it will show the whether the rule is a local setting or from a GPO.

To list the rules in the PersistentStore and RSOP stores:

```
Get-NetFirewallRule -PolicyStore PersistentStore

Get-NetFirewallRule -PolicyStore RSOP
```

When multiple GPOs contain firewall and/or IPSec rules, these rules are merged together following normal LSDOU processing of GPOs, then the final RSOP rules are merged on top of the PersistentStore rules; hence, an RSOP rule will take precedence over a

conflicting PersistentStore rule. There are also Group Policy settings to fine-tune how this is performed.

The PersistentStore is found in the registry under HKLM\SYSTEM\CurrentControlSet\Services\SharedAccess\Parameters\FirewallPolicy\FirewallRules.

StaticServiceStore
The StaticServiceStore store only contains firewall rules which have been added by Microsoft to secure the network traffic of common Windows services. For each protected service, this store will typically have rules to block all of that service's traffic by default except for the traffic specifically needed by the service.

To list the rules in the StaticServiceStore:

```
Get-NetFirewallRule -PolicyStore StaticServiceStore
```

This store was first added in Windows Vista and Server 2008 to help harden and protect built-in services from abuse (Microsoft refers to these rules as "Windows Service Hardening" rules). Other than directly editing the registry, there is no officially supported tool or API to edit the StaticServiceStore rules. These rules are hidden in that they do not appear in the graphical Windows Firewall snap-in, but they can be seen in the registry and listed with PowerShell.

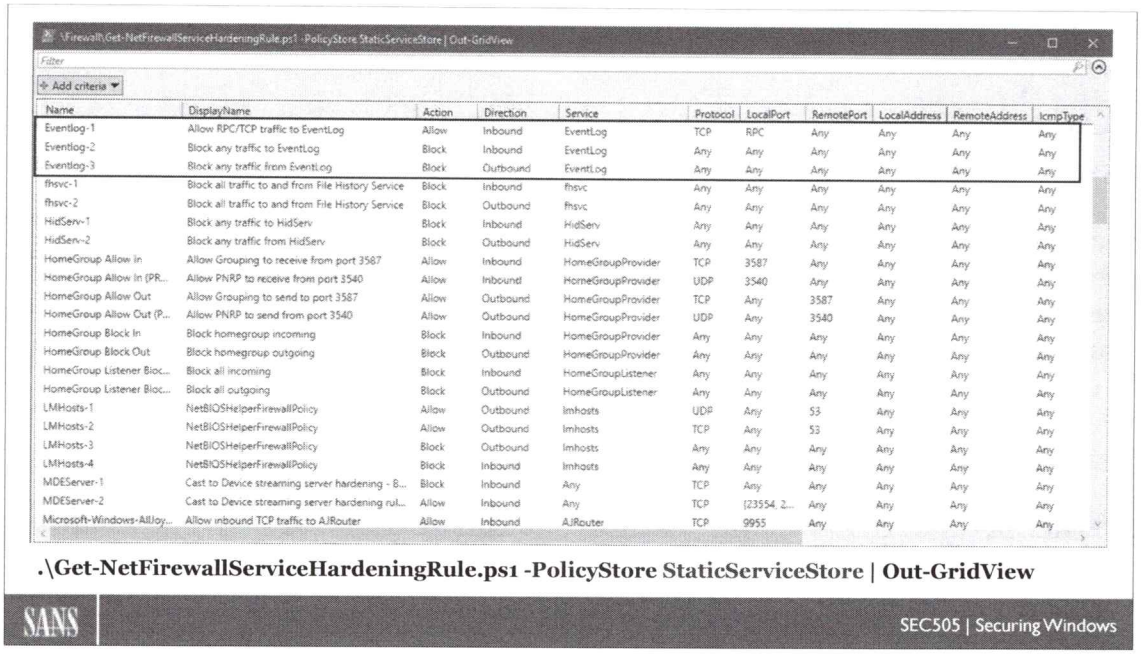
The StaticServiceStore is found in the registry under HKLM\SYSTEM\CurrentControlSet\Services\SharedAccess\Parameters\FirewallPolicy\RestrictedServices\Static\System.

ConfigurableServiceStore
The ConfigurableServiceStore store is for third-party services and applications, plus Microsoft's own Universal Apps for Windows 8 and later. Microsoft provides two scriptable COM objects (HNetCfg.FWRule and HNetCfg.FwPolicy2) so that third-party developers can manage their own ConfigurableServiceStore rules for their own products. Unfortunately, some products add rules that are bad for security. These rules are hidden in that they do not appear in the graphical Windows Firewall snap-in, but they can be seen in the registry, listed with PowerShell, and managed with the above COM objects.

To list the rules in the ConfigurableServiceStore:

```
Get-NetFirewallRule -PolicyStore ConfigurableServiceStore
```

The ConfigurableServiceStore is found in under HKLM\SYSTEM\CurrentControlSet\Services\SharedAccess\Parameters\FirewallPolicy\RestrictedServices\Configurable\System.



Get-NetFirewallServiceHardeningRule.ps1

To better visualize the hidden StaticServiceStore and ConfigurableServiceStore rules, run the following script from your courseware media (note the line wrapping below):

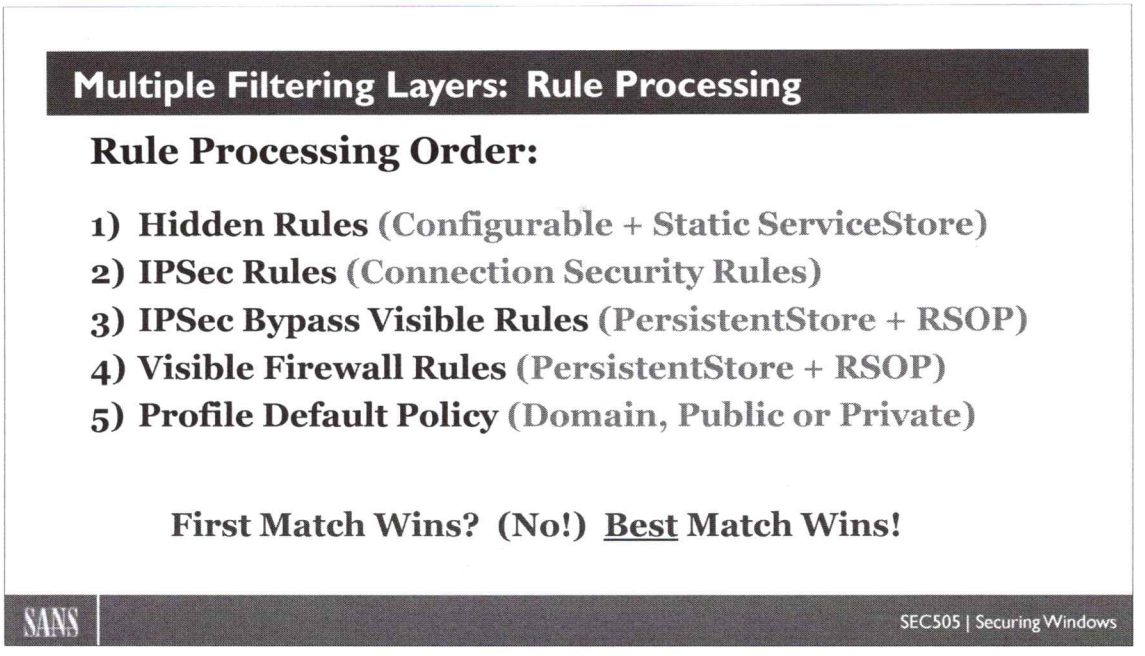
```
cd C:\SANS\Day5-IPSec\Firewall

.\Get-NetFirewallServiceHardeningRule.ps1 -PolicyStore
    StaticServiceStore | Out-GridView

.\Get-NetFirewallServiceHardeningRule.ps1 -PolicyStore
    ConfigurableServiceStore | Out-GridView
```

Using the Out-GridView cmdlet you can filter by keyword, sort columns by clicking on them, and filter by one or more property criteria (with the "Add Criteria" button).

Notice in the screenshot above how there are three rules for the EventLog service. Two of the rules block all inbound and outbound traffic for that service, and a third one allows inbound connections only to the one RPC port specifically used by the EventLog service.

Remember, if a hidden firewall rule allows an inbound connection, but a visible rule seen in the Windows Firewall snap-in blocks that same inbound connection, the connection is blocked.

It is important to remember the existence of these hidden "Windows Service Hardening rules" (as Microsoft calls them) when troubleshooting or investigating an incident.



Multiple Filtering Layers: Rule Processing

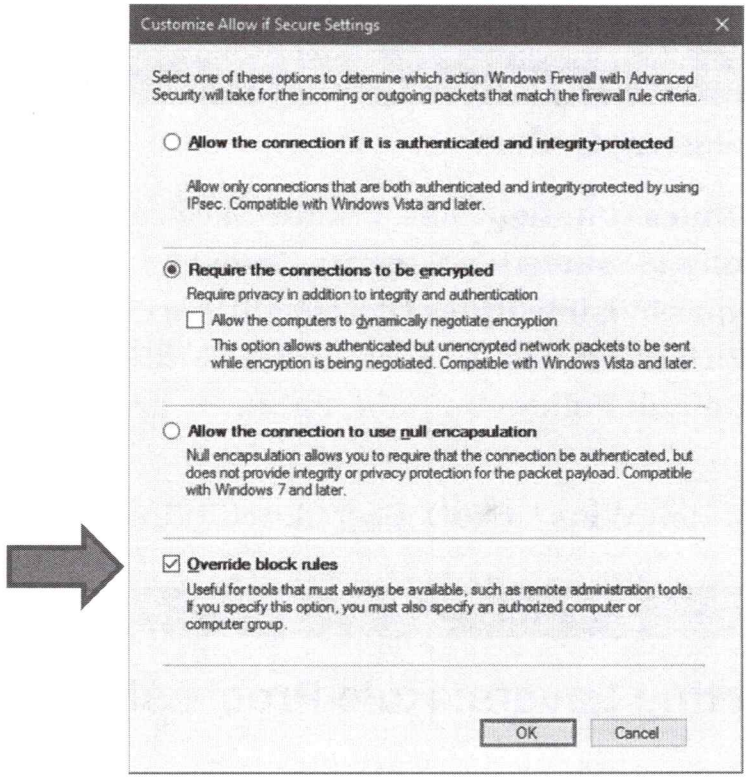
The Windows Firewall processes its rules in the following order:

1. Hidden Rules (ConfigurableServiceStore + StaticServiceStore)
2. IPSec Rules (Connection Security Rules)
3. IPSec Bypass Visible Firewall Rules (PersistentStore + RSOP)
4. Visible Firewall Rules (PersistentStore + RSOP)
5. Profile Default Policy (Domain, Public, Private)

The Hidden rules are composed of ConfigurableServiceStore and StaticServiceStore rules. These are not visible in the Windows Firewall snap-in. It appears that the StaticServiceStore rules are merged first, then followed by the ConfigurableServiceStore rules afterwards. This means that, if there is a conflict, the ConfigurableServiceStore rule will win. Hence, if a ConfigurableServiceStore rule allows an inbound connection, but a StaticServiceStore blocks it, the connection is allowed. If a ConfigurableServiceStore rule blocks an inbound connection, but a StaticServiceStore allows it, the connection is blocked.

IPSec rules are called "Connection Security Rules" in the Windows Firewall snap-in.

The IPSec Bypass Visible Firewall Rules are also called "Authenticated Bypass Rules", and these can be seen in the Windows Firewall snap-in too. Go to the properties of an allowing rule, General tab, select "Allow the connection if it is secure", click the Customize button, then check the "Override block rules" checkbox to make this rule an IPSec Bypass rule.



If this rule is an inbound rule, then at least one computer must be added on the Remote Computers tab of that rule too. Outbound rules with this box checked do not have any requirements for the Remote Computers tab. Rules that block packets cannot be configured as IPSec Bypass rules.

Windows Firewall Cannot Really Be Turned Off

Importantly, even if the Windows Firewall is turned off for all three profiles, the StaticServiceStore and ConfigurableServiceStore rules are still enforced!

In many ways, it's not really possible to turn off the Windows Firewall. Using the Windows Firewall snap-in to turn off the firewall for a profile type (Domain, Public, Private) merely deactivates the enforcement of PersistentStore and RSOP store rules for interfaces of that type, it doesn't actually disable all the filtering and IPSec activities by Windows whatsoever.

If you install a third-party firewall, it does not replace these rules, it merely adds to them. A third-party firewall is just another layer in the protocol stack.

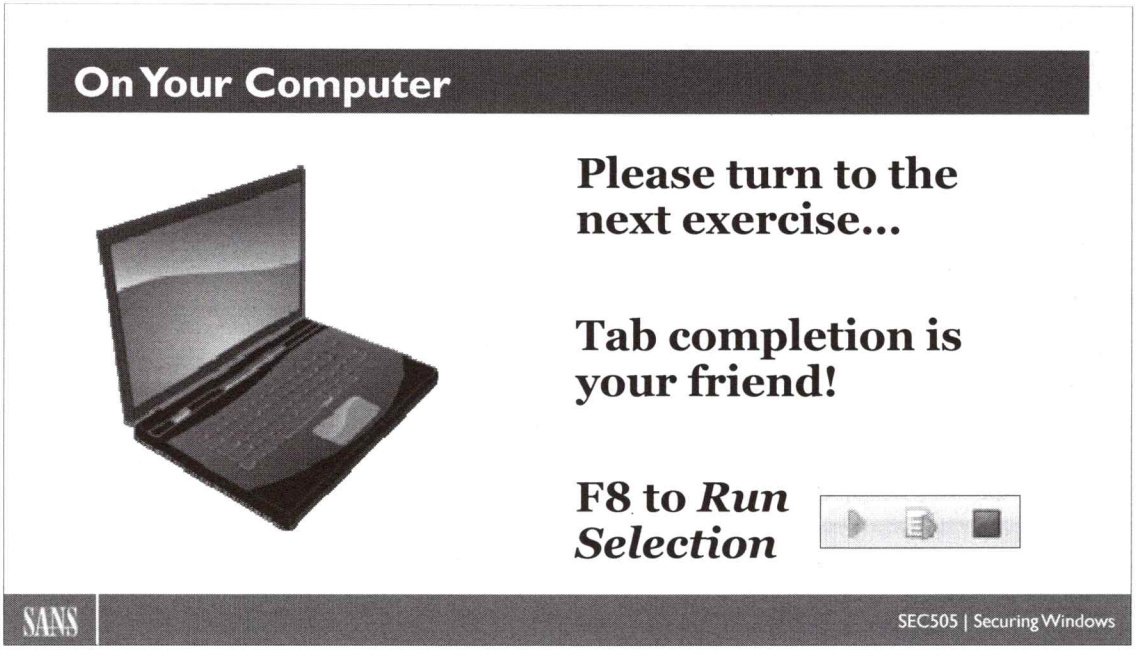
Best Match Wins (Not *First* Match Wins)

For both Hidden and Visible rules, when multiple rules apply to the same packet, the more specific rule is the winner, even if the less-specific rule is a blocking rule. It is not always the case that a blocking rule will win over an allowing rule. The winning rule is the more "specific", that is to say, includes more defining information to select packets. The profile's default policy is the least specific; a rule only mentioning source and

destination IP addresses is less specific than a rule which mentions the protocol; a rule only mentioning a protocol is less specific than a rule which also includes port numbers; and the most specific rule would be one that has defining information on every tab in the properties of that rule, such as IP addresses, protocol, ports, interface types, program or service information, etc. Unfortunately, Microsoft does not exactly document the algorithm by which rules are weighted for "specificity", but it'll be rare that two rules will be almost exactly identical such that we'd need to know.

Multiple Layers of Filtering

Keep in mind that each set of rules represents a distinct layer of filtering, each of which is capable of blocking a packet. Each layer must allow a packet through if the packet is to be successfully sent or received by the computer. It is not the case that any allowing rule can allow the packet in or out of the computer, there must be an allowing rule at each layer. Hence, if a StaticServiceStore rule and a ConfigurableServiceStore rule both allow an inbound connection, but a PersistentStore rule blocks it, the connection is still blocked. When managing visible rules in the Windows Firewall snap-in, if you have configured the visible rules to block a packet or connection, it will be blocked, no matter what the other rules may specify (just don't forget the IPSec-Authenticated Bypass rules, since these can override blocking rules).



On Your Computer

This exercise has multiple parts. Please don't forget to use tab completion!

Note: Many of these examples require Server 2012 R2, Windows 8.1, or later.

Display Networking Information

List network adapters:

```
Get-NetAdapter

Get-NetAdapter -Physical
```

List very detailed information about all physical and virtual network adapters:

```
Get-NetAdapter -Includehidden | Format-List *
```

List IPv4 and IPv6 addresses for all interfaces (replaces ipconfig.exe):

```
Get-NetIPAddress
```

List TCP connections and TCP listening ports (replaces netstat.exe):

```
Get-NetTcpConnection | Sort State
```

List UDP listening ports (replaces netstat.exe):

```
Get-NetUdpEndpoint -LocalPort 53
```

Resolve a DNS host name (replaces nslookup.exe):

```
Import-Module -Name DnsClient   #This is not usually necessary.

Resolve-DnsName -Name $env:computername -server localhost
```

View the local route table (replaces route.exe):

```
Get-Netroute
```

Ping a destination IP address or host name (replaces ping.exe):

```
Test-NetConnection -ComputerName $env:computername
```

Trace the IP route to a destination through a path of routers (replaces tracert.exe):

```
Test-NetConnection -ComputerName $env:computername -Traceroute
```

Test access to a TCP port and show the governing IPSec rule (like a port scanner):

```
Test-NetConnection -Computer $env:computername -Port 80 -Info
detailed
```

View the ARP cache (replaces arp.exe):

```
Get-NetNeighbor
```

Empty the net neighbor cache ("ARP cache") without being prompted to confirm:

```
Remove-NetNeighbor -Confirm:$False
```

Display Firewall Information

Display the connection profile (Public, Private, Domain) for each interface:

```
Get-NetConnectionProfile | Format-Table
InterfaceAlias,NetworkCategory
```

Display the state of the firewall for each interface profile type:

```
Get-NetFirewallProfile
```

> **Note:** The "persistent" policy store only contains firewall rules from the local computer (hence, they persist across reboots) while the "active" policy store includes both local rules and rules freshly applied by Group Policy, if any.

Display the names only of inbound, enabled, public, allowing firewall rules:

```
Get-NetFirewallRule -Enabled true -Direction inbound -Action
allow -Policystore ActiveStore | Where { $_.profile -match
'any|public' } | Format-Table DisplayName,Description -autosize
```

Display the names of the various firewall rule groups in a graphical application:

```
Get-NetFireWallrule | Select-Object -Property displaygroup,group
 -Unique | Sort-Object DisplayGroup | Out-GridView
```

> **Note:** The "displaygroup" is different for every culture, while the "group" field is independent of any culture, i.e., it works on every computer around the world.

Finished Already?

Do not run any of the following commands now, they are for reference only!

Review this example to configure an IP interface to use DHCP (do not run):

```
Set-NetIpInterface -InterfaceIndex 3 -Dhcp enabled

Set-DnsClientServerAddress -InterfaceIndex 3
  -resetserveraddresses

Restart-Service -Name dhcp -force
```

To change the settings on a network interface, you will usually first need to know either that interface's alias name or its index number ("ifIndex"). This information can be displayed by running "get-netadapter -includehidden".

You can also usually pipe an object representing an interface into the cmdlets which can configure interfaces. This way, you don't always have to obtain the interface index number or alias name first.

Review this example of permanently adding an IP address (do not run):

```
new-netipaddress -InterfaceIndex 3 -IpAddress 10.18.3.1
  -prefixlength 16 -defaultgateway 10.18.1.1
```

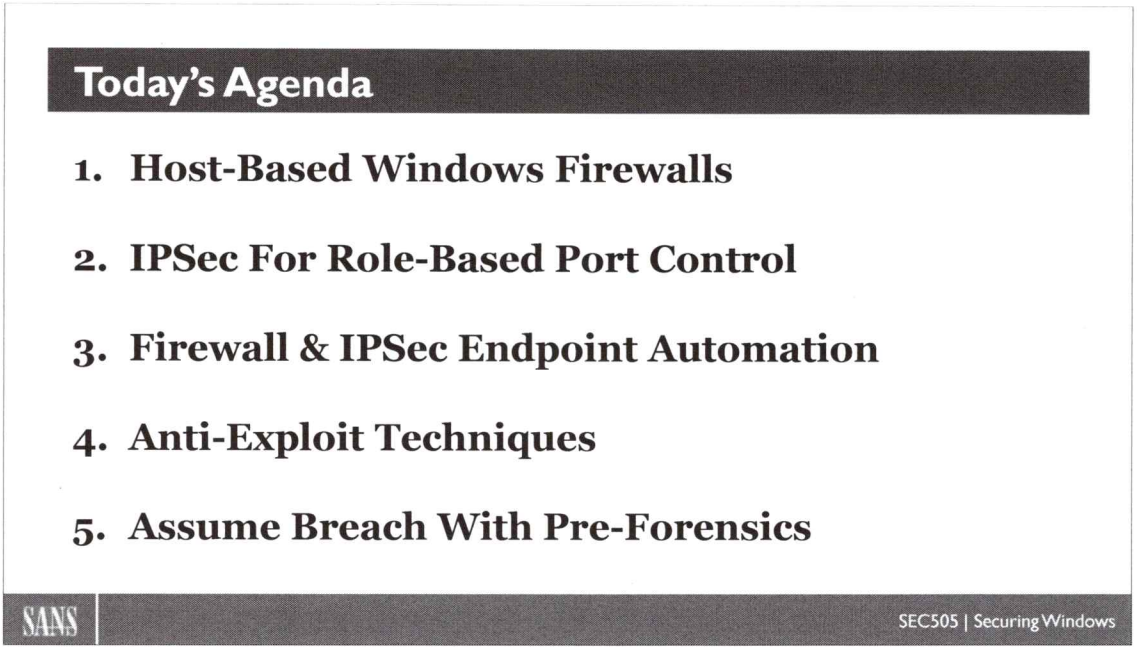
> **Note:** The prefix length is the number of 1-bits in the subnetmask, hence, a length of 16 is equivalent to 255.255.0.0 in the subnet mask.

Review this example of removing an IP address from all interfaces (do not run):

```
remove-netipaddress -IpAddress 10.18.3.1
```

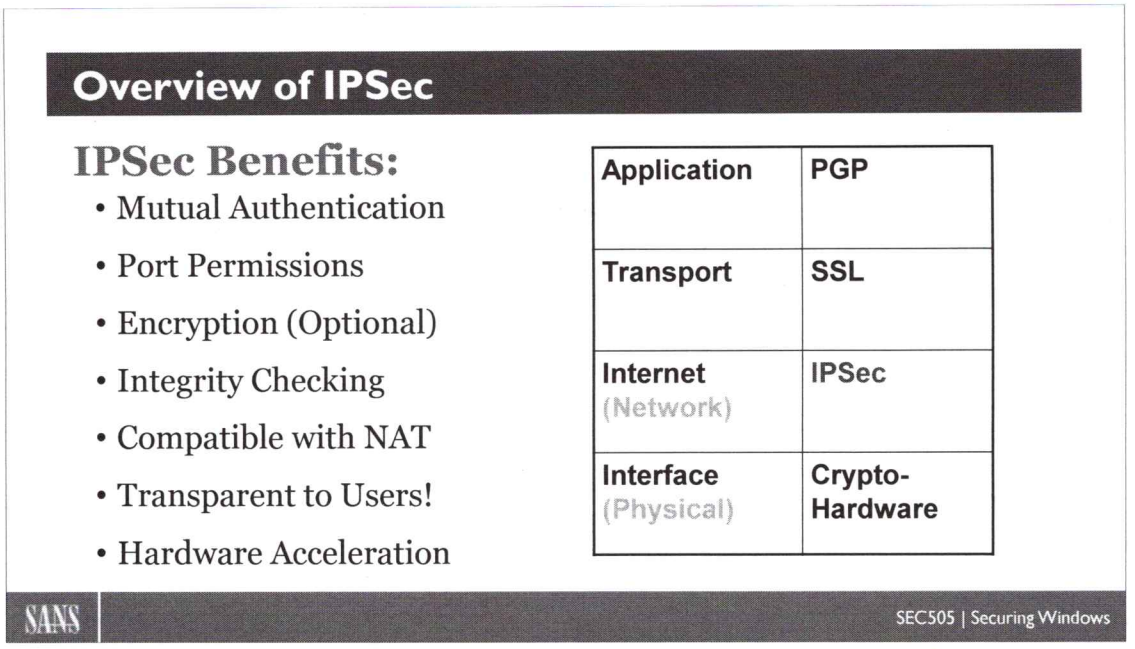
Review this example of assigning primary and secondary DNS servers to an interface which has an index number of 38 (do not run):

```
set-dnsclientserveraddress -InterfaceIndex 38
  -serveraddresses @("10.18.1.5","10.18.1.6")
```



Today's Agenda

Now that we're comfortable with the Windows Firewall, let's create an IPSec rule which can encrypt packets. When a firewall rule only allows packets in/out which have been secured with IPSec, the firewall rule doesn't automatically create the necessary IPSec rule to actually encrypt or sign those packets. Hence, the firewall rules and the IPSec rules work with each other, but they do not create or manage each other.



Overview of IPSec

Internet Protocol Security (IPSec) is a suite of protocols used for the authentication, integrity-checking, encryption, and encapsulation of TCP/IP packets.

IPSec security is implemented at the Network layer in the four-layer DoD protocol model associated with TCP/IP. This seemingly simple fact has drastic benefits!

Layer:	Contains:	Examples:
Application	Commands and data used by services and applications, e.g., HTTP, SMTP, FTP, NNTP, etc.	Pretty Good Privacy (PGP) encryption of files and e-mail
Transport	TCP and UDP port numbers, sequence and acknowledgement numbers, session control flags, etc.	Secure Sockets Layer (SSL), Transport Layer Security (TLS)
Internet (Network)	IP addresses, ICMP data, etc.	**IPSec**
Interface (Physical)	Ethernet and Token Ring hardware addresses, CRCs, etc.	Dedicated crypto-hardware devices.

An important point to understand is this: as authentication/encryption features are implemented at a lower and lower level in the protocol stack, these features become more *transparent* to users and more *compatible* with a wider range of applications and services.

Users do not have to be trained to use IPSec-compatible applications because all their applications are already IPSec-compatible, even if the original application developers have never even heard the word "IPSec" before. Services and daemons do not have to be replaced with IPSec-compatible versions because they are already compatible straight off

the CD-ROM. In short, if a piece of software sends IP packets over the wire, you can use IPSec to invisibly secure those packets. (*Including ping?* Yes, including ping packets.)

That is the real shortcoming of doing encryption/authentication at the Application and Transport layers. PGP, SSH and Stunnel have to be installed separately, for example, and users have to be trained and reminded to use them. TLS only works with applications and services specifically designed to support TLS, but at least it is more transparent than PGP because it is at the Transport layer instead of the Application layer. On the other hand, IPSec is invisible to users, does not have to be installed (it's installed already), and is compatible with all applications and services that communicates via IP.

As the table above shows, only hardware-based cryptographic devices provide security at a lower level than IPSec. But, in this case, you have to purchase special crypto-hardware! IPSec is compatible with any off-the-shelf hardware which is capable of using IP, including Ethernet, token ring, FDDI, wireless, modems, serial lines and infrared. In sum, IPSec is both independent of the underlying hardware and invisible to the upper-level protocols and applications above it -- *that* is why IPSec has become the standard.

Threats

IPSec is needed because the standard Internet Protocols --IP, ICMP, UDP, TCP, etc.-- do not provide security for themselves. In fact, these protocols are inherently insecure and archaic. They were never designed for security in the first place. They are wonderfully-designed fossils of the DARPA-net era which, through the accident of technological evolution, have been pressed into service to make an information economy.

IPSec can help to prevent attackers from causing harm when attackers try to:

- Use a port scanner to discover active TCP and UDP ports (reconnaissance).
- Spoof source IP addresses (denial-of-service and other attacks).
- Capture, modify and resend packets (replay attacks).
- Impersonate hosts (man-in-the-middle attacks).
- Extract confidential information from captured packets (attacks against privacy).
- Connect to any open TCP port if the computer is directly accessible from the Internet, even if you wish some ports were only available to your LAN users.

Standardized

IPSec standards are defined by Internet Engineering Task Force (IETF) Requests for Comments (RFCs). IPSec itself is not owned by any one vendor, and certainly not owned by Microsoft.

IPSec has become the *de facto* standard for securing TCP/IP traffic and doing Virtual Private Networking over the Internet.

There are many RFCs related to IPSec; the most important ones are at the front of this courseware. The following are the core RFCs and should be read in this order:

- RFC 2401: Security Architecture for the Internet Protocol
- RFC 2402: IP Authentication Header (AH)
- RFC 2406: IP Encapsulating Security Payload (ESP)
- RFC 2409: Internet Key Exchange (IKE)
- RFC 4306: Internet Key Exchange (IKEv2)
- RFC 4555: IKEv2 Mobility and Multihoming Protocol (MOBIKE)
- RFC 3948: Proposed: UDP Encapsulation of IPsec ESP Packets

Benefits of IPSec

There are numerous benefits of IPSec for network security:

- **Mutual Authentication Required**. Unlike SSL, both IPSec peers must authenticate to each first before an IPSec session can be established between them. Windows supports Kerberos, certificate, and pre-shared key authentication methods for IPSec. Kerberos is enabled automatically when a computer joins the domain, an Enterprise CA can distribute certificates automatically through Group Policy auto-enrollment, and pre-shared keys can be configured through Group Policy as well.

- **Strong AES Encryption**. The payload of a packet (or the entire packet itself when tunneled) can be strongly encrypted for privacy. Windows supports both AES 128/192/256-bit and the older 3DES 168-bit in Cipher Block Chaining (CBC) mode for IPSec data encryption.

- **Integrity-Checking**. Packets can be checked to verify that they have not been accidentally damaged or deliberately modified during transit using hashing algorithms, not just CRC checksums.

- **Transparent to Applications, Services and iSCSI**. Because security is implemented at the Network layer, IPSec is transparent to applications, services and even iSCSI. Applications and services do not have to be upgraded or patched to make them IPSec-compatible. Legacy applications can be used and they too will benefit from IPSec because they will be completely unaware of IPSec's operation or existence. You do not have to look for an "*IPSec Inside!*" sticker on your new software to verify that it will work with IPSec. IPSec works with all IP packet types except for broadcast and multicast packets. When you must transmit iSCSI packets over shared insecure links, IPSec can be used to mutually authenticate the initiator and target server, as well as encrypt all file transfer and CHAP authentication traffic.

- **No User Training Required**. Users do not have to be trained to use IPSec. In fact, users don't even need to be aware that IPSec has been enabled on their computers. This is one of the most important benefits of IPSec. Without this "user transparency" IPSec would be undeployable on desktops because of its complexity.

- **Group Policy Management**. Users do not have to be trained because IPSec can be centrally managed through Group Policy. Each OU or site could have a different set of IPSec policies applied to the computers in it. This feature is what enables IPSec on Windows to relatively easily scale out to thousands of machines.

- **Remote Command-Line Management**. The IPSec driver is 100% manageable from the command line with PowerShell, IPSECPOL.EXE (2000), IPSECCMD.EXE (XP) or NETSH.EXE (2003). These tools work against both local and remote systems.

- **Windows Firewall Integration**. The XP-SP2/2003-SP1 Windows Firewall has a Group Policy management option named "Allow Authenticated IPSec Bypass" which permits unsolicited in-bound connections if they are secured with IPSec. You can limit which remote computers are permitted through the firewall by defining a custom group of computer accounts and only allowing the IPSec-bypass feature for that group, e.g., for remote management and backup computers. When an IPSec policy is assigned, the Windows Firewall automatically allows incoming UDP 500 and 4500 packets for the sake of IPSec session establishment.

- **AES CPU Acceleration.** IPSec encryption on Intel or AMD processors built in Q4'2010 or later will benefit from the AES-NI instruction set designed into these processors for AES hardware acceleration. This also benefits TLS and BitLocker.

- **IPSec Hardware Acceleration NICs**. IPSec cryptographic operations can be off-loaded to smart network adapter cards. These IPSec-enabled network cards possess cryptographic processors to perform the CPU-intensive work of authenticating, integrity-checking, and encrypting packets on behalf of the operating system. Example IPSec off-load cards are the Intel Gigabit ET (1Gbs copper) and Intel Ethernet Server Adapter X520 (10Gbs fiber). (See KB254257 concerning the Intel cards and fine-tuning AH+ESP mode.) Also, some SSL accelerator cards can assist with main mode negotiations as well if the cards support CryptoAPI. Each IPSec security association consumes about 5KB of memory and a single high-end server can support over ten thousand associations. A dataflow using IPSec ESP requires about 1-3% more bandwidth than the same transfer in cleartext.

- **User Rights Integration (Windows Server 2003 and later)**. When using either Kerberos or certificate-to-computer-account-mapping authentication, Windows Server 2003 will enforce the "Access This Computer From The Network" and "Deny Access To This Computer From The Network" user rights against remote computers (2000 or later) when they initiate in-bound IPSec connections to it. Hence, you can limit by group membership which computers are permitted to open in-bound IPSec connections and then block all non-IPSec connections to the port or service you are trying to secure.

- **Firewall, NAT and IDS Compatibility**. When used by servers which must communicate through a firewall, e.g., webservers and database servers, IPSec simplifies the firewall design because the vagaries of the protocols used do not have to be predicted and managed. The servers can additionally be required to communicate over IPSec, thus providing defense in depth beyond the security provided by the firewall. Using either AH, ESP with encryption disabled, or null encapsulation will reduce the overhead of using IPSec because of the lack of encryption processing, and these cleartext packets can be examined by network Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS). If the IDS/IPS is confused by the AH or ESP header, IPSec null encapsulation means that no IPSec headers are added to packets at all after the initial IKE negotiation. Finally, the NAT-T extension permits IPSec sessions through firewalls, proxy servers and other devices that perform Network Address Translation (NAT), but only if the devices performing NAT are well-behaved (see RFC 3948).

- **Interoperable**. Because IPSec is an IETF standard, it is not owned by any single vendor. Hence, different vendors can create interoperable IPSec/VPN products. For example, PGP Desktop Security 7.0.3 and later can use PGP's IPSec client (PGPnet) to connect from a non-Windows client to a Windows server over IPSec (www.pgp.com). Also, there are a variety of Cisco Systems products which interoperate with Windows IPSec; this should not be too surprising since Cisco helped Microsoft write their IPSec code (www.cisco.com). And note that the Windows PKI can be configured to support Cisco's Simple Certificate Enrollment Protocol (SCEP); to do this, see the help for the CEPSETUP.EXE utility from the Windows Server *Resource Kit* CD-ROM (\apps\cep folder). Finally, the FreeS/WAN IPSec client for Linux is mostly interoperable with Windows (www.freeswan.org) and so is OpenBSD's `isakmpd` (www.openbsd.org).

- **Extensible**. IPSec was designed to be extensible and flexible. For example, as new ciphers and key lengths become desired, IPSec can be extended to support them while maintaining backwards compatibility.

- **IPv6 Support**. IPSec was designed with IPv6 in mind. The version of IP in widespread use today is version 4. IPv6 is the next generation of IP, with 128-bit addresses, extensible headers, and a number of new features for a heavily networked world. Windows supports IPv6 natively as well.

- **Virtual Private Networking Support**. Virtual Private Networking is the ability of a remote user or router on the Internet to connect to the corporate LAN through an encrypted secure channel. This encrypted channel will encapsulate or "tunnel" entire packets during transit. VPNs reduce long-distance charges while providing very secure remote access. While VPNs can be implemented with other protocols besides IPSec (for example, PPTP does not use IPSec) IPSec has become the industry-standard solution for VPNs. The preferred VPN protocol on Windows is L2TP, and L2TP uses IPSec for its security.

- **Network Load Balancing (NLB)**. IPSec and NLB are compatible and can be used together to create a load-balanced and fault-tolerant farm of VPN gateways (maximum of 32 gateways in the farm). With DNS round robin, multiple farms can be used for even greater scalability. For configuration steps, see KB820752 and KB323437, as well as the help menu in the Network Load Balancing Manager in the Administrative Tools folder.

Drawbacks to Using IPSec

There are a few important drawbacks to using IPSec on Windows:

- While the IPSec implementation on Windows supports NAT-T for Network Address Translation (NAT) compatibility, that does not mean all the other networking devices through which one's IPSec connections are routed will be well-behaved or configured correctly (see RFC 3948). This is especially a problem when both IPSec peers are each behind their own separate firewalls that perform NAT (this scenario is called "double NAT"). There is a registry value named "AssumeUDPEncapsulationContextOnSendRule" that may be set to help deal with some double-NAT problems, but this can only change the behavior of the two Windows IPSec peers, not the firewalls or routers along the network path between them. Double NAT is a common problem when mobile devices are accessing VMs hosted by cloud providers. Often, the only solution to the problem is to either 1) establish a non-IPSec VPN tunnel between the two networks and use IPSec within the VPN, or 2) move one of the IPSec peers out from behind its NAT-ing device and give that peer a public IP address. Once IPv6 is standard, NAT should finally go away forever.

- A Windows IPSec/VPN will not have the same throughput as a hardware-only IPSec/VPN solution, even if IPSec-enabled network adapter cards are used. If your organization is not already using Windows, or if you must support a very large number of simultaneous VPN clients, then a hardware solution may be more cost-effective.

- Windows only supports enough of IPSec tunnel mode for bare RFC compliance, and Microsoft doesn't really want you to use tunnel mode for anything other than gateway-to-gateway connections either (KB252735). There are valid reasons for this (discussed later) but the tunnel mode limitations in Windows can be disappointing.

- With the possible exception of Cisco devices, Microsoft is not particularly concerned about interoperability with non-Windows IPSec peers. Some degree of interoperability is currently possible with various flavors of UNIX/Linux, but the next Service Pack could easily change all this. The only way you can know if it'll work is to set it up and see what happens.

- Microsoft stores "pre-shared keys" for IPSec authentication in cleartext in the registry. If the key can be read by an adversary, the adversary can open his or her

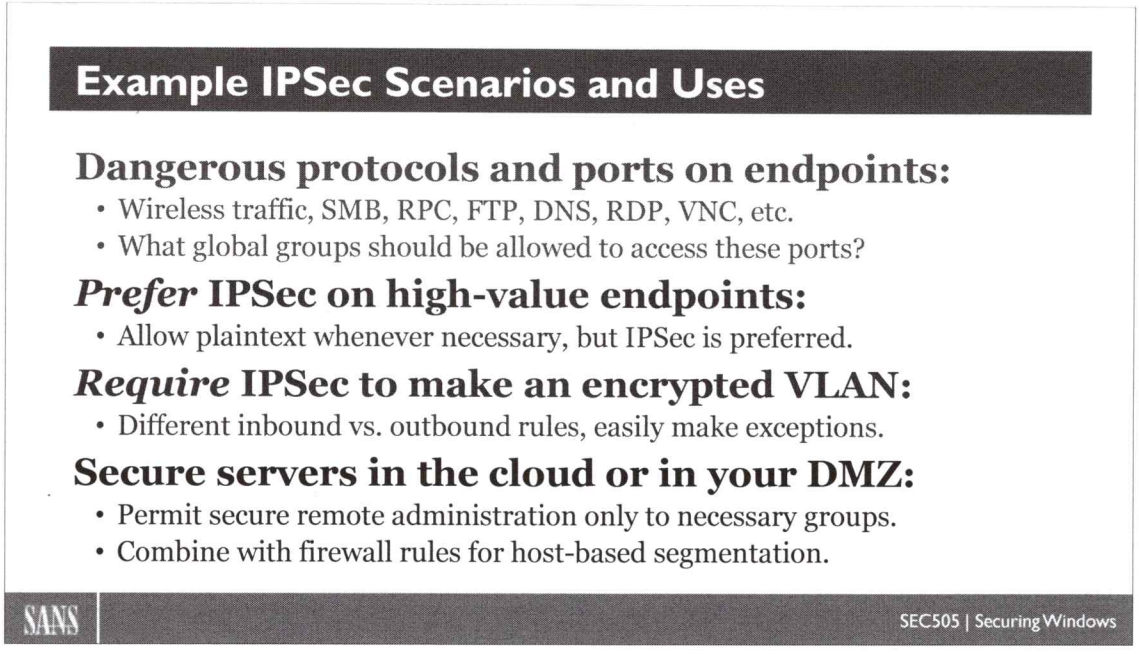
own IPSec sessions with one's systems. However, capture of this key does not enable an adversary to decrypt anyone else's sessions because the pre-shared key is only used for authentication, not data encryption. When using Group Policy to manage IPSec, any pre-shared keys will be in plaintext in the GPOs as well, which adversaries could sniff or extract from the SYSVOL share.

- Transparency to application-layer software is not always ideal. Unlike SSL/TLS, the details of an IPSec session are usually not (easily) available to applications if those applications wish to confirm the existence or details of the IPSec session, such as cipher suite or peer credentials, before proceeding with critical actions. Services and client applications are at the mercy of the operating system and the administrators who manage it.

- Like other implementations of IPSec, flooding of UDP port 500 or 4500 can prevent a victim host from establishing new IPSec sessions with other hosts. UDP 500 and 4500 are the ports used for IPSec Internet Key Exchange (IKE) negotiations, hence, they cannot be completely hidden or protected. Beyond flooding, there may also be exploitable vulnerabilities accessible through these ports, hence, it is important to quickly apply new IPSec-related patches.

- It is sometimes necessary to reduce the Maximum Transfer Unit (MTU) size of one's packets to avoid fragmentation delays and maintain compatibility with the networking infrastructure through which one's IPSec connections flow. Using PowerShell or NETSH.EXE, you might need to reduce an adapter's MTU, for example, from 1500 bytes to 1350 bytes. This problem is becoming rare.

- In rare cases, with some network adapters, especially when using UDP port 4500 encapsulation and dealing with double-NAT issues, it may be necessary to use the Device Manager tool to edit the properties of the adapter to disable "Checksum Offload" for IPv4, IPv6, UDP and TCP. There imposes a tiny performance penalty on the mainboard CPU.



Example IPSec Scenarios and Uses

IPSec is a powerful and versatile protocol. A few examples of its uses can illustrate.

Making "Dangerous" Ports, Protocols and Tools Safer on Endpoints

There are many tools and services one would like to use --especially for remote administration-- but cannot because of their security weaknesses, e.g., they transmit cleartext passwords. But if servers were configured to require an authenticated IPSec session before allowing connections to these dangerous ports, and if 3DES encryption were required for all the traffic, then these dangerous tools and services might be made secure enough to use, perhaps even on bastion hosts in the firewall's DMZ. The following is a sampling of the applications many administrators would like to use, or services they would like to enable on their servers, but cannot because of security worries, yet IPSec might make them secure enough:

- FTP
- TELNET
- SNMP
- SYSLOG
- RADIUS and LDAP authentication
- RPC-based applications (like most MMC snap-ins)
- Microsoft File and Print Sharing (SMB/CIFS)
- Remote Desktop Services (remote administration mode)
- VNC Remote Control (www.realvnc.com)
- PSEXEC.EXE -U (otherwise sends passwords in cleartext)
- Symantec pcAnywhere (http://www.symantec.com/pcanywhere/)

- And the list goes on for all those protocols, services and tools which would have been nice to run on servers in the DMZ, but were too dangerous even when protected by the firewall.

Wireless Networking

Wireless networks can be securely bound together with IPSec. Authentication will limit communication channels to just those systems participating in the domain or PKI, and encryption can provide privacy. And you don't have to require IPSec for all wireless communications either; you could rely upon the security enhancements of WPA2 for the most part, then require IPSec in addition whenever connecting to critical servers through the wireless link. If you do wish to use IPSec for all wireless traffic, install a wireless card that can run in Access Point mode in a Windows router, then configure the clients to use IPSec tunnel mode to forward all packets destined to the LAN to the wireless router instead. The Windows box will decrypt the IPSec packets and route/bridge the original packets onto the LAN. If your hardware Access Point vendor's box supports IPSec natively, then all the better! (And if one needs to push out IPSec digital certificates, one might as well push out certificates for 802.1X EAP-TLS authentication too.)

Servers in The Cloud or DMZ

Many websites are composed of a farm of webservers which act as front-ends to middleware servers which themselves communicate with multiple back-end databases (in a "three-tier" design). IPSec could be used to authenticate and integrity-check (but not encrypt) all communications among the servers and databases, then block everything else from the Internet and DMZ. A farm of webservers is also often a part of an isolated domain in the DMZ to provide for administration and content management. The domain controllers for this domain would be on a separate DMZ segment, and IPSec would be used to armor the communications between DCs and the webservers. Authentication protects the vital communications links between these servers, but without the overhead of encryption. Encryption isn't needed here because the threat of packet-sniffing is relatively small.

Critical Data Flows and Sensitive Machines (Preferring IPSec)

Defense "in depth" means providing multiple redundant layers of security. At many sites, if an attacker can penetrate the firewall, there are no other barriers between the attacker and critical internal servers. Exclusive reliance upon the firewall creates a single point of failure. IPSec can help to provide in-depth security for sensitive internal servers and workstations; for example:

- Domain controllers could replicate using IPSec,
- Databases could synchronize securely over IPSec,
- The workstations of the security administrators could require IPSec for all communications because the boxes they manage would too as well,
- The fileserver with the R&D source code might always require IPSec,
- The computers of a certain OU might be isolated from all other machines by their secret authentication keys and packet filtering rules,

- The computer OUs for corporate executives, HR and legal staff might always attempt to negotiate IPSec encryption, but fall back down to cleartext when required,
- And, in general, IPSec can help to defend a network against its own "trusted insiders" who are, in fact, according to the FBI, behind the majority of network security breaches which cause measurable financial harm.

For example, at the time of this writing, Microsoft's corporate network comprises 18 domains in six forests with cross-forest trusts and over 200,000 managed and unmanaged hosts. Yet approximately 70% of the internal traffic at Microsoft uses IPSec ESP (with no encryption enabled) to help protect the managed boxes from the unmanaged ones. (ESP is used instead of AH since ESP supports NAT-T and AH doesn't.) ESP with encryption is enabled on an as-needed basis through Group Policy on critical servers.

Very importantly, notice that in many of the examples above that IPSec is not required from the other computer, IPSec is merely preferred. This means a system can be configured to attempt to negotiate IPSec whenever a new inbound or outbound connection is being established, but, if the other computer cannot or will not authenticate with IPSec, the system will fall back to unsecured communications automatically, thus permitting the connection like normal. Hence, you do not always have to require IPSec when configuring a system, you can configure that system to merely prefer IPSec but be willing to talk to non-IPSec-capable machines as necessary. And you can either require or prefer IPSec on a case-by-case basis depending on the IP addresses, protocols or ports being used.

Getting Aggressive with Domain Isolation (Requiring IPSec)

Imagine if all your workstations and servers (or a large subset of them) required IPSec mutual authentication prior to any communication whatsoever. This would be similar to a VLAN, but instead of using switches, you're using IPSec. Only domain-joined computers would have the necessary kerberos tickets or certificates to be able to authenticate to other domain members, and all (most) systems would be configured through Group Policy to require mutual IPSec authentication. If an unauthorized computer attempted to open a TCP or UDP connection with a domain member, the connection would fail because of the inability of the rogue computer to authenticate with IPSec first.

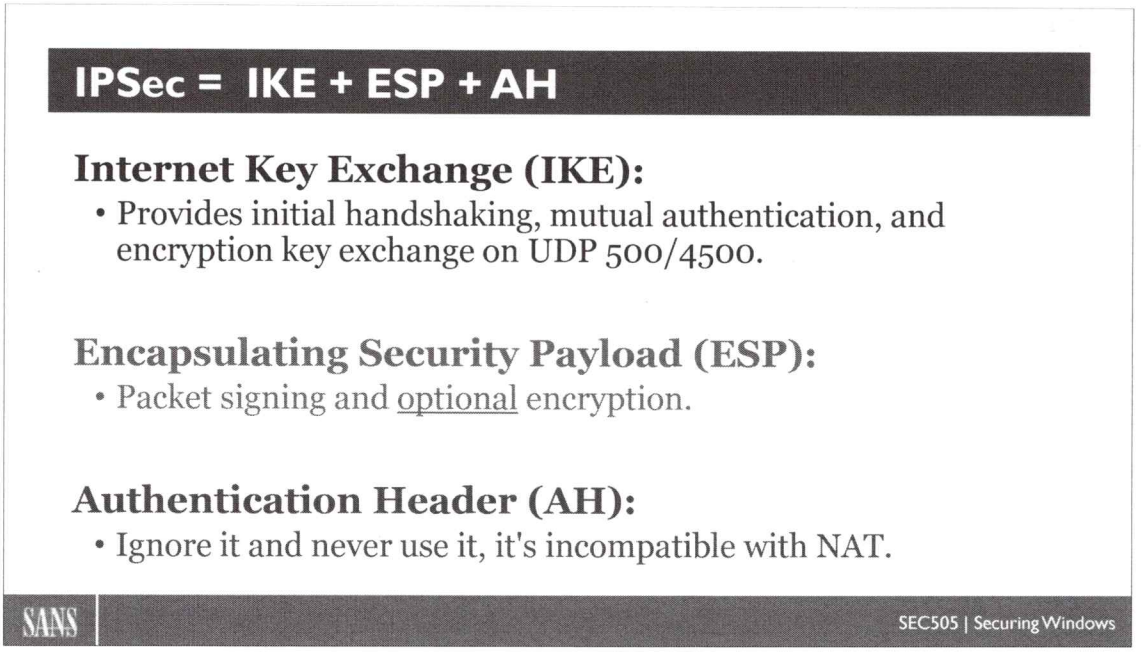
Partner Networks (Outside Your Active Directory)

You may have SMTP relays, websites, file servers, etc. to which you wish to give a partner network limited access. However, the partner company is not a part of one's forest and/or does not use Windows systems. Windows IPSec connections can be authenticated with digital certificates. These certificates can be distributed to hosts in other forests or to non-Windows systems. In short, the use of IPSec is not restricted to those within one's own organization or Active Directory forest.

Virtual Private Networking

When roaming users need to gain access to the LAN while on the road, IPSec can be used with L2TP to provide encrypted and authenticated client-to-router VPN tunnels. The client can now communicate with other hosts on the LAN just as though he or she were physically connected.

If all the users in a branch office need access to the main office, the two offices can be connected over the Internet with router-to-router IPSec. The routers perform all the encryption and authentication transparently for the users. The solution is secure, relatively easy to set up, and saves on long-distance leased lines. From the perspective of the two LANs, there just appears to be another "segment" connecting them. But that "segment" is actually an encrypted VPN tunnel through the Internet.



IPSec = IKE + ESP + AH

IPSec is a suite of protocols. The three main protocols are Internet Key Exchange (IKE), IP Authentication Header (AH) and IP Encapsulating Security Payload (ESP).

The following table summarizes the purposes of IPSec's core protocols.

Protocol	Purpose
IKE	Securely negotiate IPSec communication parameters and cryptographic keys between peers. Parameters include authentication methods, ciphers, key lengths, sequence numbers, time-to-live counters, IPSec session ID numbers, modes, etc. IKE is also what handles the mutual authentication of peers.
AH	Provides authentication of packet source, data integrity, and protection against replay attacks. Does **not** provide encryption. It's also incompatible with Network Address Translation (NAT).
ESP	Provides data encryption, authentication of packet source, data integrity, protection against replay attacks, and is compatible with Network Address Translation (NAT).

AH Signatures Include the IP Header

The most important distinction between AH and ESP is that ESP provides data encryption and AH does not. You must use ESP when you want privacy. However, it is possible to disable the encryption with ESP if desired.

AH also authenticates the *IP layer* of the packet and higher, while ESP only authenticates the *transport layer* and higher. This means that changes to the IP layer of the packet are detected by AH but not by ESP.

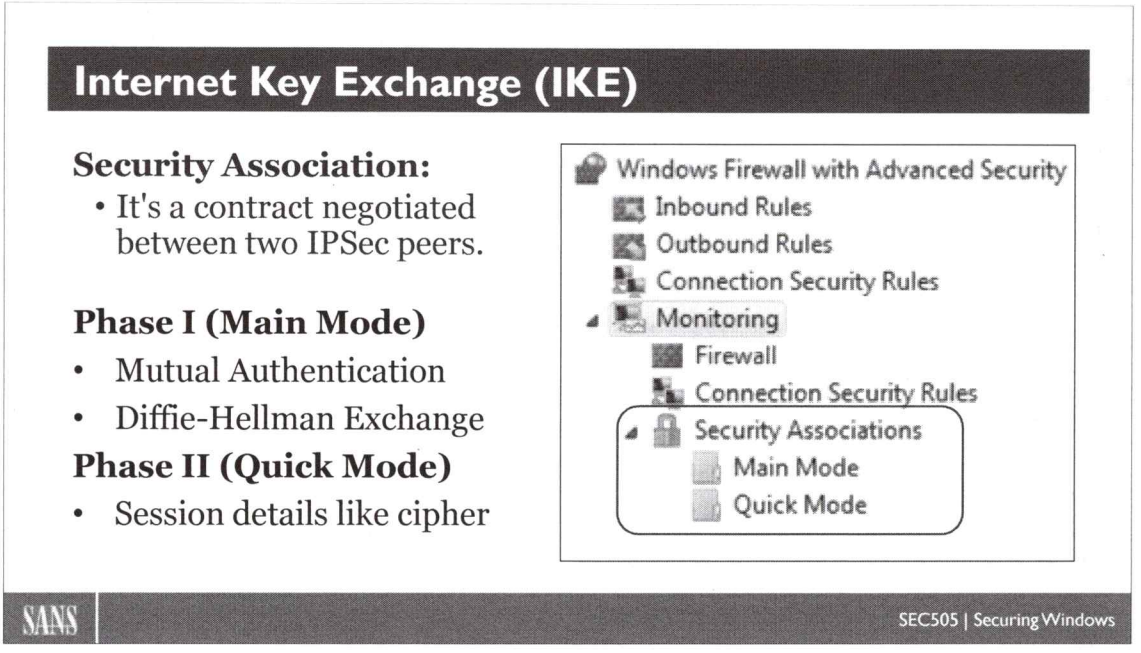
Keep in mind, though, that both AH and ESP can be used simultaneously. Hence, a single packet can use ESP to encrypt its transport layer and higher, while using AH to authenticate and integrity-check the entire packet (except the physical layer of course). Using ESP and AH simultaneously is very rarely done.

Just Never Use AH (Only Use ESP)

Very importantly, ESP can traverse devices performing Network Address Translation (NAT) without errors, while AH is always incompatible with NAT.

Because ESP can sign plaintext packets without necessarily encrypting them, and because ESP is compatible with NAT, just never use AH. If you want to sign plaintext packets, use ESP and turn off the encryption.

It's possible that the IETF will deprecate the entire AH protocol.



Internet Key Exchange (IKE)

IKE (RFC 2409) is a general-purpose protocol for negotiating communication parameters and cryptographic keys on behalf of other protocols like IPSec, RIPv2, OSPF, etc.. IKE negotiation is the first thing that occurs between two peers when they wish to communicate over IPSec. IKE is how two "**IPSec peers**" --IPSec-enabled hosts, servers or routers-- agree on how to secure their communications before secure communications can begin.

For each protocol that will use IKE negotiation, a document called a "**Domain of Interpretation (DOI)**" will define what IKE will negotiate for the protocol and how IKE will negotiate it. RFC 2407 is IPSec's DOI document for IKE.

Security Association (SA)

The end result of an IKE negotiation for IPSec, as defined in IPSec's DOI, is a data structure called a "**Security Association (SA)**". A SA is a set of parameters and cryptographic keys used by a peer to manage an IPSec session with another IPSec peer.

Every bi-directional IPSec session a host/router is party to will correspond to two SAs on that host/router: one for securing out-going packets to the other peer, and one for securing in-coming packets from that other peer. An SA is one-way, so two are required on each peer for bi-directional communications.

An SA is what maintains the context or "state" of an IPSec session. The information and keys in an SA enable the IPSec driver on a peer to successfully encrypt, decrypt, authenticate, and verify the integrity of IPSec packets.

An IPSec-enabled host or router will keep all of its SAs in its own "**Security Association Database (SADB)**". Each SA in a peer's SADB database is identified by a unique number called a "**Security Parameters Index (SPI)**". One SPI for each SA in the SADB. *(Ouch!* Now the acronym soup begins to get thick! But I promise I will only mention the important ones.)

Communicating peers will include the SPI numbers of the SAs controlling their IPSec communications in the headers of the packets they exchange. This fact is important for understanding how IPSec operates.

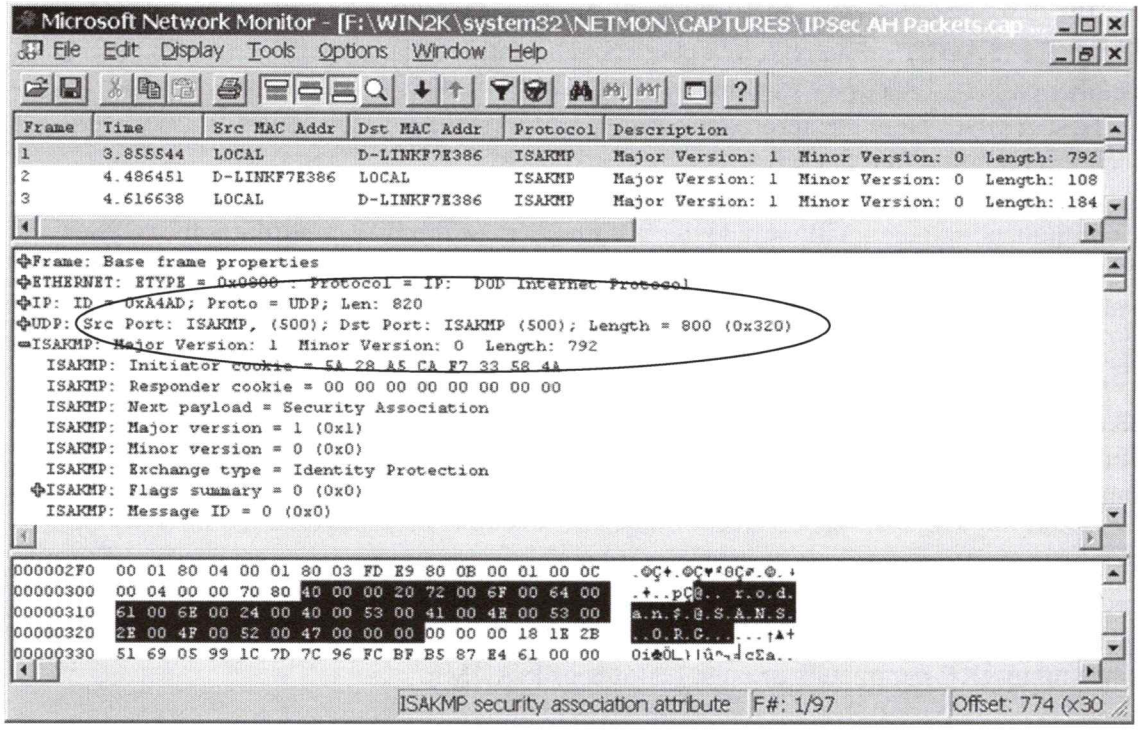
ISAKMP

IKE negotiations are carried out using a more abstract negotiation language called the **Internet Security Association and Key Management Protocol (ISAKMP)**. IKE is an implementation of ISAKMP, i.e., IKE *speaks* a dialect of ISAKMP.

IKE is less abstract/general than ISAKMP, but IKE can still be used to negotiate SAs for protocols other than IPSec; a DOI document is used to lay out the details of how IKE negotiates for other protocols, hence, a DOI will also specify some ISAKMP parameters, since many IKE parameters are just features of ISAKMP.

This can be confusing, but just keep in mind that ISAKMP is the more abstract or general-purpose, while IKE is an implementation of it. IKE-for-IPSec is an implementation of generic IKE. This course only concerns IKE-for-IPSec. The purpose of ISAKMP-IKE, in any case, is just to negotiate cryptographic keys and session parameters between the peers.

ISAKMP-IKE negotiations occur over UDP port 500 (but if NAT-T has been engaged, ISAKMP-IKE switches over to UDP 4500).



ISAKMP-IKE Negotiation Phases

ISAKMP-IKE negotiations occur in two Phases: Phase I and Phase II. The end result of each Phase is a separate SA, hence, we make a distinction between Phase I SAs and Phase II SAs. An IPSec session with a remote peer will require two SAs on each peer.

- **Phase I** negotiates an SA for ISAKMP-IKE itself (a "Main Mode IKE SA").
- **Phase II** negotiates an SA for IPSec proper (a "Quick Mode IPSec SA").

The SA negotiated in Phase I is separate from and prior to the IPSec SA produced during Phase II. Importantly, the IPSec SA from Phase II relies upon the SA negotiated in Phase I for its security. The real important key exchange occurs in Phase I for the ISAKMP-IKE SA. The IPSec SAs negotiated in Phase II derive their security from the Phase I SA.

> **Note**: Strictly speaking, both Phases are carried out by IKE, but in different "modes" (see below).

Phase I Negotiation Steps

In outline, the steps of a Phase I negotiation are:

1. **Negotiate policy**: cipher (DES, 3DES), hash (MD5, SHA1), authentication method (Kerberos, certificate, preshared key), and Diffie-Hellman "group". This is for the authentication phase only, not the bulk encryption of user data.

2. **Perform a Diffie-Hellman-Merkle key exchange**: the DH technique is used to derive an identical secret "master key" on each peer using the group information negotiated.
3. **Authenticate peer**: The master key from the prior step is used to HMAC and encrypt an authentication sequence, whose details were just negotiated.
4. **Create the ISAKMP-IKE SA**: with the above credentials and master key, the end-product we want is a set of Phase I SA parameters.

Two IPSec peers will first create a single ISAKMP-IKE SA from the Phase I negotiation, then use that single Phase I SA to negotiate one or more IPSec SAs using one or more Phase II negotiations. Over time, many IPSec SAs may come and go, but the ISAKMP-IKE SA between the two peers will remain. A single ISAKMP-IKE SA can be used to spawn many IPSec SAs.

Even when there are no active IPSec SAs between two peers, there may still be an ISAKMP-IKE SA between them. If a new IPSec SA is needed, the existing ISAKMP-IKE SA will be used to carry out a Phase II negotiation to produce the needed IPSec SA. If a new IPSec is needed, but there is no ISAKMP-IKE SA between the peers, a Phase I negotiation will occur to define the ISAKMP-IKE SA first, and then the Phase II negotiations can occur to produce the needed IPSec SAs.

Phase II Negotiation Steps
In outline, the steps of a Phase II negotiation are:

1. **Negotiate policy**: IPSec protocol (AH, ESP), cipher (DES, 3DES), and hash (MD5, SHA1). This is for the user's data, not the authentication sequence.
2. **Key generation**: The master key from Phase I is used to derive a new key for this IPSec SA. Optionally, IKE can perform another DH exchange to create an entirely new key for Perfect Forward Secrecy and greater security. Rekey intervals are defined.
3. **Create IPSec SA**: The above parameters and keys are combined to form a new SA in the SADB with a unique SPI number. The information is now available for the IPSec Driver.

When *an* SA needs to be renegotiated, perhaps because its lifetime has expired or an error has occurred, usually this means the IPSec SA needs a new Phase II negotiation, but it can also mean that a new Phase I ISAKMP-IKE negotiation is required. Because IPSec SAs rely upon ISAKMP-IKE SAs, when an ISAKMP-IKE SA is deleted all the IPSec SAs that depend on it are deleted as well.

Phase I Modes

The Phase I negotiation can occur in either "main mode" or "aggressive mode". The end result of the modes is the same (an ISAKMP-IKE SA) but their details differ (main mode is more secure, but slower). Phase II negotiations always occur in "quick mode".

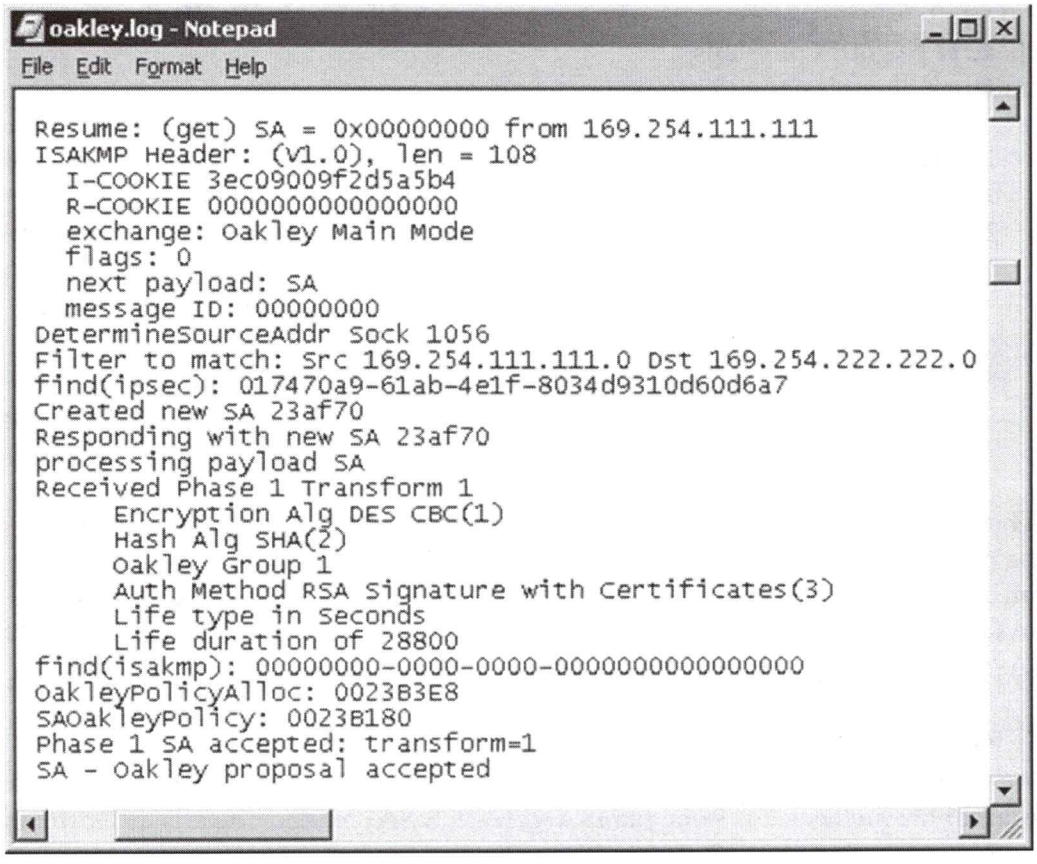
The important thing to remember is that "main/aggressive mode" is often used synonymously with "Phase I" and "IKE negotiations"; while "quick mode" is often used synonymously with "Phase II" and "IPSec negotiations".

Phase I	Main/Aggressive Mode	Negotiates "IKE SA"
Phase II	Quick Mode	Negotiates "IPSec SA"

IKE borrows these modes from the **OAKLEY** key-exchange protocol (RFC 2412), so you will sometimes also see them referred to as "OAKLEY modes" or "OAKLEY negotiations". IKE also borrows from the **SKEME** protocol for public key authentication (see Recommended Reading above concerning SKEME).

Try It Now!

On Windows you can enable logging of IKE negotiations to a text file for analysis and troubleshooting. Do this now so that the log will have data later on. You enable logging by creating a REG_DWORD value in the registry named EnableLogging set equal to 1 in HKLM\System\CurrentControlSet\Services\PolicyAgent\Oakley key. If this key does not exist, create it first (KB257225). Log entries will be written to %SystemRoot%\debug\oakley.log. The log is overwritten whenever the IPSec Policy Agent service is restarted.



ISAKMP is a complex protocol. It supports multiple payload types, multiple authentication methods, multiple key-exchange methods, multiple "modes", and

negotiates SAs by proposing sets of acceptable cryptographic algorithms in ranked order of preference. Hence, despite the large number of hash/cipher/key-length combinations, two IPSec peers can negotiate the highest security supported by both sides in common.

Diffie-Hellman-Merkle Groups

One of the most important cryptographic techniques used by IPSec is the **Diffie-Hellman-Merkle (DH)** key exchange. DH is a method for two parties to negotiate a shared secret key over an insecure channel like the Internet, but without ever sending that secret key itself over the channel-- not even in an encrypted format.

The parties must first agree which "group" to use. A **group** is a set of numbers used to control the DH exchange, consisting of a large prime number and second smaller number. The numbers can be transmitted between the peers in the clear without risk of compromising the secret key generated with them. Vanilla DH is susceptible to a man-in-the-middle attack, though, if an attacker can intercept the transmissions between the peers; the attacker can simply create two independent DH-encrypted channels with each of the peers, then ferry and/or modify the data they exchange. To combat this, IPSec authenticates the peers using Kerberos, a passphrase, or machine certificates.

The important thing to know when configuring DH is that different "groups" have different security strengths. In general, choose the group with the largest prime number if security is the most important consideration. The relevant property sheets mark the DH groups as "Low" (768-bit), "Medium" (1024-bit) or "High" (2048-bit). However, Windows 2000/XP/2003 requires a registry value to be added in order to enable 2048-bit support (KB818043):

```
          Hive: HKEY_LOCAL_MACHINE
           Key: \SYSTEM\CurrentControlSet\Services\RasMan\Parameters\
Value Name: NegotiateDH2048
Value Type: REG_DWORD
Value Data: 1  (1 to enable support, 0 to disable)
```

In Windows Vista and later, the DH exchange can also be performed using Elliptic Curve Cryptography (ECC) for ECDH. Vista supports DH group 19, which is ECC using a 256-bit random curve group, and also DH group 20, which is ECC using a 384-bit random curve group.

Authenticated IP (AuthIP)

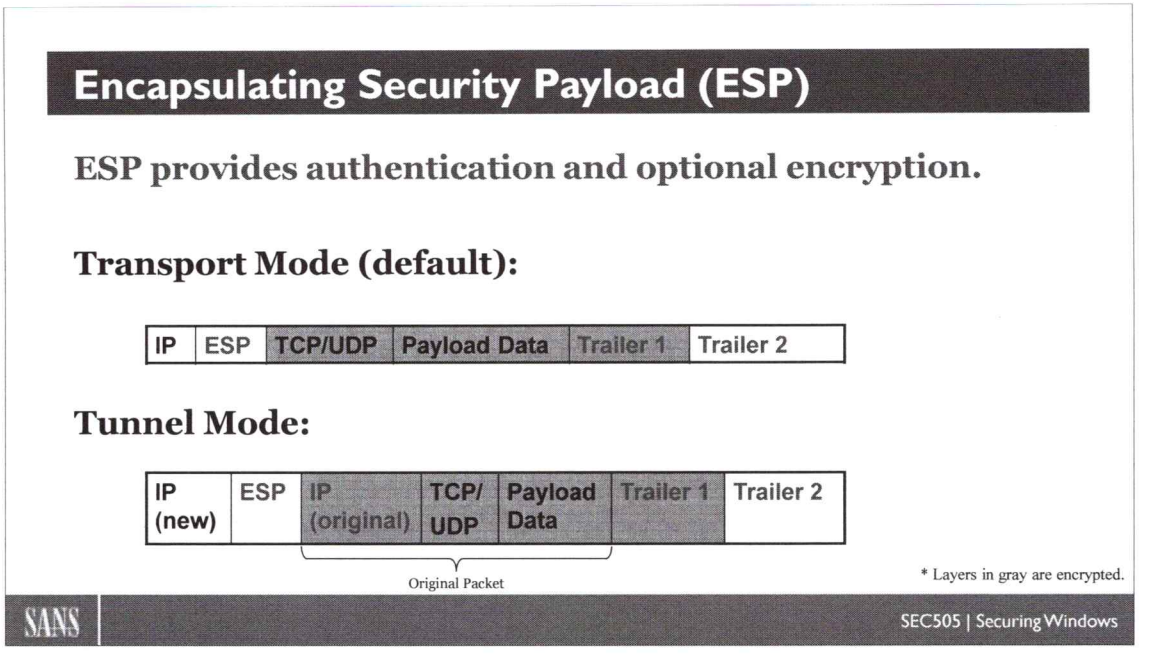
On Vista, Server 2008 and later, Authenticated IP (AuthIP) is a Microsoft proprietary enhancement to IKE for the sake of more flexible IPSec authentication. AuthIP is not the same thing as IKEv2. AuthIP will be incompatible with nearly all other (non-Microsoft) IPSec implementations, hence, AuthIP and regular IKE are negotiated in parallel at the same time, giving preference to AuthIP when the other peer supports it.

AuthIP adds significant security enhancements:

- AuthIP supports authentication of the *user*, not just the computer, of the other peer. User authentication is necessary if you wish to limit access to a TCP/UDP port with IPSec based on the user's group memberships in Active Directory. AuthIP supports Kerberos, NTLMv2 and certificate-based user authentication.

- AuthIP permits multiple authentication attempts using different protocols so that the IPSec initiator and responder do not have to use the same authentication protocol; and while an administrator could configure the computer authentication as mandatory, the user authentication might be configured as optional.

- Finally, because NTLMv2 authentication is supported, it is much easier to use IPSec between domain controllers and member computers, such as during the initial domain join, when Kerberos is unavailable and the computer has not yet received a certificate through Group Policy.



Encapsulating Security Payload (ESP)

Encapsulating Security Payload (ESP) is the core IPSec protocol which provides integrity, authentication and encryption. It can operate in either "transport mode" or "tunnel mode."

ESP Transport Mode

ESP uses both a header and a trailer. In transport mode, the ESP header is inserted just after the IP layer and before any transport layer protocols such as UDP or TCP. The ESP trailer has two parts, and both are appended to the very end.

IP	ESP	TCP/UDP	Payload Data	ESP Trailer 1	ESP Trailer 2

The layers in gray are encrypted with 56-bit DES or 168-bit 3DES in CBC mode. Encrypted layers include everything in between the ESP header and the second ESP trailer at the very end. The DES/3DES key was generated when the Phase II SA was established.

The second ESP trailer at the very end contains the Authentication Data field for the HMAC-MD5 or HMAC-SHA1 hash of the packet. Importantly, however, the scope of the authentication is smaller with ESP than in AH. In AH, the entire packet is authenticated, including the front IP layer (but not the datalink layer). In ESP, by contrast, the packet is authenticated except the front IP and the last ESP trailer.

ESP Tunnel Mode

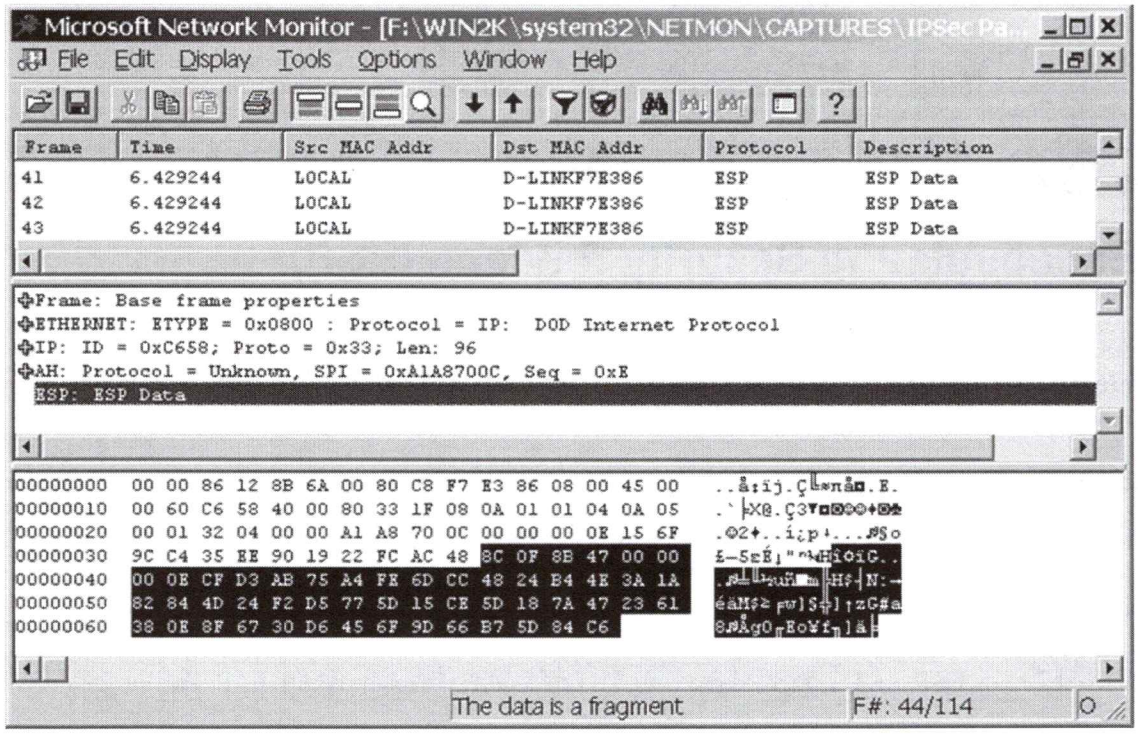
In tunnel mode, the entire original packet is placed after the ESP header and a new IP header is fabricated and placed in front of the ESP header. In tunnel mode, the entire

original packet is both encrypted and authenticated. However, the new fabricated IP header is still not authenticated. AH can be used simultaneously with ESP in order to authenticate the front IP header.

IP (new)	ESP	IP (original)	TCP/UDP	Payload	ESP 1	ESP 2

ESP tunnel mode is commonly used for Virtual Private Networking (VPN). Windows uses L2TP with ESP in transport mode, not tunnel mode, but an IKEv2 VPN does use tunnel mode.

ESP is protocol number 50.



Authentication Header (AH)

Authentication Header (AH) is the IPSec protocol which provides integrity and authentication, but not encryption (RFC 2402). AH can operate in either "transport mode" or "tunnel mode". But just ignore AH, it's incompatible with Network Address Translation (NAT) and unnecessary anyway: if you want to send plaintext packets which are digitally signed, just use ESP and turn off the encryption.

AH Transport Mode

In **transport mode**, AH authenticates the entire packet at and above the IP layer, except for a few fields in the IP layer which change during transit, e.g., time to live, type of service, etc. (these fields are zeroed out during signature construction and verification). The AH layer itself comes directly after the IP layer and just before the TCP/UDP

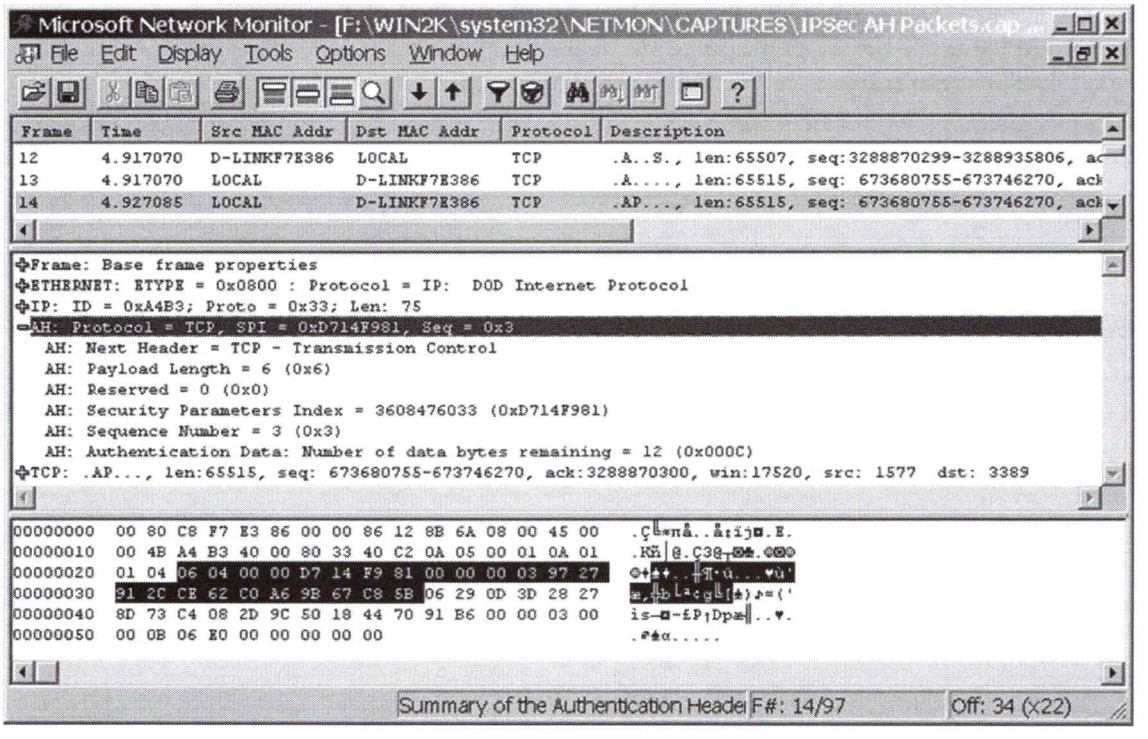
transport layer. Despite the AH header's location in the middle of the packet, the entire packet is still authenticated.

IP	AH	TCP/UDP	Payload Data

AH can be combined with ESP. In this case, the ESP data is treated no differently than any other type of data, and the AH header again comes after IP and before ESP.

IP	AH	ESP (see below for ESP)

In the AH header is the "Authentication Data" field. This field contains the authenticated hash of the packet. The hash will either be HMAC-MD5 or HMAC-SHA1. The encryption key used in the Hashed Message Authentication Code (HMAC) is negotiated when the Phase II SA is constructed. An **HMAC hash** can only be calculated and checked if both parties possess a shared secret key; this key is appended to the packet data before the packet+key is hashed with MD5/SHA1.



Interestingly, the hash value produced, whether with HMAC-MD5 or HMAC-SHA1, is truncated to 96 bits (down from 128 bits with MD5 and down from 160 bits with SHA1). This weakens the authentication a bit, but also helps to thwart brute-force searches for the shared secret key.

AH Tunnel Mode

In **tunnel mode**, the entire original packet is placed behind the AH header and a new IP header is fabricated and placed in front of the AH header. In this case, the entire original

packet is authenticated as well as the non-changing fields of the new IP header. The new IP header can have a different destination address than the original. This new destination IP address is called the "tunnel endpoint", but it is not necessarily the final destination of the original packet. The tunnel endpoint is likely to be an IPSec-enabled router.

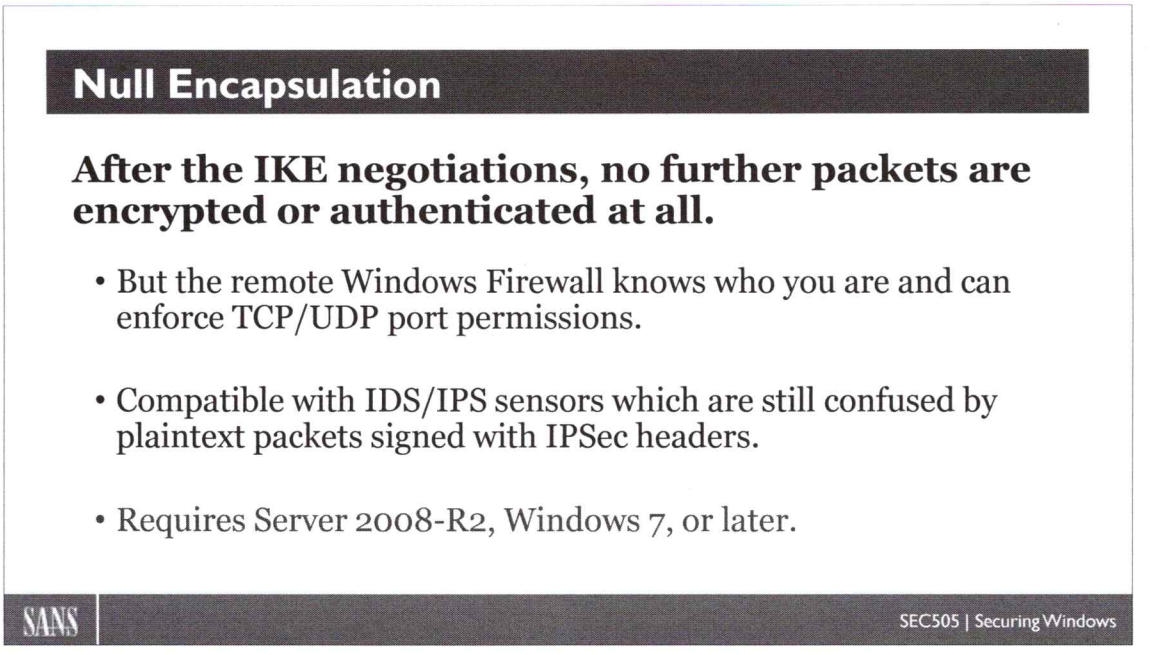
IP (new)	AH	IP (original)	TCP/UDP	Payload Data

Because of the lack of encryption, AH tunnel mode is rarely used.

AH is protocol number 51.

Summary of Differences Between AH and ESP

- AH cannot encrypt data, but ESP can.
- ESP packet encryption is optional.
- AH authenticates the entire packet, including the IP layer, while ESP only authenticates data above the IP layer.
- Both AH and ESP can be used in either transport or tunnel mode.
- AH and ESP can be combined in the same packet.
- AH cannot traverse NAT devices, but ESP can.



Null Encapsulation

Null encapsulation does not add an AH or ESP header to packets at all. After the IKE negotiations and a bit of ESP handshaking, the rest of the packets are sent normally without any modification whatsoever. This means null encapsulation provides no encryption or integrity checking of data. So what's the point then?

Advantages

The Windows Firewall can be configured to allow packets to a particular listening port from a particular source IP address, but only if these packets were sent by a peer which had first been authenticated with IKE. The successful IKE authentication is what triggers the firewall to allow access to the listening port, but only from the source IP address of the host which authenticated with IKE first (more specifically, using the AuthIP extension to IKE).

Another advantage is that null encapsulation is compatible with IDS/IPS sensors which are confused by AH or ESP headers. Even when packet payloads are in the clear, some IDS/IPS sensors just can't handle looking past the AH/ESP header to examine the plaintext payload. While this is clearly the fault of these vendors' IDS/IPS products, the issue is moot if your favorite IDS/IPS vendor won't update their code. Null encapsulation provides a partial stop-gap solution while we wait for our IDS/IPS vendors to get on the ball…

Disadvantages

IPSec null encapsulation does not encrypt, authenticate or integrity-check any packets (other than the IKE negotiation packets). Remember, "IPSec" and "encryption" are not synonyms, even when using AH or ESP.

After the IKE authentication, an attacker could spoof or modify packets from the host's IP address (which is why ESP should be used) but at least the IKE authentication requirement places another hurdle in the way of the attacker. If the client host's traffic is already encrypted and integrity-checked, such as with SSL, then the spoofing and man-in-the-middle attack risks are reduced. Again, use ESP instead whenever possible.

Requirements

IPSec null encapsulation requires Windows Server 2008-R2, Windows 7, or later operating systems.

Null encapsulation is incompatible with pre-shared key authentication.

Default IPSec Settings

Phase I (Main Mode):

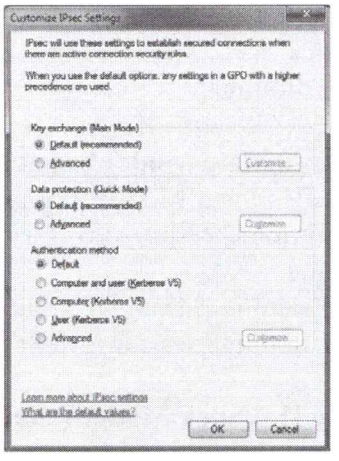
- DH: Group 2 (1024-bit)
- 128-bit AES (first)
- 168-bit 3DES (second)
- Hashing: SHA-1

Phase II: Integrity Only

- ESP (first), AH (second)
- Hashing: SHA-1
- Kerberos (computer only)

Phase II: With Encryption

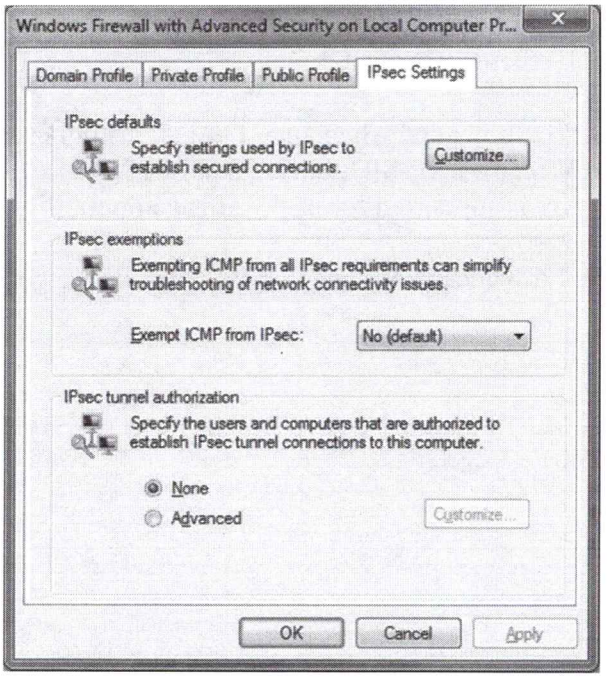
- 128-bit AES (first)
- 168-bit 3DES (second)
- Hashing: SHA-1
- Kerberos (computer only)



SANS SEC505 | Securing Windows

Default IPSec Settings

When a firewall rule or a connection security rule uses IPSec, but an IPSec option is set to "Use Default (Recommended)", what is this default and how can the defaults be changed?



The default settings come from the IPSec Settings tab on the property sheet of the WFAS snap-in itself (right-click WFAS snap-in > Properties > IPSec Setting tab). This tab also allows you to globally exempt ICMP traffic from IPSec rules.

When you click the Customize button, you'll see the dialog box shown in the slide. The defaults are as follows:

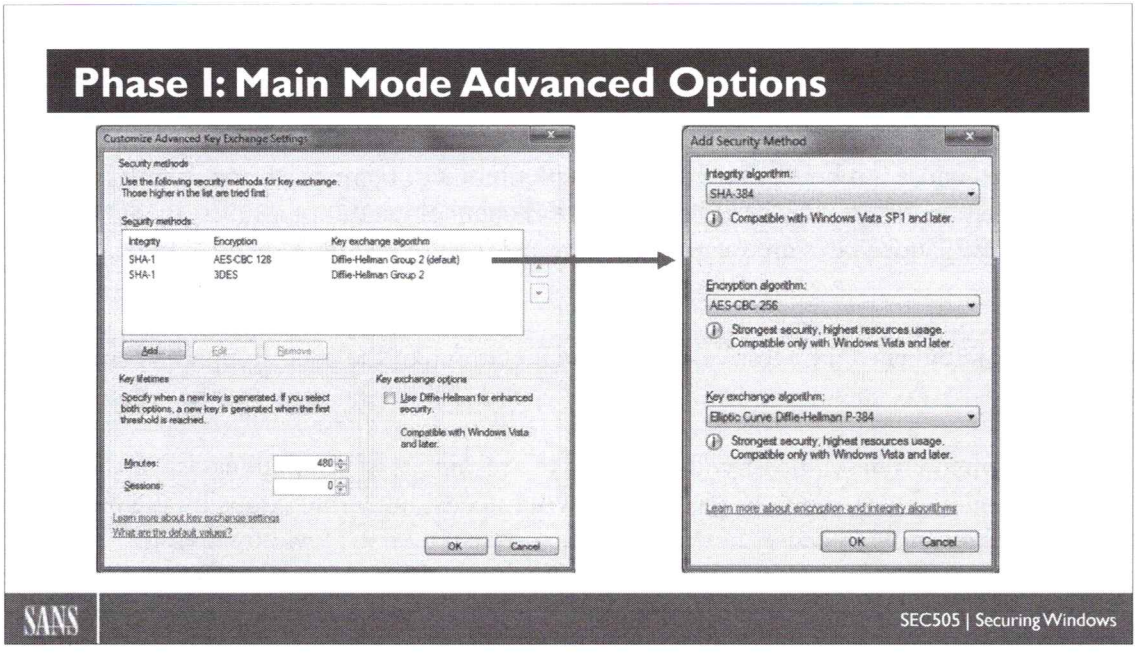
Phase I (Main Mode) Defaults:
Diffie-Hellman-Merkle: Group 2 (1024-bit prime)
Encryption: 128-bit AES is tried first (primary), then 168-bit 3DES (secondary)
Hashing: SHA-1
Key Lifetime (Minutes): 480 Minutes
Key Lifetime (Sessions): 0 Sessions (hence, only minutes decides).

Phase II (Quick Mode) Defaults for "Data Integrity" Only:
Protocol: ESP is tried first (primary), then AH is tried (secondary)
Encryption: Disabled for ESP.
Hashing: SHA-1
Key Lifetimes: 60 Minutes or 100000 KB, whichever comes first.
Authentication: Kerberos (Computer Only)

Phase II (Quick Mode) Defaults for "Data Integrity with Encryption":
Protocol: ESP
Encryption: 128-bit AES is tried first (primary), then 168-bit 3DES (secondary)
Hashing: SHA-1
Key Lifetimes: 60 minutes or 100000 KB, whichever comes first.
Authentication: Kerberos (Computer Only)

Note that SHA-1 is not recommended when an SHA-2 algorithm can be used instead. SHA-2 is supported on Windows XP-SP3, Server 2003-SP2, Vista, Server 2008, and later operating systems. SHA-1 is supported for backwards compatibility.

And when we select the "Advanced" options instead, what do we get? Let's take a look!



Phase I: Main Mode Advanced Options

The Phase I master key secures the exchange of keys negotiated in Phase II. The master key is ultimately protected using the Diffie-Hellman exchange which occurs in Phase I.

The most important option here is the key exchange algorithm. A Diffie-Hellman "group" determines the length of the prime numbers used during DH exchanges in Phase I. The larger the prime number, the more secure the encryption. DH Group 1 uses a 768-bit prime number, Group 2 (the default) uses a 1024-bit prime, and Group 14 provides 2048 bits. Elliptic Curve DH is more secure than the original DH even though the bit-sizes are smaller, e.g., 256 and 384 bits.

Two peers must support a common key exchange method in order to communicate. Note that Elliptic Curve DH is only supported in Windows Vista and later by default, but Microsoft may release updates to Windows XP/2003. DH Group 2 is the default because it has the widest compatibility with Windows and non-Windows peers, and is also very secure.

Windows 7/2008-R2 and later include a checkbox to "Use Diffie-Hellman for enhanced security". If this box is checked, DH is always used instead of AuthIP-derived keys. AuthIP is an IPSec enhancement first introduced with Vista. It's use is negotiated automatically. If AuthIP is used, then, by default, a Kerberos-derived or NTLM-derived encryption key is used instead of a key derived from DH. AuthIP key derivation is faster than DH, but less secure. You may also need to check this DH box if your environment is under regulatory restriction to always use DH, such as in a classified military network.

One of the golden rules of cryptography is to avoid using the same key for too long or to encrypt too much data. How long is too long? How much is too much? That depends on your adversaries, but in this dialog box you can specify your key lifetime in minutes and/or sessions. If you configure both, whichever maximum is hit first (minutes or sessions) will cause a rekeying and then both counters will be reset. Keep in mind that the keys referenced here are for securing the IKE channel itself, not all the gigabytes of data to follow. Those keys are negotiated in Phase II and have their own lifetime parameters.

What kind of key has a time to live here? If you edit one of the security methods on the left-hand side, you can choose the cipher/key type as well as the hashing algorithm.

Depending on the time-value of your data and the resources of your adversaries, choose the cipher and key size which is adequate for your needs, but avoid choosing the highest security levels simply because they're stronger -- they will also slow down your computers unnecessarily. For widest compatibility, consider using 3DES instead of AES. AES is currently only supported by default in Windows Vista and later.

Note that a Windows 2000 peer must have the High Encryption Pack (HEP) installed in order to support 3DES. The HEP is included with Service Pack 2 and later, and it is built into Windows XP and later by default. If a peer lacks High Encryption support, but 3DES is required, the IPSec session will fail.

For the hashing algorithm, never choose MD5, only choose SHA-1 if you must for backwards compatibility, and prefer an SHA-2 algorithm whenever possible (SHA-2 algorithms include SHA-224, SHA-256, SHA-384 and SHA-512). SHA-2 algorithms are supported on Windows XP-SP3, Server 2003-SP2, Vista, Server 2008, and later operating systems.

Perfect Forward Secrecy (PFS)

Perfect Forward Secrecy (PFS) for IPSec means that a new DH exchange is performed whenever a new symmetric key needs to be created. PFS is disabled by default. PFS can be enabled for either Phase I or Phase II keys independently using the NETSH.EXE command-line tool. In Windows 2000/XP/2003 there was a GUI checkbox for PFS, but in Windows Vista and later this GUI element was removed, hence, the necessity of using NETSH.EXE to enable it. Only very high security environments will be required to enable PFS.

Quantum Computing Resistance

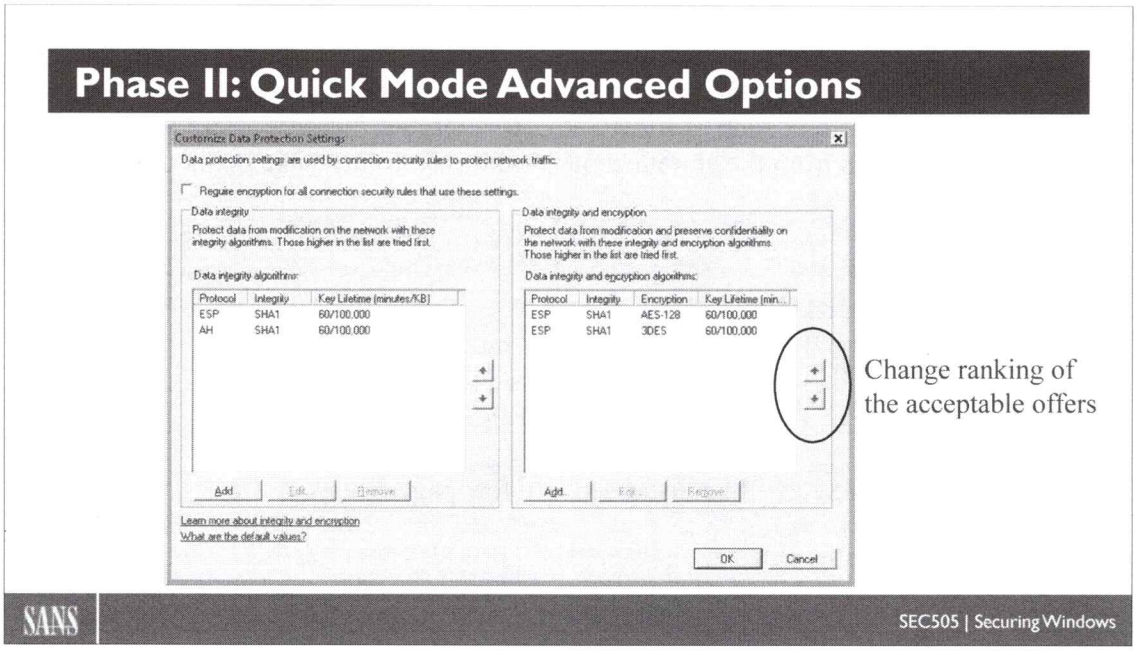
Sometime in the next 50 years there will be a "quantum apocalypse" for cryptography when quantum computing becomes widely available. The threat is real enough that in August of 2015 the National Security Agency (NSA) in the United States started to provide public guidance concerning quantum-resistant countermeasures:

https://www.nsa.gov/ia/programs/suiteb_cryptography/

The most important points from this announcement are 1) new quantum-resistant algorithms must be developed and deployed in the next several years, 2) increase the size of keys used today, and 3) pre-shared keys can provide quantum resistance if they are very large and truly random, such as for IPSec. How large is "large"? Here are the recommended key sizes from the announcement while we wait for the better algorithms:

AES: 256 bits
RSA: 3072 bits
Hashing: SHA-384
Diffie-Hellman (DH) key exchange: 3072 bits
Elliptic Curve (ECDH) key exchange: curve P-384
Elliptic Curve (ECDSA) signatures: curve P-384

Pre-shared keys with IPSec will be discussed in a few pages (below).



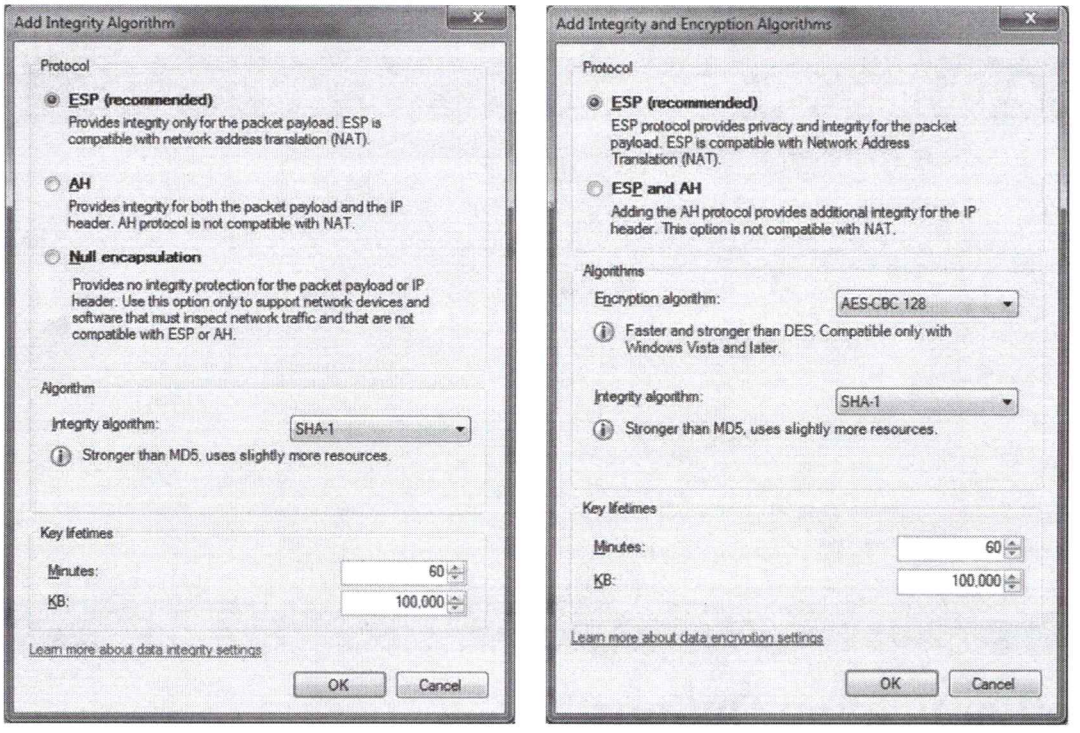
Phase II: Quick Mode Advanced Options

The left-hand side of this dialog box is for AH by itself or for ESP with the payload encryption disabled. The right-hand side is only for ESP with encryption enabled. These are the settings used by firewall rules and connection security rules that specify a "secure connection" (left side) or "secure connection with encryption" (right side).

If you check the box at the top labeled "Require encryption for all connection security rules that use these settings" then the left-hand side of the dialog box becomes grayed out: you can no longer use AH or ESP with no encryption. Connection rules don't specify Phase II/Quick Mode settings, they pull these settings from here.

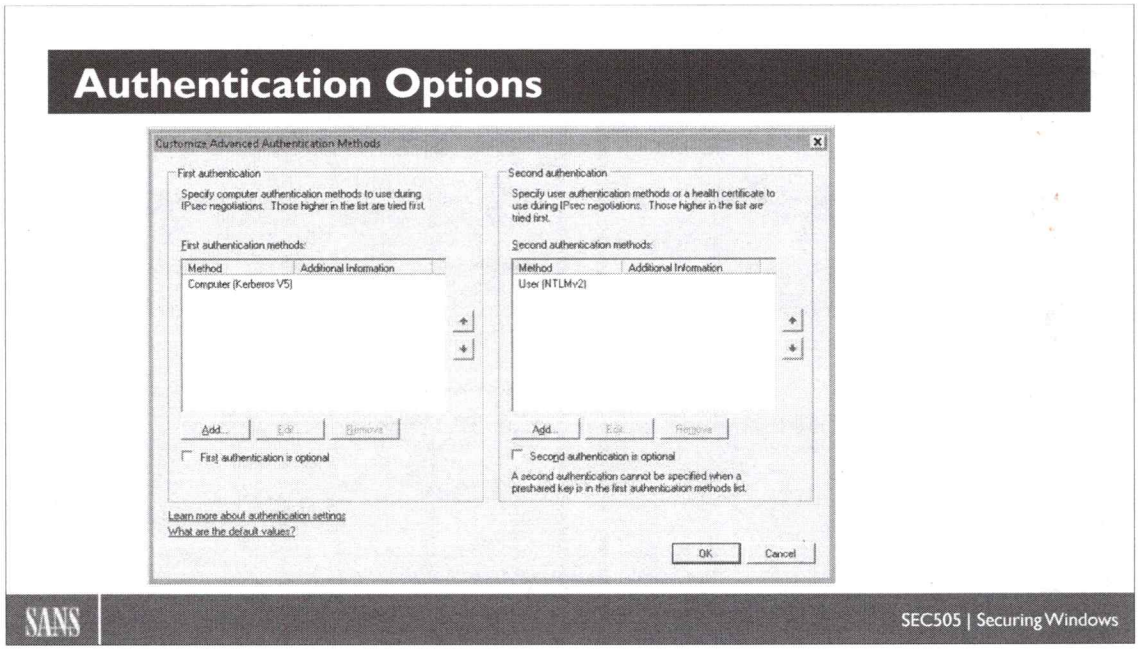
Very importantly, ESP can traverse devices like routers and firewalls that perform Network Address Translation (NAT) without the IPSec integrity checks failing. Because NAT is so widely used, ESP is preferred over AH in all cases, even when encryption for ESP is disabled. Hence, unless you have specific reason for doing so, don't use AH.

If you edit one of the integrity-only methods on the left-hand side, you can choose the algorithm (AH or ESP with encryption disabled), the hashing algorithm, and the key lifetimes.



If you edit one of the methods on the right-hand side that use ESP payload encryption, you can choose whether to use ESP by itself or ESP+AH, the encryption algorithm, hashing method, and key lifetimes. Remember that AES is not supported out-of-the-box by versions of Windows prior to Vista, and Windows 2000 requires at least Service Pack 2 to even support 168-bit 3DES.

AES Cipher Block Chaining (AES-CBC) is a mode of AES encryption in which each block of plaintext is XOR-ed with the previous block of ciphertext, making the overall message more difficult to crack. AES-CBC requires Vista or later. AES Galois Counter Mode (AES-GCM) is a mode of AES which is faster than CBC mode. Galois Message Authentication Code (GMAC) is the use of GCM for signing and integrity-checking only. Note that GCM/GMAC requires Vista+SP1 or later.



Authentication Options

The authentication methods supported by an IPSec Policy control how the computer and/or user authenticates to the other peer during Phase I negotiations. Note that in Windows 2000/XP/2003 only the computer can authenticate, not the user, but in Windows Vista and later the user can authenticate via IPSec too.

Multiple authentication methods can be enabled simultaneously. There are three authentication methods available in Windows 2000/XP/2003:

- Kerberos (Computer)
- Certificate (Computer)
- Preshared Key

In Windows Vista and later, you can also authenticate using:

- Kerberos (User)
- Certificate (User)
- Health Certificate (Computer -- deprecated)
- NTLMv2 (User)
- NTLMv2 (Computer)

The authentication methods are attempted in the order shown on the property sheet, from top to bottom. The first method acceptable to both peers is used. If two peers do not have at least one common authentication method, the authentication will fail. Once a mutually agreed-upon method fails, the next method in the list is *not* attempted in

Windows 2000/XP/2003, but Windows Vista and later can be configured to try a second time using a different authentication protocol.

In Windows 2000/XP/2003, for example, if a remote peer is configured with both Kerberos and certificate authentication, *in that order*, and the server is configured with both Kerberos and certificate authentication *in that order*, and both sides have mutually-acceptable certificates, then authentication will still fail because the remote peer cannot reach the domain controller for Kerberos. But Windows Vista and later, on the other hand, can be configured to try a second time, and, in fact, if both computers are Vista or later, each peer can use a different authentication method (as long as it is supported by the other peer, even though it is not that peer's first choice).

Kerberos Authentication

Every Windows computer in the domain has a computer account in Active Directory. The computer uses this account to authenticate to the domain when it starts up. This same account can be used with IPSec Kerberos authentication. In Windows Vista and later, the user can also use his/her account in AD for Kerberos authentication of IPSec.

Keep in mind that for Kerberos authentication to work, the IPSec peers must be in the same or trusted domains, and each peer must be able to contact a domain controller. Hence, Kerberos authentication is really not appropriate for remote access scenarios.

Certificate Authentication

Peers can be mutually authenticated with their machine certificates. To be successfully authenticated, the issuer of each computer's certificate must be trusted by the other IPSec peer, and that is all that is required. It is not the case that an IPSec peer must have a copy of the other peer's certificate beforehand, or know any of the details of the certificate's credentials. No one-to-one or many-to-one mapping occurs. Computer certificates are not mapped to computer accounts in Active Directory by default. The purpose of certificate authentication is, in part, to authenticate with peers in foreign domains or that can't be domain members.

The certificate can be issued by a Microsoft, Netscape, Entrust, VeriSign or any other CA. There are no special requirements for the X.509v3 certificate itself other than the standard validity and integrity checks.

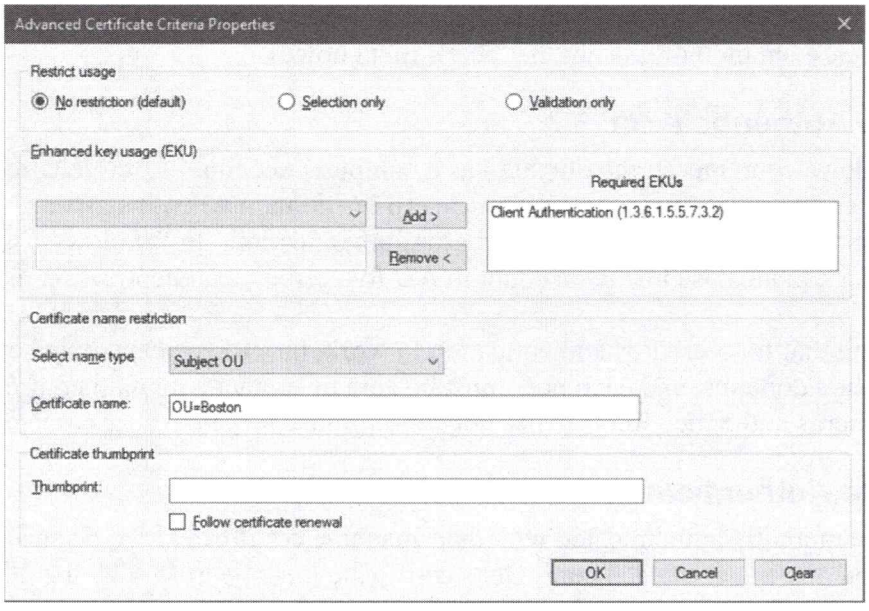
A Windows peer must have the certificate of the issuing Certification Authority (CA) of its own IPSec certificate and the other peer's certificate in its personal Trusted Root CA Store. This is the "personal store" of the computer itself, not the human sitting at it. Use Group Policy to distribute trusted CA certificates automatically (as discussed in the PKI seminar).

A Windows computer can obtain an IPSec certificate through auto-enrollment from a Windows Enterprise CA. An alternative is to manually request and install the certificate with the Certificates snap-in or the Certificate Services Website. Manual installation requires administrative rights at the system.

Certificate authentication is appropriate when the other peer does not support Kerberos, domain controllers cannot be accessed for Kerberos authentication, the other host is not in a trusted domain, or L2TP is in use.

Advanced Certificate Criteria

On Windows 8, Server 2012 and later, there is an Advanced button next to the certificate authentication options when configuring IPSec authentication. The following is the dialog box shown when clicking the Advanced button:



This allow us to reject all certificate authentications which do not meet the criteria in the dialog box, such as for Enhanced Key Usage (EKU) settings in the certificates, subject name fields, or an exact certificate hash value.

Using these filtering criteria, for example, would permit rejecting all IPSec connections from any computer or user not in a particular Organizational Unit (OU), hence, group memberships are not the only way to isolate IPSec peers.

These criteria might also be used with a custom certificate template used only for IPSec, and perhaps even a subordinate CA which issues only IPSec certificates.

Certificate Revocation Checking

The IPSec Driver does not perform certificate revocation checking by default. A certificate is "revoked" if it is on the list of revoked certificates its issuing CA publishes. A certificate might be revoked if its private key (or the private key of the CA) have been compromised. Hence, for maximum security, you should enable certificate revocation checking by setting the following registry value, but beware of performance penalties and hassles caused by configuration mistakes:

Hive: HKEY_LOCAL_MACHINE
Key: \SYSTEM\CurrentControlSet\Services\PolicyAgent\Oakley
Value Name: StrongCrlCheck
Value Type: REG_DWORD
Value Data: 1 or 2 (select "Hex" as the Radix)

1: Normal CRL checking in which the check fails only if a CRL is successfully obtained and the certificate is on it.

2: Strong CRL checking in which any error whatsoever in the download or processing of the CRL returns a failure, including finding that the certificate is on the CRL.

A certificate will include a CRL Distribution Point (CDP) field which lists the URLs where the certificate's CRL can be obtained. The IPSec peer performing the revocation checking must be able to access one or more of these URLs.

"Health Certificates" and NAP: Deprecated

IPSec supports certificate-based authentication. A special type of certificate is called a "health certificate". A health certificate is issued to a computer only after that computer has passed a variety of security health tests, e.g., confirmation of a running virus scanner, a personal firewall is engaged, recent patches applied, etc. Health certificates are a part of the much larger system of servers, protocols and procedures that comprise Microsoft's Network Access Protection (NAP) scheme, but NAP was deprecated, hence, just ignore any references to health certificates for IPSec.

Preshared Key Authentication

The "preshared key" is actually just a passphrase. The passphrase bits are mixed with and protected by the master key derived in Phase I. However, the passphrase itself is stored unencrypted in the IPSec Policy in both Active Directory and in the local registry.

Prior to Vista, the registry stores the passphrase in a value named "ipsecData" under HKLM\SOFTWARE\Policies\Microsoft\Windows\IPSec\Policy\Local\ipsecNFA{*xxxxxx xx-xxxx-xxxx-xxxx-xxxxxxxxx*}, where the "*xxx*" number is the GUID for the IPSec policy object itself. The permissions on these keys allow the local Users group read access.

With Vista and later, the pre-shared keys are found under HKLM\SYSTEM\Current ControlSet\Services\SharedAccess\Parameters\FirewallPolicy\Phase1AuthenticationSets.

> **Important**: If an adversary obtains a certificate or knows the preshared key, this only permits the adversary to open his or her own IPSec channel to your peer. The adversary *cannot* decrypt other captured sessions with this information because the Diffie-Hellman-Merkle exchange with each session is unique.

The following PowerShell command will extract all local pre-shared keys:

```
Get-NetIPsecPhase1AuthSet |
Select-Object -ExpandProperty Proposal |
Where { $_.AuthenticationMethod -eq 'PreSharedKey' } |
Select-Object -ExpandProperty PreSharedKey
```

Support for preshared key authentication is mainly intended for RFC compliance, lab testing, or when security is not an issue (KB240262). Kerberos or certificate authentication should be used on production servers instead.

The exception, ironically, is when responding to the threat of quantum computing. A random pre-shared key which is over 100 characters in length can help to withstand brute force attacks by adversaries using quantum computers. However, that being the case, if an adversary can simply steal the pre-shared key from a GPO or the registry of a previously-compromised machine, then the quantum-computing threat is irrelevant. For most organizations, avoiding pre-shared keys is still the better overall policy, at least for the time being.

Authenticated IP (AuthIP) in Windows Vista and Later

Authenticated IP (AuthIP) is an enhanced version of Internet Key Exchange (IKE) for IPSec. AuthIP has the following benefits over regular IKE:

- Authentication of the user, instead of the user's computer, using Kerberos, NTLMv2, a user certificate, or a Network Access Protection health certificate.
- The user authentication can be performed alone or after the user's computer first authenticates itself to the target (dual authentication).
- Multiple authentication attempts with different authentication protocols.
- A different authentication protocol can be used by each IPSec peer when negotiating a session, e.g., a client might use Kerberos and the server might use a certificate.

Only Windows Vista and later supports AuthIP. When communicating with Windows Server 2003 and earlier peers, IPSec on Windows Vista and later will automatically downgrade to regular IKE for backwards compatibility.

Optional vs. Mandatory Authentication

If you mark both authentication choices as "optional", authentication will no longer be required. This would permit untrusted computers to open their own IPSec channels to your machines and makes man-in-the-middle attacks more likely. Never choose this option unless you are forced to do so by your planning constraints.

If neither authentication choice is marked as "optional", then both authentication events are mandatory. You can use this feature, for example, to first force authentication of the peer's computer using a certificate, then force the user sitting at that computer to

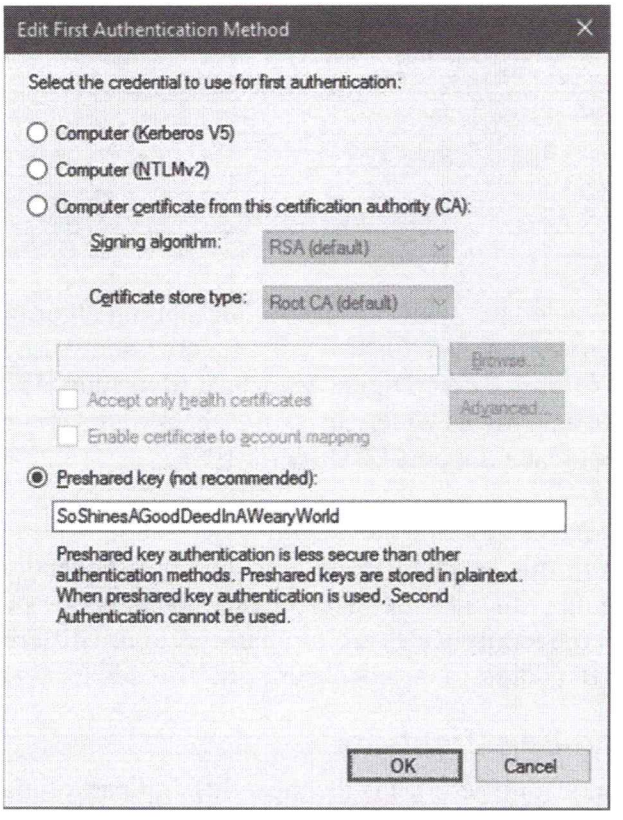
authenticate with Kerberos. If either the remote computer or user lacks the Log On Over The Network right, the IPSec channel can be blocked.

The first authentication method might safely be marked as "optional" if you intend to rely solely on the second method. This would be done, for example, when you wish to only authenticate users, not computers, because IPSec user authentication can only be enabled in the second authentication attempt, not the first one.

If you select a user-based authentication method in the second choice, any other methods on the second-choice list must be user-based as well. If you choose a computer-based method in the second-choice list, all the other methods in that list must be computer-based too. In short, you cannot mix user- and computer-based authentication methods in the second-choice list, it is all of one or the other.

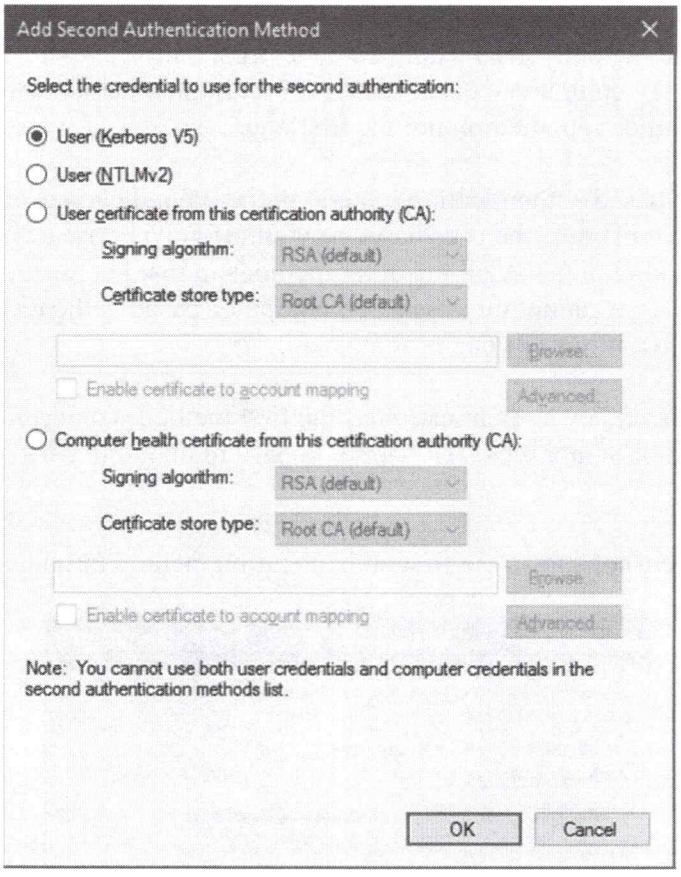
If you select pre-shared key authentication as the first method, you cannot have a second authentication attempt of any type. This is so, in part, to maintain backwards compatibility.

The following screenshot shows the first authentication method options.



The next screenshot shows the second authentication method choices, but remember that if you choose pre-shared key authentication as the first choice, you cannot add a second

choice. Note that only the second authentication choices include user-based authentication methods.



In Windows 7/2008-R2 and later, you can specify the signing algorithm used with certificate-based authentication. RSA is the default, very widely used, and backwards compatible, while the Elliptic Curve Digital Signature Algorithm (ECDSA) is newer, faster, more secure, but also not as widely used or as backwards compatible. Only Windows 7/2008-R2 and later support ECDSA for IPSec.

Because the foregoing pages describe the default IPSec settings, these will be the settings used whenever a firewall rule is marked to require a secure connection. But what if you need to use something other than the defaults on a particular server or OU of laptops? This is what Connection Security Rules are for, namely, to use different settings than the defaults for particular IP addresses, protocols or ports.

IPSec Chicken-and-Egg Problems

Configuring the authentication settings is the most difficult part of managing IPSec. One reason this is so is because if you require IPSec to access the TCP/UDP ports necessary to perform authentication, you'll never get to the ports, which means you'll never establish an IPSec connection, and so on in an infinite loop. If you require IPSec to access the Kerberos ports (TCP/UDP 88) and IPSec itself requires Kerberos

authentication, how will you ever get to the Kerberos ports? Hence, you either cannot require IPSec to access the Kerberos ports or you'll have to use a different authentication method (pre-shared key, NTLM or certificates) to access the Kerberos ports with IPSec.

There is another IPSec authentication issue too. When you use a Windows Firewall rule to restrict access to a listening port based on the group memberships of the user or computer accessing that port, how is the group memberships information conveyed to the target computer? It turns out that the target computer needs to query a domain controller for this information using various protocols, hence, the domain controllers must be configured to allow these inbound protocols and the target computer needs to be configured to allow these outbound protocols to the controllers. If you want to use IPSec with these protocols to/from the controllers, then we have to worry about the chicken-and-egg problem again; for example, you cannot restrict by group memberships access to the ports on the domain controllers for querying group memberships.

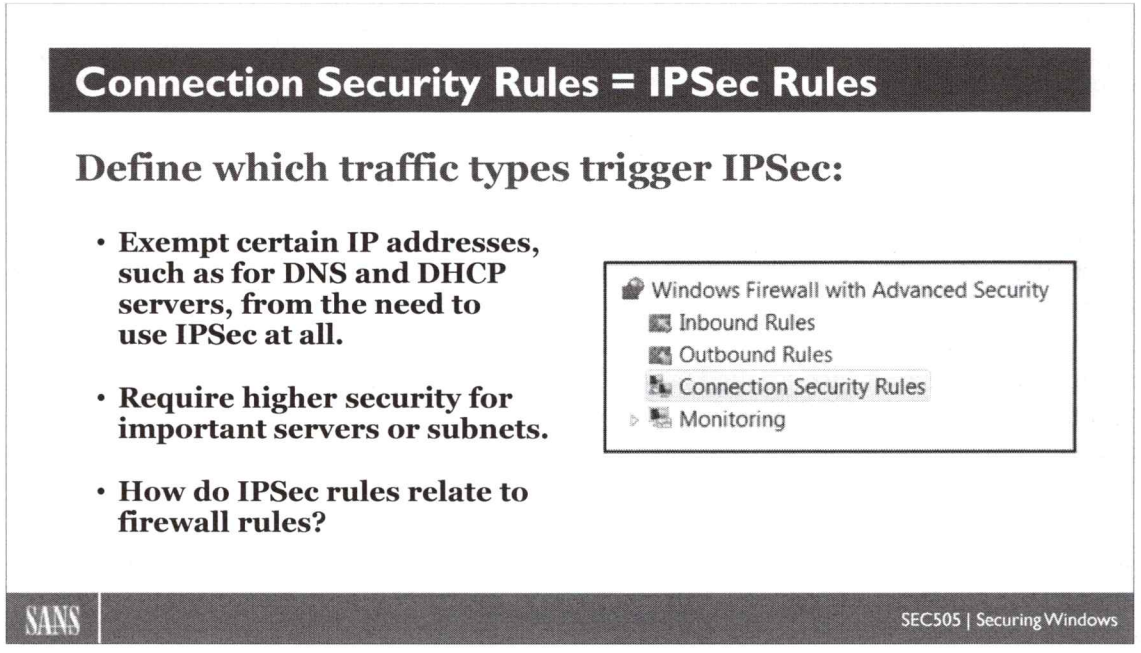
Here are the ports which must be accessible on domain controllers in order to allow the querying of group memberships when using IPSec and firewall rules to restrict access:

DNS	UDP and TCP 53
Kerberos	UDP and TCP 88
RPC	TCP 135, 49152-65535
LDAP	UDP and TCP 389

Remote Procedure Call (RPC) networking uses upper-level port numbers which are dynamically assigned. You can see what range of ports are used for RPC on your computer by running "netsh.exe int ipv4 show dynamicportrange tcp".

And don't forget the ports and protocols used by IPSec itself:

IKE	UDP 500, 4500
AH	Protocol ID 51
ESP	Protocol ID 50



Connection Security Rules = IPSec Rules

The firewall rules allow/block packets, and sometimes packets must be secured with IPSec or else they'll be blocked. But it's the Connection Security Rules which specify which IP addresses, protocols and port numbers should trigger the negotiation of IPSec. In order to use IPSec, each peer must have a Connection Security Rule which indicates that certain packets exchanged with the other peer must use IPSec.

Remember, though, that a Connection Security Rule does not by itself allow traffic through the Windows Firewall. There must still be a firewall rule enabled to allow the traffic through *after* it has been decrypted or otherwise processed by IPSec.

If a firewall rule permits a certain type of traffic but does not require IPSec for those packets, then that traffic is permitted whether or not it is secured with IPSec. If a firewall rule permits a certain type of traffic but only if it is secured with IPSec, then there must be a Connection Security Rule which successfully negotiates IPSec before the firewall examines that traffic.

The firewall must be enabled for the network profile which is active in order for any Connection Security Rules to become active too. The inbound and outbound defaults can both be set to Allow in the firewall if you wish to use IPSec but do not wish to perform any packet filtering (perhaps only for the Domain profile).

An inbound or outbound firewall rule which requires a secure connection will not by itself, in the absence of a Connection Security Rule, trigger an IPSec negotiation between the peers. At least one Connection Security Rule must be configured on each peer before IPSec will function on that peer.

Creating Connection Security Rules

There is a built-in wizard to help you create Connection Security Rules. The wizard will ask you why you are creating the rule and then will prompt you for the necessary information. However, once a rule is created, you can always go back to its properties to edit its settings, and these property sheet tabs are the same no matter what type of rule you create with the wizard, hence, let's focus on editing the rule after its been created.

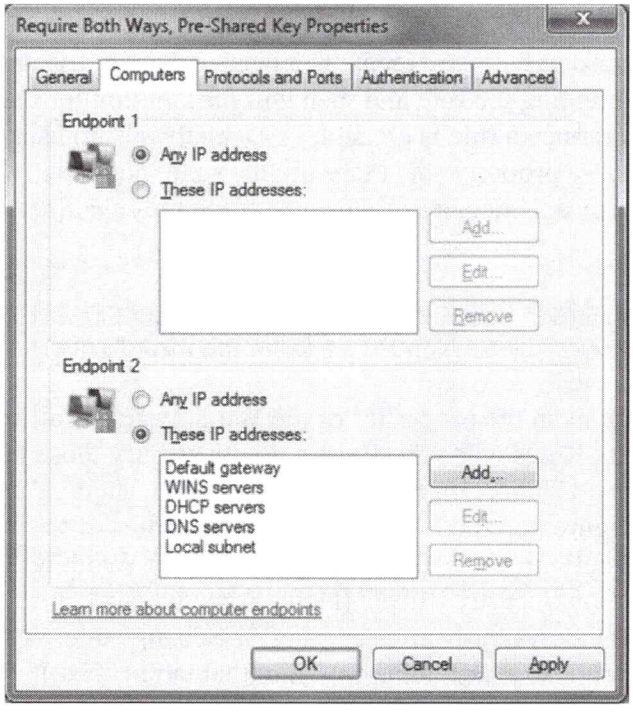
> ***Try It Now!***
>
> To create a Connection Security Rule, open the Windows Firewall snap-in > right-click Connection Security Rules > New Rule > follow the wizard's questions.

The default IPSec options in the properties of the Windows Firewall snap-in will be used by Connection Security Rules when those rules do not specify those options explicitly, i.e., when an option in a Connection Security Rule is set to "Default", the value is taken from the options configured on the IPSec tab of the properties of the Windows Firewall snap-in. To see what options are actually being used, open Windows Firewall > Monitoring > Connection Security Rules. (Note: Sometimes when you click "Default" in the IPSec tab of the Windows Firewall properties, the change doesn't stick after clicking OK, so you'll need to choose a particular option in that tab instead.)

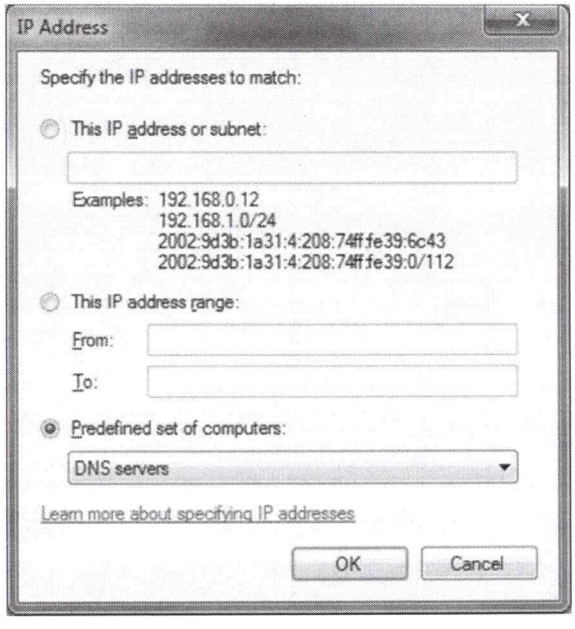
Let's discuss each of the tabs in the property sheet of a Connection Security Rule. Simply right-click the rule and select Properties.

Computers Tab

The Computers tab is for defining the endpoints or peers of the IPSec connection to which this rule applies. If a connection attempt matches the information on the Computers tab, then the Connection Security Rule is triggered for IPSec negotiations. The information on the other tabs do not apply or have any effect if the two endpoints under scrutiny do not match the endpoints defined on the Computers tab.

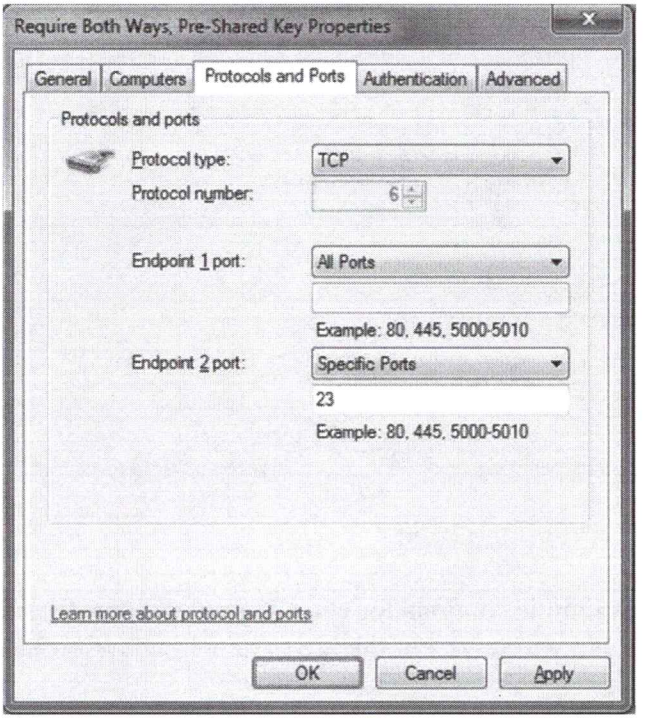


When you click Add or Edit, you can specify a single IP address or a set of IP addresses by range, subnet mask or CIDR notation (IPv4 or IPv6). You can also specify IP addresses which may change dynamically and do not need to be hard-coded into the IPSec policy. In this case, you specify endpoints as being your DNS, WINS or DHCP servers, whatever they are at the moment, or your current local subnet, as defined by your current IP address and subnet mask. These dynamic options are especially nice for roaming laptops.



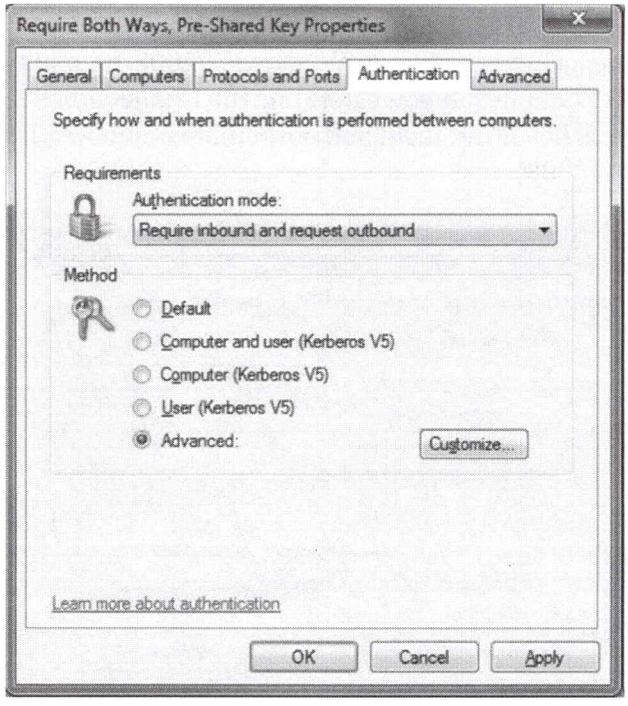
Protocols and Ports Tab

The Protocols and Ports tab exists only on Windows 7/2008-R2 and later. This tab is missing on Vista/2008. On this tab you can refine the Connection Security Rule so that it is only triggered by a particular protocol and port number combination (instead of just IP addresses, like in Vista/2008).



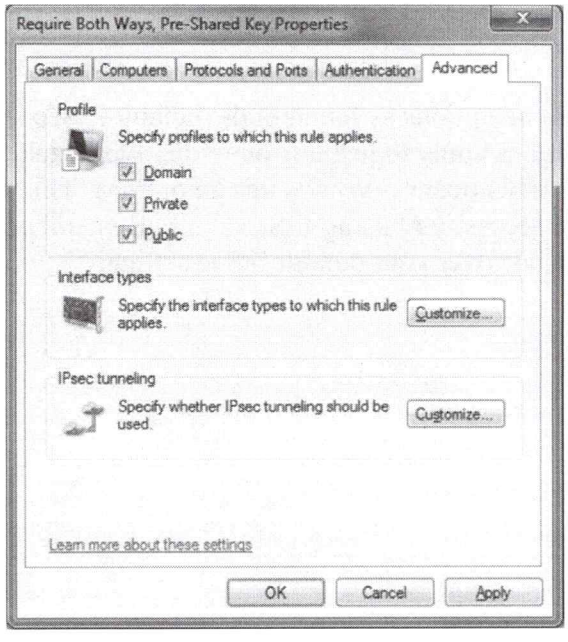
Authentication Tab

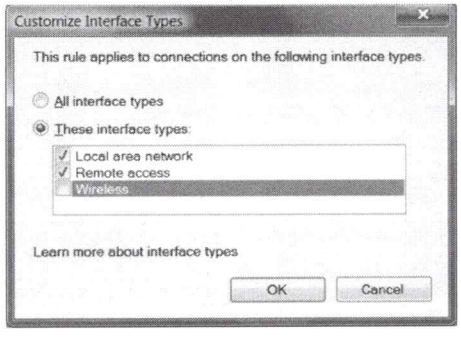
The Authentication tab provides the exact same options as found in the default IPSec settings, except that these authentication options apply to just this one rule. Please take careful note of the option named "Do Not Authenticate". Why is this an option? This is how you exempt certain endpoints from the necessity of using IPSec at all. Remember that Connection Security Rules take precedence over your default IPSec settings.



Advanced Tab

The Advanced tab has familiar options for choosing network profiles and interface types (LAN, remote access and wireless), but the IPSec tunneling options are different.



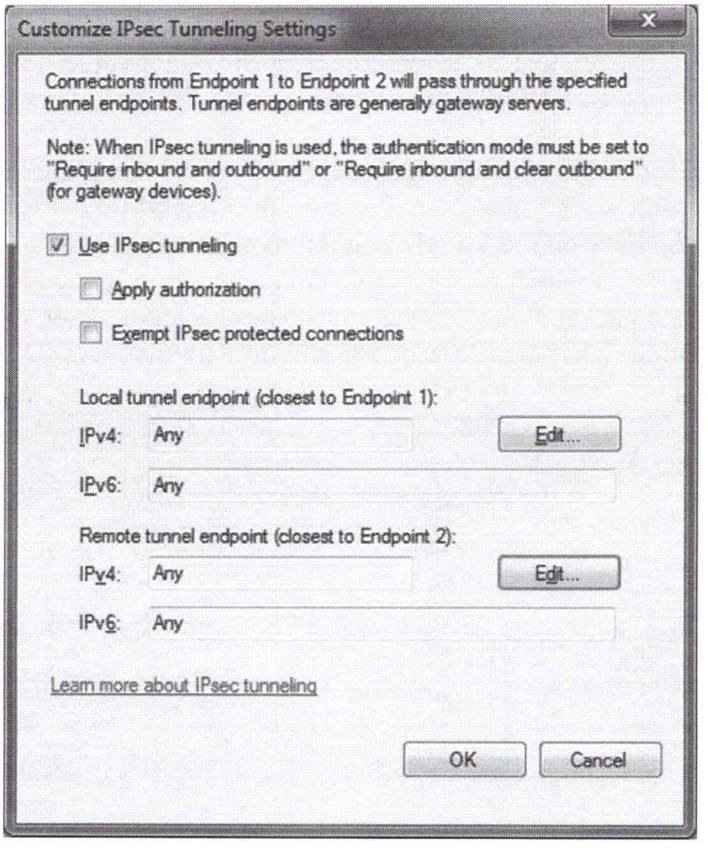


IPSec Tunneling

IPSec tunneling is very similar to Virtual Private Networking (VPN). In fact, many organizations use IPSec tunnels as their VPNs and there is a widespread misconception that "IPSec tunnel" and "IPSec VPN" are synonymous terms. Strictly speaking, though, an IPSec VPN does not have to use IPSec in tunnel mode (think of L2TP IPSec VPNs) and you can use IPSec in tunnel mode between two endpoints in a way that few people would consider to be a VPN. If you do wish to set up a VPN with Windows, though, you shouldn't use IPSec in tunnel mode, you should use L2TP with IPSec in transport mode or SSTP (or PPTP).

As discussed earlier, IPSec can operate in either transport mode or tunnel mode. In tunnel mode, the original packet is encapsulated behind a new IP header (among other changes) and the new front IP header does not have to have the same source and destination IP addresses as the original header (now encapsulated inside the new IPSec packet). In fact, the IP addresses are usually different. Usually the tunnel endpoints are gateway devices, and these gateways might be using IPv6 internally and IPv4 on their Internet interfaces.

Since the tunnel endpoint IP addresses are normally different from the IP addresses of the hosts communicating, what are the IP addresses of the endpoints? On the Advanced tab, clicking the Tunneling button and enter these IP addresses. These will be the IP addresses of the outer-most IP header created for the sake of IPSec.



When would you use such a feature instead of a full VPN? Rarely. You might have a non-Windows IPSec gateway or server that doesn't support L2TP, but normally you'll either use regular transport mode IPSec or a true VPN like L2TP or SSTP.

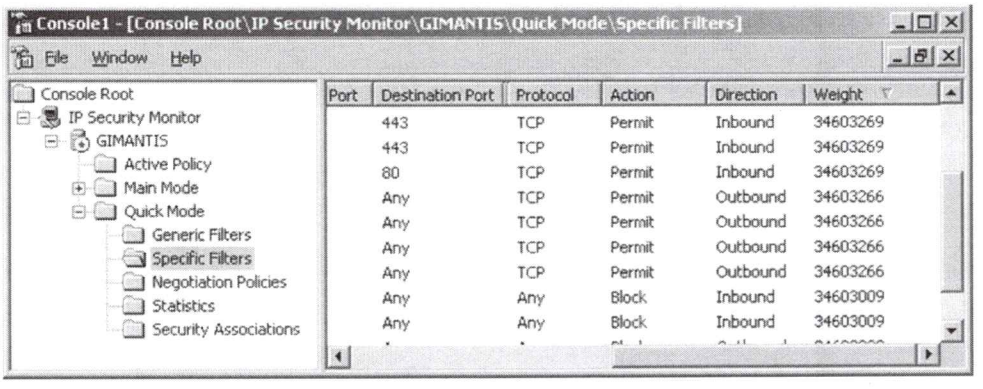
The "Apply authorization" checkbox is to limit which computers and users are permitted to tunnel through the present device. These computers and users are configured by going to the properties of the Windows Firewall snap-in > IPSec Settings tab > IPSec Tunnel Authorization section > Advanced > Customize button.

The "Exempt IPSec protected connections" checkbox will cause the computer to not pump any packets through the tunnel which have already been secured with IPSec by virtue of a different Connection Security Rule. If you want all matching packets to go through the tunnel, don't check this box.

Order of Filter Precedence

The order of the Connection Security Rules in the dialog box does not matter. If any of the rules match a packet, then that rule will be triggered. However, a single IPSec policy can contain multiple Connection Security Rules, and each rule will have its own defined endpoints, profiles and interface types. An inbound or outbound packet may match two different rules, and these rules may have conflicting actions. The ambiguity is resolved by the way the IPSec driver matches packets against rules. All the filters from all the rules in the policy are arranged in order from most specific to least specific when processed by the IPSec driver. (This is similar to the behavior of a router matching packets against its route table.) Each filter is assigned a weight value, and of the filters which match a given packet, that filter with the highest weight value (i.e., the most specific filter) is the one which will be matched.

You can see all the filters being enforced by the IPSec driver using the IP Security Monitor MMC snap-in. In the snap-in, navigate to the Quick Mode > Specific Filters container and sort the lines on the Weight column in descending order. From top to bottom, this is the order in which packets are compared against loaded filters; the first filter which matches a packet is the one that wins, and the action associated with that filter is what determines what the IPSec driver will do with that packet.



Note that if a packet matches two filters because they both specify a range of IP addresses in which the packet falls, the range defined with the greater number of subnet mask bits is the filter which will win. Also, tunnel mode filters always take precedence over transport mode filters.

However, it is still possible for two filters in two different rules to be equally specific. When this occurs, the filter associated with the more restrictive action is the filter (and rule) which takes precedence; for example, two rules with equally specific filters might select the same packet, but the rule which blocks that packet takes precedence over the rule which only encrypts it, because blocking is more restrictive than encrypting.

If you received a CD-ROM with this courseware, then the IPSec folder on the CD will contain a file named IPSec_Order_of_Specificity.csv which lists the relative weightings of the various criteria that a packet can match (highest weight at the top, smallest weight at the bottom). Double-click this file to open it in Excel.

Default Exemptions

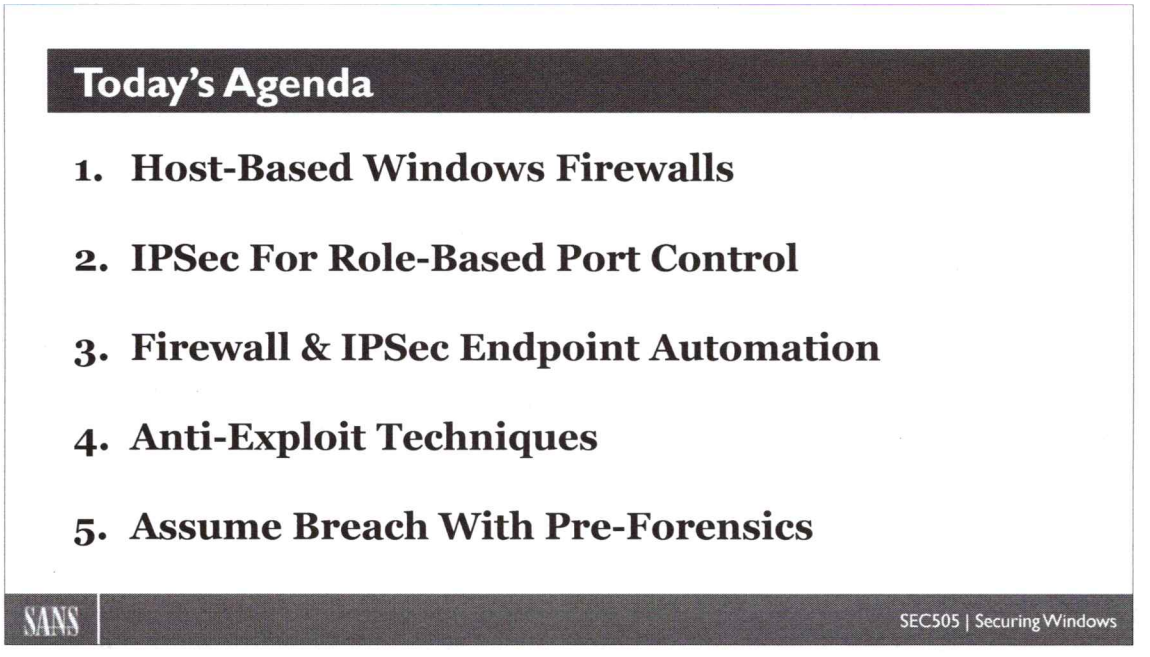
Some exemptions are built into the IPSec driver by default which causes the driver to simply ignore certain types of traffic, such as broadcast and multicast; however, a registry value can be set (NoDefaultExempt) which changes this behavior (KB811832, KB810207). Windows 2000 requires SP1 or later to enable this registry value, and note that SP4 and later *changes* this to a value of 1. Windows XP supports the value by default, but SP2 will automatically change it to 1. Windows Server 2003 and later support the value by default and it is set to 3 by default (create the value to change it to something else).

```
          Hive: HKEY_LOCAL_MACHINE
           Key: \SYSTEM\CurrentControlSet\Services\IPSEC
    Value Name: NoDefaultExempt
    Value Type: REG_DWORD
    Value Data: 0, 1, 2 or 3 (depending on operating system)

  Windows 2000: 0 or 1 possible with SP1, 0 by default, 1 by default with SP4.
    Windows XP: 0 or 1 possible, 0 by default, 1 by default with SP2.
  2003 and Later: 0, 1, 2 or 3 possible, 3 by default.

       Value 0: Multicast, broadcast, RSVP, Kerberos and IKE are exempt.
       Value 1: Multicast, broadcast and IKE are exempt.
       Value 2: RSVP, Kerberos and IKE are exempt.
       Value 3: Only IKE is exempt.
```

Note that Windows 2000/XP cannot be configured to *not* exempt broadcast or multicast traffic. And IKE can never be made not exempt too. If you wish to block IKE, it must be done with a firewall or some other filtering capability.

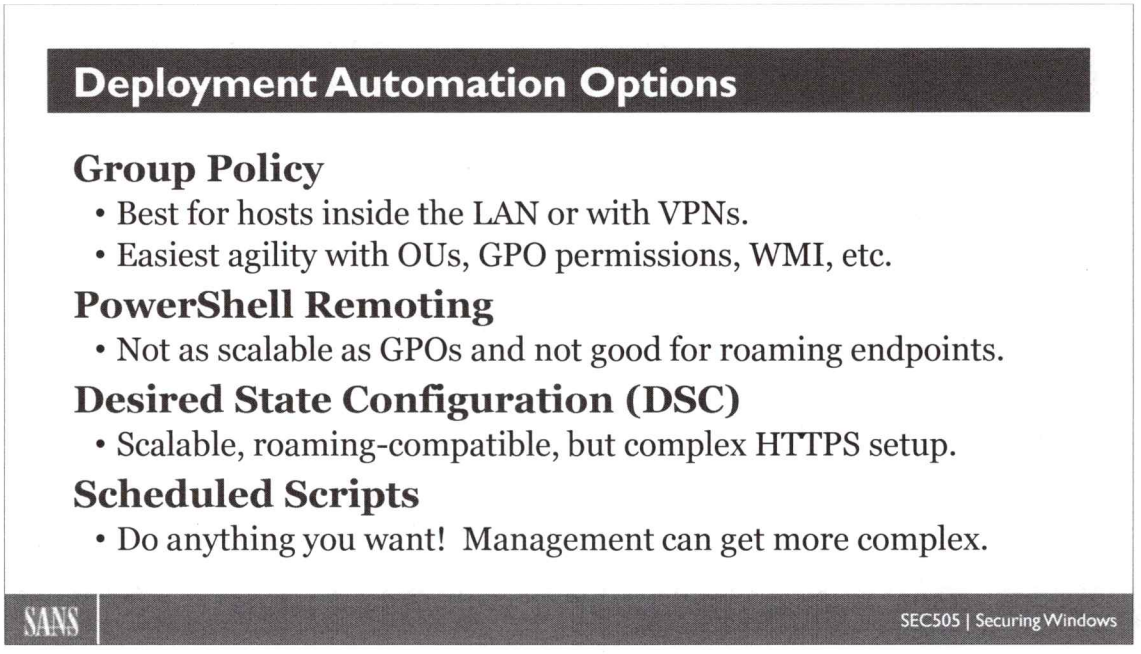
It is mandatory to set this value to 1 on 2000/XP and either 1 or 3 on Windows Server 2003 and later. This importance is highlighted by Microsoft's decision to set these values automatically with Service Packs despite the interoperability problems that may arise. The problem is that attackers can send malicious packets with any source port they wish, including the well-known port for Kerberos (UDP/TCP 88). Any defender relying on IPSec packet filtering could be compromised with this trick (perhaps using FPIPE.EXE from http://www.foundstone.com). It is also possible to send malicious broadcast packets, hence, the value should be set to 3 on Windows Server 2003 and later.

Once these more-secure values are set, keep in mind that your IPSec policies will have to take into account these once-exempted protocols when you wish to maintain functionality (KB254949, KB253169).



Today's Agenda

The wizards and graphical tools that Microsoft provides for creating individual IPSec or firewall rules are nice, but they don't help you when you need to configure these rules on thousands of systems. In the next section we'll talk about how to manage firewall and IPSec rules through Group Policy and PowerShell.



Deployment Automation Options

There are many options for deploying and managing firewall and IPSec rules across thousands of endpoints and servers. Each option has advantages and disadvantages.

Group Policy

Group Policy is best for servers and endpoints inside the LAN or which establish permanent or regular VPN tunnels back into the LAN. Servers which are hosted on cloud provider networks can also access their domain controllers.

Group Policy provides the most flexibility for the management of IPSec and firewall rules. Different GPOs can be assigned to different OUs, and GPOs can have permissions or WMI Filters to control which machines receive those GPOs.

But Group Policy only works with domain-joined machines. If you use a tool like LGPO.EXE to apply a GPO to a stand-alone computer, then it might be easier to just run a configuration script instead.

PowerShell Remoting

Remoting works on both stand-alone and domain-joined targets, but those targets must be accessible over the network, which makes this option less ideal for roaming devices on the Internet. Using remoting this way is similar to push-mode DSC, with all the pluses and minuses for scalability this entails. Nonetheless, remoting is still a great way to manage internal or cloud-hosted machines, such as servers.

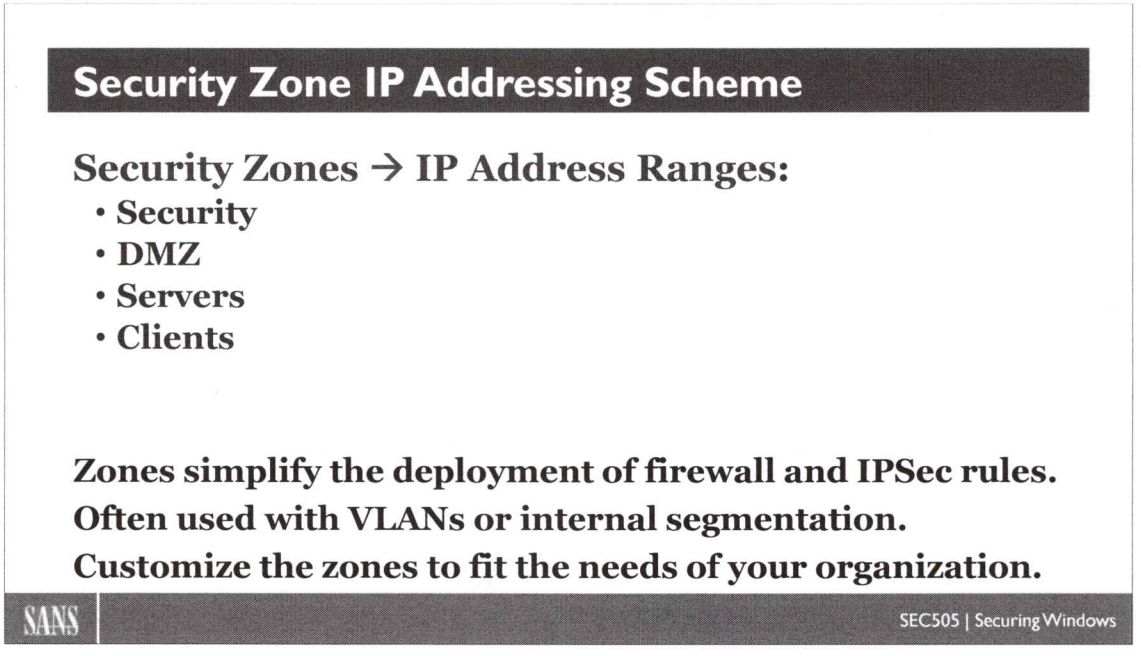
Desired State Configuration (DSC)

DSC can work in both push mode and HTTPS pull mode. Push mode is similar to generic PowerShell remoting, but DSC configuration functions might be easier to write than your own firewall and IPSec scripts (this is debatable). HTTPS pull mode has the advantage of scalability and roaming-endpoint compatibility, but is more complex to set up and maintain for large numbers of endpoints. A Microsoft or third-party graphical wrapper or management solution on top of DSC can very much help, especially when combined with a Mobile Device Management (MDM) protocol solution.

Scheduled Scripts

Scheduled scripts can run other scripts which manage IPSec and firewall rules. Scheduled jobs can be managed through Group Policy, PowerShell remoting, SCHTASKS.EXE, Desired State Configuration (DSC), and many other enterprise configuration management solutions.

Your own scheduled scripts allows for potentially great scalability and flexibility. If you can think it up, and you've got an enterprise-scale method of managing scheduled jobs, then you can probably do it! But with all this roll-your-own agility comes complexity. So, with scheduled jobs and PowerShell scripts, you can get whatever you want, but you'll have to design, deploy and maintain that solution yourself too.



Security Zone IP Addressing Scheme

Firewalling internal traffic is greatly simplified if different IP address ranges are assigned to different internal security zones. Not all computers are equally important or valuable from a security point of view. A "security zone" is a range of IP addresses allocated to a set of computers which have similar security requirements, usually because these servers and/or workstations have similar roles in the organization. Keep in mind that the following are not hard rules to be blinded followed, but general strategies to implement.

Typical Zones

Your environment will be unique, so your security zones may need to be customized, but the following security zones are typical:

- Security
- DMZ (or Extranet)
- Servers
- Clients
- Internet

The **Security Zone** includes everything which enforces or manages security for the internal network, such as domain controllers, RADIUS servers, SIEM consoles, IDS sensors, IDS consoles, administrator workstations, jump servers, log consolidation servers, and so on. These are the highest of your high-value targets. The Security Zone is often associated with special "shadow" segments or VLANs as well, and might even correspond to an entire separate AD forest for some machines like administrative workstations. Instead of a single Security Zone, you might subdivide this into special-purpose zones for monitoring, penetration testing, SIEM, domain controllers, etc.

The **DMZ (or Extranet Zone)** includes the servers exposed to the Internet, plus any associated servers or workstations which are not exclusively part of another Zone. If you have no perimeter firewalls, such as at a university, then this zone doesn't exist.

The **Servers Zone** includes all internal servers, though there will likely be some overlap with some of the DMZ and Security servers, which is fine, just allocate IP addresses in a way which makes sense for your environment. Servers include your Internet cloud-hosted VMs which are directly accessible from the LAN through a site-to-site VPN or other private tunnel into the Internet cloud provider's network.

The **Clients Zone** includes all internal workstations, laptops, tablets and smart phones, plus any remote clients with a VPN or DirectAccess connection into the LAN.

The **Internet Zone** is the rest of the world, including your own servers which are accessed over the public Internet, such as your VMs hosted by cloud providers but without a site-to-site VPN or other private tunnel.

Per-Zone IP Addressing

At each site, each zone should be assigned one or more IP address ranges. It is nice if the same subnet range for each zone is used at every site, but this isn't required, it's just easier to manage and understand. If you also wish to implement a segmentation, VLAN, firewalling or tunneling scheme on top of the IP addressing scheme, that can be handy too, but from the perspective of host-based firewalls such details are usually invisible.

The machines in the Security, DMZ and Servers Zones are typically assigned static IP addresses (or at least DHCP reservations) and the other devices in the Clients Zone will use DHCP and/or IPv6 network IDs advertised by your routers.

As an example, let's assume your organization uses 10.0.0.0/8 as the enterprise-wide network ID (we'll ignore IPv6 for now, but with its much larger addressing space, implementing the various per-site zones is actually even easier than with IPv4).

> **Note:** These examples are <u>not</u> optimized for CIDR route aggregation or conservation of scarce IP addresses, they are intended to be easy to understand for attendees who don't work at Layer 3 much and to make the management of firewall and IPSec rules simpler.

Each of your 200 sites might be assigned a different number in the second octet:

- Dallas LAN: 10.**1**.0.0/16
- DC LAN: 10.**2**.0.0/16
- San Diego LAN: 10.**3**.0.0/16
- Amsterdam LAN: 10.**4**.0.0/16

Within a site, the Security Zone might always have 1 through 5 as the third octet:

- Dallas Security: 10.1.**1-5**.0/24

- DC Security: 10.2.**1-5**.0/24
- San Diego Security: 10.3.**1-5**.0/24
- Amsterdam Security: 10.4.**1-5**.0/24

Within a site, the DMZ or Extranet Zone might always get 6-10 as the third octet:

- Dallas DMZ: 10.1.**6-10**.0/24
- DC DMZ: 10.2.**6-10**.0/24
- San Diego DMZ: 10.3.**6-10**.0/24
- Amsterdam DMZ: 10.4.**6-10**.0/24

Within a site, the Servers Zone might get 11-20 as the third octet:

- Dallas Servers: 10.1.**11-20**.0/24
- DC Servers: 10.2.**11-20**.0/24
- San Diego Servers: 10.3.**11-20**.0/24
- Amsterdam Servers: 10.4.**11-20**.0/24

Within a site, the Clients Zone might get everything else to accommodate for growth:

- Dallas Clients: 10.1.**21-255**.0/24
- DC Clients: 10.2.**21-255**.0/24
- San Diego Clients: 10.3.**21-255**.0/24
- Amsterdam Clients: 10.4.**21-255**.0/24

The important thing is not the number of bits in the CIDR mask, the names of the zones, the number of special-purpose zones, etc., the important thing is the overall strategy of associating IP address ranges with different types of devices based on their roles and their relative importance for security. You might begin with a Client Zone and the Everything-Else Zone and then see what makes sense from there. Again, it's the strategy we're interested in here, not the particular example details, because every attendees' network will be different.

Per-Zone Firewall Rules and IPSec Policies

The benefit of using per-zone IP addressing is that it will greatly simplify the management of firewall rules and IPSec policies. And not just host-based firewalls, your perimeter and internal firewall devices can also leverage these zones. Your proxy servers, IDS sensors and other network devices can benefit as well. Using Group Policy preferences with item-level targeting, different GPO policies can be applied to different Zones based just on IP address (though using GPO permissions is probably the better approach). Investing in a rational IP addressing scheme will pay off in many ways.

Consider this: do your workstations, laptops and tablets typically need to communicate directly with each other? Other than VoIP, they typically do not. So a very effective host-based firewall strategy is to have every computer in the Clients Zone block all the

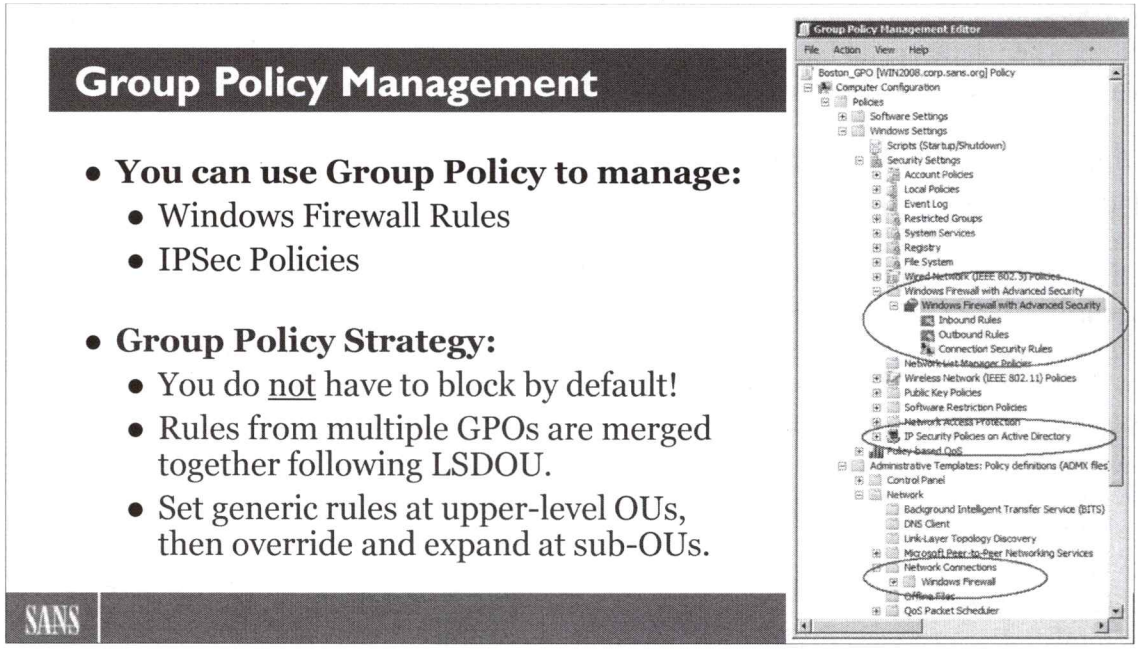
unnecessary inbound and outbound packets to every other IP address in the Clients Zone. Now, when malware and hackers take over an initial soft target in the Clients Zone, it will be more difficult for them to directly attack other client computers.

Mapping Zones to Organizational Units

Ideally, the roles a computer plays will determine its security zone and IP address, and these roles might also determine that computer's organizational unit (OU) in Active Directory. The same logic applies to both design strategies: we need to apply different security policies based on the role(s) of the computer, so some security policies affect the physical placement, Ethernet switch, VLAN, host-based firewall rules and IPSec settings of the computer (OSI Layers 1-4) while other policies affect the computer's operating system, applications and users (OSI Layers 5+).

Security zones also affect where we allow administrators and other high-value accounts to authenticate. Domain Admin users might be allowed to authenticate interactively or over the network to servers in the Security Zone, but not to the Servers Zone and certainly not the malware-infested Clients Zone. Each zone, then, might have separate administrative accounts in order to contain the harm from stolen credentials and token abuse attacks. How do we control such authentication traffic? Through Group Policy, and these GPOs are typically assigned to OUs, hence we see the possible mapping between zones and OUs again.

None of these recommendations are absolute, you will need to customize and fine-tune them for your environment, but we do want to scale up our security work to the entire enterprise while keeping a lid on the growth of complexity this entails.



Group Policy Management

How do you scale out management of IPSec and firewall policies to thousands of computers? Group Policy of course! And for your stand-alone systems, you can write a batch script to manage IPSec and firewall policies on them too. Not only Windows Vista and later, but also Windows XP/2003 can have their IPSec and firewall settings managed through Group Policy and/or custom scripts.

Export and Import Policies

IPSec policies can be exported and imported. On Vista and later, the IPSec and firewall policies are exported/imported together in one file. In Windows 2000/XP/2003, right-click the "IP Security Policies" MMC snap-in > All Tasks > Import/Export Policies. In Windows Vista/2008 and later, right-click the "Windows Firewall" snap-in > Import/Export Policy. In either case, you are saving or restoring settings from a file.

The ability to export and import policy files is important because you'll want to test your IPSec and firewall policies in an isolated lab first, then you'll later import them into a GPO for pilot testing and then eventually into another GPO for production deployment. It is also a good idea to keep snapshots of your various IPSec and firewall policies for recovery and audit reference.

Group Policy Object Paths

Windows Firewall settings for Server 2008/Vista and later are located in a GPO under Computer Configuration > Policies > Windows Settings > Security Settings > Windows Firewall With Advanced Security.

Windows Firewall settings for Windows XP/2003 are located in a GPO under Computer Configuration > Policies > Administrative Templates > Networks > Network Connections > Windows Firewall.

IPSec policies for Windows 2000/XP/2003 computers are located in a GPO under Computer Configuration > Policies > Windows Settings > Security Settings > IP Security Policies on Active Directory.

Note that all IPSec policies created in Active Directory for Windows 2000/XP/2003 will be visible in the GPO, but only one of these IPSec Policies can be activated or "assigned" at a time (notice the Assigned column). It is not the case that all visible policies will be assigned like in Vista/2008 and later.

Assigning Multiple Policies Via GPO

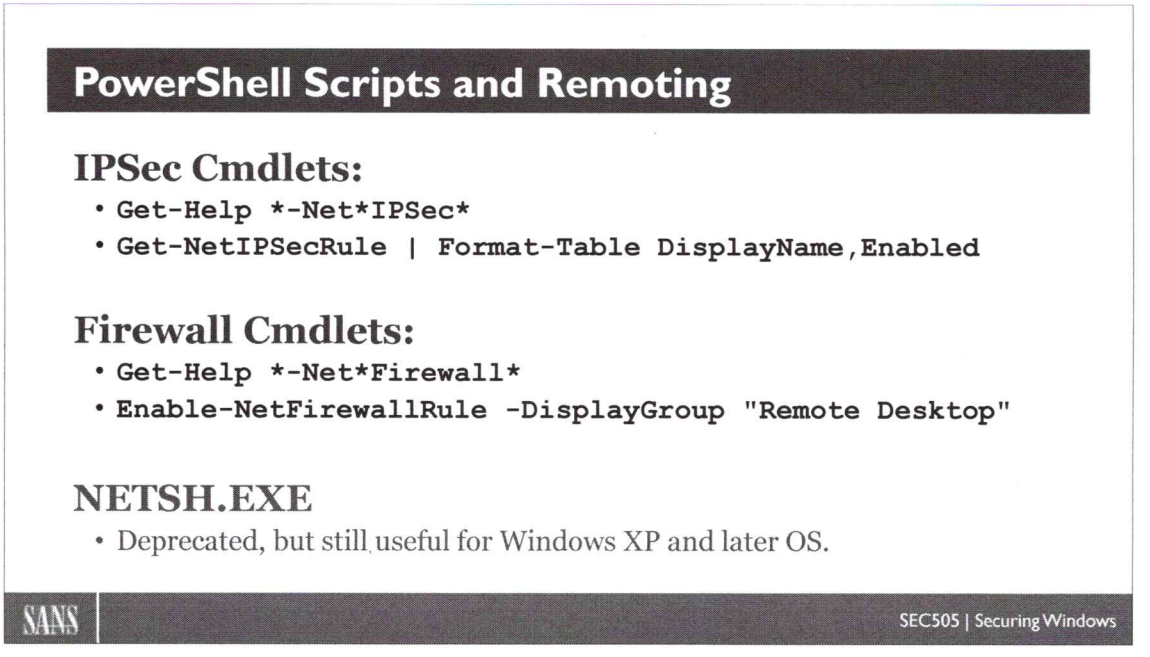
Recall that GPOs are applied in the following order: local, site, domain, and OUs from outermost to innermost (mnemonic: LSD-OU). On Windows 2000/XP/2003, the last GPO to be applied which assigns an IPSec Policy will replace and override any other IPSec policies earlier in the GPO precedence order. So, an IPSec policy assigned at the OU level will supplant any IPSec policy assigned at the domain, which will replace any IPSec policy assigned at the site, which replaces the local, and so on. IPSec policies assigned to the site, domain or OU are not merged on Windows 2000/XP/2003: the last IPSec policy assigned while GPOs are being processed is the one and only IPSec policy that becomes effective. There are no conflicts because only one IPSec policy wins in the end, i.e., the last one applied wins. And if a local IPSec policy was assigned to a machine, this local policy will be overwritten by the IPSec policy assigned from Group Policy too.

If you use Group Policy to push out an IPSec policy to Windows 2000/XP/2003 and also Windows Vista or later, that policy is accepted and enforced on all operating systems. Windows Vista and later are backwards compatible with the older IPSec policies. But, on Windows Vista and later, any additional IPSec policies configured through WFAS are also accepted and enforced at the same time. If there is a conflict, the older Windows 2000/XP/2003 IPSec policies are evaluated first and then the WFAS IPSec Connection Security Rules are evaluated afterwards.

> **Note:** On way or another, though, if a Windows Firewall rule requires certain traffic to be secured with IPSec, it still has to be secured, but it doesn't matter what triggered the IPSec negotiation process (an older Windows 2000/XP/2003 IPSec policy, a WFAS Connection Security Rule, or the firewall itself). And as a kicker, remember that dynamic mode IPSec settings are evaluated before any of the static mode settings!

Another difference to note is that WFAS IPSec policies inherited from multiple GPOs are *all* accepted, combined and enforced. In Windows 2000/XP/2003, the last IPSec policy assigned through Group Policy is the only IPSec policy which gets enforced. This is no longer the case in Vista or later. If multiple GPOs apply to a Windows Vista box, for

example, and each GPO has different WFAS IPSec quick mode settings, then all the applicable quick mode IPSec settings from all the inherited GPOs are combined and enforced. If there are conflicts, then GPOs processed later will override conflicting settings applied earlier. However, this merging of IPSec quick mode (Phase II) settings does not apply to the main mode (Phase I) settings. You can only have one main mode policy, and the last GPO to assign a main mode policy is the winner (just like in Windows 2000/XP/2003). Hence, standardize on one set of main mode configuration settings and use them everywhere on every operating system.



PowerShell Scripts and Remoting

Both IPSec and firewall settings can be scripted from the command line. These tools work on local or remote systems, as long as you are a member of the local Administrators group, and they work on both domain members and stand-alones. While Group Policy is the primary tool for managing firewall and IPSec settings, the ability to script these settings opens new possibilities.

There are different tools for different operating systems. For managing IPSec settings:

- PowerShell 3.0 or later: Over 200 cmdlets (get-help *-net*)
- Windows 2000: IPSECPOL.EXE
- Windows XP: IPSECCMD.EXE
- Windows 2003/Vista/2008/7 and later: NETSH.EXE

For managing the firewall on Windows XP and later, use only PowerShell or NETSH.EXE. In Windows XP/2003, the NETSH.EXE context is "firewall", while in Windows Vista and later the context is named "advfirewall" (execute "netsh.exe /?" to see these contexts). The NETSH.EXE program is built into Windows 2000 and later by default, but it has different capabilities on different OS versions.

To use PowerShell, you must have PowerShell 3.0 or later. There are over 200 cmdlets related to networking, IPSec and the firewall, so they can't possibly all be discussed here.

To see a listing of the cmdlets for IPSec, firewall rules, network interfaces, TCP/IP, etc.:

```
Get-Help *-net*
```

```
# Multiple wildcards can be used to narrow the output list.

Get-Help *-net*ipsec*
Get-Help *-net*firewall*
Get-Help *-net*ip*

# Get the full help for a cmdlet with the -Full switch.

Get-Help Set-Netipaddress -full
```

IPSECPOL.EXE is intended for Windows 2000 only, while IPSECCMD.EXE is only for Windows XP. The command-line switches used by the tools are almost identical. The tool was last seen as a part of the Windows XP Server Pack 2 and later Support Tools.

Dynamic Mode vs. Static Mode

The IPSec command-line tools operate in two modes: dynamic mode and static mode. When used in dynamic mode, they inject new rules directly into the IPSec driver's in-memory database of IPSec rules. Dynamic rules take effect immediately, but they do not show up in any graphical tool, hence, they can only be managed from the command line. When used to create static mode policies, these tools can create visible named policies that are permanently stored in the registry or in Active Directory. Which mode is being used depends on the command-line switches used of course.

Scheduled Scripts, Per-User Rules, and Other Creative Uses

A batch script with the desired IPSec or firewall rules could be scheduled to run every hour. Because these tools can take a command-line argument of a remote computer, domain member or stand-alone, this single script could configure hundreds of systems from a central location. Even more flexibility is available when using PowerShell or VBScript instead of plain batch files.

IPSec policies are assigned to computers, not users. However, Group Policy can be used to define logon/logoff scripts for users. These scripts could create dynamic IPSec rules when the user logs on, then delete these dynamic rules when the user logs off again. Hence, you can implement per-user IPSec settings with logon scripts for users who are local Administrators wherever they log on.

NETSH.EXE Examples

NETSH.EXE can be used on Windows XP and later to manage firewall settings, and on Windows Server 2003 and later to manage IPSec settings. Until every machine is running PowerShell 3.0 or later, you might prefer this binary for a while even though it has been deprecated by Microsoft. Here are some commands with which to experiment.

To see a summary of your profile options:

```
netsh.exe advfirewall show allprofiles
```

To dump the details of every connection security rule (IPSec):

```
netsh.exe advfirewall consec show rule name = all
```

To see how to create a connection security rule (IPSec):

```
netsh.exe advfirewall consec add rule /?
```

To dump the details of every firewall rule:

```
netsh.exe advfirewall firewall show rule name = all
```

To see how to create a firewall rule from the command line:

```
netsh.exe advfirewall firewall set rule /?
```

The following is an example of a batch file that could be used to create a static IPSec policy object on Windows Server 2003 or later. This policy would block all packets except for TCP 80/443, and it would require AH for all traffic to/from network ID 10.0.0.0 (note that many of the lines must be wrapped).

```
REM ************************************************************
REM Create The Ipsec Policy Object.
REM ************************************************************

netsh.exe ipsec static add policy name="IIS_Server_Policy"
   assign=no


REM Create Filter Lists And Add Filters To Them.
netsh.exe ipsec static add filterlist name="HTTP_Traffic"

netsh.exe ipsec static add filter filterlist="HTTP_Traffic"
   srcaddr=any dstaddr=me description="HTTP" protocol=TCP
   srcport=0 dstport=80

netsh.exe ipsec static add filter filterlist="HTTP_Traffic"
   srcaddr=any dstaddr=me description="HTTPS" protocol=TCP
   srcport=0 dstport=443

netsh.exe ipsec static add filterlist name="All_Traffic"

netsh.exe ipsec static add filter filterlist="All_Traffic"
   srcaddr=any dstaddr=me description="All Traffic"
   protocol=any srcport=0 dstport=0

netsh.exe ipsec static add filterlist name="Internal_Traffic"
```

```
netsh.exe ipsec static add filter filterlist="Internal_Traffic"
      srcaddr=10.0.0.0 srcmask=255.0.0.0 dstaddr=me
      description="Internal Traffic" protocol=any srcport=0
      dstport=0


REM ************************************************************
REM Define Filter Actions.
REM ************************************************************

netsh.exe ipsec static add filteraction name="Allow"
   action=permit

netsh.exe ipsec static add filteraction name="Block" action=block

netsh.exe ipsec static add filteraction name="AH_Only" qmpfs=yes
   soft=no inpass=yes action=negotiate
   qmsec="AH[MD5]:100000k/1000s AH[SHA1]:100000k/1000s"


REM ************************************************************
REM Now Create Rules In The Policy With The Actions And Filters.
REM ************************************************************

netsh.exe ipsec static add rule name="Allow HTTP"
   policy="IIS_Server_Policy" filterlist="HTTP_Traffic"
   kerberos=yes filteraction=Allow

netsh.exe ipsec static add rule name="Block All"
   policy="IIS_Server_Policy" filterlist="All_Traffic"
   kerberos=yes filteraction=Block

netsh.exe ipsec static add rule name="AH for LAN"
   policy="IIS_Server_Policy" filterlist="Internal_Traffic"
   psk="myPreSharedKey" filteraction=AH_Only

REM Disable The Built-In Default Response Rule.
netsh.exe ipsec static set defaultrule policy="IIS_Server_Policy"
   activate=no


REM ************************************************************
REM Now Assign The Policy.
REM ************************************************************

netsh.exe ipsec static set policy name="IIS_Server_Policy"
   assign=yes
```

On Your Computer

This exercise has multiple parts.

> **Note:** Many of these examples require Server 2012 R2, Windows 8.1 or later, but networking settings on earlier OS versions can be scripted too, just not as easily.

Configure Windows Firewall

Enable the firewall for public interfaces, block inbound connections by default, enable logging of blocked packets, and set the log size to 4096 KB:

```
Set-NetFireWallProfile -Name Public -Enabled True
  -DefaultInboundAction Block -LogBlocked True
  -LogMaxSizeKilobytes 4096
```

Enable all the firewall rules for the "File and Printer Sharing" group (in en-US culture):

```
Enable-NetFireWallRule -Displaygroup "File and Printer Sharing"
```

> **Note:** A "security filter" for a firewall rule is just all the details of that firewall rule collected into a single object to make these details easier to manage.

> **Note:** *Please* don't forget about using tab completion!

Enable the inbound "Remote Desktop" group of rules and require IPSec authentication and encryption for them:

```
Get-NetFireWallRule -DisplayGroup "Remote Desktop"
  -Direction Inbound | Enable-NetFireWallRule


Get-NetFireWallRule -Displaygroup "Remote Desktop"
  -Direction inbound | Get-NetFireWallSecurityFilter |
  Set-NetFireWallSecurityFilter -Authentication required
   -Encryption required
```

Create an inbound firewall rule to block access to TCP port 3666:

```
New-NetFireWallRule -Displayname "Drop APT Back Door"
  -Direction inbound -LocalPort 3666 -Protocol tcp -Action block
```

Note: This new rule can now be seen in the graphical Windows Firewall snap-in.

Delete a firewall rule by its display name:

```
Remove-NetFireWallRule -DisplayName "Drop APT Back Door"
```

Configure IPSec Rules

View the list of IPSec-related cmdlets:

```
Get-Help *ipsec*
```

Please switch to the C:\SANS\Day5-IPSec\IPSec folder:

```
cd C:\SANS\Day5-IPSec\IPSec
```

Use ISE to open the Add-IPSec-Rule.ps1 script and browse its contents:

```
ise .\Add-IPSec-Rule.ps1
```

Run the script to create a new IPSec rule (will be visible in the snap-in after a refresh):

```
.\Add-IPSec-Rule.ps1
```

Display the local IPSec rules:

```
Get-NetIPsecRule | Select-Object DisplayName,Enabled
```

Group Policy

Export and import firewall (and IPSec) rules to a backup file with NETSH.EXE:

```
netsh.exe advfirewall export c:\temp\firewall.wfw

netsh.exe advfirewall import c:\temp\firewall.wfw
```

Reset firewall and IPSec rules back to their factory defaults with NETSH.EXE:

```
netsh.exe advfirewall reset
```

Let's import the firewall rules into a Group Policy. Please open the Group Policy Management tool and edit the Boston_GPO.

> **Note:** The Boston_GPO was created in an earlier lab in this course. If you don't have it, just create a new GPO with the same name and link it to the Boston OU.

Inside the GPO, navigate down to Computer Configuration > Policies > Windows Settings > Security Settings > Windows Firewall with Advanced Security. Next, right-click on the Windows Firewall tool in the GPO > Import Policy > Yes button > select the C:\Temp\firewall.wfw file > Open > OK.

Generate an HTML report of the settings inside the Boston_GPO:

```
Get-GpoReport -Name "Boston_GPO" -ReportType html
  -Path c:\temp\gp.html

c:\temp\gp.html    #Will open the HTML file in a browser
```

Create Firewall Rules To Block IP Address Ranges

Please switch to the C:\SANS\Day5-IPSec\Firewall folder:

```
cd C:\SANS\Day5-IPSec\Firewall
```

View an example file of IP address ranges whose traffic we wish to block:

```
ise .\Country-BlockList.txt
```

View the help for a script which can create blocking rules from an input file:

```
Get-Help -Full .\Import-Firewall-Blocklist.ps1
```

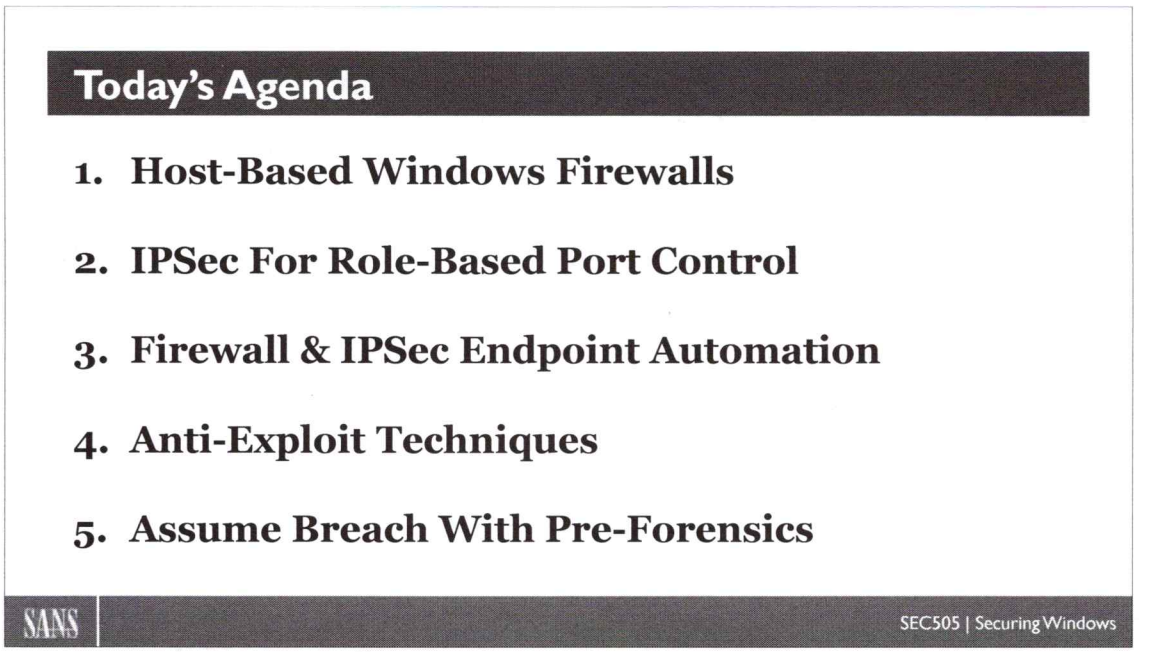
Create firewall rules to block all traffic to/from the IP addresses in the input file:

```
.\Import-Firewall-Blocklist.ps1 -InputFile
   .\Country-BlockList.txt
```

In the graphical Windows Firewall tool, refresh the lists of inbound and outbound rules. You will see new rules have been created with names like "Country-Blocklist-#XXX". In the properties of any of these rules on the Scope tab you will see the IP address ranges blocked by that rule.

Delete the firewall rules created by the script, but leave all other rules alone:

```
.\Import-Firewall-Blocklist.ps1 -InputFile
   .\Country-BlockList.txt -DeleteOnly
```

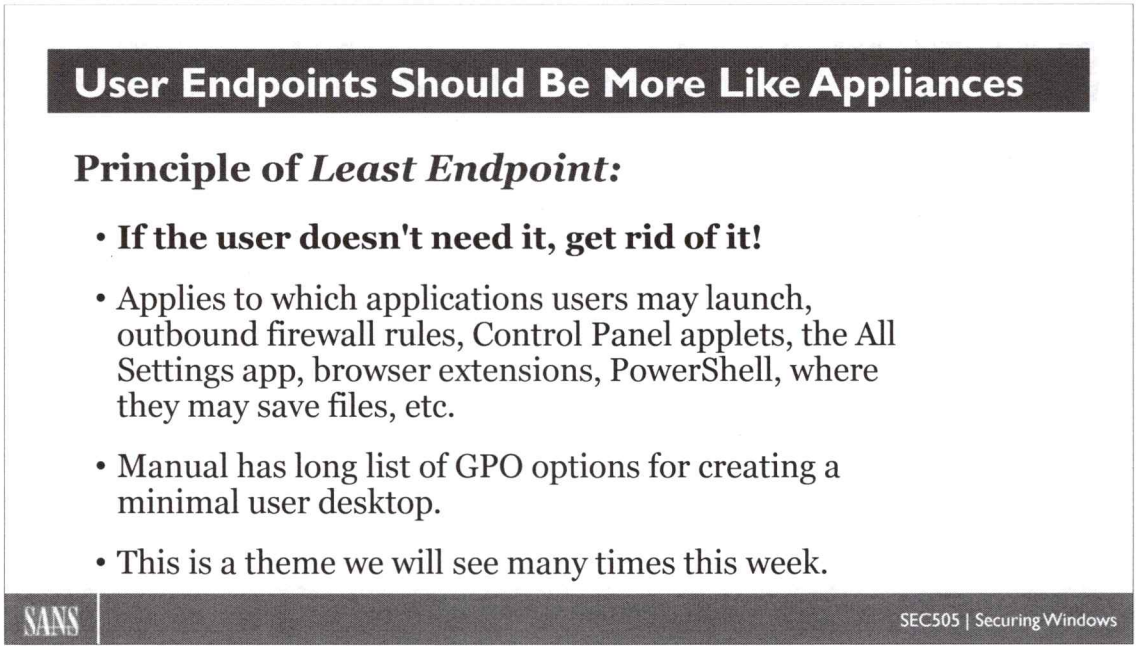


Today's Agenda

Our endpoints are fully patched. Users have no administrative privileges on their machines. What else can we do to harden our endpoints against malware exploitation?

In this section we will talk about how to make our endpoints more like minimal appliances. Part of this work is to deploy application whitelisting to control which processes users are permitted to launch. There are many whitelisting products on the market, but we will use the free, built-in solution from Microsoft: AppLocker. AppLocker can be managed through Group Policy and PowerShell.

We will also deploy a benevolent rootkit from Microsoft named EMET. EMET integrates into running applications to make these processes more resistant to compromise. EMET can also be managed through Group Policy and PowerShell.



User Endpoints Should Be More Like Appliances

Users view their Windows desktops, laptops, tablets and phones as extensions of their personalities. When they discover the Control Panel, they feel quite free to experiment with the applets in it, thus raising the Total Cost of Ownership of their Windows machines as they mess things up ("*I didn't touch a thing!*"). But this desktop-is-an-extension-of-my-Self attitude also extends to less innocuous behaviors like untrusted software installation, "poking around" the hard drives and the network, trying to get around the "annoying" security measures you've deployed, etc. These behaviors also introduce malware into the environment.

This pattern will only get worse with the BYOD trend since users will own the physical devices themselves. Yet we have to accommodate this trend somehow. What we need is to control the user's desktop (or what appears to be their desktop) on all their devices no matter who owns the hardware.

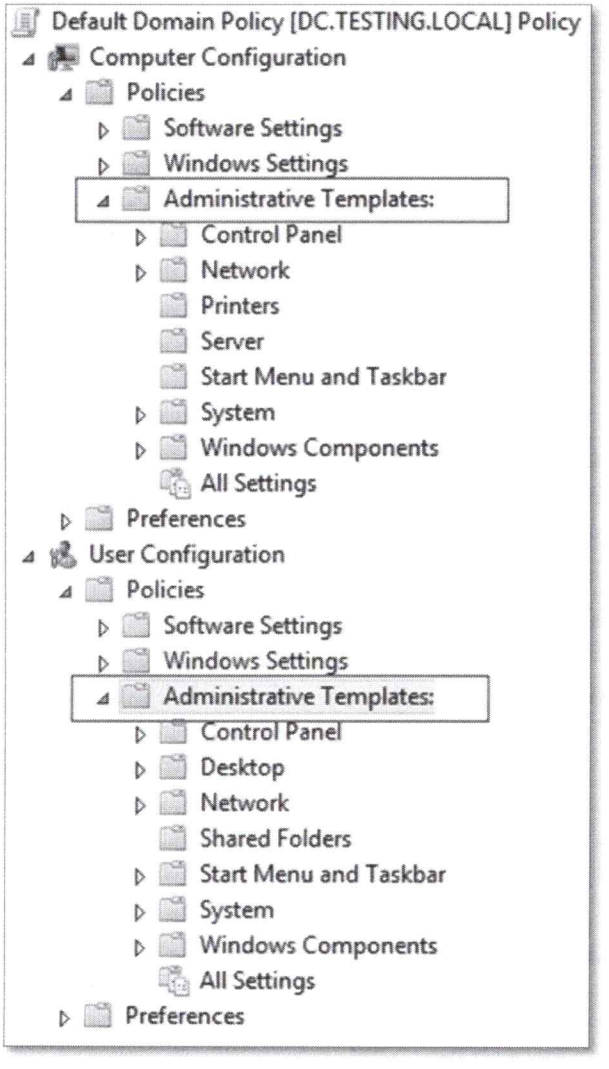
If you can get away with it politically, consider rolling out managed desktops instead of the omni-purpose desktops that install by default. A managed desktop follows the principle of least privilege to its logical extreme: deny *everything* not absolutely required for the user to get his or her work done. Through Group Policy you can transform the desktop into something unrecognizable, into something more like an *appliance* which allows access to just the few applications and network locations the user really needs.

Of course, this assumes you've gotten users out of the local Administrators group on their computers. If a user has administrative control over their own computer, you don't really manage it, they *let you* manage it. Through Group Policy, your patch/application management system, and the free Microsoft Application Compatibility Toolkit, it really

isn't necessary for the vast majority of users to be local Administrators group members anymore.

Group Policy Desktop Settings

We can use Group Policy to make our users' desktops more like managed appliances. There are hundreds of settings which can restrict users' behavior in useful ways. Many of the more interesting settings are listed below, and of course we already discussed other settings like AppLocker and Windows Firewall rules. The GPO settings listed below are found under the Administrative Templates container unless otherwise specified.



A few of the items below need some explanation:

- "Custom user interface" -- The first program which launches when someone logs on is their user interface process. By default, it is EXPLORER.EXE, which creates the taskbar, the Start Menu, and icons on the desktop. You don't have to

have these. The "interface" program could be any program you wish, e.g., NOTEPAD.EXE, CMD.EXE, Remote Desktop Services client in full-screen mode, Microsoft Outlook with a "digital dashboard", a Microsoft Access form, a simple MMC console with three icons for the three applications users are permitted to run (see screenshot below), etc. When combined with the "Disable Task Manager" and "Disable The Command Prompt" options, a custom user interface can help reduce user mischief.

- "Hide these specified drives in My Computer" and "Prevent access to drives in My Computer" -- The hide version will not show the particular hidden drive letters in My Computer, File Explorer and in Open File dialog boxes. But users can still access files in hidden drives if the path is manually entered in the application, Run line or CMD.EXE window. Hence, make sure to disable the Run line and CMD.EXE windows when merely hiding. Hiding drive letters is mainly intended to prevent casual snooping. The prevent access version of this option, on the other hand, still shows the restricted drive letters, but any access to those drives is denied. This is the version to use when regulating access to flash drives, CD-ROM drives, and the boot partition. Both versions can be used together, and the My Documents folder can be redirected to the user's home folder share.

GPO > Computer or User > Policies > Administrative Templates >

Windows Components > Internet Explorer:

- Disable changing proxy settings
- Disable changing Automatic Configuration settings
- Disable changing ratings settings
- Disable changing certificate settings
- Disable AutoComplete for forms
- Do not allow AutoComplete to save passwords

Windows Components > Internet Explorer > Internet Control Panel:

- Disable the General page
- Disable the Security page
- Disable the Content page
- Disable the Connections page
- Disable the Programs page
- Disable the Advanced page

Windows Components > Internet Explorer > Browser Menus:

- File menu: Disable closing the browser and Explorer windows
- Tools menu: Disable Internet Options...menu option
- Disable 'Save this program to disk' option

Windows Components > Windows Explorer:

- Remove "Map Network Drive" and "Disconnect Network Drive"

- Hide these specified drives in My Computer
- Prevent access to drives from My Computer
- No "Computers Near Me" in My Network Places
- No "Entire Network" in My Network Places

Windows Components > Microsoft Management Console:

- Restrict the user from entering author mode
- Restrict users to the explicitly permitted list of snap-ins

Windows Components > Task Scheduler:

- Hide Property Pages [of tasks]
- Prevent Task Run or End
- Disable Drag-and-Drop [of .job files into the Tasks folder]
- Disable New Task Creation
- Disable Task Deletion
- Disable Advanced Menu
- Prohibit Browse [to schedule arbitrary programs or scripts]

Windows Components > Windows Installer:

- Disable media source for any install [prevents user-selection of MSI files]

Start Menu & Taskbar:

- Remove common program groups from Start Menu
- Remove Run menu from Start Menu
- Disable and remove the Shut Down command

Desktop:

- Hide all icons on desktop
- Remove My Documents icon from desktop
- Remove My Documents icon from Start Menu
- Hide My Network Places icon on desktop
- Hide Internet Explorer icon on desktop
- Prohibit user from changing My Documents path
- Don't save settings at exit

Desktop > Active Desktop:

- Enable Active Desktop
- Disable Active Desktop
- Disable all items
- Prohibit changes
- Prohibit closing items
- Add/Delete items
- Active Desktop Wallpaper

Desktop > Active Directory:

- Hide Active Directory folder [in My Network Places]

Control Panel:

- Disable Control Panel
- Hide specified Control Panel applets
- Show only specified Control Panel applets

Control Panel > Add/Remove Programs:

- Disable Add/Remove Programs
- Hide the "Add a program from CD-ROM or floppy disk" option

Control Panel > Display:

- Disable Display in Control Panel
- Hide Background tab
- Disable changing wallpaper
- Hide Appearance tab
- Hide Settings tab
- Hide Screen Saver tab
- Activate screen saver
- Screen saver executable name
- Password protect the screen saver
- Screen saver timeout

Network > Offline Files:

- Disable user configuration of Offline Files
- Synchronize all offline files before logging off
- Action on server disconnect
- Non-default server disconnect actions
- Disable 'Make Available Offline'
- Prevent use of Offline Files folder
- Administratively assigned offline files

Network > Network and Dial-Up Connections:

- Prohibit deletion of RAS connections
- Prohibit access to properties of a LAN connection
- Prohibit access to current user's RAS connection properties
- Prohibit access to properties of RAS connections available to all users
- Prohibit access to the Dial-Up Preferences item on the Advanced menu
- Prohibit access to the Advanced Settings item on the Advanced menu
- Prohibit configuration of connection sharing
- Prohibit TCP/IP advanced configuration

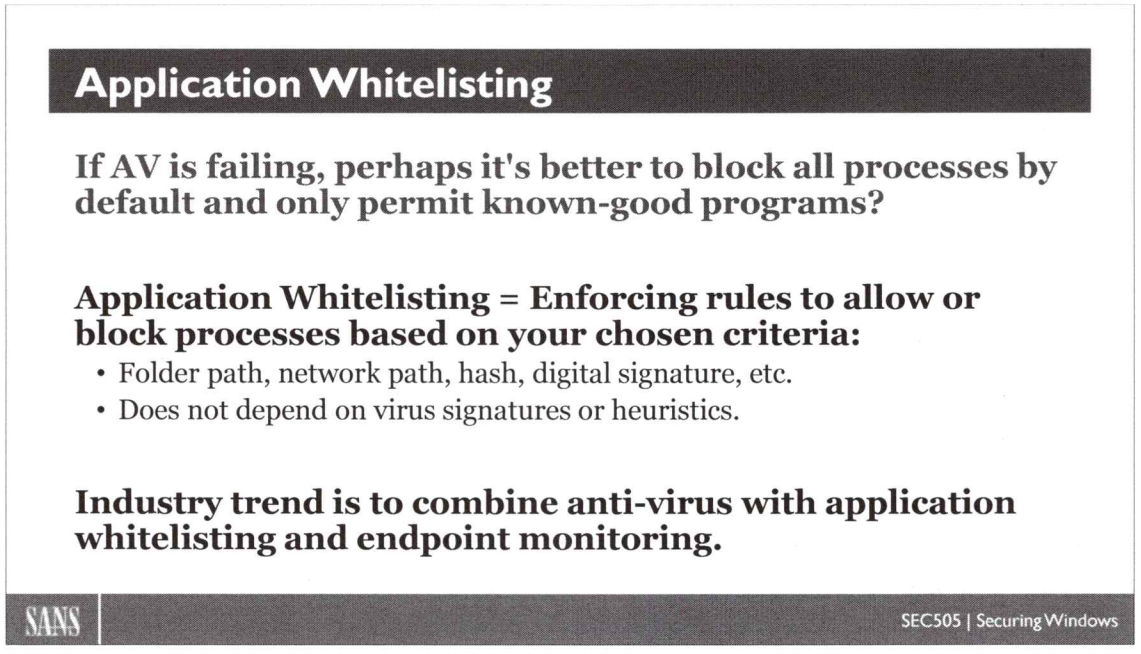
System:

- Custom user interface
- Disable the command prompt
- Disable registry editing tools
- Run only allowed Windows applications
- Don't run specified Windows applications
- Disable Autoplay [on CD-ROM or all drives]

System > Logon/Logoff:

- Disable Task Manager
- Disable Lock Computer
- Disable Change Password
- Disable Logoff
- Exclude directories in roaming profile
- Run these programs at user logon
- Disable the run once list
- Disable legacy run list

Note that the options above named "Run only allowed Windows applications" and "Don't run specified Windows applications" (in the System area) only take effect if the user's desktop interface is the default EXPLORER.EXE *and* it is through EXPLORER.EXE that the user is attempting to launch the program, i.e., through a Start Menu, task bar, or desktop icon. Task Manager, CMD.EXE and PowerShell do not enforce these rules. These settings might be used as a reinforcement for real application whitelisting products, such as Bit9 or AppLocker, but provide only modest restrictions on their own.



Application Whitelisting

If AV can't stop malware, the hope is that whitelisting will. Application whitelisting products enforce rules concerning which processes can and cannot run. In a strict environment, every process is blocked unless there is a rule which specifically allows it. In a permissive environment, all processes are allowed unless specifically blocked.

Whitelisting can be combined with anti-virus scanning in a single integrated product, and this is the direction of the industry right now. Because signature-based anti-virus scanners will never be able to keep up with the mutation rate, and because heuristic, reputation-based and other fuzzy anti-virus scanning techniques are not reliable enough yet, application whitelisting should be an important part of your arsenal. And because even AV plus whitelisting will eventually fail, we also need continuous monitoring of all the endpoints, i.e., host-based intrusion detection.

A few of the more popular application whitelisting solutions are the following:

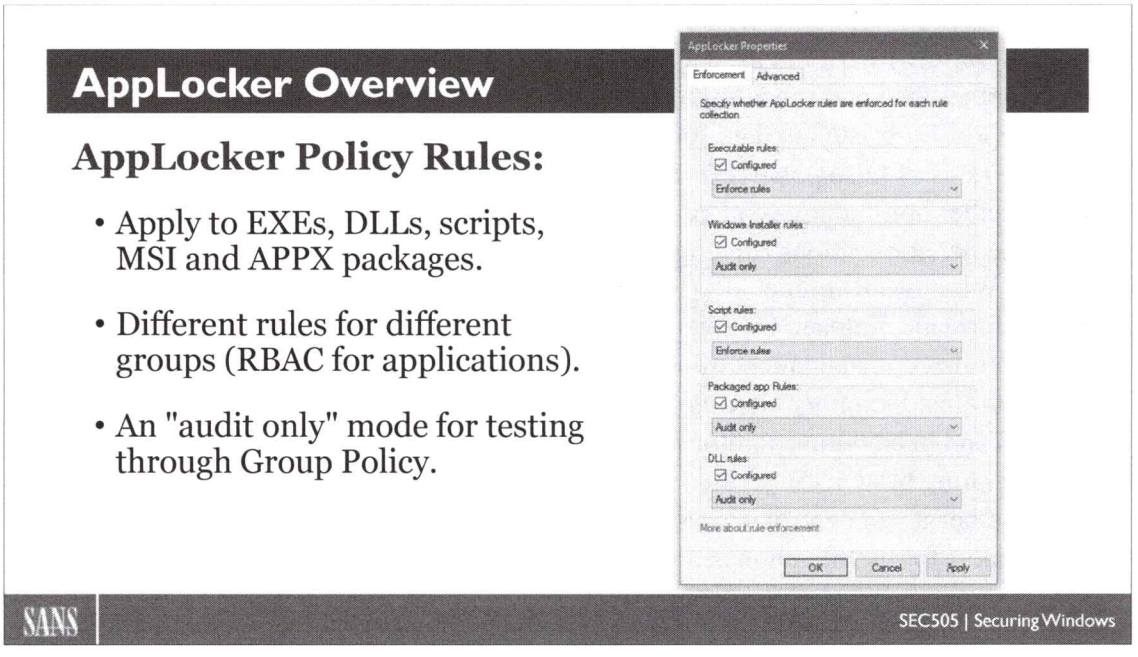
- Bit9 Security Platform (www.bit9.com)
- Lumension Application Control (www.lumension.com)
- McAfee Application Control (www.mcafee.com)
- Microsoft Software Restriction Policies and AppLocker (www.microsoft.com)
- SignaCert Enterprise Trust (www.signacert.com)

But how to choose which is best? Whitelisting products are mainly differentiated by the following features or lack thereof:

- Licensing fees, since these products can get expensive.

- Whether all processes are regulated or only the user-initiated processes.
- The variety of criteria used to define rules, such as hashes and digital signatures.
- Whether the vendor continuously supplies new signatures for popular products.
- Support for trusted update sources (WSUS, EMS, etc.) for auto-approval.
- The rates of false positive and false negative classifications.
- Whether non-binaries can be regulated too, such as scripts, macros and MSIs.
- Whether applications running inside a Java VM, the .NET Framework or top of the Windows Runtime API (Metro apps) are regulated.
- Performance impact, especially when integrated with anti-virus scanning.
- Centralized management of off-site devices which only connect infrequently.
- Centralized logging, alerting and custom reporting features.
- IDS and behavioral-monitoring features blended in, similar to AV heuristics.
- Operating systems supported besides Windows.
- Whether it can control access to removable devices too, such as flash drives.
- Support for mobile phones, Point of Sale (POS) terminals, and other devices.

Let's look at a built-in whitelisting technology called "AppLocker" and how to manage it through Group Policy and PowerShell.



AppLocker Overview

Application Control Policy (AppLocker) regulates what programs, scripts, modern APPX packages, and MSI installer packages users can run. AppLocker can allow/block applications based on digital signatures, version, file system path, network path, SHA-256 hash value. But be aware of its requirements, though, it doesn't work on XP, 2003, Windows 7 Professional, or even Windows 8.1 Pro. You must have the Enterprise version of Windows.

Requirements

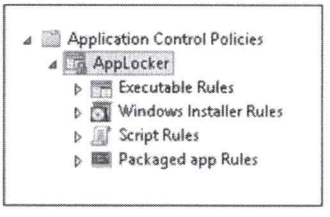
AppLocker has the following requirements for deployment:

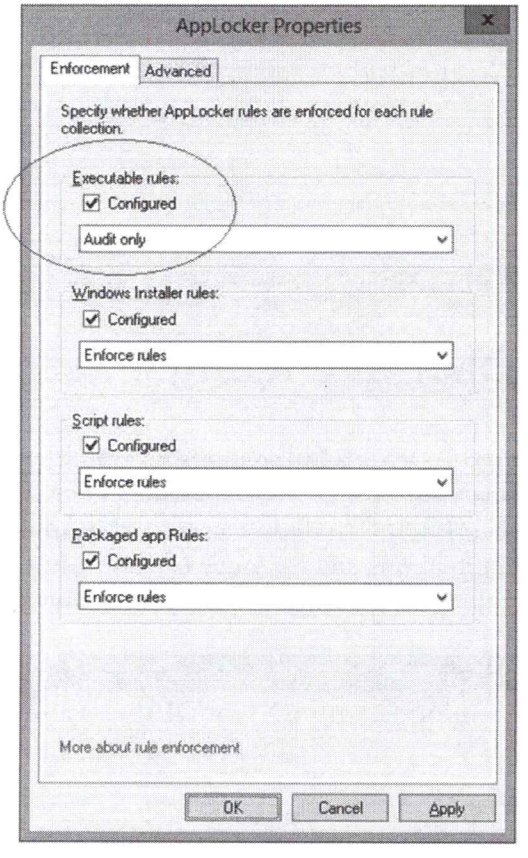
- AppLocker works on any edition of Windows Server 2008-R2, or later, but not on Windows Server 2008 or earlier.
- AppLocker works on Windows 7 Ultimate or Enterprise (but not on Professional), and Windows 8 Enterprise (but not on Pro), or later clients. There is no Ultimate version of Windows 8 or later.
- At least one domain controller must be running Windows Server 2008-R2 or later.
- The Application Identity service must be running on clients, so don't disable it, and consider setting the service to start automatically too.
- APPX package rules are only available on Windows 8 Enterprise and later.

Audit-Only Mode

When first creating an AppLocker policy, set the enforcement to "Audit Only" to help identify executables, MSI packages and scripts which should (not) be allowed to run. When in audit mode, nothing will be blocked by AppLocker.

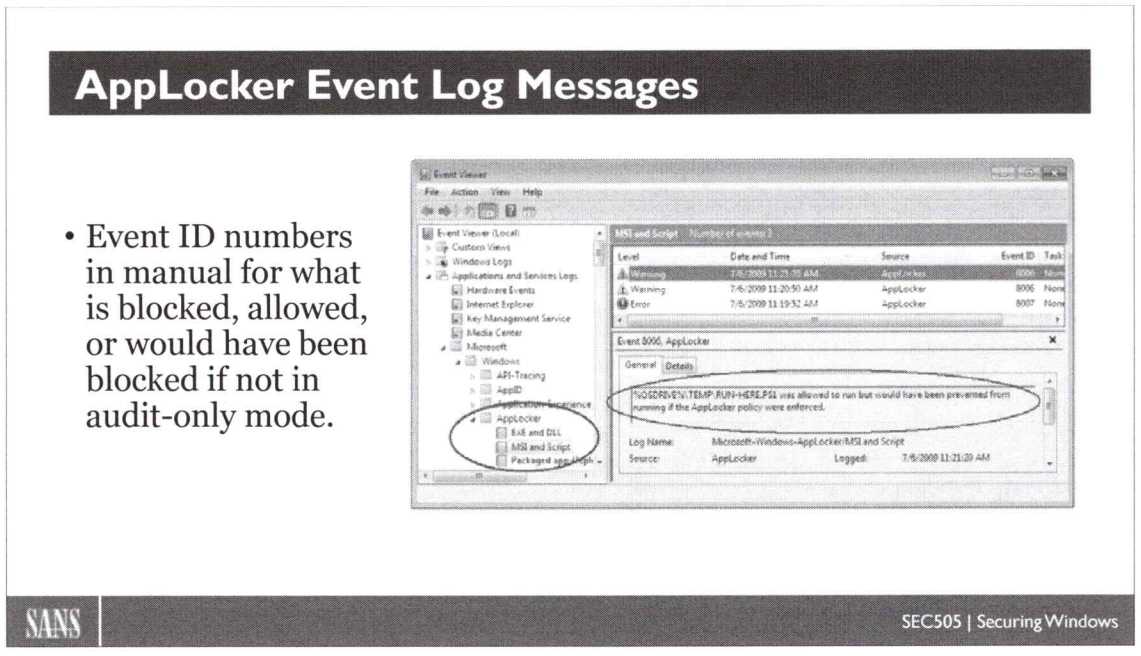
You can choose to enforce rules or merely audit them separately for each of the three categories of applications: executables, Windows installer packages (like .MSI files) and scripts.





Try It Now!

To configure the enforcement and/or auditing of AppLocker rules, open the relevant GPO > Computer Configuration > Policies > Windows Settings > Security Settings > Application Control Policies > right-click on AppLocker > Properties > Enforcement tab > choose "Enforce Rules" or "Audit Only" for executables, packages and scripts. (Choose "Audit Only" here in the lab until after the necessary rules are created.)

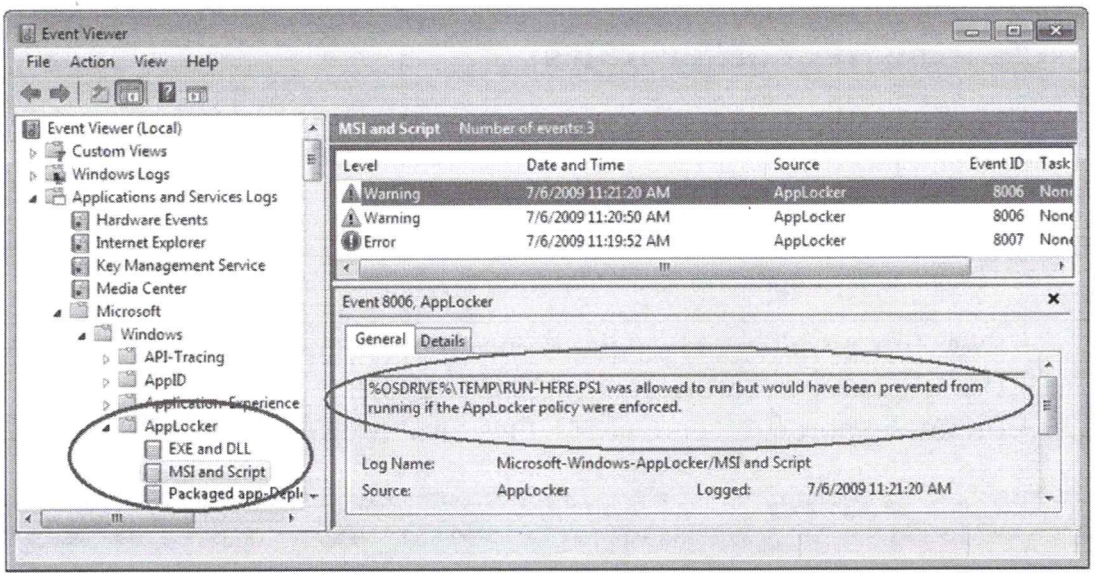


AppLocker Event Log Messages

AppLocker has its own set of Windows event logs that can be used for troubleshooting, monitoring user application/script/package launch, and rule creation with the PowerShell Get-AppLockerFileInformation cmdlet. The AppLocker log is located under Event Viewer > Applications and Services Logs > Microsoft > Windows > AppLocker. The following table describes the types of events you can find there.

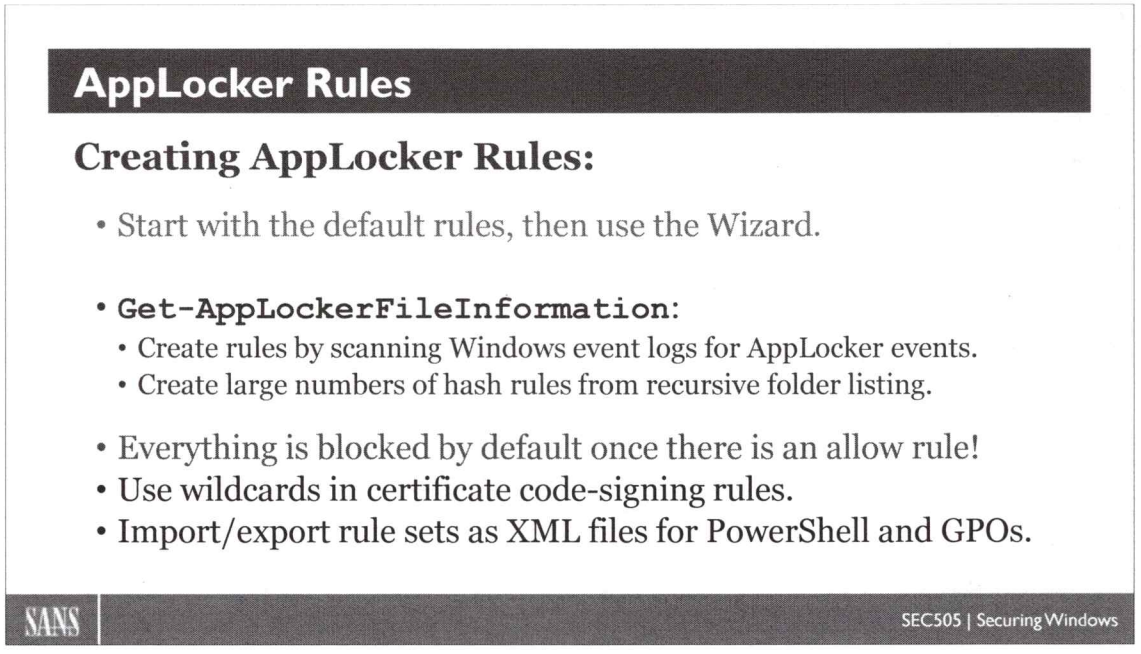
Event ID	Level	Message
8002	Information	EXE or DLL was allowed to run.
8003	Warning	EXE or DLL was allowed to run but would have been blocked if the AppLocker policy were enforced.
8004	Error	EXE or DLL was not allowed to run.
8005	Information	Script or MSI was allowed to run.
8006	Warning	Script or MSI was allowed to run but would have been blocked if the AppLocker policy were enforced.
8007	Error	Script or SMI was not allowed to run.
8007	Error	AppLocker disabled on this edition of Windows.
8020	Information	Packaged app allowed (Windows 8 and later).
8021	Information	Packaged app audited (Windows 8 and later).
8022	Information	Packaged app disabled (Windows 8 and later).
8023	Information	Packaged app installation allowed (Windows 8 and later).
8024	Information	Packaged app installation audited (Windows 8 and later).
8025	Warning	Packaged app installation disabled (Windows 8 and later).
8027	Warning	No Packaged app rule configured (Windows 8 and later).

For example, event ID 8006 is for applications that would have been blocked, and ID 8007 is for applications that were actually blocked.



In PowerShell, if you'd like to extract the block-event messages related to AppLocker:

```
# Script: Get-AppLockerBlockEvent.ps1

Get-WinEvent -Filterhashtable @{ LogName='Microsoft-Windows-
AppLocker/EXE and DLL'; Level=3 } -ErrorAction SilentlyContinue

Get-WinEvent -Filterhashtable @{ LogName='Microsoft-Windows-
AppLocker/MSI and Script'; Level=3 } -ErrorAction
SilentlyContinue

Get-WinEvent -Filterhashtable @{ LogName='Microsoft-Windows-
AppLocker/Packaged app-Deployment'; Level=3 } -ErrorAction
SilentlyContinue

Get-WinEvent -Filterhashtable @{ LogName='Microsoft-Windows-
AppLocker/Packaged app-Execution'; Level=3 } -ErrorAction
SilentlyContinue
```



AppLocker Rules

Keep in mind that once a single AppLocker rule is added to allow something, everything else not explicitly allowed will be blocked. There is an implicit default rule to deny any script or binary from running once you add at least one allow rule to a collection, so you must add all the rules necessary to explicitly allow to run what you want to run.

AppLocker rules can apply to the follow types of files: .exe, .com, .ps1, .bat, .cmd, .vbs, .js, .msi, .msp, .mst, .dll, .ocx, and .appx.

Creating Default Rules

AppLocker can create a set of default rules and it is highly recommended that you at least start with the default rules in order to prevent legitimate applications from being blocked (especially here in the lab). The default rules allow local Administrators to run anything and the Everyone group to run anything under %PROGRAMFILES% or %WINDIR%. You can always delete/edit these rules later.

> ***Try It Now!***
>
> To create default rules for executables, installers or scripts, open the relevant GPO > Computer Configuration > Policies > Windows Settings > Security Settings > Application Control Policies > AppLocker > right-click the relevant container > Create Default Rules.

Note that deny rules take precedence over allow rules.

Automatically Generate Rules

In real life, you'll install all the Microsoft and third-party software your target users need on a known-good reference VM or physical workstation, then generate and test your

AppLocker rules on that system before doing a wider deployment. AppLocker has a built-in wizard which can scan a folder and its subdirectories in order to auto-generate rules to allow the executables, install packages and scripts found there to run. You can edit the generated rules afterwards of course.

> ***Try It Now!***
>
> To generate allow rules automatically, install all desired software, packages and scripts on a reference system you trust, then open the relevant GPO > Computer Configuration > Policies > Windows Settings > Security Settings > Application Control Policies > AppLocker > right-click the relevant container > Automatically Generate Rules > answer the questions of the wizard and proceed.

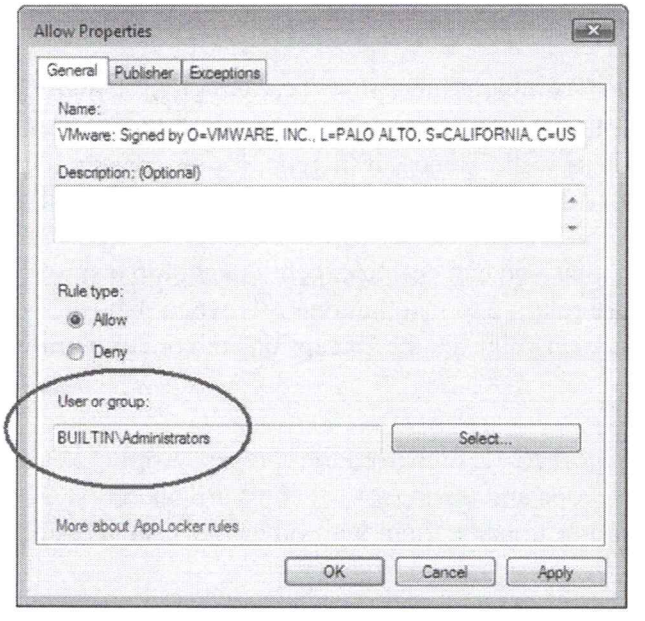
Notice in the wizard that it prefers using digital signature rules, but it can create path or hash rules as preferred for the unsigned items. When possible, the wizard will consolidate signature and path rules to reduce the total number of rules generated.

Create New Rules, Define Exceptions and Specify Groups

Now that you have your default and auto-generated rules, it's easy to add more by hand.

> ***Try It Now!***
>
> To create an AppLocker rule by hand, open the relevant GPO > Computer Configuration > Policies > Windows Settings > Security Settings > Application Control Policies > AppLocker > right-click the relevant container > Create New Rule > answer the questions of the wizard.

Importantly, keep in mind that you can apply different AppLocker rules to different groups. When combined with GPO permissions, WMI filters, OU nesting, etc., you can achieve very precisely targeted AppLocker policies.



One of the best features of AppLocker is the ability to precisely define the characteristics of a digital signature for the sake of allowing/denying a signed item to run. Unlike SRP, AppLocker gives you control over which fields of the signature are relevant and even what data must appear in some of these fields, including the file name and file version number.

And for signature or path rules, you can have exceptions to these rules based on digital signature, path or hash information. Go to the properties of an existing non-hash rule and see the Exceptions tab.

Rule Precedence

Deny rules take precedence over allow rules. If any type of deny rule applies to a file (publisher, path, or hash) then that file is blocked, even if there are other allow rules which apply to that file as well. Hence, if there is, for example, a hash rule that allows an EXE to run, and a path rule that blocks that EXE, that EXE file will be denied.

Keep in mind, though, that you can define exceptions within a single allow or deny rule, but these exceptions are relative to just that one rule when that rule is active. In the properties of an AppLocker rule, see the Exceptions tab on both allow or deny rules.

Export/Import XML

You can import and export XML files used to represent AppLocker rules. This is useful for version control, backups and testing. XML files are plain text, so it's relatively easy to modify them by hand or to parse them with other tools, such as for reporting.

Try It Now!

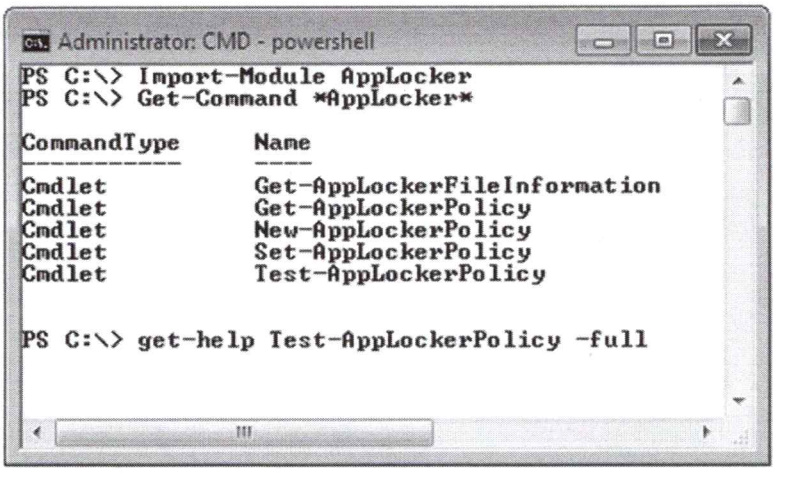
To import/export your AppLocker policy XML, open the relevant GPO > Computer Configuration > Policies > Windows Settings > Security Settings > Application Control Policies > right-click AppLocker > Import/Export Policy.

PowerShell For AppLocker

Not only can AppLocker manage PowerShell scripts and the PowerShell host binaries, but there are PowerShell cmdlets for managing AppLocker policies too. Use the get-help cmdlet to read about their uses and parameters ("get-help applocker").

Imagine you need to frequently re-create an updated XML policy file on a reference machine where you frequently change its configuration. Using the Get-AppLockerFileInformation and New-AppLockerPolicy cmdlets, you could easily script the creation of the new XML policy file after the tools re-scan the system.

If you have multiple XML policy files that you would like to merge together with your hand-built rules, use the Set-AppLockerPolicy cmdlet to merge them into the current local policy, then use Get- AppLockerPolicy to export all your current rules to a new XML file.



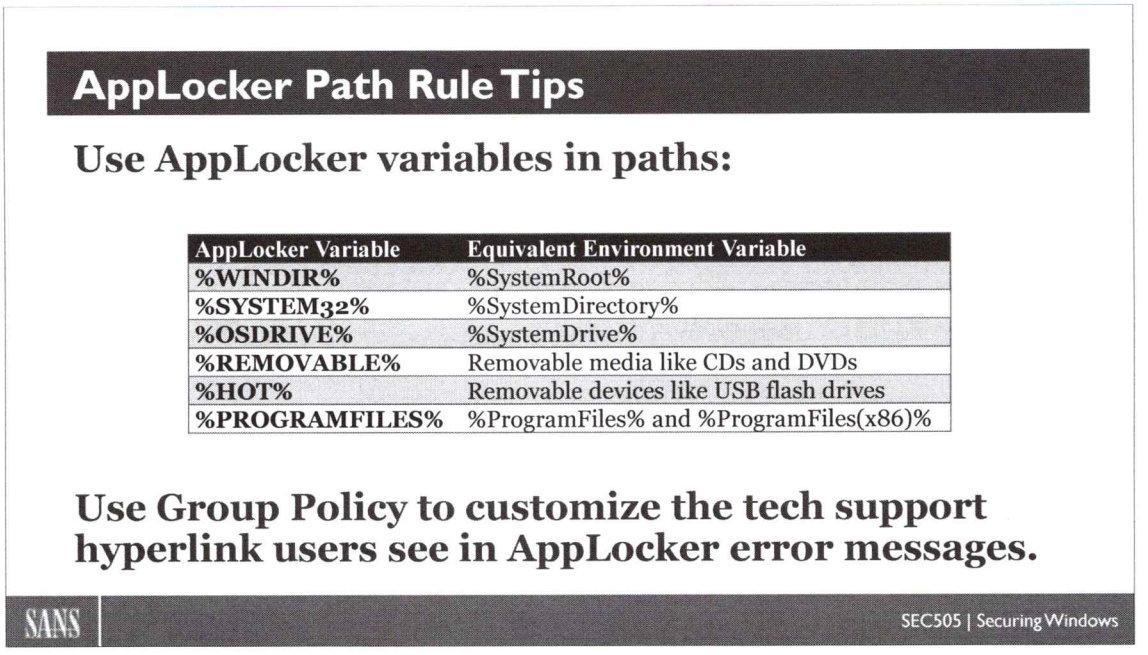
After running AppLocker in audit-only mode for a while on a machine you presume is clean, you could use Get-AppLockerFileInformation to scan the machine's event logs to extract the AppLocker warning messages and use these messages as the basis for creating new allow rules (see the help examples for the -EventLog parameter).

If you are creating a new complex AppLocker policy and would like to know whether it would allow/deny all the executables, install packages and scripts found in a variety of folders on your test machine, but you don't want to actually double-click all those things, then use the test-applockerpolicy cmdlet to report on what would be allowed/denied in those folders, plus it'll give you the reason why anything was denied.

Group Policy Processing

When multiple GPOs apply to a computer and each of those GPOs include AppLocker rules, all of the rules will be combined and applied simultaneously. It is not the case that the last GPO applied which has AppLocker rules will define the only AppLocker rules on a system. Deny rules take precedence over allow rules, no matter the source of the rules.

However, it is the case that the last GPO applied which defines the global enforcement options of "Audit Only" or "Enforce Rules" (on the Enforcement tab) will be the final enforcement options for the system; for example, if a GPO linked at the domain container configures scripts processing to be "Audit Only", and another GPO for an OU configures scripts processing to be "Enforce Rules", then for the computers in that OU the final effective setting will be "Enforce Rules".



AppLocker Path Rule Tips

Keep in mind that the variables used in AppLocker paths are not environment variables, they are special variables just for AppLocker. Here are the valid AppLocker variables:

AppLocker Variable	Equivalent Environment Variable
%WINDIR%	%SystemRoot%
%SYSTEM32%	%SystemDirectory%
%OSDRIVE%	%SystemDrive%
%REMOVABLE%	Removable **media** like CDs and DVDs
%HOT%	Removable **devices** like USB flash drives
%PROGRAMFILES%	%ProgramFiles% and %ProgramFiles(x86)%

When first getting started, use the "Create Default Rules" option for executables, but, for the sake of malware, also add deny rules for the following paths:

- %OSDRIVE%\$Recycle.Bin*
- %OSDRIVE%\Recovery*
- %OSDRIVE%\System Volume Information*
- %HOT%*

NTFS Permissions

When using paths for application blocking, the AppLocker rules can be reinforced with good NTFS permissions and audit settings; for example, NTFS execute permission can

be denied to various groups at various folders, and attempted program execution can be logged in these folders too.

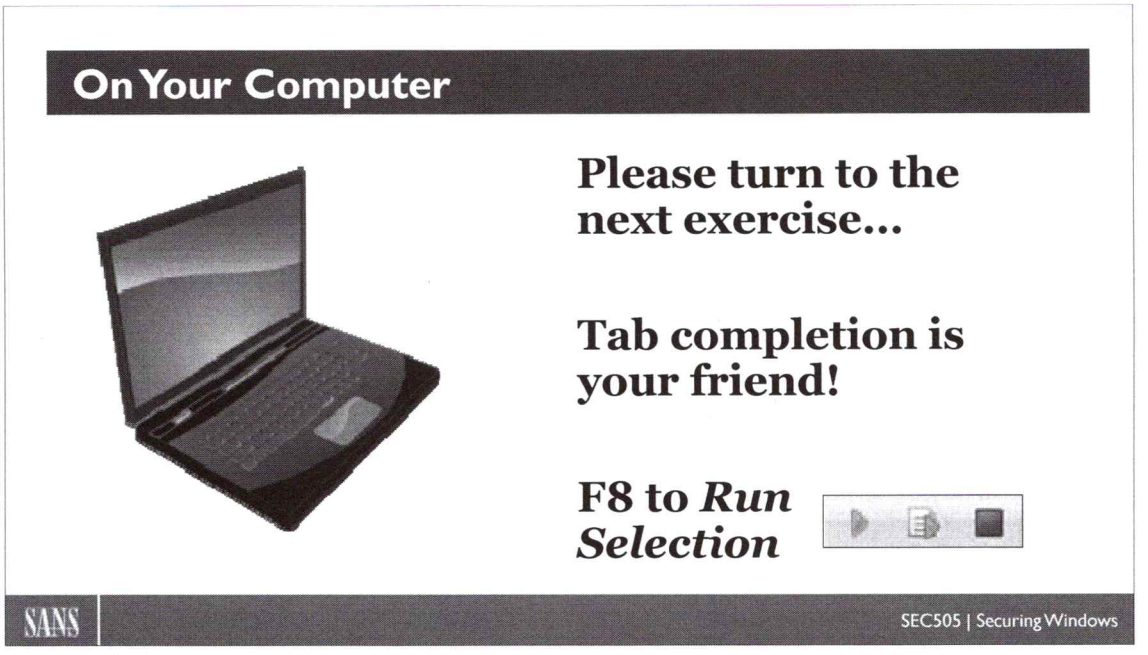
For example, deny NTFS execute permission to the local Users group on the files in the following folders, but do not allow these permissions to be inherited by subdirectories. The deny permissions should apply only to the files in these folders, not any folders:

- %USERPROFILE%
- %USERPROFILE%\AppData\LocalLow
- %APPDATA%
- %PROGRAMDATA%
- %PROGRAM FILES%
- %PROGRAM FILES(x86)%
- %SYSTEMDRIVE%

Finally, don't forget that deny rules take precedence over allow rules. This is true not just for path rules, but for all rule types.

Customize The Tech Support Hyperlink In Error Messages

There is a GPO option to display a custom hyperlink for an internal web page when a user is prevented from running a process or installing a package. Your help page can describe the issue, calm the user and provide a phone number or e-mail address if the user believes they should be allowed to run the program or script. The GPO option is located here: Computer Configuration > Policies > Administrative Templates > Windows Components > Windows Explorer > Set a support web page link.



On Your Computer

Make a snapshot or checkpoint of your VM first, if your VM software supports this.

Import the AppLocker module:

```
import-module applocker
```

List the AppLocker cmdlets from the module:

```
get-command -module applocker
```

Display file information which AppLocker can use to create rules:

```
dir c:\windows\system32\*.exe |
  get-applockerfileinformation | format-list *
```

Now use that information to create a set of AppLocker rules, saved as an XML file:

```
dir c:\windows\system32\*.exe |
   get-applockerfileinformation | new-applockerpolicy -ruletype
   publisher,path -user everyone -optimize -xml |
   out-file .\rules.xml
```

Glance at the XML rules file just produced:

```
notepad.exe rules.xml
```

Close Notepad.

Test the rules to confirm that the target executables will be allowed/blocked as expected:

```
test-applockerpolicy -xmlpolicy rules.xml -path
   c:\windows\system32\*.exe | format-list *
```

Merge the AppLocker rules from the XML file into the local computer's current set of AppLocker rules for testing or hand editing (without -merge, current rules are lost):

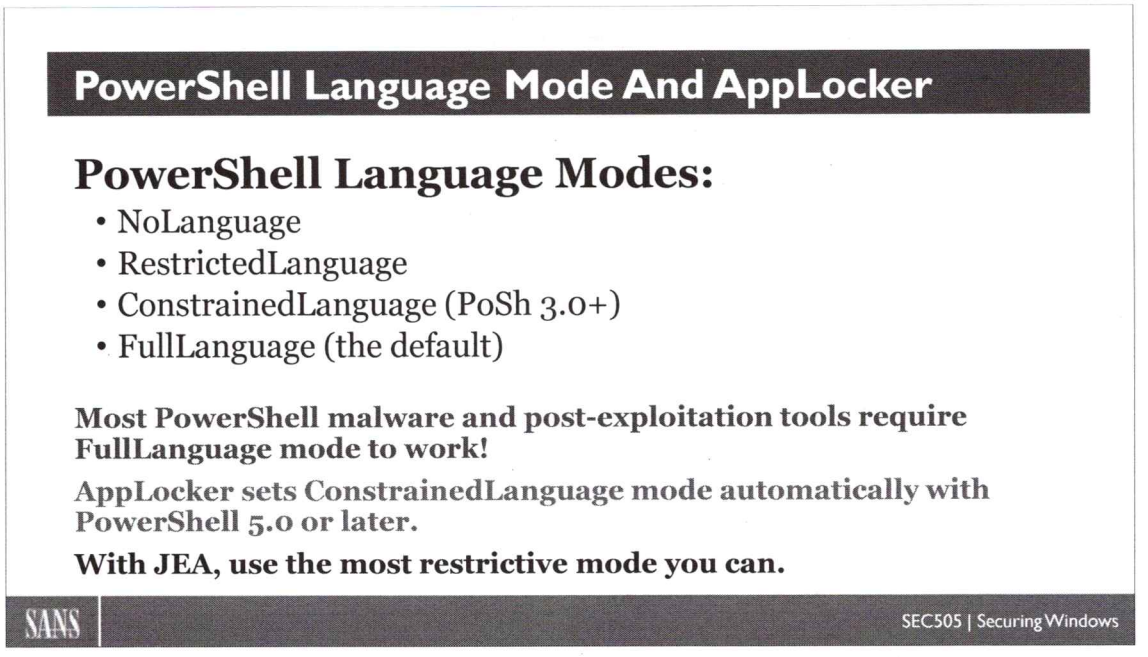
```
set-applockerpolicy -xmlpolicy rules.xml -merge
```

To view the AppLocker rules just created, don't try to do a refresh, instead, open a new MMC console (mmc.exe) > File menu > Add/Remove Snap-in > select "Group Policy Object Editor" > Add button > Finish > OK.

In the Local Computer Policy GPO, navigate to Computer Configuration > Windows Settings > Security Settings > Application Control Policies > AppLocker > Executable Rules. Here you can see the rules created from the XML file.

Close the MMC console without saving changes (click No).

In real life, once the AppLocker rules have been thoroughly tested, they can be exported from the local computer and then imported into a domain GPO using the Group Policy Management console.



PowerShell Language Mode And AppLocker

We've seen how AppLocker can be used to restrict which PowerShell scripts are permitted to run. AppLocker can also restrict binary executables which host the PowerShell engine, such as powershell.exe and powershell_ise.exe, or attempt to load PowerShell-related DLLs.

Once you add even a single AppLocker allow rule, then anything not explicitly allowed by AppLocker is blocked by default. Having an allow rule switches AppLocker into "Allow Mode", i.e., block by default. But there's more. PowerShell 5.0 and later automatically detects whether AppLocker is in Allow Mode, and, if it is, PowerShell itself switches to a different mode: Constrained Language Mode.

PowerShell Language Modes

PowerShell has several "language modes", each of which determines the features of PowerShell that are permitted to execute or be accessed in that mode. A language mode is defined for a local or remote session, and may be customized with a session configuration file (also called a "remoting endpoint").

These are the available language modes for a PowerShell session:

- NoLanguage
- RestrictedLanguage
- ConstrainedLanguage (in PowerShell 3.0+)
- FullLanguage (the default)

To see your current language mode (either directly or in an error message):

```
$ExecutionContext.SessionState.LanguageMode
```

To read the built-in help about language modes:

```
get-help about_Language_Modes
```

What cannot be accessed or executed in the various language modes?

- **NoLanguage**: No scripting language features may be used at all, only commands may be run; but the cmdlets or scripts that are run will have full access to PowerShell language elements inside of them.

- **RestrictedLanguage**: Cannot run scriptblocks; very few variables are accessible; very few operators may be used; cannot invoke method calls; cannot access property references; and no assignments are permitted. This is only slightly more permissive than NoLanguage mode.

- **ConstrainedLanguage**: Cannot call into the Windows API (Win32); only a very short list of allowed types from the .NET Framework may be created, accessed or used; the Add-Type cmdlet cannot load arbitrary C# code; and very few COM objects may be accessed. This mode aims to strike a reasonable balance between security and functionality.

- **FullLanguage**: Nothing is blocked. This is the default.

Again, when AppLocker has even one allow rule, the PowerShell language mode automatically switches to ConstrainedLanguage (with PowerShell 5.0 and later). This is very important for security because virtually all currently-known PowerShell malware and post-exploitation tools require FullLanguage mode in order to run.

Just Enough Admin (JEA)

Even better, with Just Enough Admin (JEA), you can set the language mode for the JEA session configuration endpoint. Ideally, set the mode to NoLanguage, but, in general, set the mode to the most restrictive possible while still allowing users to get their work done. Strive to use at least ConstrainedLanguage mode at a minimum however.

Set The Mode By Environment Variable (Not Recommended)

The proper way to use language modes for security is through AppLocker and/or JEA. We have to work quite a bit to even try to stop skilled adversaries.

But it is possible to set the default language mode with a machine-wide or system environment variable named "__PSLockdownPolicy". When set to 4, for example, this variable will make ConstrainedLanguage the default mode. (Apparently the other modes are not currently assignable with different numbers this way.)

To permanently set the environment variable to make ConstrainedLanguage the default:

```
[Environment]::SetEnvironmentVariable("__PSLockdownPolicy","4","M
achine")
```

To delete that environment variable to revert back to the default language mode (Full):

```
[Environment]::SetEnvironmentVariable("__PSLockdownPolicy",$null,
"Machine")
```

You can manage environment variables through Group Policy as well: GPO > Computer Configuration > Preferences > Windows Settings > Environment.

But if hackers or malware have taken over a machine, they can modify environment variables defined in the registry too. AppLocker and/or JEA are recommended instead.

Blocking Unsigned WSH Scripts (Not Using AppLocker)

A related whitelisting feature to be used alongside AppLocker is unsigned script blocking. You can digitally sign PowerShell and Windows Script Host (WSH) scripts, and Windows will check the validity of these signatures before running the scripts. If a signature is missing, corrupted or untrusted, you can either block the script entirely or simply warn the user with a pop-up message.

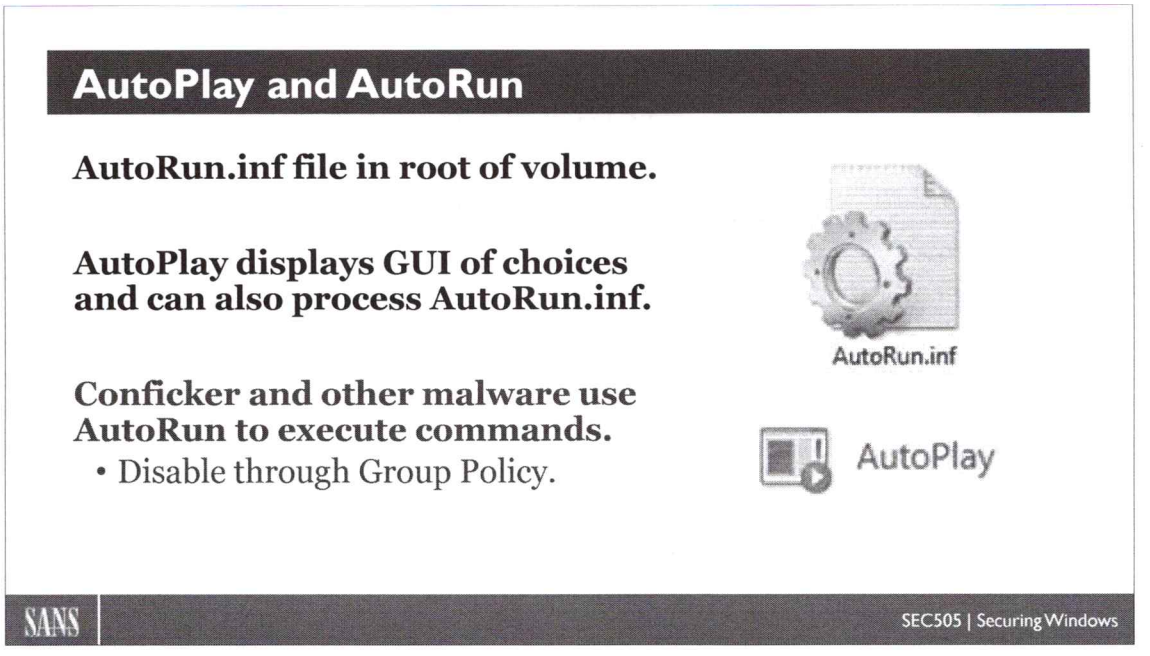
Windows Script Host

The Windows Script Host (WSH) is composed of two binaries, CSCRIPT.EXE and WSCRIPT.EXE, and is what executes VBScript and JScript scripts. You can also get Perl and Python WSH plug-ins too (www.activestate.com). POWERSHELL.EXE is the main interpreter for PowerShell scripts. PowerShell 1.0, WSH 5.6 and later can be used to digitally sign scripts and to verify signatures in scripts. The signature does not encrypt the script. The signature itself is a block of code appended to the end of the cleartext script, and, because this block is commented out, it can be ignored by the interpreter if need be. Contained in this block is a hash of the script encrypted with the private key of the original script developer and the developer's corresponding public key code-signing certificate (plus other information). If the developer's code-signing certificate was issued by a Certification Authority (CA) the user trusts, the user's computer can check the hash in the signature to verify the script hasn't been modified. Through Group Policy, you can manage exactly which CAs your users will trust. The idea is that you will have your own code-signing certificate(s) and you will make your users' computers trust your certificate(s) through Group Policy.

When WSH 5.6 or later is installed, a REG_DWORD registry value named "TrustPolicy" determines whether users can execute scripts which are unsigned or untrusted (HKCU\SOFTWARE\Microsoft\Windows Script Host\Settings\TrustPolicy). The TrustPolicy value can take one of three settings:

0 = Run the script, signed or not (the default).

1 = Prompt user whether to run the unsigned/untrusted script.
2 = Prevent the unsigned/untrusted script from running.



AutoPlay and AutoRun

AutoPlay and AutoRun are Windows features that control the execution of commands and/or the display of a GUI dialog box when inserting a CD/DVD, flash drive, USB peripheral device or when mapping a drive letter to a shared folder. AutoRun is the older feature more narrowly associated with the execution of a command in the autorun.inf file stored in the root folder of a mounted volume. AutoPlay is the newer and more broad term which also includes the options shown on the pop-up GUI asking what to do when new media is mounted. For many years, malware has exploited the AutoPlay/AutoRun feature to execute malicious commands, including Conficker.

Fortunately, by editing the registry through Group Policy, scripts or some other means, you can disable AutoRun and manage the AutoPlay defaults to be more safe. The registry options available are too numerous to list here, so please see the references below, but the main GPO settings can be found here:

- Windows 2000/XP/2003: Computer Configuration > Administrative Templates > System > Turn Off AutoPlay.

- Windows Vista and later: Computer Configuration > Policies > Administrative Templates > Windows Components > AutoPlay Policies

While you will want to disable AutoRun, you may wish to keep AutoPlay enabled as-is or with some modification for the sake of compatibility, but, if in doubt, you can disable both. It is only a slight inconvenience for the user to open File Explorer.

Hardening Layer 8

Unfortunately, even with AutoRun and AutoPlay disabled, users can still be tricked into executing malicious programs or opening malware-infected data files on USB flash drives and other removable media; for example, simply giving these files enticing names can lure users into opening them to take a peek. If removable media and devices aren't blocked entirely, the only solution is to train users to avoid doing such bad things.

It can seem pointless at times, but user security awareness training can help to reduce infections. Since you don't manage users' personal computers at home, user training can help reduce the flow of home infections being transported back into the office by flash drive, iPod, netbook and e-mail. In fact, you might consider simply giving AV scanners to your users for use at home for free (including the yearly updates) on a DVD you build for them which includes other security software too, e.g., updated browser installers, alternative PDF reader, personal firewall, patch management utilities, etc. Often, your AV site license includes a free personal use license for each business license you purchase, so you might not have to spend an extra penny.

Here are some tips about format and delivery of anti-malware training:

- Make it mandatory for new employees and require a short quiz afterwards.
- Focus on how bad habits can directly affect that user's productivity, performance reviews and compensation.
- Emphasize how good habits can protect them at home too, e.g., identity theft, privacy, lost data, etc.
- Use real examples, if possible, from current employees who have suffered harm.
- Avoid technical jargon and academic background knowledge; keep it plain, simple and practical.
- Use videos, screenshots and demonstrations as much as possible.
- Incorporate games, jokes, stories and audience interaction; avoid 100% lecture.
- Follow up with monthly e-mail reminders, but include cartoons, jokes, tips or some other teaser.
- Try to identify those groups who get infected most often and focus on them.
- Don't have one generic training course for all users, customize the content.

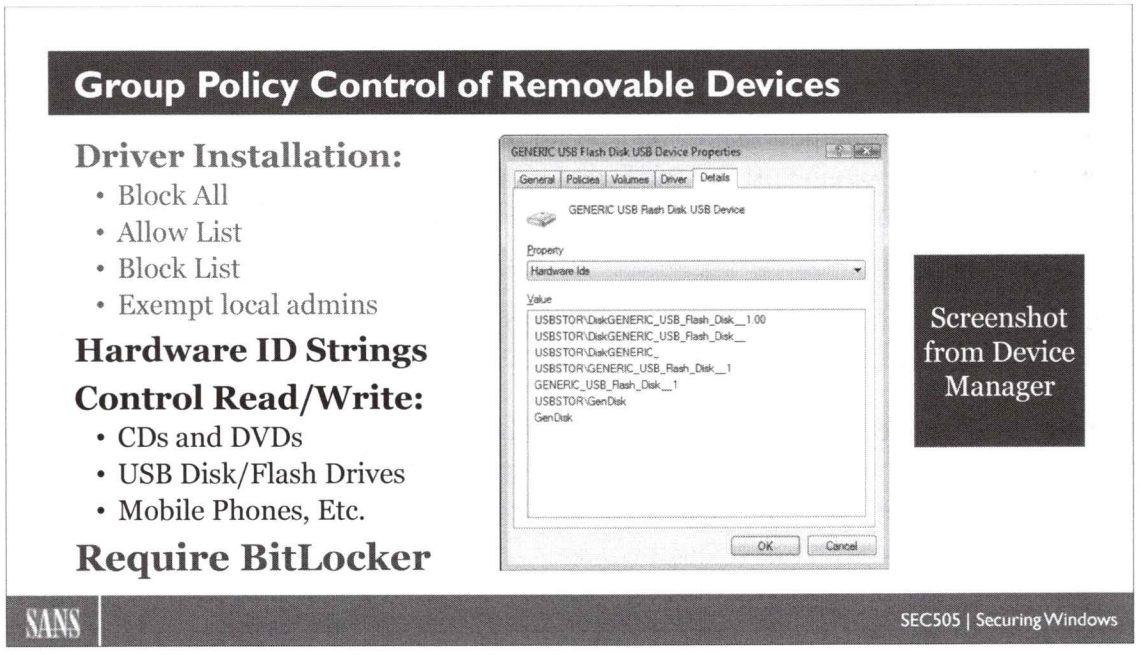
The training topics should at least include the following, but this list really will be determined by the details of your environment, e.g., which browser you use, which e-mail client, AV pop-up dialog boxes, and so on:

- "Human IDS" examples of how and when to alert the help desk.
- Personal and corporate harm resulting from bad practices.
- Handling suspicious e-mail file attachments.
- Recognizing phishing attempts.
- Safe web browsing habits.
- Screenshots of the company's legitimate in-use AV software.
- Dangers of peer-to-peer applications.

- AutoPlay dialog boxes.
- Policies concerning removable media.
- Policies for installing new software.
- Policies for using personal computers for doing office work.
- Necessity of logging off at day's end.
- Good security practices at home, e.g., AV, firewall, passwords, browsers, etc.

The SANS Institute sponsors the "Securing The Human" project for end-user security training that's relevant and fun (www.securingthehuman.org). The project has both free and commercial resources for trainers.

You may also be interested in NIST Special Publication 800-50, *Building an Information Technology Security Awareness and Training Program*, which can be downloaded for free from http://csrc.nist.gov/publications/nistpubs/800-50/NIST-SP800-50.pdf.



Group Policy Control of Removable Devices

Using Group Policy, you can regulate which type(s) of USB devices are permitted to be connected to Windows Vista and later. You can block all USB devices, enforce an allow list of approved devices and deny all else, enforce a block list and allow all else, or control read and/or write access to removable devices generally.

The device driver policy settings are found in the GPO under Computer Configuration > Policies > Administrative Templates > System > Device Installation > Device Installation Restrictions. With the exception of "Allow administrators to override device installation policy", these device driver policies apply to *anyone* logging on at the managed computer, hence, you cannot assign different device driver installation policies to different users or groups.

The option to control read/write access to removable media is found under User or Computer Configuration > Policies > Administrative Templates > System > Removable Storage Access. These removable storage policies can be applied to users and user groups, hence, they can be different for different groups in AD. If configured differently in both User and Computer Configuration, the Computer Configuration settings will win.

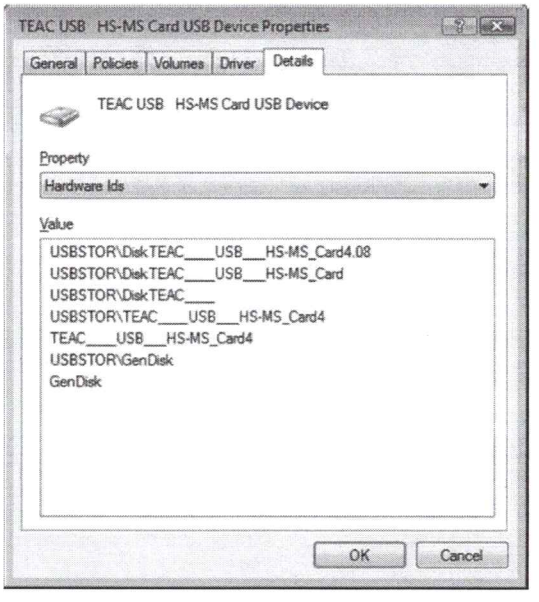
In both sets of policies, there are options to regulate devices based on their hardware ID strings and setup GUID strings.

Hardware Identifier Strings and Setup Classes

Hardware manufacturers encode one or more identifier strings into their products which Windows can read. Windows matches these strings from the device against similar strings in the .INF files associated with their device drivers, helping Windows to locate

the correct driver for a particular device. Identifier strings can be very exact, specifying the exact make and model of a particular version of hardware, or they can be more general and abstract, which assists is locating a compatible driver when the precisely correct driver cannot be found (for more information about the procedure, see http://msdn.microsoft.com/en-us/library/ff546228.aspx).

Similarly, manufacturers can label their products with GUID numbers that identify the "setup class" of their devices, and Windows can also use these GUID number to help install the device.



To see an identifier string for a device, open Control Panel > System > Device Manager > properties of plug-n-play device, such as a NIC or USB device > Details tab > select "Hardware Ids" from the property menu. ID strings such as these will be used in Group Policy Objects to regulate device driver installation.

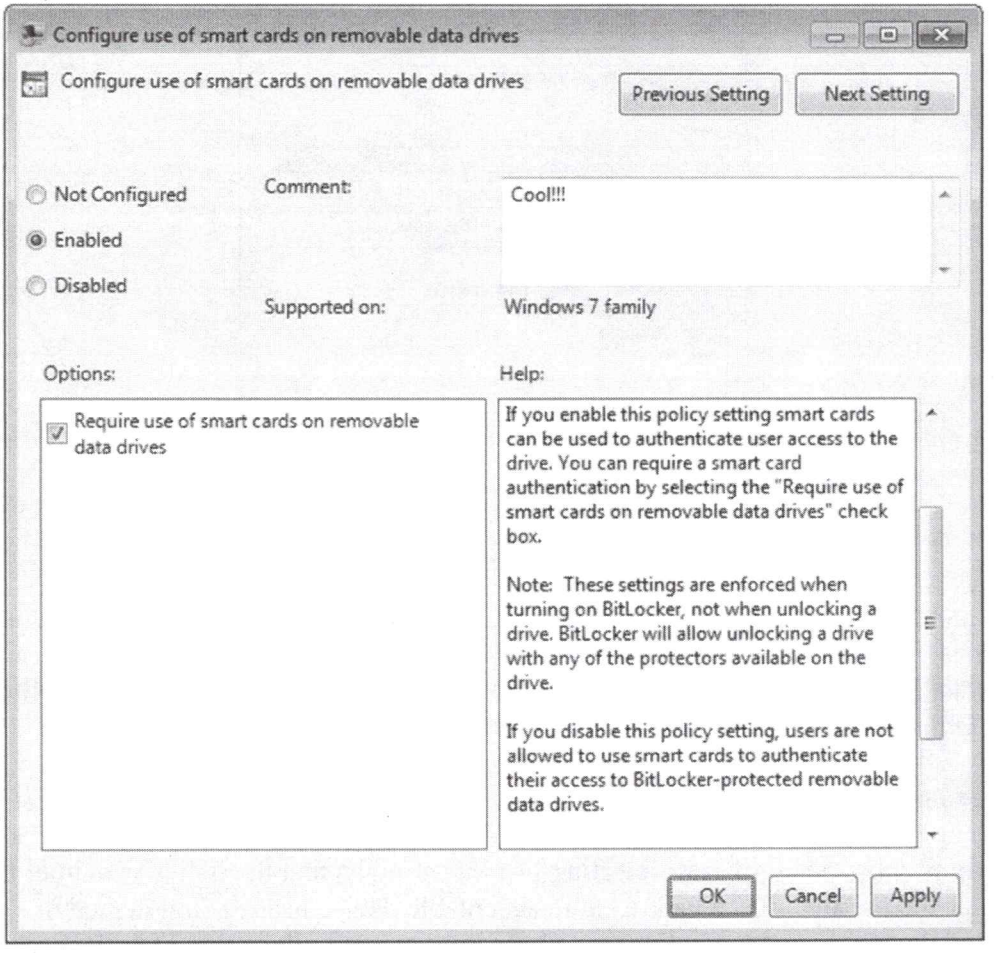
When examining the Details tab of device, hardware ID strings are listed from most specific at the top to least specific at the bottom. The primary difficulty in using these new Group Policy features is in selecting just the string so that the policy is neither too broad nor too narrow, which can result in acceptable device usage being denied or unwanted device usage being allowed. There is no magic bullet solution to the problem of choosing the right ID string(s), only trial-and-error testing will yield the desired results. Microsoft's documentation on these features often either gives contrived examples that are uselessly narrow or simply hand-waves the difficulties away without offering much real-world advice, so please do test it all in a lab first.

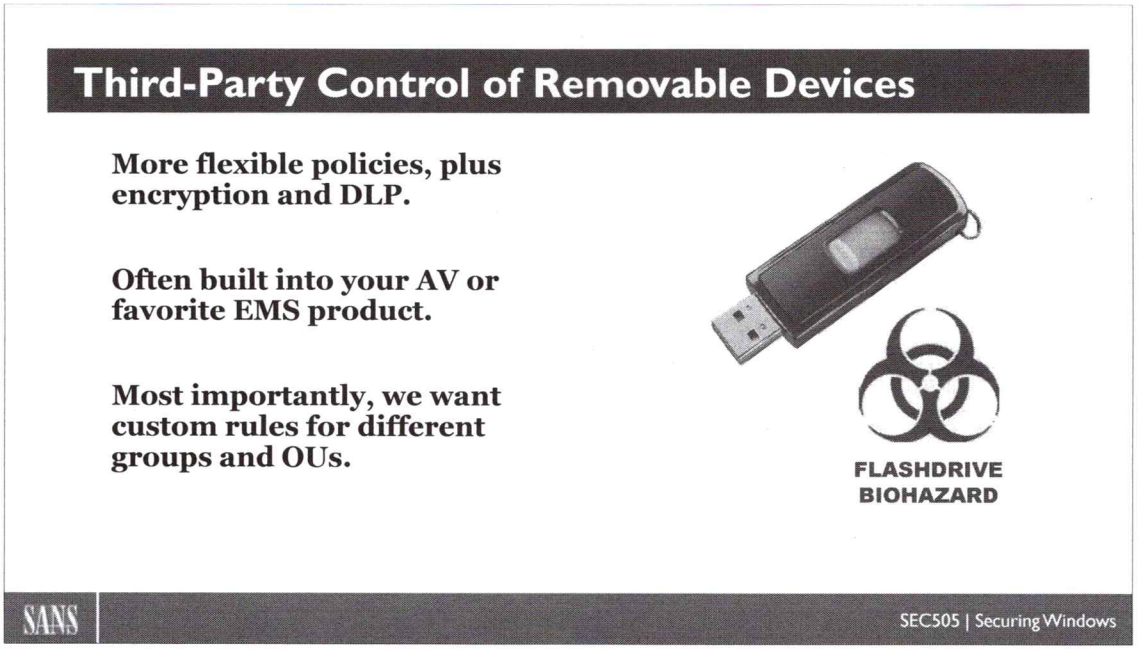
Manage BitLocker Requirements On Removable Drives

You have Group Policy control over most BitLocker options and requirements, including the use of BitLocker To Go on removable drives. For example, you can use Group Policy to enforce the following:

- Deny write access to removable drives not protected by BitLocker.
- Require a smart card to access a BitLocker encrypted drive.
- Do not allow BitLocker To Go on FAT-formatted removable drives.
- Require minimum length and complexity for BitLocker passphrases.
- Require a TPM for BitLocker on fixed (non-removable) drives.
- Require a minimum PIN length for BitLocker on fixed drives.

You can find these options in a GPO by navigating to Computer Configuration > Policies > Administrative Templates > Windows Components > BitLocker Drive Encryption.





Third-Party Control of Removable Devices

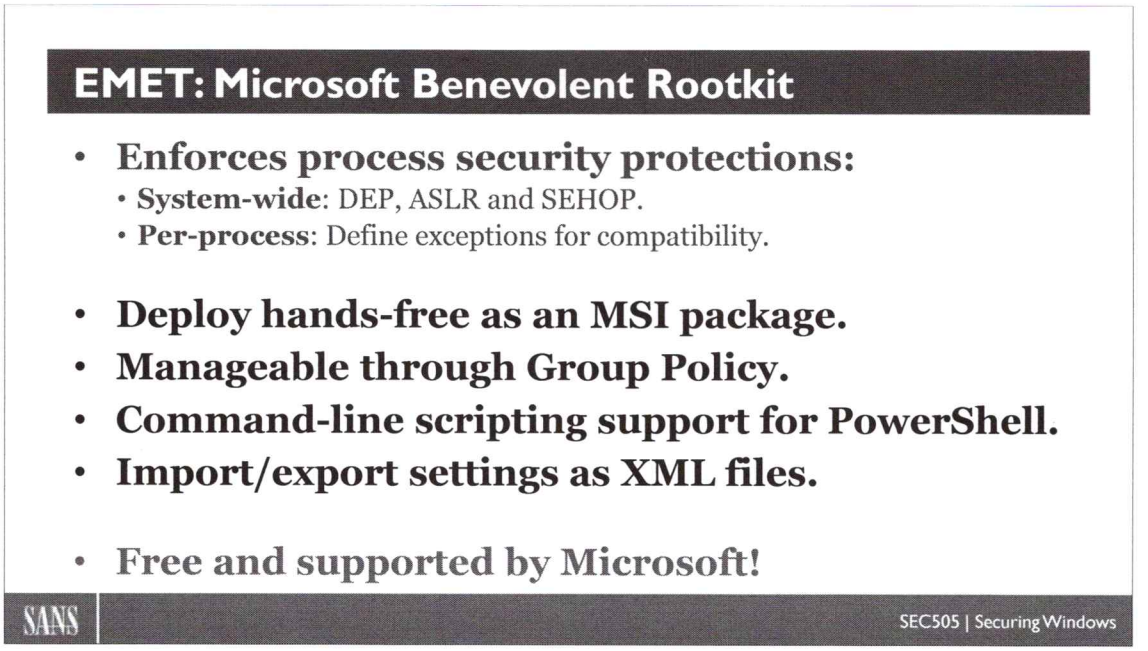
While Group Policy control of removable devices is more-or-less free, it is also a bit crude in comparison to the third-party products available for device control. There are third-party products whose sole purpose is to provide centralized device control and these generally are the most flexible. There is also an excellent chance that your favorite AV or EMS already has device control as a feature, hence, you could save money by just using what you have. For a few examples of what's available, these vendors provide device control, and there are certainly many more:

- Bit9 Security Platform (www.bit9.com)
- CoSoSys Endpoint Protector (www.endpointprotector.com)
- DeviceLock (www.devicelock.com)
- Kaspersky Device Control (www.kaspersky.com)
- Lumension Device Control (www.lumension.com)
- McAfee Device Control (www.mcafee.com)
- Sophos SafeGuard (www.sophos.com)
- Symantec Endpoint Protection (www.symantec.com)

However, there are a few important facts to keep in mind when evaluating device control products. The aim here is to prevent the spread of malware through removable devices, but most of these vendors are focused on data encryption and Data Loss Prevention (DLP). To prevent malware transmission, we mainly want to prevent read and write access to devices, not encrypt them. In fact, if a USB flash drive is encrypted, that might make it more difficult for your AV to scan it. Take care not to waste money on features

you don't intend to use, and if Group Policy provides adequate device control for your needs, there's no reason to purchase something else.

For regulating read/write access, the most important feature is the ability to define flexible rules on the basis of Active Directory group membership and organizational unit location. You will have some users who should not be permitted to use removable devices at all, others who can only use non-storage devices, and others who will need read/write access to anything they wish. If you don't have this flexibility, there will be political backlash against your harsh policies and you might end up just turning off all the restrictions again. Group Policy can target device control policies to specific groups and OU's, but not all AV or EMS products can do so, you'll have to read the fine print.



EMET: Microsoft Benevolent Rootkit

The free Enhanced Mitigation Experience Toolkit (EMET) utility from Microsoft allows you to bolt security features onto a process even after its executables have been compiled. In effect, EMET operates as a benevolent rootkit which you control.

EMET uses the same API hooking techniques as the Application Compatibility Toolkit (ACT) to modify how a process calls certain functions provided by the operating system. EMET adds "shims" to a process such that a function call normally handled by the OS is first processed by EMET's shim code instead before routing to the OS.

EMET includes the following features:

- Free.
- Compatible with Windows XP, Server 2003, and later.
- Compatible with both 32-bit x86 and 64-bit x64 processes.
- MSI-packaged for installation through script, SCCM or Group Policy.
- Can manage system-wide protections like DEP and SEHOP.
- Can manage per-application protections.
- Each protection can be enabled/disabled on a per-application basis.
- Manageable through GUI, scripts, Group Policy, or SCCM.
- Writes troubleshooting data and alerts to the Application event log.
- Includes sample XML and ADMX files to quickly get started.

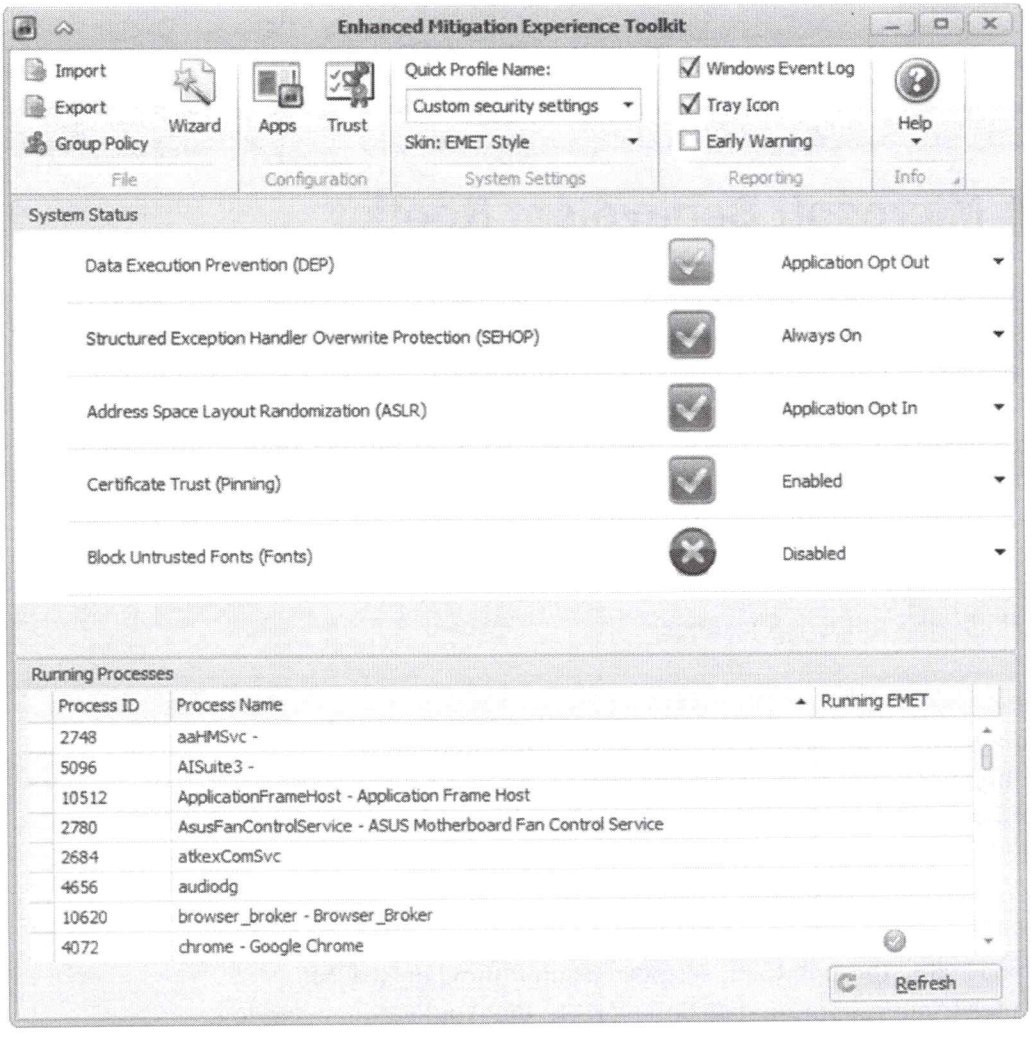
Installation

To get EMET, see KB2458544 or search on "site:microsoft.com emet download" to get the latest version.

EMET is compatible with Windows XP-SP3, Vista-SP1, Server 2003-SP1 and later. Windows 10, Server 2016 and later require at least EMET 5.5.

EMET requires the .NET Framework 4.0 or later to be installed.

After installing EMET, look in its installation directory (%ProgramFiles(x86)%\EMET) for the PDF user's guide, the \Deployment subdirectory with GPO files, and command-line management tool (EMET_Conf.exe).



Graphical Interface

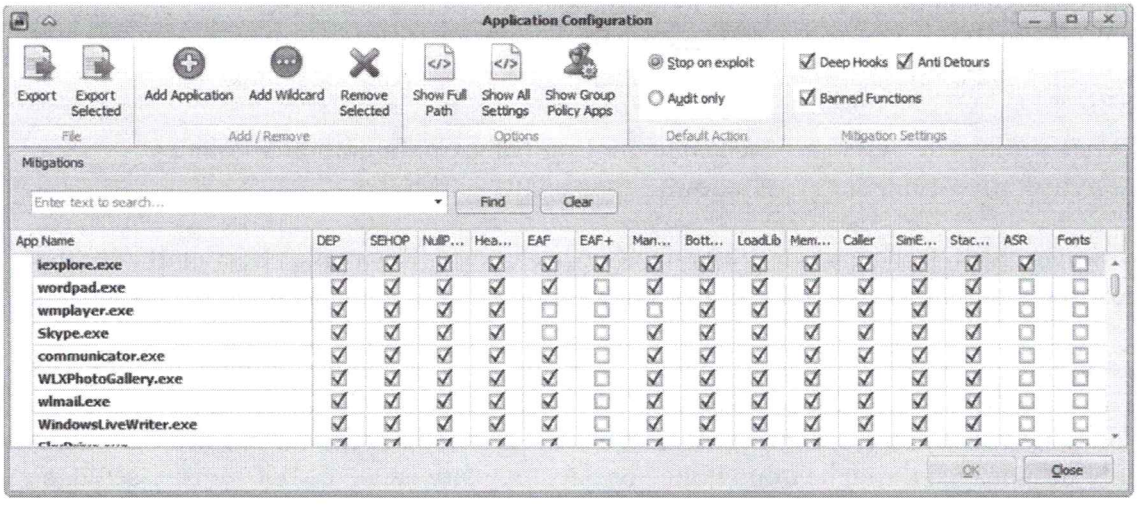
EMET's graphical management program is EMET_GUI.EXE. It can be launched from the EMET icon in the notification area of the taskbar next to the clock.

Try It Now!

After installation, right-click on the EMET icon and select "Run As Administrator". In the Start screen, just type "emet" to bring up the GUI app icon.

In the EMET interface, the "Configure System" button is for setting system-wide defaults for DEP, ASLR and SEHOP. These defaults can also be managed from the command line (EMET_Conf.exe --System) and Group Policy.

> **Warning!** If BitLocker is enabled using a TPM, a change in boot-up DEP settings will trigger BitLocker recovery mode at the next reboot! You will need the BitLocker recovery PIN or USB flash drive to reboot successfully again.



In the EMET interface, the "Configure Apps" button is for adding particular applications whose execution should be protected by EMET. Ideally, every protection checkbox would be enabled for protected applications, but notice that incompatible protections can be separately disabled on a per-application basis. These settings can also be managed from the command line (EMET_Conf.exe --Set) and Group Policy.

Command-Line Scripting and Protection Profiles

The PDF user guide in EMET's installation folder (EMET User's Guide.pdf) contains examples and guidance for scripting the management of EMET's protection settings. The command-line switches are pretty simple, and scripting changes to EMET is actually easier than using Group Policy. EMET configuration scripts would likely be deployed through Group Policy as start-up scripts.

An EMET "Protection Profile" is just an XML file with EMET settings. You can import or export EMET settings from the command line to simplify administration; for example, a long list of customized protection settings might be created, tested and exported from one machine, then imported with a single-line script on many other machines. The XML file and associated scripts could be placed in a read-only shared folder.

Look in the %ProgramFiles(x86)%\EMET\Deployment\Protection Profiles\ folder for example XML files, especially the All.xml file.

Note that EMET settings configured through Group Policy take precedence over locally-administered settings when there is a conflict. Deleting local EMET settings will not delete any Group Policy EMET settings, which will still be enforced.

Group Policy Support

EMET can be installed and managed through Group Policy. The ADMX template for EMET is located in %ProgramFiles(x86)%\EMET\Deployment\Group Policy Files\.

On the local computer, the EMET.admx file should be copied into the %WinDir%\PolicyDefinitions folder, and the EMET.adml file should be copied into the %WinDir%\PolicyDefinitions\en-US folder (or whatever your locale may be).

EMET settings in a GPO are located under Computer Configuration > Policies > Administrative Templates > Windows Components > EMET.

However, EMET changes made through Group Policy will not take effect until after the next reboot or until after "EMET_Conf.exe --refresh" is executed, whichever comes first.

Group Policy EMET settings take precedence over locally-configured EMET settings whenever there is a conflict. Deleting local EMET settings will not delete any Group Policy EMET settings, which will still be enforced. Group Policy EMET settings can only be managed through Group Policy or direct registry edits. EMET registry settings are located under HKLM\SOFTWARE\Policies\Microsoft\EMET.

Group Policy EMET settings are preceded with a ">" character in the output of the "EMET_Config.exe --list" command.

Troubleshooting

See the Application event log for troubleshooting events from the EMET source. Whenever possible, EMET will log both problems and blocked exploits. The EMET icon in the notification area of the taskbar will also display a pop-up message when there is an issue.

Data Execution Prevention (DEP)

Data Execution Prevention (DEP) is a security feature which prevents the execution of code in pages of memory that are not explicitly marked as executable. Some exploits require the attacker's input to be copied to an unexpected location in memory and then executed there; one purpose of DEP is to prevent this from happening. DEP requires XP-SP2, 2003-SP1 or later operating systems, and an Intel CPU with XD (execute disable) or an AMD CPU with NX (no execute) support too, hence, DEP has both hardware and software requirements.

DEP has four possible system-wide settings:

- **OptIn**: DEP disabled for all 32-bit processes unless explicitly requested by a process; enabled for 64-bit processes by default, but a process can choose to opt out; this is the default for Windows XP/Vista/7/8 and later.

- **OptOut**: DEP is enabled for all processes by default, but a process can choose to opt out; this is the default for Server 2003/2008/2012 and later.

- **AlwaysOn**: DEP is enabled for all processes, no exceptions.

- **AlwaysOff**: DEP is disabled for all processes, no exceptions.

To confirm DEP is enabled for a process, add the "DEP Status" column in Process Hacker, or see the General tab in the properties of a process in Process Hacker.

> **Warning!** If BitLocker is enabled using a TPM, a change in boot-up DEP settings will trigger BitLocker recovery mode at the next reboot. You will need the BitLocker recovery PIN or USB flash drive to reboot successfully again!

To check the system-wide DEP status from the command line on Vista/2008 and later:

```
bcdedit.exe | findstr.exe nx
```

To set the system-wide DEP status to OptOut on Vista/2008 and later:

```
bcdedit.exe /set nx optout
```

To check DEP status in the Control Panel, open the System applet > Advanced System Settings link > Advanced tab > Performance Settings button > Data Execution Prevention tab.

> **Note:** There is something called "software-enforced DEP" which has little to do with what's being discussed here. Software DEP is related to structured exception handling in software, not to NX/XD support in the CPU.

Define Exceptions to DEP Outside of EMET

Using the Application Compatibility Toolkit (ACT), sysadmins can deploy shims enterprise-wide to opt out applications which are incompatible with DEP, but only on 32-bit systems (the fix does not exist for 64-bit platforms). This is done with the "DisableNX" compatibility fix in ACT.

But on both 32-bit and 64-bit systems, the registry can be modified to define which applications should be exempted from DEP. To make the change, use a script to add a new string value to the HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\AppCompatFlags\Layers key; the name of the value should be the full path to the executable, the data in the value should be "DisableNXShowUI".

Between the ACT fix and the registry exemptions, the OptOut setting can be the default on all systems for DEP, not just servers.

ASLR

Starting with Windows Vista/2008 and later, the memory locations of EXE images, DLL images, thread stacks, process environment blocks (PEB), and process heaps can be set to unpredictable addresses when processes are launched. Across two reboots of a computer, there is only a small chance that a given EXE or DLL will load to the exact same memory location twice. Additionally, the memory just before the heap of a process is deallocated so that an attempt to search for the beginning of the heap will likely cause an access violation and kill the process. These features are collectively called "Address Space Layout Randomization" or ASLR.

ASLR's lack of predictability makes it more difficult for exploit coders. If a piece of malware cannot rely on finding useful executable code or data structures at fixed memory locations, the malware will either have to search for what it needs or simply do without. Because searching is difficult to code into an exploit and prone to crashing the target process, malware must often be written without hard-coded assumptions about the locations of these useful functions or data. Especially when ASLR is combined with DEP, exploits are more difficult to develop, and compromised systems are less likely to be *completely* taken over if they are infected (a hang or system crash is better than silent subversion).

While it is possible to force ASLR on or off for all processes (using the MoveImages registry value), for compatibility it's best to let a binary indicate whether it's ASLR-compatible or not in its PE header. You can determine whether a running process is using ASLR in Process Hacker by adding the "ASLR" column in the list of running processes (right-click any column header > Choose Columns > check ASLR Enabled).

As an example, CVE-2010-3971 describes an IE8 exploit which attacks a DLL in the browser that was not compiled for ASLR and DEP support, but if IE is forced to run with full ASLR and DEP support, such as with Microsoft's Enhanced Mitigation Experience Toolkit (EMET), the exploit fails. In IE9 and later, all the browser's DLLs are compiled with ASLR and DEP support.

On 64-bit Windows 8 and later, ASLR is even more effective because of the larger memory address space and increased randomization. With the appropriate patch, Windows 7 supports this as well (KB2639308). It is also possible to force an application to submit to ASLR randomization --even if the application was not compiled with ASLR support originally-- by modifying a registry value named "MitigationOptions", but using EMET is probably easier and more effective anyway.

In short, ASLR best practices are:

- Prefer software which is compatible with ASLR over software which is not, and especially aim for the use of ASLR and DEP together.

- Use 64-bit Windows 8 or later to benefit from improvements in ASLR, especially with 64-bit applications that have been compiled with ASLR support by design.

SEHOP

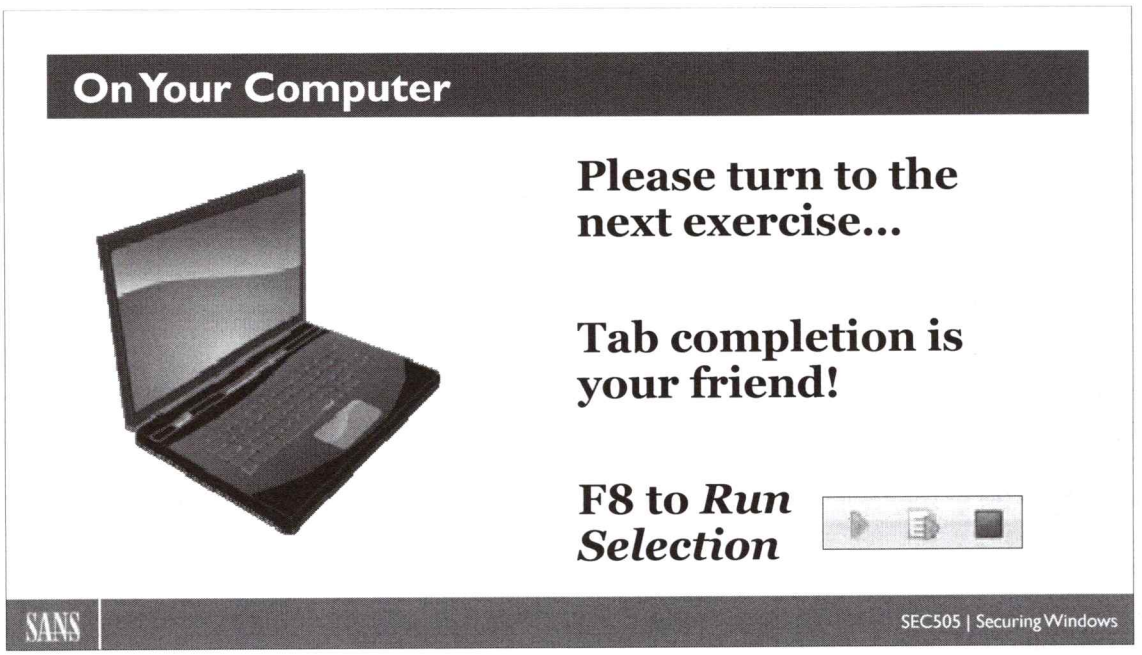
Structured Exception Handler Overwrite Protection (SEHOP) is not a compiler option, it is a runtime defensive strategy implemented by the operating system. SEHOP aims to thwart a class of stack-based buffer overflow exploits which attempt to get around the protections provided by structured exception handling generally and the /SAFESEH compiler option specifically.

SEHOP is enabled by default in Server 2008 and later server-based operating systems, but is disabled by default in client operating systems for backwards compatibility reasons. It cannot be enabled globally in Windows 2000/XP. SEHOP is enabled system-wide by editing the registry value named DisableExceptionChainValidation (as described in KB956607). It can also be enabled with EMET for specific applications, even on XP/2003.

To enable SEHOP, set the following registry value and reboot:

> Key: HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\Kernel
> Value: DisableExceptionChainValidation
> Type: DWORD (32-bit)
> Data: 0

The vast majority of applications are compatible with SEHOP, but we'll need to know for sure. The best way to test an application for SEHOP compatibility is to install and run the application on a machine with SEHOP enabled. If a rare application is incompatible with SEHOP, you don't have to disable SEHOP globally, you can disable SEHOP on a per-application basis in the registry through Group Policy (and then give that vendor grief for their bad coding practices).



On Your Computer

Install EMET silently:

```
msiexec.exe /i c:\sans\tools\emet\emet-setup.msi /qn /norestart
```

Note: In the msiexec.exe command above, please use the full path to the MSI file.

Switch to the EMET application installation folder:

```
cd 'C:\Program Files (x86)\EMET 5.5'          #Same version?
```

Note: Your installation of EMET might use a more recent version number.

View the starter configuration XML files which come with EMET:

```
dir '.\Deployment\Protection Profiles'
```

Import one of the starter XML configuration files into EMET:

```
.\EMET_Conf.exe --import '.\Deployment\Protection
  Profiles\Popular Software.xml'
```

Note: Your version of EMET may come with different XML configuration files by default. Choose any XML file for this lab. Ignore any red text warnings.

Configure system-wide settings, such as SEHOP and ASLR (ignore red warnings):

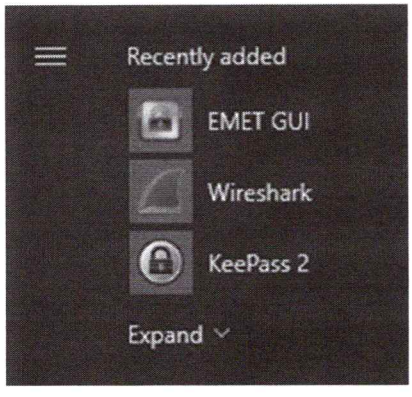
```
.\EMET_Conf.exe --system SEHOP=AlwaysOn

.\EMET_Conf.exe --system ASLR=ApplicationOptIn
```

View your current EMET settings inside PowerShell:

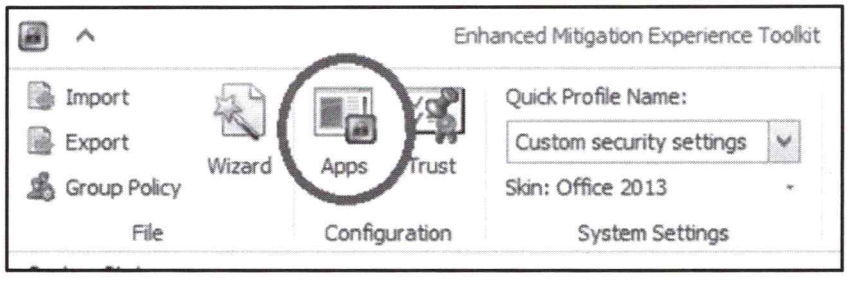
```
.\EMET_Conf.exe --list
```

EMET GUI

View your current EMET settings in the graphical EMET application by right-clicking the grey lock icon for EMET in the notification area of the taskbar (next to the clock) and selecting Open EMET. You may have to click the up-arrow button ("^") to see it.



In the EMET graphical tool, click the **Apps** button in the toolbar to see the application protection settings imported from the XML file.



In the GUI EMET application, now that you are viewing the App settings, click the "Add Application" button in the toolbar > select "C:\Program Files (x86)\KeePass Password Safe 2\KeePass.exe" > Open > OK. KeePass has been added to the list of protected applications for which you could (un)check different mitigations as necessary.

In PowerShell, export your current EMET settings to a new XML configuration file:

```
.\EMET_Conf.exe --export .\MySettings.xml
```

> **Note:** In the following command, the folder path to the program is just an example, the folder and program file do not need to exist for this lab.

Add a new application to the current settings being enforced, but disable DEP for it:

```
.\EMET_Conf.exe --set --force "*\folder\program.exe" -DEP
```

> **Note:** The --force switch allows the protection setting to be added even if the path or program executable is not currently installed. You can only use wildcards in the folder portion of the path, not in the name of the executable. Ignore any warnings in red fonts.

Delete all application settings, except those applied through Group Policy:

```
.\EMET_Conf.exe --delete_all
```

If you list your EMET settings again, notice that they have been deleted:

```
.\EMET_Conf.exe --list
```

Group Policy EMET

Recall that a GPO Administrative Template (.ADMX file) can be loaded to display more yellow containers and settings in a GPO than exist by default. An ADMX template uses strings for the user's language by loading these strings from an associated ADML file. Both files need to be placed in the correct folders on the local computer before the desired settings can be seen while editing a GPO.

Prepare to manage EMET through Group Policy by copying the necessary ADMX and ADML files into the folders used by the Group Policy Management console:

```
copy '.\Deployment\Group Policy Files\EMET.admx'
  $env:WinDir\PolicyDefinitions

copy '.\Deployment\Group Policy Files\EMET.adml'
  $env:WinDir\PolicyDefinitions\en-US
```

> **Note:** Please note the "en-US" at the end of the second command above. If your VM's language locale is not US English (en-US), then copy the ADML file into the appropriate directory named for your locale.

Open the Group Policy Management console and navigate to Forest > Domains > testing.local > right-click the Default Domain Policy GPO > Edit > Computer Configuration > Policies > Administrative Templates > Windows Components > EMET.

In the EMET container of the GPO, right-click "Default Protections for Popular Software" > Edit > select Enabled > OK.

Refresh Group Policy in your VM and wait a minute:

```
gpupdate.exe /force
```

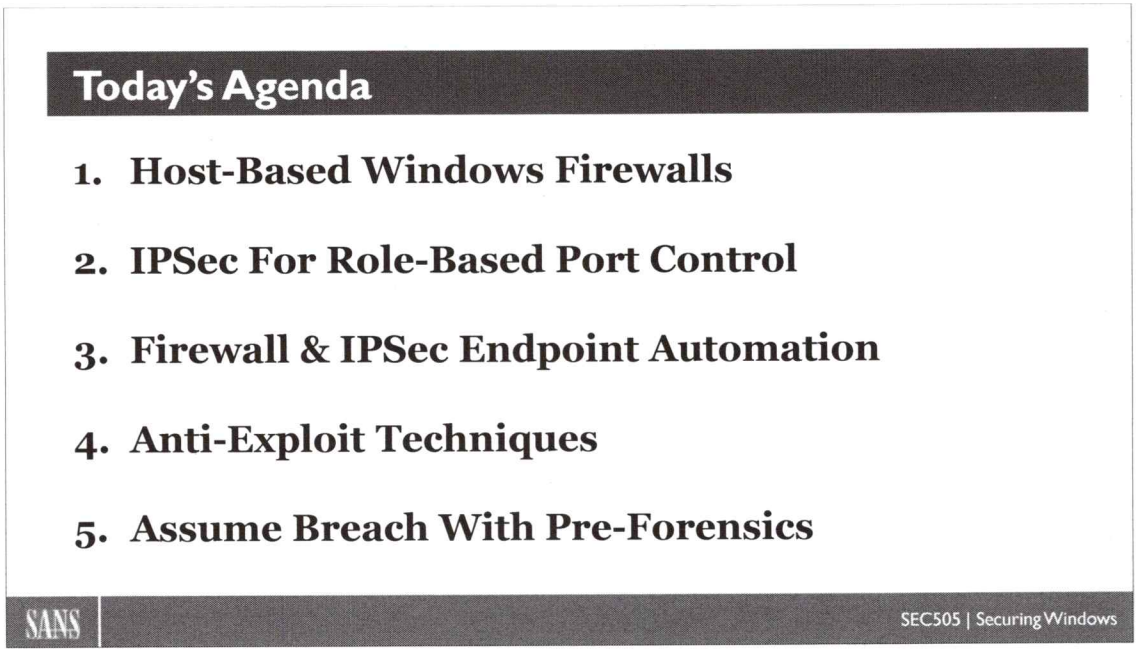
In PowerShell, view the current EMET settings and notice that the list of registry settings is blank (we deleted those) but the list of GPO mitigations is long:

```
.\EMET_Conf.exe --list
```

If you open the EMET GUI application and click the Apps button, the list will be empty. The EMET GUI only shows the current EMET settings stored in the registry, not the settings applied through Group Policy.

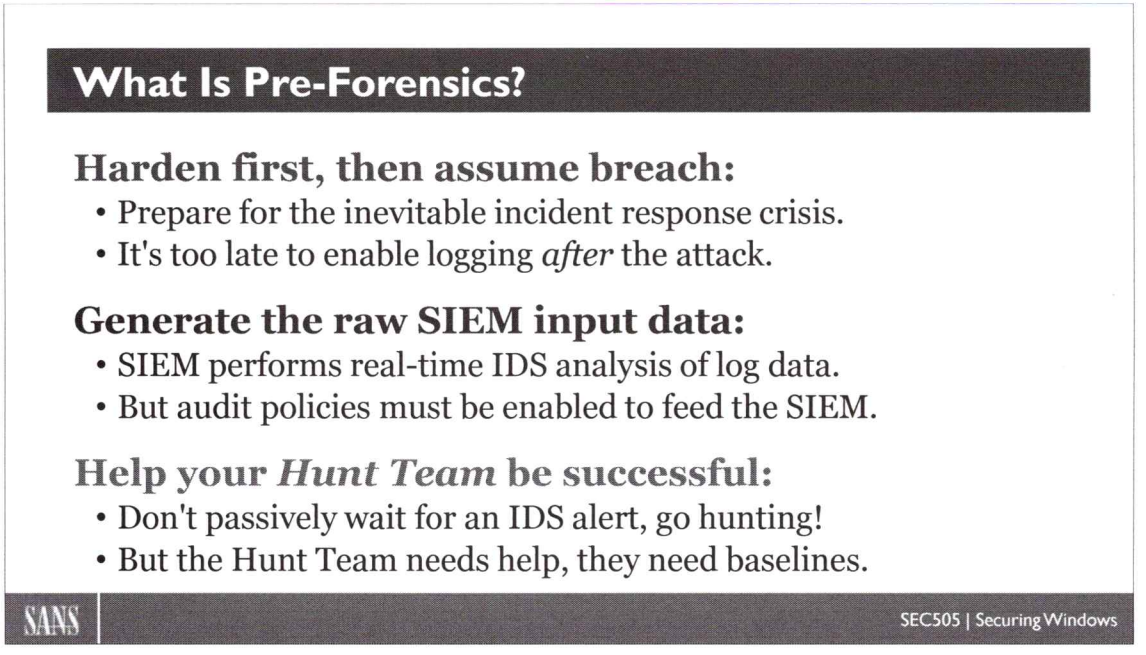
The GPO mitigations were loaded from the local 'Popular Software.xml' file, not downloaded from the domain controller. By replacing this XML file on each client computer, you can change which application mitigations are enforced. There is a Group Policy preference which can copy a file from a shared folder to overwrite another file on all the GPO clients.

EMET changes made through Group Policy will not take effect until after the next reboot or until after "EMET_Conf.exe --refresh" is executed, whichever comes first. Group Policy EMET settings take precedence over locally-configured EMET settings whenever there is a conflict.



Today's Agenda

We do the best we can to secure our networks against compromise, but we have to assume our defenses will eventually fail. Pre-forensics is what we need to do prior to a compromise to assist with forensics and incident response. Pre-forensics also includes steps to help the Hunt Team discover signs of active threats, hopefully before any further harm can be committed.



What Is Pre-Forensics?

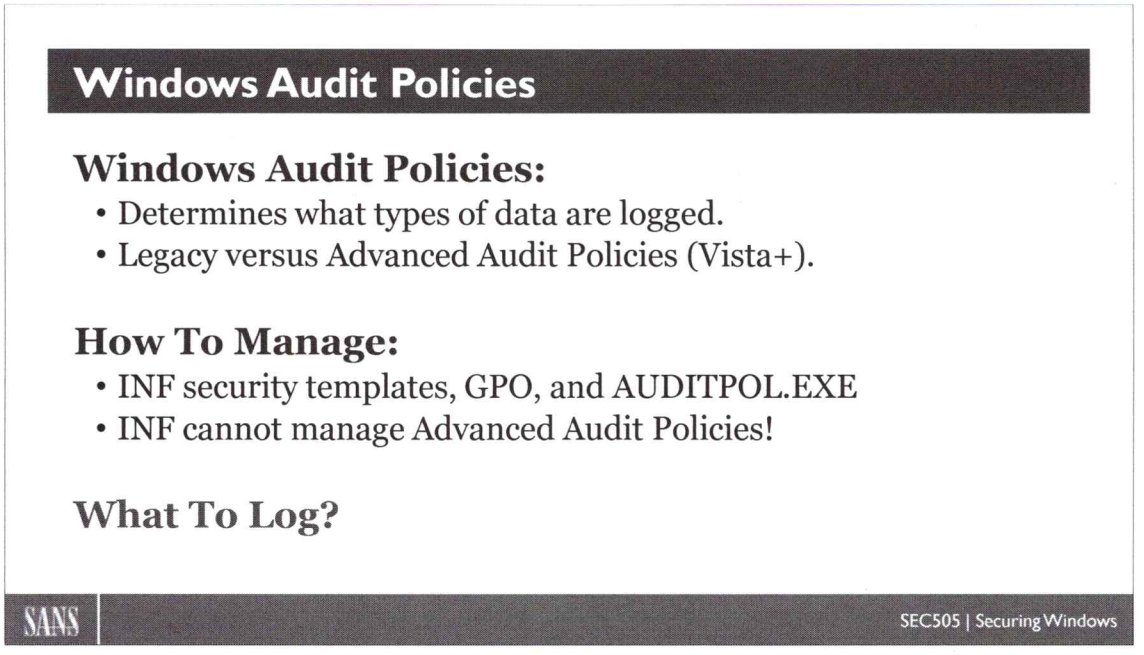
This is not a forensics, incident response, or IDS course. SANS has excellent courses on all of these topics, but each of these requires a full week of training, so we just can't cover everything here.

"Pre-forensics" is everything that should be done on Windows first during the hardening phase to assist with later forensics, incident response, and intrusion detection. It's the standard formula for network security: prevention first, detection second, incident response third. It's now, during the prevention or hardening step, that we prepare and facilitate the steps which follow.

We prevent as much harm as possible, then assume there will be a breach anyway. This is just prudence, not pessimism. You drive as safely as you can, but you still have air bags and seat belts just in case. If we have poor prevention habits, then our forensics and IR teams will be overwhelmed. Prevention reduces the potential damage we could suffer as much as practical, then IDS, forensics and incident response fill in the gaps. Together, we all aim for the same thing: reducing harm.

Your Security Information Event Management (SIEM) system can create dashboards and real-time IDS alerts, but only if it is fed the raw data it requires to analyze. If the right kinds of log data are not generated in the first place, then the SIEM will be blind to those aspects of the network. In this case, we are focusing on Windows endpoints and servers, but your SIEM will need to be fed from many sources, such as your firewalls and routers too.

When we assume there will be inevitable breaches, we must also assume that our IDS and SIEM products will be blind to some of those breaches. Hence, to help fill in the gaps in our IDS/SIEM coverage, we should also pre-emptively go hunting for signs of undetected breaches in our environments. The "Hunt Team" are those security personnel who look for things which our IDS/SIEM systems cannot see. This is a hard job! Pre-forensics also includes tasks to make it easier for the Hunt Team to be successful. For example, if the Hunt Team has no pre-compromise or "clean" baseline to compare against, then it is hard to find indicators of hacker or malware activity post-compromise. The Hunt Team mainly uses Big Data tools to find the needles in the haystack, but there first has to be haystack of data to search!



Windows Audit Policies

To detect and respond to attacks, log data of various types must be generated, sifted for alerting, consolidated, and analyzed. This includes log data from all sources, including firewalls, IDS sensors, RADIUS servers, DNS, domain controllers, IIS, Exchange, and Windows. This course only covers Windows, but it is expected that other logging sources will be appropriately configured and consolidated too.

Windows Event Logs

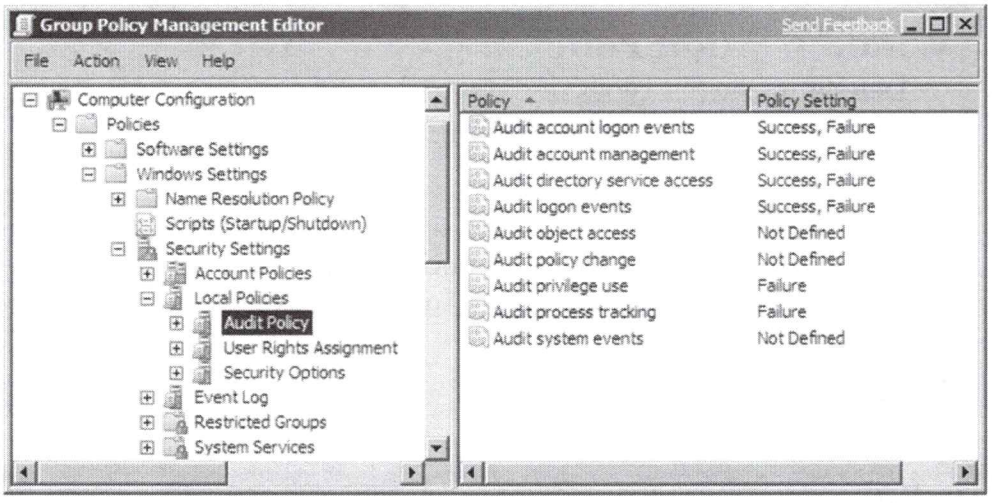
In Event Viewer there are the three standard logs: System, Security and Application. The System log is for events related to operating system performance. It is mainly used for troubleshooting. The Security log is for recording authentication events, access to resources, invocations of user rights, and other items of interest for intrusion detection and incident response. The Application log is set aside for any other events application developers want, it's the everything-else category.

Windows provides good auditing capabilities, but they are not enabled by default. Facilities for centrally collecting and managing this audit data --like something akin to a Windows Syslog-- are sorely lacking (and the event forwarding capability isn't enough). This section is concerned only with enabling the auditing; managing the log data almost always requires a third-party product; and real-time analysis and alerting always requires another product, it cannot be done by hand.

Basic Audit Policy

All audit policy settings on both servers and workstations can be remotely configured through Group Policy. The older, basic audit policy options are set under Computer

Configuration > Policies > Windows Settings > Security Settings > Local Policies > Audit Policy.



Auditing is enabled for the success and/or failure of certain types of events. However, these audit policy settings are mainly for Windows XP, Server 2003, and earlier operating systems. Current operating systems may use these policies, but it is better to focus on the advanced audit policies instead (see below). Nonetheless, the basic audit policies still work, so the following is a list of the recommended minimal audit policies to enable, if you are going to use them:

- **Audit Account Logon Events** (Success, Failure)-- This tracks authentication requests processed by the domain controllers even when the access is not to the domain controller itself. These authentication requests are sent by users' machines when a user logs on locally at those desktops, and by servers when a client logs on remotely to those servers, e.g., when a user maps a drive letter to a server's shared folder. Think of DCs as providing a service for the sake of other machines on the network (checking usernames and passphrases) and this category logs whenever that service is provided. When this policy is enabled on the workstations and servers themselves, then it applies only to the local accounts on those machines, hence, it only applies to authenticated access to those machines. Strictly speaking, this audit policy logs information at the machine where the account in use is physically stored, i.e., either in the global AD database on a DC or in the local SAM database on a member server.

- **Audit Account Management** (Success, Failure)-- This monitors user and group tasks such as account creation, deletion, modification, and group membership changes. Note that only the bare fact that an account or group has been modified will be logged through this policy, not the detailed data made available with "Audit Directory Service Access".

- **Audit Directory Service Access** (Success, Failure)-- This is required to begin logging access to AD objects as defined on those objects' individual SACLs. By analogy, NTFS SACLs also do not cause data to be written to the Event Log unless the "Audit Object Access" policy is enabled.

- **Audit Logon Events** (Success, Failure)-- This tracks interactive and over-the-network logons to the target computer itself. Strictly speaking, it logs information on the machine where the Security Access Token (SAT) is created for the local or remote user that is performing the logon, but the SIDs for that SAT do not all have to come from the target computer itself: the global SIDs will come from a DC and the local SIDs will come from the target machine, but the target machine is still the location where the SAT is created.

- **Audit Object Access** (Success, Failure)-- This is required to begin logging access to NTFS folders and files, registry keys, and shared printers. It is not the case that enabling this category will cause all filesystem, registry and printer access to be logged. Rather, enabling the category makes it possible to have the audit ACLs (the System Access Control Lists) on those objects take effect. It takes two changes to audit access to a file, for example; this category must be enabled and that particular file must be configured to audit some kind of access to it.

- **Audit Policy Change** (Success, Failure)-- Tracks changes to the audit policies themselves, and changes to user rights assignments.

- **Audit Privilege Use** (Not Defined)-- Monitors the exercise of the various user rights on the machine, e.g., take ownership, change system time, etc. Enable this policy on an as-needed basis only to avoid filling the logs.

- **Audit Process Tracking** (Not Defined)-- This is rarely enabled, and usually only by programmers who are debugging their own code. This category tracks program execution, process loading and unloading, filesystem handle creation and release, indirect object access, and other low-level OS behaviors. Enabling this category will cause a vast amount of extra log data and will slow the system down considerably.

- **Audit System Events** (Success, Failure)-- Tracks system startup, shutdown, and other system-wide events. This also records clearing of the System and Security logs.

Advanced Audit Policy Configuration

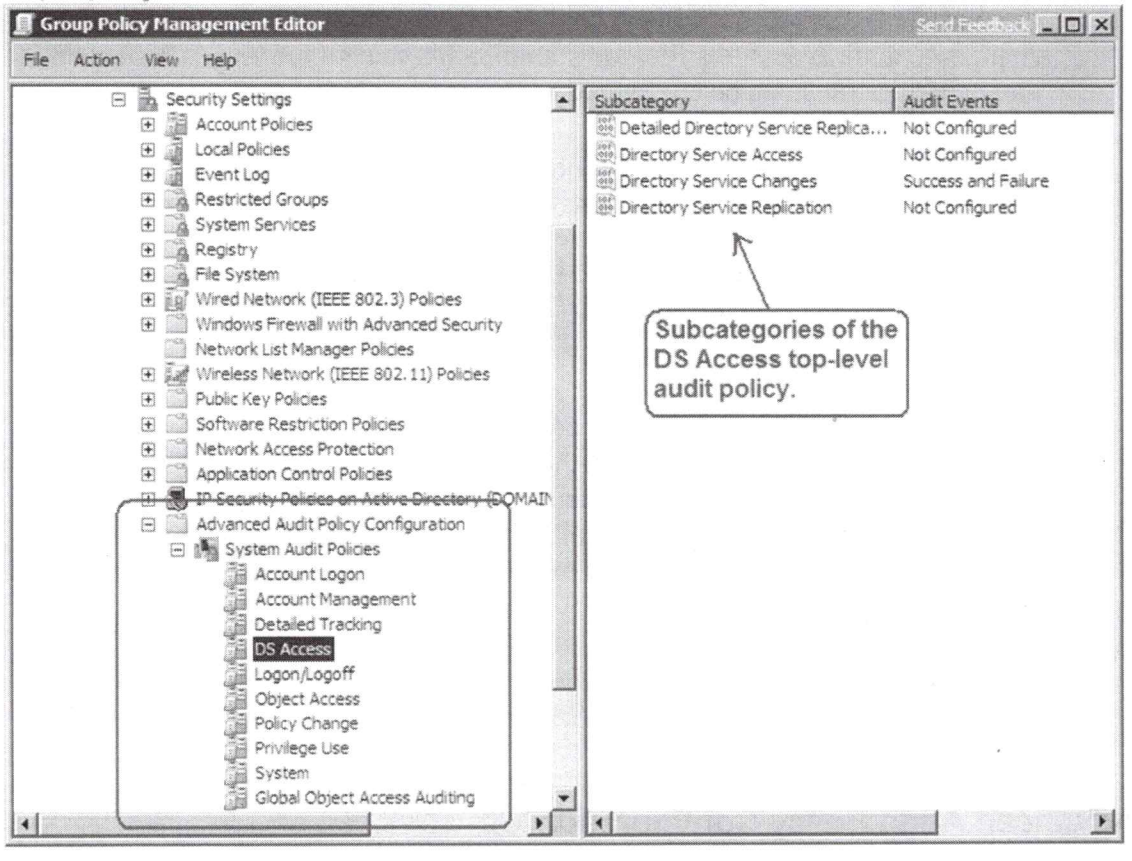
In Server 2008, Vista and later, the basic top-level audit policy categories (above) have been broken down into subcategories, and each subcategory of audit policy can be enabled or disabled separately from the other subcategories. This is much more precise, and this approach to auditing is the recommended method instead of using the basic audit settings.

Indeed, the recommendation is now to simply ignore the basic audit policies when possible and only use the advanced audit policies. To make sure there are no conflicts or confusion between the basic and advanced audit policies, make sure to enable the following GPO setting: Computer Configuration > Policies > Windows Settings > Security Settings > Local Policies > Security Options > Audit: Force audit policy subcategory settings (Windows Vista or later) to override audit policy category settings.

The above GPO option will ensure that the basic audit policies are ignored.

Managing advanced audit settings can be done from the command line with AUDITPOL.EXE or through Group Policy. Note that using Group Policy for this requires Server 2008 R2, Windows 7 or later; on Server 2008 and Vista only, you must still use AUDITPOL.EXE.

To manage advanced audit policies via GPO, navigate to Computer Configuration > Policies > Windows Settings > Security Settings > Advanced Audit Policy Configuration.



Be aware that many tools designed for Windows security do not correctly identify the active audit policies on a computer, including some of Microsoft's own tools. When in doubt, use AUDITPOL.EXE to view the audit policies currently live on a machine.

To list your current audit policy configuration with AUDITPOL.EXE:

```
auditpol.exe /get /category:*
```

For reference, here is a listing of the advanced audit policy subcategories:

System:

- Security System Extension
- System Integrity
- IPsec Driver
- Other System Events
- Security State Change

Logon/Logoff:

- Logon
- Logoff
- Account Lockout
- IPsec Main Mode
- IPsec Quick Mode
- IPsec Extended Mode
- Special Logon
- Other Logon/Logoff Events
- Network Policy Server

Object Access:

- File System
- Registry
- Kernel Object
- SAM
- Certification Services
- Application Generated
- Handle Manipulation
- File Share
- Filtering Platform Packet Drop
- Filtering Platform Connection
- Other Object Access Events

Privilege Use:

- Sensitive Privilege Use
- Non Sensitive Privilege Use
- Other Privilege Use Events

Detailed Tracking:

- Process Termination
- DPAPI Activity
- RPC Events
- Process Creation

Policy Change:

Audit Policy Change
Authentication Policy Change
Authorization Policy Change
MPSSVC Rule-Level Policy Change
Filtering Platform Policy Change
Other Policy Change Events

Account Management:

User Account Management
Computer Account Management
Security Group Management
Distribution Group Management
Application Group Management
Other Account Management Events

DS Access:

Directory Service Changes
Directory Service Replication
Detailed Directory Service Replication
Directory Service Access

Account Logon:

Kerberos Service Ticket Operations
Other Account Logon Events
Kerberos Authentication Service
Credential Validation

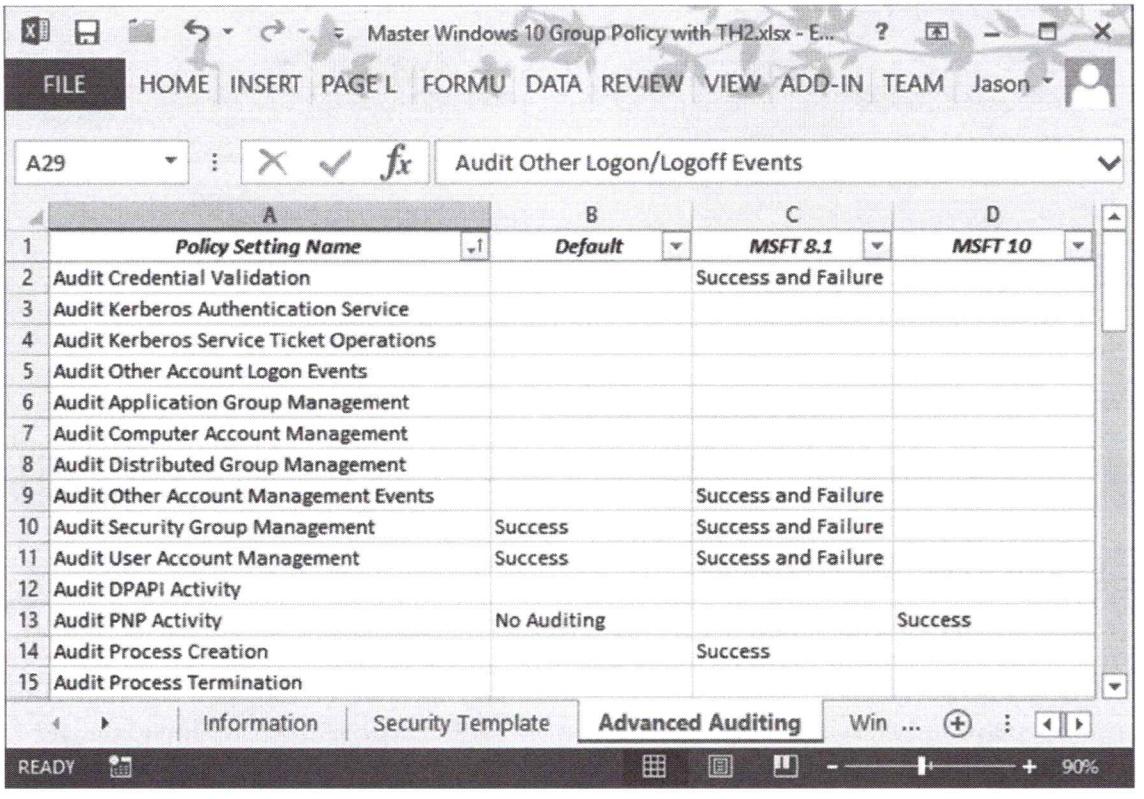
What Should Be Logged?

What should be logged? This is not easy to answer. Too much logging slows down the system, wastes bandwidth, and consumes drive space unnecessarily. Too little and we may be blind to the activities of hackers and malware. And different types of machines require different audit policies, e.g., different policies for non-classified laptops versus domain controllers.

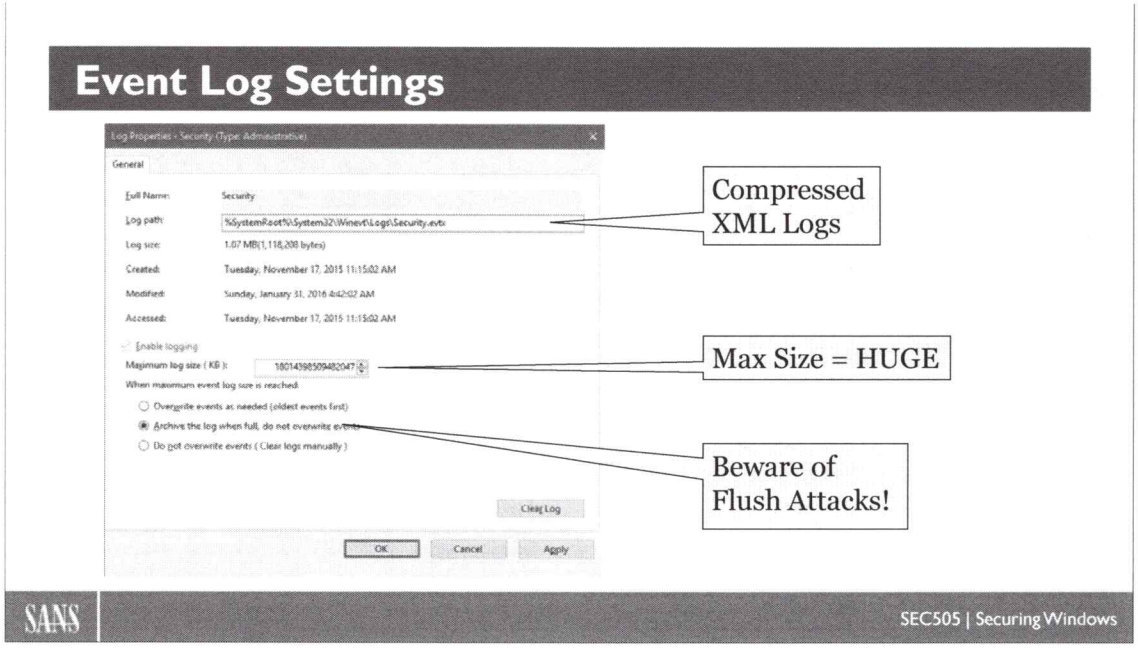
So, as discussed earlier in the course in the section on security templates, download the latest version of Microsoft's Security Compliance Manager (SCM) and choose the baseline most appropriate for each of your different types of systems. Microsoft tries to strike a balance between visibility and impact, so these baselines are a good place to start. Don't worry, you can always log more categories of information later, but there are too many baselines and too many settings to list here.

To download the latest version of the Security Compliance Manager (SCM), just do a search on the name at https://www.microsoft.com/security.

For example, an SCM security baseline will include a spreadsheet of security settings, including the advanced audit policies. Here is a screenshot from the security baseline spreadsheet for Windows 10.

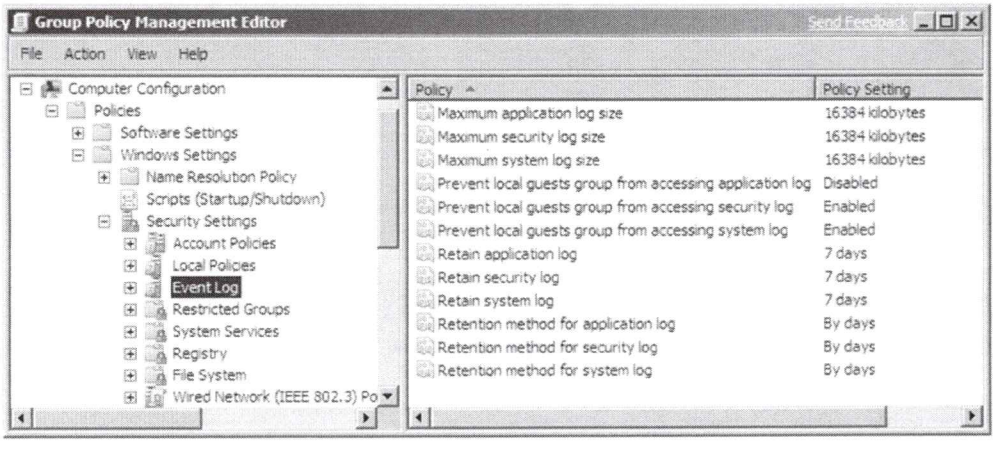


	A	B	C	D
1	*Policy Setting Name*	*Default*	*MSFT 8.1*	*MSFT 10*
2	Audit Credential Validation		Success and Failure	
3	Audit Kerberos Authentication Service			
4	Audit Kerberos Service Ticket Operations			
5	Audit Other Account Logon Events			
6	Audit Application Group Management			
7	Audit Computer Account Management			
8	Audit Distributed Group Management			
9	Audit Other Account Management Events		Success and Failure	
10	Audit Security Group Management	Success	Success and Failure	
11	Audit User Account Management	Success	Success and Failure	
12	Audit DPAPI Activity			
13	Audit PNP Activity	No Auditing		Success
14	Audit Process Creation		Success	
15	Audit Process Termination			



Event Log Settings

Event Log settings mainly concern the size and wrapping options for the separate log files themselves. To get to the dialog box in the slide, open Event Viewer > right-click a log > Properties. In a security template or domain GPO, these settings are found under Computer Configuration > Policies > Windows Settings > Security Settings > Event Log.



Log Size

Each log is finite in size. The larger the log size the more events it can hold before the wrapping options engage. On Server 2008, Vista and later, the logs have been transformed into compressed XML files with more-or-less unlimited maximum sizes: more than 16 million terabytes (16384PB)!

On Server 2000/2003 and Windows XP, even though a log can be set to a maximum size of 4.2GB in the interface, the log will not grow larger than about 300MB (KB183097). There is no 300MB artificial limit for Vista/2008 and later boxes.

Wrapping Options

When a log file fills to its maximum capacity, that log file's wrapping options engage. There are three wrapping options:

- Overwrite Events As Needed
- Overwrite Events Older Than *X* Days (number of days is configurable)
- Do Not Overwrite Events (Clear Log Manually)

If you do not choose to Overwrite Events As Needed and the log fills up, then new events will simply not be written. The computer will continue operating normally, unless the CrashOnAuditFail option is enabled (see below).

Recommended Settings for Security

Ensure that there is adequate free space in the Boot partition (the one containing the operating system files) for both the log files and a paging file. Strongly consider placing additional paging files in other partitions, e.g., a RAID 0 volume dedicated to the paging file is ideal, but expensive. As a rule of thumb, the maximum size of the paging file in the Boot partition should be equal to the amount of RAM installed plus 50 megabytes. The combined maximum sizes of the log files should not be greater than the amount of hard drive free space after subtracting the amount of space used by the paging file when the paging file is at its maximum size.

The appropriate size of the log files depends upon one's security policy and the wrapping options used.

The wrapping option to Overwrite Events As Needed should not be used. This option allows an attacker to flush out log files with meaningless entries. After the intruder causes damage or steals data, he can launch a batch file which endlessly loops as it attempts to perform a denied action; the Security log will then overflow with access denied entries, flushing out the entries which audited the original damage or theft.

In most environments, the option to Overwrite Events Older Than *X* Days is the best choice. The number of days set should correspond to the log's backup/export schedule. This schedule is in turn determined by the maximum size of the log and the rate at which the log becomes filled. Avoid the situation where a log fills to capacity before it has been exported to a storage server or backed up. Each event should be backed up twice before being purged, i.e., if you make a backup once every seven days, then your log size should be set large enough so that wrapping does not occur until after about 15+ days. If your logs fill up faster than that, you'll need to make more frequent backups.

The option to Do Not Overwrite Events is preferred in very high security environments. This option requires the log to be manually cleared before the log fills to maximum

capacity and logging ceases. Clear the log just after it has been exported or backed up. This option might be used with the CrashOnAuditFail registry setting (see below).

> **Note**: Windows Server 2003 and later can be configured with custom Read, Write and Clear permissions on each of its Event Logs to regulate who can perform these actions. This is configured through registry edits (KB323076).

The Security log is the most important and should have a relatively large size and be archived frequently to prevent its overflow. There are many variables which must be considered when choosing a log file size, including:

- Free space on the partition.
- Average rate at which the log fills.
- Export/backup schedule for the log.
- Audit options on the SACLs of objects.
- Security policy requirements of one's environment.

These variables must be considered and configured together as a set.

If you set the AutoBackupLogFiles registry value (KB312571) and an event log fills to maximum size, then Windows will automatically create an archive file of the event log and then clear the working log. Beware that these archive files will accumulate until either you move/delete them or the server runs out of free hard drive space. If you have the money for centralized archival of event log data, hopefully with host-based IDS monitoring too, this would be far better.

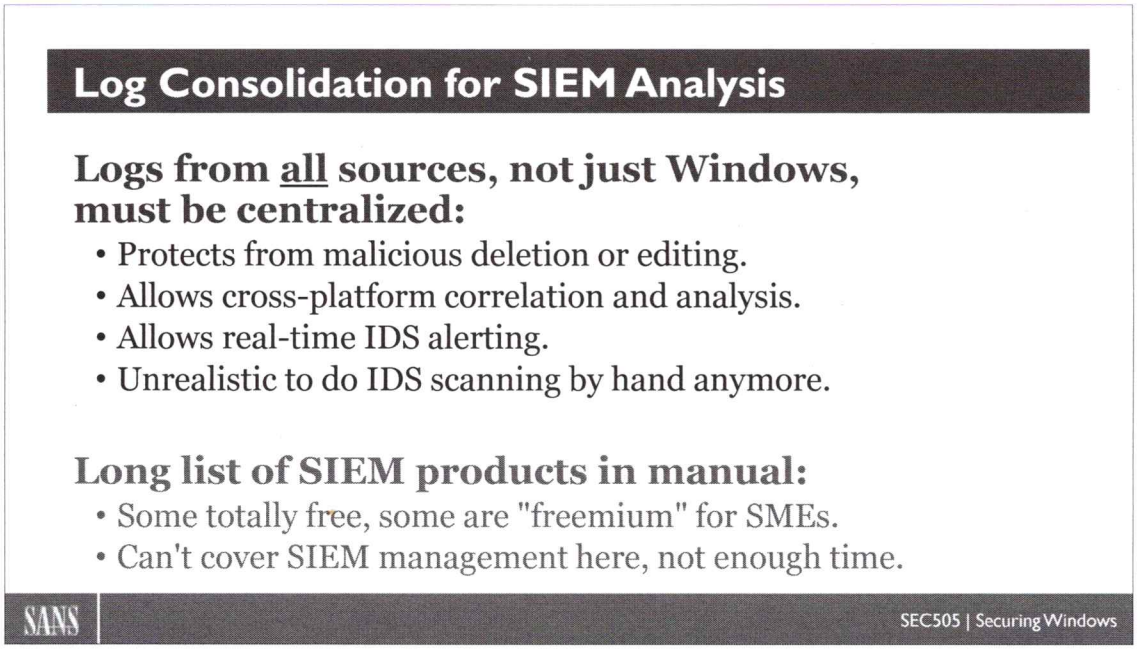
Shut Down System Immediately If Unable To Log Security Audits

When a log file reaches its maximum size, and the option to Overwrite Events As Needed is not used, then the system continues to function normally except that no new events are written to the overflowed log. Logging stops but the server stays up and running.

In very high security environments it may be undesirable for the server to remain operational when no logging is occurring. On these networks, it is preferable for a server to shut down than to operate without auditing.

When a server does shut down because of this setting, a Blue Screen of Death (BSoD) will occur. When the server is rebooted, only an administrator will be able to log on (interactively or over the network). The administrator should then back up the Security log, clear it, and reboot the system.

> **Important:** If an attacker knows that this option is enabled, then the attacker can use it for Denial of Service attacks!



Log Consolidation for SIEM Analysis

Ideally, as local log data is being produced, copies of the log data would be sent over the network to a centralized log consolidation, archival, alerting and analysis server. Hackers and malware might attempt to disable logging and delete all log data after compromising a machine, so ideally the log event copies would be sent over the network in real time, though it is common to batch up new log entries for at least a few minutes first.

There is no Syslog equivalent built into Windows (the subscriptions feature in Event Viewer doesn't count), so installing a third-party log consolidation product is mandatory. There are both commercial and FOSS options available. Whatever product is chosen, though, we want it to scale to one's enterprise, not consume excessive bandwidth, not slow down monitored systems too much, to provide flexible alerting capabilities, and to make it as easy as possible to perform forensics analysis and speedy incident response.

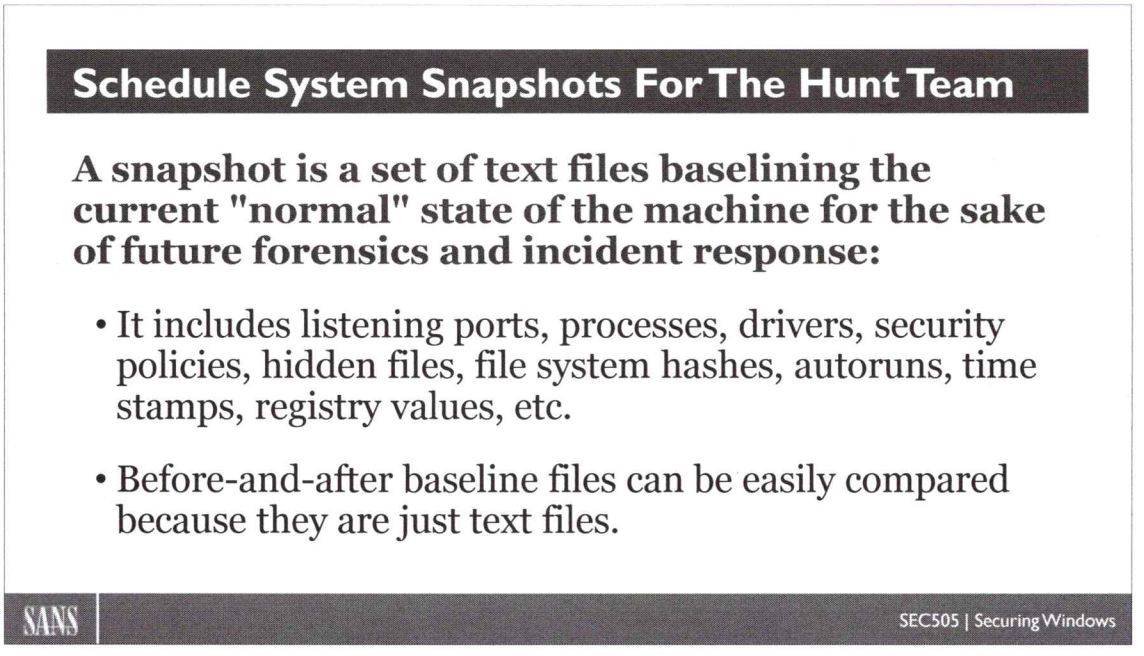
The product should also include a Security Information Event Management (SIEM) console, or at least be compatible with one's existing SIEM. Many products in this space are a combination of log consolidator and SIEM analytics in one package.

Having a SIEM is essential for receiving near real-time IDS alerting. As logs from all sources, not just Windows, are consolidated, the SIEM can perform pattern matching to look for indications of attacks, compromise, malware infection, or post-exploitation actions. Today, it is totally unrealistic to attempt to perform this cross-platform, multi-source, real-time IDS analysis by hand. There is just too much data streaming in and the patterns to identify themselves change too frequently (these patterns are "fuzzy" heuristic or statistical patterns, not simple signatures). Fortunately, there are some great SIEMs which are free!

Here are a few log consolidation products to consider, though this is not intended as an exhaustive list or as specific recommendations:

- BlackStratus Storm (www.blackstratus.com)
- CorreLog (www.correlog.com)
- Dell InTrust (www.dell.com)
- Elastic (www.elastic.co [no "m"])
- HP ArcSight (www.hp.com)
- IBM Q1 Labs (www.q1labs.com)
- LogRhythm (www.logrhythm.com)
- McAfee Enterprise Security Manager (www.mcafee.com)
- NetIQ Sentinel (www.netiq.com)
- OSSEC (www.ossec.net)
- SolarWinds Log & Event Manager (www.solarwinds.com)
- Splunk (www.splunk.com)
- Symantec Security Information Manager (www.symantec.com)
- TIBCO LogLogic (www.tibco.com)
- TripWire SIEM (www.tripwire.com)
- TrustWave SIEM (www.trustwave.com)

Note: The SIEM space is changing rapidly. If your favorite SIEM is not on the list above, or if a product no longer exists, please let the instructor know!



Schedule System Snapshots For The Hunt Team

A *system snapshot* is a collection of data that documents the configuration and running state of the machine at a point in time. Its purpose is to provide a baseline against which later snapshots can be compared in order to detect changes. Presumably we'll have a *before* snapshot, when we assume the machine is working fine and has not been compromised, and an *after* snapshot, taken after problems began or after a suspected compromise, or just because it's time for another audit. The *before* and *after* snapshots can be compared so that only the differences are listed. This process is useful for troubleshooting, intrusion detection, incident response and forensics. When you are on the "Hunt Team" looking for signs of intrusion or compromise on Windows machines, these snapshots are golden!

Can't We Just Use Our Backups?

An ideal snapshot of a server, you would think, would be a binary image backup of its hard drive. This is often required for forensics, but binary images and backup archives suffer from a few shortcomings: they produce too much data to be stored easily; that data cannot be compared against other such snapshots quickly or easily; and analysis usually requires special forensics tools and training. Fortunately, we are already making regular backups of our critical servers. So whatever advantages these backups provide for auditing, we should view them as fulfilling this role in addition to providing disaster recovery.

Because the purpose of a snapshot is to provide a baseline for comparison, we want to capture data in a form which is easily compared against other snapshots; we want that data to be highly compressible for storage; and we want to be able to automate the entire snapshot-making process. What fits the bill? Plain text files produced by custom scripts

and command-line tools are perfect for system snapshots. Through scripts and command-line tools you can gather almost any type of data you wish. It's trivial to redirect this data to a text file, and large text files can be compressed to a fraction of their original size.

Store your textual snapshot files in an NTFS folder with compression enabled or script a tool like 7-Zip to compress the snapshot files into a single archive (www.7-zip.org). The NTFS compression will be transparent to your other auditing tools, while archive files will need to be extracted first.

How to Structure the Data

The purpose of making a snapshot is to have it for later comparison with prior snapshots; hence, the names of the snapshot files and their internal structure should be geared towards this end. You can have one large snapshot file per server, or you might prefer creating one folder per server with the contents of the snapshot separated into multiple files.

The names of snapshot files might include the computer's name and the date. For example, a snapshot file might be named *SERVER47-2014-08-29.txt*. The snapshot data should include a file (like README.TXT) that includes the computer's name, the date and time, the script used to create the snapshot, the username and domain of the person running the script, and any other identifying information you'll later need.

The snapshot files should be labeled uniquely and standardized across as many of the snapshots as possible. In practice, this means you should try to use the same script each time. Again, anticipate how file-comparison tools or other auditing scripts will later try to use this data, hence, make your snapshots as "digestible" as possible to software, i.e., make it well-formatted and standardized.

Snapshot Contents

Ideally, a snapshot should include all the information necessary to help discover precisely what changes an expert-level intruder with administrative privileges has covertly made to your system. This is almost impossible, but it's the goal to shoot for. Hence, a good snapshot should include:

- All local user accounts, with as many of their properties as possible.
- All local groups and their memberships, especially the local Administrators group.
- Shared folders, their local paths, and their permissions.
- Local audit, lockout, and passphrase policies.

- List of all user rights and the various users/groups who have these rights on the machine.
- List of running processes and their properties.
- List of drivers and their properties.
- Running services, and the startup settings (Automatic, Manual or Disabled) of all services, running or not.
- All networking configuration settings, including IP addresses, the route table, NetBIOS names, DNS/WINS servers, IPSec configuration, etc.
- Environmental variables.
- The entire registry, or at least the keys which control services, drivers, and automatically executed commands.
- List of all files, or at least the files in %SYSTEMDRIVE%, including their sizes, last-modified dates, and file attributes (especially the hidden files).
- List of all folders with the number of files in each folder and the total number of bytes consumed by all the files in each folder.
- Dump of all NTFS permissions, or at least the NTFS permissions of everything under %SYSTEMROOT%.
- If SQL Server is installed, then include lists of all the users and groups who occupy the various "roles" in SQL Server, especially the SysAdmin role.
- The IIS XML configuration files.

Comparing Snapshot Files

The whole point of gathering all this snapshot and logging data is to be able to detect covert changes to our systems and explain how the changes were made. Detection and analysis enables you to stop the spread of further damage, to hopefully repair the damage that already has been done, and to learn what vulnerability made it possible in the first place so that you can prevent it from happening again.

If hackers have formatted your hard drives, this is easy to detect, but other changes will be invisible unless you specifically look for them. This is where comparing the current snapshot against earlier ones really helps. What's the best way to do the comparison? You always can do an "eyeball audit" with two copies of Notepad side-by-side, a snapshot in each. Fortunately, there are tools that can compare two similar text files and print only their differences. One of these is PowerShell's Compare-Object. Another

built-in tool is FC.EXE. Both tools can take two files, compare them, and prints each set of mismatches.

A graphical version of FC is WINDIFF.EXE from Microsoft (and it can work from the command-line too). WINDIFF highlights the mismatching lines from the files in different colors, and you can jump back and forth between mismatches easily. It's like FC and LIST combined into one tool. A similar tool with more features is CSDIFF.EXE and it's free (do a search on its name for the download link).

The limitations of these tools, though, are that the snapshots must be formatted very similarly to each other if you're to avoid hundreds of mismatches, and they only can detect changes in the snapshots. But the big limitation is unavoidable: it takes a human being to understand and analyze these snapshots.

Detecting File Modifications

The script can be modified to create SHA-256 hashes of whatever files you wish, such as all the operating system and boot-up files. There is a free tool named SHA256DEEP.EXE which supports an -X switch to perform comparisons against prior hashings. For example, using the -X switch of SHA256DEEP.EXE, you can compare the current hashes of all the same files against the previously-recorded hashes. The tool will show you all the files which have changed. In fact, the tool will also show you all the new files and all the missing files too.

On Your Computer

In PowerShell, please switch to the C:\SANS\Day5-IPSec folder:

```
cd C:\SANS\Day5-IPSec
```

In this folder, open the Snapshot.ps1 pre-forensics script in ISE for perusal:

```
ise .\Snapshot.ps1
```

Note: In the next command, ignore any error messages or graphical pop-ups.

Run the Snapshot.ps1 script and wait a few minutes:

```
.\Snapshot.ps1
```

After a few minutes, depending on the speed of your computer, a new subdirectory will be created and filled with a variety of XML, TXT and CSV files. The folder is named after your computername and the current time, e.g., *COMPUTER*-2016-11-3-1-22.

See the new subdirectory created by the snapshot script:

```
dir -directory
```

Switch into the new subdirectory and list all the files created:

```
cd COMPUTERNAME-2016-11-3-1-22  #Not this folder exactly
```

```
dir
```

Some files are just flat TXT or CSV files, so they can be easily examined by hand:

```
ise MSINFO32-Report.txt

Get-Content Audit-Policy.txt
```

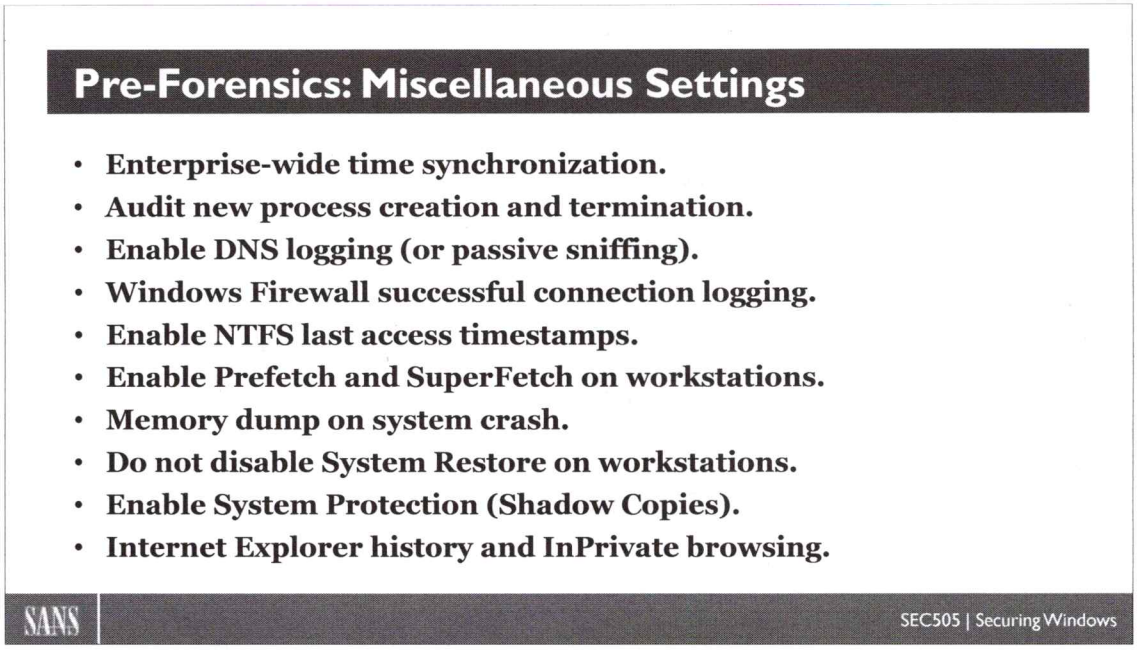
Most files are XML, which makes them easier to use when performing snapshot comparisons, but if you want to examine the data with your eyes, try these examples:

```
Import-CliXml services.xml

Import-CliXml processes.xml | out-gridview
```

And you can always convert the XML data to a CSV file or to some other text format which is easier to examine or compare (starting with XML, it's easy to convert data):

```
Import-CliXml drivers.xml |
   Select Name,Description,State,StartMode |
   Export-Csv drivers.csv

ise drivers.csv
```

What about comparing the files across multiple snapshots? These could be compared using a variety of free tools, including PowerShell's Compare-Object, Notepad++, fc.exe, diff.exe, etc. (http://en.wikipedia.org/wiki/Comparison_of_file_comparison_tools). There is even a script on your course USB\CD named "Compare-ExportedXmlFiles.ps1" for this purpose to get you started if you want to do it in PowerShell alone.



Pre-Forensics: Miscellaneous Settings

In addition to operational snapshots and conventional log data, there are more sources of information for forensics and incident response which can be enabled. But we will need to take care in configuring them. There will often be a trade-off between the potential forensics value of this data and the CPU cycles, storage space, and I/O performance impact required to capture it.

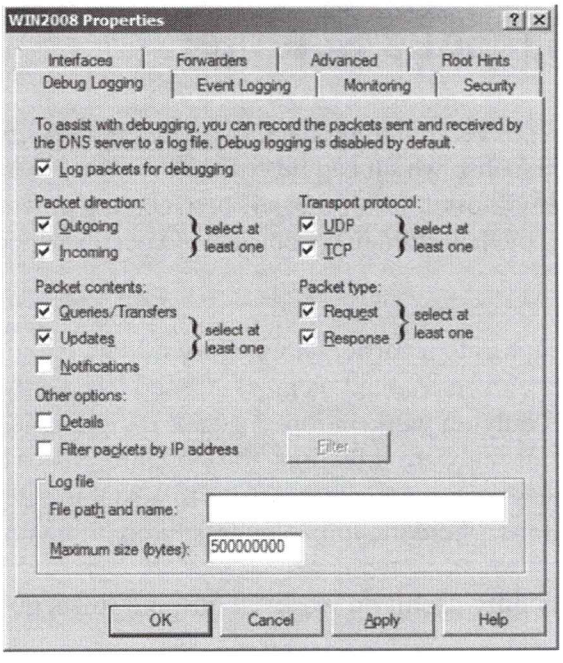
If it does turn out that the overhead of capturing a particular category of data is too great on servers, remember that these changes can be enabled on an as-needed basis only during a crisis, and might be configured only on workstations instead; for example, after discovering initial signs of compromise or malware infection, additional forensics data could be captured for the duration of the response and clean up, disabled again afterwards on servers, but left enabled on workstations where the impact is less likely to be noticed.

Some of the following settings are really only useful on workstations, but it is a good time to mention them all in any case.

Time Synchronization
Ensure that the clocks of all switches, routers, firewalls, IDS sensors, proxies, servers and workstations are synchronized and correct. Domain-joined Windows systems sync their clocks with their domain controllers by default. Further control of NTP settings can be obtained with the W32TM.EXE tool or through Group Policy (Computer Configuration > Policies > Administrative Templates > System > Windows Time Service).

Audit Process Creation and Termination
In the Advanced Audit Policy Configuration section of a GPO (discussed earlier), go to the Detailed Tracking category, and enable the logging of successful process creation events (ID 4688) and successful process termination events (ID 4689) to the Security log. This is very useful because it includes the username, executable path, PID number, and exit time of each process launched.

Enable DNS Logging
Logging DNS query requests provides a gold mine of information. To enable DNS query logging, go to the Properties of the DNS server in the DNS snap-in > Debug Logging tab > check the boxes only for Incoming, Queries/Transfers, and Request > OK. By logging only requests, the logging overhead can be reduced, but if the IP addresses returned by the DNS server are desired too, responses will also have to be logged. If the I/O impact of this logging on the DNS servers is too great, then separate network IDS sensors can be installed which passively sniff and log all DNS queries and responses. Network sensors are preferred especially when hosts are permitted to use any DNS server, internal or on Internet, such as the OpenDNS servers.



Another DNS logging option is available only on Server 2012 R2 and later called "DNS diagnostics logging" (see KB2956577). It requires a few steps to set up, and it will still impose a performance impact, but it provides more information than debug logging.

Windows Firewall Log Size and Log Successful Connections in the Domain Profile
As discussed elsewhere in the course, the Windows Firewall can be managed through Group Policy, NETSH.EXE and PowerShell cmdlets. To help track the movements of hackers and worms inside the corporate LAN, increase the size of the firewall log to at least 20MB and enable the logging of successful connections for the Domain Profile on

workstations. The logging of Internet-bound connections can be done at the perimeter firewall, but internal-only connections do not traverse the perimeter. If the I/O overhead of successful connection logging is too great, remember that this recommendation is only for workstations, and this logging might be enabled only temporarily during an incident response on both servers and workstations.

Enable NTFS Last Access Timestamps
The NTFS file system can record a last-accessed timestamp on folders and files, which is useful for creating a timeline of the activity of a hacker or a piece of malware. Enable NTFS last access timestamps by setting the following registry value to zero:

Key: HKLM\SYSTEM\CurrentControlSet\Control\FileSystem
Value: NtfsDisableLastAccessUpdate
Data: 0 [REG_DWORD]

You can also set this registry value to zero with the following command:

```
fsutil.exe behavior set disablelastaccess 0
```

On a file server with deeply-nested directories and thousands of files in each directory, this setting can impose a 5% overhead when enumerating files and folders. On a workstation, it is unlikely that a user would notice any performance penalty.

Enable Prefetch and SuperFetch
Prefetch and SuperFetch are memory management features designed to speed the bootup, logon and application launch procedures. One or both are usually disabled on servers, and SuperFetch is disabled by default on workstations with solid state drives. But these features also record very useful information for forensics. To enable either or both of these features, set the following registry values:

Key: HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\Memory Management\PrefetchParameters
Value: EnablePrefetcher
Value: EnableSuperfetch
Data: 3 [REG_DWORD] for both values

Prefetch and SuperFetch are hotly debated features on Internet blogs, especially concerning computers with less than 4GB memory or with solid state drives. Both should be enabled on all workstations, even with SSD storage, but Prefetch should only be enabled on servers temporarily during an incident response when it is known that Prefetch will capture relevant data, i.e., it should not be enabled by default on servers.

Memory Dump on Crash
Every computer should be configured to write at least a small memory dump when a system crash occurs (it will be 256KB or less in size). If possible, though, a crash should instead save either a kernel memory dump or a complete memory dump. Be aware that

memory dumps have specific paging file and partition requirements, and even a kernel-only dump will add several minutes to the time required to reboot after a crash.

Server 2012, Windows 8 and later also have an option set by default called "Automatic memory dump" which can change the dump settings dynamically based on the computer's prior four-week crash history (you can manually configure these options in Control Panel > System > Advanced System Settings > Advanced tab > Settings button for Startup and Recovery). This is set by default and should be kept enabled.

Because of the complexities involved, please see KB969028 and do an Internet search on "CrashDumpEnabled", which is the name of the main registry value involved.

System Restore and System Protection (Volume Shadow Copies)
System Restore allows a workstation's OS, drivers and program files to be restored to an earlier state, called a "restore point", usually for troubleshooting purposes. System Protection uses the volume shadow copy feature of the workstation to recover previous versions of users' data files, such as when a file is damaged or accidentally deleted. On a workstation, both of these are configured by going to Control Panel > System > System Protection link.

On servers, System Restore is not available (without hacking the machine), and System Protection is simply called "Volume Shadow Copies", which is not enabled by default, but can be enabled by opening File Explorer > right-click any local drive > Configure Shadow Copies.

System Restore is enabled by default on workstations and should be left enabled for its forensics and other benefits. It can be disabled by setting a registry value named "DisableSR" or through Group Policy (Computer Configuration > Policies > Administrative Templates > System > System Restore), but it's best not to do so.

Keep in mind that restore points will be deleted automatically to free up drive space as needed, and restore points can be scheduled to only occur at night if desired. This is one of the main (erroneous) objections to enabling restore points.

On servers, do not enable System Restore. This is unsupported by Microsoft and full system backups are probably performed nightly on servers anyway.

On file servers with high-value shared folders, enabling Shadow Copies is useful for many reasons, not just forensics. The maximum amount of storage space which may be used by Shadow Copies is also configurable if you're worried about a run-away loss of free space. However, no blanket recommendation can be made for Shadow Copies in general, you will have to decide what's best for each shared folder on each server.

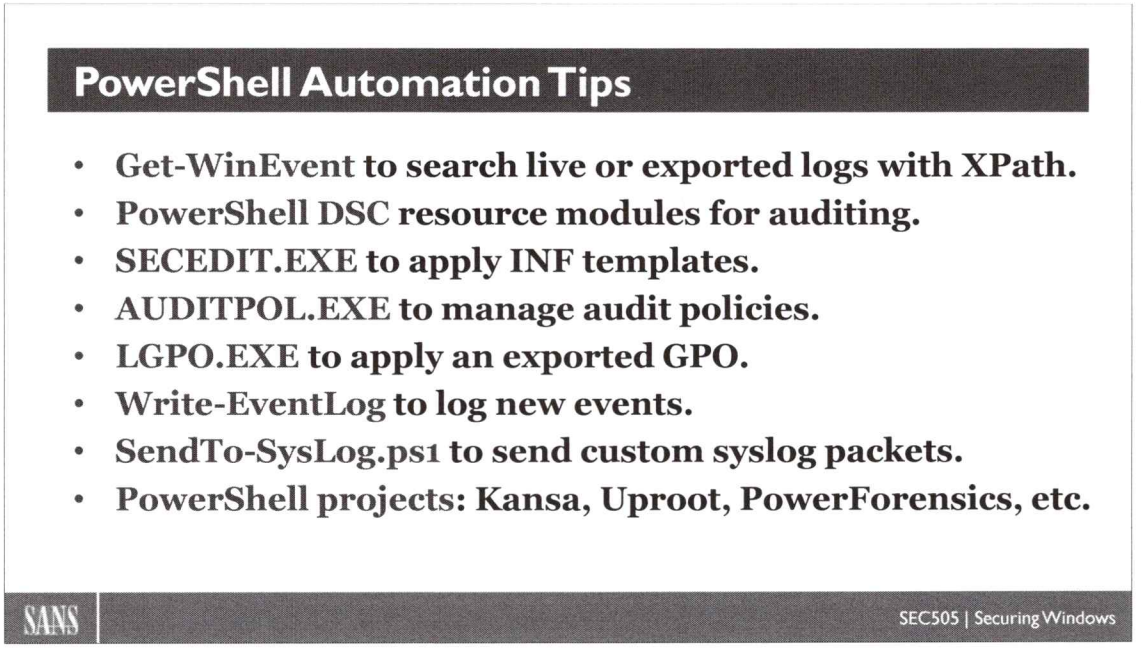
Internet Explorer History and InPrivate Browsing
Malware is often downloaded through the browser. For forensics purposes, we want to increase the number of days that browser history is maintained and also to disable the

InPrivate Browsing feature of IE, which indirectly limits browser history. We're not concerned here with users attempting to cover their tracks as they violate acceptable use policies, we mainly want to retain potential indicators of compromise. The following Group Policy settings are recommended for all servers and workstations:

- Computer Configuration > Policies > Administrative Templates > Windows Components > Internet Explorer > Delete Browsing History > enable the "Disable Configuring History" option and set the number of days to keep pages in history to at least 90 days.

- Computer Configuration > Policies > Administrative Templates > Windows Components > Internet Explorer > Privacy > set "Turn off InPrivate Browsing" to Enabled.

If you have standardized on a different browser, such as Google Chrome or Mozilla Firefox, then try to set the equivalent options in that browser too, e.g., the Chrome "incognito" feature.



PowerShell Automation Tips

Just as a reminder, there are many PowerShell resources for automation:

- The **Get-WinEvent** cmdlet can search local or remote live event logs using the very flexible XPath query language. It can also search offline, exported event log files (.evtx) using XPath.

- There are **Desired State Configuration (DSC)** resource modules on the PowerShell gallery for managing audit policies and event log settings (www.PowerShellGallery.com).

- **SECEDIT.EXE** can be scripted to apply an INF security template, and these templates can manage legacy audit policies, event log settings, and NTFS/ReFS audit settings. INF templates, however, cannot be used to set Advanced Audit Policies found on Windows Vista and later.

- **AUDITPOL.EXE** can query and manage all audit policies, including the Advanced Audit Policies, on Vista and later.

- **LGPO.EXE** is a free download from Microsoft which can be used to apply an exported GPO to a stand-alone computer. This GPO may include legacy and advanced audit policies, event log settings, and NTFS/ReFS audit settings.

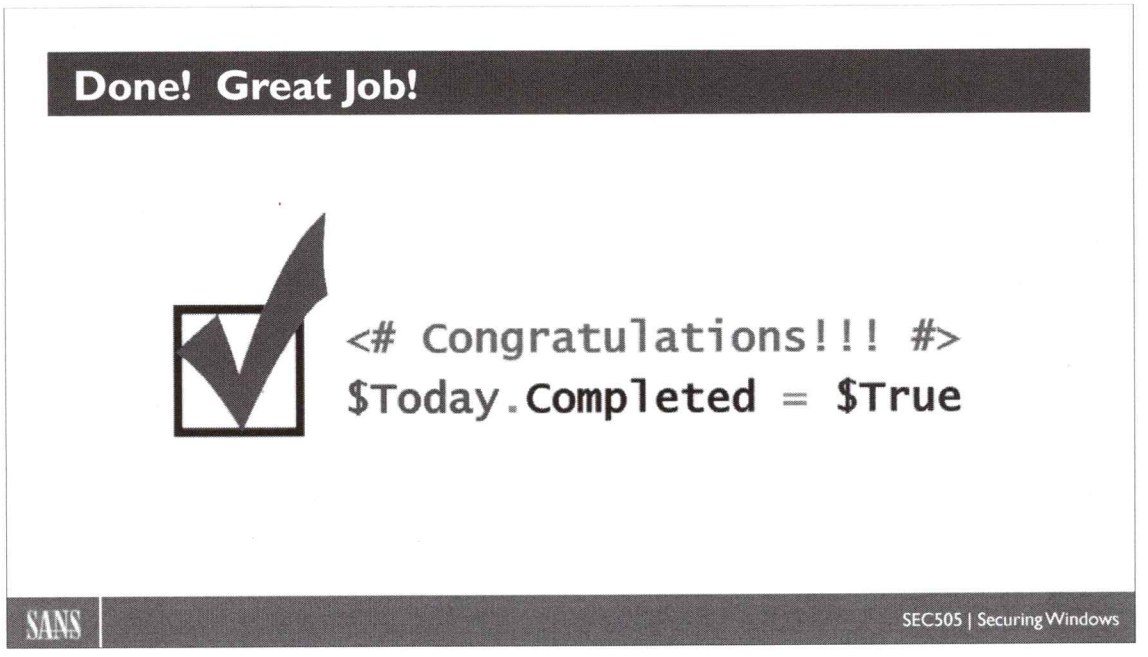
- **Write-EventLog** can log new events and create new event logs. There are many other scripts and binary tools available on the Internet for writing to Windows event logs too.

- **SendTo-SysLog.ps1** is a script on your courseware media (USB or DVD) which can send properly-formatted UDP messages to a Syslog server. It's command-line parameters allow you to easily set the facility, severity, message payload, and Syslog server IP address or FQDN.

Finally, there is a growing list of open source PowerShell projects related to forensics, post-exploitation, hacking, and security in general. These projects may come and go, but here is a partial list projects to check out:

- PowerShell Empire (http://www.powershellempire.com) for post-exploitation.
- Kansa (https://github.com/davehull/Kansa) for baselines and hunt teams.
- PowerForensics (https://github.com/Invoke-IR/) for forensics.
- Uproot (https://github.com/Invoke-IR/) for WMI, agentless HIDS.
- PowerSploit (https://github.com/PowerShellMafia/) for penetration testing.
- Posh-SecMod (https://github.com/darkoperator/) for security functions.

Search GitHub for "PowerShell" to see what else comes up! Have fun!



Congratulations!

You have finished the course!

Please complete the evaluation form and return it to the room monitor or drop it in the "Evaluations" box. Written comments are especially appreciated, and play a large role in how we update the manual.

Thank You for attending this seminar!