

ITPro<sup>TM</sup>  
SERIES

WindowsITPro

 **eBooks**

Keeping Your Business  
SAFE from Attack:

# Passwords and Permissions

By Roger Grimes

**Microsoft<sup>®</sup>**



## Contents

<b>Chapter 5 Auditing to Detect Intrusions</b> .....	<b>103</b>
<b>Intrusion Detection</b> .....	<b>103</b>
<b>Windows Auditing</b> .....	<b>104</b>
Security Log .....	104
Event Message Fields .....	105
Other Logs .....	109
<b>Audit Policy</b> .....	<b>110</b>
Maximum Log Size .....	110
Archiving Events .....	111
<b>Audit Categories</b> .....	<b>112</b>
Audit Account Logon Events .....	113
Audit Account Management .....	114
Audit Directory Service Access .....	115
Audit Logon Events .....	115
Audit Object Access .....	115
Audit Policy Change .....	117
Audit Privilege Use .....	117
Audit Process Tracking .....	118
Audit System Events .....	118
Per-User Selective Auditing .....	118
Auditing Isn't Perfect .....	119
<b>Event Log Management</b> .....	<b>120</b>
Measuring Baselines .....	120
Synchronizing Time .....	120
Logging Security Events .....	121
Determining Log Rotation and Permanence .....	121
Centralizing Data Collection .....	121
Event Viewer Console .....	121
EventCombMT .....	122
Log Parser .....	123
Microsoft Audit Collection System .....	123
Standardizing Terms .....	123
Filtering Data .....	123

Correlating Data .....	125
Extracting Useful Information .....	125
Setting up an Alerting System .....	125
Simple Windows Alerting Mechanisms .....	126
Windows Event Triggers .....	127
Other Third-Party Alert Utilities .....	128
<b>Auditing Best Practices .....</b>	<b>128</b>
<b>Summary .....</b>	<b>128</b>

## Chapter 5:

# Auditing to Detect Intrusions

No matter how well you secure a computer, malware and rogue hackers will still assess its weaknesses and try to exploit any potential vulnerability. Windows has a fantastic feature that lets you monitor attempted and successful exploits – auditing. When configured correctly, auditing can monitor and detect most intrusions. The more you learn about Windows security, the more you'll be involved with the audit logs.

With Windows auditing, you can configure settings in nine different categories of events; in this way, you can capture most events occurring on a workstation or server. To successfully enable auditing, you need to understand what each audit category does, understand the different components of a successful audit policy, and know how to read the resulting messages. Chapter 5 covers basic intrusion detection, Windows auditing, audit policy, auditing categories, event log management, and audit policy best practices.

## Intrusion Detection

Intrusion detection is the process of noticing an unauthorized event, an unauthorized action, or unauthorized content. Unauthorized events can be caused intentionally by outside hackers, legitimate insiders performing unauthorized activities, or automated malware, or even by insiders innocently causing accidents.

According to the Microsoft product support teams that handle hacking incidents, the most common symptoms of unauthorized malicious events are

- Unexplained account lockouts
- Unexplained significant, rapid, sustained increase in CPU utilization (indicating a possible worm)
- Unexplained significant, rapid sustained increase in usage of local or wide-area network bandwidth (indicating a possible worm or a denial-of-service attack)
- Unexplained significant decrease in free disk space (indicating the possibility of a hacker using storage space)
- Unexpected server or workstation reboots (all instances should be investigated thoroughly)
- Unexpected process crashes (indicating buggy hacker code)
- Unexplained STOP errors (a.k.a. Blue Screens of Death)
- Unexplained network connections
- Unexpected installation of new services or programs (which is hard to track unless you are on top of the services and programs that should be running)
- Unexpected installation of new software patches (indicating the possibility that hackers are closing holes that allowed them in to protect their newly acquired asset from other hackers)
- Unexpected file or registry modifications (a very common indication of automated malware)
- Suspension or disappearance of antivirus software
- Deletion of Admin shares

- Appearance of new, unauthorized user accounts
- Unexpected activation of a high number of DHCP leases
- Unexplained clearing of the security log — which is different from a security log that has never logged any security events

Although any of these events can have non-malicious causes, any instance of these symptoms should be immediately researched to conclusion. All of these types of events can be detected manually through observation or using auditing and other related mechanisms.

## Windows Auditing

In Windows, you can audit nine categories of policies in up to six default log files: Application, Directory service, DNS Server, File Replication Service, Security, and System. Security events are usually posted to the Security log, although some are sent to the Application log file. Log files can be accessed using Event Viewer (Eventvwr.exe located in the System32 folder). Users, by default, have Read and Execute NTFS permissions on the Event Viewer application. It can be launched and used on the command line, through a Microsoft Management Console (MMC) application (for example, Computer Management or Local Computer Policy), through group policy, or through administrative utilities such as Systems Management Server or Message-Oriented Middleware.

The Application, Security, and System log files are available on all operating systems in the NT family, to both clients and servers. Any application can write to the Application log file, but the application developer must specifically code the application to do so; Windows doesn't automatically write application errors to the Application log. The Windows OS writes most audit messages to the Security log, although other applications can also use the security log if they are specifically programmed and allowed to do so. The System log file typically contains only event messages that are generated by the operating system. The Directory Service and File Replication Service log files are available only on computers with Windows 2000 or later file server OSs. The DNS Server log file is available on Windows 2000 or later file servers running the DNS Service. This chapter concentrates on the Security log file.

## Security Log

The security log file is called SecEvent.Evt and is located in %SystemRoot%\System32\Config along with the other default log files. To change the location of the Security log file, modify the HKLM\System\CurrentControlSet\Services\Security\File registry key. Several other log-related values can be modified in this registry area.

By default, two different services write to the Security log: the Security Reference Monitor (SRM) and the Local Security Authority Service (LSASS). The SRM reports on interactions with objects and LSASS on all other operating system audited events. By default, only Administrators and the LocalSystem accounts have Full Control access to the default log files. Non-administrative users can view (but not manage or clear) the Application and System logs, but they have no access the Security log file. A group policy (or registry edit) setting called *Prevent local guests group from accessing security log* determines whether members of the Guests local group can access the security log. Guest access is disabled by default in Windows XP and above, although enabling the setting is usually unnecessary anyway because the default Guest account is disabled by default. Guest access to the security logs was necessary for some legacy applications.

User accounts must possess the *Manage auditing and security log* user privilege to access the Security log. Unfortunately, user accounts with this right can also clear the Security log. This right also lets individual users enable object access auditing (covered below) on a per-object basis, but does not enable general *Audit Object Access* auditing. By default, only Administrators have this right.

Another related setting is *Generate security audits*. Accounts with this user right can add events to the Security log. By default, only the Local Service and Network Service accounts have this privilege in Windows XP and 2003. To write to the Security log, other applications or services must run in a user account context that possesses this user right. This right was created to prevent hackers or programs from writing random security events to the Security log file in an attempt to hide legitimate entries.

For some events, such as detailed Kerberos events, to be written to the security log, logging must be turned on explicitly. By default, most Kerberos authentication events are logged; however, not all are included. To ensure that all Kerberos events are logged, you must enable detailed Kerberos logging by creating a new registry value. Create the HKLM\System\CurrentControlSet\Control\Lsa\Kerberos\Parameters\LogLevel and set its value to 1.

## Event Message Fields

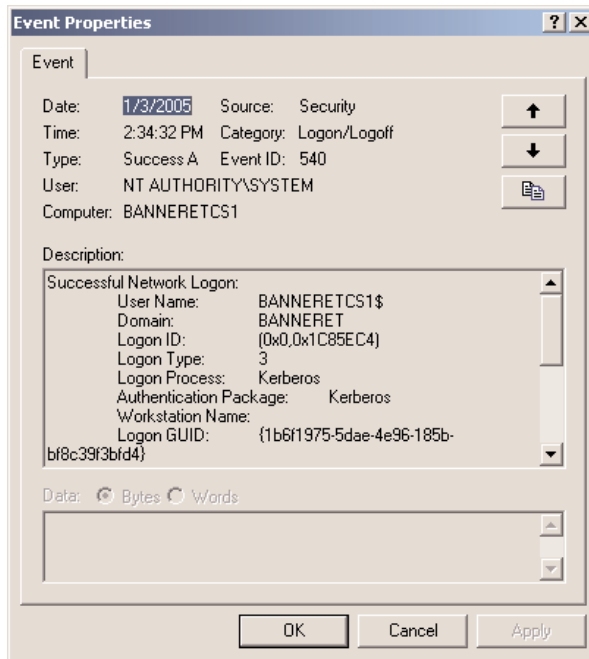
Every event log message contains one or more fields. Table 5-1 shows the main Security log event message fields:

**Table 5-1 Security event log message fields**

Field	Description
Date	The date the event occurred.
Time	The time the event occurred.
Type	This field has two possible values: Success Audit or Failure Audit. Typically, a success audit is logged when some event has been successful (for example, a successful logon or permitted object access), and a failure audit is logged when some event has not been successful (for example, a failed logon or denied object access). Some success audits are generated for events that failed or were denied (successfully so, I guess).
User	The security principal account involved in the security event. This account can be a user, computer (computer accounts end with a \$), a service, a built-in account (for example, System or Anonymous_Login), or other security principal.
Computer	The computer where the event happened or was authenticated.
Source	The process, service, or application involved in the event. Unfortunately, in the Security log, the Source field always contains the value, "Security," which provides little additional help.
Category	In the Security log, the category classifies the type of event, such as account management or object access.
Event ID	The numeric value of the event message. Microsoft tracks event messages using this value; this value also identifies the message in event log databases. An excellent online source for event IDs is <a href="http://www.eventid.net">http://www.eventid.net</a> . It's not always 100 percent accurate or inclusive, but it is as good as you'll find.
Description	This field contains more event-specific information (see Table 5-2 for example fields). The information displayed varies by event type.

Figure 5-1 is an example of a Security event log message showing many of the common fields. The example shows Event ID 540, a successful logon/authentication event of a workstation account to the server.

**Figure 5-1**  
*Example security event log message*



**Table 5-2 Example security event description fields**

Field	Description
Event Description	A short text field, usually one sentence or less, that tracks to the Event ID.
User Name	The account involved in the event. Could be an actual user logon name, a computer name (always followed by a \$), or a built-in account like System.
Domain	The name of the NetBIOS domain where the (user name) account is located. If the user account is from the local Security Accounts Manager, this field has the local computer's NetBIOS name. For built-in local accounts, this field has the value of NT Authority.
Logon ID	A number unique to the logon session. It is not the user's logon account name. The Logon ID is unique until the computer is restarted. This field can be used to track related events.
Logon Type	The method the account logged on with (see Table 5-3). It indicates whether the account logged on locally (interactive), over the network, or in some other way.
Logon Process	The name of process that performed the logon (see Table 5-4).
Authentication Package	The authentication software package used in the authentication event (see Table 5-5).
Workstation Name	The NetBIOS computer name.
Logon GUID	A number that is like the Logon ID field but is globally unique.
Transited Service	The Kerberos delegation extension field ( <a href="http://msdn.microsoft.com/msdnmag/issues/03/04/SecurityBriefs">http://msdn.microsoft.com/msdnmag/issues/03/04/SecurityBriefs</a> ).
Source Network Address	The source IP address, if available (and if it's part of the event message).
Source Port	The source transport port number, if available (and if it's part of the event message).

The description information that is displayed varies widely according to the Event ID. For example, Event ID 776, *Certificate Services published the certificate revocation list (CRL)*, has description fields labeled Base CRL, CRL No, Key Container, Next Published, and Published URLs. For more information about each event message, search on each Event ID in Windows Help and Support, online help, or on one of the online support sites (such as <http://www.eventid.net>).

**Table 5-3 Logon types**

Value	Description
2	An interactive logon; a local logon, not done using a network. However, in Windows NT and 2000, Terminal Services and remote desktop logons are recorded as local logons.
3	A network logon; a logon that occurred remotely over network. However, this value can also be recorded if a local user browses a computer using Network Neighborhood.
4	A batch logon; this value indicates accounts that log on with the Logon as batch job privilege. Normally, this value is recorded only for Task Scheduler jobs, but any logon account can be given that right.
5	A service logon; the logon was done by a service configuration manager on behalf of a service using a service account (that is, a service started and logged on).
6	A proxy logon
7	An unlock workstation event; a user logged back into a workstation at a console locked from the screensaver.
8	A network cleartext logon; this value is usually associated with an IIS logon using Basic Authentication.
9	A Newcredentials logon; a caller (process, thread, or program) cloned its current SID token and specified new credentials for outbound connections, creating a new logon session.
10	A RemoteInteractive logon; this value indicates a Remote desktop (RDP) or terminal services logon process (new in XP and above).
11	A logon process that used cached credentials. By default, all computers and users can log on using previously cached credentials if a domain controller is not available to authenticate their logon request.

**Table 5-4 Logon processes**

Value	Description
AdvApi	An application called LogonUser initiated the logon
Kerberos	Kerberos initiated the logon IIS Microsoft IIS initiated the logon
MS.Radius	A Remote Authenticated Dial-In User Service (RADIUS) such as Internet Authentication Service (IAS) initiated the logon
Ntlmssp	NTLM performed authentication
SCMgr	A service account logged on
User32 or Winlogon/MSGina	A normal interactive (local) logon process, often initiated by the end-user hitting Ctrl-Alt-Del sequence to log on

**Table 5-5 Default authentication package values**

Value	Description
Microsoft_Authentication_Package_V1_0	Capable of LM or NTLM (version 1 or 2) authentication
Kerberos	Kerberos authentication. Not valid for local logons; available only in Windows 2000 and above
Negotiate	Client and server negotiate authentication protocol. Currently valid only between NTLM and Kerberos.
SChannel	IIS authentication. SSL or TLS support.
Digest	Authentication IIS authentication.

Because authentication packages can be customized (for example, for advanced cryptography or smart cards), the values listed above can change.

Microsoft is aware that their event log messages aren't always easy to understand for the new user, but every version of the operating system has easier-to-understand messages. Event log messages in Windows XP and above often contain direct links to Microsoft articles about that Event log message. In Windows 2000, the user can copy the event log message text to the clipboard with one click, so that it can then be posted easily into an Internet query. Microsoft is also working to minimize the number of less-useful messages (that is, "noise") to make the logs more relevant.

## **Other Logs**

Many other Windows applications have their own log files, including IIS, Windows Firewall, DHCP, DNS, and PPP. IIS has multiple logs (located by default in %SystemRoot%\System32\Logfiles), including separate log files for HTTP, FTP, and the SMTP virtual server. IIS 6.0 splits invalid HTTP requests from the normal web log to a separate log file under the \Httperr folder.

Each of these logs can be customized to change the frequency the log rotates, its location, and the information it contains. The IIS logs can be used with the Security log to track events. For example, successful or failed IIS authentication events are tracked to the security log; however, in IIS versions before 6.0, the IP address of the computer attempting to authenticate was tracked only in the IIS web service log file. Used together, both logs paint a more complete picture. In Windows Server 2003, the security log in IIS 6.0 security log can track IP addresses for IIS authentication events.

The Windows Firewall log, when enabled, saves events to %SystemRoot%\Pfirewall.log. DHCP saves detailed log files to %SystemRoot%\System32\DHCP. Each log file begins with a description of the various DHCP events tracked. Perhaps the most detailed logs are generated by DNS. Windows Server 2003 DNS can track every DNS event down to packet-level data.

When setting up a Windows auditing scheme, you should research the logs that are available to the various applications enabled on each computer; you might find that there are advantages to enabling logging there, too. Follow the instructions in KB Article 234014 (<http://support.microsoft.com/default.aspx?scid=kb;%5BLN%5D;234014>) to enable PPP logging of dial-up connections. IAS and RRAS have similar instructions to enable logging.

The Performance Monitoring utility (called the Performance Console in XP and above) can track statistics that provide useful security information. For example, worms, buffer overflows, and denial-of-service trojans often cause sudden, unexplained, sustained spikes in processor or network utilization. The worms start thread after thread in their search for more machines to exploit. Buffer overflows can cause spikes of 100% utilization because of the unnatural way programs end. Savvy security administrators can enable Performance Monitor alerts to warn them of sustained high utilization levels.

In the larger picture of detecting malicious intrusions, don't forget the logging facilities of your other network devices, such as firewalls, switches, antivirus services, mail gateways, routers, and other security devices. When a malicious event occurs, every bit of extra auditing helps in forensic analysis and recovery.

## Audit Policy

As is the case with any security tool, successful audit policies require a little bit of forethought and pre-planning. Enabling auditing should involve these three overall steps:

1. Decide what to audit.
2. Configure and enable auditing.
3. Monitor audit event messages.

Here are some basic questions to consider:

- How large can the audit logs become?
- What audit categories should I enable?
- What objects should I audit?
- How should I collect and monitor event messages?
- Who has ownership of the audit policy?
- How should the event logger treat log files when they become their maximum size?
- If the audit log fills up, should the computer “freeze”?
- How should old event logs be archived and for how long?

The rest of this chapter will cover these topics.

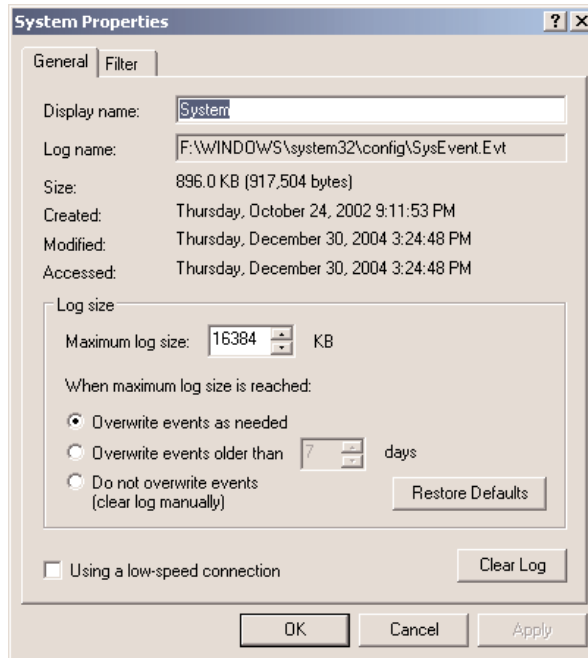
## Maximum Log Size

The first order of business is to determine the maximum size of the Security log (see Figure 5-2). Although the theoretical maximum size of the log file is the maximum file size that is available to the NTFS file system (4 GB in Windows 2000 and later), the real maximum size is much smaller. Bill Boswell, columnist for Redmond Magazine (<http://www.redmondmag.com/columns/article.asp?EditorialsID=743>) discovered that the maximum size is 1 GB, and because of other related processes, the maximum practical size is somewhere around the area of 300 MB.

In any case, very large logs are cumbersome and slow; as the size grows, the log works even more slowly. It takes longer to open Event Viewer, takes event log management tools longer to perform queries, and makes sorting and extracting information sluggish. On the opposite end of the spectrum, if you create separate smaller log files, you must combine them to get trends and find events that span multiple time periods. Regardless, at some point, your logs fill and have to be cleared, archived, or overwritten.

You can set the maximum size for the log file in three ways: in the registry, in the user interface (see Figure 5-2), or using group policy.

**Figure 5-2**  
User interface for setting maximum log file size



Log sizes must be multiples of 64KB. The default log size for Windows Server 2003 is 16 MB; for XP Pro, it is 512KB. Choose any size that is practical for your organization. For example, if your security log captures 1 MB of data per day and you need 7 days of data in one log, make your Security log maximum size 7MB (or a bit more to include long weekends and holidays).

If you are unsure what size to make your log, start with a 16MB file and adjust up or down, as your experience dictates. As a general rule, domain controllers and file servers probably need large log files and end user workstations smaller logs.

## Archiving Events

When the log file has reached its maximum size, you have three options: clear the log, allow it to be overwritten, or save the log file. In setting group policy, this setting is known as *Retention method for log*. The default setting (see Figure 5-2) is to *Overwrite events as needed*. This option overwrites oldest events first. You can also choose to *Overwrite events older than x days* old. You may also choose to not overwrite events, which requires that you clear the event log manually. If you choose this option and the event log fills up, no events will be logged until the log is cleared.

In high-security environments, you can enable a security setting to shut down the system if the Security log becomes full; in group policy, choose Local Policies, Security Options, *Audit: Shut down system immediately if unable to log system audits* or by setting a registry key value (see <http://support.microsoft.com/?kbid=888179> ). This setting is available so that security events are never dropped. If events can't be recorded, the system becomes unavailable. This defense was created

specifically to prevent a hacker from generating enough bogus security events to fill the Security log and make it stop logging. If the system does stop processing because of this setting, it generates a STOP error. The system can be restarted, but only an administrator can log on. The administrator must re-set a registry value, save and clear the log, and restart the system.

It makes sense to save and back up your logs before you clear them or allow events to be overwritten by new events. Often, the event you are troubleshooting, malicious or not, had its origins days or months before you noticed the symptom. An archive of old logs makes forensic investigation easier; creating a plan to archive audit logs is covered below. In high-security environments, you should consider backing up all logs to write-one, read-many media. This will help support the logs as stronger evidence, if the need arises.

## Audit Categories

You can log nine categories of events (see Figure 5-3):

- Audit account logon events
- Audit account management
- Audit directory service access
- Audit logon events
- Audit object access
- Audit policy change
- Audit privilege use
- Audit process tracking
- Audit system events

**Figure 5-3**  
*Audit categories*

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

Categories can be enabled or disabled using Local Computer Policy or group policy. You can enable each category for success, for failure, or for both. You should audit both the success and failure of high-risk events (such as logons), failure-only for events that monitor attempted unauthorized access, and both success and failure for high-risk events for which any access should be noted.

Generally, you should audit success *and* failure events

- if the asset has a high risk of compromise
- if the asset has high value to the organization
- if the asset, if successfully exploited, leads to widespread compromise of other network resources (for example, the Domain controller, DNS server, or the Certificate Services server)
- any system events
- any policy changes

In general, you should audit failure events for most audit categories *except* process tracking and policy change. Generally, you should audit success events for Audit account management.

Table 5-6 shows the recommended settings from in the Microsoft Windows Server 2003 Security Guide.

**Table 5-6 Windows Server 2003 Security Guide recommendations**

Audit Category	Computer Environment		
	Legacy	Enterprise	High Security
Account logon events	Success, Failure	Success, Failure	Success, Failure
Account management	Success, Failure	Success, Failure	Success, Failure
Directory service access	Success, Failure	Success, Failure	Success, Failure
Logon events	Success, Failure	Success, Failure	Success, Failure
Object access	Success, Failure	Success, Failure	Success, Failure
Policy change	Success	Success	Success
Privilege use	No Auditing	Failure	Success, Failure
Process tracking (see Note below)	No Auditing	No Auditing	No Auditing
System events	Success	Success	Success



### Note

The Security Guide recommends a setting of **No Auditing for Process Tracking**; however, this chapter recommends enabling it on domain controllers. See the “Audit Process Tracking” section below for more discussion.

When enabling these overall categories, most categories are “all or nothing” — you can’t set them for individual objects or users. For example, if you enable Audit privilege use or Audit system events, all uses of those privileges (such as user right assignments) or all system events for all security principles will be recorded. However, the Audit object access category must be set on a per-object basis, and on a per-user or group basis.

Now, we’ll cover each category in more detail.

## Audit Account Logon Events

When Audit account logon events is enabled, all domain-based authentication events are logged to the authenticating domain controller. For example, if a domain user on Computer A connects to a file share on MemberServerB, the authentication event is logged only on the domain controller that

authenticated the event. This setting does not track logons or authentication events involving local accounts. When tracking hackers, an administrator has to check only participating domain controllers to find relevant authentication events. This setting was not available until Windows 2000.

When checking events in this category, you look for sustained patterns of logon failures, although not all logon intrusion events will trigger a failure event. For example, Event 644-Account Lockout is a success event. If you find suspicious patterns, note the logon time, type, and process. Multiple, successive logon attempts performed in a very short time period usually indicate an automated attack.



**Note**

**Both Audit account logon events and Audit logon events monitor successful and failed attempts to logon or to authenticate.**

## ***Audit Account Management***

This setting tracks the creation, change, or deletion of a user account or group and records when a password is set or changed. Hackers often create new user accounts or add themselves to administrative groups. For example, a hacker could use a privilege escalation attack to add a regular end-user account to the Administrators group or to enable a previously inactive account. Events in this category tell you what was affected and what account initiated the change. Accounts that are locked out are recorded in this category, as well as passwords that are reset or changed.

Important account management events include

- 624 User account created
- 627 User password reset
- 628 User password set
- 630 User account deleted
- 629, 626 Account disabled, or re-enabled
- 632 Member added to global group
- 635 New local group created
- 636 Member added to local group
- 643 Domain policy changed
- 644 User account locked
- 645 Computer account created
- 658 Universal group created
- 660 Member added to universal group
- 685 Name of an account changed

## ***Audit Directory Service Access***

This category records when a security principal accesses an Active Directory object on a domain controller. In the Default Domain Controller Group Policy object (GPO), this value is set, by default, to no auditing and remains undefined for workstations and servers where it has no meaning. Although security experts often ignore this category, they shouldn't — it is good for tracking hacker activities during a known compromise across the network.

## ***Audit Logon Events***

This category tracks local logons and authentication events at the computer where the resource is accessed. Events to watch for in this category include

- 528 User logged on
- 529 Logon failure; bad name or password
- 530 Logon failure; outside allowed logon time
- 531 Logon failure; logon attempted to disabled account
- 532 Logon failure; logon attempted using expired account
- 533-539 Logon failures for various reasons
- 540 User logged on to network
- 548 and 549 Logon failure-Filtered SID
- 550 Possible denial of service (DoS) attack

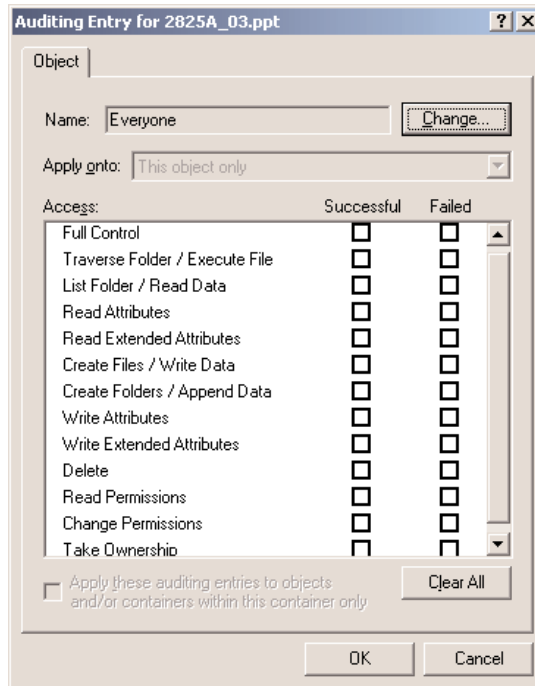
In most cases, enabling Audit logon events and Audit account logon events at the same time is a good way to make sure you capture all logon and authentication events, which will let you watch for password and account hacking. For example, if you have renamed your Administrator account, be on the lookout for attempted logon attempts to the old account name, which will trigger the message *Event ID 529 - Unknown User name or bad password*. If you have disabled your bogus Administrator account, look for *Event ID 531 - Logon Failure-Account Disabled* or *Event ID 675 - Pre-authentication failed* - the user typed bad password.

## ***Audit Object Access***

This category, Audit object access, is perhaps the juiciest auditing feature in Windows. It tracks any access or attempted access of any Windows object with a system access control list (SACL; covered in an earlier chapter). In tracking hackers and malware, auditing object access is the most valuable player.

Objects that can be audited include files, folders, registry keys, and printers. Objects can be audited overall, per group, or per user. The specific attributes and accesses that can be audited vary, because files, registry keys, and printers are distinctly different objects. Attributes available for file auditing (see Figure 5-4) include read access, delete attempts, write attempts, execution, changing permissions, and taking ownership.

**Figure 5-4**  
*File auditing attributes*



Object access auditing is particularly tricky because it is a two-step process: it must be enabled overall (in the category setting) and then on each individual object you want to monitor. However, enabling auditing on every object on a computer system creates a lot of overhead and takes literally hours to complete, even if you perform it at the root level. Therefore, you should enable object access auditing on areas that are most likely to be compromised, such as auto-run registry keys, Startup folders, and the System32 folder. Other sensitive areas to monitor include

- Write success or failure on system or program files to monitor for virus execution
- Write success of HOST file
- Deletion or modification of antivirus program files

Notable Event IDs include

- 563 Attempt was made to delete file
- 564 Object deleted
- 567 Permission was executed on object
- 570 Object access attempted

## ***Audit Policy Change***

This security setting can log every change to user rights assignment policies, audit policies, or trust policies. Note that this setting does not include changes to group policy (as often believed) or changes to password policy. It might be helpful in catching a hacker clearing the event log or in attempting to elevate privileges. Notable events include

- 608 User right assigned
- 609 User right removed
- 612 Audit policy changed
- 621 System access granted to an account



### ***Note***

**Because of a bug, the Failure setting does not capture data; therefore, the administrator can check only for successful Audit Policy Changes. There is no easy way around this bug until Microsoft fixes it.**

## ***Audit Privilege Use***

This security setting can track each instance in which a security principal exercises a user right (that is, a privilege). To see a list of possible privileges, open any group policy object and view the User Right Assignments container object or visit <http://www.microsoft.com/resources/documentation/WindowsServ/2003/standard/proddocs/en-us/Default.asp?url=/resources/documentation/windowsserv/2003/standard/proddocs/en-us/520.asp>. User rights assignments are considered privileges because they convey additional permissions that cannot be communicated solely using NTFS object security.

When enabled, this category tracks the use or attempted use of any privilege by a security principal of any user right except these:

- Bypass traverse checking
- Debug programs
- Create a token object
- Replace process level token
- Generate security audits
- Back up files and directories
- Restore files and directories

Auditing the user rights listed above tends to generate too many (usually uninteresting) events in the security log, which impedes performance without adding security value. For example, if backing up and restoring files were audited by default, every routine data backup would generate thousands to millions of events each time it ran. However, if you choose, you can enable the auditing of these rights in the security options of group policy.

### ***Audit Process Tracking***

This category creates detailed information for events such as program activation, process exit, handle duplication, and indirect object access. It is usually not set unless it is needed in ongoing program development or hacker tracking. Often, security guides recommend not enabling process tracking because of the “noise” it could create, but you should enable it on any computer where you suspect malicious or unauthorized activity.

You should also enable it on any domain controllers. Domain controllers aren’t usually starting and stopping a lot of applications, so any “noise” it creates is minimal. Also, any domain controller is a high-value target, so enabling Audit process tracking is compulsory for competent security auditing.

Notable events for this category include

- 592 New process created
- 593 Process exited
- 595 Indirect access to an object was obtained
- 601 User attempted to install service

### ***Audit System Events***

This category tracks the times that a user restarts or shuts down the computer or when an event occurs that affects either the system security or the Security log. Events to be aware of in this category include

- 512 Windows is starting up
- 516 Resources exhausted; security events lost
- 517 Audit log cleared
- 519 Process used invalid local procedure call in an attempt to impersonate a client
- 520 System time changed

If any of these events happen unexpectedly, you should investigate the event until you are assured that it was not an intrusion.

### ***Per-User Selective Auditing***

A new feature added in XP SP2 (and slated for Windows Server 2003 SP1) is Per-User Selective Auditing. Introduced to meet a Common Criteria (<http://csrc.nist.gov/cc>) objective, it allows exceptions to overall audit policy on a per-user basis (you cannot set up exceptions for groups). It also ignores audit exceptions set for administrative accounts.

Using the AuditUsr.exe tool located in %SystemRoot%\System32, you can define exceptions to the overall audit policy for all nine categories (see Figure 5-5).

**Figure 5-5**  
*AuditUsr.exe command line syntax*

```

c:\ F:\WINDOWS\system32\cmd.exe
F:\WINDOWS\system32>auditusr /?
Auditusr - sets per user auditing policy.
/? prints command-line help.
If no parameters are given then the current settings will be displayed.
Specify one of:
/is <security principal>:<list of comma-delimited categories>
  adds or changes an include-success entry
/if <security principal>:<list of comma-delimited categories>
  adds or changes an include-failure entry
/es <security principal>:<list of comma-delimited categories>
  adds or changes an exclude-success entry
/ef <security principal>:<list of comma-delimited categories>
  adds or changes an exclude-failure entry
/r <security principal>
  removes all per-user auditing entries for that security principal
/ra
  removes all per-user auditing entries
/e <filename>
  exports current per-user auditing settings to a file
/i <filename>
  imports current per-user auditing settings from a file
Valid categories are:
System Event
Logon/Logoff
Object Access
Privilege Use
Detailed Tracking
Policy Change
Account Management
Directory Service Access
Account Logon
  
```

With this tool, you can import user accounts and settings in a comma-delimited file and export current exceptions to a file for confirmation purposes. Unfortunately, this feature is not very well documented.

### ***Auditing Isn't Perfect***

Although Windows auditing can capture most unauthorized events, it isn't perfect. To begin with, most event log messages are summaries. If you need the details of an attack, you'll need to rely on another auditing tool, like Network Monitor. Second, Windows auditing uses in-band tools, meaning that the data collection happens on the system that is being monitored. Therefore, a hacker can find out whether auditing is enabled, and if it is, manipulate the outcome. Finally, auditing misses many successful attacks, including many buffer overflows. If a buffer overflow from a single packet attack compromises a particular executable, the event logs don't always note the event as malicious. For this reason, always use Windows auditing in conjunction with other monitoring tools, such as antivirus or intrusion detection systems.

## Event Log Management

Managing your event logs requires more than simply collecting and reading event messages. It includes

- Measuring baselines
- Synchronizing time
- Logging security events
- Determining log rotation and permanence
- Collecting data in a central place
- Filtering data
- Correlating data
- Extracting useful information
- Setting up an alerting system

All part of the process must be planned, coordinated, and tested if intrusion detection is to be successful and useful.

### *Measuring Baselines*

Detecting unauthorized activity is impossible if you don't know what normal baseline activities are to start with. Before you start actively managing your event logs, you should monitor all the computers involved for normal activity. As any system administrator knows, Windows constantly creates event messages on any system. The trick is to learn what is and isn't normal. You should troubleshoot any unexpected events that come up during the baseline monitoring sessions before you start actively managing your event logs for security events. You should note and document baseline activity to use in future comparisons.

### *Synchronizing Time*

It is important that all computers involved be synchronized, including the logging and monitoring workstations. Make sure the time, date, and time zone settings are identical. Unsynchronized systems can make event correlation much harder than it needs to be. As a side benefit, guaranteed accurate time synchronization will make hacking evidence hold up better in court.

Modern Windows computers use the Windows Time Service and a protocol called Simple Network Time Protocol (SNTP, documented in IETF RFC 2030). The Windows Time Service can be fed its time reading from an internal computer clock (the default) or use an external time source, such as an Internet NTP time server. Window workstations participating in a Windows 2000 or later domain and using the default Kerberos authentication must be time-synchronized within 5 minutes of the authenticating domain controller to complete a successful login.

In a Windows domain environment, the domain controller that fulfills the PDC Emulator role is the centralized time sync server for the domain. The PDC Emulator computer should use a very accurate internal PC clock or should be configured to get its time from an external NTP server source (<http://support.microsoft.com/kb/216734/EN-US>). Several free NTP clients, including NetTime (<http://nettime.sourceforge.net>), are available for legacy Windows systems that do not have an NTP-compatible application.

## ***Logging Security Events***

Logging should be done anywhere it can be done. Start by using the Windows event logs, and use third-party tools as desired. All network devices in the path of data communications headed into and out of your network should have logging enabled — detailed logging when possible. You may have logs from Windows, firewalls, IDS, routers, switches, gateways, antivirus software, and anything else that may track packets or network communications.

## ***Determining Log Rotation and Permanence***

Logs should be collected and rotated frequently enough that data is not overwritten. It is important that logs be rotated on a schedule that balances accuracy and performance. Logs can quickly become large. A compromised computer with full logging enabled can have daily logs that are tens of megabytes big.

Security events should be logged to a permanent write-once, read-many media source if it is possible that the data will ever be used in court. Courts often support the “best evidence” doctrine, which states that nearly anything (except evidence known as hearsay) can be used for evidence in a court of law. However, evidence that is professionally collected and resistant to tampering will be more convincing in court than evidence without the same protections. If you can prove that your data was collected in a time-synchronized environment, tracked through its chain of custody, and was difficult to manipulate after collection, you have a pretty good evidence trail. Thoughtful decision-making must accompany log file rotation and archiving.

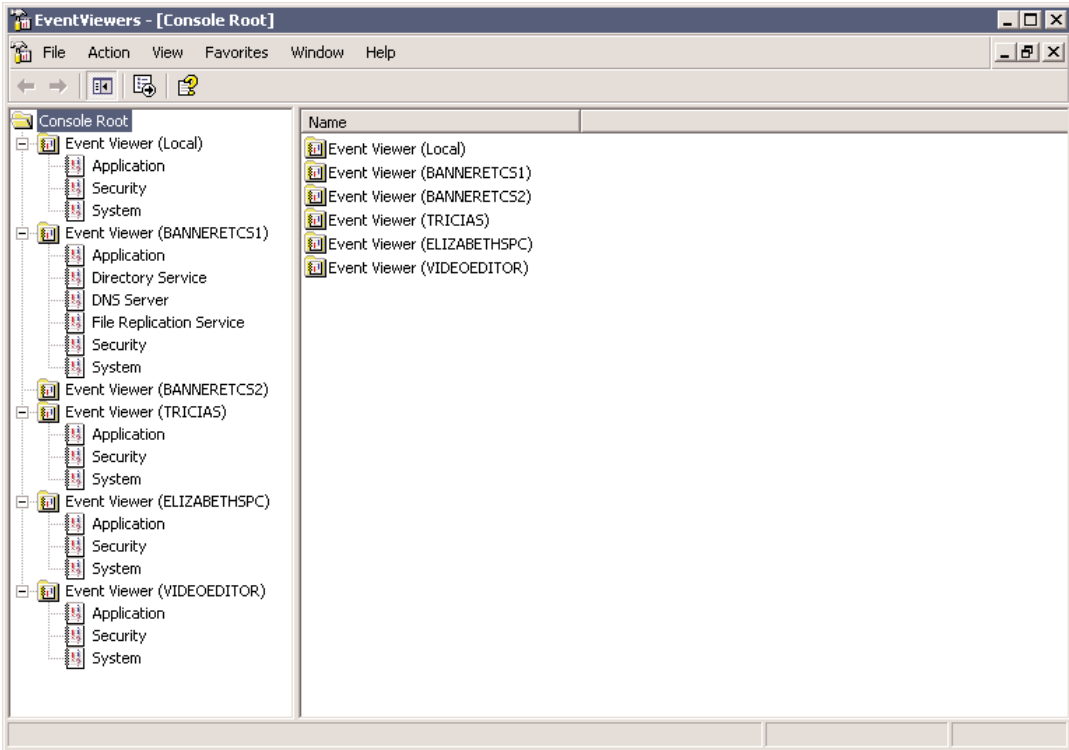
## ***Centralizing Data Collection***

It is the rare network that has only one log. Most have dozens or hundreds of logs. A setup that requires an administrator to check each log manually almost guarantees that the logs won't be checked routinely. Smart administrators collect as many security events as they can to a centralized location. It is nearly impossible for one centralized data collection system to collect all events, but the more you can collect centrally, the better off you will be. Microsoft has several free tools; dozens of third-party applications also collect and prioritize Windows log files. The four we look at are the Event Viewer Console, EventCombMT, the Log Parser, and the Microsoft Audit Collection System.

### ***Event Viewer Console***

If you have Windows 2000 or later, you can use the Event Viewer MMC snap-in to view event logs on the local machine and/or one or more remote machines. You can create an Event Viewer console that contains multiple computers' event logs in one location, as shown in Figure 5-6.

**Figure 5-6**  
*Event Viewer snap-in console monitoring several computers*



To view events remotely, you must have local administrator rights and you must have the Remote Registry and the Server services enabled on the remote machine. Although the Event Viewer console lets you view multiple computers' event logs in a central location, each log and its events are still separated.

## EventCombMT

Microsoft provides several ways to remotely collect multiple security event logs into a centralized database where they can be viewed, sorted, and prioritized at once. The oldest of these tools is EventCombMT (<http://support.microsoft.com/default.aspx?scid=kb;en-us;824209&Product=winsvr2003>). It lets you query multiple computer event logs and get the results in a common file. The file can be imported into SQL Server, Microsoft Excel, or another tool for analysis. EventCombMT works on Windows NT and later.

Although EventCombMT works well, it is not a real-time utility. Each time you want to collect data, you must initiate an EventCombMT query. Running a query against multiple machines, or even on a single machine with tens of thousands of records, can take a long time.

## Log Parser

Microsoft released a new tool called Log Parser in the IIS 6 Resource Kit (<http://www.microsoft.com/downloads/details.aspx?displaylang=en&familyid=8cde4028-e247-45be-bab9-ac851fc166a4>) that, although it is simply a rudimentary command-line tool, can extract data and events from a wide range of log sources. It does not require IIS 6 to work.

Log Parser uses SQL-like statements to query log file sources. Queries can be basic and extract all events or pull just specific records. You can save data to SQL databases, CSV files, and in many other formats. If you aren't used to SQL queries, the complexity and exacting syntax can be a hurdle. And like EventCombMT, Log Parser is a batch process. You can learn more about LogParser at <http://www.logparser.com>.

## Microsoft Audit Collection System

Microsoft's newest addition to the log collection family is the Microsoft Audit Collection System (or MACS). MACS represents the future of Microsoft's log collection tools. It is still in limited beta release and is not available to the general public. You can read more about MACS at [www.windowsboston.com/downloads/doc/MACS\\_beta\\_Overview.doc](http://www.windowsboston.com/downloads/doc/MACS_beta_Overview.doc).

MACS is a real-time system that works only with Microsoft security event logs. Multiple computers' security log files can be collected to a centralized SQL database. Each participating client runs a MACS client service that communicates with the centralized server.

Microsoft also supports event log collection using SMS (<http://www.microsoft.com/smsserver/default.asp>) and Microsoft Operations Manager (<http://www.microsoft.com/mom/default.mspcx>), known as MOM.

## Standardizing Terms

Although setting up a central location for data may seem like a challenge, it is really the easy part. The hard part is a result of the fact that no two OSs or security devices collect and define data in the same way, leading to a "Tower of Babel" situation. For example, competing antivirus scanners are notorious for giving the same malware two different names. So, if you have log entries from two different antivirus platforms protecting your honeypot, one may report the SQL Slammer worm as SQL.Slammer.worm.A and another as SQL.Worm.32.

Even in a pure Windows environment, different events may be called different things or be handled differently between OS versions. For example, in Windows 2000, failed account logon events are logged to Event IDs 681 and 676. In Server 2003, they are logged to 680 and 672. In Windows 2000, RDP and Terminal Server logons are considered interactive logons. In Windows XP and later, they are called Type 10 Logons instead (see Table 5-3 above). The different versions of Windows contain many other auditing changes. There is no easy solution for this problem — administrators simply must become familiar with the common Event IDs used by each Windows version.

## Filtering Data

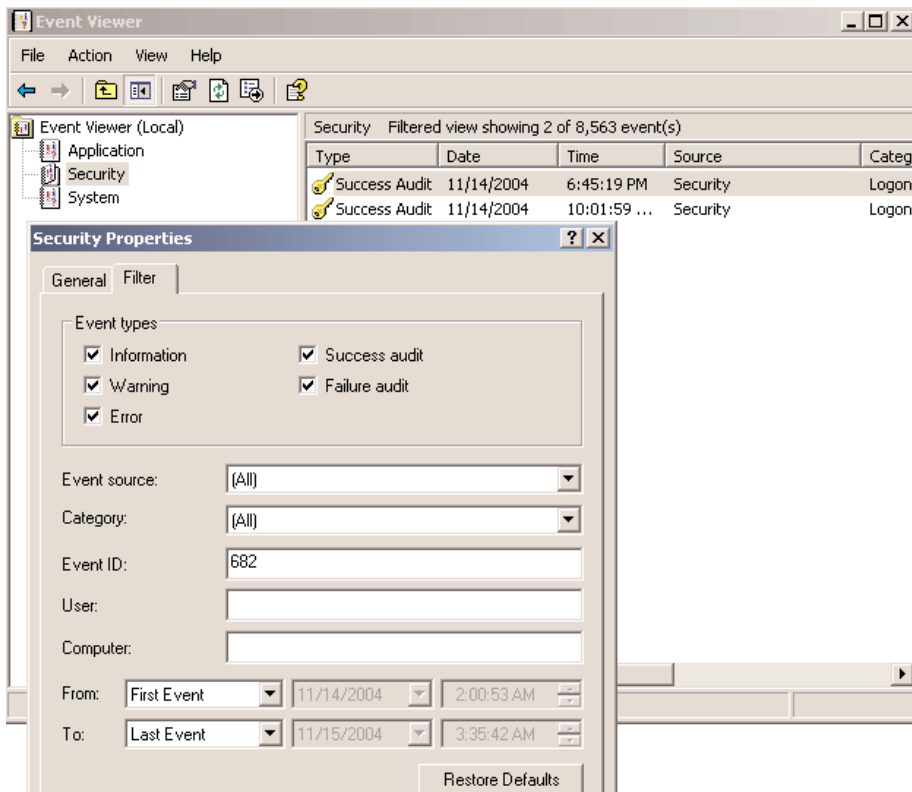
Once all the data is collected, it needs to be sorted and prioritized. High-priority items should be brought to the attention of the appropriate person, and low-priority messages should be filtered and saved to a log file. The hard part is determining what is and isn't high priority. Unfortunately, the Windows Security log does not come with built-in levels of criticality, like the other default logs have. And even if it did, a low-priority event to one administrator may be a high-priority event to another.

The trick is to develop a customized set of priorities for your environment based upon measured baselines, evaluated risks, and expected attacks. You should develop several levels of priority and then evaluate and sort all the different collected events into your priority levels.

Low-priority events should be simply recorded. High-priority events should be acted on immediately. However, never delete any security event log messages, because something innocuous might end up being important information later. Sometimes, what didn't happen at a particular time is as important as what did happen.

Many data collection systems also contain data filtering abilities. All of Microsoft's event log collection tools contain query features that you can use to sort collected events. Even the Event Viewer utility has simple query features. For example, Figure 5-7 shows the Event Viewer application filtering all events (tens of thousands) except Event 682 (Successful Re-Connection to a Winstation), which shows successful RDP reconnections, including the login name used. A user could construct a simple filter that queries for a handful of high-priority events.

**Figure 5-7**  
*Event Viewer filtering successful logins*



Most third-party event log collectors also have a query feature. A common query that could be constructed in most data filters could collect all Warning and Audit Failure events.

## Correlating Data

Data correlation is the grouping of data into useful sets of information instead of relying on one data point. A simple example is noting whether you are dealing with a hacker who is attempting to log on to one computer several times using various guesses, or one who is scanning your entire network for any weakly protected resources. The latter type of attack will be successful more often than the former.

Data correlation tools are just starting to make their debut. MOM (<http://www.microsoft.com/mom/default.mspx>) is Microsoft's data correlation tool. It includes predefined event ID and alert thresholds. A new microcosm of vendors specializing in *Security Event Management (SEM)*, also known as *Security Incident Management (SIM)* is cropping up. SEM/SIM vendors perform event log management tasks, as described in the preceding sections, and even merge vulnerability ratings, patch management, and risk exposure into the cycle. Vendors such as ArcSight (<http://www.arcsight.com>), netForensics (<http://www.netforensics.com>), Computer Associates (<http://www.ca.com>), and IBM (<http://www.ibm.com>) are heavily investing in their software, hardware, and managed service SEM/SIM products. See *InfoWorld* magazine's excellent summary article on SIM/SEM at [http://www.infoworld.com/infoworld/article/04/10/29/44FEbigsecure\\_1.html](http://www.infoworld.com/infoworld/article/04/10/29/44FEbigsecure_1.html) for more details. Microsoft is developing similar, robust solutions.

## Extracting Useful Information

After all this careful log planning, only the useful, critical, information should be culled and presented for action. For logged security events to provide you with useful information, they must be *relevant* to your environment. Relevancy in this case is a measure of how likely it is that a particular attack will be successful in your environment.

For example, do you care if a hacker tried Unix exploits against your Windows Server 2003 machines? Some system administrators do and some don't. The last piece in the puzzle in event log management is to make sure that only relevant critical events get passed to the administrator. Events with a low relevancy should be noted and logged but not forwarded to an administrator for action.

## Setting up an Alerting System

An event logging system must have a way of alerting appropriate people when a high-priority event occurs. Alert messages should be short, so that they can be sent to consoles, pagers, and cell phones, which are the preferred methods of receiving a high-priority alert. However, alert messages should also include enough information to let the responder assess the situation.

At a minimum, an alert message should carry the following information:

- Date and time of alert
- Message text indicating identified threat
- Priority
- Location of threat

The following line shows a sample alert message:

```
07-04-03 01:03:04.2345 High priority; Slammer probe; worm; DMZ IIS 6.0 Server2;
```

Typically, the alert is sent to an Internet e-mail address that corresponds to an alphanumeric pager or cell phone. Because sending messages via the Internet can sometimes be unreliable,

especially during a high-priority attack, many alert systems use dial-up modems that connect to a proprietary messaging service belonging to the pager or cell phone company.

Sending alerts via the console works only if you are on the local network when the alert is sent. Sending an alert via e-mail won't mean much if you aren't reading your e-mail that very second, and it will mean even less if it's buried in hundreds of other e-mail messages.

Alerting is more of an art than it sounds. If you simply set up an alerting mechanism to go off each time high-priority activity occurs, you could end up with a backlog of a hundred alerts in a few minutes. The alerting system must be smart enough to alert you only once for each related event; this setup is called *alert throttling* or *message throttling*. The idea is that after the system alerts you, it should sit idle for a predetermined amount of time if further activity appears to be coming from the same source and in the same priority level.

Also consider who should be alerted. If you are out of town or otherwise unavailable, who should respond in your place? You may even want to define response time guidelines according to the threat level. Whatever your alerting mechanism is, above all else, it should be reliable.

### Simple Windows Alerting Mechanisms

In Windows, you can use the NET SEND command, Msg.exe, and many other programs to send alerts. These programs can be used for sending short console messages across networks.

NET SEND has been around since at least Windows 95, but it may have been available in even earlier Microsoft products. NET SEND is a subcommand under the larger umbrella functionality available with the Net.exe program. Although the NET command is usually used to map drive shares (such as NET USER X: \\fileserver\sharename) or list users (NET USERS), it can also be used to send console messages. Each message arrives with a bell sound to alert any nearby users.

The Messenger service must be enabled on the computers involved. Because of potential spam message harassment, the Messenger service is disabled by default in Windows Server 2003 and Windows XP Service Pack 2. NET SEND can send messages to a user, domain, workgroup, or IP address. Messages can contain up to 128 characters. NET SEND's syntax is as follows:

```
NET SEND {<user> or /domain:<domain> or /users or <IPAddr>} <message>
```

The /domain parameter sends the supplied message to all users in the specified domain or workgroup. The /users option sends the message to all users with active connected sessions to the computer it is sent on. On Windows XP Service Pack 2, when sending to a single user, the /domain parameter must also be entered. The message can be plainly typed without any quotation marks unless you use non-text characters, such as a slash. Here are two NET SEND examples:

```
NET SEND admin There are failed logons on IIS 6 Server 3
NET SEND 192.168.1.56 "Disabled account has been enabled on FS3"
```

You can even incorporate external programs to extend the functionality of NET SEND. For example, with a bit of command-line coding and the free Showmbrs program, you can send messages to a Windows group (<http://www.jsiinc.com/SUBB/tip0700/rh0757.htm>). Other monitoring tools often use NET SEND as a quick and easy way to alert the administrator to activity in the honeypot, although it does not scale well over routed networks.

**Note**

Windows 9x computers need to run `Winpopup.exe` to accept NET SEND messages.

## Windows Event Triggers

Windows XP and Server 2003 even allow the NET SEND command to be triggered off a local or remote Windows event log message. The very useful and powerful `Eventtriggers.exe` program lets you create, delete, list, and query *trigger events*, as they are called (see Table 5-7 for syntax). Once created, trigger events are active until deleted, even surviving a system reboot.

The `EVENTTRIGGERS` command syntax is as follows:

```
EVENTTRIGGERS /Create [/S system [/U username [/P [password]]]] /TR triggername /TK
  taskname [/D description] [/L log] { [/EID id] [/T type] [/SO source] } [/RU username
  [/RP password]]
```

Type `EVENTTRIGGERS /?` or `EVENTTRIGGERS /Create /?` to see the full syntax options.

**Table 5-7 EVENTTRIGGERS /Create options**

Parameter	Variable	Description
/S	System	Specifies the remote system to connect to.
/U	[domain\]user	Specifies the user context under which the command should execute.
/P	[password]	Specifies the password for the given user context; prompts for input if omitted.
/TR	Triggername	Specifies a friendly name to associate with the event trigger.
/L	Log	Specifies the NT event log(s) to monitor events from. Valid types include Application, System, Security, DNS Server Log, and Directory Log. The wildcard (*) may be used, and the default value is *.
/EID	Id	Specifies a specific event ID the event trigger should monitor for.
/T	Type	Specifies an event type that the trigger should monitor for. Valid values include ERROR, INFORMATION, WARNING, SUCCESSAUDIT, and FAILUREAUDIT.
/SO	Source	Specifies a specific event source the event trigger should monitor for.
/D	Description	Specifies the description of the event trigger.
/TK	Taskname	Specifies the task to execute when the event trigger conditions are met.
/RU	Username	Specifies the user account (user context) under which the task runs. For the system account, the value must be "".
/RP	Password	Specifies the password for the user. To prompt for the password, the value must be either * or none. The password is not needed for the SYSTEM account.

You can create as many trigger events as you like and display them using the `EVENTTRIGGERS /query /v` command.

Trigger events can be used along with the NET SEND command for alerting purposes. For example, the following `EVENTTRIGGERS` command alerts the administrator if an invalid password is used during a login:

```
EVENTTRIGGERS.exe /create /L security /eid 529 /tr IncorrectLogon /tk "NET SEND
  administrator Incorrect Logon on FileServer1"
```

This trigger event, called `IncorrectLogon`, triggers event ID 529 (Bad Password or User Account Name) and sends a message to the administrator.

The next example triggers an alert if the security log is cleared:

```
Eventtriggers.exe /create /l security /eid 517 /tr LogCleared /tk "Net Send  
administrator IIS Log Cleared"
```

The `EVENTTRIGGERS` command is very versatile. See <http://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/eventtriggers.mspx> for more details.

Of course, Microsoft's MOM and SMS are perfect tools for event log alerting.

### Other Third-Party Alert Utilities

You can use a plethora of other utilities to send alerts from your honeypot or monitoring system. One of the most popular choices is a public domain utility called `Blat` (<http://www.blat.net>). `Blat` is a very small SMTP client that allows messages and files to be sent using the SMTP protocol to port 25 (or any other port number). It uses multiple sender profiles and retries if the receiving computer is busy. A DLL version can be directly installed and renamed to send messages directly from the honeypot. You can send messages with predefined subjects, messages, and attached files. It's perfect for sending alerts to e-mail systems or small-form computers and PDAs. Scripts and programs that need more functions than `NET SEND` can provide commonly use `Blat`.

Other third-party message sending programs include `Net Send Command Line` and `Net Send Lite` (<http://www.rjlsoftware.com>), `Febooti Command Line` (<http://www.febooti.com>), and `WinMessenger` (<http://www.vypress.com>). `ServerSentry` (<http://www.datatribe.net>) also monitors Windows event logs and services and sends trigger messages.

As you have learned, an event log management system entails more than simply turning on auditing and collecting event messages.

## Auditing Best Practices

Audit policy is best implemented with a little foresight and research. Here are some best practices to follow:

- Create an audit plan before turning on auditing
- Decide what events you will audit and where
- Decide how often logs will be reviewed and by whom
- Configure event log settings
- Configure using Domain Controller's policy and Domain policy
- Implement an audit log management tool
- Collect and archive audit logs and critical events across your organization

## Summary

Chapter 5 discussed event auditing as a primary tool in detecting attempted intrusions into your Windows system. If auditing is appropriately configured, it can capture critical information about most unauthorized events.

Chapter 6 will consider two-factor authentication and smart cards.