**Windows Logs**

|  |  |
| --- | --- |
| **Symptoms** | **Solutions** |
| Event ID 41: Kernel power error |  |
| Kernel power errors get logged when your system didn't shut down properly. You will see the message "The system has rebooted without cleanly shutting down first. This error could be caused if the system stopped responding, crashed, or lost power unexpectedly." | These errors are usually caused by issues in power supply. They are a concern especially when they occur on your critical assets. Your web servers and other critical servers must always be up and running. So, it is a must to track this event and be alerted if it occurs, so you can quickly take steps to resolve it. |
| Event ID 6008: Unexpected system shutdown |  |
| Event ID 6008 gets logged to the system event log when a system shuts down unexpectedly. You will see the message "The previous system shutdown at time on date was unexpected." | Unexpected system shutdowns need to be investigated immediately when they happen to your critical servers as they can affect business continuity. The system shutdown might be caused by a hardware issue, such as overheating or power supply, but could also be a security threat. |
| Event ID 7000: Service start failed |  |
| The Service Control Manager (SCM) logs this event when a service fails or hangs while starting. This is a serious concern for administrators because it can affect business continuity. The error message tells you why the service failed while starting. | With EventLog Analyzer, you can receive an alert for services that failed and for other related critical events. You can also search for other events logged by Service Control Manager to identify the cause for the error and quickly troubleshoot. |
| Event ID 7031: Service crash |  |
| Event ID 7031 gets logged when a service crashes. The Service Control Manager logs this event when a service stops unexpectedly. The message says which service failed, how many times it failed and the corrective action that will be taken. | You can change the automatic recovery actions that need to be taken when the service fails to ensure your critical services are always up and running. The recovery option can be set in Control Panel > Administrative Tools > Services |
| Event ID 4625: Failed logon |  |
| Event ID 4625 gets logged when an account fails to logon. The log data contains the information about the reason for the failed logon such as a bad username or password. | While failed logons occur routinely in your network, a sudden spike in failed logons would indicate a potential threat as it could be a brute force attack attempt. Administrators must keep tabs on failed logon activity and know the reason for the logon failure in order to ensure network security. |
| **Event ID 10028** |  |
| This event indicates a communication problem between the local computer and the one specified in the event. | The problem can be caused by several factors: - remote computer is offline - the network is experiencing problems (cabling, switches, routers, etc) - firewalls may block the traffic between the two computers - the DNS servers may be unavailable or they may provide the wrong IP address for that particular host name. Verify each of the potential issues mentioned above and ensure that they are not blocking the communication between the computers. |

|  |  |
| --- | --- |
| **Event ID 1000: Application error** |  |
| Event ID 1000 signifies a general application error. While they could be a routine browser crash, some application errors and crashes such as those on critical systems are a major concern for administrators and require their immediate attention. | The log data with event ID 1000 gives you information on the faulting application, module, process ID, and more. You will see other event IDs such as 1001, 1002 and 1004 pertaining to application errors logged along with event ID 1000 that might give you more information on what exactly happened with faulting application. |
| **Event ID 1001: Windows Error Reporting** |  |
| This error event gets logged by Windows Error Reporting infrastructure in the Application event log. This could be due to a variety of reasons such as blue screen errors or other application errors. | The information contained in the log message can help you analyze the reason for the error. By tracking associated application event IDs such as 1000 and 1002 logged in Event Viewer, you can diagnose the cause and take steps to troubleshoot. |
| Event ID 1002: Application Hang |  |
| Event ID 1002 occurs due to an application hang. While non-responsive applications are quite routine, an application that keeps hanging repeatedly can result in a dip in business productivity and must be checked. For example, a DHCP error can prevent users from connecting to the internet. | A hanging application usually fixes itself upon restart. The information in the log data gives you details on the faulting program and you can to look at Action Center for further information on it. |

**Linux Logs**

|  |
| --- |
| **Shutdown Logs** |
|  |
| Dec 28 21:03:38 ip-172-31-34-37 sshd[1172]: pam\_unix(sshd:session): session opened for user ubuntu by (uid=0)  Dec 28 21:03:38 ip-172-31-34-37 systemd: pam\_unix(systemd-user:session): session opened for user ubuntu by (uid=0)  Dec 28 21:03:41 ip-172-31-34-37 sudo: ubuntu : TTY=pts/0 ; PWD=/home/ubuntu ; USER=root ; COMMAND=/sbin/shutdown -r now |
| **Solution** |
| If someone ran the shutdown command manually, you can see it in the auth log file. Here you can see that someone remotely logged in as the user Ubuntu and then shut the system down. |
|  |
| **When server restarted or crashed (Kernel Initializing) logs** |
|  |
| Dec 28 21:04:22 ip-172-31-34-37 kernel: [ 0.000000] Linux version 4.15.0-1039-aws (buildd@lgw01-amd64-034) (gcc version 7.3.0 (Ubuntu 7.3.0-16ubuntu3)) #41-Ubuntu SMP Wed May 8 10:43:54 UTC 2019 (Ubuntu 4.15.0-1039.41-aws 4.15.18)  Dec 28 21:04:22 ip-172-31-34-37 kernel: [ 0.000000] Command line: BOOT\_IMAGE=/boot/vmlinuz-4.15.0-1039-aws root=LABEL=cloudimg-rootfs ro console=tty1 console=ttyS0 nvme\_core.io\_timeout=4294967295 |
| **Solution** |
| If you want to see when the server restarted regardless of reason (including crashes), you can search the kernel log file(/var/log/kern.log). Syslog also applies the “kern” facility to kernel logs. The following logs were generated immediately after boot. Note the timestamp between the brackets is 0: this tracks the amount of time since the kernel started. You can also find boot logs by searching for “BOOT\_IMAGE”. |
|  |
| Detect Memory Problems Logs |
|  |
| Dec 28 21:30:26 ip-172-31-34-37 kernel: [ 1575.404070] Out of memory: Kill process 16471 (memkiller) score 838 or sacrifice child  Dec 28 21:30:26 ip-172-31-34-37 kernel: [ 1575.408946] Killed process 16471 (memkiller) total-vm:144200240kB, anon-rss:562316kB, file-rss:0kB, shmem-rss:0kB  Dec 28 21:30:27 ip-172-31-34-37 kernel: [ 1575.518686] oom\_reaper: reaped process 16471 (memkiller), now anon-rss:0kB, file-rss:0kB, shmem-rss:0kB |
| **Solution** |
| There are several reasons a server might crash, but one common cause is running out of memory.  When RAM and swap space are completely exhausted, the kernel will [start killing processes](https://www.kernel.org/doc/gorman/html/understand/understand016.html)—typically those using the most memory and the most short-lived. The error occurs when your system is using all of its memory, and a new or existing process attempts to access additional memory. Look in your log files for strings like “Out of Memory” or for kernel warnings. These strings indicate your system intentionally killed the process or application rather than allowing the process to crash.  You can find these logs in the kernel log (/var/log/kern.log) or in the syslog (/var/log/syslog). You can use a tool like grep to search for the relevant entries:  $ grep "Out of memory" /var/log/syslog  Dec 28 21:30:26 ip-172-31-34-37 kernel: [ 1575.404070] Out of memory: Kill process 16471 (memkiller) score 838 or sacrifice child  Keep in mind grep itself uses memory, so you might cause an out-of-memory error just by running grep. This is another reason why it’s a fabulous idea to centralize your logs! |
|  |
| Logging Cron Job Errors |
| **Solution** |
| The cron daemon is a scheduler that runs commands at specified dates and times. If the process fails to run or fails to finish, then a cron error appears in your log files. You can find these files in /var/log/cron, /var/log/messages, and /var/log/syslog depending on your distribution. There are many reasons a cron job can fail. Usually the problems lie with the process rather than the cron daemon itself.  By default, cron jobs output to syslog and appear in the /var/log/syslog file. You can also redirect the output of your cron commands to another destination, such as standard output or another file. In this example, we pipe “Hello world” to the logger command. This creates two log events: one from cron, and one from the logger command. The -t parameter sets the app name to “helloCron”::  $ crontab -e  \*/5 \* \* \* \* echo 'Hello World' 2>&1 | /usr/bin/logger -t helloCron  Which creates the log entries:  Dec 28 22:20:01 ip-172-31-11-231 CRON[15296]: (ubuntu) CMD (echo 'Hello World!' 2>&1 | /usr/bin/logger -t helloCron)  Dec 28 22:20:01 ip-172-31-11-231 helloCron: Hello World!  Each cron job will log differently based on the specific type of job and how it outputs data. Hopefully there are clues to the root cause of problems within the logs, or you can add additional logging as needed. In most cases, you should simply let cron log the output of your commands. |