

Event Viewer - Windows

The log files with the .evtx file extension typically reside in C:\Windows\System32\winevt\Logs .

Elements of a Windows Event Log

Event logs are crucial for troubleshooting any computer incident and help understand the situation and how to remediate the incident. To get this picture well, you must first understand the format in which the information will be presented. Windows offers a standardized means of relaying this system information.

First, we need to know what elements form event logs in Windows systems. These elements are:

- **System Logs:** Records events associated with the Operating System segments. They may include information about hardware changes, device drivers, system changes, and other activities related to the device.
- **Security Logs:** Records events connected to logon and logoff activities on a device. The system's audit policy specifies the events. The logs are an excellent source for analysts to investigate attempted or successful unauthorized activity.
- **Application Logs :**Records events related to applications installed on a system. The main pieces of information include application errors, events, and warnings.
- **Directory Service Events:** Active Directory changes and activities are recorded in these logs, mainly on domain controllers.
- **File Replication Service Events:** Records events associated with Windows Servers during the sharing of Group Policies and logon scripts to domain controllers, from where they may be accessed by the users through the client servers.
- **DNS Event Logs:** DNS servers use these logs to record domain events and to map out.
- **Custom Logs:** Events are logged by applications that require custom data storage. This allows applications to control the log size or attach other parameters, such as ACLs, for security purposes.

There are three main ways of accessing these event logs within a Windows system:

1. **Event Viewer** (GUI-based application)
2. **Wevtutil.exe** (command-line tool)
3. **Get-WinEvent** (PowerShell cmdlet)

Operational Number of events: 44				
Level	Date and Time	Source	Event ID	Task Category
Information	12/10/2020 10:38:01 AM	PowerShell (Microsoft-Windows-PowerShell)	4103	Executing Pipeline

From the above image, notice the event provider's name and the number of events logged. In this case, there are 44 events logged. You might see a different number. No worries, though. Each column of the pane presents a particular type of information as described below:

- **Level:** Highlights the log recorded type based on the identified event types specified earlier. In this case, the log is labeled as **Information**.
- **Date and Time:** Highlights the time at which the event was logged.
- **Source:** The name of the software that logs the event is identified. From the above image, the source is PowerShell.
- **Event ID:** This is a predefined numerical value that maps to a specific operation or event based on the log source. This makes Event IDs not unique, so Event ID 4103 in the above image is related to Executing Pipeline but will have an entirely different meaning in another event log.
- **Task Category:** Highlights the Event Category. This entry will help you organize events so the Event Viewer can filter them. The event source defines this column.

The middle pane has a split view. More information is displayed in the bottom half of the middle pane for any event you click on.

This section has two tabs: **General** and **Details**.

- General is the default view, and the rendered data is displayed.
- The Details view has two options: Friendly view and XML view.

Below is a snippet of the General view.

Event 4103, PowerShell (Microsoft-Windows-PowerShell)

General Details

CommandInvocation(PSConsoleHostReadLine): "PSConsoleHostReadLine"

Context:

- Severity = Informational
- Host Name = ConsoleHost
- Host Version = 5.1.17763.592
- Host ID = 3154c106-cb32-4b2f-8aee-ba629a2e1ce8
- Host Application = C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe

Log Name: Microsoft-Windows-PowerShell/Operational

Source: PowerShell (Microsoft-Wind

Logged: 12/10/2020 10:38:01 AM

Event ID: 4103

Task Category: Executing Pipeline

Level: Information

Keywords: None

User: WIN-100UJBNP9G7\Admini

Computer: WIN-100UJBNP9G7

OpCode: To be used when operation i

More Information: [Event Log Online Help](#)

wevtutil.exe

Ok, you played around with Event Viewer. Imagine you have to sit there and manually sift through hundreds or even thousands of events (even after filtering the log). Not fun. It would be nice if you could write scripts to do this work for you. We will explore some tools that will allow you to query event logs via the command line and/or PowerShell.

The wevtutil.exe tool "enables you to retrieve information about event logs and publishers. You can also use this command to install and uninstall event manifests, to run queries, and to export, archive, and clear logs."

As with any tool, access its help files to find out how to run the tool. An example of a command to do this is `wevtutil.exe /?`.

Powershell - wevtutil.exe

```
PS> C:\Users\Administrator> wevtutil.exe /?
Windows Events Commandline Utility.
Enables you to retrieve information about event logs and publishers, install
and uninstall event manifests, run queries, and export, archive and clear
logs.
```

Usage:

You can use either the short (for example, `ep /uni`) or long (for example, `enum-publishers /unicode`) version of the command and option names. Commands, options and option values are not case-sensitive.

Variables are noted in all upper-case.

```
wevtutil COMMAND [ARGUMENT [ARGUMENT] ...] [/OPTION:VALUE [/OPTION:VALUE]
...]
```

Commands:

<code>el</code>	<code> enum-logs</code>	List log names.
<code>gl</code>	<code> get-log</code>	Get log configuration information.
<code>sl</code>	<code> set-log</code>	Modify configuration of a log.
<code>ep</code>	<code> enum-publishers</code>	List event publishers.
<code>gp</code>	<code> get-publisher</code>	Get publisher configuration information.
<code>im</code>	<code> install-manifest</code>	Install event publishers and logs from manifest.
<code>um</code>	<code> uninstall-manifest</code>	Uninstall event publishers and logs from manifest.
<code>qe</code>	<code> query-events</code>	Query events from a log or log file.
<code>gli</code>	<code> get-log-info</code>	Get log status information.
<code>epl</code>	<code> export-log</code>	Export a log.
<code>al</code>	<code> archive-log</code>	Archive an exported log.

```
cl | clear-log          Clear a log.
```

From the above snippet, under **Usage**, you are provided a brief example of how to use the tool. In this example, `ep` (**enum-publishers**) is used. This is a **command** for `wevtutil.exe`.

Below, we can find the **Common options** that can be used with Windows Events Utility.

Powershell - `wevtutil.exe`

Common Options:

```
{r | remote}:VALUE
```

If specified, run the command on a remote computer. VALUE is the remote computer name. Options `/im` and `/um` do not support remote operations.

```
{u | username}:VALUE
```

Specify a different user to log on to the remote computer. VALUE is a user name in the form of `domain\user` or `user`. Only applicable when option `/r` is specified.

```
{p | password}:VALUE
```

Password for the specified user. If not specified, or if VALUE is `"*"`, the user will be prompted to enter a password. Only applicable when the `/u` option is specified.

```
{a | authentication}:[Default|Negotiate|Kerberos|NTLM]
```

Authentication type for connecting to remote computer. The default is `Negotiate`.

```
{uni | unicode}:[true|false]
```

Display output in Unicode. If `true`, then output is in Unicode.

To learn more about a specific command, type the following:

```
wevtutil COMMAND /?
```

Notice at the bottom of the above snapshot, `wevtutil COMMAND /?`. This will provide additional information specific to a command. We can use it to get more information on the command `qe` (**query-events**).

Powershell - `wevtutil.exe`

```
PS> C:\Users\Administrator> wevtutil qe /?
```

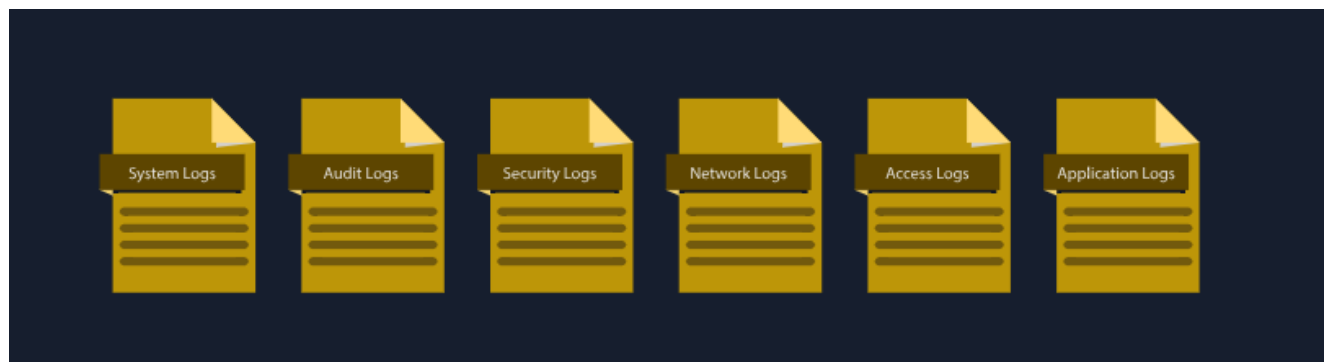
```
Read events from an event log, log file or using a structured query.
```

```
Usage:
```

```
wevtutil {qe | query-events} [/OPTION:VALUE [/OPTION:VALUE]...]
```

Look over the information within the help menu to fully understand how to use this command.

Types Of Logs



Log Type	Usage	Example
System Logs	The system logs can be helpful in troubleshooting running issues in the OS. These logs provide information on various operating system activities.	<ul style="list-style-type: none">- System Startup and shutdown events- Driver Loading events- System Error events- Hardware events
Security Logs	The security logs help detect and investigate incidents. These logs provide information on the security-related activities in the system.	<ul style="list-style-type: none">- Authentication events- Authorization events- Security Policy changes events- User Account changes events- Abnormal Activity events
Application Logs	The application logs contain specific events related to the application. Any interactive or non-interactive activity happening inside the application will be logged here.	<ul style="list-style-type: none">- User Interaction events- Application Changes events- Application Update events- Application Error events
Audit Logs	The Audit logs provide detailed information on the system changes and user events. These logs are helpful for compliance requirements and can play a vital role in security monitoring as well.	<ul style="list-style-type: none">- Data Access events- System Change events- User Activity events- Policy Enforcement events
Network Logs	Network logs provide information on the network's outgoing and incoming traffic. They play crucial roles in troubleshooting network issues and can also be handy during incident investigations.	<ul style="list-style-type: none">- Incoming Network Traffic events- Outgoing Network Traffic events- Network Connection Logs - Network Firewall Logs
Access Logs	The Access logs provide detailed information about the access to different resources. These resources can be of different types, providing us with information on their access.	<ul style="list-style-type: none">- Webserver Access Logs- Database Access Logs - Application Access Logs- API Access Logs

This is how a Windows event log looks. It has different fields. The major fields are discussed below:

Description: This field has a detailed information of the activity.

Log Name: The Log Name indicates the log file name.

Logged: This field indicates the time of the activity.
Event ID: Event IDs are unique identifiers for a specific activity.

Numerous event IDs are available in Windows event logs. We can use these event IDs to search for any specific activity. For example, event ID 4624 uniquely identifies the activity of a successful login, so you only need to search for this event ID 4624 when investigating successful logins.

Here is a table of some important Event IDs in Windows Operating System.

Event ID	Description
4624	A user account successfully logged in
4625	A user account failed to login
4634	A user account successfully logged off
4720	A user account was created
4724	An attempt was made to reset an account's password
4722	A user account was enabled
4725	A user account was disabled
4726	A user account was deleted

Web Server Access Logs Analysis

Apache web server access log file which can be found in the directory:
`/var/log/apache2/access.log`

- **IP Address:** “172.16.0.1” - The IP address of the user who made the request.
- **Timestamp:** “[06/Jun/2024:13:58:44]” - The time when the request was made to the website.
- **Request:** The request details.

- **HTTP Method:** “GET” - Tells the website what action to be performed on the request.
- **URL:** “/” - The requested resource.
- **Status Code:** “200” - The response from the server. Different numbers indicate different response results.
- **User-Agent:** “Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_3) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.110 Safari/537.36” - Information about the user’s Operating System, browser, etc. when making the request.

cat is a popular utility for displaying the contents of a text file. We can use the cat command to display the contents of a log file, as they are typically in the text format.

cat command usage

```
root@kali$ cat access.log
172.16.0.1 - - [06/Jun/2024:13:58:44] "GET /products HTTP/1.1" 404 "-"
"Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/58.0.3029.110 Safari/537.36"
10.0.0.1 - - [06/Jun/2024:13:57:44] "GET / HTTP/1.1" 404 "-" "Mozilla/5.0
(Macintosh; Intel Mac OS X 10_12_3) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/58.0.3029.110 Safari/537.36"
192.168.1.1 - - [06/Jun/2024:13:56:44] "GET /about HTTP/1.1" 500 "-" "Moz
```

we may need to combine two log files. The cat command-line utility can also be helpful in this case. We can combine the results of multiple files within one single file, as shown below.

cat command usage

```
root@kali$ cat access1.log access2.log > combined_access.log
```

grep

It is a very useful command line utility that allows you to search for strings and patterns inside a log file. For example, you may need to search if a specific IP address is present in your log file. You can do this by using the following command: The following command will search the access.log file for the string ‘192.168.1.1’ and display all the lines with this string.

grep command usage

```
root@kali$ grep "192.168.1.1" access.log
192.168.1.1 - - [06/Jun/2024:13:56:44] "GET /about HTTP/1.1" 500 "-"
"Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/58.0.3029.110 Safari/537.36"
192.168.1.1 - - [06/Jun/2024:13:53:44] "GET /products HTTP/1.1" 404 "-"
"Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
```

```
Gecko) Chrome/58.0.3029.110 Safari/537.36"
192.168.1.1 - - [06/Jun/2024:13:46:44] "GET /about HTTP/1.1" 200 "-" "Mo
```

The `less` command is helpful for handling multiple log files. You may need to analyze specific chunks one by one. For this, you can use the `less` command-line utility, which helps you view one page at a time.

less command usage

```
root@kali$ less access.log
172.16.0.1 - - [06/Jun/2024:13:52:44] "GET /products HTTP/1.1" 404 "-"
"Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/58.0.3029.110 Safari/537.36"
10.0.0.1 - - [06/Jun/2024:13:48:44] "GET /about HTTP/1.1" 404 "-"
"Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/58.0.3029.110 Safari/537.36"
192.168.1.1 - - [06/Jun/2024:13:46:44] "GET /about HTTP/1.1" 200 "-"
"Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/58.0.3029.110 Safari/537.36":
```