


How it works


 Note

This guide is based on the official [Audit guide](#).

Audit uses a set of rules to define what is to be captured in the log files. There are three types of Audit rules that can be specified:

- **Control rules** allow the Audit system’s behavior and some of its configuration to be modified.
- **File system rules**, also known as file watches, allow the auditing of access to a particular file or a directory.
- **System call rules** allow logging of system calls that specified programs makes.

Audit rules can be specified interactively with the *auditctl* command-line utility, but to make changes persistent, edit */etc/audit/audit.rules*.

 Warning

All commands that interact with the Audit service and the Audit log files require root privileges, so you will need to be root or use `sudo` to execute these commands.

Control rules

Some examples that illustrate how to modify the behaviour of the Audit system:

auditctl -b	Set the maximum amount of existing Audit buffers in the kernel.
auditctl -e	Enable/disable the Audit system or lock its configuration.
auditctl -s	Report the status of the Audit system.
auditctl -l	List all currently loaded Audit rules.
auditctl -D	Delete all currently loaded Audit rules.

File System Rules

To define a file system rule, use the following syntax:

```
-w <path> -p <permissions> -k <key_name>
```

where:

-w <path>	Specify what file or directory to audit with <path>		
-p <permissions>	<permissions> are the permissions that are to auditing, including the following:		
	Values	r	read access to a file or a directory.
		w	write access to a file or a directory.
		x	execute access to a file or a directory.
		a	change in the file’s or directory’s attribute.
-k <key_name>	<key_name> is an optional string to identify which rule/set of rules generates a particular log line. This argument is required by Wazuh in order to analyze the logs more accurately.		

For example, to define a rule that logs all write access to, and every attribute change of, the */etc/passwd* file, execute the following command::

```
$ auditctl -w /etc/passwd -p wa -k passwd_changes
```

System Call Rules

To define a system call rule, use the following syntax::

```
-a action,filter -S system_call -F field=value -k key_name
```

where:

-a <action>, <filter>	Tells the kernel's rule matching engine to append a rule at the end of the rule list.		
	We must specify which rule list to append it to and what action to take when it triggers.		
	<action>	always	read access to a file or a directory.
		never	write access to a file or a directory.
	The <filter> value specifies which kernel rule-matching filter is applied to the event		
	<filter>	task	Only audit events fork or clone syscalls. This is rarely used in practice.
exit		All syscall and file system audit requests are evaluated.	
user		This is used to remove some events that originate in user space. By default, any event originating in user space is allowed.	
exclude		This is used to exclude certain events from being logged. <i>msgtype</i> is used to tell the kernel which message to filter out. For more granular control over which events to audit: use the user and exit filters instead.	
-S <system_call>	This specifies which <i>system_call</i> to audit. Multiple system calls can be specified in a single rule. A list of all system calls can be found with the command <code>ausyscall --dump</code> .		
-F <field=value>	Use <i>field=value</i> to specify additional criteria to narrow down which events to audit, based on: architecture, group ID, process ID, etc..., Multiple -F options can be used in a single rule.		
-k <key_name>	<key_name>is an optional string to identify which rule/set of rules generates a particular log line. This argument is required by Wazuh in order to analyze the logs more accurately.		

For example, to define a rule that creates a log entry every time a file is deleted or renamed by a system user whose ID is 500 or larger, use the following. Note that the *-F auid!=4294967295* option is used to exclude users whose login UID is not set.

```
$ auditctl -a always,exit -S unlink -S unlinkat -S rename -S renameat -F auid>=500 -F auid!=4294967295 -k delete
```

It is also possible to define a file system rule using the system call rule syntax. The following command creates a rule for system calls that is analogous to the **-w /etc/shadow -p wa** file system rule::

```
$ auditctl -a always,exit -F path=/etc/shadow -F perm=wa
```