Hat ⌄

Table of co

7.6.

PDF

By defa otation
is enabl

The following Audit rule logs every attempt to read or modify the `/etc/ssh/sshd_config` file:

```
-w /etc/ssh/sshd_config -p warx -k sshd_config
```

If the `auditd` daemon is running, for example, using the following command creates a new event in the Audit log file:

```
~]$ cat /etc/ssh/sshd_config
```

This event in the `audit.log` file looks as follows:

```
type=SYSCALL msg=audit(1364481363.243:24287): arch=c000003e syscall=2 success=no exit=-13 a0=7f
type=CWD msg=audit(1364481363.243:24287):  cwd="/home/shadowman"
```

```
type=PATH msg=audit(1364481363.243:24287): item=0 name="/etc/ssh/sshd_config" inode=409248 dev=
type=PROCTITLE msg=audit(1364481363.243:24287) : proctitle=636174002F6574632F7373682F737368645F
```

The above event consists of four records, which share the same time stamp and serial number. Records always start with the `type=` keyword. Each record consists of several `name=value` pairs separated by a white space or a comma. A detailed analysis of the above event follows:

First Record

`type=SYSCALL`

The `type` field contains the type of the record. In this example, the `SYSCALL` value specifies that this record was triggered by a system call to the kernel.

For a list of all possible type values and their explanations, see Audit Record Types.

`msg=audit(1364481363.243:24287):`

The `msg` field records:

- a time stamp and a unique ID of the record in the form `audit(time_stamp:ID)`. Multiple records can share the same time stamp and ID if they were generated as part of the same Audit event. The time stamp is using the Unix time format - seconds since 00:00:00 UTC on 1 January 1970.

- various event-specific `name=value` pairs provided by the kernel or user space applications.

`arch=c000003e`

The `arch` field contains information about the CPU architecture of the system. The value, `c000003e`, is encoded in hexadecimal notation. When searching Audit records with the `ausearch` command, use the `-i` or `--interpret` option to automatically convert hexadecimal values into their human-readable equivalents. The `c000003e` value is interpreted as `x86_64`.

`syscall=2`

The `syscall` field records the type of the system call that was sent to the kernel. The value, `2`, can be matched with its human-readable equivalent in the `/usr/include/asm/unistd_64.h` file. In this case, `2` is the `open` system call. Note that the **ausyscall** utility allows you to convert system call numbers to their human-readable equivalents. Use the `ausyscall --dump` command to display a listing of all system calls along with their numbers. For more information, see the ausyscall(8) man page.

`success=no`

The `success` field records whether the system call recorded in that particular event succeeded or failed. In this case, the call did not succeed.

`exit=-13`

The `exit` field contains a value that specifies the exit code returned by the system call. This value varies for different system call. You can interpret the value to its human-readable equivalent with the following command:

```
~]# ausearch --interpret --exit -13
```

Note that the previous example assumes that your Audit log contains an event that failed with exit code `-13`.

`a0=7fffd19c5592`, `a1=0`, `a2=7fffd19c5592`, `a3=a`

The `a0` to `a3` fields record the first four arguments, encoded in hexadecimal notation, of the system call in this event. These arguments depend on the system call that is used; they can be interpreted by the **ausearch** utility.

`items=1`

The `items` field contains the number of PATH auxiliary records that follow the syscall record.

`ppid=2686`

The `ppid` field records the Parent Process ID (PPID). In this case, `2686` was the PPID of the parent process such as `bash`.

`pid=3538`

The `pid` field records the Process ID (PID). In this case, `3538` was the PID of the `cat` process.

`auid=1000`

The `auid` field records the Audit user ID, that is the loginuid. This ID is assigned to a user upon login and is inherited by every process even when the user's identity changes, for example, by switching user accounts with the `su - john` command.

`uid=1000`

The `uid` field records the user ID of the user who started the analyzed process. The user ID can be interpreted into user names with the following command: `ausearch -i --uid UID`.

`gid=1000`

The `gid` field records the group ID of the user who started the analyzed process.

`euid=1000`

The `euid` field records the effective user ID of the user who started the analyzed process.

`suid=1000`

The `suid` field records the set user ID of the user who started the analyzed process.

`fsuid=1000`

The `fsuid` field records the file system user ID of the user who started the analyzed process.

`egid=1000`

The `egid` field records the effective group ID of the user who started the analyzed process.

`sgid=1000`

The `sgid` field records the set group ID of the user who started the analyzed process.

`fsgid=1000`

The `fsgid` field records the file system group ID of the user who started the analyzed process.

`tty=pts0`

The `tty` field records the terminal from which the analyzed process was invoked.

`ses=1`

The `ses` field records the session ID of the session from which the analyzed process was invoked.

`comm="cat"`

The `comm` field records the command-line name of the command that was used to invoke the analyzed process. In this case, the `cat` command was used to trigger this Audit event.

```
exe="/bin/cat"
```

The `exe` field records the path to the executable that was used to invoke the analyzed process.

```
subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

The `subj` field records the SELinux context with which the analyzed process was labeled at the time of execution.

```
key="sshd_config"
```

The `key` field records the administrator-defined string associated with the rule that generated this event in the Audit log.

Second Record

```
type=CWD
```

In the second record, the `type` field value is `CWD` — current working directory. This type is used to record the working directory from which the process that invoked the system call specified in the first record was executed.

The purpose of this record is to record the current process's location in case a relative path winds up being captured in the associated PATH record. This way the absolute path can be reconstructed.

```
msg=audit(1364481363.243:24287)
```

The `msg` field holds the same time stamp and ID value as the value in the first record. The time stamp is using the Unix time format - seconds since 00:00:00 UTC on 1 January 1970.

```
cwd="/home/user_name"
```

The `cwd` field contains the path to the directory in which the system call was invoked.

Third Record

`type=PATH`

In the third record, the `type` field value is `PATH`. An Audit event contains a `PATH`-type record for every path that is passed to the system call as an argument. In this Audit event, only one path (`/etc/ssh/sshd_config`) was used as an argument.

`msg=audit(1364481363.243:24287):`

The `msg` field holds the same time stamp and ID value as the value in the first and second record.

`item=0`

The `item` field indicates which item, of the total number of items referenced in the `SYSCALL` type record, the current record is. This number is zero-based; a value of `0` means it is the first item.

`name="/etc/ssh/sshd_config"`

The `name` field records the path of the file or directory that was passed to the system call as an argument. In this case, it was the `/etc/ssh/sshd_config` file.

`inode=409248`

The `inode` field contains the inode number associated with the file or directory recorded in this event. The following command displays the file or directory that is associated with the `409248` inode number:

```
~]# find / -inum 409248 -print
/etc/ssh/sshd_config
```

`dev=fd:00`

The `dev` field specifies the minor and major ID of the device that contains the file or directory recorded in this event. In this case, the value represents the `/dev/fd/0` device.

`mode=0100600`

The `mode` field records the file or directory permissions, encoded in numerical notation as returned by the `stat` command in the `st_mode` field. See the `stat(2)` man page for more information. In this case, `0100600` can be interpreted as `-rw-------`, meaning that only the root user has read and write permissions to the `/etc/ssh/sshd_config` file.

`ouid=0`

The `ouid` field records the object owner's user ID.

`ogid=0`

The `ogid` field records the object owner's group ID.

`rdev=00:00`

The `rdev` field contains a recorded device identifier for special files only. In this case, it is not used as the recorded file is a regular file.

`obj=system_u:object_r:etc_t:s0`

The `obj` field records the SELinux context with which the recorded file or directory was labeled at the time of execution.

`objtype=NORMAL`

The `objtype` field records the intent of each path record's operation in the context of a given syscall.

`cap_fp=none`

The `cap_fp` field records data related to the setting of a permitted file system-based capability of the file or directory object.

`cap_fi=none`

The `cap_fi` field records data related to the setting of an inherited file system-based capability of the file or directory object.

`cap_fe=0`

The `cap_fe` field records the setting of the effective bit of the file system-based capability of the file or directory object.

`cap_fver=0`

The `cap_fver` field records the version of the file system-based capability of the file or directory object.

Fourth Record

`type=PROCTITLE`

The `type` field contains the type of the record. In this example, the `PROCTITLE` value specifies that this record gives the full command-line that triggered this Audit event, triggered by a system call to the kernel.

```
proctitle=636174002F6574632F7373682F737368645F636F6E666967
```

The `proctitle` field records the full command-line of the command that was used to invoke the analyzed process. The field is encoded in hexadecimal notation to not allow the user to influence the Audit log parser. The text decodes to the command that triggered this Audit event. When searching Audit records with the `ausearch` command, use the `-i` or `--interpret` option to automatically convert hexadecimal values into their human-readable equivalents. The `636174002F6574632F7373682F737368645F636F6E666967` value is interpreted as `cat /etc/ssh/sshd_config`.

The Audit event analyzed above contains only a subset of all possible fields that an event can contain. For a list of all event fields and their explanation, see Audit Event Fields. For a list of all event types and their explanation, see Audit Record Types.

Example 7.6. Additional `audit.log` Events

The following Audit event records a successful start of the `auditd` daemon. The `ver` field shows the version of the Audit daemon that was started.

```
type=DAEMON_START msg=audit(1363713609.192:5426): auditd start, ver=2.2 format=raw kernel=2.6
```

The following Audit event records a failed attempt of user with UID of 1000 to log in as the root user.

```
type=USER_AUTH msg=audit(1364475353.159:24270): user pid=3280 uid=1000 auid=1000 ses=1 subj=u
```

Previous

Next