Use of sudo commands on a server

The sudo command provides authenticated users elevated access to the system. This access might result in undesirable changes leading to an incident or outage. You want the ability to see which sudo commands were executed on the server prior to an incident, and by whom, so you can quickly identify root cause or determine who to consult for further investigation.

Data required

*nix: Security logs

Procedure

1. Install the Splunk Add-on for Unix and Linux.
2. Run the following search. You can optimize it by specifying an index and adjusting the time range.

   sourcetype= linux_secure process= sudo COMMAND= * host= *
   | rex "COMMAND= (?<raw_command> . *)"
   | eval COMMAND= coalesce (raw_command, COMMAND)
   | table _time host USER PWD COMMAND

Search explanation

The table provides an explanation of what each part of this search achieves. You can adjust this query based on the specifics of your environment.

<table>
<thead>
<tr>
<th>Splunk Search</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>sourcetype= linux_secure</td>
<td>Search only the linux_secure source type.</td>
</tr>
<tr>
<td>process= sudo COMMAND= * host= *</td>
<td>Return all sudo command processes on any host.</td>
</tr>
<tr>
<td>rex &quot;COMMAND= (?&lt;raw_command&gt; . *)&quot;</td>
<td>Capture the entire command text and name it raw_command.</td>
</tr>
<tr>
<td>eval COMMAND= coalesce (raw_command,</td>
<td>Return commands that are set in different ways than</td>
</tr>
</tbody>
</table>

The information provided in Splunk Lantern is intended for informational and educational purposes only. All information is provided in good faith, however, Splunk disclaims any and all representations and warranties, express and implied, regarding the information provided, including without limitation any warranties and representations regarding the completeness, adequacy or accuracy of the information. You agree to take full responsibility for the results arising from the use of the information provided.
**Next steps**

The resulting table shows which users ran which command with administrative privileges. Knowing this information can identify problems with the system that are the result of system configuration changes. For example, if user Tom used sudo to edit important config files found in the `/etc`, you might be able to confirm those changes caused a service to malfunction.

<table>
<thead>
<tr>
<th>_time</th>
<th>host</th>
<th>user</th>
<th>pwd</th>
<th>command</th>
</tr>
</thead>
<tbody>
<tr>
<td>2020-09-28T20:21:01.000+0000</td>
<td>10.2.1.35</td>
<td>root</td>
<td>/etc/apache2/sites-enabled</td>
<td>/etc/init.d/apache2 reload</td>
</tr>
<tr>
<td>2020-09-28T20:17:21.000+0000</td>
<td>10.2.3.35</td>
<td>adm_abapiwa</td>
<td>/etc/apache2/sites-enabled</td>
<td>/usr/bin/vim convergence_wsgi</td>
</tr>
<tr>
<td>2020-09-28T20:16:24.000+0000</td>
<td>10.2.5.1</td>
<td>root</td>
<td>/etc/apache2/sites-disabled</td>
<td>/bin/ln -s ../sites-available/default-ssl 000-default-ssl</td>
</tr>
<tr>
<td>2020-09-28T20:16:04.000+0000</td>
<td>10.2.1.33</td>
<td>adm_SamiiHoran</td>
<td>/etc/apache2/sites-enabled</td>
<td>/etc/init.d/apache2 reload</td>
</tr>
</tbody>
</table>

Note that the above sample output shows two instances of root as a user, which indicates direct root access to the system. It is a best practice to not allow direct root access to the system but rather to always require sudo to gain access to privileged functions. In the case above, we have no idea what identity was behind the root user who logged in to 10.2.1.35. However, we do for host 10.2.3.35 and for 10.2.1.33. This makes it possible to trace back privileged actions.
to known users on the system rather than unknown users who gain direct access as root but know the root password.

The above report is also useful for compliance because many controls, such as the CIS Controls, state that an organization must show controlled use of administrative privileges. Sudo is one way to control such privileges and the report generated by this search can be used to discover and give attestation to compliance to the control. Controlling administrative privileges is not only a security measure but also an aid in root cause for misconfigurations.

The data in this example is usually found in the Linux auth.log and is given the sourcetype of linux_secure by the Splunk Add-on. Below are some sample full events.

```
Oct 23 19:43:12 acmepayroll sshd[29253]: Accepted password for root from 10.11.36.11 port 2958 ssh2
```

```
Oct 23 19:39:30 HOST0170 sshd[25089]: [ID 800047 auth.info] Accepted publickey for naughtyuser from 10.11.36.49 port 50241 ssh2
```

```
Oct 23 19:36:55 HOST0170 sshd[25089]: [ID 800047 auth.info] Accepted publickey for naughtyuser from 10.11.36.5 port 50241 ssh2
```

Interesting fields that are extracted by the add-on include dest (destination), pid (process id), process (process name), src_port (port of the authentication process), sshd_protocol, and user. These fields can be used to analyze other authentication related metrics, such as users logged in at the same time from multiple remote locations.

Finally, you might also want to look at other similar searches to this in our article Managing *nix system user account behavior.