Documentation › User manual › Reference › Local configuration › command

command

XML section name

```
<command>
</command>
```

In a command configuration section, you define a command to be used by one or more active responses. It is possible to have as many commands as needed, but each one must be in their own separate <command> section.

Options

- name
- executable
- expect
- timeout_allowed

name

This field specifies the name of the command which is called in the active-response section.

Default value	n/a
Allowed values	Any name
use	Required

executable

This must be a file (with the execute permission set) inside `/var/ossec/active-response/bin`. You don't need to provide the path.

Default value	n/a
Allowed values	Any file name
use	Required

expect

This is a list of zero or more names of extracted fields that are to be passed as parameters to the command. If any of the listed fields were not extracted in a certain instance, those field values would be passed as a dash (`-`) instead of as no value at all. A good example is the firewall-block command which expects the `srcip` field so it knows which IP to block. Multiple expected field names are comma separated.

Default value	n/a
Allowed values	Names of extracted fields, like **srcip** or **username**, separated by commas if there is more than one.
use	Required

Note

You can specify no fields by using `<expect></expect>`. That is the valid setting when no options need to be passed to the active-response command.

timeout_allowed

If yes, this indicates that the command is stateful, and will be called again in a certain length of time and instructed to undo its original action.

Default value	yes
Allowed values	yes/no

Example of configuration

```xml
<!-- For Unix systems -->
<command>
  <name>host-deny</name>
  <executable>host-deny.sh</executable>
  <expect>srcip</expect>
  <timeout_allowed>yes</timeout_allowed>
</command>

<!-- For Windows systems -->
<command>
  <name>win_route-null</name>
  <executable>route-null.cmd</executable>
  <expect>srcip</expect>
  <timeout_allowed>yes</timeout_allowed>
</command>
```