Centralized configuration

Introduction

Agents can be configured remotely by using the `agent.conf` file. The following capabilities can be configured remotely:

- File Integrity monitoring (**syscheck**)
- Rootkit detection (**rootcheck**)
- Log data collection (**localfile**)
- Security policy monitoring (**rootcheck**, **wodle name="open-scap"**)
- Anti-flooding mechanism (**bucket options**)
- Labels for agent alerts (**labels**)

Below, is the proper syntax of `agent.conf` and the process of pushing the configuration from the manager to the agent.

agent.conf

XML section name

```
<agent_config>
    ...
</agent_config>
```

The `agent.conf` is only valid on server installations.

The `agent.conf` exists in `/var/ossec/etc/shared`. It should be readable by the ossec user.

Options

name	Allows assignment of the block to one particular agent.	
	Allowed values	Any agent name
os	Allows assignment of the block to an operating system.	
	Allowed values	Any OS family
profile	Allows assignment of a profile name to a block. Any agent configured to use the defined profile may use the block.	
	Allowed values	Any defined profile

Examples

```
<agent_config name="agent01">
...
<agent_config os="Linux">
...
<agent_config profile="UnixHost">
```

Centralized configuration process

Here we are going to explain how a centralized configuration can be done.

1. Configuration

Edit the file `/var/ossec/etc/shared/agent.conf`. If the file does not exist, create it:

```
$ touch /var/ossec/etc/shared/agent.conf
$ chown ossec:ossec /var/ossec/etc/shared/agent.conf
$ chmod 640 /var/ossec/etc/shared/agent.conf
```

Several configurations may be created according to the `name`, `OS` or `profile` of an agent.

```
<agent_config name="agent_name">
    <localfile>
        <location>/var/log/my.log</location>
        <log_format>syslog</log_format>
    </localfile>
</agent_config>

<agent_config os="Linux">
    <localfile>
        <location>/var/log/linux.log</location>
        <log_format>syslog</log_format>
    </localfile>
</agent_config>

<agent_config profile="database">
    <localfile>
        <location>/var/log/database.log</location>
        <log_format>syslog</log_format>
    </localfile>
</agent_config>
```

2. Run /var/ossec/bin/verify-agent-conf and if any errors are reported, fix them and return to step one. Failure to perform this step may allow errors to be pushed to agents, preventing them from running. If that happens, you may be forced to visit each agent manually to recover them.
3. Push of the configuration to the agents

Each time agents check-in to the manager (10 minute default), they pull a fresh copy of `agent.conf` if a new version is available. However, the new `agent.conf` is not used by the agent until the next time the agent is restarted in step 5. Restarting the manager will speed up how quickly it makes the new `agent.conf` available to the agents.

4. Check if the agent received the configuration

Once an agent received the configuration, the “Client version” field will have the md5sum of the `agent.conf` file.

```
$ md5sum /var/ossec/etc/shared/agent.conf
078b0711a8b2ee8b18e839afdafe6be0  /var/ossec/etc/shared/agent.conf
```

```
$ /var/ossec/bin/agent_control -i 1032

Wazuh agent_control. Agent information:
   Agent ID:   1032
   Agent Name: vpc-agent-ubuntu
   IP address: 10.0.0.122
   Status:     Active

   Operating system:    Linux vpc-agent-ubuntu.wazuh.com 3.13.0-57-generic #95-Ubuntu SMP Fri Jun 19 09:28:15
UTC 2015 x86_64
   Client version:      OSSEC Wazuh v1.2 / 078b0711a8b2ee8b18e839afdafe6be0
   Last keep alive:     Wed Feb 15 15:35:15 2017

   Syscheck last started  at: Wed Feb 15 13:24:32 2017
   Rootcheck last started at: Wed Feb 15 13:37:11 2017
```

Also, the API returns the md5sum of `agent.conf` in the field `sharedSum`:

```
$ curl -u foo:bar -k http://127.0.0.1:55000/agents/1032?pretty

{
   "error": 0,
   "data": {
      "status": "Active",
      "name": "vpc-agent-ubuntu",
      "ip": "10.0.0.122",
      "dateAdd": "2016-12-22 11:59:08",
      "version": "OSSEC Wazuh v1.2",
      "sharedSum": "078b0711a8b2ee8b18e839afdafe6be0",
      "lastKeepAlive": "2017-02-15 15:44:57",
      "os": "Linux vpc-agent-ubuntu.wazuh.com 3.13.0-57-generic #95-Ubuntu SMP Fri Jun 19 09:28:15 UTC 2015
x86_64",
      "id": "1032"
   }
}
```

5. Restart the agent

In order to apply the changes, you must restart the agent which may be completed remotely:

```
$ /var/ossec/bin/agent_control -R -u 1032

Wazuh agent_control: Restarting agent: 1032
```

Precedence

It's important to know which is the precedence between `ossec.conf` and `agent.conf`. The local and the shared configuration are merged. `ossec.conf` is read before the shared `agent.conf`, the last definition of any setting will overwrite any previous appearance. Also, the settings that includes a path to file, will be concatenated.

For example:

Let's say we have this configuration on the `ossec.conf` file.

```
<rootcheck>
  <disabled>no</disabled>
  <check_unixaudit>no</check_unixaudit>
  <check_files>yes</check_files>
  <check_trojans>no</check_trojans>
  <check_dev>yes</check_dev>
  <check_sys>yes</check_sys>
  <check_pids>yes</check_pids>
  <check_ports>yes</check_ports>
  <check_if>yes</check_if>
  <system_audit>/var/ossec/etc/shared/system_audit_rcl.txt</system_audit>
</rootcheck>
```

and the `agent.conf`.

```
<rootcheck>
  <check_unixaudit>yes</check_unixaudit>
  <rootkit_files>/var/ossec/etc/shared/rootkit_files.txt</rootkit_files>
  <rootkit_trojans>/var/ossec/etc/shared/rootkit_trojans.txt</rootkit_trojans>
  <system_audit>/var/ossec/etc/shared/cis_debian_linux_rcl.txt</system_audit>
  <system_audit>/var/ossec/etc/shared/cis_rhel_linux_rcl.txt</system_audit>
  <system_audit>/var/ossec/etc/shared/cis_rhel5_linux_rcl.txt</system_audit>
</rootcheck>
```

The final configuration will overwrite `check_unixaudit` to “yes” because it appears on the `agent.conf`. The path listed with `system_audit` option will be concatenated, so `system_audit_rcl.txt` (on the `ossec.conf`) will be as valid as `cis_debian_linux_rcl.txt` (on the `agent.conf`).