

# OSSEC Rules

## Global ossec.conf Settings

OSSEC comes with a server-wide configuration file. Its important to look for and modify this file on the host that runs the server your agents connect to. This configuration will control the alerting and rules used on the server and its agents. Located at `/var/ossec/etc/ossec.conf`, see the below examples for some changes that can be made within the file -

### Email

Change the below appropriately to send emails to yourself. Note that routing mail to localhost IP (Running Postfix IP) allows for forwarding mail through the host as the user defined in our [Postfix Configuration](#)

```
<global>
  <email_notification>yes</email_notification>
  <email_to>emailaddress@gmail.com</email_to>
  <smtp_server>127.0.0.1</smtp_server>
  <email_from>ossec@hostname</email_from>
</global>
```

### Syscheck

Syscheck is the block that configures the many settings that run various checks on your system. Define these specifically to match your environment, ignoring or adding directories as needed based on alerting. Any rules with the XML tags `<group>syscheck</group>` within their definition are checked when this test is ran.

```
<syscheck>
  <!-- Frequency that syscheck is executed - default to every 22 hours -->
  <frequency>79200</frequency>

  <!-- Directories to check (perform all possible verifications) -->
  <directories check_all="yes">/etc,/usr/bin,/usr/sbin</directories>
  <directories check_all="yes">/bin,/sbin,/boot</directories>
```

```
<!-- Files/directories to ignore -->
<ignore>/etc/mntab</ignore>
<ignore>/etc/mnttab</ignore>
<ignore>/etc/hosts.deny</ignore>
<ignore>/etc/mail/statistics</ignore>
<ignore>/etc/random-seed</ignore>
<ignore>/etc/adjtime</ignore>
<ignore>/etc/httpd/logs</ignore>
<ignore>/etc/utmpx</ignore>
<ignore>/etc/wtmpx</ignore>
<ignore>/etc/cups/certs</ignore>
<ignore>/etc/dumpdates</ignore>
<ignore>/etc/svc/volatile</ignore>

<!-- Windows files to ignore -->
<ignore>C:\WINDOWS\System32\LogFiles</ignore>
<ignore>C:\WINDOWS\Debug</ignore>
<ignore>C:\WINDOWS\WindowsUpdate.log</ignore>
<ignore>C:\WINDOWS\iis6.log</ignore>
<ignore>C:\WINDOWS\system32\wbem\Logs</ignore>
<ignore>C:\WINDOWS\system32\wbem\Repository</ignore>
<ignore>C:\WINDOWS\Prefetch</ignore>
<ignore>C:\WINDOWS\PCHEALTH\HELPCTR\DataColl</ignore>
<ignore>C:\WINDOWS\SoftwareDistribution</ignore>
<ignore>C:\WINDOWS\Temp</ignore>
<ignore>C:\WINDOWS\system32\config</ignore>
<ignore>C:\WINDOWS\system32\spool</ignore>
<ignore>C:\WINDOWS\system32\CatRoot</ignore>
</syscheck>
```

## Command Monitoring

Partition alerts -

```
<rule id="530" level="0">
  <if_sid>500</if_sid>
  <match>^ossec: output: </match>
  <description>OSSEC process monitoring rules.</description>
  <group>process_monitor,</group>
</rule>
```

```

<rule id="531" level="7" ignore="7200">
  <if_sid>530</if_sid>
  <match>ossec: output: 'df -P': /dev/</match>
  <regex>100%</regex>
  <description>Partition usage reached 100% (disk space monitor).</description>
  <group>low_diskspace,</group>
</rule>

```

## Process Monitoring Command Monitoring - Book

## Custom Monitoring

```

<!-- local_rules.xml -->
<rule id="100004" level="7">
  <if_sid>531</if_sid>
  <match>snap</match>
  <description>Ignore snap partition size alerts</description>
</rule>

```

```

<!-- ossec.conf -->
<localfile>
  <log_format>full_command</log_format>
  <command>ps</command>
  <frequency>60</frequency>
</localfile>

```

## Custom Monitoring

## Reporting

You can see a simple agent status report using the below command

```

# List agents and status
sudo /var/ossec/bin/agent_control -r -l

```

`agent_control` has various other uses, see the output of `agent_control -h` below -

OSSEC HIDS agent\_control: Control remote agents.

Available options:

-h      This help message.

- l List available (active or not) agents.
- lc List active agents.
- i <id> Extracts information from an agent.
- R <id> Restarts agent.
- r -a Runs the integrity/rootkit checking on all agents now.
- r Runs the integrity/rootkit checking on one agent now.
  
- b <ip> Blocks the specified ip address.
- f <ar> Used with -b, specifies which response to run.
- L List available active responses.
- m Show the limit of agents that can be added.
- s Changes the output to CSV (comma delimited).
- j Changes the output to JSON .
- u <id> Used with -r and -b Specifies the agent to use.

The below rules are used to generate reports on changes made to a system using OSSEC's built in grouping policies.

```
<rule id="550" level="7">
  <category>ossec</category>
  <decoded_as>syscheck_integrity_changed</decoded_as>
  <description>Integrity checksum changed.</description>
  <group>syscheck,</group>
</rule>

<rule id="551" level="7">
  <category>ossec</category>
  <decoded_as>syscheck_integrity_changed_2nd</decoded_as>
  <description>Integrity checksum changed again (2nd time).</description>
  <group>syscheck,</group>
</rule>

<rule id="552" level="7">
  <category>ossec</category>
  <decoded_as>syscheck_integrity_changed_3rd</decoded_as>
  <description>Integrity checksum changed again (3rd time).</description>
  <group>syscheck,</group>
</rule>
```

The above rules and groups can be passed through reporting tools provided by OSSEC. Note that `001` below is our agent ID we want to generate the report for. OSSEC will automatically lookup alerts that have fired related to the rules above.

```
# Manually run root, syschecks
# This checks the OSSEC server and its agents for any new changes
sudo /var/ossec/bin/agent_control -r -u 001
# Generate integrity check report (for email reporting?)
sudo /var/ossec/bin/syscheck_control -i 001
```

For example, an integrity check report looks something like the below -

```
Changes for 2019 Sep 06:
2019 Sep 06 09:18:04,0 - /usr/bin/docker
2019 Sep 06 09:18:13,0 - /usr/bin/docker-proxy
2019 Sep 06 09:18:35,0 - /usr/bin/dockerd
2019 Sep 06 09:21:36,0 - /bin/docker
2019 Sep 06 09:21:44,0 - /bin/docker-proxy
2019 Sep 06 09:22:06,0 - /bin/dockerd
2019 Sep 06 18:44:42,0 - /var/ossec/etc/internal_options.conf
2019 Sep 06 18:46:01,0 - /etc/ossec-init.conf
```

any of the above commands can be set to run as cronjobs and paired with `mail -s` to manually schedule a daily report with more specific output. Below, we sent the integrity check report we manually generated above to a specific E-Mail address.

```
sudo /var/ossec/bin/syscheck_control -i 001 | mail -s "OSSEC Daily Report: Agent 001 Integrity Check"
email@somedomain.com
sudo /var/ossec/bin/agent_control -r -l | mail -s "OSSEC Daily Report: Agent Status" email@somedomain.com
```

In this way, it is easy to setup a cronjob to send these reports for you on a specified schedule.

Setting up automatic daily reporting (12:01AM) for OSSEC can also be done within the `/var/ossec/etc/ossec.conf` file on the OSSEC Server by adding the options below -

```
<ossec_config>
...
<reports>
  <category>syscheck</category>
  <title>Daily report: File changes</title>
  <email_to>example@test.com</email_to>
```

```

    <level>10</level>

    <title>Daily report: Alerts with level higher than 10</title>

    <email_to>example@test.com</email_to>
</reports>

...
</ossec_config>

```

This will generate reports for any alerting done within the `syscheck` group, and another report for any alerts of severity `level 10` or greater. These reports are sent to the email addresses based on the settings provided to reports

Alternatively, you can manually generate these reports within the command prompt of your OSSEC Server by running the below commands -

```

# Example 1: Show Successful Logins
cat /var/ossec/logs/alerts/alerts.log | /var/ossec/bin/ossec-reportd -f group authentication_success

# Example 2: Show Alerts Level 10 and Greater
cat /var/ossec/logs/alerts/alerts.log | /var/ossec/bin/ossec-reportd -f level 10

# Example 3: Show the srcip for all users
cat /var/ossec/logs/alerts/alerts.log | /var/ossec/bin/ossec-reportd -f group authentication -r user srcip

# Example 4: Show Changed files as reported by Syscheck
cat /var/ossec/logs/alerts/alerts.log | /var/ossec/bin/ossec-reportd -f group syscheck -r location filename

```

Further filtering of alerts can be handled through properties of the XML tags, an example of filtering alerts for recipients based on various settings is seen below -

```

<ossec_config>
...
<email_alerts>
    <email_to>alice@test.com</email_to>
    <event_location>server1|server2</event_location>
</email_alerts>
<email_alerts>
    <email_to>is@test.com</email_to>
    <event_location>/log/secure$</event_location>
</email_alerts>
<email_alerts>
    <email_to>bob@test.com</email_to>
    <event_location>192.168.</event_location>
</email_alerts>
<email_alerts>

```

```
<email_to>david@test.com</email_to>
<level>12</level>
</email_alerts>
...
</ossec_config>
```

Note that the above is an example providing different uses in filtering emails in general, but the syntax for reports should be used if the desire is to filter the daily generated report distribution based on various settings. While the properties are the same, the tags are named differently within their blocks - depending on where they are nested, within an `<email_alerts>` or `<reports>` XML block.

## Custom Local Rules

When running an OSSEC with remote agents, you'll need to configure the alerting and rules specific to the needs of your environment. To do this, edit `/var/ossec/rules/local_rules.xml` and blocks similar to the format below.

```
<!-- This example will ignore NXDOMAIN alerts -->
<rule id="100002" level="0"> <!--Define the rule ID we are creating-->
  <if_sid>1002</if_sid> <!-- Specify rule ID we are altering -->
  <program_name>systemd-resolved</program_name> <!-- Optional cross check with rule program name -->
  <match>Server returned error NXDOMAIN</match> <!-- Match error text -->
  <description>Usless systemd-resolvd log message</description> <!-- local description -->
</rule>

<!-- This example will ignore ssh failed logins for the user name XYZABC. --
<rule id="100020" level="0">
  <if_sid>5711</if_sid>
  <user>XYZABC</user>
  <description>Example of rule that will ignore sshd </description>
  <description>failed logins for user XYZABC.</description>
</rule>
```

Above, we use the `<rule>` xml tag with various values passed to OSSEC to identify our rule. Below, we can see all the available attributes for this tag

rule -  
Defines a rule

Attributes:

## level

Specifies the level of the rule. Alerts and responses use this value.

Allowed: Any number (0 to 16)

## id

Specifies the ID of the rule.

Allowed: Any number from 100 to 99999

## maxsize

Specifies the maximum size of the event.

Allowed: Any number from 1 to 99999

## frequency

Specifies the number of times the rule must have matched before firing. The number that triggers the rule is actually 2 more than this setting.

Allowed: Any number from 1 to 999

Example: frequency="2" would mean the rule must be matched 4 times

## timeframe

The timeframe in seconds.

This option is intended to be used with the frequency option.

Allowed: Any number from 1 to 9999

## ignore

The time (in seconds) to ignore this rule after firing it (to avoid floods).

Allowed: Any number from 1 to 9999

## overwrite

Used to supercede an OSSEC rule with local changes.

This is useful to change the level or other options of rules included with OSSEC.

Allowed yes

You'll notice that the [OSSEC Docs - Rule Syntax](#) will be a great resource when creating these rules as there are many different values you can add or edit, but a few of the important settings can be



seen below -

### **<program\_name>**

Program name is decoded from syslog process name.

Allowed: any OS\_Match/sregex Syntax

### **<if\_sid>**

Matches if the ID has matched.

Allowed: Any rule id

### **<if\_level>**

Matches if the level has matched before.

Allowed: Any level from 1 to 16

### **<match>**

Any string to match against the log event.

Allowed: Any OS\_Match/sregex Syntax

### **<regex>**

Any regex to match against the log event.

Allowed: Any OR\_Regex/regex Syntax

Complete list of ossec rules within `/var/ossec/rules/` -

```
apache_rules.xml  ms1016_usbdetect_rules.xml  sendmail_rules.xml
apache_rules.xml~  ms_dhcp_rules.xml  sendmail_rules.xml~
apparmor_rules.xml  ms_dhcp_rules.xml~  smbd_rules.xml
apparmor_rules.xml~  ms_firewall_rules.xml  smbd_rules.xml~
arpwatch_rules.xml  ms_ftpd_rules.xml  solaris_bsm_rules.xml
arpwatch_rules.xml~  ms_ftpd_rules.xml~  solaris_bsm_rules.xml~
asterisk_rules.xml  ms_ipsec_rules.xml  sonicwall_rules.xml
asterisk_rules.xml~  ms_powershell_rules.xml  sonicwall_rules.xml~
attack_rules.xml  msauth_rules.xml  spamd_rules.xml
attack_rules.xml~  msauth_rules.xml~  spamd_rules.xml~
cimserver_rules.xml  mysql_rules.xml  squid_rules.xml
cimserver_rules.xml~  mysql_rules.xml~  squid_rules.xml~
cisco-ios_rules.xml  named_rules.xml  sshd_rules.xml
cisco-ios_rules.xml~  named_rules.xml~  sshd_rules.xml~
clam_av_rules.xml  netscreenfw_rules.xml  symantec-av_rules.xml
clam_av_rules.xml~  netscreenfw_rules.xml~  symantec-av_rules.xml~
courier_rules.xml  nginx_rules.xml  symantec-ws_rules.xml
courier_rules.xml~  nginx_rules.xml~  symantec-ws_rules.xml~
dnsmasq_rules.xml  nsd_rules.xml  syslog_rules.xml
```

```

dovecot_rules.xml[] openbsd-dhcpd_rules.xml    syslog_rules.xml~
dovecot_rules.xml~[] openbsd_rules.xml[]    sysmon_rules.xml
dropbear_rules.xml[] openbsd_rules.xml~[]    sysmon_rules.xml~
dropbear_rules.xml~[] opensmtpd_rules.xml[]    systemd_rules.xml
exim_rules.xml[] opensmtpd_rules.xml~    systemd_rules.xml~
firewall_rules.xml[] ossec_rules.xml[]    telnetd_rules.xml
firewall_rules.xml~[] ossec_rules.xml~[]    telnetd_rules.xml~
firewalld_rules.xml[] owncloud_rules.xml[]    topleveldomain_rules.xml
firewalld_rules.xml~[] pam_rules.xml[]    trend-osce_rules.xml
ftpd_rules.xml[] pam_rules.xml~[]    trend-osce_rules.xml~
ftpd_rules.xml~[] php_rules.xml[]    unbound_rules.xml
hordeimp_rules.xml[] php_rules.xml~[]    unbound_rules.xml~
hordeimp_rules.xml~[] pix_rules.xml[]    vmpop3d_rules.xml
ids_rules.xml[] pix_rules.xml~[]    vmpop3d_rules.xml~
ids_rules.xml~[] policy_rules.xml[]    vmware_rules.xml
imapd_rules.xml[] policy_rules.xml~[]    vmware_rules.xml~
imapd_rules.xml~[] postfix_rules.xml[]    vpn_concentrator_rules.xml
kesl_rules.xml[] postfix_rules.xml~[]    vpn_concentrator_rules.xml~
last_rootlogin_rules.xml postgresql_rules.xml    vpopmail_rules.xml
linux_usbdetect_rules.xml postgresql_rules.xml~    vpopmail_rules.xml~
local_rules.xml[] proftpd_rules.xml[]    vsftpd_rules.xml
local_rules.xml~[] proftpd_rules.xml~[]    vsftpd_rules.xml~
mailscanner_rules.xml[] proxmox-ve_rules.xml    web_appsec_rules.xml
mailscanner_rules.xml~[] psad_rules.xml[]    web_appsec_rules.xml~
mcafee_av_rules.xml[] pure-ftpd_rules.xml[]    web_rules.xml
mcafee_av_rules.xml~[] pure-ftpd_rules.xml~    web_rules.xml~
mhn_cowrie_rules.xml[] racoon_rules.xml[]    wordpress_rules.xml
mhn_dionaea_rules.xml[] racoon_rules.xml~[]    wordpress_rules.xml~
ms-exchange_rules.xml[] roundcube_rules.xml[]    zeus_rules.xml
ms-exchange_rules.xml~[] roundcube_rules.xml~    zeus_rules.xml~
ms-se_rules.xml[] rules_config.xml
ms-se_rules.xml~[] rules_config.xml~

```

## Debugging OSSEC Rules

If you're having trouble configuring / testing rules, try running the commands below to test your settings on command by causing the alerts to fire.

```

# Watch for new OSSEC logs within /var/ossec/logs/
tail -f /var/ossec/logs/active-response.log

```

```
tail -f /var/ossec/logs/ossec.log
```

```
## Test the OSSEC response as if the echo contents were parsed within some monitored logs
```

```
logger "Segmentation Fault"
```

```
# logger will have no output but will respond as if the rule really happened
```

```
# Will give debug info, does not actually carry out any actions
```

```
# Shows action that the log represents on the system, as well as OSSEC response if the alert would fire
```

```
echo "pam_unix(sudo:auth): conversation failed" | sudo /var/ossec/bin/ossec-logtest
```

```
2019/09/07 04:50:59 ossec-testrule: INFO: Reading local decoder file.
```

```
2019/09/07 04:50:59 ossec-testrule: INFO: Started (pid: 26407).
```

```
ossec-testrule: Type one log per line.
```

```
**Phase 1: Completed pre-decoding.
```

```
  full event: 'pam_unix(sudo:auth): conversation failed'
```

```
  hostname: 'monitor'
```

```
  program_name: '(null)'
```

```
  log: 'pam_unix(sudo:auth): conversation failed'
```

```
**Phase 2: Completed decoding.
```

```
  No decoder matched.
```

```
**Phase 3: Completed filtering (rules).
```

```
  Rule id: '1002'
```

```
  Level: '2'
```

```
  Description: 'Unknown problem somewhere in the system.'
```

```
**Alert to be generated.
```

```
# Re-scan a log that fired alerts to check what actions might happen if they reoccur
```

```
# This will scan syslogs for all events within the log
```

```
cat /var/log/syslog | /var/ossec/bin/ossec-logtest -a
```

```
# We can build on the above command and scan the ossec alert.log
```

```
# Passing this into reportd, we can generate a report on the alerts fired within the given log
```

```
cat /var/ossec/logs/alerts/alerts.log | /var/ossec/bin/ossec-logtest -a | /var/ossec/bin/ossec-reportd
```

```
2019/09/07 05:20:57 ossec-reportd: INFO: Started (pid: 26840).
```

```
2019/09/07 05:20:57 ossec-testrule: INFO: Reading local decoder file.
```

```
2019/09/07 05:20:57 ossec-testrule: INFO: Started (pid: 26839).
```

```
Report completed. ==
```

-----  
->Processed alerts: 249  
->Post-filtering alerts: 249  
->First alert: 2019 Sep 07 05:20:57  
->Last alert: 2019 Sep 07 05:20:57

Top entries for 'Source ip':

-----  
100.200.300.400 |11 |  
200.300.400.100 |2 |  
300.400.100.200 |2 |  
::ffff:400.100.200.300 |1 |  
::ffff:500.600.700.800 |1 |  
::ffff:600.700.800.500 |1 |  
::ffff:700.800.500.600 |1 |

Top entries for 'Username':

-----  
user1 |60 |  
user2 |6 |  
root |6 |

Top entries for 'Level':

-----  
Severity 3 |230 |  
Severity 2 |5 |  
Severity 4 |5 |  
Severity 7 |5 |  
Severity 13 |2 |  
Severity 5 |2 |

Top entries for 'Group':

-----  
syslog |244 |  
pam |153 |  
authentication\_success |86 |  
sudo |57 |  
sshd |13 |  
dpkg |8 |

errors	7	
config_changed	5	
connection_attempt	4	
vsftpd	4	
ossec	3	
apache	2	
fts	2	

Top entries for 'Location':

host3->stdin	115	
host2->stdin	82	
host->stdin	50	
(host)->stdin	2	

Top entries for 'Rule':

5502 - Login session closed.	82	
5501 - Login session opened.	71	
5402 - Successful sudo to ROOT executed	54	
5715 - SSHD authentication success.	13	
1002 - Unknown problem somewhere in the system.	5	
2902 - New dpkg (Debian Package) installed.	5	
11401 - FTP session opened.	4	
2901 - New dpkg (Debian Package) requested to install.	3	
5403 - First time user executed sudo.	3	
591 - Log file rotated.	3	
1003 - Non standard syslog message (size too large).	2	
10100 - First time user logged in.	2	
31303 - Nginx critical message.	2	

# Run a report limited to alerts level 7 and above within the passed log file

# Filters allowed: group, rule, level, location,

# user, srcip, filename

# Examples:

# -f group authentication\_success (to filter on login success)

# -f level 10 (to filter on level >= 10)

# -f group authentication -r user srcip (to show srcip for all users)

cat /var/ossec/logs/alerts/alerts.log | /var/ossec/bin/ossec-logtest -a | /var/ossec/bin/ossec-reportd -f level 7

2019/09/07 05:26:49 ossec-reportd: INFO: Started (pid: 26871).

2019/09/07 05:26:49 ossec-testrule: INFO: Reading local decoder file.

2019/09/07 05:26:49 ossec-testrule: INFO: Started (pid: 26870).

2019/09/07 05:26:54 ossec-reportd: INFO: Report completed. Creating output...

Report completed. ==

-----  
->Processed alerts: 249

->Post-filtering alerts: 7

->First alert: 2019 Sep 07 05:26:49

->Last alert: 2019 Sep 07 05:26:49

Top entries for 'Level':

-----  
Severity 7 |5 |  
Severity 13 |2 |

Top entries for 'Group':

-----  
syslog |7 |  
config\_changed |5 |  
dpkg |5 |  
errors |2 |

Top entries for 'Location':

-----  
monitor->stdin |7 |

Top entries for 'Rule':

-----  
2902 - New dpkg (Debian Package) installed. |5 |  
1003 - Non standard syslog message (size too large). |2 |

---

Revision #8

Created 6 September 2019 23:41:53 by Shaun Reed

Updated 20 May 2020 07:06:31 by Shaun Reed