# Monitoring

- Fail2Ban
- OSSEC Ubuntu Server
- OSSEC Rules

# Fail2Ban

## Links & Installation

Fail2ban Documentation

Linode Fail2ban Guide

```
sudo yum install fail2ban
sudo apt install fail2ban
sudo pacman -Syu fail2ban
```

etc..

## Configuration Files

`/etc/fail2ban/`

To modify configs, copy any `fail2ban.conf` to `fail2ban.local` and modify the copied `fail2ban.local` configuration file. Fail2ban will automatically override the settings in `fail2ban.conf` with those in `fail2ban.local`

```
sudo cp /etc/fail2ban/fail2ban.conf /etc/fail2ban/fail2ban.local
```

This file is also not intended to be modified directly. Run the command below to create a local configuration to edit -

```
sudo cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local
```

### jail.local

This file holds all of your local fail2ban settings for this host. Some of the important lines and settings to look at are seen below

- ignoreip
  - An IP address for fail2ban to ignore
- maxrety
  - Number of retries before being locked out

```
#... File Reduced ...
```

```
# Destination email address used solely for the interpolations in
# jail.{conf,local,d/*} configuration files.
destemail = user@gmail.com

# Sender email address used solely for some actions
sender = admin@hostname

# E-mail action. Since 0.8.1 Fail2Ban uses sendmail MTA for the
# mailing. Change mta configuration parameter to mail if you want to
# revert to conventional 'mail'.
mta = mail

# Default protocol
protocol = tcp

#... File Reduced ...

# Choose default action.  To change, just override value of 'action' with the
# interpolation to the chosen action shortcut (e.g.  action_mw, action_mwl, etc) in jail.local
# globally (section [DEFAULT]) or per specific section
action = %(action_mwl)s

#
# JAILS
#

#
# SSH servers
#

[sshd]
# To use more aggressive sshd modes set filter parameter "mode" in jail.local:
# normal (default), ddos, extra or aggressive (combines all).
# See "tests/files/logs/sshd" or "filter.d/sshd.conf" for usage example and[Definition]

failregex = ^<HOST> -.*GET.*(\.php|\.asp|\.exe|\.pl|\.cgi|\.scgi)

ignoreregex = details.
#mode   = normal
enabled = true
```

```
port    = 22
logpath = %(sshd_log)s
backend = %(sshd_backend)s

#... File Reduced ...

[nginx-http-auth]

enabled = true
port    = http,https
logpath = %(nginx_error_log)s

# To use 'nginx-limit-req' jail you should have `ngx_http_limit_req_module`
# and define `limit_req` and `limit_req_zone` as described in nginx documentation
# http://nginx.org/en/docs/http/ngx_http_limit_req_module.html
# or for example see in 'config/filter.d/nginx-limit-req.conf'
[nginx-limit-req]
port    = http,https
logpath = %(nginx_error_log)s

[nginx-botsearch]

enabled = true
port     = http,https
logpath  = %(nginx_error_log)s
maxretry = 2

#... File Reduced ...
```

its important to notice the line `action = %(action_mwl)s` - this line defines which default action we will take when a fail2ban is broken. These actions are defined within our `jail.local`, but Ill paste them here as well

```
# Action shortcuts. To be used to define action parameter

# Default banning action (e.g. iptables, iptables-new,
# iptables-multiport, shorewall, etc) It is used to define
# action_* variables. Can be overridden globally or per
# section within jail.local file
banaction = iptables-multiport
```

```
banaction_allports = iptables-allports

# The simplest action to take: ban only
action_ = %(banaction)s[name=%(__name__)s, bantime="%(bantime)s", port="%(port)s",
protocol="%(protocol)s", chain="%(chain)s"]

# ban & send an e-mail with whois report to the destemail.
action_mw = %(banaction)s[name=%(__name__)s, bantime="%(bantime)s", port="%(port)s",
protocol="%(protocol)s", chain="%(chain)s"]
            %(mta)s-whois[name=%(__name__)s, sender="%(sender)s", dest="%(destemail)s",
protocol="%(protocol)s", chain="%(chain)s"]

# ban & send an e-mail with whois report and relevant log lines
# to the destemail.
action_mwl = %(banaction)s[name=%(__name__)s, bantime="%(bantime)s", port="%(port)s",
protocol="%(protocol)s", chain="%(chain)s"]
             %(mta)s-whois-lines[name=%(__name__)s, sender="%(sender)s", dest="%(destemail)s",
logpath=%(logpath)s, chain="%(chain)s"]

# See the IMPORTANT note in action.d/xarf-login-attack for when to use this action
#
# ban & send a xarf e-mail to abuse contact of IP address and include relevant log lines
# to the destemail.
action_xarf = %(banaction)s[name=%(__name__)s, bantime="%(bantime)s", port="%(port)s",
protocol="%(protocol)s", chain="%(chain)s"]
             xarf-login-attack[service=%(__name__)s, sender="%(sender)s", logpath=%(logpath)s, port="%(port)s"]

# ban IP on CloudFlare & send an e-mail with whois report and relevant log lines
# to the destemail.
action_cf_mwl = cloudflare[cfuser="%(cfemail)s", cftoken="%(cfapikey)s"]
                %(mta)s-whois-lines[name=%(__name__)s, sender="%(sender)s", dest="%(destemail)s",
logpath=%(logpath)s, chain="%(chain)s"]

# Report block via blocklist.de fail2ban reporting service API
#
# See the IMPORTANT note in action.d/blocklist_de.conf for when to use this action.
# Specify expected parameters in file action.d/blocklist_de.local or if the interpolation
# `action_blocklist_de` used for the action, set value of `blocklist_de_apikey`
# in your `jail.local` globally (section [DEFAULT]) or per specific jail section (resp. in
# corresponding jail.d/my-jail.local file).
```

```
#
action_blocklist_de  = blocklist_de[email="%(sender)s", service=%(filter)s, apikey="%(blocklist_de_apikey)s",
agent="%(fail2ban_agent)s"]

# Report ban via badips.com, and use as blacklist
#
# See BadIPsAction docstring in config/action.d/badips.py for
# documentation for this action.
#
# NOTE: This action relies on banaction being present on start and therefore
# should be last action defined for a jail.
#
action_badips = badips.py[category="%(__name__)s", banaction="%(banaction)s", agent="%(fail2ban_agent)s"]
#
# Report ban via badips.com (uses action.d/badips.conf for reporting only)
#
action_badips_report = badips[category="%(__name__)s", agent="%(fail2ban_agent)s"]

# Report ban via abuseipdb.com.
#
# See action.d/abuseipdb.conf for usage example and details.
#
action_abuseipdb = abuseipdb
```

You can use any action above that you'd like, and even create your own or modify them as you see fit.

# Custom Jails

Using Regex and fail2ban's filters, we can create our own jails within fail2ban to define rules specific to requests we may be recieving on our application. For example, add the below to `/etc/fail2ban/jail.local` to define a rule to block attempts to run scripts on the webserver.

```
# Add these lines to /etc/fail2ban/jail.local
[nginx-noscript]

enabled  = true
port     = http,https
filter   = nginx-noscript
logpath  = /var/log/nginx/access.log
maxretry = 6
```

Now, we need to define the filter for the jail we just created. Fail2ban stores these within `/etc/fail2ban/filter.d/`. So, for our example we will create the `nginx-noscript.conf` file within this directory. Its important that the name we choose here corresponds with what we named our jail in `jail.local`.

```
# New definition for /etc/fail2ban/filter.d/nginx-noscript.conf jail
[Definition]

failregex = ^<HOST> -.*GET.*(\.php|\.asp|\.exe|\.pl|\.cgi|\.scgi)

ignoreregex =
```

If you want to test this regex, you can use `fail2ban-regex` to do so on any log file. The command below is an example of testing a regex statement on an nginx log. This command will output all the matching lines within the log that are captured by the regex, which would result in a ban from fail2ban -

```
sudo fail2ban-regex /var/log/nginx/access.log '^<HOST>.*\"(.|)\\x.*\"$' --print-all-matched
```

## Jail Status

To check on the status of running jails, see the command below

```
sudo fail2ban-client status
```

When users are banned under a jail, you can see a list of them by running the following, where `nginx-http-auth` can be changed out for any name of a running jail.

```
sudo fail2ban-client status nginx-http-auth
```

To unban an IP from a jail, run the below

```
sudo fail2ban-client set nginx-http-auth unbanip 124.45.123.777.
```

To unban all IPs from a given jail

```
sudo fail2ban-client restart --unban nginx-http-auth
```

# Log Files

Fail2ban's logs will look similar to the below -

```
sudo cat /var/log/fail2ban.log | tail
2019-11-24 17:16:24,307 fail2ban.filter         [27297]: INFO    [nginx-noscript] Found 62.234.108.37 - 2019-11-24 17:16:24
2019-11-24 17:16:25,009 fail2ban.filter         [27297]: INFO    [nginx-noscript] Found 62.234.108.37 - 2019-11-24 17:16:24
2019-11-24 17:16:25,425 fail2ban.filter         [27297]: INFO    [nginx-noscript] Found 62.234.108.37 - 2019-11-24 17:16:25
2019-11-24 17:16:25,481 fail2ban.actions        [27297]: NOTICE  [nginx-noscript] Ban 62.234.108.37 - 2019-11-24 17:17:25
```

Fail2ban keeps these logs within the `/var/log/fail2ban.log` file, we can use these logs with the commands below to create useful reports for hardening your server or tuning your rules. Take notice of the difference in the use of `zgrep` and `grep` below, where we are either searching recent logs or all the logs stored on the system.

```
# Report on all logs for summary of bans triggered sorted by jail, grouped by dates -
sudo zgrep -h "Ban " /var/log/fail2ban.log* | awk '{print $6,$1}' | sort | uniq -c
3 [nginx-noscript] 2019-10-27
4 [nginx-noscript] 2019-10-28
4 [nginx-noscript] 2019-10-29
6 [nginx-jail2] 2019-10-27
1 [nginx-jail2] 2019-10-28
2 [nginx-jail2] 2019-10-29
```

```
# Log report for bans triggered today only, grouped by IP and hostname -
grep "Ban " /var/log/fail2ban.log | grep `date +%Y-%m-%d` | awk '{print $NF}' | sort | awk '{print $1,"("$1")"}' | logresolve | uniq -c | sort -n
      1 217.147.85.78 (217.147.85.78)
      1 61-219-11-153.HINET-IP.hinet.net (61.219.11.153)
      1 85.93.20.70 (85.93.20.70)
      1 ip50.ip-51-83-234.eu (51.83.234.50)
      1 vmi185089.contaboserver.net (5.189.189.207)
      1 vmi214529.contaboserver.net (213.136.87.57)
      2 62.234.108.37 (62.234.108.37)
```

```
# Log report for bans triggered today only, grouped by IP and jail -
sudo grep "Ban " /var/log/fail2ban.log | awk '{print $6,$8}' | sort | uniq -c | sort -n
      1 [nginx-jail1] 61.219.11.153
      1 [nginx-jail1] 85.93.20.70
      1 [nginx-jail2] 138.68.247.104
```

```
      1 [nginx-jail2] 213.136.87.57
      1 [nginx-jail2] 217.147.85.78
      1 [nginx-jail2] 5.189.189.207
      1 [nginx-noscript] 51.83.234.50
      2 [nginx-noscript] 62.234.108.37
      5 [nginx-noscript] 51.83.234.50
      4 [nginx-noscript] 62.234.108.37
```

```
# Log report for all bans triggered within a logfile, sorted by date, grouped by jail
sudo grep "Ban " /var/log/fail2ban.log.1 | awk '{print $1, $6}'|sort | uniq -c
      1 2020-02-17 [nginx-noscript]
      3 2020-02-18 [nginx-noscript]
      1 2020-02-18 [nginx-wplogin]
    158 2020-02-19 [nginx-noscan]
      2 2020-02-19 [nginx-noscript]
     15 2020-02-19 [sshd-badproto]
# Since nginx-noscan is a permanent ban, the high number above is the jail restoring bans after a manual reboot on the 19th
```

```
# Report on all logs for summary of bans triggered, grouped by IP and jail -
sudo awk '($(NF-1) = /Ban/){print $NF,"("$NF")"}' /var/log/fail2ban.log* | sort | logresolve | uniq -c | sort -n
      1 mail.grayson-college.info (162.253.219.14)
      1 new.wigroup.com.br (159.89.144.7)
      1 srvcpanel02.ativy.com (201.7.210.50)
      1 vps-01.naftalie.net (142.44.240.254)
      2 106.12.54.100 (106.12.54.100)
      2 106.13.228.250 (106.13.228.250)
      2 111.20.55.66 (111.20.55.66)
```

```
# Log report for bans triggered today only, grouped and sorted by IP -
sudo awk '($(NF-1) = /Ban/){print $NF}' /var/log/fail2ban.log | sort | uniq -c | sort -n
      1 131.108.164.19
      1 138.68.247.104
      1 213.136.87.57
      1 217.147.85.78
      1 5.189.189.207
      1 51.83.234.50
      1 61.219.11.153
      1 85.93.20.70
      2 62.234.108.37
```

```
# Report on all logs for summary of bans triggered, grouped and sorted by IP -
sudo awk '($(NF-1) = /Ban/){print $NF,"("$NF")"}' /var/log/fail2ban.log* | sort | logresolve | uniq -c | sort -n
     2 79.143.186.114
      3 79.143.187.243
      2 79.143.188.161
      2 80.211.6.136
      2 80.211.85.67
      2 80.241.220.101
      2 80.241.221.67
      3 80.82.70.118
      1 85.93.20.70
      2 87.98.136.163
      3 89.208.209.125
      2 89.238.186.229
      2 91.121.106.6
      2 91.121.157.178
      3 91.121.70.155
      3 91.121.76.97
      2 91.123.204.139
      3 91.194.90.159
      1 94.180.250.158
```

```
# Report on all logs for summary of bans triggered, grouped and sorted by truncated IPs -
zgrep -h "Ban " /var/log/fail2ban.log* | awk '{print $NF}' | awk -F\. '{print $1"."$2"."}' | sort | uniq -c  | sort -n |
tail
      1 101.200.
      1 103.60.
      1 103.98.
      1 106.120.
      1 106.13.
      1 106.54.
      1 107.6.
      1 114.115.
      1 114.215.
      1 122.51.
      1 123.207.
      1 129.146.
 ...
      Output reduced
      ...
```

```
  8 46.101.
 10 91.121.
 11 159.65.
 11 79.143.
 12 51.68.
 13 51.38.
 19 207.180.
 22 173.212.
 22 5.189.
 33 173.249.
```

```
# Report on all logs for summary of bans triggered, grouped and sorted by truncated IPs -
# Pipe through tail to create a smaller report of most offensive subnets
zgrep -h "Ban " /var/log/fail2ban.log* | awk '{print $NF}' | awk -F\. '{print $1"."$2"."}' | sort | uniq -c  | sort -n | tail
  8 46.101.
 10 91.121.
 11 159.65.
 11 79.143.
 12 51.68.
 13 51.38.
 19 207.180.
 22 173.212.
 22 5.189.
 33 173.249.
```

> “ Art of the web

# OSSEC Ubuntu Server

OSSEC is a useful tool in monitoring for malicious activity across various servers. It's lightweight, and easy to install an agent and have it reporting to the master server within minutes. Unfortunately, there is no automated solution to configuring agents remotely via Ansible or other tools that I am aware of.

## OSSEC Server Configuration

Its important to note that we are installing the server in these instructions, and not an agent manager. An Agent manager is a much lighter installation from the same tarball that allows connecting to this server and reporting alerts through one host.

### Creating OSSEC User

Once you are logged in to the host you wish to act as the server sending email alerts and recieving reports from agents and create a new user to manage OSSEC -

```
admin@host:~$ git clone https://github.com/shaunrd0/klips

Cloning into 'klips'...
remote: Enumerating objects: 295, done.
remote: Counting objects: 100% (295/295), done.
remote: Compressing objects: 100% (187/187), done.
remote: Total 295 (delta 109), reused 255 (delta 72), pack-reused 0
Receiving objects: 100% (295/295), 47.48 KiB | 3.96 MiB/s, done.
Resolving deltas: 100% (109/109), done.
admin@host:~$ cp klips/scripts/adduser.sh .
admin@host:~$ sudo ./adduser.sh ossec 5555
Adding user `ossec' ...
Adding new group `ossec' (5555) ...
Adding new user `ossec' (5555) with group `ossec' ...
Creating home directory `/home/ossec' ...
Copying files from `/etc/skel' ...

Enter 1 if ossec should have sudo privileges. Any other value will continue and make no changes
1
Configuring sudo for ossec...
```

```
Enter 1 to set a password for ossec, any other value will exit with no password set
1
Changing password for ossec...
New password:
Retype new password:
passwd: password updated successfully
```

## Dependencies / Installation Files

Now that we have our user created, lets become them and prepare to install the OSSEC server

```
admin@host:~$ sudo -iu ossec

To run a command as administrator (user "root"), use "sudo <command>".
see "man sudo_root" for details.
```

OSSEC Official Downloads provides the official download sources, and after selecting one the file can be downloaded right to your working directory using `wget` -

```
# Download the tar for linux server / agent installation
wget https://github.com/ossec/ossec-hids/archive/3.3.0.tar.gz
# Extract it
tar xf 3.3.0.tar.gz
```

Now, we have created the below directory -

```
ossec-hids-3.3.0/
BUGS       CONFIG       INSTALL  README.md   active-response  doc  install.sh
CHANGELOG  CONTRIBUTORS  LICENSE  SUPPORT.md  contrib         etc  src
```

We should prepare to start installing by grabbing basic OSSEC dependencies -

```
sudo apt install build-essential gcc make
```

> Below we will cover the several error cases I've encountered installing OSSEC on Ubuntu servers 18.04 and later. If you just want to get through the install, feel free to skip below and install / extract all the dependencies that fixed the many errors I've encountered during the installation process. Otherwise, only install or correct the packages which give you errors during installation

## Installing OSSEC Server

> During installation, OSSEC will ask for a server hostname / IP address. Using anything other than a direct IP has always given me issues. If you install and want to change the IP of your OSSEC server, edit the `/var/ossec/etc/agent.conf` file.

Now we have our user created, permissions granted, and dependencies / files we need to install OSSEC. Navigate within the `ossec-hids-3.3.0/` directory and run `sudo ./install.sh`. You will be prompted to select preferred settings for this installation. Pay attention to the prompts and respond accordingly, this is where the difference is seen in installing an Agent vs installing the OSSEC Monitoring Server.

You may see the below error for a missing dependency - pcre2.

```
5- Installing the system
 - Running the Makefile
cd external/pcre2-10.32/ && \
./configure \
        --prefix=/home/kossec/ossec-hids-3.3.0/src/external/pcre2-10.32//install \
        --enable-jit \
        --disable-shared \
        --enable-static && \
make install-libLTLIBRARIES install-nodist_includeHEADERS
/bin/sh: 1: cd: can't cd to external/pcre2-10.32/
Makefile:770: recipe for target 'external/pcre2-10.32//install/lib/libpcre2-8.a' failed
make: *** [external/pcre2-10.32//install/lib/libpcre2-8.a] Error 2


 Error 0x5.
 Building error. Unable to finish the installation.
```

Continue on by running the below commands, which will add the required files to your extracted `ossec-hids-3.3.0/` directory -

```
# Error - build fails because of missing pcre2
# Run these commands within the installation directory
cd ossec-hids-3.3.0
wget https://ftp.pcre.org/pub/pcre/pcre2-10.32.tar.gz
tar xzf pcre2-10.32.tar.gz -C src/external
```

Now run `sudo PCRE2_SYSTEM=no ./install` to start the installation, and keep in mind should you need to restart the install later for any reason you will need to run `sudo PCRE2_SYSTEM=no ./install` and NOT `sudo ./install`.

After fixing the above error, you may see another when attempting to install again. The error below is due to the missing `libz-dev` dependency -

```
os_zlib/os_zlib.c:13:10: fatal error: zlib.h: No such file or directory
 #include <zlib.h>
          ^~~~~~~~
compilation terminated.
Makefile:727: recipe for target 'os_zlib/os_zlib.o' failed
make: *** [os_zlib/os_zlib.o] Error 1


 Error 0x5.
 Building error. Unable to finish the installation.
```

If you see *this* error, you'll need to install zlib using the below command

```
# Error Making os_auth
sudo apt install -y libz-dev
```

Now we may see yet another error -

```
client-agent/start_agent.c:15:10: fatal error: event.h: No such file or directory
 #include <event.h>
          ^~~~~~~~~
compilation terminated.
Makefile:1014: recipe for target 'client-agent/start_agent.o' failed
make: *** [client-agent/start_agent.o] Error 1
 Error 0x5.
 Building error. Unable to finish the installation.
```

To correct this problem, we can run `sudo apt install libevent-dev` but it doesn't end there..

We might see the problem below -

```
#include <openssl/opensslv.h> /* For OPENSSL_VERSION_NUMBER */
          ^~~~~~~~~~~~~~~~~~~~
compilation terminated.
Makefile:1017: recipe for target 'ossec-agentd' failed
make: *** [ossec-agentd] Error 1
 Error 0x5.
 Building error. Unable to finish the installation
```

To fix this problem, we run `sudo apt install libssl-dev`

That's it! :)

## Enable EMail Notifications

To enable email alerting from the OSSEC server and the remote agents it monitors, follow the instructions on Configuring Postfix and then make the appropriate changes to the mail settings in `/var/ossec/etc/ossec.conf` below -

```
<ossec_config>
  <global>
    <email_notification>yes</email_notification>
    <email_to>someone@somedomain.com</email_to>
    <smtp_server>127.0.0.1</smtp_server>
    <email_from>ossec@host</email_from>
  </global>
```

There is no need to point `smtp_server` to a mail server directly, handling mail this way simply bounces the messages off the localhost's configuration we already have setup and validated working from Configuring Postfix.

If you can send mail with the below command, chances are the issue is within OSSEC and not your servers postifx configuration or gmail authentication.

```
echo "This email confirms that Postfix is working" | mail -s "Testing Posfix" emailuser@example.com
```

If this command fails, go back and check that you've configured Postfix correctly with GMail, and once you have that verified come back here to finish up with OSSEC.

# Managing Agents

> I haven't had much luck with using domain names or floating IPs with OSSEC agents or servers, so in general I'd recommend just using a direct IP address. If there is a way around this, I'm not aware of it.

To use the manage_agent utility that comes with OSSEC, run `/var/ossec/bin/manage_agents` either as root or with sudo (`sudo /var/ossec/bin/manage_agents`)

If you are running the manage_agents utility on an OSSEC Server -

```
****************************************
* OSSEC HIDS v3.3.0 Agent manager.     *
* The following options are available: *
****************************************
   (A)dd an agent (A).
   (E)xtract key for an agent (E).
   (L)ist already added agents (L).
   (R)emove an agent (R).
   (Q)uit.
Choose your action: A,E,L,R or Q: q
```

If you are running the manage_agents utility on an OSSEC Agent -

```
****************************************
* OSSEC HIDS v3.3.0 Agent manager.     *
* The following options are available: *
****************************************
   (I)mport key from the server (I).
   (Q)uit.
Choose your action: I or Q:
```

This tool is used to define an agent on the server, export its key, then import the server's key within the ossec-agent on the remote host, allowing the connection. Its important to follow these steps carefully, as any discrepency in IP or `client.keys` will result in a connection failing.

# Defining Agents

To start, on the OSSEC Server, run the manage_agents utility and add an agent -

```
****************************************
* OSSEC HIDS v3.3.0 Agent manager.     *
* The following options are available: *
****************************************
   (A)dd an agent (A).
   (E)xtract key for an agent (E).
   (L)ist already added agents (L).
   (R)emove an agent (R).
   (Q)uit.
Choose your action: A,E,L,R or Q: a


- Adding a new agent (use '\q' to return to the main menu).
```

```
  Please provide the following:
   * A name for the new agent: test
   * The IP Address of the new agent: 0.0.0.0
   * An ID for the new agent[007]: 007
Agent information:
   ID:007
   Name:test
   IP Address:0.0.0.0

Confirm adding it?(y/n): y
Agent added.
```

# Extracting Server Keys (Server)

We've define our agent with the local OSSEC Server, and prepared it for the connection. Now we need to continue through the prompts and extract the key for the agent to copy over onto our remote host -

```
****************************************
* OSSEC HIDS v3.3.0 Agent manager.     *
* The following options are available: *
****************************************
   (A)dd an agent (A).
   (E)xtract key for an agent (E).
   (L)ist already added agents (L).
   (R)emove an agent (R).
   (Q)uit.
Choose your action: A,E,L,R or Q: e

Provide the ID of the agent to extract the key (or '\q' to quit): 007

Agent key information for '007' is:
MDA2Ig4LjQzIDkxZDYmRlZGN5mOG5NzY5Nd325dmFmMTU0NzZkddaDM1ND456431MWY1ODhhMDyjukuMDYzg
4MzA5MmM=
** Press ENTER to return to the main menu.
```

Exit the prompts, and copy this key or temporarily store it in the text file. We will need it to register the agent with its remote server.

# Importing Server Keys (Agent)

Open a terminal on the host you have installed the OSSEC Agent on, and run the manage_agents utility just as we did on the last host (`sudo /var/ossec/bin/manage_agents`) -

```
****************************************
* OSSEC HIDS v3.3.0 Agent manager.     *
* The following options are available: *
****************************************
   (I)mport key from the server (I).
   (Q)uit.
Choose your action: I or Q: i



* Provide the Key generated by the server.
* The best approach is to cut and paste it.
*** OBS: Do not include spaces or new lines.

Paste it here (or '\q' to quit):
MDA2Ig4LjQzIDkxZDYmRlZGN5mOG5NzY5Nd325dmFmMTU0NzZkddaDM1ND456431MWY1ODhhMDyjukuMDYzg
4MzA5MmM

Agent information:
   ID:007
   Name:test
   IP Address:0.0.0.0

Confirm adding it?(y/n): y
Added.
** Press ENTER to return to the main menu.
```

# Starting OSSEC Agent

That's it! Press enter and exit through the prompts, then just `sudo /var/ossec/bin/ossec-control restart` to apply our changes on both the server and the agent.

Sometimes, when initially starting an agent on a new host you will get like the below -

```
ossec-syscheckd(1226): ERROR: Error reading XML file '/var/ossec/etc/shared/agent.conf': XMLERR: File
'/var/ossec/etc/shared/agent.conf' not found. (line 99).
ossec-syscheckd(1226): ERROR: Error reading XML file '/var/ossec/etc/shared/agent.conf': XMLERR: File
'/var/ossec/etc/shared/agent.conf' not found. (line 99).
```

Just `touch /var/ossec/etc/shared/agent.conf` and `sudo /var/ossec/bin/ossec-control restart` / `sudo /var/ossec/bin/ossec-control start` again. The services should start normally. This is an old bug I came across in this GitHub issue.

# Troubleshooting Agent Connections

If an agent isnt connecting, try the following commands to check for common problems -

## Verify Agent Server Configuration

When attempting to start the OSSEC agent, you may see errors like the following -

```
Deleting PID file '/var/ossec/var/run/ossec-logcollector-23324.pid' not used...
Deleting PID file '/var/ossec/var/run/ossec-agentd-23320.pid' not used...
ossec-logcollector not running ..
ossec-syscheckd not running ..
ossec-agentd not running ..
Killing ossec-execd ..
OSSEC HIDS v2.9.0 Stopped
Starting OSSEC HIDS v2.9.0 (by Trend Micro Inc.)...
Started ossec-execd...
2019/11/04 17:17:16 ossec-agentd: INFO: Using notify time: 600 and max time to reconnect: 1800
Started ossec-agentd...
Started ossec-logcollector...
2019/11/04 17:17:19 ossec-syscheckd(1210): ERROR: Queue '/var/ossec/queue/ossec/queue' not accessible:
'Connection r
efused'.
2019/11/04 17:17:19 rootcheck(1210): ERROR: Queue '/var/ossec/queue/ossec/queue' not accessible:
'Connection refused
'.
```

Checking the logs within `/var/ossec/logs/ossec.log` reveals that no there is no valid OSSEC server configuration. To resolve this, ensure that you used the correct IP during installation for your OSSEC Server, then use the `/var/ossec/bin/manage_agents` tool on both the server and the agent to Add / Export Agent Keys and then run `sudo /var/ossec/bin/ossec-control restart` to restart the agent.

## Check Agent Status on Server

There are many ways to check the agent status on the OSSEC Server host itself. One of which is within the manage_agents utility used to register new agents. Simply run `sudo /var/ossec/bin/manage_agents` and select L to list added agents and their current status with the server.

```
****************************************
* OSSEC HIDS v3.3.0 Agent manager.     *
* The following options are available: *
****************************************
   (A)dd an agent (A).
   (E)xtract key for an agent (E).
   (L)ist already added agents (L).
   (R)emove an agent (R).
   (Q)uit.
Choose your action: A,E,L,R or Q: l

Available agents:
   ID: 001, Name: host, IP: 100.200.300.400
   ID: 002, Name: host2, IP: 200.200.300.400
   ID: 003, Name: host3, IP: 300.200.300.400
   ID: 004, Name: host4, IP: 400.200.300.400

** Press ENTER to return to the main menu.
```

We can also list all agents with the built in OSSEC utility list_agents, run the following command to list all agents, even if they are disconnected or have never been connected in the first place -

```
admin@host:~# /var/ossec/bin/list_agents -a
host-100.200.300.400 is active.
host2-200.200.300.400 is active.
host3-300.200.300.400 is active.
host4-400.200.300.400 is active.
```

## Check OSSEC Logs

OSSEC is an HIDS, which means it takes a ton of logs. Use them to your advantage, in this care we can easily check for general OSSEC errors within the `/var/ossec/logs/ossec.log`

Logs such as the below could indicate an incorrect `client.key`, which is configured when adding the agent to the OSSEC server, and again on the remote host when importing the generated key. If you see errors like this, try going back and removing the agent from the OSSEC server, create a new one and be sure you are using the correct IP for your agent.

```
2019/08/31 18:14:05 ossec-remoted(1213): WARN: Message from '100.200.300.400' not allowed.
2019/08/31 18:14:43 ossec-remoted(1213): WARN: Message from '100.200.300.400' not allowed.
2019/08/31 18:14:49 ossec-remoted(1213): WARN: Message from '100.200.300.400' not allowed.
```

```
2019/08/31 18:14:53 ossec-remoted(1213): WARN: Message from '100.200.300.400' not allowed.
2019/08/31 18:14:58 ossec-remoted(1213): WARN: Message from '100.200.300.400' not allowed
```

It's important to note that when installing the agent on the remote host you wish to monitor, you are prompted to input the Server IP - this is an important step and if done incorrectly just re-run the installation as if you were starting on a fresh host, and when prompted to update OSSEC input NO. Follow the prompts and install in the same location you did previously, and OSSEC will prompt you to remove the previous installation before reinstalling. Do so, and input the correct IP of the host running your OSSEC Server.

A useful trick when debugging mail issues with `ossec-maild` -

```
tail -f /var/ossec/logs/ossec.log | grep ossec-maild
```

For a live feed, filtered log on `ossec-maild` within the base OSSEC server logs.

## Check IPTables

You may need to allow traffic on ports used by OSSEC with `iptables`, to do so, run the following commands. If you make any changes to iptables, be sure to `sudo apt install iptables-persistent`, which will enable iptables to carry over your settings if your host is restarted.

View current iptables -

```
iptables -nL
```

Allow the Agent to connect to our OSSEC Server host on the specified port, run -

```
iptables -A INPUT -p UDP --dport 1514 -s your_agent_ip -j ACCEPT
```

Allow the OSSEC Server to connect to our agent on the specified port, run -

```
iptables -A INPUT -p UDP --dport 1514 -s your_server_ip -j ACCEPT
```

Allow all outbound traffic, assuming no malicious activity will come from within -

```
iptables -A OUTPUT -j ACCEPT
```

## Check Network Traffic

Still not sure why your agent isn't connecting? Try to monitor network traffic on the ports you're running OSSEC across - this could give you some idea of if there is traffic moving on one host and not the other, and lead you to where it is being stopped.

The output below is healthy traffic from my OSSEC server monitoring a few hosts -

```
tcpdump -i eth0 port 1514
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
21:48:48.984246 IP 100.200.300.400.53359 > monitor.1514: UDP, length 265
21:48:48.984314 IP 100.200.300.400.53359 > monitor.1514: UDP, length 265
21:48:49.991446 IP 100.200.300.400.53359 > monitor.1514: UDP, length 233
21:48:49.992233 IP monitor.1514 > 159.65.224.81.53359: UDP, length 73
21:48:53.914955 IP 100.200.300.400.48049 > monitor.1514: UDP, length 265
21:48:54.990058 IP 200.200.300.400.53359 > monitor.1514: UDP, length 249
```

## Duplicate Counter Errors

Sometimes, when attempting an update or reinstall / reconfiguration of the OSSEC server or agents you may see the errors below within your `/var/ossec/logs/ossec.log` -

```
2019/11/05 01:58:16 ossec-remoted: WARN: Duplicate error:  global: 0, local: 7316, saved global: 0, saved local:7317
2019/11/05 01:58:16 ossec-remoted(1407): ERROR: Duplicated counter for 'box1'.
2019/11/05 01:59:20 ossec-remoted: WARN: Duplicate error:  global: 0, local: 5910, saved global: 0, saved local:5911
2019/11/05 01:59:20 ossec-remoted(1407): ERROR: Duplicated counter for 'box2'.
```

OSSEC uses a counter system to avoid repeat alerts on the same attacks. If you recently refactored your monitoring, it is easy to get your wires crossed in the process and knock the OSSEC server out of sync with its agents. This can happen any number of ways, but a resolution can be found in the OSSEC FAQ Docs. The process is detailed below.

```
On every agent:

    stop ossec
    go to: .../ossec/queue/rids (or ossec-agent/rids on Windows) and remove every file in there.

Go to the server:

    Stop ossec
    Remove the rids file with the same name as the agent id that is reporting errors.

Restart the server
Restart the agents.
```

In my case, the error above was exactly as stated - I had left the previous ossec agent running while installing the newer version, so there were two agents trying to report to the same IP. The agents were running on the same box, and therefore had the same source IP which was allowed, so OSSEC was confused and sending groups of the errors below

```
2019/11/05 02:44:38 ossec-remoted: WARN: Duplicate error:  global: 0, local: 6997, saved global: 0, saved local:6999
2019/11/05 02:44:38 ossec-remoted(1407): ERROR: Duplicated counter for 'kapps'.
2019/11/05 02:46:17 ossec-remoted(1403): ERROR: Incorrectly formatted message from '1.2.3.4'.
2019/11/05 02:46:23 ossec-remoted(1403): ERROR: Incorrectly formatted message from '1.2.3.4'.
```

To fix this, I just opened `htop` and hit <F3> to type `ossec` and search for any running agents. After finding them, I killed them with <F9> within `htop`

If you can't seem to chase down what is causing these errors but notice that your OSSEC server and agents are running normally despite noisy alerts, you can disable this feature. This is not reconmmended as it removes a security feature of OSSEC but it can be done by viewing the settings within `/var/ossec/etc/internal_options.conf`. Note that you should not edit this file, but copy any settings you wish to modify or override to `/var/ossec/etc/local_internal_options.conf` and set the values to your liking.

For these errors, to stop monitoring for duplicated messages, we add the following line to `/var/ossec/etc/local_internal_options.conf`, save the file and then restart the OSSEC agent.

```
# Default value is set to 1
# Verify msg id (set to 0 to disable it)
remoted.verify_msg_id=0
```

## Disconnected Agents

I have been neglecting server maintenance for a few months recently, and came to find out that all of my OSSEC agents had crashed.. A few months ago.

On my OSSEC server, `/var/ossec/logs/ossec.log` shows the following logs

```
2022/06/09 00:01:55 INFO: Connected to 127.0.0.1 at address 127.0.0.1, port 25
2022/06/09 03:56:52 rootcheck: INFO: Starting rootcheck scan.
2022/06/09 03:59:48 rootcheck: INFO: Ending rootcheck scan.
2022/06/09 17:44:48 ossec-syscheckd: INFO: Starting syscheck scan.
2022/06/09 18:32:39 ossec-syscheckd: INFO: Ending syscheck scan.
2022/06/09 23:55:01 ossec-reportd: INFO: Started (pid: 1143516).
2022/06/09 23:55:06 ossec-reportd: INFO: Report completed. Creating output...
2022/06/09 23:55:06 ossec-reportd: INFO: Started (pid: 1143527).
```

2022/06/09 23:55:11 ossec-reportd: INFO: Report completed. Creating output...
2022/06/10 00:00:20 ossec-monitord: INFO: Starting daily reporting for 'Daily report: OSSEC Summary'
2022/06/10 00:00:25 ossec-monitord: INFO: Report 'Daily report: OSSEC Summary' completed. Creating output...
2022/06/10 00:00:25 INFO: Connected to 127.0.0.1 at address 127.0.0.1, port 25
2022/06/10 00:02:39 rootcheck: INFO: Starting rootcheck scan.
2022/06/10 00:05:35 rootcheck: INFO: Ending rootcheck scan.
2022/06/10 16:35:35 ossec-syscheckd: INFO: Starting syscheck scan.
2022/06/10 17:23:32 ossec-syscheckd: INFO: Ending syscheck scan.
2022/06/10 20:08:32 rootcheck: INFO: Starting rootcheck scan.
2022/06/10 20:11:28 rootcheck: INFO: Ending rootcheck scan.
2022/06/10 23:55:01 ossec-reportd: INFO: Started (pid: 1155364).
2022/06/10 23:55:06 ossec-reportd: INFO: Report completed. Creating output...
2022/06/10 23:55:06 ossec-reportd: INFO: Started (pid: 1155375).
2022/06/10 23:55:11 ossec-reportd: INFO: Report completed. Creating output...
2022/06/11 00:00:50 ossec-monitord: INFO: Starting daily reporting for 'Daily report: OSSEC Summary'
2022/06/11 00:00:55 ossec-monitord: INFO: Report 'Daily report: OSSEC Summary' completed. Creating output...
# NOTE: Above this line is normal activity

2022/06/11 00:00:55 INFO: Connected to 127.0.0.1 at address 127.0.0.1, port 25
2022/06/11 01:00:39 ossec-logcollector: socketerr (not available).
2022/06/11 01:00:39 ossec-logcollector(1224): ERROR: Error sending message to queue.
2022/06/11 01:00:42 ossec-logcollector(1210): ERROR: Queue '/var/ossec/queue/ossec/queue' not accessible: 'Connection refused'.
2022/06/11 01:00:42 ossec-logcollector(1211): ERROR: Unable to access queue: '/var/ossec/queue/ossec/queue'. Giving up..
2022/06/11 01:00:59 ossec-remoted: socketerr (not available).
2022/06/11 01:00:59 ossec-remoted(1210): ERROR: Queue '/queue/ossec/queue' not accessible: 'Connection refused'.
2022/06/11 01:01:02 ossec-remoted(1210): ERROR: Queue '/queue/ossec/queue' not accessible: 'Connection refused'.
2022/06/11 01:01:02 ossec-remoted(1211): ERROR: Unable to access queue: '/queue/ossec/queue'. Giving up..
2022/06/11 01:31:20 ossec-monitord: socketerr (not available).
2022/06/11 01:31:20 ossec-monitord(1224): ERROR: Error sending message to queue.
2022/06/11 15:26:28 ossec-syscheckd: INFO: Starting syscheck scan.
2022/06/11 15:26:28 ossec-syscheckd: socketerr (not available).
2022/06/11 15:26:28 ossec-syscheckd(1224): ERROR: Error sending message to queue.
2022/06/11 15:26:31 ossec-syscheckd(1210): ERROR: Queue '/var/ossec/queue/ossec/queue' not accessible: 'Connection refused'.
2022/06/11 15:26:31 ossec-syscheckd(1211): ERROR: Unable to access queue: '/var/ossec/queue/ossec/queue'. Giving up..

```
2022/06/11 23:01:04 agent_control(1210): ERROR: Queue '/queue/alerts/ar' not accessible: 'Connection refused'.
2022/06/11 23:01:04 agent_control(1301): ERROR: Unable to connect to active response queue.
2022/06/11 23:55:01 ossec-reportd: INFO: Started (pid: 1157527).
2022/06/11 23:55:06 ossec-reportd: INFO: Report completed. Creating output...
2022/06/11 23:55:06 ossec-reportd: INFO: Started (pid: 1157538).
2022/06/11 23:55:11 ossec-reportd: INFO: Report completed. Creating output...
2022/06/12 00:01:21 ossec-monitord: INFO: Starting daily reporting for 'Daily report: OSSEC Summary'
2022/06/12 00:01:26 ossec-monitord: INFO: Report 'Daily report: OSSEC Summary' completed and zero alerts post-filter.
2022/06/12 00:01:26 ossec-monitord: INFO: Report 'Daily report: OSSEC Summary' empty.
2022/06/12 23:01:04 agent_control(1210): ERROR: Queue '/queue/alerts/ar' not accessible: 'Connection refused'.
2022/06/12 23:01:04 agent_control(1301): ERROR: Unable to connect to active response queue.
2022/06/12 23:55:01 ossec-reportd: INFO: Started (pid: 1159635).
2022/06/12 23:55:06 ossec-reportd: INFO: Report completed. Creating output...
2022/06/12 23:55:06 ossec-reportd: INFO: Started (pid: 1159646).
2022/06/12 23:55:11 ossec-reportd: INFO: Report completed. Creating output...
2022/06/13 00:01:51 ossec-monitord: INFO: Starting daily reporting for 'Daily report: OSSEC Summary'
2022/06/13 00:01:51 ossec-monitord: ERROR: Unable to open alerts file to generate report.
2022/06/13 00:01:51 ossec-monitord: INFO: Report 'Daily report: OSSEC Summary' empty.
2022/06/13 00:02:21 ossec-monitord: File '/logs/alerts/2022/Jun/ossec-alerts-12.log' not found. MD5 checksum skipped.
2022/06/13 00:02:21 ossec-monitord: File '/logs/alerts/2022/Jun/ossec-alerts-12.log' not found. SHA1 checksum skipped.
2022/06/13 23:01:04 agent_control(1210): ERROR: Queue '/queue/alerts/ar' not accessible: 'Connection refused'.
```

On my OSSEC agents, `/var/ossec/logs/ossec.log` shows the following logs

```
2022/06/09 23:08:48 ossec-syscheckd: INFO: Starting syscheck scan.
2022/06/09 23:24:21 ossec-syscheckd: INFO: Ending syscheck scan.
2022/06/10 19:09:21 rootcheck: INFO: Starting rootcheck scan.
2022/06/10 19:12:12 rootcheck: INFO: Ending rootcheck scan.
2022/06/10 21:27:12 ossec-syscheckd: INFO: Starting syscheck scan.
2022/06/10 21:42:46 ossec-syscheckd: INFO: Ending syscheck scan.
2022/06/10 23:02:46 rootcheck: INFO: Starting rootcheck scan.
2022/06/10 23:05:37 rootcheck: INFO: Ending rootcheck scan.
2022/06/10 23:05:37 ossec-syscheckd: INFO: Starting syscheck scan.
2022/06/10 23:21:12 ossec-syscheckd: INFO: Ending syscheck scan.
# NOTE: Above this line is normal activity

2022/06/11 01:03:33 ossec-agentd(1218): ERROR: Unable to send message to 'server'.
```

```
2022/06/11 01:03:34 ossec-agentd(1218): ERROR: Unable to send message to 'server'.
2022/06/11 01:09:36 ossec-agentd(1218): ERROR: Unable to send message to 'server'.
2022/06/11 01:09:37 ossec-agentd(1218): ERROR: Unable to send message to 'server'.
2022/06/11 01:09:38 ossec-agentd(1218): ERROR: Unable to send message to 'server'.
2022/06/11 01:09:39 ossec-agentd(1218): ERROR: Unable to send message to 'server'.
2022/06/11 01:09:40 ossec-agentd(1218): ERROR: Unable to send message to 'server'.
2022/06/11 01:09:41 ossec-agentd(1218): ERROR: Unable to send message to 'server'.
2022/06/11 01:15:36 ossec-agentd(1218): ERROR: Unable to send message to 'server'.
2022/06/11 01:15:37 ossec-agentd(1218): ERROR: Unable to send message to 'server'.
2022/06/11 01:15:38 ossec-agentd(1218): ERROR: Unable to send message to 'server'.
2022/06/11 01:21:38 ossec-agentd(1218): ERROR: Unable to send message to 'server'.
2022/06/11 01:21:39 ossec-agentd(1218): ERROR: Unable to send message to 'server'.
2022/06/11 01:21:40 ossec-agentd(1218): ERROR: Unable to send message to 'server'.
2022/06/11 01:27:41 ossec-agentd(1218): ERROR: Unable to send message to 'server'.
2022/06/11 01:27:42 ossec-agentd(1218): ERROR: Unable to send message to 'server'.
2022/06/11 01:27:43 ossec-agentd(1218): ERROR: Unable to send message to 'server'.
2022/06/11 01:29:36 ossec-agentd: WARN: Server unavailable. Setting lock.
2022/06/11 01:29:46 ossec-agentd(1218): ERROR: Unable to send message to 'server'.
2022/06/11 01:29:58 ossec-agentd(1218): ERROR: Unable to send message to 'server'.
2022/06/11 01:29:59 ossec-agentd(4101): WARN: Waiting for server reply (not started). Tried: '159.203.190.63'.
2022/06/11 01:30:01 ossec-agentd: INFO: Trying to connect to server 159.203.190.63, port 1514.
2022/06/11 01:30:01 INFO: Connected to 159.203.190.63 at address 159.203.190.63, port 1514
2022/06/11 01:30:11 ossec-agentd(1218): ERROR: Unable to send message to 'server'.
2022/06/11 01:30:21 ossec-logcollector: WARN: Process locked. Waiting for permission...
2022/06/11 01:30:23 ossec-agentd(1218): ERROR: Unable to send message to 'server'.
2022/06/11 01:30:24 ossec-agentd(4101): WARN: Waiting for server reply (not started). Tried: '159.203.190.63'.
2022/06/11 01:30:44 ossec-agentd: INFO: Trying to connect to server 159.203.190.63, port 1514.
2022/06/11 01:30:44 INFO: Connected to 159.203.190.63 at address 159.203.190.63, port 1514
2022/06/11 01:30:54 ossec-agentd(1218): ERROR: Unable to send message to 'server'.
2022/06/11 01:31:06 ossec-agentd(1218): ERROR: Unable to send message to 'server'.
2022/06/11 01:31:07 ossec-agentd(4101): WARN: Waiting for server reply (not started). Tried: '159.203.190.63'.
# ... Literally 3 months of logs like this; try, fail
```

Fixed with the following commands

On the OSSEC server -

```
sudo su
cd /var/ossec/bin/
./agent_control -l
```

```
OSSEC HIDS agent_control. List of available agents:
   ID: 000, Name: not (server), IP: 127.0.0.1, Active/Local
   ID: 001, Name: real, IP: 1.1.1.1, Disconnected
   ID: 002, Name: agent, IP: 2.2.2.2, Disconnected
   ID: 003, Name: names, IP: 3.3.3.3, Disconnected
```

These agents are all normally connected. My email logs are showing only reports for the OSSEC server for the last 3 months. Nice.

My server runs this command every day at a certain time, and if it fails I get an email alert. I've been ignoring them, but I guess the alerting has done its job here. I knew there was a problem, I just didn't care.

```
./agent_control -r -a
2022/09/12 23:17:08 agent_control(1210): ERROR: Queue '/queue/alerts/ar' not accessible: 'Connection ref
used'.
2022/09/12 23:17:08 agent_control(1301): ERROR: Unable to connect to active response queue.

** Unable to connect to remoted.
```

To fix this all I had to do was restart the agents. On my server *and* the remote agents. OSSEC makes this simple, and it's just a single command that is the same for the server and agents.

```
cd /var/ossec/bin
./ossec-control restart
```

The worst part of fixing this was logging into all the servers and resetting them manually. I feel they're decently secure for what they are, but I have not made authenticating as root a simple process, even for myself. Anyhow - after running the command above on all your servers things should be fine.

```
cd /var/ossec/bin/
./agent_control -l

OSSEC HIDS agent_control. List of available agents:
   ID: 000, Name: not (server), IP: 127.0.0.1, Active/Local
   ID: 001, Name: real, IP: 1.1.1.1, Active
   ID: 002, Name: agent, IP: 2.2.2.2, Active
   ID: 003, Name: names, IP: 3.3.3.3, Active
```

# Links

More links:

OSSEC Installation Tutorial

# 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/mtab</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 intergrity 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 properies 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 |