**Installing Wazuh:**

To start setting up Wazuh, add the Wazuh repository to the server.

1. root@manager# install the necessary packages for the installation

root@manager# apt install curl apt-transport-https unzip wget libcap2-bin software-properties-common lsb-release gnupg

1. Install the GPG key:

root@manager# curl -s https://packages.wazuh.com/key/GPG-KEY-WAZUH | apt-key add -

1. Add repository

root@manager# echo "deb https://packages.wazuh.com/4.x/apt/ stable main" | tee -a /etc/apt/sources.list.d/wazuh.list

1. Update the package information:

root@manager# apt-get update

### **Installing the Wazuh manager**

1. Install the Wazuh manager package:

root@manager# apt-get install wazuh-manager=4.2.7-1

1. Enable and start the Wazuh manager service

root@manager# systemctl daemon-reload

root@manager# systemctl enable wazuh-manager

root@manager# systemctl start wazuh-manager

1. Run the following command to check if the Wazuh manager is active:

root@manager# systemctl status wazuh-manager

## Installing Elasticsearch:

1. Install Elasticsearch OSS and Open Distro for Elasticsearch:

root@manager# apt install elasticsearch-oss opendistroforelasticsearch

1. Configuring Elasticsearch

root@manager# curl -so /etc/elasticsearch/elasticsearch.yml <https://packages.wazuh.com/resources/4.2/open-distro/elasticsearch/7.x/elasticsearch_all_in_one.yml>

1. Run the following commands to add the Wazuh users and additional roles in Kibana:

root@manager#curl -so /usr/share/elasticsearch/plugins/opendistro\_security/securityconfig/roles.yml <https://packages.wazuh.com/resources/4.2/open-distro/elasticsearch/roles/roles.yml>

root@manager # curl -so /usr/share/elasticsearch/plugins/opendistro\_security/securityconfig/roles\_mapping.yml <https://packages.wazuh.com/resources/4.2/open-distro/elasticsearch/roles/roles_mapping.yml>

root@manager# curl -so /usr/share/elasticsearch/plugins/opendistro\_security/securityconfig/internal\_users.yml <https://packages.wazuh.com/resources/4.2/open-distro/elasticsearch/roles/internal_users.yml>

### Certificates creation

1. Remove the demo certificates:

root@manager# rm /etc/elasticsearch/esnode-key.pem /etc/elasticsearch/esnode.pem /etc/elasticsearch/kirk-key.pem /etc/elasticsearch/kirk.pem /etc/elasticsearch/root-ca.pem –f

1. Generate the new certificate and deploy the certificates:

Download tools from the wazuh repository

curl -so ~/wazuh-cert-tool.sh <https://packages.wazuh.com/resources/4.2/open-distro/tools/certificate-utility/wazuh-cert-tool.sh>

curl -so ~/instances.yml <https://packages.wazuh.com/resources/4.2/open-distro/tools/certificate-utility/instances_aio.yml>

Run the wazuh-cert-tool.sh to create the certificates:

#bash ~/wazuh-cert-tool.sh

1. Move the Elasticsearch certificates to their corresponding location:

# mkdir /etc/elasticsearch/certs/

# mv ~/certs/elasticsearch\* /etc/elasticsearch/certs/

# mv ~/certs/admin\* /etc/elasticsearch/certs/

# cp ~/certs/root-ca\* /etc/elasticsearch/certs

1. Enable and start the Elasticsearch service:

# systemctl daemon-reload

# systemctl enable elasticsearch

# systemctl start elasticsearch

1. Run the Elasticsearch securityadmin script to load the new certificates information and start the cluster:

# export JAVA\_HOME=/usr/share/elasticsearch/jdk/ && /usr/share/elasticsearch/plugins/opendistro\_security/tools/securityadmin.sh -cd /usr/share/elasticsearch/plugins/opendistro\_security/securityconfig/ -nhnv -cacert /etc/elasticsearch/certs/root-ca.pem -cert /etc/elasticsearch/certs/admin.pem -key /etc/elasticsearch/certs/admin-key.pem

1. Run the following command to ensure that the installation is successful:

#curl -XGET https://localhost:9200 -u admin:admin –k

An example response should look as follows:

{

"name" : "node-1",

"cluster\_name" : "elasticsearch",

"cluster\_uuid" : "tWYgqpgdRz6fGN8gH11flw",

"version" : {

"number" : "7.10.2",

"build\_flavor" : "oss",

"build\_type" : "rpm",

"build\_hash" : "747e1cc71def077253878a59143c1f785afa92b9",

"build\_date" : "2021-01-13T00:42:12.435326Z",

"build\_snapshot" : false,

"lucene\_version" : "8.7.0",

"minimum\_wire\_compatibility\_version" : "6.8.0",

"minimum\_index\_compatibility\_version" : "6.0.0-beta1"

},

"tagline" : "You Know, for Search"

}

## Installing Filebeat:

Filebeat is the tool on the Wazuh server that securely forwards alerts and archived events to Elasticsearch.

1. Install the Filebeat package:

#apt-get install filebeat

1. Download the preconfigured Filebeat configuration file used to forward the Wazuh alerts to Elasticsearch:

#curl -so /etc/filebeat/filebeat.yml https://packages.wazuh.com/resources/4.2/open-distro/filebeat/7.x/filebeat\_all\_in\_one.yml

1. Download the preconfigured Filebeat configuration file used to forward the Wazuh alerts to Elasticsearch

#curl -so /etc/filebeat/wazuh-template.json <https://raw.githubusercontent.com/wazuh/wazuh/4.2/extensions/elasticsearch/7.x/wazuh-template.json>

# chmod go+r /etc/filebeat/wazuh-template.json

1. Download the Wazuh module for Filebeat

#curl -s https://packages.wazuh.com/4.x/filebeat/wazuh-filebeat-0.1.tar.gz | tar -xvz -C /usr/share/filebeat/module

1. Copy the Elasticsearch certificates into /etc/filebeat/certs:

#mkdir /etc/filebeat/certs

#cp ~/certs/root-ca.pem /etc/filebeat/certs/

#mv ~/certs/filebeat\* /etc/filebeat/certs/

1. Enable and start the Filebeat service:

systemctl daemon-reload

systemctl enable filebeat

systemctl start filebeat

1. To ensure that Filebeat is successfully installed, run the following command:

filebeat test output

elasticsearch: https://127.0.0.1:9200...

parse url... OK

connection...

parse host... OK

dns lookup... OK

addresses: 127.0.0.1

dial up... OK

TLS...

security: server's certificate chain verification is enabled

handshake... OK

TLS version: TLSv1.3

dial up... OK

talk to server... OK

version: 7.10.2

## Installing Kibana:

1. Install the Kibana package:

#apt-get install opendistroforelasticsearch-kibana

1. Download the Kibana configuration file:

curl -so /etc/kibana/kibana.yml https://packages.wazuh.com/resources/4.2/open-distro/kibana/7.x/kibana\_all\_in\_one.yml

1. Create the /usr/share/kibana/data directory:

# mkdir /usr/share/kibana/data

# chown -R kibana:kibana /usr/share/kibana/data

1. Install the Wazuh Kibana plugin. The installation of the plugin must be done from the Kibana home directory as follows:

# cd /usr/share/kibana

# sudo -u kibana /usr/share/kibana/bin/kibana-plugin install <https://packages.wazuh.com/4.x/ui/kibana/wazuh_kibana-4.2.7_7.10.2-1.zip>

1. Copy the Elasticsearch certificates into /etc/kibana/certs:

#mkdir /etc/kibana/certs

#cp ~/certs/root-ca.pem /etc/kibana/certs/

#mv ~/certs/kibana\* /etc/kibana/certs/

#chown kibana:kibana /etc/kibana/certs/\*

1. Link Kibana socket to privileged port 443

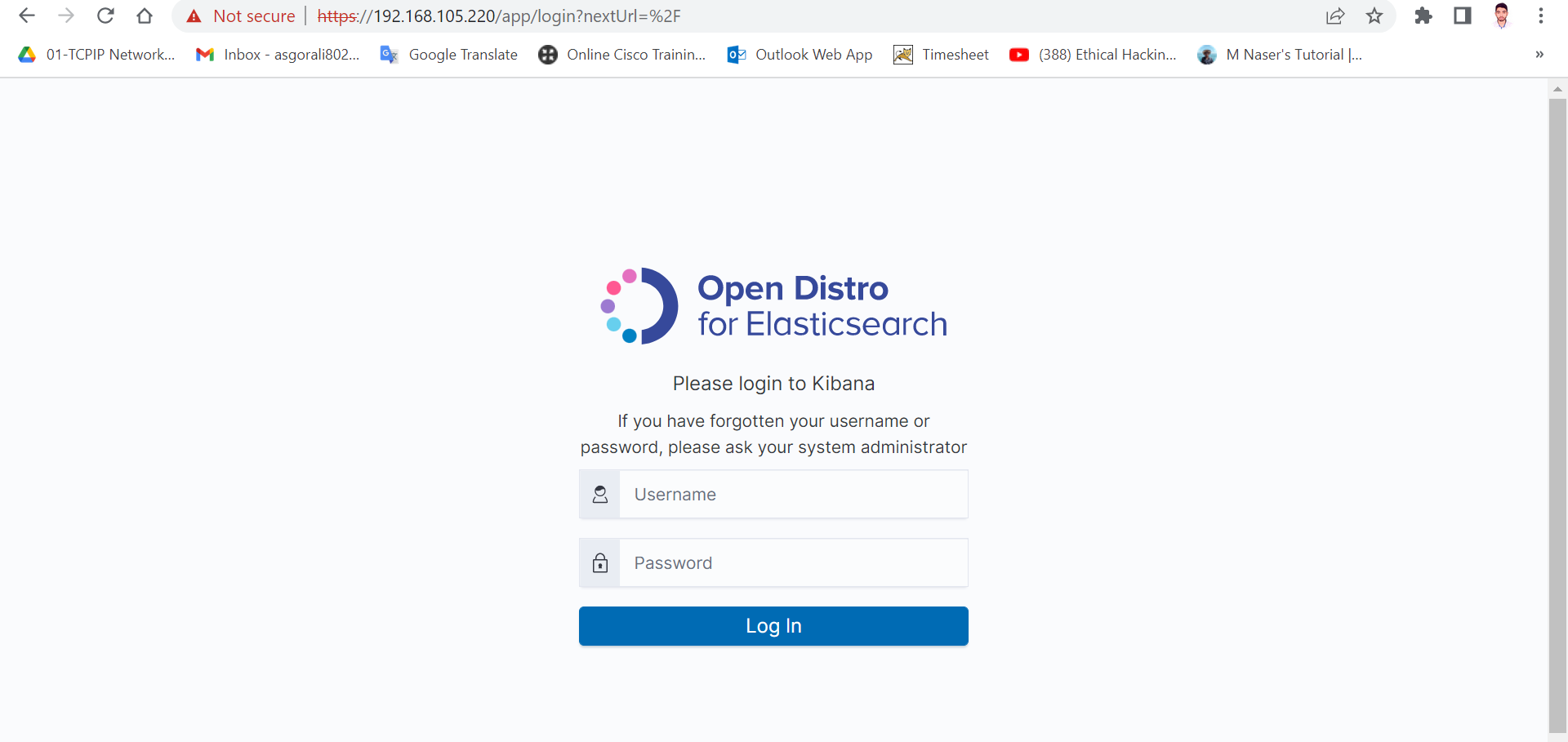
#setcap 'cap\_net\_bind\_service=+ep' /usr/share/kibana/node/bin/node

1. Enable and start the Kibana service:

# systemctl daemon-reload

# systemctl enable kibana

# systemctl start kibana



**System Configuration:**

<ossec\_config>

<!—This Configurations for Log collection from Agent-->

<global>

<jsonout\_output>yes</jsonout\_output>

<alerts\_log>yes</alerts\_log>

<logall>no</logall>

<logall\_json>no</logall\_json>

<email\_log\_source>alerts.log</email\_log\_source>

<agents\_disconnection\_time>10m</agents\_disconnection\_time>

<agents\_disconnection\_alert\_time>0</agents\_disconnection\_alert\_time>

</global>

<alerts>

<log\_alert\_level>3</log\_alert\_level>

<email\_alert\_level>12</email\_alert\_level>

</alerts>

<!-- Choose between "plain", "json", or "plain,json" for the format of internal logs -->

<logging>

<log\_format>plain</log\_format>

</logging>

<remote>

<connection>secure</connection>

<port>1514</port>

<protocol>tcp</protocol>

<queue\_size>131072</queue\_size>

</remote>

<!-- Policy monitoring -->

<!-- Frequency that rootcheck is executed - every 12 hours -->

<rootcheck>

<disabled>no</disabled>

<check\_files>yes</check\_files>

<check\_trojans>yes</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>

<frequency>43200</frequency>

<rootkit\_files>etc/rootcheck/rootkit\_files.txt</rootkit\_files>

<rootkit\_trojans>etc/rootcheck/rootkit\_trojans.txt</rootkit\_trojans>

<skip\_nfs>yes</skip\_nfs>

</rootcheck>

<!-- VirusTotal API Integration -->

<integration>

<name>virustotal</name><api\_key>abd74847f974f22ee9099ee78b548be96b3eed4e19f170369272c7a5ccf40b66</api\_key>

<group>syscheck</group>

<alert\_format>json</alert\_format>

</integration>

<!—OS query integration -->

<wodle name="osquery">

<disabled>yes</disabled>

<run\_daemon>yes</run\_daemon>

<log\_path>/var/log/osquery/osqueryd.results.log</log\_path>

<config\_path>/etc/osquery/osquery.conf</config\_path>

<add\_labels>yes</add\_labels>

</wodle>

<!-- System inventory Configuarion -->

<wodle name="syscollector">

<disabled>no</disabled>

<interval>1h</interval>

<scan\_on\_start>yes</scan\_on\_start>

<hardware>yes</hardware>

<os>yes</os>

<network>yes</network>

<packages>yes</packages>

<ports all="no">yes</ports>

<processes>yes</processes>

<!-- Database synchronization settings -->

<synchronization>

<max\_eps>10</max\_eps>

</synchronization>

</wodle>

<sca>

<enabled>yes</enabled>

<scan\_on\_start>yes</scan\_on\_start>

<interval>12h</interval>

<skip\_nfs>yes</skip\_nfs>

</sca>

<vulnerability-detector>

<enabled>yes</enabled>

<interval>5m</interval>

<ignore\_time>6h</ignore\_time>

<run\_on\_start>yes</run\_on\_start>

<!-- Ubuntu OS vulnerabilities -->

<provider name="canonical">

<enabled>yes</enabled>

<os>trusty</os>

<os>xenial</os>

<os>bionic</os>

<os>focal</os>

<update\_interval>1h</update\_interval>

</provider>

<!-- Debian OS vulnerabilities -->

<provider name="debian">

<enabled>yes</enabled>

<os>stretch</os>

<os>buster</os>

<update\_interval>1h</update\_interval>

</provider>

<provider name="debian">

<enabled>yes</enabled>

<os>stretch</os>

<os>buster</os>

<update\_interval>1h</update\_interval>

</provider>

<!-- RedHat OS vulnerabilities -->

<provider name="redhat">

<enabled>yes</enabled>

<os>5</os>

<os>6</os>

<os>7</os>

<os>8</os>

<update\_interval>1h</update\_interval>

</provider>

<!-- Windows OS vulnerabilities -->

<provider name="msu">

<enabled>yes</enabled>

<update\_interval>1h</update\_interval>

</provider>

<!-- Aggregate vulnerabilities -->

<provider name="nvd">

<enabled>yes</enabled>

<update\_from\_year>2010</update\_from\_year>

<update\_interval>1h</update\_interval>

</provider>

</vulnerability-detector>

<!-- File integrity monitoring -->

<syscheck>

<disabled>no</disabled>

<!-- Frequency that syscheck is executed default every 12 hours -->

<frequency>43200</frequency>

<scan\_on\_start>yes</scan\_on\_start>

<!-- Generate alert when new file detected -->

<alert\_new\_files>yes</alert\_new\_files>

<!-- Don't ignore files that change more than 'frequency' times -->

<auto\_ignore frequency="10" timeframe="3600">no</auto\_ignore>

<!-- Directories to check (perform all possible verifications) -->

<directories>/etc,/usr/bin,/usr/sbin</directories>

<directories>/bin,/sbin,/boot</directories>

<!-- Files/directories to ignore -->

<ignore>/etc/mtab</ignore>

<ignore>/etc/hosts.deny</ignore>

<ignore>/etc/mail/statistics</ignore>

<ignore>/etc/random-seed</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>

<!-- File types to ignore -->

<ignore type="sregex">.log$|.swp$</ignore>

<!-- Check the file, but never compute the diff -->

<nodiff>/etc/ssl/private.key</nodiff>

<skip\_nfs>yes</skip\_nfs>

<skip\_dev>yes</skip\_dev>

<skip\_proc>yes</skip\_proc>

<skip\_sys>yes</skip\_sys>

<!-- Nice value for Syscheck process -->

<process\_priority>10</process\_priority>

<!-- Maximum output throughput -->

<max\_eps>100</max\_eps>

<!-- Database synchronization settings -->

<synchronization>

<enabled>yes</enabled>

<interval>5m</interval>

<max\_interval>1h</max\_interval>

<max\_eps>10</max\_eps>

</synchronization>

</syscheck>

<command>

<name>disable-account</name>

<executable>disable-account</executable>

<timeout\_allowed>yes</timeout\_allowed>

</command>

<command>

<name>restart-wazuh</name>

<executable>restart-wazuh</executable>

</command>

<command>

<name>firewall-drop</name>

<executable>firewall-drop</executable>

<timeout\_allowed>yes</timeout\_allowed>

</command>

<command>

<name>host-deny</name>

<executable>host-deny</executable>

<timeout\_allowed>yes</timeout\_allowed>

</command>

<command>

<name>route-null</name>

<executable>route-null</executable>

<timeout\_allowed>yes</timeout\_allowed>

</command>

<command>

<name>win\_route-null</name>

<executable>route-null.exe</executable>

<timeout\_allowed>yes</timeout\_allowed>

</command>

<!--

<active-response>

<active-response>

<disabled>no</disabled>

<command>firewall-drop</command>

<location>local</location>

<rules\_id>31168</rules\_id>

<timeout>300</timeout>

</active-response>

-->

<!-- Log analysis -->

<localfile>

<log\_format>command</log\_format>

<command>df -P</command>

<frequency>360</frequency>

</localfile>

<localfile>

<log\_format>full\_command</log\_format>

<command>netstat -tulpn | sed 's/\([[:alnum:]]\+\)\ \+[[:digit:]]\+\ \+[[:digit:]]\+\ \+\(.\*\):\([[:digit:]]\*\)\ \+\([0-9\.\:\\*]\+\).\+\ \([[:digit:]]\*\/[[:alnum:]\-]\*\).\*/\1 \2 == \3 == \4 \5/' | sort -k 4 -g | sed 's/ == \(.\*\) ==/:\1/' | sed 1,2d</command>

<alias>netstat listening ports</alias>

<frequency>360</frequency>

</localfile>

<localfile>

<log\_format>full\_command</log\_format>

<command>last -n 20</command>

<frequency>360</frequency>

</localfile>

<ruleset>

<!-- Default ruleset -->

<decoder\_dir>ruleset/decoders</decoder\_dir>

<rule\_dir>ruleset/rules</rule\_dir>

<rule\_exclude>0215-policy\_rules.xml</rule\_exclude>

<list>etc/lists/audit-keys</list>

<list>etc/lists/amazon/aws-eventnames</list>

<list>etc/lists/security-eventchannel</list>

<list>etc/lists/blacklist-alienvault</list>

<!-- User-defined ruleset -->

<decoder\_dir>etc/decoders</decoder\_dir>

<rule\_dir>etc/rules</rule\_dir>

</ruleset>

<rule\_test>

<enabled>yes</enabled>

<threads>1</threads>

<max\_sessions>64</max\_sessions>

<session\_timeout>15m</session\_timeout>

</rule\_test>

<!-- Configuration for wazuh-authd -->

<auth>

<disabled>no</disabled>

<port>1515</port>

<use\_source\_ip>no</use\_source\_ip>

<force\_insert>yes</force\_insert>

<force\_time>0</force\_time>

<purge>yes</purge>

<use\_password>no</use\_password>

<ciphers>HIGH:!ADH:!EXP:!MD5:!RC4:!3DES:!CAMELLIA:@STRENGTH</ciphers>

<!-- <ssl\_agent\_ca></ssl\_agent\_ca> -->

<ssl\_verify\_host>no</ssl\_verify\_host>

<ssl\_manager\_cert>etc/sslmanager.cert</ssl\_manager\_cert>

<ssl\_manager\_key>etc/sslmanager.key</ssl\_manager\_key>

<ssl\_auto\_negotiate>no</ssl\_auto\_negotiate>

</auth>

<cluster>

<name>wazuh</name>

<node\_name>node01</node\_name>

<node\_type>master</node\_type>

<key></key>

<port>1516</port>

<bind\_addr>0.0.0.0</bind\_addr>

<nodes>

<node>NODE\_IP</node>

</nodes>

<hidden>no</hidden>

<disabled>yes</disabled>

</cluster>

</ossec\_config>

<ossec\_config>

<localfile>

<log\_format>syslog</log\_format>

<location>/var/ossec/logs/active-responses.log</location>

</localfile>

<localfile>

<log\_format>syslog</log\_format>

<location>/var/log/auth.log</location>

</localfile>

<localfile>

<log\_format>syslog</log\_format>

<location>/var/log/syslog</location>

</localfile>

<localfile>

<log\_format>syslog</log\_format>

<location>/var/log/dpkg.log</location>

</localfile>

<localfile>

<log\_format>syslog</log\_format>

<location>/var/log/kern.log</location>

</localfile>

</ossec\_config>