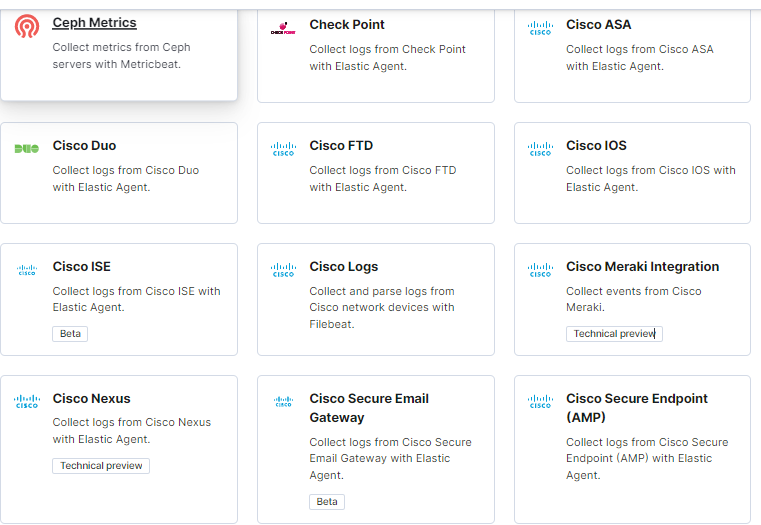
Cisco modules to be monitor by ELK:



* asa fileset: supports Cisco ASA firewall logs.
* amp fileset: supports Cisco AMP API logs.
* ftd fileset: supports Cisco Firepower Threat Defense logs.
* ios fileset: supports Cisco IOS router and switch logs.
* nexus fileset: supports Cisco Nexus switch logs.
* meraki fileset: supports Cisco Meraki logs.
* umbrella fileset: supports Cisco Umbrella logs.

## CISCO ASA:

input {

# Receive Cisco ASA logs on the standard syslog UDP port 514

udp {

port => 21002

#type => "cisco-asa"

type => "ASA Firewall 2"

}

udp {

port => 21008

#type => "cisco-asa"

type => "ASA Firewall 8"

}

#tcp {

# port => 21002

# type => "cisco-asa"

#}

#tcp {

# port => 21008

# type => "cisco-asa"

#}

}

filter {

# Split the syslog part and Cisco tag out of the message

grok {

match => ["message", "%{CISCO\_TAGGED\_SYSLOG} %{GREEDYDATA:cisco\_message}"]

}

# Parse the syslog severity and facility

syslog\_pri { }

# Parse the date from the "timestamp" field to the "@timestamp" field

date {

match => ["timestamp",

"MMM dd HH:mm:ss",

"MMM d HH:mm:ss",

"MMM dd yyyy HH:mm:ss",

"MMM d yyyy HH:mm:ss"

]

timezone => "America/Chicago"

}

# Clean up redundant fields if parsing was successful

if "\_grokparsefailure" not in [tags] {

mutate {

rename => ["cisco\_message", "message"]

remove\_field => ["timestamp"]

}

}

#if [%{port}] == "21002" {

# mutate {

# add\_field => ["device", "ASA Firewall 2"]

# }

#}

#else if "port" == "21008" {

# mutate {

# add\_field => ["device", "ASA Firewall 8"]

# }

#}

#else{

# mutate {

# add\_field => ["device","THIS WAS AN ELSE"]

# #add\_field => ["JUUUUUUNKKKK", {port}] - bad syntax

# #add\_field => ["JUUUUUUNKKKK", "{port}"] - "JUUUUUUNKKKK" => "{port}",

# #add\_field => ["JUUUUUUNKKKK", port] - "JUUUUUUNKKKK" => "port",

# add\_field => ["EXTRA TYPE", "%{type}"]

# add\_field => ["JUUUUUUNKKKK", %{port}] - bad syntax

# add\_field => ["JUUUUUUNKKKK", "%{port}"]

#

#

# }

#}

# Extract fields from the each of the detailed message types

# The patterns provided below are included in Logstash since 1.2.0

grok {

match => [

"message", "%{CISCOFW106001}",

"message", "%{CISCOFW106006\_106007\_106010}",

"message", "%{CISCOFW106014}",

"message", "%{CISCOFW106015}",

"message", "%{CISCOFW106021}",

"message", "%{CISCOFW106023}",

"message", "%{CISCOFW106100}",

"message", "%{CISCOFW110002}",

"message", "%{CISCOFW302010}",

"message", "%{CISCOFW302013\_302014\_302015\_302016}",

"message", "%{CISCOFW302020\_302021}",

"message", "%{CISCOFW305011}",

"message", "%{CISCOFW313001\_313004\_313008}",

"message", "%{CISCOFW313005}",

"message", "%{CISCOFW402117}",

"message", "%{CISCOFW402119}",

"message", "%{CISCOFW419001}",

"message", "%{CISCOFW419002}",

"message", "%{CISCOFW500004}",

"message", "%{CISCOFW602303\_602304}",

"message", "%{CISCOFW710001\_710002\_710003\_710005\_710006}",

"message", "%{CISCOFW713172}",

"message", "%{CISCOFW733100}"

]

}

}

output {

# Archive Cisco ASA firewall logs on disk based on the event's timestamp

# Results in directories for each year and month, with conveniently-named log files, like:

# /path/to/archive/cisco-asa/2014/2014-09/cisco-asa-2014-09-24.log

stdout { codec => rubydebug }

#file {

# path => "/opt/logstash\_logs/%{type}/%{+YYYY}/%{+YYYY-MM}/%{type}-%{+YYYY-MM-dd}.log"

#}

# Also output to ElasticSearch for review in Kibana

elasticsearch {

hosts => ["localhost:9200"]

}

}

## CISCO AMP:

input{

exec{

command => "bash /etc/run-amp.sh"

codec => json

type => "Cisco\_AMP\_EP"

interval => "300"

}

}

filter {

if "Cisco\_AMP\_EP" in [type] {

json { source => "message" }

split{ field => "data" }

#split { field => "[data]" }

split { field => "[data][computer][network\_addresses]" }

mutate {

remove\_field => [ "command" ]

remove\_field => [ "[metadata]" ]

#remove\_field => [ "[data][vulnerabilities]" ]

rename => { "[data][computer][hostname]" => "hostname" }

rename => { "[data][computer][external\_ip]" => "external\_ip" }

rename => { "[data][computer][network\_addresses][ip]" => "internal\_ip" }

rename => { "[data][computer][network\_addresses][mac]" => "mac\_address" }

rename => { "[data][computer][active]" => "active" }

rename => { "[data][event\_type]" => "event\_type" }

rename => { "[data][file][disposition]" => "file\_disposition" }

rename => { "[data][file][identity][sha256]" => "sha256" }

rename => { "[data][computer][connector\_guid]" => "connector\_guid" }

rename => { "[data][computer][links][computer]" => "link\_computer" }

rename => { "[data][computer][links][group]" => "link\_group" }

rename => { "[data][computer][links][trajectory]" => "link\_trajectory" }

rename => { "[data][computer][user]" => "username" }

rename => { "[data][detection]" => "detection" }

rename => { "[data][file][archived\_file][disposition]" => "archived\_file\_disposition" }

rename => { "[data][file][archived\_file][identity][sha256]" => "archived\_file\_sha256" }

rename => { "[data][file][parent][file\_name]" => "parent\_filename" }

rename => { "[data][file][parent][identity][md5]" => "parent\_md5" }

rename => { "[data][file][parent][identity][sha1]" => "parent\_sha1" }

rename => { "[data][file][parent][process\_id]" => "parent\_process\_id" }

rename => { "[data][error][description]" => "error\_description" }

rename => { "[data][error][code]" => "error\_code" }

rename => { "[data][start\_timestamp]" => "start\_timestamp" }

rename => { "[data][start\_date]" => "start\_date" }

rename => { "[data][scan][description]" => "data\_scan\_description" }

rename => { "[data][date]" => "date" }

rename => { "[data][detection\_id]" => "detection\_id" }

rename => { "[data][event\_type\_id]" => "event\_type\_id" }

rename => { "[data][group\_guids]" => "group\_guids" }

rename => { "[data][id]" => "id" }

rename => { "[data][timestamp]" => "timestamp" }

rename => { "[data][timestamp\_nanoseconds]" => "timestamp\_nanoseconds" }

rename => { "[data][file][file\_name]" => "file\_name" }

rename => { "[data][file][file\_path]" => "file\_path" }

rename => { "[data][file][parent][disposition]" => "parent\_disposition" }

rename => { "[data][file][parent][identity][sha256]" => "parent\_sha256" }

rename => { "[data][generic\_ioc][description]" => "generic\_ioc\_description" }

rename => { "[data][generic\_ioc][short\_description]" => "generic\_ioc\_short\_description" }

rename => { "[data][network\_info][dirty\_url]" => "dirty\_url" }

rename => { "[data][network\_info][local\_ip]" => "local\_ip" }

rename => { "[data][network\_info][remote\_ip]" => "remote\_ip" }

rename => { "[data][network\_info][local\_port]" => "local\_port" }

rename => { "[data][network\_info][parent][disposition]" => "network\_info\_parent\_dispostion" }

rename => { "[data][network\_info][parent][file\_name]" => "network\_info\_parent\_filename" }

rename => { "[data][network\_info][parent][identity][md5]" => "network\_info\_parent\_md5" }

rename => { "[data][network\_info][parent][identity][sha1]" => "network\_info\_parent\_sha1" }

rename => { "[data][network\_info][parent][identity][sha256]" => "network\_info\_parent\_sha256" }

rename => { "[data][network\_info][parent][process\_id]" => "network\_info\_parent\_process\_id" }

rename => { "[data][network\_info][remote\_port]" => "network\_info\_remote\_port" }

}

if [data][vulnerabilities] {

split { field => "[data][vulnerabilities]" }

mutate {

rename => { "[data][vulnerabilities][name]" => "vulnerable\_software" }

rename => { "[data][vulnerabilities][score]" => "vulnerable\_software\_score" }

rename => { "[data][vulnerabilities][cve]" => "vulnerable\_software\_cve" }

rename => { "[data][vulnerabilities][url]" => "vulnerable\_software\_url" }

rename => { "[data][vulnerabilities][version]" => "vulnerable\_software\_version" }

}

}

if "Cisco\_AMP\_EP" in [type] {

grok {

match => [ "hostname", "%{DATA:barcode}\.%{DATA}\.%{DATA}\.%{DATA}" ]

}

}

if "Cisco\_AMP\_EP" in [type] {

mutate {

copy => { "internal\_ip" => "srcip" }

copy => { "remote\_ip" => "dstip" }

add\_field => [ "Virus\_Total\_lookup", "https://www.virustotal.com/#/file/%{sha256}/detection" ]

add\_field => [ "Virus\_Total\_lookup", "https://www.virustotal.com/#/file/%{parent\_sha256}/detection" ]

add\_field => [ "Service\_Now\_Asset\_lookup", "https://airproducts.service-now.com/nav\_to.do?uri=%2Fcmdb\_ci\_pc\_hardware\_list.do%3Fsysparm\_offset%3D%26sysparm\_list\_mode%3Dgrid%26sysparm\_query%3D123TEXTQUERY321%253D%{barcode}" ]

add\_field => [ "AMP\_EndPoints\_Barcode\_lookup", "https://console.amp.cisco.com/search?query=%{barcode}" ]

}

}

}

}

output {

if "Cisco\_AMP\_EP" in [type] {

elasticsearch {

hosts => localhost

index => "apci-ampep-%{+YYYY.MM.dd}" }

}

}

## CISCO FTD :

input {  
tcp {  
port => 5000  
}  
tcp {  
port => 5514  
type => "FTDlog"  
}  
udp {  
port => 5514  
type => "FTDlog"  
}  
}  
filter {  
if [type]== "FTDlog" {  
grok {  
match=> {  
"message"=>"<%{POSINT:syslog\_pri}>%{TIMESTAMP\_ISO8601:timestamp}%{SPACE}\%FTD-%{DATA:severity}-%{DATA:eventid}: %{GREEDYDATA:msg}"  
}  
}  
if "\_grokparsefailure" not in [tags] {  
kv {  
source => "msg"  
field\_split => ","  
value\_split => ":"  
trim\_key => " "  
}  
geoip {  
source => "DstIP"  
target => "GeoDstIp"  
}  
geoip {  
source => "SrcIP"  
target => "GeoSrcIP"  
}  
}  
}  
mutate { remove\_field => [ "msg"] }  
}  
output {  
elasticsearch {  
hosts => "elasticsearch:9200"  
user => "elastic"  
password => "changeme"  
}  
}

--------------------------------------------------------------------------------------

input {

beats {

port => "5046"

}

}

output {

if [event.dataset] == "cisco.ios" {

elasticsearch {

hosts => ["http://esnode1.cluster.com:9200","http://esnode2.cluster.com:9200"]

index => "network-%{[event.dataset]}-%{+yyyy.MM.dd}"

user => "elastic"

password => "XXXXXXXXXXXX"

pipeline => "%{[@metadata][pipeline]}"

manage\_template => "false"

ilm\_enabled => "false"

}

}

else if [event.dataset] == "cisco.nexus" {

elasticsearch {

hosts => ["http://esnode1.cluster.com:9200","http://esnode2.cluster.com:9200"]

index => "network-%{[event.dataset]}-%{+yyyy.MM.dd}"

user => "elastic"

password => "XXXXXXXXXXXX"

pipeline => "%{[@metadata][pipeline]}"

manage\_template => "false"

ilm\_enabled => "false"

}

}

else if [event.dataset] == "cisco.asa" {

elasticsearch {

hosts => ["http://esnode1.cluster.com:9200","http://esnode2.cluster.com:9200"]

index => "network-%{[event.dataset]}-%{+yyyy.MM.dd}"

user => "elastic"

password => "XXXXXXXXXXXX"

pipeline => "%{[@metadata][pipeline]}"

manage\_template => "false"

ilm\_enabled => "false"

}

}

else if [event.dataset] == "cisco.ftd" {

elasticsearch {

hosts => ["http://esnode1.cluster.com:9200","http://esnode2.cluster.com:9200"]

index => "network-%{[event.dataset]}-%{+yyyy.MM.dd}"

user => "elastic"

password => "XXXXXXXXXXXX"

pipeline => "%{[@metadata][pipeline]}"

manage\_template => "false"

ilm\_enabled => "false"

}

}

else if [event.dataset] == "cef.log" {

elasticsearch {

hosts => ["http://esnode1.cluster.com:9200","http://esnode2.cluster.com:9200"]

index => "network-%{[event.dataset]}-%{+yyyy.MM.dd}"

user => "elastic"

password => "XXXXXXXXXXXX"

pipeline => "%{[@metadata][pipeline]}"

manage\_template => "false"

ilm\_enabled => "false"

}

}

else if [event.dataset] == "panw.panos" {

elasticsearch {

hosts => ["http://esnode1.cluster.com:9200","http://esnode2.cluster.com:9200"]

index => "network-%{[event.dataset]}-%{+yyyy.MM.dd}"

user => "elastic"

password => "XXXXXXXXXXXX"

pipeline => "%{[@metadata][pipeline]}"

manage\_template => "false"

ilm\_enabled => "false"

}

}

stdout {codec => rubydebug}

}

## CISCO LOG:

input {

udp {

port => "514"

type => "syslog-udp-cisco"

}

}

filter {

grok {

#Remember, the syslog message looks like this : <190>%LINK-I-Up: gi1/0/13

match => { "message" => "^<%{POSINT:syslog\_facility}>%%{DATA:cisco\_code}: %{GREEDYDATA:syslog\_message}" }

}

}

output {

elasticsearch {

hosts => ["http://127.0.0.1:9200"]

index => "cisco-switches-%{+YYYY.MM.dd}"

}

}