Graylog to CrowdSec

Background

Crowdsec's architecture allows running several agents, each parsing the local logs on the server it's running, and sending events to a local API. While this approach works and is flexible, it might not be the most efficient. In my case, all my servers are already sending their logs to a Graylog instance. Running one crowdsec agent on all of those VM would be a waste :

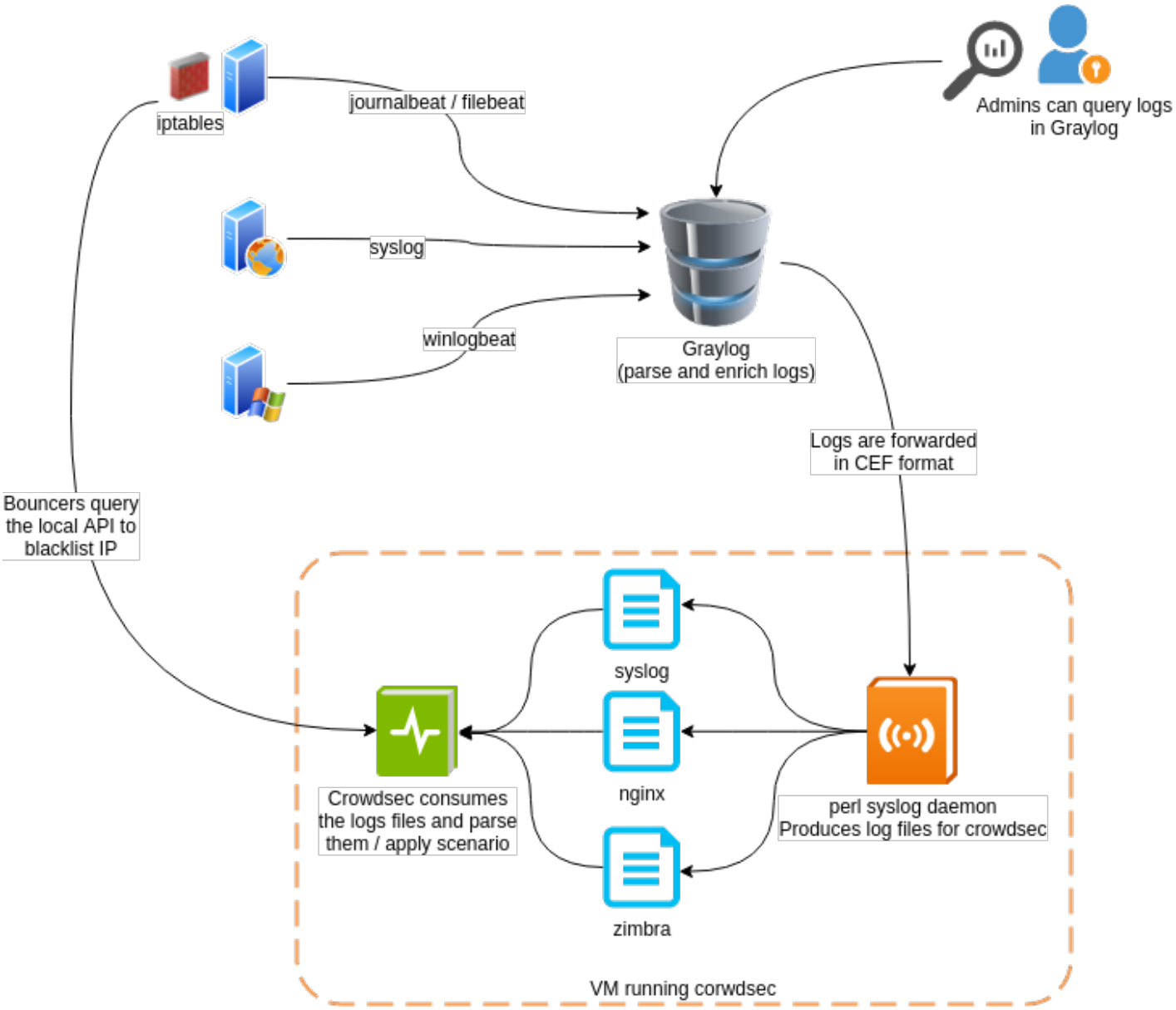
- I really like the idea behind the Journal (systemd-journald), it's very conveniant. But it has a major drawback : it's slow has hell ! Better have everything on SSD, or reading the journal will slows everything down. As I already have journalbeat collecting logs from the Journal, I prefer not adding another Journal reader, which will slows things down even further
- Let's assume we have 40 VM on which we'd like crowdsec agent running. This means something like 40x80MB = 3.2GB of RAM, just for crowdsec
- I have to manage crowdsec conf on those 40 VM. Of course, ansible to the rescue, it wouldn't require manual config everywhere, but I'd still have to setup which logs to parse on which VM, which scenario to apply etc.

So, I looked for an alternative setup, and here's what I came with

Send logs from Graylog to ?

As I already have all my logs in Graylog, it'd be better to send this stream of logs to a single crowdsec installation. But, for now, crowdsec doesn't have network logs input, it can only reads files and the Journal (I've opened a ticket for this). So, the idea is to somehow forward the logs I want from Graylog to a small daemon, which would write logs for crowdsec to consume.

Here's the global flow



g2cs

I wrote a small perl daemon, named g2cs (Graylog to Crowdsec). It's available here. It'll simply listen on a UDP port, waiting for messages to consume from Graylog. It assumes the logs are sent using the CEF format (so, this is the format we'll choose later, for Graylog output). Using a structured log format between Graylog and g2cs allows some filtering in g2cs (for example, to recognize nginx log and put them in a dedicated file, separated from the general syslog). This daemon is very simple

```
perl g2cs.pl --port 514 --logdir /tmp/crowdsec/ --maxlines 20000
```

- Port is the port number g2cs will listen on
- logdir is where it'll write logs for crowdsec to consume. Inside this logdir, g2cs will create

sub directories, like :
  - syslog.log
  - nginx
    - access.log
    - error.log
  - httpd
    - access.log
    - error.log
  - zimbra
    - mailbox.log
- maxlines is the number of lines each file will get before being truncated. As those log files are only to feed crowdsec, the g2cs daemon will truncate them if they reach this number of lines, so they do not grow indefinitely

> You can choose a directory on a tmpfs filesystem to improve performance, as you do not need those logs to be persistent

> We could probably do the same thing with rsyslog, but its configuration is arcane to me, so, it was easier to write the small g2cs daemon instead

Configure crowdsec

Now that we have our g2cs daemon running, you can configure crowdsec acquisition to read these files. Something like

```
---
filenames:
- /run/g2cs/logs/syslog.log
labels:
    type: syslog


---
filenames:
- /run/g2cs/logs/nginx/access.log
- /run/g2cs/logs/nginx/error.log
labels:
    type: nginx


---
```

```
filenames:
- /run/g2cs/logs/httpd/access.log
- /run/g2cs/logs/httpd/error.log
labels:
    type: apache2

---
filenames:
- /run/g2cs/logs/zimbra/mailbox.log
labels:
    type: zimbra
```

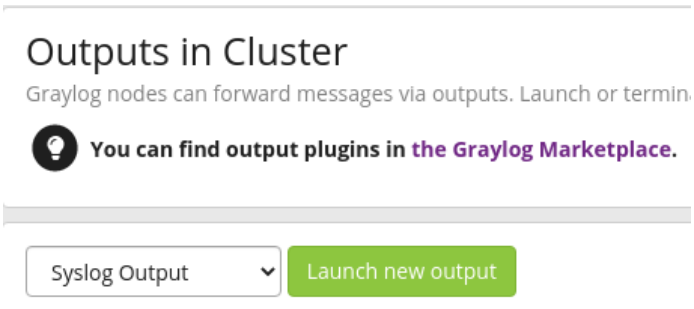
Install the syslog output plugin on Graylog

OK, now that we have crowdsec and g2cs ready, we need to send our logs from Graylog to g2cs. For this, we'll use the syslog output plugin. Just download the jar from github, place it in your Graylog plugin dir (this depends on how you have installed graylog), and restart graylog-server.

Create a syslog output

Now in Graylog, you can create a new output. Go in System → Outputs. Select the “Syslog output” and click launch new output

Image not found or type unknown



Now, configure your Syslog output like this :

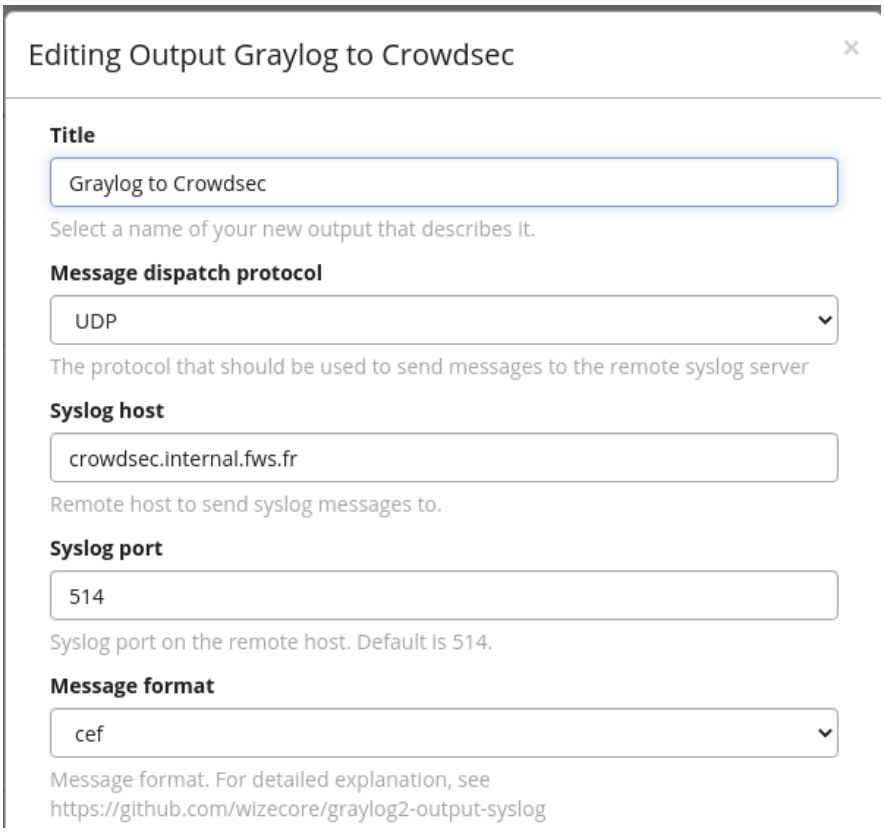


Image not found or type unknown

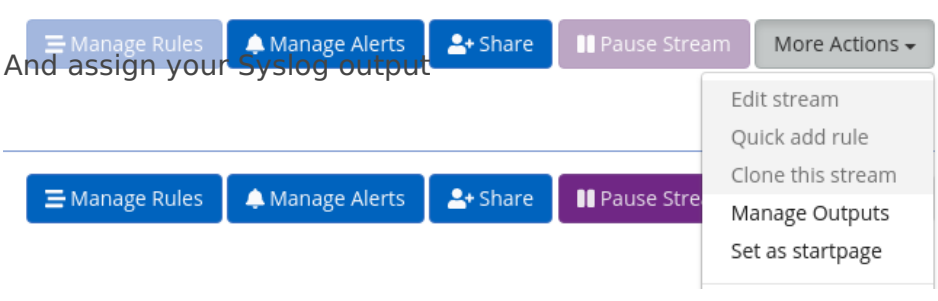
- Choose the UDP protocol
- Enter the DNS name or IP address of your server running g2cs
- Choose the port on which g2cs bind
- Choose the CEF message format

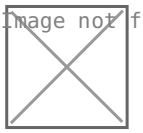
Assign output to streams

Now, you can assign in Graylog your new output to the streams you want. Go in the Stream menu, then, “Manage outputs”

Image not found or type unknown

And assign your Syslog output





You should now see logs flowing from Graylog, to crowdsec. I'm using this on a small graylog setup, ingesting about 400msg/sec, out of which ~200msg/sec are parsed by my single crowdsec install. I just have to install the bouncers where I want to react to all the agressive IP collected on all my servers.

Run crowdsec with less privileges

Bonus point : as crowdsec only access logs from the g2cs daemon, you can run both as a less privileged user, instead of root. First, create an unprivileged user

```
useradd -r -s /sbin/nologin g2cs
```

Now adapt the systemd unit for crowdsec, eg in **/etc/systemd/system/crowdsec.service.d/user.conf**

```
[Service]
User=g2cs
Group=g2cs
```

And create a systemd unit for g2cs itself, **/etc/systemd/system/g2cs.service**

```
[Unit]
Description=Graylog to Crowdsec syslog daemon
After=syslog.target

[Service]
Type=simple
ExecStart=/usr/local/bin/g2cs --port=514 --logdir=/run/g2cs/logs
User=g2cs
Group=g2cs
Restart=always
PrivateTmp=yes
PrivateDevices=yes
ProtectSystem=full
ProtectHome=yes
NoNewPrivileges=yes
SyslogIdentifier=g2cs
```

```
# Allow binding on privileged ports
CapabilityBoundingSet=CAP_NET_BIND_SERVICE
AmbientCapabilities=CAP_NET_BIND_SERVICE


[Install]
WantedBy=multi-user.target
```

Révision #8
Créé 17 mars 2021 22:04:05 par Daniel Berteaud
Mis à jour 2 septembre 2021 15:30:06 par Daniel Berteaud