Viewing TTY Logs with Lighttpd

This section provides the steps to convert the ttylogs captured by the DShield sensor into an HTML formatted document that can be accessed from Kibana interface.

Enable tty logging in each DShield sensors:

guy@picollector: sudo vi /srv/cowrie/cowrie.cfg

Search for ttylog = false change to: *ttylog = true*

Default ttylog → /srv/cowrie/var/lib/cowrie/tty

$ sudo systemctl restart isc-agent

$ sudo /srv/dshield/status.sh

Install txt2html to convert the ttylogs to an html format to have to ability to access them from ELK

$ sudo apt-get install txt2html

Install lighttpd in the ELK server

$ sudo apt-get install lighttpd

$ sudo systemctl enable lighttpd

$ sudo systemctl start lighttpd

Log location: /var/www/html

$ ls -la /var/www/html

Confirm the webserver is accessible and running then move the default index out of the default directory

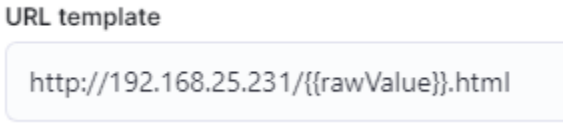
Using browser access: http://elk

$ sudo mv /var/www/html/index.lighttpd.html .

Configure TTYLog in Kibana

Management -> Kibana -> Data Views -> cowrie*

Edit TTYLog and change the IP 192.168.25.231 to your ELK server IP or Name



In ELK server, create SSH Shared Keys and don't put a password:

$ ssh-keygen

Copy id_rsa.pub over to each sensor(s). Likely the easiest way to copy the public key over might be to scp from DShield sensor

$ scp guy@192.168.25.231:/home/guy/.ssh/id_rsa.pub .

$ cat id_rsa.pub >> .ssh/authorized_keys

$ rm id_rsa.pub

Test SSH between the ELK server to DShield sensor

$ ssh -l guy 192.168.25.165 -p 12222

Setup DShield Sensor for TTYLogs to HTML Format

There are 2 scripts provided for the sensor

- replayttylog.sh → Used when setting up TTYLogs parsing for the first time
- ttylog.sh → Runs on a hourly cronjob

If the sensor has been running for a while, un `replayttylog.sh` until it is completed. This could take a while depending on the number of logs is currently stored. The converted hashes will be stored in the home directory in ~/ttylog.

$ mkdir scripts

$ cd scripts

$ curl https://raw.githubusercontent.com/bruneaug/DShield-SIEM/main/sensor_scripts/replayttylog.sh -o replayttylog.sh

$ curl https://raw.githubusercontent.com/bruneaug/DShield-SIEM/main/sensor_scripts/ttylog.sh -o ttylog.sh

$ chmod 755 *.sh

$ ~/scripts/replayttylog.sh

When the script finish to run, it is time to manually transfer the logs to the Elastic server.

Setup Elastic Server for TTYLogs Transfer

The script has completed the log conversion to HTML, the Elastic server should be ready to receive the logs (we assume the webserver is setup as per above) and manually transfer them via scp to ELK:

$ sudo scp -P 12222 guy@192.168.25.165:~/ttylog/* /var/www/httpd

The last step is to enable the hourly cronjob on the server to convert and transfer the TTYLogs over to the web server.

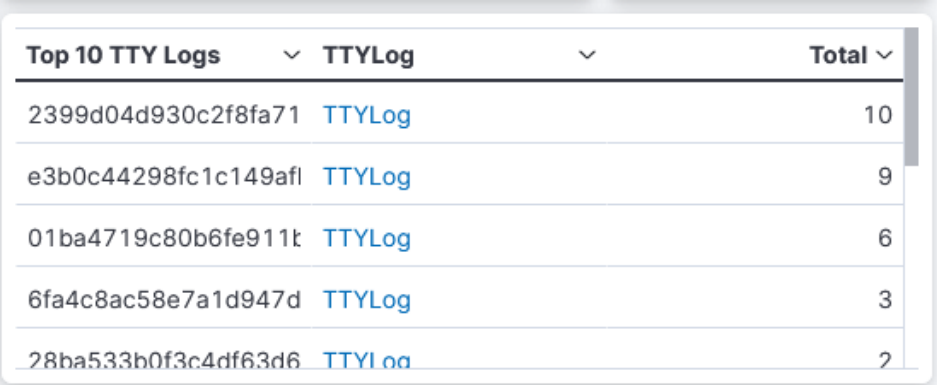
$ crontab -e

1 * * * * /root/scripts/parsing_tty.sh > /dev/null 2>1&

This complete converting to HTML the ttylogs and having them available from the ELK server.

Accessing TTYLogs from Kibana

In Kibana, lookf for the Top 10 TTY Logs summary and just select TTYLog to go to the webserver page containing the log.



Top 10 TTY Logs	TTYLog	Total
2399d04d930c2f8fa71	TTYLog	10
e3b0c44298fc1c149afl	TTYLog	9
01ba4719c80b6fe911t	TTYLog	6
6fa4c8ac58e7a1d947d	TTYLog	3
28ba533b0f3c4df63d6	TTYLog	2

This is an example of the output: