Writing to and rotating custom log files

smp, 2010-27-05

Sometimes I need to log information from iRules to debug something. So I add a simple log statement, like this:

```
when HTTP_REQUEST {
 if { [HTTP::uri] equals "/secure" } {
  log local0. "[IP::remote_addr] attempted to access /secure"
 }
}
```

This is fine, but it clutters up the /var/log/ltm log file. Ideally I want to log this information into a separate log file. To accomplish this, I first change the log statement to incorporate a custom string - I chose the string "##":

```
when HTTP_REQUEST {
 if { [HTTP::uri] equals "/secure" } {
  log local0. "##[IP::remote_addr] attempted to access /secure"
 }
}
```

Now I have to customize syslog to catch this string, and send it somewhere other than /var/log/ltm. I do this by customizing syslog with an include statement:

```
tmsh modify sys syslog include '"
filter f_local0 {
   facility(local0) and not match(\": ##\");
};

filter f_local0_customlog {
   facility(local0) and match(\": ##\");
};

destination d_customlog {
   file(\"/var/log/customlog\" create_dirs(yes));
};

log {
   source(local);
   filter(f_local0_customlog);
   destination(d_customlog);
};
"'
```

save the configuration change:

```
tmsh save / sys config
```

and restarting the syslog-ng service:

```
tmsh restart sys service syslog-ng
```

The included "f_local0" filter overrides the built-in "f_local0" syslog-ng filter, since the include statement will be the last one to load. The "not match" statement is regex which will prevent any statement containing a “##” string from being written to the /var/log/ltm log. The next filter, "f_local0_customlog", catches the "##" log statement and the remaining include statements handle the job of sending them to a new destination which is a file I chose to name "/var/log/customlog".

You may be asking yourself why I chose to match the string ": ##" instead of just "##". It turns out that specifying just "##" also catches AUDIT log entries which (in my configuration) are written every time an iRule with the string "##" is modified. But only the log statement from the actual iRule itself will contain the ": ##" string. This slight tweak keeps those two entries separated from each other.

So now I have a way to force my iRule logging statements to a custom log file. This is great, but how do I incorporate this custom log file into the log rotation scheme like most other log files? The answer is with a logrotate include statement:

```
tmsh modify sys log-rotate syslog-include '"
/var/log/customlog {
   compress
   missingok
   notifempty
}"'
```

and save the configuration change:

```
tmsh save / sys config
```

Logrotate is kicked off by cron, and the change should get picked up the next time it is scheduled to run.

And that's it. I now have a way to force iRule log statements to a custom log file which is rotated just like every other log file. It’s important to note that you must save the configuration with "**tmsh save / sys config**" whenever you execute an include statement. If you don't, your changes will be lost then next time your configuration is loaded. That's why I think this solution is so great - it's visible in the bigip_sys.conf file - not like customizing configuration files directly. And it's portable.

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | f5.com

F5 Networks, Inc. Corporate Headquarters info@f5.com	F5 Networks Asia-Pacific apacinfo@f5.com	F5 Networks Ltd. Europe/Middle-East/Africa emeainfo@f5.com	F5 Networks Japan K.K. f5j-info@f5.com