heka-cef Documentation

Release

Victor Ng

December 02, 2013

Contents

heka-py-cef is a plugin extension for heka-py. heka-py-cef provides an extension to log send CEF messages to a heka server.

More information about how Mozilla Services is using heka (including what is being used for a router and what endpoints are in use / planning to be used) can be found on the Read The Docs page.

CHAPTER 1

Configuration

Configuration is normally handled through Heka's configuration system using INI configuration files. A CEF plugin must use the *heka_cef.cef_plugin:config_plugin* as the provider of the plugin. The suffix of the configuration section name is used to set the method name on the Heka client. Any part after *heka_plugin_* will be used as the method name.

In the following example, we will bind a method *cef* into the Heka client where we will allow network messages to be sent to the Heka server.

```
[heka_plugin_cef]
provider=heka_cef.cef_plugin:config_plugin
```

The CEF plugin provides some optional configuration settings for setting the syslog facility, syslog ident and syslog priority.

By default, the syslog facility will be set to LOCAL4.

Valid facility settings are :

- KERN
- USER
- MAIL
- DAEMON
- AUTH
- LPR
- NEWS
- UUCP
- CRON
- LOCAL0
- LOCAL1
- LOCAL2
- LOCAL3
- LOCAL4
- LOCAL5

- LOCAL6
- LOCAL7

Valid priority settings are :

- EMERG
- ALERT
- CRIT
- ERR
- WARNING
- NOTICE
- INFO
- DEBUG

Syslog options are not supported as they do not make sense in the context of running a hekad daemon. The PID is always captured in a Heka message in the PID field.

Here is one sample configuration demonstrating using all available configuration keys

```
[heka_plugin_cef]
provider=heka_cef.cef_plugin:config_plugin
syslog_facility=KERN
syslog_ident=my_funny_app
syslog_priority=EMERG
```

CHAPTER 2

Usage

Obtaining a client can be done in multiple ways, please refer to the heka documentation for complete details.

That said, if you are impatient you can obtain a client using *get_client*. We strongly suggest you do not do this though.

```
from heka.holder import get_client
```

Logging CEF records is similar to using the raw CEF library. Constants from the *cef* library have been exported in the *heka_cef* module.

For existing code that uses the *cef* library, you will use the *cef* method of the heka client. Your code will change from this

```
from cef import log_cef, AUTH_FAILURE

...

log_cef("Authentication attemped without username", 5,
        request.environ, request.registry.settings,
        "", signature=AUTH_FAILURE)
```

to this

```
from heka.holder impot get_client
import heka_cef

...

client = get_client('heka_cef')
client.cef("Authentication attemped without username", 5,
        request.environ, request.registry.settings,
        "", signature=heka_cef.AUTH_FAILURE)
```

Note that the CEF plugin has exported important constants into the *heka_cef* module.

Constants exported are:

- AUTH_FAILURE
- CAPTCHA_FAILURE
- OVERRIDE_FAILURE
- ACCOUNT_LOCKED

- PASSWD_RESET_CLR

See the cef library for details on each of the constants.

CHAPTER 3

Detailed message layout

A complete CEF message is written out into the payload section of the heka message.

CEF metadata including syslog priority, syslog ident, and syslog facilty are passed as string fields in the Heka message.

The following shows a capture of an example CEF message being captured by hekad.

```
2013/09/23 12:15:09 <
    Timestamp: 2013-09-23 12:15:09.134116864 -0400 EDT
    Type: cef
    Hostname: Victors-MacBook-Air.local
    Pid: 80776
    UUID: 95833933-db90-515f-9c43-469733c560e4
    Logger:
    Payload: Sep 23 12:15:09 Victors-MacBook-Air.local CEF:0|mozilla|weave|3|xx\|x|xx\|x|5|cs1Label=
    EnvVersion: 0.8
    Severity: 6
    Fields: [name:"cef_meta.syslog_priority" value_type:STRING representation:"" value_string:"EMERG"
             name:"cef_meta.syslog_ident" value_type:STRING representation:"" value_string:"my_funny_
             name:"cef_meta.syslog_facility" value_type:STRING representation:"" value_string:"KERN"]
```

CHAPTER 4

API documentation

CHAPTER 5

Indices and tables

- *genindex*
- *modindex*
- *search*