

6. Other Topics

The Event Log

Cobalt Strike operators and scripts communicate global events to the shared event log. Aggressor Scripts may respond to this information too. The event log events begin with `event_`. To list for global notifications, use the `event_notify` hook.

```
on event_notify {  
    println("I see: $1");  
}
```

To post a message to the shared event log, use the `&say` (`functions.html#say`) function.

```
say("Hello World");
```

To post a major event or notification (not necessarily chit-chat), use the `&elog` (`functions.html#elog`) function. The deconfliction server will automatically timestamp and store this information. This information will also show up in Cobalt Strike's Activity Report.

```
elog("system shutdown initiated");
```

Timers

If you'd like to execute a task periodically, then you should use one of Aggressor Script's timer events. These events are `heartbeat_X`, where `X` is 1s, 5s, 10s, 15s, 30s, 1m, 5m, 10m, 15m, 20m, 30m, or 60m.

```
on heartbeat_10s {  
    println("I happen every 10 seconds");  
}
```

Dialogs

Aggressor Script provides several functions to present and request information from the user. Use `&show_message` (`functions.html#show_message`) to prompt the user with a message. Use `&show_error` (`functions.html#show_error`) to prompt the user with an error.

```
bind Ctrl+M {  
    show_message("I am a message!");  
}
```

Use `&prompt_text` ([functions.html#prompt_text](#)) to create a dialog that asks the user for text input.

```
prompt_text("What is your name?", "Joe Smith", {  
    show_message("Please $1 $+ , pleased to meet you");  
});
```

The `&prompt_confirm` ([functions.html#prompt_confirm](#)) function is similar to `&prompt_text` ([functions.html#prompt_text](#)), but instead it asks a yes/no question.