How to Parse the Barracuda Email Security Gateway Syslog

https://campus.barracuda.com/doc/27459873/

For Programmers: Parsing the Barracuda Syslog

For general information about using the syslog, see Syslog and the Barracuda Email Security Gateway.

Syslog messages generated by the Barracuda Email Security Gateway can be parsed for reporting purposes or for building of a custom message log. It is easiest to think of each syslog line in terms of the main components, and the INFO portion can then be parsed based on that service.

The following Perl code illustrates a simple parsing of the log lines. It takes a line and places the resulting message information into a hash – pushing that hash onto a global array of messages when it completes.

```
sub parse_log_line

{

    # Grab the line we were given and create a new message hash for our message

    my($line) = @_;

    my %message = ();



    # These are the components we may have parsed out of the message based on the service

    my ($ip, $id, $start_time, $end_time, $name, $info, $domain);

    my ($enc, $sender, $recip, $score, $action, $reason, $reason_extra, $subject);



    # Grab the main components from the line (IP, MSG_ID, START_TIME, END_TIME, SERVICE, INFO)
```

```
#
#
# NOTE: If this is for the SEND log line then the IP, as well as the START/END times are
# bogus values of 127.0.0.1 and 0/0 respectively
if( $line =~ /\s+:\s+([^\s]+) ([^\s]+) (\d+) (\d+) (RECV|SCAN|SEND) (.*)$/)
{
   # Grab the main pieces of the log entry and the process specific info
   ($ip, $id, $start_time, $end_time, $name, $info) = ($1, $2, $3, $4, $5, $6);

   # Set the connecting IP, message-id, start-time, and end-time if this wasn't
   # for the SEND service
   if( $name !~ /SEND/ )
   {
      $message{client} = $ip;
      $message{id} = $id;
      $message{start_time} = $start_time;
      $message{end_time} = $end_time;
   }

   # Break out the process specific pieces from the info portion
```

```
if( $name =~ /RECV/ )
{
   # Break the MTA info up into sender/recip/action/reason/reason_extra
   if( $info =~ /([^\s]+)\s([^\s]+)\s(\d+)\s(\d+)\s(.*)$/ )
   {
      ($sender, $recip, $action, $reason, $reason_extra) = ($1, $2, $3, $4, $5);

      # Store the readable time of this message based on when it was started by
      # converting the unix time to its components and then sprintf'ing into readable form
      my ($sec,$min,$hour,$mday,$mon,$year,$wday,$yday,$isdst) = localtime($start_time);
      $message{time} = sprintf("%02d/%02d/%02d %02d:%02d:%02d", $mon+1, $mday,
$year-100, $hour, $min, $sec);

      # Store the sender if we had one
      if( $sender ne '-' )
      {
         $message{from} = $sender;
      }

      # Store the recipient if we had one
      if( $recip ne '-' )
```

```
        {
           $message{mailto} = $recip;
        }

        # Set our action/reason codes
        $message{action_id} = $action;
        $message{reason_id} = $reason;

        # Pull in the reason_extra field. This should never be anything other
        # than ASCII since the mta doesn't have any multi-byte functionality
        # ... thus we don't need to eval it.
        if( $reason_extra ne '-' )
        {
           $message{reason_extra} = " ($reason_extra)";
        }
     }
  }
  elsif( $name =~ /SCAN/ )
  {
     # Break the scanner info up into
encrypted/sender/recip/score/action/reason/reason_extra/subject
```

```
if( $info =~ /([^\s]+)\s([^\s]+)\s([^\s]+)\s([-\.\d+]+)\s(\d+)\s(\d+)\s(.*)\sSUBJ:(.*)$/ )

{



   ($enc, $sender, $recip, $score, $action, $reason, $reason_extra, $subject) =

      ($1, $2, $3, $4, $5, $6, $7, $8);



   # Store the readable time of this message based on when it was started by

   # converting the unix time to its components and then sprintf'ing into readable form

   my ($sec,$min,$hour,$mday,$mon,$year,$wday,$yday,$isdst) = localtime($start_time);

   $message{time} = sprintf("%02d/%02d/%02d %02d:%02d:%02d", $mon+1, $mday,
$year-100, $hour, $min, $sec);



   # Store the sender if we had one

   if( $sender ne '-' )

   {

      $message{from} = $sender;

   }



   # Store the recipient if we had one and build the msg_file path

   if( $recip ne '-' )
```

```
{
   $message{mailto} = $recip;
}

# Set the subject line
if( $subject )
{
   eval
   {
      # Note: if this is encoded you may want to decode it here and that
      # is why this section is in an eval – since nothing guarantees the
      # sender encoded the subject properly.
      $message{subject} = decode( $subject );
   };
}

# Set the score if we had one
if( $score ne '-' )
{
   $message{spam_score} = $score;
```

```
        }

        # Set our action/reason codes
        $message{action_id} = $action;
        $message{reason_id} = $reason;

        # Pull in the reason_extra field. This has the extra info the filter that matched
        # and other things that might be multi-byte so it should probably be eval'd
        eval
        {
           if( $reason_extra ne '-' )
           {
              $message{reason_extra} = decode( $reason_extra );
           }
        }
     }
  }
  elsif( $name =~ /SEND/ )
  {
     # Break the Outbound MTA info up into encrypted/action/queue_id/response
```

```
        if( $info =~ /([^\s]+)\s(\d+)\s([^\s]+)\s(.*)$/ )
        {
            my ($enc, $action, $queue_id, $reason) = ($1, $2, $3, $4);

            # Do whatever you would like with the delivery transactions – just keep in
            # mind that a single message may have multiple outbound entries because of
            # being deferred by the downstream server.
        }
    }

    # Put a ref to this message onto our array of messages so we can use it later
    push(@message_list, \%message);

    # Send back whatever info you would like to the caller here. In this case
    # we are sending back the end time as an example that could handle tracking
    # last seen message time or something similar
    return( $end_time );
}

# No message info to send back
```

```
    return undef;

}
```

For questions after reading this document, please contact Barracuda Networks Technical Support.