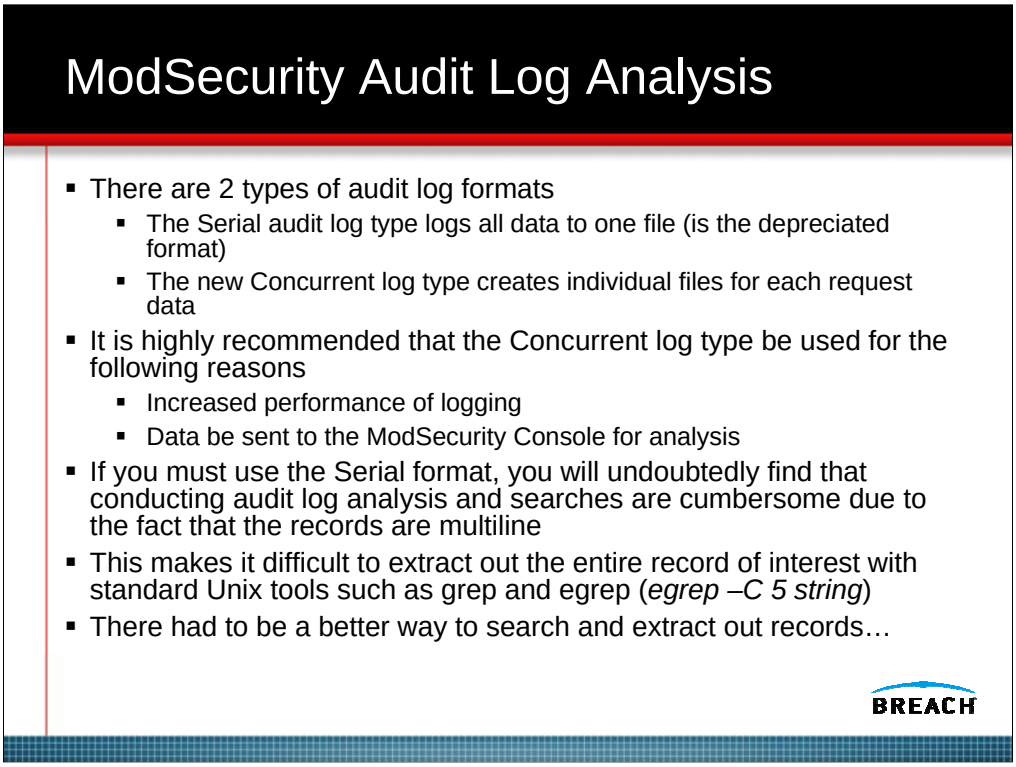
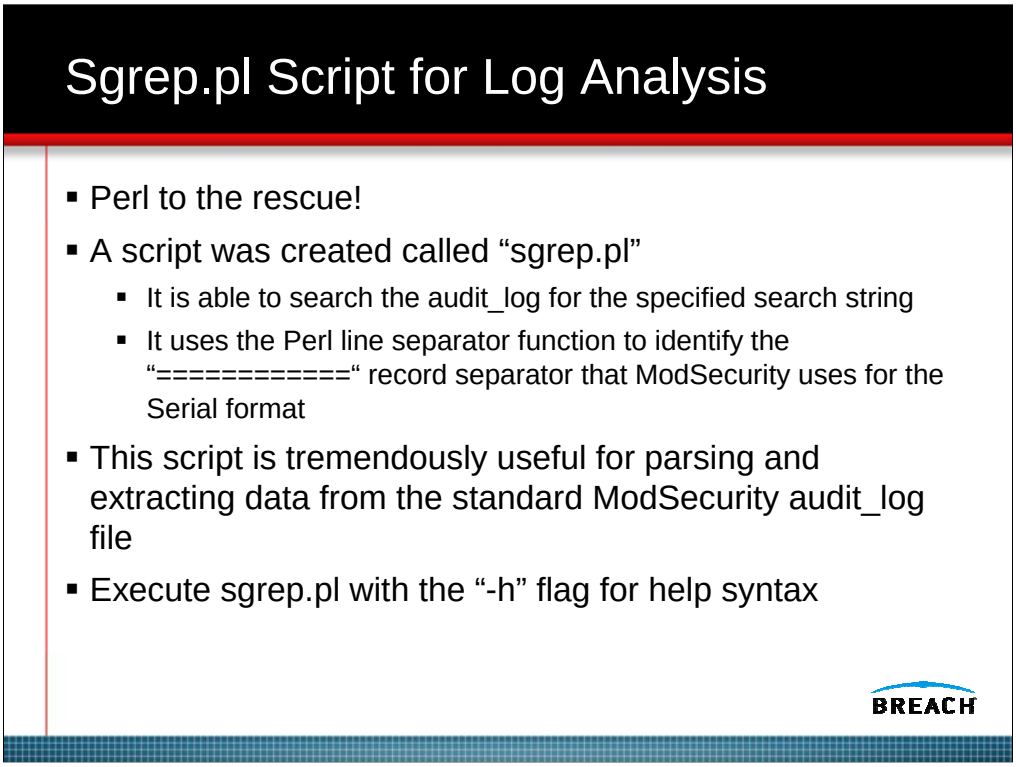


ModSecurity currently supports 2 logging formats – Serial and Concurrent with the later being the most currently and preferred method. The data provided by the ModSecurity audit_log file is invaluable when conducting incident response for web attacks. The data represents the entire client request including the client headers and data payloads. While the value of this data is undeniable, there are some practical issues to deal with if this data is to be used in an efficient manner. For those of you who have been using ModSecurity for some time, you have almost certainly run into this issue that I am about to discuss.

While using grep/egrep searches does show more data from the log entry, it does not perform well in bulk searches as there are never a consistent number of client headers being sent, so the accuracy of the returned data is a bit off. What I needed was a script that functioned similarly to egrep; however, it would be able to understand the ModSecurity audit_log format and be able to extract out the multi-line record.



This lead to the creation of a script called sgrep.pl. Sgrep is a PERL script that accomplishes the tasks that we need to successfully parse and extract entire audit_log records that contain the desired search text string.

Sgrep.pl Help Menu

```
# ./sgrep.pl –h
This program does...
usage: $0 [-hf:s:v:]
    if a file is compressed then it will be uncompressed on the
  fly

    default   : display usage
    -f file   : file to search through
    -s string : string to match on - enclosed in quotes if it
                contains spaces
    -v level  : verbose output
    -h        : this (help) message

example:
         $0
         $0 -f
         $0 -s
         $0 -v level  Verbose/Debug messages, where level = 0..9
```

BREACH

Here is the help menu for the sgrep.pl script.

Sgrep.pl Usage Example

```
# /tools/sgrep.pl -f audit_log -s "200.189.60.251"
=======================================
Request: 200.189.60.251 - - [18/May/2005:23:58:17 --0400] "GET /awstats/awstats
.pl?configdir=|echo%20;echo%20;id;echo%20;echo| HTTP/1.0" 403 743
Handler: cgi-script
---------------------------------------
GET /awstats/awstats.pl?configdir=|echo%20;echo%20;id;echo%20;echo| HTTP/1.0
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/vnd.ms-
   powerpoint, application/vnd.ms-excel, application/msword, */*
Accept-Language: en-us
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0b; Windows NT 5.0)
Host: www.companyx.com
Via: 1.0 lilith.persistelecom.com.br:3128 (squid/2.5.STABLE8)
X-Forwarded-For: 10.1.1.9
Cache-Control: max-age=259200
Connection: keep-alive
ModSecurity-message: Access denied with code 403. Pattern match "\;id" at THE_REQUEST
ModSecurity-action: 403

HTTP/1.0 403 Forbidden
Connection: close
```

BREACH

As you can see, sgrep.pl was able to extract out the entire record for this awstats command injection attempt from the specified IP address 200.189.60.251.

Pipelining Sgrep Commands

```
# ./sgrep.pl -f audit_log -s "ccbill" | ./sgrep.pl -f - -s 68.16.164.147
=======================================
Request: 68.16.164.147 - - [Thu Mar 11 15:29:49 2004] "GET
  http://www.access.ccbill.com/jettis/add-passwd-old.cgi HTTP/1.0" 500 454
Handler: proxy-server
Error: The proxy server could not handle the request <EM><A
  HREF="http://www.access.ccbill.com/jettis/add-passwd-
  old.cgi">GET http://www.access.ccbill.com/jettis/add-passwd-
  old.cgi</A></EM>.<P>
Reason: <STRONG>Host not found</STRONG>
---------------------------------------
GET http://www.access.ccbill.com/jettis/add-passwd-old.cgi HTTP/1.0
Cache-Control: no-cache
Connection: close
Host: www.access.ccbill.com
Pragma: no-cache
Proxy-Connection: keep-alive
Referer: http://www.access.ccbill.com/jettis/add-passwd-old.cg
```

BREACH

Sometimes, the search string used with Sgrep produces too many results. In order to narrow down the results and/or make a more complex search, you can used multiple sgrep.pl searches together. It is possible to take the output of one sgrep.pl search and use it as the input to another search. This is accomplished by using the Unix pipe character “|” to join 2 separate sgrep.pl commands together. On the second sgrep.pl command, you must specify a dash “-” character to the “-f” file flag. This tells Perl to use standard input instead of reading an actual file.