EVLOG

Linux Event Logging for Enterprise Class Systems

Murat Koç

murat.koc@frontsite.com.tr

&

murat.koc@frontsite.de

Başlıklar

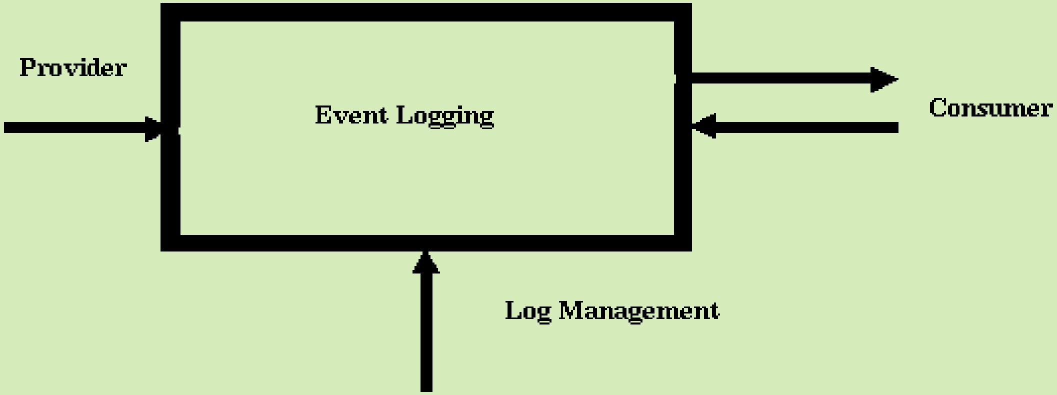
- Sysklogd
- Yapı
- Kurulum
- Test
- Event girdi yapısı
- Event tipleri
- Flag tipleri
- Facility tipleri
- Severity tipleri
- Veri format tipleri
- Kullanılan komutlar

Sysklogd

- printk/klog(kernel events) ve syslog dan oluşuyor.
- Event ler text formatında tutuluyor ve gerekli bazı bilgiler kayıt edilmiyor.
- Sadece ilgilenilen kayıtları görmek için sınırlı sayıda arabirime sahip.
- Sınırlı sayıda event sağlayıcısına sahip(LOCAL0-7)
- Event bildirme yetenekleri sınırlı
- Log dosyalarının yönetilmesi ve limitlenmesi ancak logrotate ile sınırlı bir şekilde yapılabiliyor.

Yapı

- 2 adet birincil arabirim (Provider&Consumer)
- 1 adet ikincil arabirim (Log Management)



Yapı

- ***Provider arabirimi***
- **Event leri bildirmek için kullanılır**
- **Event verisinin, analiz ve loglama için kullanılabilir olması için**
- **belli bir bilgi seti sağlanmak zorundadır.**

Yapı

- *Event Logging*
  - Provider tarafından sağlanan veriye ek olarak topladığı bilgilerle event log girdisini oluşturur.
    - Printk(), syslog(), vsyslog() mesajlarını alarak POSIX-uyumlu event kayıtları
    - şeklinde loglayabilir.
    - Belirli değerlere bağlı olarak, birbiri ardına hızlı bir şekilde olan tekrarlı event
    - lerin kayıtların tutulmasını engelleyebilir.
    - Belirli tanımlamalara bağlı olarak, loglanan event leri gösterebilir.
    - Belirli tanımlamalara bağlı olarak logdan okunan event leri filtereyebilir.
    - Kayıtlı consumer a, consumer tarafından belirtilmiş event lerin oluşması halinde
    - haber verebilir.
    - Event in buffer boyutunu ayarlanmasına olanak sağlar

Yapı

Consumer arabirimi

Log dan belirli tanımlamalara bağlı olarak event verisini alır

Event kayıtlarını standart veya consumer tarafından tanımlanmış formatta görüntüler

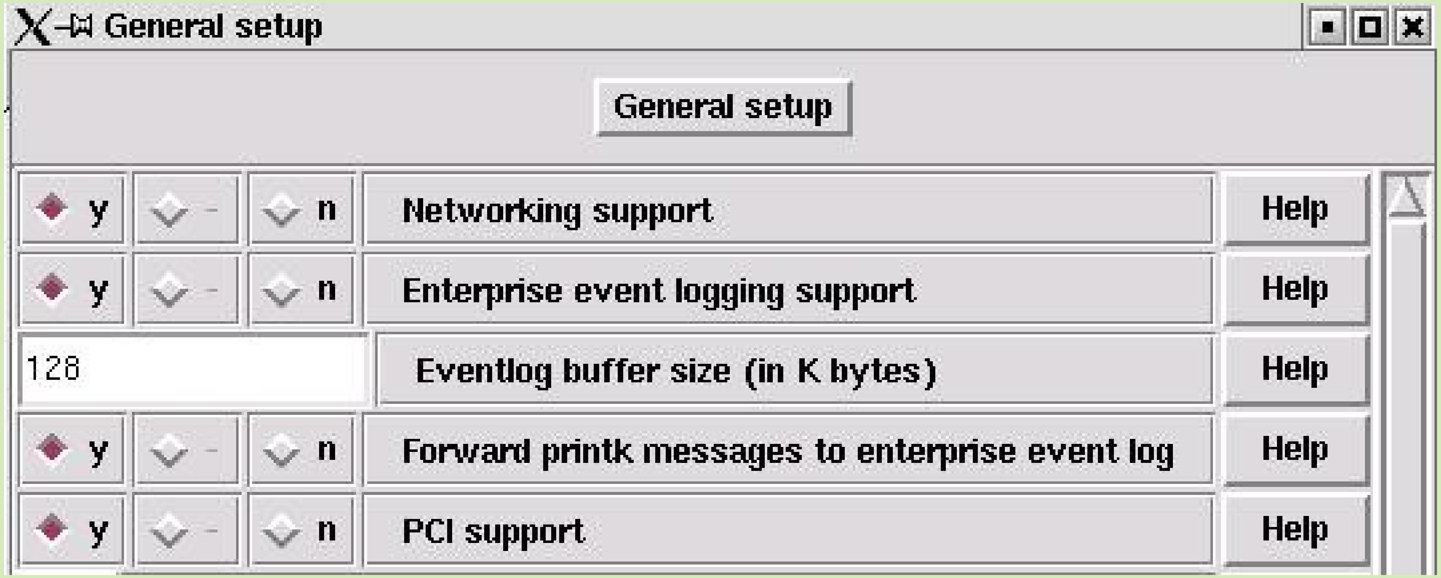
Log a, consumer tarafından tanımlanmış özelliklerde event yazıldığı zaman haberdar etmek için kayıt eder.

Yapı

- *Log Management*
-
- **Event log unun boyutunu yönetmek**
- **Otomatik olarak daha fazla ilginelimeyen log ların silinmesi, alanın düzenlenmesini sağlamak, log u temizlemek**
-
- **için çeşitli yöntemler sağlar**
-

Kurulum

- Kernel için gerekli patch yapılır.
- Desteği aktif hale getirilir.



General setup

General setup

◆ y ◇ - ◇ n Networking support Help

◆ y ◇ - ◇ n Enterprise event logging support Help

128 Eventlog buffer size (in K bytes) Help

◆ y ◇ - ◇ n Forward printk messages to enterprise event log Help

◆ y ◇ - ◇ n PCI support Help

Kurulum

```
#tar xvfz evlog-1.4.2-2.tar.gz
#cd evlog-1.4.2
#make
#make install
```

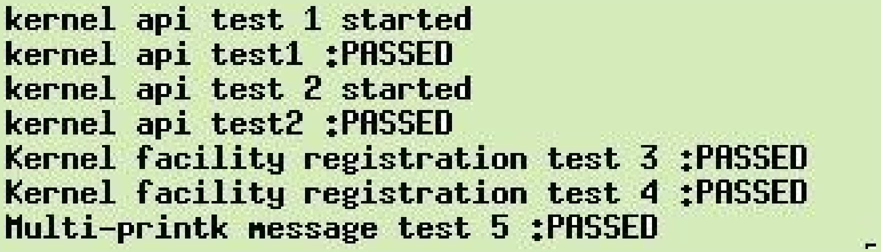
Kernel space testi

```
#cd kernel/test
#make
#./runtests.sh
```

User space testi

```
#cd /var/evlog/test
#./runtests.sh
```

Test – Kernel Space



```
kernel api test 1 started
kernel api test1 :PASSED
kernel api test 2 started
kernel api test2 :PASSED
Kernel facility registration test 3 :PASSED
Kernel facility registration test 4 :PASSED
Multi-printk message test 5 :PASSED
```

```
recid=541, size=80, format=BINARY, event_type=0x7d0, facility=KERN,
severity=ERR, uid=root, gid=root, pid=3852, pgrp=3839,
time=Tue Oct 29 13:45:04 2002, flags=0x2 (KERNEL), thread=0x0, processor=0
00000000 0A 00 00 00 0B 00 00 00  0C 00 00 00 0D 00 00 00 | ........ ........
00000010 0E 00 00 00 0A 00 00 00  0B 00 00 00 0C 00 00 00 | ........ ........
00000020 0D 00 00 00 0E 00 00 00  01 00 02 00 03 00 04 00 | ........ ........
00000030 05 00 6C 6F 76 65 1E 00  00 00 1F 00 00 00 20 00 | ..love.. ...... .
00000040 00 00 21 00 00 00 22 00  00 00 61 62 63 64 66 00 | ..!...". ..abcdf.
```

```
recid=589, size=79, format=STRING, event_type=0x64, facility=test_sneezy,
severity=INFO, uid=root, gid=root, pid=3871, pgrp=3839,
time=Tue Oct 29 13:45:07 2002, flags=0x2 (KERNEL), thread=0x0, processor=0
Registration of facility "test_sneezy" succeeded.  Facility code = 0xdd866b39.
recid=591, size=40, format=STRING, event_type=0x2, facility=KERN,
severity=ALERT, uid=root, gid=root, pid=3880, pgrp=3839,
time=Tue Oct 29 13:45:10 2002, flags=0x22 (KERNEL|PRINTK), thread=0x0,
processor=0
Hey! The disk is on fire! Do something!
```

Test – User Space

```
< Hey!  Something happened at line 71 of defaultTest.c:
---
> Hey!  Something happened at line 67 of defaultTest.c:
13c13
< defaultTest.c:78: This event record has a message string,
    plus another string.
---
> defaultTest.c:75: This event record has a message string,
    plus another string.
17c17
< Hey!  Something happened at line 86 of defaultTest.c:
---
> Hey!  Something happened at line 82 of defaultTest.c:
templates defaultTest :FAILED
templates stattest :PASSED
templates misc test :PASSED
test of degenerate templates/records :PASSED
test of array delimiter :PASSED
test 1 started
test1   :PASSED
-----------------
test2   :PASSED
-----------------
This test is testing for access denied
Access denied.
```

```
Test3   :PASSED
-----------------
test4   :PASSED
----------------
test5   :PASSED
-----------------
test6   :PASSED
-----------------
Action was executed with recid=1121 as an arg
Action was executed with recid=1127 as an arg
Action was executed with recid=1123 as an arg
Action was executed with recid=1129 as an arg
Action was executed with recid=1131 as an arg
Action was executed with recid=1125 as an arg
Action was executed with recid=1133 as an arg
Action was executed with recid=1135 as an arg
Action was executed with recid=1139 as an arg
Action was executed with recid=1137 as an arg

test7   :PASSED
-----------------
testevlog1 :PASSED
testfac1        :PASSED
```

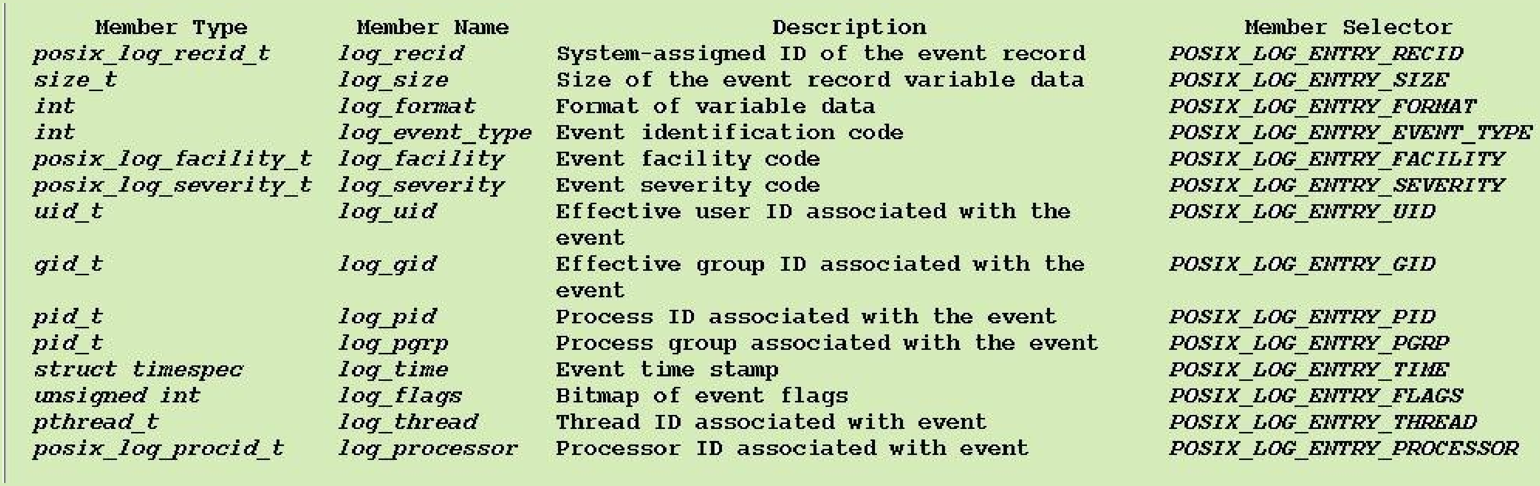
Test – User Space

```
Recid=1147, size=68, format=STRING, event_type=0x1, facility=LOGMGMT,
severity=INFO, uid=root, gid=root, pid=1396, pgrp=1395,
time=Tue Oct 29 13:54:51 2002, flags=0x0, thread=0x400, processor=0
Discarded 3 duplicate events, event_type = 0x7d2, facility = LOCAL0

recid=1149, size=11, format=STRING, event_type=0x64, facility=LOCAL1,
severity=INFO, uid=root, gid=root, pid=4217, pgrp=3988,
time=Tue Oct 29 13:54:51 2002, flags=0x0, thread=0x400, processor=0
some junk3

recid=1151, size=32, format=STRING, event_type=0x65, facility=0x6bebb75e,
severity=INFO, uid=root, gid=root, pid=4269, pgrp=3988,
time=Tue Oct 29 13:54:58 2002, flags=0x0, thread=0x400, processor=0
ROOT guy logs this info message

recid=1153, size=41, format=STRING, event_type=0x65, facility=0x6f154fd,
severity=INFO, uid=root, gid=root, pid=4298, pgrp=3988,
time=Tue Oct 29 13:55:13 2002, flags=0x0, thread=0x400, processor=0
testfac3:ROOT guy logs this info message

recid=1175, size=74, format=STRING, event_type=0x2, facility=LOGMGMT,
severity=NOTICE, uid=root, gid=root, pid=4322, pgrp=3988,
time=Tue Oct 29 13:55:15 2002, flags=0x0, thread=0x400, processor=0
Log compaction on /var/evlog/eventlog starts at Tue Oct 29 13:55:15 2002

recid=1177, size=69, format=STRING, event_type=0x3, facility=LOGMGMT,
severity=NOTICE, uid=root, gid=root, pid=4322, pgrp=3988,
time=Tue Oct 29 13:55:15 2002, flags=0x0, thread=0x400, processor=0
Log compaction on /var/evlog/eventlog ended. 10 events were removed.
```

Event girdi yapısı

- **POSIX 1003.25 tanımı**

Recid=1179, size=155, format=STRING, event_type=0x2, facility=KERN, severity=WARNING, uid=root, gid=root, pid=0, pgrp=0, time=Tue Oct 29 13:56:20 2002, flags=0x32 (KERNEL|PRINTK|INTERRUPT), thread=0x0, processor=0



Member Type	Member Name	Description	Member Selector
posix_log_recid_t	*log_recid*	System-assigned ID of the event record	*POSIX_LOG_ENTRY_RECID*
size_t	*log_size*	Size of the event record variable data	*POSIX_LOG_ENTRY_SIZE*
int	*log_format*	Format of variable data	*POSIX_LOG_ENTRY_FORMAT*
int	*log_event_type*	Event identification code	*POSIX_LOG_ENTRY_EVENT_TYPE*
posix_log_facility_t	*log_facility*	Event facility code	*POSIX_LOG_ENTRY_FACILITY*
posix_log_severity_t	*log_severity*	Event severity code	*POSIX_LOG_ENTRY_SEVERITY*
uid_t	*log_uid*	Effective user ID associated with the event	*POSIX_LOG_ENTRY_UID*
gid_t	*log_gid*	Effective group ID associated with the event	*POSIX_LOG_ENTRY_GID*
pid_t	*log_pid*	Process ID associated with the event	*POSIX_LOG_ENTRY_PID*
pid_t	*log_pgrp*	Process group associated with the event	*POSIX_LOG_ENTRY_PGRP*
struct timespec	*log_time*	Event time stamp	*POSIX_LOG_ENTRY_TIME*
unsigned int	*log_flags*	Bitmap of event flags	*POSIX_LOG_ENTRY_FLAGS*
pthread_t	*log_thread*	Thread ID associated with event	*POSIX_LOG_ENTRY_THREAD*
posix_log_procid_t	*log_processor*	Processor ID associated with event	*POSIX_LOG_ENTRY_PROCESSOR*

Event tipleri

- EVL_PRINTK_MESSAGE (0x2) – event in printk() fonksiyonu tarafından yazıldığını
- belirtir. Bu event tiplerinin log_facility leri daima LOG_KERN dir.
- EVL_SYSLOG_MESSAGE (0x1) – event in syslog() veya vsyslog() ile yazıldığını belirtir.
- log_facility, syslog() u kullanan program tarafından belirtilir.
- EVL_BUFFER_OVERRUN (0x6) – kernel event buffer ın boyutunun üstüne çıkıldğı için
- event veya event lerin gözardı edildiğini belirtir. Bu tip event lerin log_facility leri daima
- LOG_LOGMGMT dir.
- EVL_DUPS_DISCARDED (0x7) – bir veya daha fazla tekrarlı event in gözardı edildiğini
- belirtir.Bu tip event lerin log_facility leri daima LOG_LOGMGMT dir.
- EVLOG_REGISTER_FAC (40) – kernel da evl_register_facility() ile çağrılan özel bir event
- olduğunu belirtir.

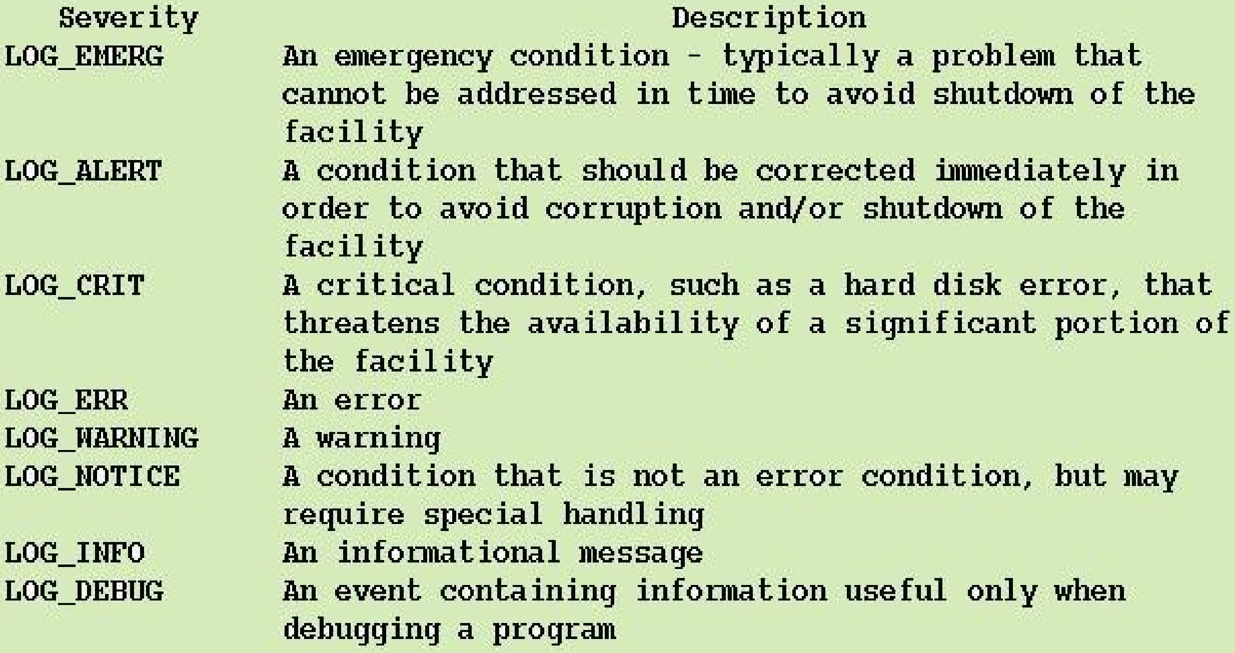
Flag tipleri

- POSIX_LOG_TRUNCATE (0x1) - POSIX standardında tanımlanmıştır
- EVL_KERNEL_EVENT (0x2) – event in kernel içinden loglandığını belirtir.
- EVL_KERNTIME_LOCAL (0x8) – Bu bit log_time değerinin local time a göre
- düzenlendiğini ve GMT e göre evlogd tarafından düzenlenmesi gerektiğini belirtir.
- EVL_INTERRUPT (0x10) – Bu bit event in interrupt durumundan loglandığını
- belirtir.
- EVL_PRINTK_MESSAGE (0x20) – event in printk() ile yazıldığını belirtir.
- 0x40 ve 0x80 ilerideki event loglama kullanımları için ayrılmıştır.

Facility tipleri

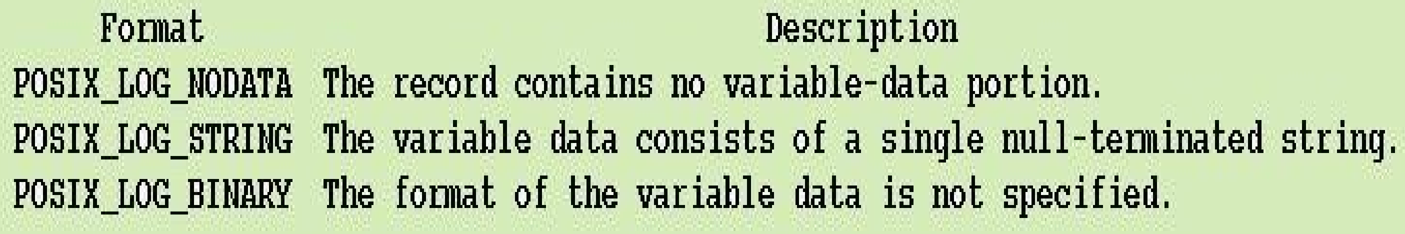
Facility	Description
LOG_AUTH	The authorization system
LOG_CRON	The system that schedules periodic tasks in the system
LOG_DAEMON	Background services that are not explicitly provided facility codes
LOG_KERN	The kernel
LOG_LPR	The printer system
LOG_MAIL	The mail system
LOG_NEWS	The news system
LOG_SYSLOG	The system logging process
LOG_LOGMGMT	The system for managing event logs
LOG_USER	A catch-all facility code for applications that choose not to employ a more descriptive code
LOG_LOCAL0	Reserved for local use
LOG_LOCAL1	Reserved for local use
LOG_LOCAL2	Reserved for local use
LOG_LOCAL3	Reserved for local use
LOG_LOCAL4	Reserved for local use
LOG_LOCAL5	Reserved for local use
LOG_LOCAL6	Reserved for local use
LOG_LOCAL7	Reserved for local use

Severity tipleri



Severity	Description
LOG_EMERG	An emergency condition - typically a problem that cannot be addressed in time to avoid shutdown of the facility
LOG_ALERT	A condition that should be corrected immediately in order to avoid corruption and/or shutdown of the facility
LOG_CRIT	A critical condition, such as a hard disk error, that threatens the availability of a significant portion of the facility
LOG_ERR	An error
LOG_WARNING	A warning
LOG_NOTICE	A condition that is not an error condition, but may require special handling
LOG_INFO	An informational message
LOG_DEBUG	An event containing information useful only when debugging a program

Veri Format tipleri



Format	Description
POSIX_LOG_NODATA	The record contains no variable-data portion.
POSIX_LOG_STRING	The variable data consists of a single null-terminated string.
POSIX_LOG_BINARY	The format of the variable data is not specified.

Kullanılan komutlar

- **evlconfig**
- **evlfacility**
- **evlnotify**
- **evlogmgr**
- **evlsend**
- **evltc**
- **evlview**

Evlconfig

NAME

evlconfig - Configure logging daemon

SYNOPSIS

```
evlconfig  -l | --list
evlconfig  -s | --screen   filter | no filter
evlconfig  -i | --interval seconds
evlconfig  -c | --count events
evlconfig  -d | --discarddups  on | off
evlconfig  -o|  --output   severity-level | off
```

DESCRIPTION

The evlconfig command lets you change the default settings for event logging.

Unless otherwise noted, root permission is required to use the following evlconfig options.

Evlconfig örnek

```
#evlconfig -s uid!=muratkoc
#evlconfig -d on
#evlconfig -i 5
#evlconfig -c 20
#evlconfig -o INFO
```

```
#evlconfig -l
Discard Dups = on
Discard Interval = 5 seconds
Discard Count = 20 identical events
Event Screen:
        uid!=muratkoc
Console display level = INFO
```

Evlfacility

- NAME
- evlfacility - Manage facility registry
- SYNOPSIS
- evlfacility -l|--list
- evlfacility -v|--verify [file]
- evlfacility -r|--replace file
- evlfacility -a|--add facility-name [-k|--kernel] [-p|--private] [-f|--filter filter]
- evlfacility -d|--delete facility-name
- evlfacility -c|--change facility-name [-k|--kernel] | [-u|--user] | [-p|--private] | [-n|-
- -noprivate] | [-f|--filter filter|nofilter]
- DESCRIPTION
- The evlfacility command lets you list contents of the event logging facility
- registry, replace the entire facility registry, add facilities (with options) to the
- facility registry, delete facilities, or modify an existing facility.
- For evlfacility options that modify the facility registry, event logging applications
- already in progress will start using the modified registry within 5 seconds. The
- following options are accepted (unless otherwise noted, root permission is required

Evlfacility örnek

```
#evlfacility -l

0 KERN
8 USER
16 MAIL
24 DAEMON
32 AUTH
40 SYSLOG
48 LPR
56 NEWS
64 UUCP
72 CRON
80 AUTHPRIV     private
88 FTP
96 LOGMGMT
128 LOCAL0
136 LOCAL1
144 LOCAL2
152 LOCAL3
160 LOCAL4
168 LOCAL5
176 LOCAL6
184 LOCAL7
```

```
0xd8c0bf86 test_kfacreg kernel
0xcd14f75 test_bashful kernel
0xeb5232b8 test_doc kernel
0x8c69ff96 test_dopey kernel
0x36c38aa1 test_grumpy kernel
0x531dfca3 test_happy kernel
0x6526797e test_sleepy kernel
0xdd866b39 test_sneezy kernel
```

Evlnotify

- NAME
- Evlnotify – Event Notification
- SYNOPSIS
- evlnotify -l | --list
- evlnotify -a | --add notify-action [-o | --once-only] [-f | --filter filter] [-p | --persistent] [-u | --uid userid]
- evlnotify -d | --delete notify-id ...
- evlnotify -c | --change new-notify-action notify-id
- evlnotify -F | --file cmd-file
- DESCRIPTION
- The evlnotify command lets you register actions to be taken when a specified event occurs. An action (command or shell
- script) will be executed on behalf of the user who registered it, except if that user is root; root has the option of specifying an
- alternate user id (other than root).
- The default environment settings are specified in /etc/evlog.d/action_profile, which can be modified only by root. The action
- command, or script, can alter its environment. The initial default is as follows:
- PATH=/usr/bin:/bin:.
- PWD=/tmp
- A user attempting to issue the evlnotify command must have access to at least one of the log files (/var/evlog/eventlog or
- /var/evlog/privatelog). In addition to verifying that the user has read access to at least one of the log files, an additional check is
- performed using a mechanism similar to the crontab command:
- · If the file /etc/evlog.d/action.allow exists, then the users listed therein can issue this command.
- · Otherwise, if the file /etc/evlog.d/action.deny exists, but the action.allow file does not exist, then users listed in
- /etc/evlog.d/action.deny will not be allowed to issue the evlnotify command, and all others will be allowed to issue the command.
- · If neither the action.allow nor action.deny files exist, then all users can issue this command. Note that notifications can only
- be modified or deleted by the users who created them.

Evlnotify örnek

```
#evlnotify -a /tmp/mail_at -f 'facility=KERN'

#evlnotify -l
17:severity=INFO:uid=muratkoc:root:0:0
18:facility=KERN:/tmp/mail_at:root:0:0

muratkoc@linux:~> /sbin/evlnotify -a /tmp/mail_at -f 'severity=INFO'
muratkoc@linux:~> /sbin/evlnotify -l
19:severity=INFO:/tmp/mail_at:muratkoc:0:0
```

Evlogmgr

NAME

evlogmgr - Event log manager

SYNOPSIS

```
evlogmgr   -c | --compact  filter
 [ -F | --force ] [ -C | --compr-bak ]
 [[ -p | --private  ] | [ -l | --log  srcfile  ]]
evlogmgr   -f | --fix
 [[ -p | --private  ] | [ -l | --log  srcfile  ]]
evlogmgr   -s | --show-status  filter
 [[ -p | --private  ] | [ -l | --log  srcfile  ]]
```

DESCRIPTION

The evlogmgr command performs log management on the event log, on the private log, or optionally, on a log file that you specify. You also specify which events are to be deleted. The space freed by deleted events is reused for undeleted events (a process referred to as compaction) and the log file is truncated, thus reducing its overall size.

You must have root permission to use this command.

Evlogmgr örnek

```
#evlogmgr -c 'severity=DEBUG' -C

recid=1587, size=70, format=STRING, event_type=0x3, facility=LOGMGMT,
severity=NOTICE, uid=root, gid=root, pid=18361, pgrp=18361,
time=Tue Oct 29 20:50:55 2002, flags=0x0, thread=0x400, processor=0
Log compaction on /var/evlog/eventlog ended. 284 events were removed.
```

Evlsend

NAME

evlsend - event generation utility

SYNOPSIS

evlsend

-f | --facility facility -t | --type event_type

[-s | --severity severity] [-m | --message message-string] [-b | --binary attr_type,

attr_value ...]

DESCRIPTION

The evlsend utility lets you send an event message to the system.

Evlsend örnek

- **#evlsend -f MAIL -t 0x32 -s INFO -m "super event yaparim"**
-
- **recid=1641, size=20, format=STRING, event_type=0x32,**
- **facility=MAIL,severity=INFO, uid=root, gid=root, pid=18428,**
- **pgrp=18428, time=Tue 29 Oct 2002 08:57:05 PM EET, flags=0x0,**
- **thread=0x400, processor=0 super event yaparim**
-

Evltc

- NAME
  - evltc - Compile formatting templates
- SYNOPSIS
  - evltc sourcefile [-f | --func] [-n | --noto]
    - [-c | --cpp] [cpp_options]
  - or
  - evltc binfile.to
- DESCRIPTION
  - The evltc command reads the formatting template specification(s) in sourcefile and creates a binary template file for each specification.
- sourcefile may not define two templates with the same event_type even if for different facilities. Binary files are created in the directory where
- sourcefile resides.
  - If the name of the specified file ends in .to, it is assumed to be a binary template file. It is read, and if it contains an event-record template, the
- corresponding sample call to the evl_log_write() function is printed, as with -f. In this case, the -f and -n options (if specified) are ignored, and
- any cpp-related options are flagged as errors.
  - The binary file for an event-record template is named eventtype.to, where eventtype is the decimal event type. If eventtype is negative, the
- minus sign (-) is converted to an equals sign (=). The binary file for a struct template is named structname.to.
  - If the source file contains any errors, no binary files are produced, and error messages are written to stderr.
  - The algorithm for finding a struct template that is referenced by another template is as follows:
  - (1) The struct path from the import statement is converted into a relative pathname by replacing all periods with slashes and appending .to.
- For example, gui.graphics.point becomes gui/graphics/point.to.
  - (2) This relative pathname is applied to each of the following directories in turn until a file is found:
  - (a) the directory in which the current template source file resides.
  - (b) the directories specified in the EVLTMPLPATH environment variable (a colon-delimited list of directories). If the EVLTMPLPATH
- environment variable is not defined, the directory /var/evlog/templates is searched.
  - (3) The selected template file is read into memory if it is not already there.
  - It is neither necessary nor permitted to import a template with the same name as one previously defined in the current template source file,
- unless the imported template is from a different directory. See Example 3 (below).

Evltc örnek

```
/* HEADER SECTION */
facility "LOCAL1";
event_type 0x3115;
/* CONST-ATTRIBUTES SECTION */
const {
        string    repair_action = "Replace SCSI adapter";
}
/* RECORD-ATTRIBUTES SECTION */
attributes {
        char       unit_ser_no[8]    "(%c)";
        ushort     lun               "%u";
        char       sense_bytes[12]   "%t";
/* For the next attribute, the various sections of the format_spec are
 * automatically concatenated into a single string.
 */
        uchar      recovery_stat     "%
                b/0x40/INTERFACE_WAS_RESET/"
          "0x20/RECOVERY_ACTION_STARTED/"
          "0x10/RECOVERY_ACTION_FAILED/";
/
```

```
* The final attribute specification says to display the rest of the
 * bytes in dump format.
 */
       char       extra_data[_R_]   "%t";
}
/* FORMATTING SECTION */
format
SCSI interface error: Adapter Serial Number/LUN = %unit_ser_no%/\
%lun%
\tRecovery Status: %recovery_stat%
\tSense Bytes:
%sense_bytes%
  \tRecommended repair action:
\t\t%repair_action%\n\n
%extra_data%
```

Evltc örnek

```
#evltc -f scsi.t
evl_log_write(LOG_LOCAL1, 12565 /* 0x3115 */, severity, flags,
        "char[]",       8,      unit_ser_no,
        "ushort",        lun,
        "char[]",       12,     sense_bytes,
        "uchar",        recovery_stat,
        "char[]",       _R_,    extra_data,
        "endofdata");
```

```
SCSI interface error: Adapter Serial Number/LUN = XSCSI178/3
Recovery Status: 0x50(INTERFACE_WAS_RESET|RECOVERY_ACTION_FAILED)
Sense Bytes:
00000000 61 62 63 64 65 66 67 68  61 62 63 64          | abcdefgh abcd
Recommended repair action:
Replace SCSI adapter
00000000 26 B3 B3 25 AB BC CD                          | &..%...
```

Evlview

- NAME
  - evlview - View log events
- SYNOPSIS
  - evlview --help
  - OR
  - evlview [input] [output] [-f | --filter filter]
  - [-b | --templates] [-B | --notemplates]
  - input (defaults to /var/evlog/eventlog, or to /var/evlog/privatelog with -p | --private):
  - [-n | --new][-T | --timeout nsec][-R | --recid rid]
  - OR
  - [-l | --log srclogfile] [-t | --tail nrec]
  - [-r | --reverse]
  - output (defaults to stdout):
  - [-o | --out destlogfile]
  - OR
  - [-S | --formatstr format-string] [format_opts]
  - OR
  - [-F | --formatfile format-file] [format_opts]
  - OR
  - [-c | --compact] [-s | --separator sep] [format_opts]
  - OR
  - [-m | --syslog]
  - format_opts:
  - [-N | --newlines n] [-d | --datefmt date-format]
- DESCRIPTION
  - The evlview utility lets you view events from an event log, view events in real time, or read records from an event log and write the
- records to another file.

Evlview örnek

```
#evlview -f 'data contains "eth"'
recid=221, size=77, format=STRING, event_type=0x2, facility=KERN,
severity=INFO, uid=root, gid=root, pid=573, pgrp=451,
time=Tue 29 Oct 2002 01:19:29 PM EET, flags=0x22 (KERNEL|PRINTK),
 thread=0x0, processor=0
eth0: RealTek RTL8139 Fast Ethernet at 0xd2856000, 00:60:67:00:a5:54, IRQ 10


#evlview -f 'facility==MAIL'
recid=1453, size=35, format=STRING, event_type=0x1, facility=MAIL,
severity=NOTICE, uid=root, gid=dialout, pid=18133, pgrp=17580,
time=Tue 29 Oct 2002 08:35:08 PM EET, flags=0x0, thread=0x400, processor=0
 Starting mail and news send/fetch
```

Sorular ?

Teşekkürler….