M-ICE
Modular Intrusion Detection and Countermeasure Environment

Admin Guide

Version 0.4

Changelog

Version	Date	Author	Changes, Problems, Notes
0.1	2003−12−07	Thomas	− initial version
0.2	2003−12−30	Thomas	− added chapter about assumptions
0.3	2004−01−18	Thomas	− added section about alert database − metioned ntp and libidmef RPM files − create user and group needed for daemons
0.4	2004−02−01	Thomas	− added chapter about LAuS

Copyright and Licence Notes

Abstract

This paper will provide you with the essential knowledge to understand and setup the M-ICE system. Several chapters will guide you from the abstract architecture to example configurations of it the M-ICE components. The example systems are running SUSE LINUX 9.0 professional on i386 architecture. [1]

[1] I am not responsible for any damage caused by my software or by this documentation!

Contents

Chapter 1

Assumptions

To make our design work theoretically, we need to create and accept some assumptions. Making assumptions is often used in theoretic science to make a situation cleaner and simpler (remeber mathematic proves). To be honest this project is far beyond theory but some things are too random and too complex, that is the same reason why security certifications, like CAPP [7], need to make assumptions too, to keep them realistic. [1] We have to trust some system entities that we can not fully control norverify but that are essential for the operation of our system. The administrator needs to make sure that these assumptions do not get violated. [2] All trusted system components:

- System Hardware
- Operating-System Kernel
- Super-User (root)
- all Daemons running as root (especially: syslogd, auditd, apache)
- setuid Programs
- LAuS Subsystem
- M-ICE components

For more information about security-certified operating-systems read the security-guide and LAuS documentation delivered with SuSE Linux Enterprise Server 8 + ServicePak3 + certification-eal3*.rpm.

[1]get SuSE Linux Enterprise Server 8 + ServicePak3 to run a certified system with security-level EAL3+

[2]some rules like the trust in the root user may get violated but these violation should be detected by the system itself

Chapter 2

Architecture

Before we go to the details of the systems architecture I should make clear the goals I tried to reach. First I want to provide a system that can be customized easily to fit everybodys needs and to close the gap between IDS research and IDS application in real-life.

M-ICE is splited up into several basic components. These basic components can be customized by applying dynamic loadable modules. All components exchange information via a TCP/IP network or via local interprocess communication. All parts of M-ICE can be run on a single system or can be installed on a dedicated system and distributed in a network.

First our IDS needs to get the audit-data. These data is generated on a host, the client in the IDS context, and can be application-level data like syslog or kenel-level data like syscall logs (BSM, LAuS, ...). Forwarding these data will by done by the `DataForwarder`. The `DataForwarder` has to load three modules, one to filter the raw data, one to pseudonymize the user related identifiers, and another one to convert the audit-data to a standard format. After passing all modules the `DataForwarder` sends the data to an analysis-agent and/or to a database.

M-ICE uses two kinds of databases, one for archiving raw audit-data and another one for collecting all detected alerts. In the case of uncirtainness about on system compromise the first databases can be queried manually to obtain useful informations.

The raw-log database is assembled by using a gerneric network daemon that is called `BufferDaemon` and two modules. One module is used to decode the data received from the network, the other module processes the network data after a user-defined delay and send it to a local MySQL database.

The analysis-agent is assembled in an equal way by using the `BufferDaemon`, a decoding module, and a postprocessing module. The postprocessing module is the code part where the analysis happens. The analysis-agent can send its data to another analysis-agent and/or to the `ManagementHost`.

The `ManagementHost` is a combination of a `BufferDaemon`, a decoding module, a postprocessing module, several reaction-agents, and a MySQL database for archiving received alerts. The `ManagementHost` receives the alerts of the M-ICE analysis-agents in the IDMEF [1] format. By using a standardized message format in combination of a flexible relation of alert-IDs and reaction-types it is possible to handle even other Intrusion Detection Systems like Snort (with the apropriate output module).

If a reaction has to be executed, a message (RPC like) will be send from the `ManagementHost` to the `ReactionDaemon` on the client. The reactions are implemented as a dynamic loadable module. The modules are identified by an ID that is send within the message from the `ManagementHost`. By this way reactions can be added at runtime and are very flexible.

Chapter 3

Software

The M-ICE project is hosted at Sourceforge [5]. All documentation, source-code and binary RPM files can be downloaded from there. The project provides a RPM file for every logical component of the M-ICE system and RPM files for logical sets of components, like `mice-all-client-0.1-0.i386.rpm` for all files needed for the client system. A list of RPM files:

- mice-0.1-0.i386.rpm
- mice-all-analysis-0.1-0.i386.rpm
- mice-all-client-0.1-0.i386.rpm
- mice-all-management-0.1-0.i386.rpm
- mice-all-rawlogdb-0.1-0.i386.rpm
- mice-bufferdaemon-0.1-0.i386.rpm
- mice-bufferdaemon-aa-0.1-0.i386.rpm
- mice-bufferdaemon-mngmthost-0.1-0.i386.rpm
- mice-bufferdaemon-rawlogdb-0.1-0.i386.rpm
- mice-dataforwarder-0.1-0.i386.rpm
- mice-module-dec-idmef-twofish-0.1-0.i386.rpm
- mice-module-dec-logformat-twofish-0.1-0.i386.rpm
- mice-module-dec-webstat-unixdomain-0.1-0.i386.rpm
- mice-module-filter-regex-0.1-0.i386.rpm
- mice-module-format-simple-0.1-0.i386.rpm

- mice-module-pop-aa-regex-0.1-0.i386.rpm
- mice-module-pop-aa-webstat-fifo-0.1-0.i386.rpm
- mice-module-pop-act-generic-0.1-0.i386.rpm
- mice-module-pop-mysql-0.1-0.i386.rpm
- mice-module-pop-syslog-0.1-0.i386.rpm
- mice-module-pop-webstat-idmef-0.1-0.i386.rpm
- mice-module-pseudo-dummy-0.1-0.i386.rpm
- mice-module-rct-dummy-0.1-0.i386.rpm
- mice-module-rid-countermeasure-0.1-0.i386.rpm
- mice-module-rid-savetofile-0.1-0.i386.rpm
- mice-module-rid-sendtoalertdb-0.1-0.i386.rpm
- mice-module-rid-writetosyslog-0.1-0.i386.rpm
- mice-reactiondaemon-0.1-0.i386.rpm

Additional RPM packages can be found at the download-section of the M-ICE Sourceforge homepage too, like:

- libidmef-0.7.2-0.i586.rpm
- libidmef-devel-0.7.2-0.i586.rpm
- ntp-4.2.0-0.i586.rpm
- LAuS'ized Linux kernel

Chapter 4

Client Setup

In the M-ICE context the client is the system we want to monitor. Ideally the client system runs the `DataForwarder` and the `ReactionDaemon`. When we think about our IDS as an circle of processing data, the client system is the conjunction between the start-point (`DataForwarder`) and the end-point (`ReactionDaemon`).

4.1 The Linux Audit Subsystem - LAuS

The Linux Audit Subsystem (LAuS) was developed by the SuSE Linux Security-Team during a security certification (EAL3). LAuS enables an admin to log syscalls and to have an audit-trail to follow the track of every login.

If you are running an SuSE system you can download the RPM package for the binary kernel or the kernel-source. Otherwise you need to download the kernel- patches from SuSE to patch and compile your own kernel. Additionally to the modified kernel you need to download and install the user-space tools and modified system tools. Most of these system tools are just available for SuSE Linux Enterprise Server 8 with ServicePack 3. Fortunately everything we need for syscall-logging is available for free. [1]

4.1.1 Setup

At this point I assume a system with a LAuS'ized kernel and all user-space tools installed. To get an overview of LAuS read `laus(7)`

The audit-subsystem can easily be startet:

```
root@client  # rcaudit start
Starting audit subsystem                                                    done
root@client  # rcaudit status
Checking for audit daemon:                                                  running
```

[1] all necessary links are listed at the download-section of the M-ICE homepage

To read or search in the logs use `aucat(1)` or `augrep(1)`, like:

```
root@client:~ # aucat
2004-02-01T15:44:12      0   1787     root [AUDIT_start] audit system started
2004-02-01T15:45:18      0   1787     root [AUDIT_stop] audit system stopped
2004-02-01T15:45:23      0   1833     root [AUDIT_start] audit system started
root@client:~ # augrep -h
     TIME            PID     LOGIN   DATA
2004-02-01T15:44:12   1787     root [AUDIT_start] audit system started
2004-02-01T15:45:18   1787     root [AUDIT_stop] audit system stopped
2004-02-01T15:45:23   1833     root [AUDIT_start] audit system started
root@client:~ #
```

The filter rules for the syscalls can be found in `/etc/audit/filter.conf` and the corresponding man-pages are `audit-filesets.conf(5)` and `audit-filter.conf(5)`. The default filter-rules should suffice.

The last step would be to enable the audit-subsystem at boot-time. We have two possibilities to start auditing. The service that authenticate users can attach processes to the audit-subsystem [2] or we can simply attach every process. I choose the latter mothod by creating and editing file `/etc/sysconfig/audit`, activating service audit and rebooting the machine:

```
root@client:~ # echo "AUDIT_ATTACH_ALL=1" > /etc/sysconfig/audit
root@client:~ # chkconfig -set audit
root@client:~ # reboot
```

Hopefully everything wents fine and your system comes back with a full featured audit-subsystem. Depending on your machine and on your config you may recognize a higher level of load. But now we have a lot more log information available:

```
root@client:~ # aucat | wc -l
   3297
root@client:~ # aucat | tail
2004-02-01T16:19:36   3322   1302   -1 [PROC_realgid] setgroups32(1, [51]);
                                       result=0
2004-02-01T16:19:36   3323   1302   -1 [PRIV_userchange] setresuid32(-1, 51, -1);
                                       result=0
2004-02-01T16:19:36   3324   1302   -1 [PRIV_userchange] setresuid32(-1, 0, -1);
                                       result=0
2004-02-01T16:19:36   3325   1302   -1 [PROC_realgid] setgid32(0); result=0
2004-02-01T16:19:36   3326   1302   -1 [PROC_realgid] setgroups32(1, [0]);
                                       result=0
```

[2] this requires modified system tools and at least pam_laus.so

```
2004-02-01T16:19:36   3327   1302   -1 [PRIV_userchange] setuid32(0); result=0
2004-02-01T16:19:41   3328   1754   -1 [PROC_execute] execve("/usr/sbin/aucat",
                                       ["aucat"], [data, len=0])
2004-02-01T16:19:41   3329   1755   -1 [PROC_execute] execve("/usr/bin/wc",
                                       ["wc", "-l"], [data, len=0])
2004-02-01T16:19:41   3330   1754   -1 [FILE_open] open("/var/log/audit.d/bin.0",
                                       O_RDONLY); result=3
2004-02-01T16:19:48   3331   1756   -1 [PROC_execute] execve("/usr/sbin/aucat",
                                       ["aucat"], [data, len=0])
```

I advise you to read all the man-pages to get a clue about this powerful system and to customize it. If you have access to the `certification-eal3.rpm` read the security-guide included.

4.2 Data Forwarder

At least the client needs to forward the audit-data to the database and/or the analysis-agent. The `DataForwarder` will switch to user `datafwd` and group `mice` after it has setup itself correctly. Please create this user and group.

4.2.1 Installation

For this purpose we have to install the `DataForwarder`.

```
root@client # rpm -ivh mice-dataforwarder-0.1-0.i386.rpm
mice-dataforwarder         ##################################################
root@client  # rpm -ql mice-dataforwarder
/etc/M-ICE
/etc/M-ICE/dataforwarder.conf
/etc/init.d/MICE-dataforwarder
/usr/sbin/dataforwarder
/usr/share/doc/packages/mice-dataforwarder
/usr/share/doc/packages/mice-dataforwarder/AUTHORS
/usr/share/doc/packages/mice-dataforwarder/COPYING
/usr/share/doc/packages/mice-dataforwarder/ChangeLog
/usr/share/doc/packages/mice-dataforwarder/INSTALL
/usr/share/doc/packages/mice-dataforwarder/NEWS
/usr/share/doc/packages/mice-dataforwarder/README
/usr/share/doc/packages/mice-dataforwarder/TODO
```

Additionally we have to install the filter module, pseudonymizer module, and the format module.

```
root@client # rpm -ivh mice-module-filter-regex-0.1-0.i386.rpm
mice-module-filter-regex    ##################################################
root@client # rpm -ql mice-module-filter-regex
/etc/M-ICE/mice_mod_flt_regex.conf
/usr/lib/mice_mod_flt_regex.a
/usr/lib/mice_mod_flt_regex.la
/usr/lib/mice_mod_flt_regex.so.1.0.0
/usr/share/doc/packages/mice-module-filter-regex
/usr/share/doc/packages/mice-module-filter-regex/AUTHORS
/usr/share/doc/packages/mice-module-filter-regex/COPYING
/usr/share/doc/packages/mice-module-filter-regex/ChangeLog
/usr/share/doc/packages/mice-module-filter-regex/INSTALL
/usr/share/doc/packages/mice-module-filter-regex/NEWS
/usr/share/doc/packages/mice-module-filter-regex/README
/usr/share/doc/packages/mice-module-filter-regex/TODO

root@client # rpm -ivh mice-module-pseudo-dummy-0.1-0.i386.rpm
mice-module-pseudo-dummy    ##################################################
root@client # rpm -ql mice-module-pseudo-dummy
/etc/M-ICE/mice_mod_psd_empty.conf
/usr/lib/mice_mod_psd_empty.a
/usr/lib/mice_mod_psd_empty.la
/usr/lib/mice_mod_psd_empty.so.1.0.0
/usr/share/doc/packages/mice-module-pseudo-dummy
/usr/share/doc/packages/mice-module-pseudo-dummy/AUTHORS
/usr/share/doc/packages/mice-module-pseudo-dummy/COPYING
/usr/share/doc/packages/mice-module-pseudo-dummy/ChangeLog
/usr/share/doc/packages/mice-module-pseudo-dummy/INSTALL
/usr/share/doc/packages/mice-module-pseudo-dummy/NEWS
/usr/share/doc/packages/mice-module-pseudo-dummy/README
/usr/share/doc/packages/mice-module-pseudo-dummy/TODO

root@client # rpm -ivh mice-module-format-simple-0.1-0.i386.rpm
mice-module-format-simple   ##################################################
root@client # rpm -ql mice-module-format-simple
/etc/M-ICE/mice_mod_fmt_simple.conf
/usr/lib/mice_mod_fmt_simple.a
/usr/lib/mice_mod_fmt_simple.la
/usr/lib/mice_mod_fmt_simple.so.1.0.0
/usr/share/doc/packages/mice-module-format-simple
/usr/share/doc/packages/mice-module-format-simple/AUTHORS
/usr/share/doc/packages/mice-module-format-simple/COPYING
/usr/share/doc/packages/mice-module-format-simple/ChangeLog
```

```
/usr/share/doc/packages/mice-module-format-simple/INSTALL
/usr/share/doc/packages/mice-module-format-simple/NEWS
/usr/share/doc/packages/mice-module-format-simple/README
/usr/share/doc/packages/mice-module-format-simple/TODO
```

4.2.2 Configuration

The configuration file is written in the MS Windows INI-format. [3] The following shows the default configuration file for the `DataForwarder` (`/etc/M-ICE/dataforwarder.conf`):

```
#
# This is the Config File for the Data-Forward Agent - Part of M-ICE IDS
#
# Thomas Biege <thetom@uin4d.de>
#

[SQL_SERVER]
SQLIP     = 172.16.0.10
SQLPORT   = 3366
Server
SQLPROTO  = TCP

[ANALYSIS_SERVER]
ASIP      = 172.16.0.10
ASPORT    = 3367
ASPROTO   = TCP


[SECURITY_AND_PRIVACY]
ENCRYPT   = yes
PSEUDO    = no
FILTER    = yes
USER      = datafwd
GROUP     = mice
CHROOT    = /var/datafwd/
ENCKEY    = "changeme"
CRYPTMOD  = "twofish"
DEVRANDOM = /dev/random


[LOGFILE_LIST]
```

[3] People may not like it but it's a well-known format and easy to understand. It may be replaced by a more sophisticated format in the future

```
FILE      = /var/log/messages
#FILE      = /var/log/xferlog
#FILE      = /var/log/secure_server.log
#FILE      = /var/log/ntp
#FILE      = /var/log/faillog
#FILE      = /var/log/http.access_log
#FILE      = /var/log/http.error_log

[MODULES_SEARCH_PATH]
MODPATH   = /usr/lib/

[MODULES]
MOD_FILTER    = mice_mod_flt_regex
MOD_LOGFORMAT = mice_mod_fmt_simple
MOD_PSEUDONYM = mice_mod_psd_empty

[MODULES_CONFIG_FILE]
FILTERCONF      = /etc/M-ICE/mice_mod_flt_regex.conf
LOGFORMATCONF   = /etc/M-ICE/mice_mod_fmt_simple.conf
PSEUDONYMCONF   = /etc/M-ICE/mice_mod_psd_empty.conf

[MISC]
SHMSIZE     = 10240
SLEEPINV    = 2
RECONNECT   = 15
PIDPATH     = /var/run/M-ICE/
#SYSLOGFAC = LOG_DAEMON
```

The file is subdevided in sections. Sections are starting with `[SECTION_NAME]`. Each section accommodates various items of different types: integer, strings, array. Comments start with #.

The first section `[SQL_SERVER]` is used to specify the connection to the raw-log database. The item `SQLIP` and `SQLPORT` can be used to define the address and the port number of the database server. If both are set to 0 no data will be send. `SQLPROTO` defines the protocol. The default is TCP, but it can also be UDP or something else. [4]

The same goes for section `[ANALYSIS_SERVER]`.

Section `[SECURITY_AND_PRIVACY]` contains the parameters that influence the security and privacy settings of the `DataForwarder` (what a suprise).

ENCRYPT	"yes" or "no" to encrypt the data sent over the network or not [5]

[4] at the moment just TCP is supported

[5] disable encryption if you encounter checksum errors when receiving data

PSEUDO	“yes“ or “no“ to pseudonymize user-related audit-data or not
FILTER	“yes“ or “no“ to filter the audit-data or not
USER	this defines the user to change to after setting up all the needed functions that need root privileges
GROUP	same as USER for the group ID
CHROOT	directory to chroot to [6]
ENCKEY	Key to use for encryption [7]
CRYPTOMOD	crypto module to load, this module is the crypto algorithm to use [8]
DEVRANDOM	source of randomness [9]

A list of data-sources have to be set in section `[LOGFILE_LIST]`. The item type is a string array and can contain plain-text application log file (`sysklog(8)`) or character device files that output data line-by-line.

FILE	name of file or char device can be defined multiple time

The next three sections are needed to provide information for the dynamic loadable modules. Section `[MODULES_SEARCH_PATH]` is a list of pathes where to look for the modules to load. It works the same way as section `[LOGFILE_LIST]`.

MODPATH	name of path where modules are be located

Section `[MODULES]` contains the name of the filter, format and pseudonymisation module (without the trailing “.so“).

MOD_FILTER	filter module
MOD_LOGFORMAT	format module
MOD_PSEUDONYM	pseudonymisation module

The config file for the appropriate module can be specified in section `[MODULES_CONFIG_FILE]`.

FILTERCONF	filter module config file
LOGFORMATCONF	format module config file
PSEUDONYMCONF	pseudonymisation module config file

[6] not implemented at the moment

[7] M-ICE uses symmetric cryptograpie

[8] at the moment just “twofish“ should be used

[9] not used at the moment

The last section [MISC] contains various information that do not fit in the previous sections.

SHMSIZE	size of shared memory to buffer raw log data
SLEPPINV	delay interval between file checks in seconds
RECONNECT	number of failed reconnects to database server and/or analysis-agent before giving up
PIDPATH	directory to place PID file to

Please read the configuration files of the modules too. The most interessting one is the filter config file which contains a list of regular expressions used to filter the raw log data.

4.2.3 Controlling

Controlling the DataForwarder can easily be done by using the SysV-like rc-file, like usual.

```
root@client # rcMICE-dataforwarder start
M-ICE IDS: Starting DataForwarder                                   done
root@client # rcMICE-dataforwarder status
M-ICE IDS: Status of DataForwarder
2099
                                                            done
root@client # rcMICE-dataforwarder stop
M-ICE IDS: Stopping DataForwarder                                   done
```

4.3 Reaction Daemon

IF you want to execute reactions on the client you have to install and configure the ReactionDaemon. The following sections will describe all the needed steps.

4.3.1 Installation

Installation of the appropriate RPM packages.

```
root@client # rpm -ivh mice-reactiondaemon-0.1-0.i386.rpm
mice-reactiondaemon         ##################################################
root@client # rpm -ql mice-reactiondaemon
/etc/M-ICE
/etc/M-ICE/reactiondaemon.conf
/etc/init.d/MICE-reactiondaemon
/usr/sbin/reactiondaemon
```

```
/usr/share/doc/packages/mice-reactiondaemon
/usr/share/doc/packages/mice-reactiondaemon/AUTHORS
/usr/share/doc/packages/mice-reactiondaemon/COPYING
/usr/share/doc/packages/mice-reactiondaemon/ChangeLog
/usr/share/doc/packages/mice-reactiondaemon/INSTALL
/usr/share/doc/packages/mice-reactiondaemon/NEWS
/usr/share/doc/packages/mice-reactiondaemon/README
/usr/share/doc/packages/mice-reactiondaemon/TODO
```

Additionally we have to install the reaction module(s).

```
root@client # rpm -ivh mice-module-rct-dummy-0.1-0.i386.rpm
mice-module-rct-dummy      ##################################################
root@client # rpm -ql mice-module-rct-dummy
/etc/M-ICE
/etc/M-ICE/mice_mod_rct_dummy.conf
/usr/lib/mice_mod_rct_dummy.a
/usr/lib/mice_mod_rct_dummy.la
/usr/lib/mice_mod_rct_dummy.so.1.0.0
/usr/share/doc/packages/mice-module-rct-dummy
/usr/share/doc/packages/mice-module-rct-dummy/AUTHORS
/usr/share/doc/packages/mice-module-rct-dummy/COPYING
/usr/share/doc/packages/mice-module-rct-dummy/ChangeLog
/usr/share/doc/packages/mice-module-rct-dummy/INSTALL
/usr/share/doc/packages/mice-module-rct-dummy/NEWS
/usr/share/doc/packages/mice-module-rct-dummy/README
/usr/share/doc/packages/mice-module-rct-dummy/TODO
```

4.3.2 Configuration

The following shows the default configuration file (`/etc/M-ICE/reactiondaemon.conf`):

```
#
# This is the Config File for the ReactionDaemon - Part of M-ICE IDS
#
# Thomas Biege <thetom@uin4d.de>
#

[FUNCTION_ID]
FID = 0x000001
#FID = 0x000002
#FID = 0x000003
#FID = 0x000004
#FID = 0x0000FF
```

```
[REACTION_MODULES]
RCT_MOD = mice_mod_rct_system

[REACTION_MODULES_CONFIG_FILE]
RCT_FILE = /etc/M-ICE/mice_mod_rct_system.conf

[NETWORK]
IP    = 0.0.0.0
PORT  = 1977

[SECURITY]
PASSWORD  = changeme
CRYPTMOD  = twofish
CHROOT    = /var/M-ICE/chroot/reactiondaemon/

[MISC]
PIDPATH   = /var/run/M-ICE/
MODPATH   = /lib/M-ICE/
BACKLOG   = 16
```

The array sections `[FUNCTION_ID]` and `[REACTION_MODULES]` are needed to connect a function-ID to an appropriate reaction module. A function-ID is an unique number that represents a reaction type/module and is part of the reaction message send from the `ManagementHost` to the client. In our example the function-ID `0x000001` corresponse to the module `mice_mod_rct_system`.

The config files for the reaction modules are defined in array section `[REACTION_MODULES_CONFIG_FILE]`. The first entry in section `[REACTION_MODULES]` coresponse to the first entry in section `[REACTION_MODULES_CONFIG_FILE]` and so on.

Section `[NETWORK]` set the network related parameters

IP	IP adress to listen to
PORT	port number to listen to

Security settings are controlled by the parameters in section `[SECURITY]` and are similar to the parameters of the `DataForwarder`.

The same goes for section `[MISC]`. The parameter `BACKLOG` defines the connection backlog for the socket.

4.3.3 Controlling

Controlling of the `ReactionDaemon` can easily be done by using the SysV-like rc-file, like usual.

```
root@client # rcMICE-reactiondaemon start
M-ICE IDS: Starting ReactionDaemon                                   done
root@client # rcMICE-reactiondaemon status
M-ICE IDS: Status ReactionDaemon:
3068
                                                             done
root@client # rcMICE-reactiondaemon stop
M-ICE IDS: Stopping ReactionDaemon
```

4.4 What else?

The `DataForwarder` should be started after all other components were set up. The `ReactionDaemon` can be started immediately because it just accepts connections. That's all...

Chapter 5

Raw-Log Database

Most of the servers used by M-ICE are a combination of the `BufferDaemon` plus a decoding and a post–processing module. The database can be installed on the client or on a different system.

5.1 BufferDaemon

The `BufferDaemon` is a generic network server. It reads data from the network, writes it in a ring-buffer and post-processes the buffered data after a user-defined time interval. Network servers often just differ in data-encoding and data-processing but most basic parts are the same. That is the reason for the `BufferDaemon`, it provides all the basic tasks and can be customized by using dynamic loadable decoding modules and post–processing modules. The `BufferDaemon` will switch to user `datafwd` and group `mice` after it has setup itself correctly. Please create this user and group.

5.1.1 Installation

There is only one `BufferDaemon`, but different kinds of `BufferDaemon` RPM files that include the appropriate config files.

```
root@rawlogdb # rpm -ivh mice-bufferdaemon-0.1-0.i386.rpm
mice-bufferdaemon          ##################################################
root@rawlogdb # rpm -ql mice-bufferdaemon
/etc/M-ICE
/etc/M-ICE/bufferdaemon.conf
/etc/init.d/mice-bufferdaemon
/usr/sbin/bufferdaemon
/usr/share/doc/packages/mice-bufferdaemon
/usr/share/doc/packages/mice-bufferdaemon/AUTHORS
/usr/share/doc/packages/mice-bufferdaemon/COPYING
```

```
/usr/share/doc/packages/mice-bufferdaemon/ChangeLog
/usr/share/doc/packages/mice-bufferdaemon/INSTALL
/usr/share/doc/packages/mice-bufferdaemon/NEWS
/usr/share/doc/packages/mice-bufferdaemon/README
/usr/share/doc/packages/mice-bufferdaemon/TODO

root@rawlogdb # rpm -ivh mice-bufferdaemon-rawlogdb-0.1-0.i386.rpm
mice-bufferdaemon-rawlogdb ##################################################
root@rawlogdb # rpm -ql mice-bufferdaemon-rawlogdb
/etc/M-ICE/bufferdaemon-rawlogdb.conf
/etc/init.d/mice-bufferdaemon-rawlogdb

root@rawlogdb # rpm -ivh mice-module-dec-logformat-twofish-0.1-0.i386.rpm
mice-module-dec-logformat-twofish ###########################################
root@rawlogdb # rpm -ql mice-module-dec-logformat-twofish
/etc/M-ICE
/etc/M-ICE/mice_mod_dec_logformat_twofish.conf
/usr/lib/mice_mod_dec_logformat_twofish.a
/usr/lib/mice_mod_dec_logformat_twofish.la
/usr/lib/mice_mod_dec_logformat_twofish.so.1.0.0
/usr/share/doc/packages/mice-module-dec-logformat-twofish
/usr/share/doc/packages/mice-module-dec-logformat-twofish/AUTHORS
/usr/share/doc/packages/mice-module-dec-logformat-twofish/COPYING
/usr/share/doc/packages/mice-module-dec-logformat-twofish/ChangeLog
/usr/share/doc/packages/mice-module-dec-logformat-twofish/INSTALL
/usr/share/doc/packages/mice-module-dec-logformat-twofish/NEWS
/usr/share/doc/packages/mice-module-dec-logformat-twofish/README
/usr/share/doc/packages/mice-module-dec-logformat-twofish/TODO
```

5.1.2 Configuration

Again the default config file.

```
#
# This is the Config File for the Buffer-Daemon - Part of M-ICE IDS
#
# Thomas Biege <thetom@uin4d.de>
#

[ADDRESS]
BINDTO  = 0.0.0.0

[PORT_NUMBER]
```

```
PORT  = 3366

[MODULES_SEARCH_PATH]
MODPATH = /lib/M-ICE/

[DECODING_MODULES]
DEC_MOD = mice_mod_dec_logformat_twofish

[DECODING_MODULES_CONFIG_FILE]
DEC_FILE = /etc/M-ICE/mice_mod_dec_logformat_twofish.conf

[POSTPROC_MODULES]
POST_MOD = mice_mod_pop_mysql

[POSTPROC_MODULES_CONFIG_FILE]
POST_FILE = /etc/M-ICE/mice_mod_pop_mysql.conf

[TIME_INTERVALS]
TIMEINV = 2

[CACHE_AND_RINGBUFFER]
MMPATH  = /var/M-ICE/cache.\texttt{BufferDaemon}
MMSIZE  = 0
RBSIZE  = 100

[SECURITY_AND_PRIVACY]
PSEUDO    = no
USER      = datafwd
GROUP     = mice
CHROOT    = /var/M-ICE/chroot/bufferdaemon/

[MISC]
PIDPATH   = /var/run/M-ICE/rawlogdb/
BACKLOG   = 16
```

Section [ADDRESS] and [PORT_NUMBER] specify the 2-tuple to listen to. These two sections are directly related to the [ENCODING_MODULES]. In our example it means that the BufferDaemon listens to all interfaces on port number 1977 and forwards all data received at this port to the appropriate decoding module. [1] The plain data will be placed in a ring-buffer and processed by a post–processing module after a user-defined time interval. This interval is speci-

[1] Please note that one BufferDaemon is able to handle multiple decoding and post–processing modules at once

fied in section [TIME_INTERVALS]. The post–processing module entry in section [POSTPROC_MODULES] is directly related to the entry in section [ENCODING_MODULES]. For the BufferDaemon configuration we have alot of related sections that may be confusing at the first time but the following item list may throw some light on it's data-processing:

1. listen on IP 0.0.0.0
2. wait for data on port 1977
3. accept data
4. data will be decoded by the module mice_mod_dec_logformat_twofish
5. the plain-text data will be buffered in a ring-buffer
6. after 2 seconds the
7. post–processing module mice_mod_pop_mysql will be called

The ring-buffer can be hold in memory or in a memory-mapped file [2]. Both can be configured in section [CACHE_AND_RINGBUFFER].

MMPATH	path fro memory-mapped file
MMSIZE	size of mmapped file in number of entries (0 means "ignore")
RBSIZE	size of ring-buffer in number of entries (0 means "ignore")

I think the remaining sections are self-explaining. [3]

5.1.3 Controlling

```
root@rawlogdb # rcmice-bufferdaemon-rawlogdb start
M-ICE IDS: Starting BufferDaemon (RawLog DB)                        done
root@rawlogdb # rcmice-bufferdaemon-rawlogdb status
M-ICE IDS: Status BufferDaemon (RawLog DB):
16233 16241 16245
                                                            done
root@rawlogdb # rcmice-bufferdaemon-rawlogdb stop
M-ICE IDS: Stopping BufferDaemon (RawLog DB)                        done
```

[2] it's not tested at the moment

[3] If not, let me know.

5.2 MySQL Database

You may choose one MySQL database for archiving both raw log-data and detected alarms or just separate both, that is up to you.

5.2.1 Installation

Just install the MySQL package available for your system. It should be part of the CDs shipped with the carton box you bought or you can fetch it from the internet [6].

5.2.2 Configuration

I will just describe the basic steps needed to create a database table and the appropriate user to access this table. First we check if the server is running and create a database.

```
root@rawlogdb # rcmysql status
Checking for service MySQL:                                          running

root@rawlogdb # mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2 to server version: 4.0.15

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> create database mice_rawlog_tab
    -> ;
Query OK, 1 row affected (0.00 sec)

mysql> quit
Bye
root@rawlogdb #
```

Now we have to fill this database with our structure. For this purpose I have written a text file that can be used as input for the `mysql` command. These textfile is part of the `mice-0.1-0.i386.rpm` file.

```
root@rawlogdb # mysql mice_rawlog_tab < \
/usr/share/doc/packages/mice-all-rawlogdb/create_mysql_tab_for_raw_logs.txt
```

The following tables were created. [4]

[4] Please note, these tables really need to be improved.

```
CREATE TABLE rawlog_line  ( hostname    TEXT,
                    domain      TEXT,
                    ip          TEXT,
                    osystem     TEXT,
                    release     TEXT,
                    version     TEXT,
                    date        DATE,
                    time        TIME,
                    logline     TEXT,
                    signature   TEXT );

CREATE TABLE scslog_line  ( hostname    TEXT,
                    domain      TEXT,
                    ip          TEXT,
                    osystem     TEXT,
                    release     TEXT,
                    version     TEXT,
                    date        DATE,
                    time        TIME,
                    syscall     TEXT,
                    program     TEXT,
                    pid         INT     UNSIGNED  NOT NULL,
                    uid         INT     UNSIGNED  NOT NULL,
                    euid        INT     UNSIGNED  NOT NULL,
                    call        TEXT,
                    comment     TEXT );

CREATE TABLE firewall_line  ( action      TEXT,
                      if_in       TEXT,
                      if_out      TEXT,
                      mac         TEXT,
                      source      TEXT,
                      destination TEXT,
                      ip_length   TEXT,
                      tos         TEXT,
                      prec        TEXT,
                      ttl         TEXT,
                      id          TEXT,
                      protocol    TEXT,
                      src_port    TEXT,
                      dst_port    TEXT,
                      pac_length  TEXT,
                      date        DATE,
```

```
                      time        TIME );
```

At the last step we will create the account for the MySQL user used by M-ICE.

```
root@rawlogdb # mysql mice_rawlog_tab
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 6 to server version: 4.0.15

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> grant INSERT,SELECT on rawlog_line.* to micedb@127.0.0.1 \
       IDENTIFIED BY 'changeme2';
Query OK, 0 rows affected (0.00 sec)

mysql> grant INSERT,SELECT on scslog_line.* to micedb@127.0.0.1 \
       IDENTIFIED BY 'changeme2';
Query OK, 0 rows affected (0.00 sec)

mysql> grant INSERT,SELECT on firewall_line.* to micedb@127.0.0.1 \
       IDENTIFIED BY 'changeme2';
Query OK, 0 rows affected (0.00 sec)
```

Let us check the tables.

```
mysql> select * from rawlog_line;
Empty set (0.02 sec)

mysql> select * from scslog_line;
Empty set (0.00 sec)

mysql> select * from firewall_line;
Empty set (0.00 sec)

mysql>
```

That's it so far.

5.2.3 Controlling

Please consult the documentation of your vendors MySQL package for information about controlling the MySQL server.

5.3 MySQL Module

The MySQL module acts as post–processing (pop) module for the `BufferDaemon` and is responsible for parsing the raw data and putting the coocked data in the MySQL database.

5.3.1 Installation

Standard procedure (note: `libmysqlclient.so.10` is need by this RPM package):

```
root@rawlogdb # rpm -ivh mice-module-pop-mysql-0.1-0.i386.rpm
mice-module-pop-mysql       ##################################################
root@rawlogdb # rpm -ql mice-module-pop-mysql
/etc/M-ICE
/etc/M-ICE/mice_mod_pop_mysql.conf
/usr/lib/mice_mod_pop_mysql.a
/usr/lib/mice_mod_pop_mysql.la
/usr/lib/mice_mod_pop_mysql.so.1.0.0
/usr/share/doc/packages/mice-module-pop-mysql
/usr/share/doc/packages/mice-module-pop-mysql/AUTHORS
/usr/share/doc/packages/mice-module-pop-mysql/COPYING
/usr/share/doc/packages/mice-module-pop-mysql/ChangeLog
/usr/share/doc/packages/mice-module-pop-mysql/INSTALL
/usr/share/doc/packages/mice-module-pop-mysql/NEWS
/usr/share/doc/packages/mice-module-pop-mysql/README
/usr/share/doc/packages/mice-module-pop-mysql/TODO
```

5.3.2 Configuration

The file of interest is named `/etc/M-ICE/mice_mod_pop_mysql.conf` and just contains a handfull options.

```
#
# This is the Config File for the mice_mod_mysql - Part of M-ICE IDS
#
# Thomas Biege <thetom@uin4d.de>
#

[MYSQL_SERVER]
HOSTNAME  = 127.0.0.1
PORT      = 3306
USER      = micedb
PASSWORD  = changeme2
```

```
DBNAME    = mice_rawlog_tab
MAXERR    = 10
```

HOSTNAME	address of MySQL server [5]
PORT	port number of MySQL server
USER	MySQL user
PASSWORD	password of MySQL user
DBNAME	MySQL table name
MAXERR	threshold for errors accepted due communication

5.4 Usage

Just start the `BufferDaemon`.

```
root@rawlogdb # rcmice-bufferdaemon-rawlogdb start
M-ICE IDS: Starting BufferDaemon (RawLog DB)                          done
```

The database can be read out by using your favorite mysql-client.

[5] link is unencrypted and should be localhost or tunneled otherwise

5.4.1 Reading out the Raw-Log Database

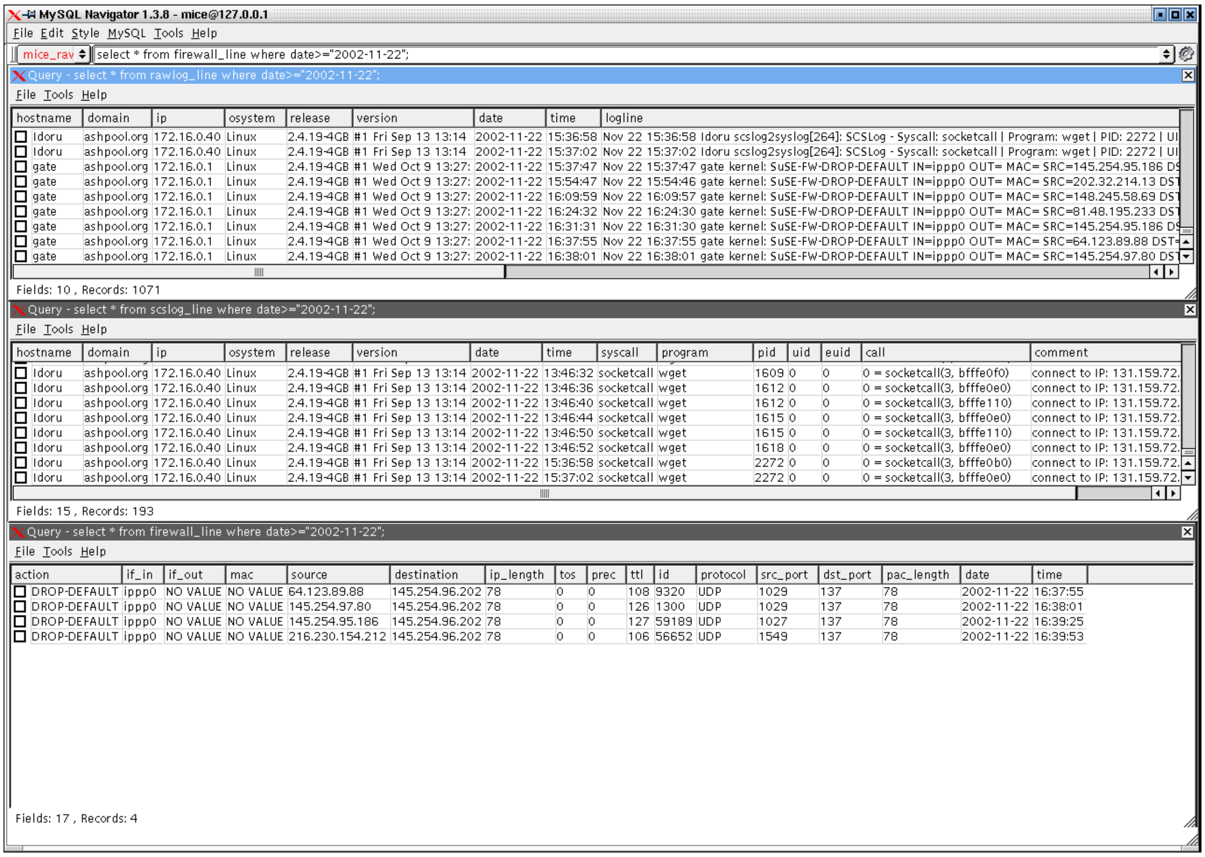

Figure 5.1: raw-log, scslog, firewall table

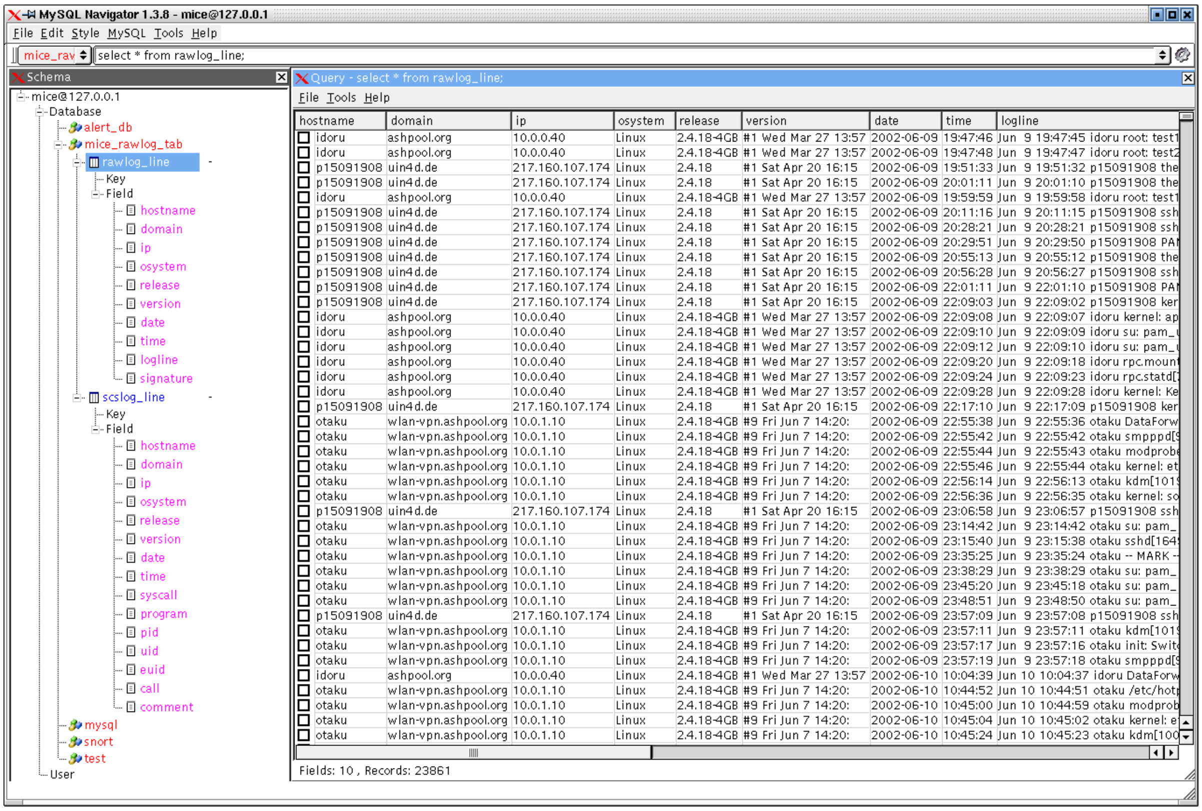

Figure 5.2: raw-log table

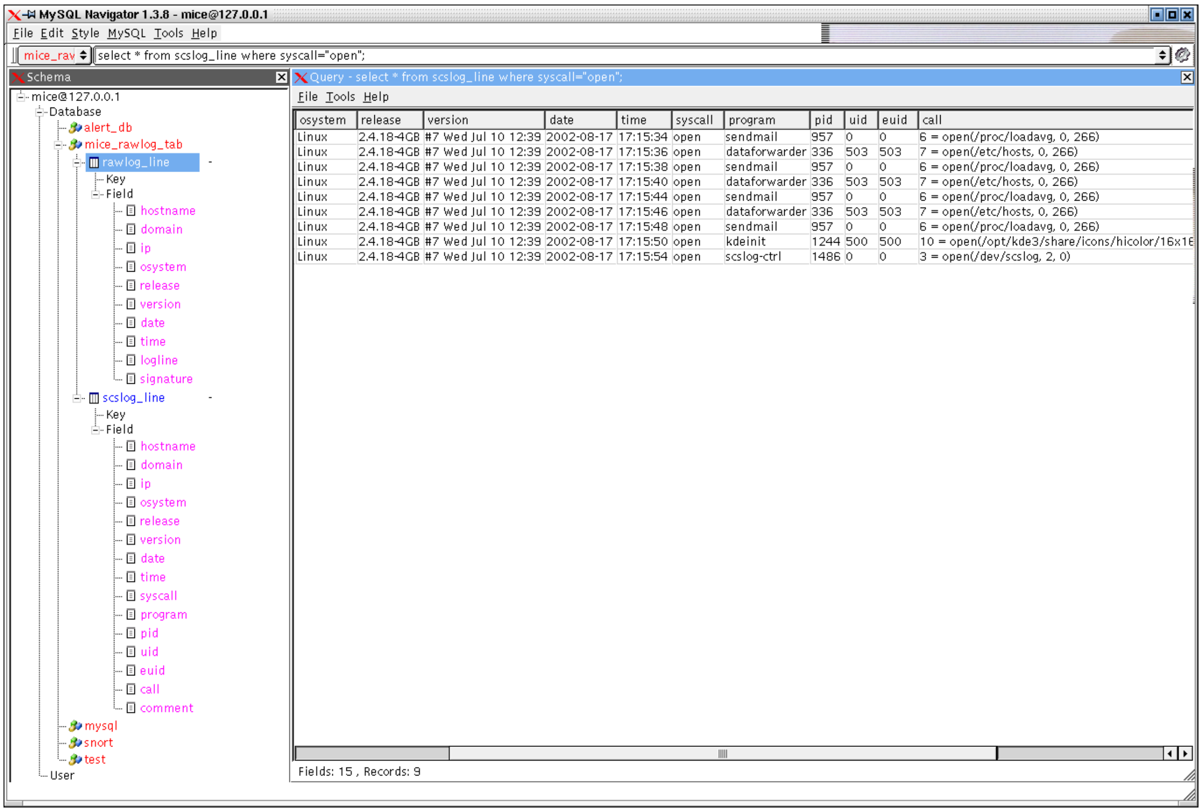

Figure 5.3: scslog table

Chapter 6

Analysis Agent

The analysis-agent can be the second target of the raw-log data. At the moment I will just describe a setup using one analysis-agent and not more or less.

The setup is nearly the same as the setup of the raw log database. The only thing that differs is the post–processing module.

6.1 BufferDaemon

The `BufferDaemon` will switch to user `datafwd` and group `mice` after it has setup itself correctly. Please create this user and group.

6.1.1 Installation

There is only one `BufferDaemon`, but different kinds of `BufferDaemon` RPM files that contain the appropriate config files.

```
root@aa # rpm -ivh mice-bufferdaemon-0.1-0.i386.rpm
mice-bufferdaemon          ##################################################
root@aa # rpm -ql mice-bufferdaemon
/etc/M-ICE
/etc/M-ICE/bufferdaemon.conf
/etc/init.d/mice-bufferdaemon
/usr/sbin/bufferdaemon
/usr/share/doc/packages/mice-bufferdaemon
/usr/share/doc/packages/mice-bufferdaemon/AUTHORS
/usr/share/doc/packages/mice-bufferdaemon/COPYING
/usr/share/doc/packages/mice-bufferdaemon/ChangeLog
/usr/share/doc/packages/mice-bufferdaemon/INSTALL
/usr/share/doc/packages/mice-bufferdaemon/NEWS
/usr/share/doc/packages/mice-bufferdaemon/README
/usr/share/doc/packages/mice-bufferdaemon/TODO
```

```
root@aa# rpm -ivh mice-bufferdaemon-aa-0.1-0.i386.rpm
mice-bufferdaemon-aa ##################################################
root@aa # rpm -ql mice-bufferdaemon-aa
/etc/M-ICE/bufferdaemon-aa.conf
/etc/init.d/mice-bufferdaemon-aa

root@aa # rpm -ivh mice-module-dec-logformat-twofish-0.1-0.i386.rpm
mice-module-dec-logformat-twofish ###########################################
root@aa # rpm -ql mice-module-dec-logformat-twofish
/etc/M-ICE
/etc/M-ICE/mice_mod_dec_logformat_twofish.conf
/usr/lib/mice_mod_dec_logformat_twofish.a
/usr/lib/mice_mod_dec_logformat_twofish.la
/usr/lib/mice_mod_dec_logformat_twofish.so.1.0.0
/usr/share/doc/packages/mice-module-dec-logformat-twofish
/usr/share/doc/packages/mice-module-dec-logformat-twofish/AUTHORS
/usr/share/doc/packages/mice-module-dec-logformat-twofish/COPYING
/usr/share/doc/packages/mice-module-dec-logformat-twofish/ChangeLog
/usr/share/doc/packages/mice-module-dec-logformat-twofish/INSTALL
/usr/share/doc/packages/mice-module-dec-logformat-twofish/NEWS
/usr/share/doc/packages/mice-module-dec-logformat-twofish/README
/usr/share/doc/packages/mice-module-dec-logformat-twofish/TODO
```

6.1.2 Configuration

Again, the default config file.

```
#
# This is the Config File for the Buffer-Daemon - Part of M-ICE IDS
#
# Thomas Biege <thetom@uin4d.de>
#

[ADDRESS]
BINDTO  = 0.0.0.0

[PORT_NUMBER]
PORT  = 3366

[MODULES_SEARCH_PATH]
MODPATH = /lib/M-ICE/
```

```
[DECODING_MODULES]
DEC_MOD = mice_mod_dec_logformat_twofish

[DECODING_MODULES_CONFIG_FILE]
DEC_FILE = /etc/M-ICE/mice_mod_dec_logformat_twofish.conf

[POSTPROC_MODULES]
POST_MOD = mice_mod_pop_aa_regex

[POSTPROC_MODULES_CONFIG_FILE]
POST_FILE = /etc/M-ICE/mice_mod_pop_aa_regex.conf

[TIME_INTERVALS]
TIMEINV = 2

[CACHE_AND_RINGBUFFER]
MMPATH  = /var/M-ICE/cache.bufferdaemon
MMSIZE  = 0
RBSIZE  = 100

[SECURITY_AND_PRIVACY]
PSEUDO     = no
USER       = datafwd
GROUP      = mice
CHROOT     = /var/M-ICE/chroot/bufferdaemon/

[MISC]
PIDPATH    = /var/run/M-ICE/rawlogdb/
BACKLOG    = 16
```

Section `[ADDRESS]` and `[PORT_NUMBER]` specify the 2-tuple to listen to. These two sections are directly related to the `[ENCODING_MODULES]`. For our example that means that the `BufferDaemon` listens to all interfaces on port number 1977 and forwards all data received at this port to the appropriate decoding module. The plain data will be placed in a ring-buffer and processed by a post–processing module after a user-defined time interval. This interval is specified in section `[TIME_INTERVALS]`. The post–processing module entry in section `[POSTPROC_MODULES]` is directly related to the entry in section `[ENCODING_MODULES]`. For the `BufferDaemon` configuration we have alot of related sections that may be confusing at the first time but the following item list may throw some light on it:

1. listen on IP `0.0.0.0`

2. wait for data on port `1977`
3. accept data
4. data will be decoded by the module `mice_mod_dec_logformat_twofish`
5. the plain-text data will be buffered in a ring-buffer
6. after `2` seconds the
7. post–processing module `mice_mod_pop_mysql` will be called

The ring-buffer can be hold in memory or in a memory-mapped file [1]. Both can be configured in section `[CACHE_AND_RINGBUFFER]`.

MMPATH	path fro memory-mapped file
MMSIZE	size of mmapped file in number of entries (0 means "ignore")
RBSIZE	size of ring-buffer in number of entries (0 means "ignore")

I think the remaining sections are self-explaining.

6.1.3 Controlling

```
root@aa # rcmice-bufferdaemon-aa start
M-ICE IDS: Starting BufferDaemon (analysis-agent)                    done
root@rawlogdb # rcmice-bufferdaemon-aa status
M-ICE IDS: Status BufferDaemon (analysis-agent):
1425 1426 1427
                                                             done
root@aa # rcmice-bufferdaemon-aa stop
M-ICE IDS: Stopping BufferDaemon (analysis-agent)                    done
```

6.2 Analysis Module

The analysis-module does the tricky job of detection intrusion attempts and categorizing the analysis results. The results will be converted to an IDMEF message and send to the `ManagementHost`.

6.2.1 Installation

Take care this module depends on the existence of `libidmef.so.0` and was just created as an testing module. At the moment there is no sophisticated way to analyse the data.

[1] it's not tested at the moment

```
root@aa # rpm -ivh  mice-module-pop-aa-regex-0.1-0.i386.rpm
mice-module-pop-aa-rege         ###########################################
root@aa # rpm -ql mice-module-pop-aa-regex
/etc/M-ICE
/etc/M-ICE/mice_mod_pop_aa_regex.conf
/usr/lib/mice_mod_pop_aa_regex.a
/usr/lib/mice_mod_pop_aa_regex.la
/usr/lib/mice_mod_pop_aa_regex.so.1.0.0
/usr/share/doc/packages/mice-module-pop-aa-regex
/usr/share/doc/packages/mice-module-pop-aa-regex/AUTHORS
/usr/share/doc/packages/mice-module-pop-aa-regex/COPYING
/usr/share/doc/packages/mice-module-pop-aa-regex/ChangeLog
/usr/share/doc/packages/mice-module-pop-aa-regex/INSTALL
/usr/share/doc/packages/mice-module-pop-aa-regex/NEWS
/usr/share/doc/packages/mice-module-pop-aa-regex/README
/usr/share/doc/packages/mice-module-pop-aa-regex/TODO
```

6.2.2 Configuration

These file is a bit more interessting then the once above. It contains all the various rules for the analysis-module to match raw-log data to certain kinds of classifications.

```
#
# This is the Config File for the mice_mod_aa-regex - Part of M-ICE IDS
#
# Thomas Biege <thetom@uin4d.de>
#

[HOST_INFO]
Management  = 172.16.0.10:4455
Agent       = 172.16.0.10:2266
Encryption  = yes
MngmntKey   = changeme1
AgentKey    = changeme2

[IDMEF_INFO]
AlertID     = 1
AlertIDFile = /var/log/M-ICE/mice_mod_aa_regex-alertid
DTDFile     = /etc/M-ICE/idmef-message.dtd


# Authentication Messages
```

```
[AUTH]
AuSendTo  = "M"
AuS_Rule  = ".*session started for user.*"
AuS_Rule  = ".*Accepted password.*"
AuF_Rule  = ".*FAILED SU.*"
AuF_Rule  = ".*Failed password.*"
AuF_Rule  = ".*Failed rsa.*"
AuF_Rule  = ".*ROOT LOGIN REFUSED FROM.*"

# root Actions
[ROOT]
RoSendTo  = "N"
RoW_Rule  = ""
RoR_Rule  = ""
RoO_Rule  = ""
RoS_Rule  = ""
RoE_Rule  = ""


# read(2)
[READ]
ReSendTo  = "N"
ReS_Rule  = ""
ReF_Rule  = ""


# write(2)
[WRITE]
WrSendTo  = "N"
WrS_Rule  = ""
WrF_Rule  = ""


# Monitor special UID/GID
[MONITORING]
MoSendTo  = "N"
Mo_UID    = ""
Mo_GID    = ""


# Special Applications
[APPS]
ApSendTo  = "M"
```

```
Ap_Name   = ".*sshd.*"


# Exploit Patterns
[EXPLOIT]
ExSendTo = "M"
Ex_Rule = ".*crc32 compensation attack: network attack detected.*"
Ex_Rule = ".*GET /scripts/root.exe?/c+dir.*"
Ex_Rule = ".*RECV: 163129 bytes of data, 0 total lines.*"
Ex_Rule = ".*REMOTE TSI ?%.*"

# Firewall Patterns
[FIREWALL]
FwSendTo = "M"
FwD_Rule = ".*SuSE-FW-DROP.*"
FwR_Rule = ""
FwA_Rule = ".*SuSE-FW-ACCEPT.*"
FwI_Rule = ".*SuSE-FW-ILLEGAL.*"

# Socket Operations
#[SOCKET]
#SoSendTo = "N"
#SoS_Rule = ""   # Success
#SoF_Rule = ""   # Failure
#SoC_Rule = ""   # Connect
#SoA_Rule = ""   # Accept
#SoL_Rule = ""   # Listen
#SoR_Rule = ""   # Raw Socket
#SoT_Rule = ""   # TCP
#SoU_Rule = ""   # UDP
#SoI_Rule = ""   # ICMP

# Default
[DEFAULT]
DeSendTo = "N"


# EOF
```

This looks a bit cryptic. Most of the sections contain regular expression to classify the raw log data. First a short description:

HOST_INFO	information about the management host and another analysis-agent

IDMEF_INFO	just some info we need for IDMEF, like the DTD file
AUTH	rules for matching authentication failures and success
ROOT	rules for matching root actions
READ	rules for matching read operations
WRITE	rules for matching read operations
MONITORING	rules for monitoring specific groups and/or users
APPS	rules for monitoring applications
EXPLOIT	rules for matching exploit patterns
FIREWALL	rules for classifying firewall output
SOCKET	not used at the moment
DEFAULT	default action to take if nothing matches

HOST_INFO In this section you can set the IP and port number of another analysis-agent and/or a `ManagementHost`. For security reasons you should switch encryption on and change the default keys.

Every rule-section has a send-to parameter. This parameter is just one character "M" for `ManagementHost`, "A" for analysis-agent, "B" for both types of components and "N" for none of the components.

AUTH The authentication section contains tags for successful (`AuS_Rule`) and failed (`AuF_Rule`) authentications. The configuration file contains some comman examples.

ROOT To monitor interessting root actions like write (`RoW_Rule`), read (`RoR_Rule`), open (`RoO_Rule`), socket operations (`RoS_Rule`) and execution of processes (`RoE_Rule`). These rules need a monitoring system on syscall-level on the client system.

READ Rules for successful (`ReS_Rule`) and failed (`ReF_Rule`) read operations in general can be set here. These rules need a monitoring system on syscall-level on the client system.

WRITE Same as `READ` section for write operations. These rules need a monitoring system on syscall-level on the client system.

MONITORING In this section you can define the group- (`Mo_GID`) and/or user-IDs (`Mo_UID`) to monitor.

APPS Special applications can be monitored by setting the appropriate rule in this section.

FIREWALL This section contains the tags for classifying output from, i.e. SuSEfirewall2, like drop (`FwD_Rule`), reject (`FwR_Rule`), accept (`FwA_Rule`) and illegal (`FwI_Rule`). Note that these example rules can easily be replaced by rules appropriate for your firewall logs.

Chapter 7

Management Host

The ManagementHost is the central point where all the analysis-results will be collected. Depending on the alert-ID the administrator can define reactions to be taken. Reactions are not hardcoded, they are implemented in small programs, that read a simple reaction-message from a FIFO and extract the needed informations from it to fulfil their tasks.

As mentioned before the ManagementHosts expects messages in IDMEF format. IDMEF is a flexible, open and extensible format based on UML and XML. This even allows the ManagementHost to receive messages from other Intrusion Detection Systems [1]. The decoding module may remove a shell sourrounding the IDMEF message, like headers or encryption. To be able to handle arbitrary alert-IDs the ManagementHost, or to be more precise the post–processing module, needs a table that matches alert-IDs to reactions. But more on this later.

7.1 BufferDaemon

Nothing new here. Please keep in mind that we just need ONE BufferDaemon per system but, at the moment, different configuration files. This will change in future so we have just ONE BufferDaemon and ONE configuration file. The BufferDaemon will switch to user datafwd and group mice after it has setup itself correctly. Please create this user and group.

7.1.1 Installation

We have to install the BufferDaemon, the buferdaemon config for the ManagementHost and the decoding module. For our purpose we use the decoding module for IDMEF messages encapsulated in a twofish encryption. [2]

[1] If I remember correctly Snort has an IDMEF output plugin

[2] All you need to adjust here is the key.

```
root@mh # rpm -ivh mice-bufferdaemon-0.1-0.i386.rpm
mice-bufferdaemon           ##################################################
root@mh # rpm -ql mice-bufferdaemon
/etc/M-ICE
/etc/M-ICE/bufferdaemon.conf
/etc/init.d/mice-bufferdaemon
/usr/sbin/bufferdaemon
/usr/share/doc/packages/mice-bufferdaemon
/usr/share/doc/packages/mice-bufferdaemon/AUTHORS
/usr/share/doc/packages/mice-bufferdaemon/COPYING
/usr/share/doc/packages/mice-bufferdaemon/ChangeLog
/usr/share/doc/packages/mice-bufferdaemon/INSTALL
/usr/share/doc/packages/mice-bufferdaemon/NEWS
/usr/share/doc/packages/mice-bufferdaemon/README
/usr/share/doc/packages/mice-bufferdaemon/TODO

root@mh # rpm -ivh mice-bufferdaemon-mngmthost-0.1-0.i386.rpm
mice-bufferdaemon-mngmthost##################################################
root@mh # rpm -ql mice-bufferdaemon-mngmthost
/etc/M-ICE/bufferdaemon-mngmthost.conf
/etc/init.d/mice-bufferdaemon-mngmthost

root@mh # rpm -ivh mice-module-dec-idmef-twofish-0.1-0.i386.rpm
mice-module-dec-idmef-twofi##################################################
root@mh # rpm -ql mice-module-dec-idmef-twofish
/etc/M-ICE
/etc/M-ICE/mice_mod_dec_idmef_twofish.conf
/usr/lib/mice_mod_dec_idmef_twofish.a
/usr/lib/mice_mod_dec_idmef_twofish.la
/usr/lib/mice_mod_dec_idmef_twofish.so.1.0.0
/usr/share/doc/packages/mice-module-dec-idmef-twofish
/usr/share/doc/packages/mice-module-dec-idmef-twofish/AUTHORS
/usr/share/doc/packages/mice-module-dec-idmef-twofish/COPYING
/usr/share/doc/packages/mice-module-dec-idmef-twofish/ChangeLog
/usr/share/doc/packages/mice-module-dec-idmef-twofish/INSTALL
/usr/share/doc/packages/mice-module-dec-idmef-twofish/NEWS
/usr/share/doc/packages/mice-module-dec-idmef-twofish/README
/usr/share/doc/packages/mice-module-dec-idmef-twofish/TODO
```

7.1.2 Configuration

The configuration file looks almost the same expect for the decoding and post-processing module.

```
#
# This is the Config File for the Buffer-Daemon - Part of M-ICE IDS
#
# Thomas Biege <thetom@uin4d.de>
#

[ADDRESS]
BINDTO  = 0.0.0.0

[PORT_NUMBER]
PORT  = 4455

[MODULES_SEARCH_PATH]
MODPATH = /lib/M-ICE/

[ENCODING_MODULES]
ENC_MOD = mice_mod_enc_idmef_twofish

[ENCODING_MODULES_CONFIG_FILE]
ENC_FILE = /etc/M-ICE/mice_mod_enc_idmef_twofish.conf

[OUTPUT_MODULES]
OUT_MOD = mice_mod_out_act_generic

[OUTPUT_MODULES_CONFIG_FILE]
OUT_FILE = /etc/M-ICE/mice_mod_out_act_generic.conf

[TIME_INTERVALS]
TIMEINV = 2

[CACHE_AND_RINGBUFFER]
MMPATH  = /var/M-ICE/cache.\texttt{BufferDaemon}
MMSIZE  = 0
RBSIZE  = 100

[SECURITY_AND_PRIVACY]
PSEUDO    = no
USER      = datafwd
GROUP     = mice
CHROOT    = /var/M-ICE/chroot/bufferdaemon/

[MISC]
PIDPATH   = /var/run/M-ICE/mngmthost/
```

```
BACKLOG   = 16
```

7.1.3 Controlling

```
root@mh # rcmice-bufferdaemon-mngmthost start
M-ICE IDS: Starting BufferDaemon (Management Host)                   done
root@mh # rcmice-bufferdaemon-mngmthost status
M-ICE IDS: Status BufferDaemon (Management Host):
2324 2326 2327
                                                             done
root@mh # rcmice-bufferdaemon-mngmthost stop
M-ICE IDS: Stopping BufferDaemon (Management Host)                   done
```

7.2 Management Module

The post–processing module used to implement the management operations is called `mice_mod_out_act_generic`.

7.2.1 Installation

This package needs `libidmef` to be installed.

```
root@mh # rpm -ivh mice-module-pop-act-generic-0.1-0.i386.rpm
mice-module-pop-act-gener##################################################
root@mh # rpm -ql mice-module-pop-act-generic
/etc/M-ICE
/etc/M-ICE/match_file_aa_regex.txt
/etc/M-ICE/match_file_aa_webstat.txt
/etc/M-ICE/mice_mod_pop_act_generic.conf
/usr/lib/mice_mod_pop_act_generic.a
/usr/lib/mice_mod_pop_act_generic.la
/usr/lib/mice_mod_pop_act_generic.so.1.0.0
/usr/share/doc/packages/mice-module-pop-act-generic
/usr/share/doc/packages/mice-module-pop-act-generic/AUTHORS
/usr/share/doc/packages/mice-module-pop-act-generic/COPYING
/usr/share/doc/packages/mice-module-pop-act-generic/ChangeLog
/usr/share/doc/packages/mice-module-pop-act-generic/INSTALL
/usr/share/doc/packages/mice-module-pop-act-generic/NEWS
/usr/share/doc/packages/mice-module-pop-act-generic/README
/usr/share/doc/packages/mice-module-pop-act-generic/TODO
```

7.2.2 Configuration

First lets look at the configuration file `mice_mod_pop_act_generic.conf`.

```
#
# This is the Config File for the mice_mod_aa-regex - Part of M-ICE IDS
#
# Thomas Biege <thetom@uin4d.de>
#

# List of Reaktion IDs
[REACTION_ID]
RID = 1
RID = 2
RID = 3
RID = 4


# and associated named pipes
[RID_PIPENAME]
PIPE = rid_1_write_to_syslog
PIPE = rid_2_send_to_alert_db
PIPE = rid_3_save_to_file
PIPE = rid_4_countermeasure


# Every Analyzer must send an ID...
[ANALYZER_ID]
ANAID = "MICE-RegEx"  # IDMEF Analyzer ID field


# ... to select the right file for matching Alert ID -> Reaction ID
[ANALYZER_MATCHFILE]
MF = match_file_aa_regex.txt


[IDMEF_INFO]
DTDFile     = /etc/M-ICE/idmef-message.dtd


[MISC]
MOD_PATH    = /lib/M-ICE
PIPE_PATH   = /var/run/M-ICE
PID_PATH    = /var/run/M-ICE
```

```
MATCH_PATH  = /etc/M-ICE/matchfiles
USER        = datafwd
GROUP       = mice

# EOF
```

Section [REACTION_ID], [RID_PIPENAME] and section [ANALYZER_ID], [ANALYZER_MATCHFILE] are directly related.

Let me describe them a bit more in detail:

REACTION_ID	Just a growing list of numerical identifiers for the reaction-modules.
PIPENAME	Name of FIFO, every reaction-agent has its own FIFO.
ANALYZER_ID	ID string of IDMEF message (IDS identifier).
ANALYZER_MATCHFILE	This is the most important part and includes a table for matching an alert-ID to an alert-description and to the appropriate reaction.

7.2.3 The Match File

This file is used by the act-generic post–processing module to match alert-IDs to descriptions and reactions. The following is an example of the regex analysis post–processing module.

```
#
# First we have to describe the Alarm we detect.
# This depends on the Analysis Module
#


# AlertID from IDMEF Classification-->name
[ALERT_ID]
AID = 0x1010    # RT_AUTH_S
AID = 0x1020    # RT_AUTH_F

AID = 0x2010    # RT_ROOT_W
AID = 0x2020    # RT_ROOT_O
AID = 0x2030    # RT_ROOT_R
AID = 0x2040    # RT_ROOT_S
AID = 0x2050    # RT_ROOT_E

AID = 0x3010    # RT_READ_S
AID = 0x3020    # RT_READ_F
```

```
AID = 0x4010    # RT_WRITE_S
AID = 0x4020    # RT_WRITE_F

AID = 0x5010    # RT_MONI_U
AID = 0x5020    # RT_MONI_G

AID = 0x6010    # RT_APPS_N

AID = 0x7010    # RT_EXPL

AID = 0x8010    # RT_FW_D
AID = 0x8020    # RT_FW_R
AID = 0x8030    # RT_FW_A
AID = 0x8040    # RT_FW_I


# Description for AlertID
[ALERT_ID_DESCRIPTION]
AID_DESC = "Authentication Success"
AID_DESC = "Authentication Failure"
AID_DESC = "Superuser writes to File"
AID_DESC = "Superuser opens File"
AID_DESC = "Superuser reads from File"
AID_DESC = "Superuser does Operation on Socket"
AID_DESC = "Superuser executes Code"
AID_DESC = "Read Success"
AID_DESC = "Read Failure"
AID_DESC = "Write Success"
AID_DESC = "Write Failure"
AID_DESC = "Monitoring UserID"
AID_DESC = "Monitoring GroupID"
AID_DESC = "Monitoring Application"
AID_DESC = "Exploit Signature detected"
AID_DESC = "Firewall Rule: Drop"
AID_DESC = "Firewall Rule: Reject"
AID_DESC = "Firewall Rule: Accept"
AID_DESC = "Firewall Rule: Illegal"


#
# Which Reaction should be triggered by which Alert(s)?
#
```

```
# Write to Syslog
[RID_1]
AID_1 = 0x1010
AID_1 = 0x1020
AID_1 = 0x2010
AID_1 = 0x2020
AID_1 = 0x2030
AID_1 = 0x2040
AID_1 = 0x2050
AID_1 = 0x3010
AID_1 = 0x3020
AID_1 = 0x4010
AID_1 = 0x4020
AID_1 = 0x5010
AID_1 = 0x5020
AID_1 = 0x6010
AID_1 = 0x7010

# Send to Alert DB
[RID_2]
AID_2 = 0x1010
AID_2 = 0x1020
AID_2 = 0x2010
AID_2 = 0x2020
AID_2 = 0x2030
AID_2 = 0x2040
AID_2 = 0x2050
AID_2 = 0x3010
AID_2 = 0x3020
AID_2 = 0x4010
AID_2 = 0x4020
AID_2 = 0x5010
AID_2 = 0x5020
AID_2 = 0x6010
AID_2 = 0x7010
AID_2 = 0x8010
AID_2 = 0x8020
AID_2 = 0x8030
AID_2 = 0x8040


# Save to File
```

```
[RID_3]
AID_3 = 0x1010
AID_3 = 0x1020
AID_3 = 0x2010
AID_3 = 0x2020
AID_3 = 0x2030
AID_3 = 0x2040
AID_3 = 0x2050
AID_3 = 0x3010
AID_3 = 0x3020
AID_3 = 0x4010
AID_3 = 0x4020
AID_3 = 0x5010
AID_3 = 0x5020
AID_3 = 0x6010
AID_3 = 0x7010


# Countermeasure
[RID_4]
AID_4 = 0x6010
AID_4 = 0x7010


# EMPTY
[RID_5]

# EMPTY
[RID_6]

# EMPTY
[RID_7]

# EMPTY
[RID_8]

# EMPTY
[RID_9]

# EMPTY
[RID_10]
```

```
# EOF
```

ALERT_ID	Numerical identifier to look for in IDMEF message.
ALERT_ID_DESCRIPTION	Description of alert.
RID_1	List of alert-IDs that trigger reaction with ID 1
RID_2	List of alert-IDs that trigger reaction with ID 2
RID_x	List of alert-IDs that trigger reaction with ID x

7.3 MySQL Database

This is the second database we have to establish. Its purpose is storing the detected alerts that were received by the `ManagementHost`.

7.3.1 Installation

Just install the MySQL package available for your system. It should be part of the CDs shipped with the carton box you bought or you can fetch it from the internet [6].

7.3.2 Configuration

I will just describe the basic steps needed to create a database table and the appropriate user to access this table. First we check if the server is running and create a database.

```
root@mh # rcmysql status
Checking for service MySQL:                                         running

root@mh # mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2 to server version: 4.0.15

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> create database alert_db
    -> ;
Query OK, 1 row affected (0.00 sec)

mysql> quit
Bye
root@mh #
```

Now we have to fill this database with our structure. For this purpose I have written a text file that can be used as input for the `mysql` command. These textfile is part of the `mice-0.1-0.i386.rpm` file.

```
root@mh # mysql alertdb < \
/usr/share/doc/packages/mice/create_mysql_tab_for_alert_logs.txt
```

The following table is created. [3]

```
CREATE TABLE alert_entry  ( alertid           TEXT,
                    alertdesc         TEXT,
                    classification    TEXT,
                    date              DATE,
                    time              TIME,
                    analyzerid        TEXT,
                    source_address    TEXT,
                    source_user       TEXT,
                    source_process    TEXT,
                    source_service    TEXT,
                    target_address    TEXT,
                    target_user       TEXT,
                    target_process    TEXT,
                    target_service    TEXT,
                    idmefmsg          TEXT,
                    signature         TEXT );
```

Next we have to create the account for the MySQL user used by the reaction-agent to access the database.

```
root@mh # mysql alert_db
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 6 to server version: 4.0.15

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> grant INSERT,SELECT on alert_entry.* to alertdb@127.0.0.1 \
       IDENTIFIED BY 'hallo';
Query OK, 0 rows affected (0.00 sec)
```

Let us check the table.

```
mysql> select * from alert_entry;
Empty set (0.03 sec)
```

That's it so far.

[3] Please note, these tables really need to be improved.

7.3.3 Controlling

Please consult the documentation of your vendors MySQL package for information about controlling the MySQL server.

7.4 Reaction Agents

As mentioned before the reaction-agents are small tools that read a message from a FIFO and fulfil their job. There exist four default reaction-agents handling the following actions:

- save IDMEF message to file
- send IDMEF message to alert-DB
- send IDMEF message to syslog
- countermeasure

7.4.1 Installation

The RPM packages are installed as always (`libidmef` and `libmysqlclient` needs to be installed).

```
root@mh # rpm -ivh mice-module-rid-savetofile-0.1-0.i386.rpm
mice-module-rid-savetofil##################################################
root@mh # rpm -ql mice-module-rid-savetofile
/etc/M-ICE
/etc/M-ICE/rid_3_save_to_file.conf
/etc/init.d/MICE-rid_3_save_to_file
/usr/sbin/rid_3_save_to_file
/usr/share/doc/packages/mice-module-rid-savetofile
/usr/share/doc/packages/mice-module-rid-savetofile/AUTHORS
/usr/share/doc/packages/mice-module-rid-savetofile/COPYING
/usr/share/doc/packages/mice-module-rid-savetofile/ChangeLog
/usr/share/doc/packages/mice-module-rid-savetofile/INSTALL
/usr/share/doc/packages/mice-module-rid-savetofile/NEWS
/usr/share/doc/packages/mice-module-rid-savetofile/README
/usr/share/doc/packages/mice-module-rid-savetofile/TODO

root@mh # rpm -ivh mice-module-rid-sendtoalertdb-0.1-0.i386.rpm
mice-module-rid-sendtoale###############################################
root@mh # rpm -ql mice-module-rid-sendtoalertdb
/etc/M-ICE
```

```
/etc/M-ICE/rid_2_send_to_alert_db.conf
/etc/init.d/MICE-rid_2_send_to_alert_db
/usr/sbin/rid_2_send_to_alert_db
/usr/share/doc/packages/mice-module-rid-sendtoalertdb
/usr/share/doc/packages/mice-module-rid-sendtoalertdb/AUTHORS
/usr/share/doc/packages/mice-module-rid-sendtoalertdb/COPYING
/usr/share/doc/packages/mice-module-rid-sendtoalertdb/ChangeLog
/usr/share/doc/packages/mice-module-rid-sendtoalertdb/INSTALL
/usr/share/doc/packages/mice-module-rid-sendtoalertdb/NEWS
/usr/share/doc/packages/mice-module-rid-sendtoalertdb/README
/usr/share/doc/packages/mice-module-rid-sendtoalertdb/TODO

root@mh # rpm -ivh mice-module-rid-writetosyslog-0.1-0.i386.rpm
mice-module-rid-writetosy##############################################
root@mh # rpm -ql mice-module-rid-writetosyslog
/etc/M-ICE
/etc/M-ICE/rid_1_write_to_syslog.conf
/etc/init.d/MICE-rid_1_write_to_syslog
/usr/sbin/rid_1_write_to_syslog
/usr/share/doc/packages/mice-module-rid-writetosyslog
/usr/share/doc/packages/mice-module-rid-writetosyslog/AUTHORS
/usr/share/doc/packages/mice-module-rid-writetosyslog/COPYING
/usr/share/doc/packages/mice-module-rid-writetosyslog/ChangeLog
/usr/share/doc/packages/mice-module-rid-writetosyslog/INSTALL
/usr/share/doc/packages/mice-module-rid-writetosyslog/NEWS
/usr/share/doc/packages/mice-module-rid-writetosyslog/README
/usr/share/doc/packages/mice-module-rid-writetosyslog/TODO

root@mh # rpm -ivh mice-module-rid-countermeasure-0.1-0.i386.rpm
mice-module-rid-counter  ##############################################
root@mh # rpm -ql mice-module-rid-countermeasure
/etc/M-ICE
/etc/M-ICE/rid_4_countermeasure.conf
/etc/init.d/MICE-rid_4_countermeasure
/usr/sbin/rid_4_countermeasure
/usr/share/doc/packages/mice-module-rid-countermeasure
/usr/share/doc/packages/mice-module-rid-countermeasure/AUTHORS
/usr/share/doc/packages/mice-module-rid-countermeasure/COPYING
/usr/share/doc/packages/mice-module-rid-countermeasure/ChangeLog
/usr/share/doc/packages/mice-module-rid-countermeasure/INSTALL
/usr/share/doc/packages/mice-module-rid-countermeasure/NEWS
/usr/share/doc/packages/mice-module-rid-countermeasure/README
/usr/share/doc/packages/mice-module-rid-countermeasure/TODO
```

7.4.2 Configuration

All configuration files include a line that describes the path to the FIFO. In general most reaction-agents just need little configuration modifications. The sections are self-explaining. [4]

7.5 Usage

The database can be read out by using your favorite mysql-client.

7.5.1 Reading out the Alert Database

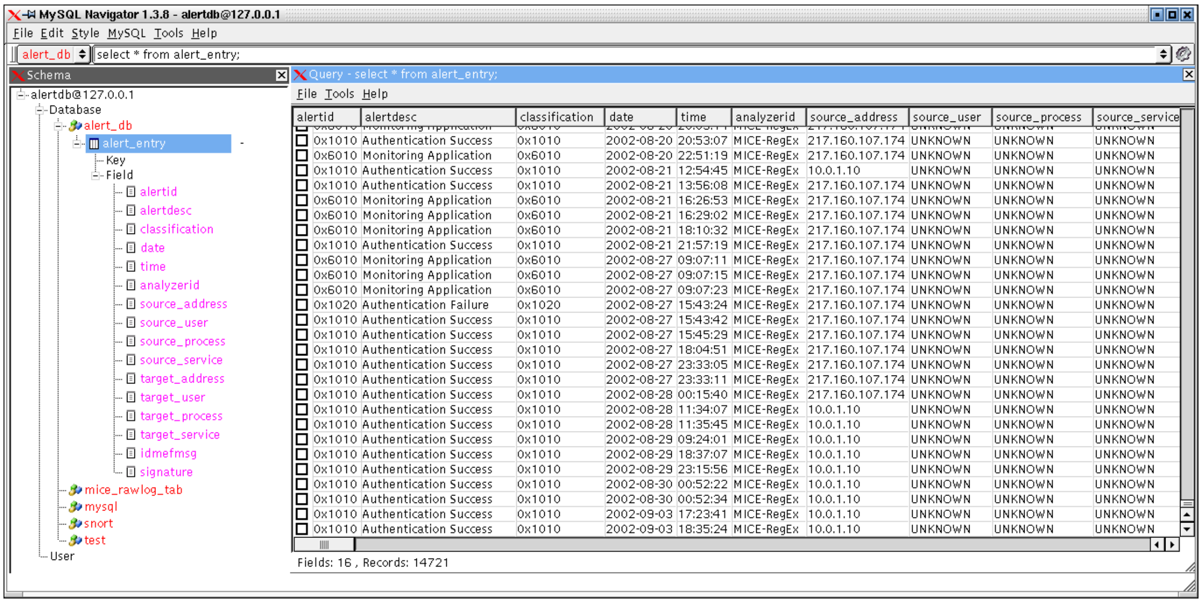

Figure 7.1: read all records and show table contents

[4] As always, let me know if not.

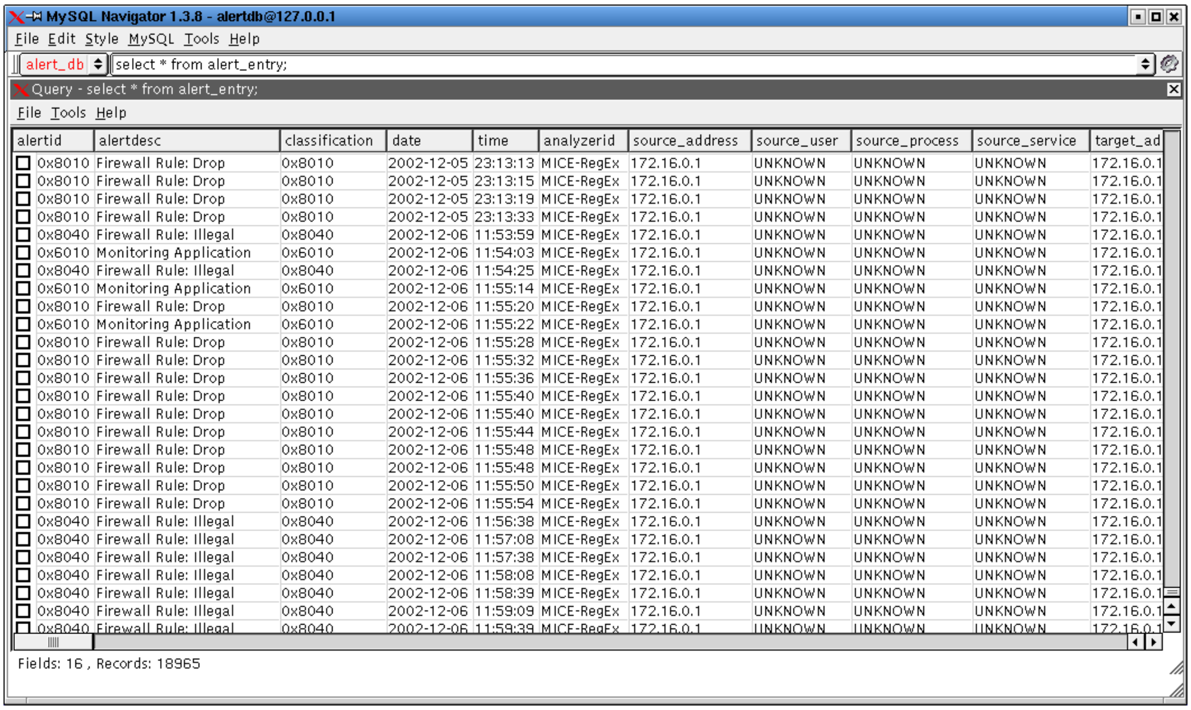

Figure 7.2: read all records

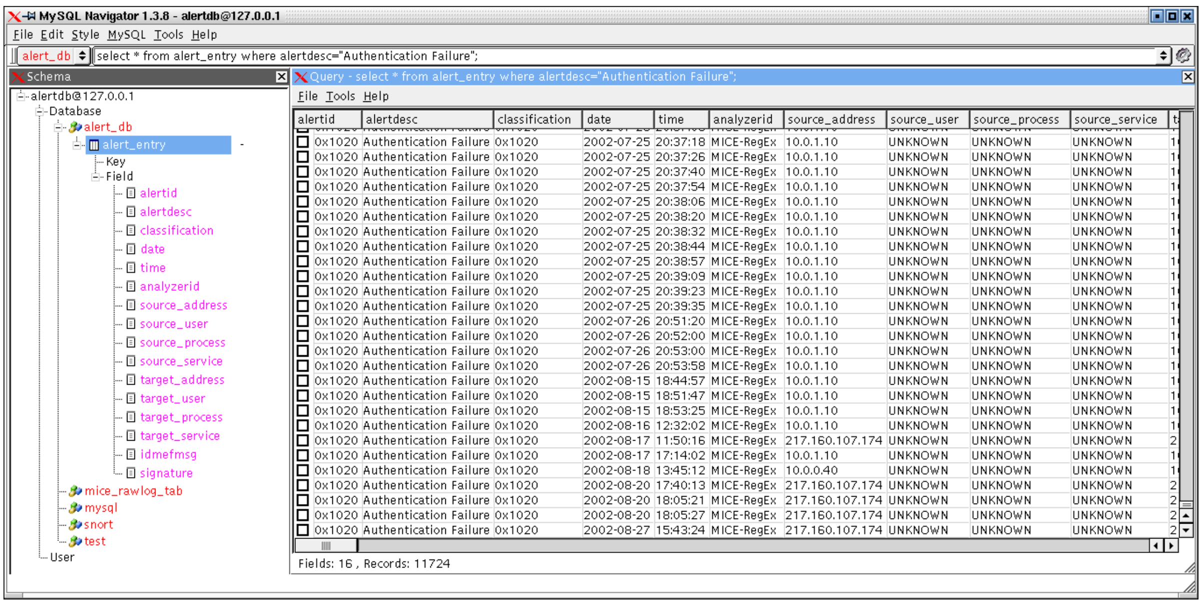

Figure 7.3: list authentication-failures

Chapter 8

ToDo

- test if uptodate

Appendix A

Abbreviations

DAC Discretionary Access Control (all files are protected by DAC rules)

DoS Denial–of–Service

FIFO First In, First Out; Named Pipe; local Interprocess Communication

GNU GNU's Not Unix!, Projekt of the *Free Software Foundation*

GUI Graphical User Interface

IDMEF Intrusion Detection Message Exchange Format

IDS Instrusion Detection System

IP Internet Protocol, s. RFC–791 [3]

LAuS Linux Audit-Subsystem

LKM Loadable Kernel Modul

PAM Pluggable Authentication Module

SO Security Officer

SQL Structured Query Language

SSL Secure Socket Layer, Encryption on presentationlayer

Syslog native Unix Logging System

TCP Transmission Control Protocol, s. RFC–793 [3]

UDP User Datagram Protocol, s. RFC–768 [3]

UML Unified Modeling Language

XML Extensible Markup Language

Appendix B

List of Figures

Appendix C

Bibliography

[1] D. Curry, H. Debar, Intrusion Detection Message Exchange Format — Data Model and Extensible Markup Language (XML) Document Type Definition, IDWG, February 2002

[2] LibIDMEF, http://www.silicondefense.com/idwg/libidmef/index.htm

[3] RFC Datenbank, http://www.rfc-editor.org/

[4] GNU Free Documentation Licence, http://www.gnu.org/copyleft/fdl.html

[5] M-ICE project at Sourceforge, http://sourceforge.net/projects/m-ice/

[6] MySQL, http://www.mysql.com

[7] CAPP Version 1d, http://www.radium.ncsc.mil/tpep/library/protection_profiles/CAPP-1.d.pdf