GSE Frühjahrestagung 2013
Z/VSE und z/VM mit Linux auf System z, U30 - zTalents

Leipzig, 22.-24. April 2013

VM02

Zentralisiertes Auditing für Linux auf System z mit z/OS SMF

Dr. Manfred Gnirss – gnirss@de.ibm.com

Guillaume Lasmayous – guillaume.lasmayous@fr.ibm.com

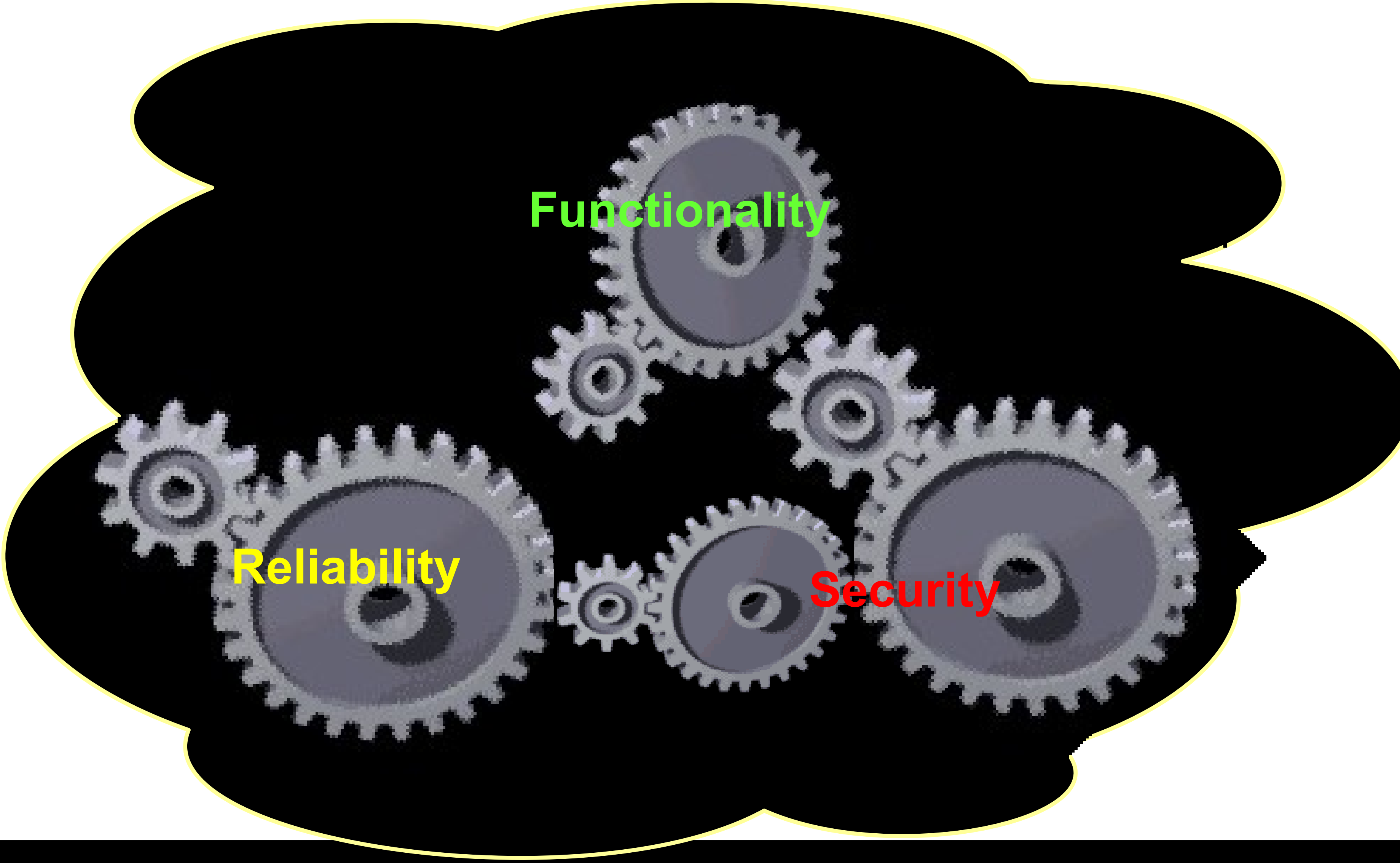
Auditing ?

- “An IT audit is an examination of the management controls within an Information technology (IT) infrastructure. The evaluation of obtained evidence determines if the information systems are safeguarding assets, maintaining data integrity, and operating effectively to achieve the organization's goals or objectives. These reviews may be performed in conjunction with a financial statement audit, internal audit, or other form of attestation engagement.”

From Wikipedia, the free encyclopedia (https://en.wikipedia.org/wiki/Information_technology_audit)

- What will be discussed here is the process to gather the evidence, not the evaluation of the evidence itself.

Security vs. Functionality and Reliability

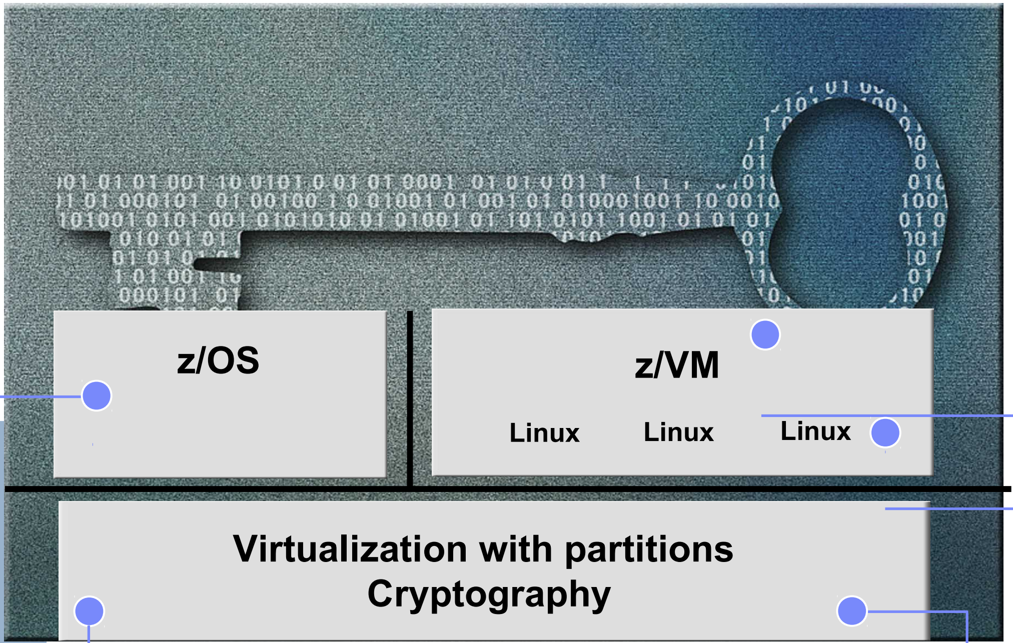


System z Evaluations and Certifications

The Common Criteria program establishes an organizational and technical framework to evaluate the trustworthiness of IT Products and protection profiles

z/OS

- **Common Criteria EAL4+**
  - with CAPP and LSPP
  - z/OS 1.7 → 1.10 + RACF
  - z/OS 1.11 + RACF (OSPP)
  - z/OS 1.12 + RACF (OSPP)
  - z/OS 1.13 + RACF (OSPP)
- **Common Criteria EAL5 +**
  - **z/OS RACF 1.12 (OSPP)**
- **z/OS 1.10 IPv6 Certification by JITC**
- **IdenTrust™ certification for z/OS PKI Services**
- **FIPS 140-2**
  - System SSL z/OS 1.10 →1.12
  - z/OS ICSF PKCS#11 Services
    - z/OS 1.11



System z9 EC and z9 BC System z10 EC and z10 BC

Common Criteria EAL5 with specific target of evaluation -- LPAR: Logical partitions

- **System z196 , z114 and zEC12**
  - **Common Criteria EAL5+ with specific target of Evaluation -- LPAR**
- **Crypto Expr.3 & Crypot Expr.4s Coprocessors**
  - FIPS 140-2 level 4 Hardware Evaluation
  - Approved by German ZKA
- **CP Assist**
  - FIPS 197 (AES)
  - FIPS 46-3 (TDES)
  - FIPS 180-3 (Secure Hash)

z/VM

- System Integrity Statement
- Common Criteria
  - z/VM 5.3 is EAL 4+ for CAPP and LSPP
  - z/VM 6.1 is EAL 4+ for OSPP

Linux on System z

- **Common Criteria**
  - SUSE SLES10 certified at EAL4+ with CAPP
  - Red Hat EL5 EAL4+ with CAPP and LSPP
  - SUSE SLES 11 EAL4+ with OSPP
  - RedHat EL6 EAL 4+ with OSPP
- **OpenSSL - FIPS 140-2 Level 1 Validated**
- **CP Assist - SHA-1 validated for FIPS 180-1 - DES & TDES validated for FIPS 46-3**

Notes:

- **Common Criteria Certification with Protection Profiles CAPP and LSPP or OSPP requires auditing capabilities**
- **z/OS, z/VM: via RACF**
- **Linux: via Linux Audit Framework**

Linux Audit Framework (LAF)

- Linux Audit Framework: important requirement of CC-CAPP/LSP or CC-OSPP certification

- Linux Audit Framework is a system to:

  – Collect information regarding events occurring on the running system

    - Kernel events (system-calls)
    - User events (audit-enabled programs)

  – Form and log a record describing each event using information collected from that event

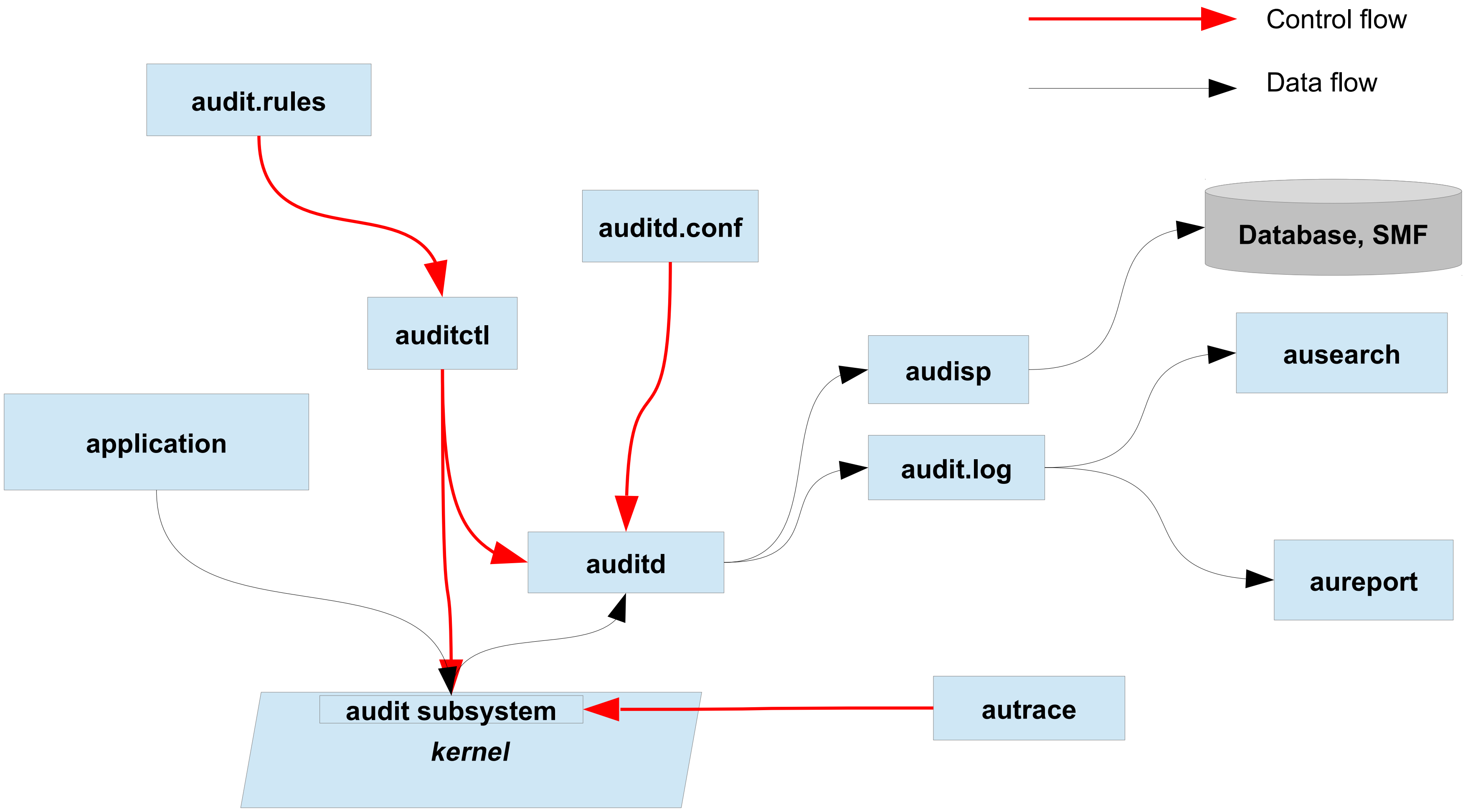
    - Syscall arguments, subject attributes, object attributes, time, and so on

  – Analyze the log of records

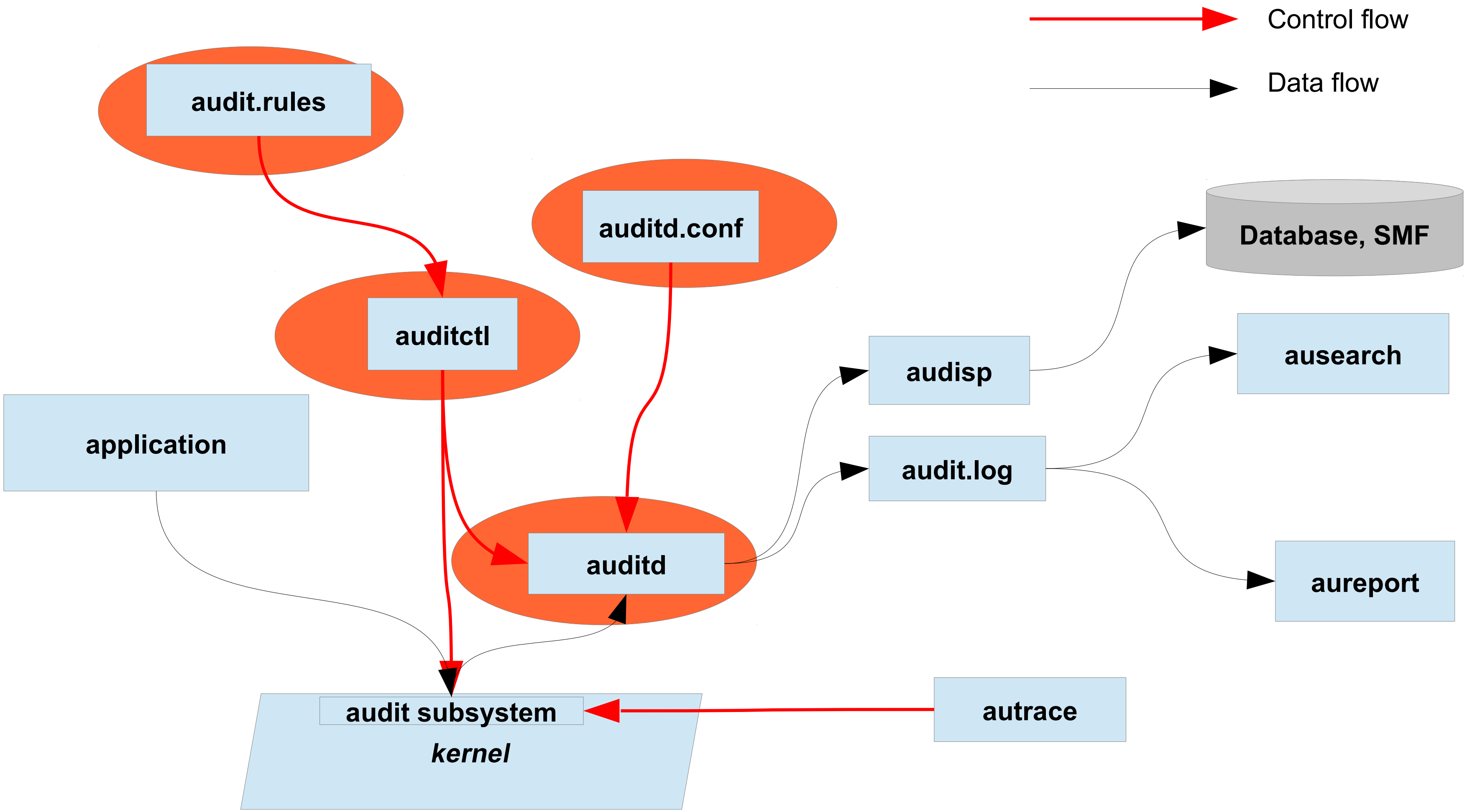
Linux Audit Framework (LAF) . . .

- Linux Audit Framework helps to make system more secure:
  - Analyse what is happening on your system
  - However, it does not provide additional security by itself. It does not protect from code malfunctions or any kind of exploits
  - But it is useful for traking these issues and helps you to take additional security measures to prevent them (in future).

Linux Audit Framework components



Linux Audit Framework components



Configuring the Audit Daemon

Default in /etc/audit/auditd.conf is reasonable, but not CC-CAP/LSPP or CC-OSPP compliant. You may want must(?) adapt the file.

Example:

```
log_file=path_to_separate_partition/audit.log
log_format = RAW
priority_boost = 4
flush = SYNC                                  ### or DATA
freq = 20
num_logs = 4
dispatcher = /sbin/audispd
disp_qos = lossy
max_log_file = 5
max_log_file_action = KEEP_LOGS
space_left = 75
space_left_action = EMAIL
action_mail_acct = root
admin_space_left = 50
admin_space_left_action = SINGLE     ### or HALT
disk_full_action = SUSPEND           ### or HALT
disk_error_action = SUSPEND          ### or HALT
```

Enable Audit for System Calls

Configure how audit daemon is started in /etc/sysconfig/auditd

```
AUDITD_LANG="en_US"
AUDITD_DISABLE_CONTEXTS="no"
```

For permanent auditing of system calls incl. file and directory watches set to “no” and restart audit daemon (Note: default is “yes”)

For temporary enabling of auditing of system calls incl. file and directory watches execute the following command as root:

```
auditctl -e 1
```

For temporary disabling of auditing of system calls incl. file and directory watches execute the following command as root:

```
auditctl -e 0
```

Setting Up Audit Rules

Audit rules can be passed to audit daemon using
- auditctl command line,
- or with a rule set in /etc/audit/audit.rules

Main auditctl commands :

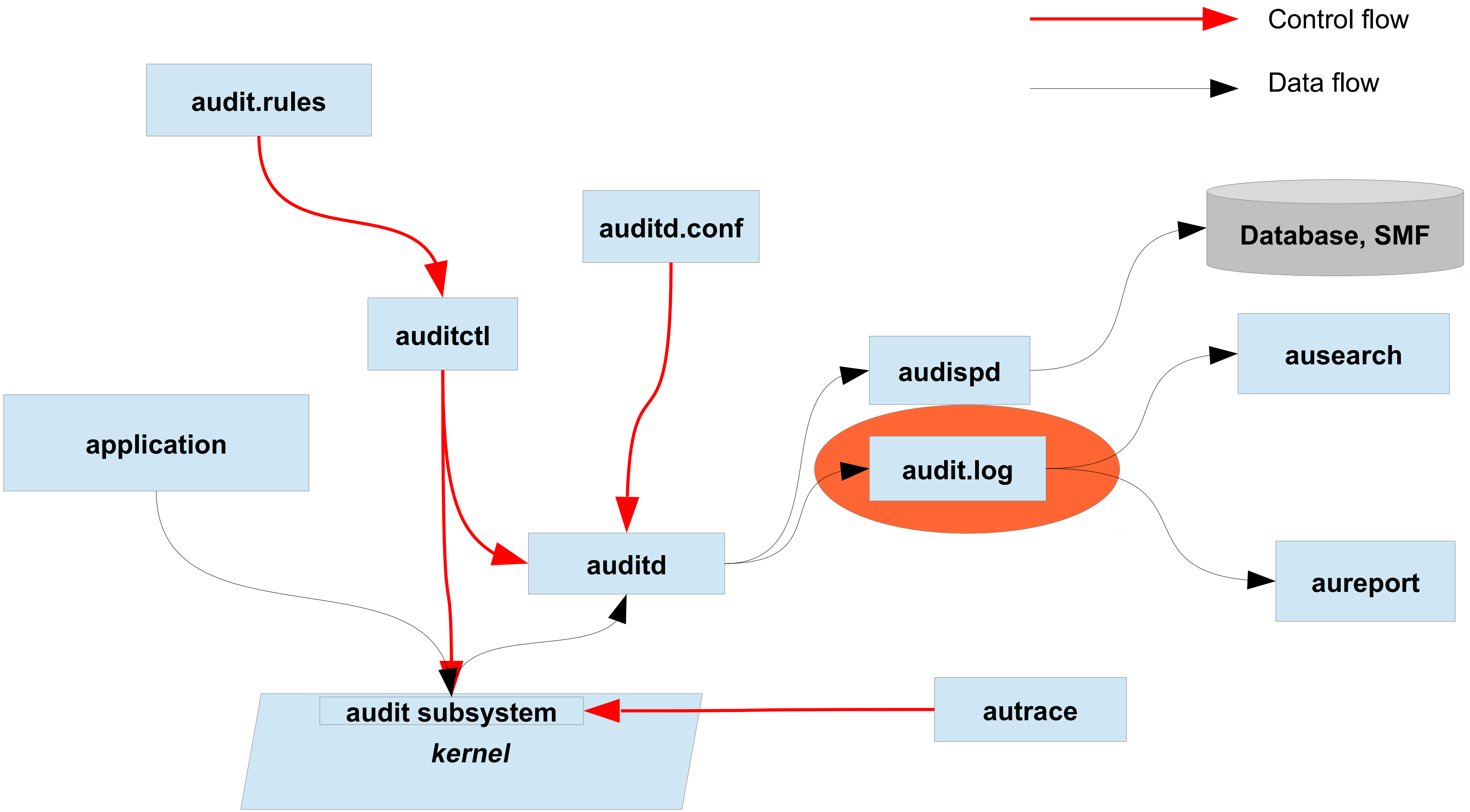
auditctl -e to enable or disable audit
auditctl -f to control the failure flag
auditctl -r to control the rate limit for audit messages
auditctl -b to control the backlog limit
auditctl -s to query the current status of the audit daemon

-e, -f, -r and -b also for audit.rules

Example:

```
-D
-b 8192
-f 1
-e 1
-w /etc/shadow
-w /etc/passwd -k fk_passwd -p rwxa
-a entry,always -S mkdir
```

Linux Audit Framework components



Linux Audit Framework – logs and reports

- Auditd writes by default its log files into /var/log/audit/audit.log
- Logged info after a “vim /var/log/audit/audit.log”:

```
type=CWD msg=audit(1316683236.198:956):  cwd="/var/log/audit"
type=PATH msg=audit(1316683236.198:956): item=0 name="." inode=10374 dev=5e:01 mode=040700
ouid=0 ogid=0 rdev=00:00
type=SYSCALL msg=audit(1316683236.198:957): arch=80000016 syscall=5 per=400000 success=yes
exit=5 a0=801aa20a a1=0 a2=0 a3=1 items=1 ppid=31216 pid=31848 auid=0 uid=0 gid=0 euid=0 suid=0
fsuid=0 egid=0 sgid=0 fsgid=0 tty=pts0 ses=663 comm="vim" exe="/bin/vim-normal" key="LOG_audit"
type=CWD msg=audit(1316683236.198:957):  cwd="/var/log/audit"
```

- SELinux, when activated and enforcing, also logs data in the audit framework log file
- Verbosity controlled by audit rules defined in audit.rules. SuSE and RedHat ships pre-established rules based on the Common Criteria CAPP/LSPP or OSPP profiles.
- Audit.conf controls the behavior of the audit daemon:
  - What to do when the log files reached a certain size
  - What to do when running out of disk space
  - Where to write log files...

Anatomy of an audit record – 1/3

```
type=SYSCALL msg=audit(1316683236.198:957): arch=80000016 syscall=5 per=400000 success=yes
exit=5 a0=801aa20a a1=0 a2=0 a3=1 items=1 ppid=31216 pid=31848 auid=0 uid=0 gid=0 euid=0
suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=pts0 ses=663 comm="vim" exe="/bin/vim-normal"
key="LOG_audit"
type=CWD msg=audit(1316683236.198:957):  cwd="/var/log/audit"
```

- **`type=SYSCALL`**: category of event happening: here it's a SYSCALL; can be LOGIN, USER_{ACCT, START, END}, AVC, {ADD,DEL}_{GROUP, USER}....

  Complete list can be found at https://fedorahosted.org/audit/browser/trunk/lib/msg_typetab.h

- **`msg=audit(1316683236.198:957)`**: ID of the audited event, based on timestamp and # of events since startup.

  Timestamp is a number of seconds since Epoch (use **`date(1)`** to convert it to a human readable format):

```
$ date --date='@1316683236.198'
Thu Sep 22 11:20:36 CEST 2011)
```

Anatomy of an audit record – 2/3

```
type=SYSCALL msg=audit(1316683236.198:957): arch=80000016 syscall=5 per=400000 success=yes
exit=5 a0=801aa20a a1=0 a2=0 a3=1 items=1 ppid=31216 pid=31848 auid=0 uid=0 gid=0 euid=0
suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=pts0 ses=663 comm="vim" exe="/bin/vim-normal"
key="LOG_audit"
type=CWD msg=audit(1316683236.198:957):  cwd="/var/log/audit"
```

- **`arch=8000016`**: ELF(5) architecture flags – s390x here.
  - Architectures supported by LAF are: alpha, armeb, i386, ia64, ppc, s390, s390x, x86_64 (c000003e)

- **`syscall=5 a0=801aa20a a1=0.. exit=5:`** syscall numerical IDs, parameters and return code.
  - Correspondance between IDs and syscall name is found in https://fedorahosted.org/audit/browser/trunk/lib/*arch*_table.h
  - List is architecture dependent, see for example: https://fedorahosted.org/audit/browser/trunk/lib/s390x_table.h
  - In our case here, syscall 5 is open(2)

Anatomy of an audit record – 3/3

```
type=SYSCALL msg=audit(1316683236.198:957): arch=80000016 syscall=5 per=400000 success=yes
exit=5 a0=801aa20a a1=0 a2=0 a3=1 items=1 ppid=31216 pid=31848 auid=0 uid=0 gid=0 euid=0
suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=pts0 ses=663 comm="vim" exe="/bin/vim-normal"
key="LOG_audit"
type=CWD msg=audit(1316683236.198:957):  cwd="/var/log/audit"
```

- **`comm=”vim”:`** command typed.

- **`exe=”/bin/vim-normal”:`** real executable launched.

- **`uid, gid:`** user and group ID used when launching the command, effective user and group IDs...

- **`pid, ppid`**: Parent Process ID and Process ID of the command launched.

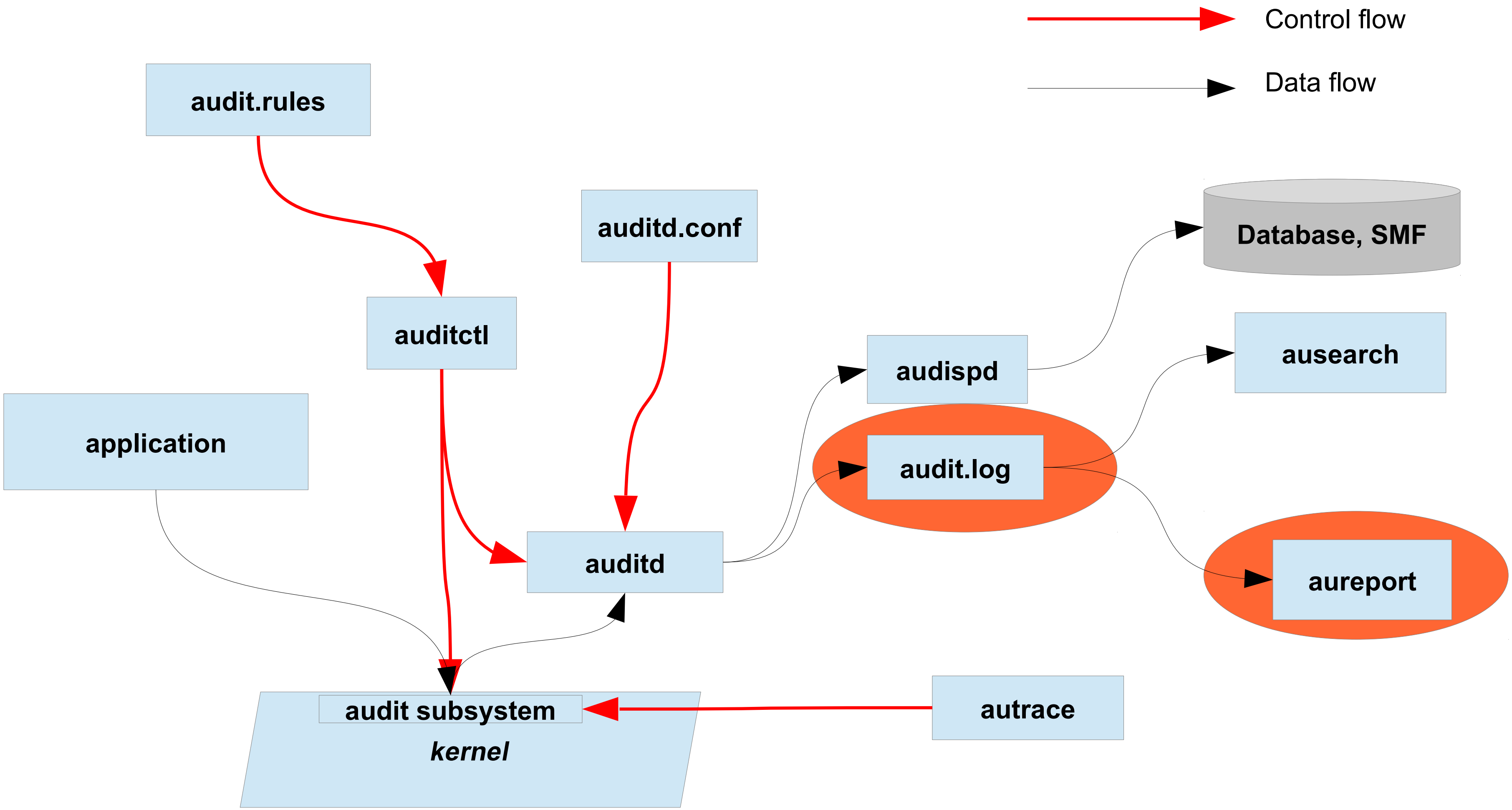
Linux Audit Framework – SELinux event record

- SELinux logs some additional events called “AVC” (Access Vector Cache):

```
type=AVC msg=audit(1346747742.883:20914): avc:  denied  { getattr } for  pid=2645 comm="httpd"
path="/home/guigui" dev=dasda1 ino=39443 scontext=unconfined_u:system_r:httpd_t:s0
tcontext=unconfined_u:object_r:user_home_dir_t:s0 tclass=dir
type=SYSCALL msg=audit(1346747742.883:20914): arch=80000016 syscall=107 per=400000 success=no
exit=-13 a0=2aae0c47a20 a1=3fffff7df80 a2=3fffff7df80 a3=3fffd263cc0 items=0 ppid=2641
pid=2645 auid=0 uid=48 gid=48 euid=48 suid=48 fsuid=48 egid=48 sgid=48 fsgid=48 tty=(none)
ses=1 comm="httpd" exe="/usr/sbin/httpd" subj=unconfined_u:system_r:httpd_t:s0 key=(null)
```

- SELinux event records include information about the SELinux context
- Here in this example, the lstat(2) syscall (which id is 107 on s390x) failed when the httpd process tried to access a file in my home directory.
- That's because SELinux boolean httpd_enable_homedirs is off, denying access by the Apache daemon to user home directories.

Linux Audit Framework components



Linux Audit Framework – logs and reports

- Set of userland tools to query the logs files and generate reports: ausearch and aureport

```
[root@lxclust2 tp]# aureport

Summary Report
======================
Range of time in logs: 03/19/2012 14:48:09.049 - 09/04/2012 14:01:01.533
Selected time for report: 03/19/2012 14:48:09 - 09/04/2012 14:01:01.533
Number of changes in configuration: 6
Number of changes to accounts, groups, or roles: 9
Number of logins: 5
Number of failed logins: 1
Number of authentications: 13
Number of failed authentications: 7
Number of users: 2
Number of terminals: 8
Number of host names: 2
Number of executables: 12
Number of files: 1
Number of AVC's: 4
Number of MAC events: 2
Number of failed syscalls: 4
Number of anomaly events: 0
Number of responses to anomaly events: 0
Number of crypto events: 63
Number of keys: 0
Number of process IDs: 111
Number of events: 470
```

Linux Audit Framework – report on authentication

```
[root@lxclust2 tp]# aureport -au

Authentication Report
============================================
# date time acct host term exe success event
============================================
1. 09/04/2012 08:38:46 root 10.254.1.10 ? /usr/sbin/sshd yes 21466
2. 09/04/2012 08:38:46 root 10.254.1.10 ? /usr/sbin/sshd yes 21467
3. 09/04/2012 08:38:46 root 10.254.1.10 ssh /usr/sbin/sshd yes 21470
4. 09/04/2012 08:38:57 root ? console /bin/login yes 21486
5. 09/04/2012 09:02:14 root 10.254.1.10 ssh /usr/sbin/sshd no 21528
6. 09/04/2012 09:02:17 root 10.254.1.10 ssh /usr/sbin/sshd yes 21529
7. 09/04/2012 09:02:17 root 10.254.1.10 ssh /usr/sbin/sshd yes 21532
8. 09/04/2012 09:02:42 root 10.254.1.10 ssh /usr/sbin/sshd no 21554
9. 09/04/2012 09:02:46 root 10.254.1.10 ssh /usr/sbin/sshd no 21555
10. 09/04/2012 09:02:46 root 10.254.1.10 ssh /usr/sbin/sshd no 21556
11. 09/04/2012 09:02:52 root 10.254.1.10 ssh /usr/sbin/sshd no 21557
12. 09/04/2012 09:02:52 root 10.254.1.10 ssh /usr/sbin/sshd no 21558
13. 09/04/2012 09:02:58 root 10.254.1.10 ssh /usr/sbin/sshd yes 21559
14. 09/04/2012 09:02:58 root 10.254.1.10 ssh /usr/sbin/sshd yes 21562
15. 09/04/2012 09:47:05 root 10.254.1.10 ssh /usr/sbin/sshd no 20822
16. 09/04/2012 09:47:10 root 10.254.1.10 ssh /usr/sbin/sshd yes 20823
17. 09/04/2012 09:47:10 root 10.254.1.10 ssh /usr/sbin/sshd yes 20826
18. 09/04/2012 09:55:56 root 10.254.1.10 ssh /usr/sbin/sshd yes 20846
19. 09/04/2012 09:55:56 root 10.254.1.10 ssh /usr/sbin/sshd yes 20849
20. 09/04/2012 10:11:25 guigui ? pts/0 /bin/su yes 20886
```

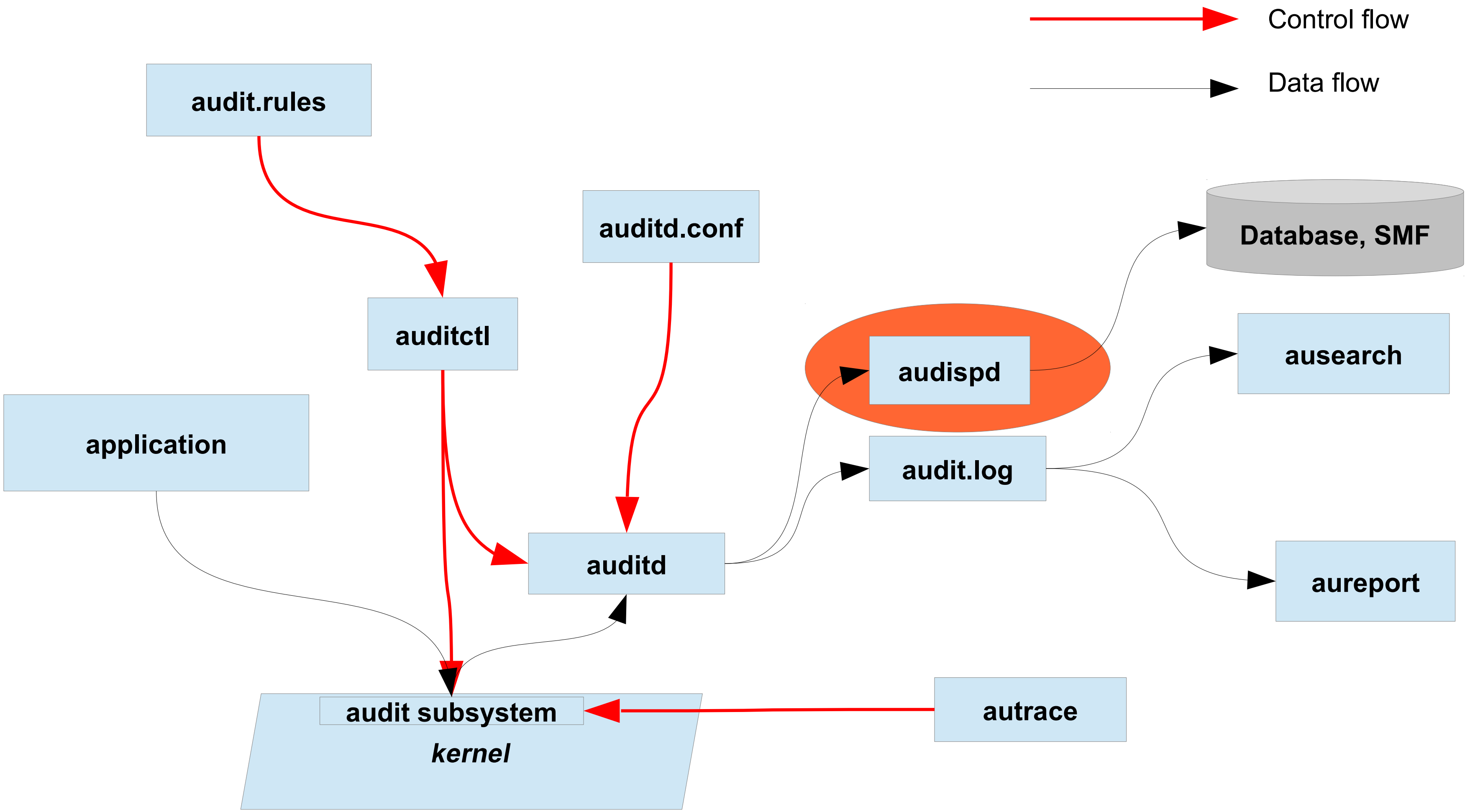
Linux Audit Framework – report on login

Login report for given time interval and logs contained in myfile

```
aureport -l -ts 14:00 -te 15:00 -if myfile
Login Report
=====================================================
# date time auid host term exe success event
=====================================================
1. 17/02/09 14:21:09 root: 192.168.2.100 sshd /usr/sbin/sshd no 7718
2. 17/02/09 14:21:15 0 jupiter /dev/pts/3 /usr/sbin/sshd yes 7724
```

And there are many more possibilities . . .

Linux Audit Framework components



Linux Audit Framework – SMF backend

- Auditd can make use of an event multiplexer, called audispd, to provide audit log data to external tools for analysis.
  - Even real time analysis might be possible
- Plugin-based architecture to interface with external tools.
- One of the plugins is audispd-zos-remote.

- audispd-zos-remote is a remote-auditing plugin for the Linux Audit framework. It forwards audit data to a z/OS SMF database, through IBM Tivoli Directory Server configured for Remote Audit Service.

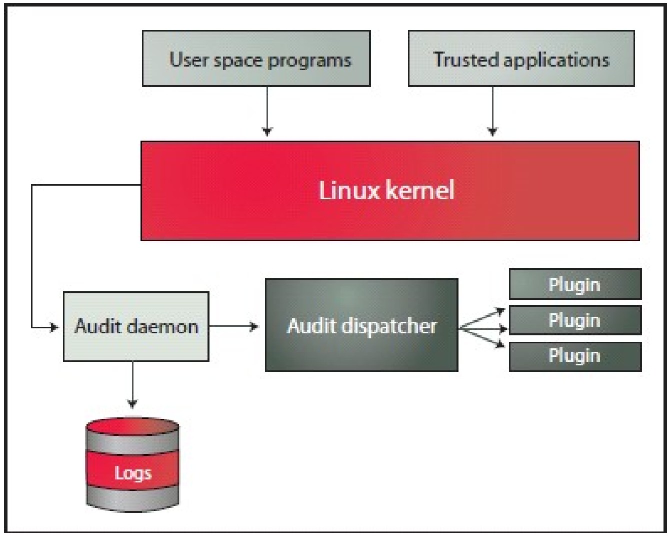
- Works with z/OS and z/VM. Additional requirements on z/VM includes HLASM to customize RACF.

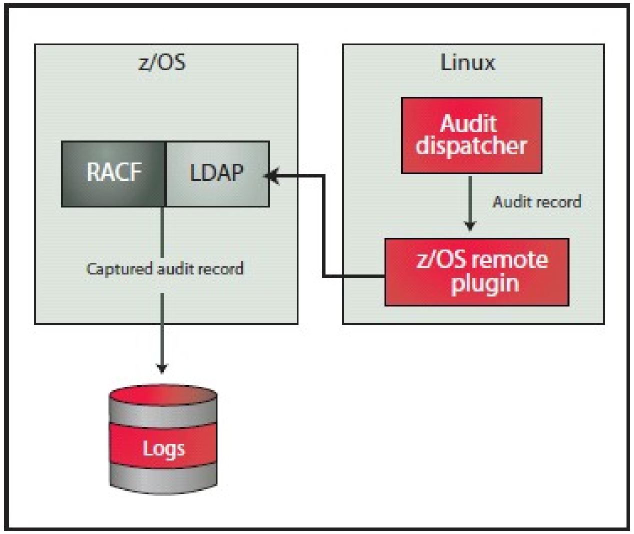
- The ITDS Remote Audit service will cut SMF records of type 83 subtype 4 everytime it processes a request. This plugin will issue a remote audit request for every incoming Linux Audit record (meaning that one Linux record will map to one SMF record), and fill this type's records.
- For more details about the SMF record format, refer to audispd-zos-remote(8) man page.

/etc/audisp/audispd.conf	for config of audit dispatcher
/etc/audisp/plugins.d	contains config of plugins

z/OS Configuration

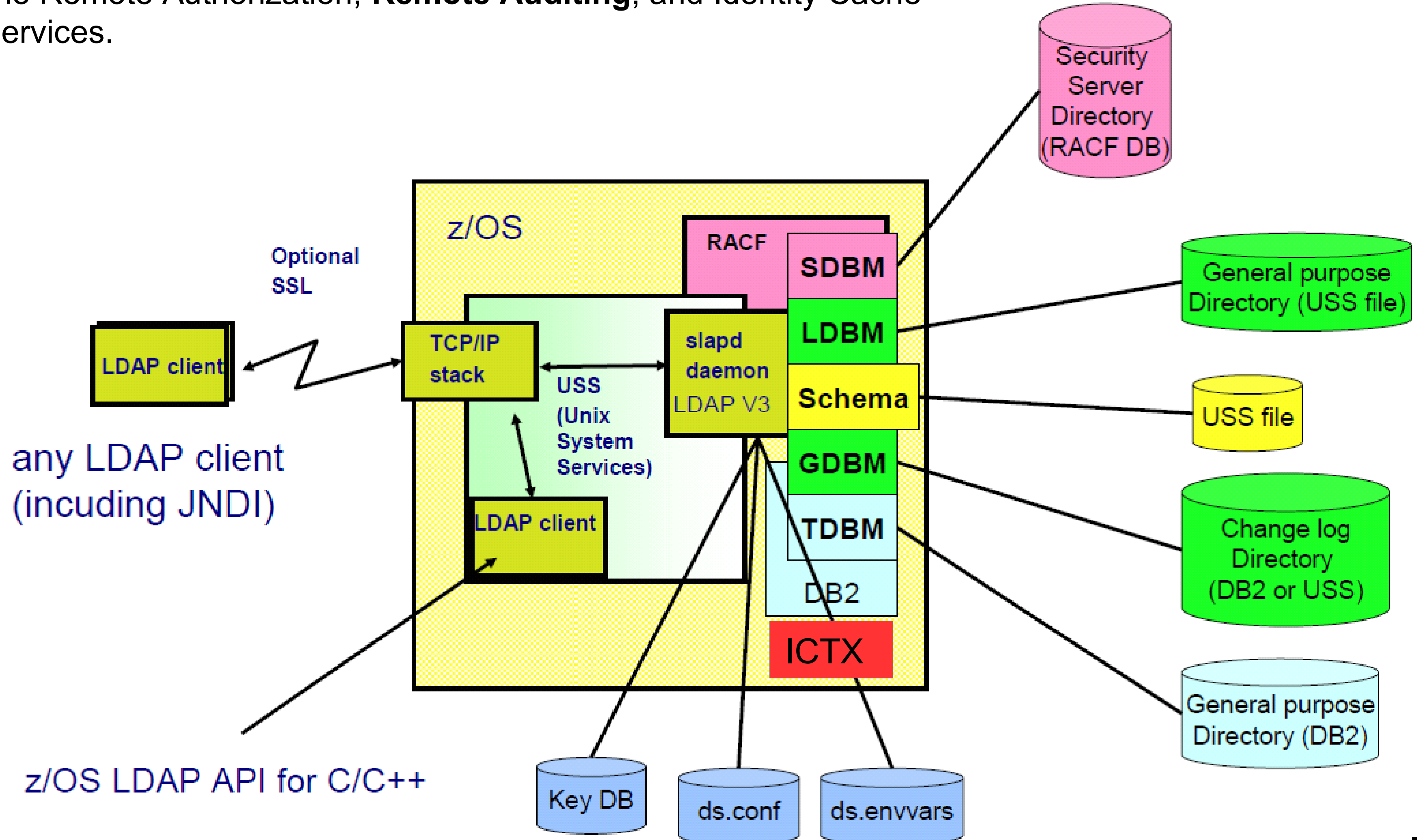
- What do we need to configure on z/OS?
  - z/OS IBM Tivoli Directory Server (ITDS)
    - Configured for Remote Services Support
  - RACF Configuration
    - Profiles authorization for working with Remote Services





z/OS IBM Tivoli Directory Server

- z/OS ITDS Remote Services:
  - The extended operation (EXOP) ICTX backend interfaces with RACF for the Remote Authorization, **Remote Auditing**, and Identity Cache services.


Security Server Directory (RACF DB)
z/OS
RACF
SDBM
Optional SSL
LDAP client
TCP/IP stack
slapd daemon
LDAP V3
LDBM
General purpose Directory (USS file)
Schema
USS file
USS (Unix System Services)
GDBM
any LDAP client (incuding JNDI)
LDAP client
TDBM
Change log Directory (DB2 or USS)
DB2
ICTX
General purpose Directory (DB2)
z/OS LDAP API for C/C++
Key DB
ds.conf
ds.envvars

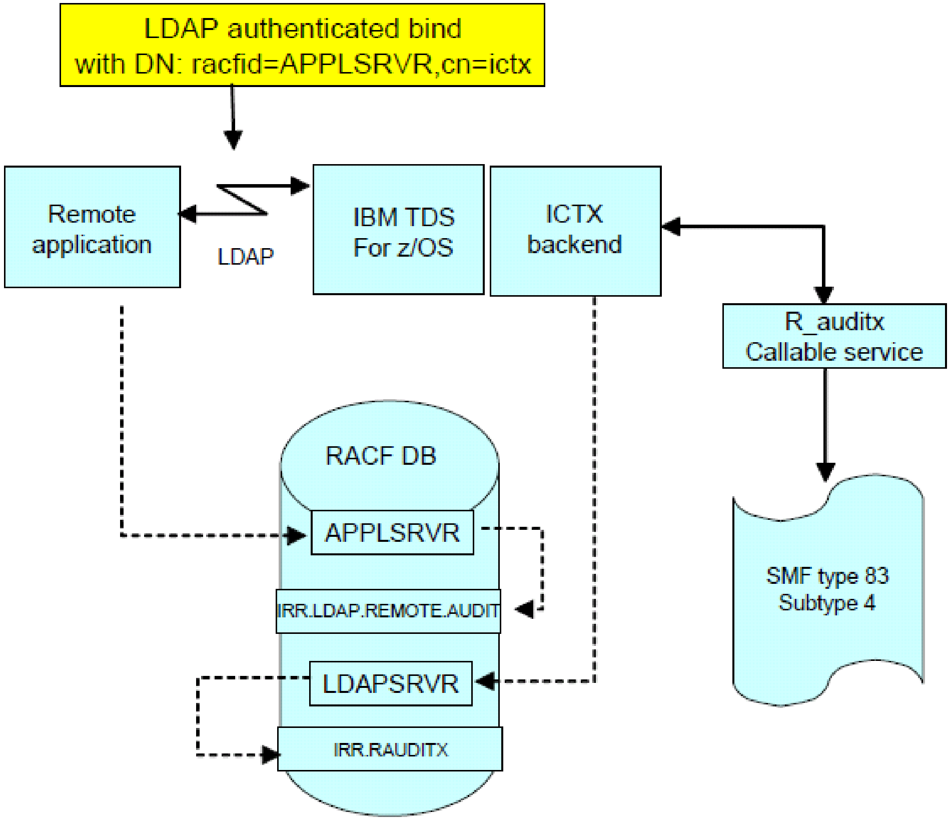
z/OS ITDS Remote Auditing

The ICTX backend has a suffix fixed by IBM: **cn=ictx.**

It requires an authenticated LDAP bind by the application calling the service.

The authenticated bind is done using a distinguished name that contains a RACF **userID** in the format indicated below, along with a RACF **password**:

- DN: racfid=<RACF userID>,cn=ictx



Remote auditing is performed via the ICTX backend which receives parameters sent by the requesting client application and uses them to invoke the R_auditx RACF callable service.

The R_auditx service generates an **SMF type 83 subtype 4** record that contains the requestor-provided information.

z/OS ITDS Remote Auditing Flow

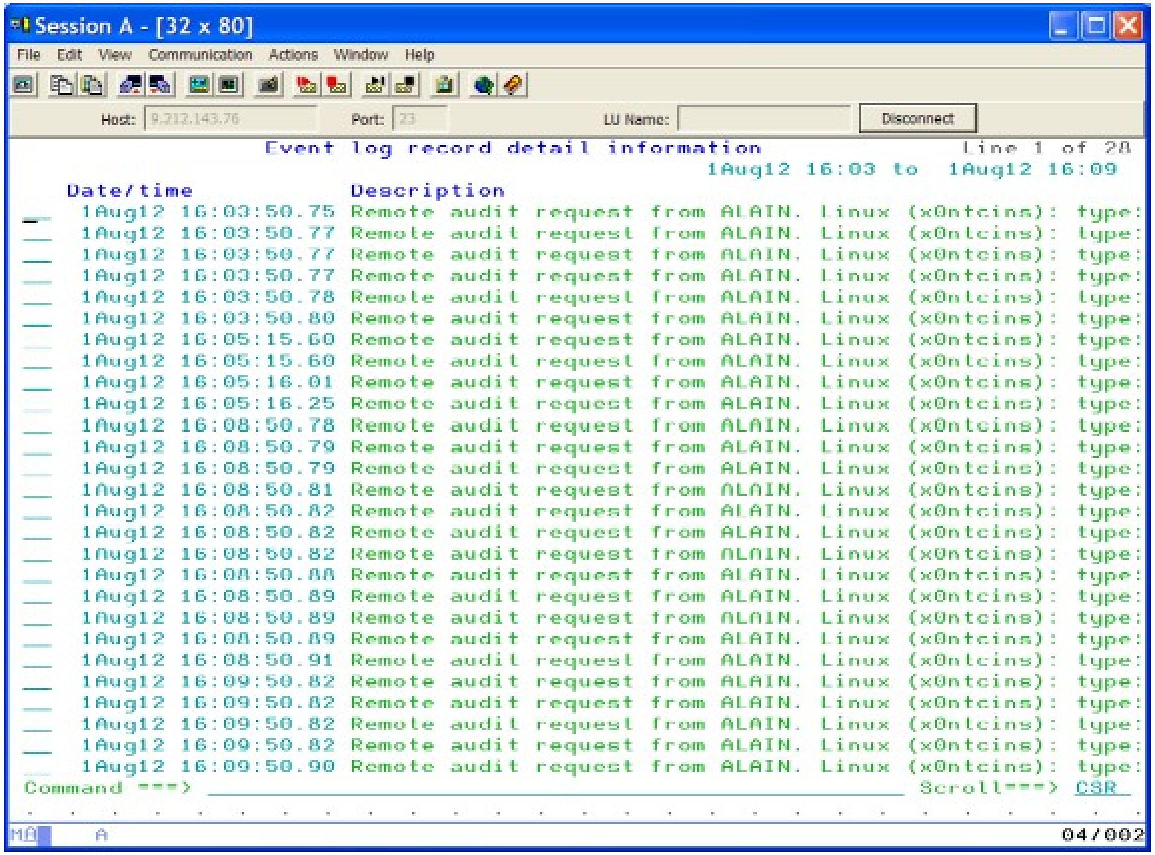
- The application must perform a simple bind to the server using an authorized racfid=userid,cn=ictx bind distinguished name.
- The application must build a DER-encoded extended operation request having the defined ASN.1 syntax that is specific to the audit request.
- The z/OS IBM TDS receives the request and routes it to the ICTX component, where it is decoded and processed.
- ICTX verifies the correct syntax and the requestor's authority before invoking the SAF audit service to satisfy the request.
- The result of the SAF service is a DER-encoded response that LDAP returns.
- The application must decode the response in order to interpret the results.

z/OS ITDS RACF Profiles

- Define the LINUX CDT Class
  - REDEFINE CDT @LINUX CDT INFO(POSIT(493) FIRST(ALPHA,NATIONAL,NUMERIC,SPECIAL) OTHER(ALPHA,NATIONAL,NUMERIC,SPECIAL) RACLIST(REQUIRED) CASE(ASIS) GENERIC(ALLOWED) DEFAULTUACC(NONE) MAXLENGTH(246))
  - RDEFINE @LINUX * UACC(NONE) AUDIT(ALL(READ))
  - SETR RACLIST(@LINUX) REFRESH
- Authorize the Bind userID to the IRR.LDAP.REMOTE.AUDIT RACF profile
  - RDEFINE FACILITY IRR.LDAP.REMOTE.AUDIT UACC(NONE)
  - PERMIT IRR.LDAP.REMOTE.AUDIT CLASS(FACILITY) ID(ALAIN) ACCESS(READ)
- Authorize the LDAP server userID to the IRR.RAUDITX RACF profile
  - RDEFINE FACILITY IRR.RAUDITX UACC(NONE)
  - PERMIT IRR.RAUDITX CLASSS(FACILITY) ID(GLDSRV) ACCESS(READ)

Exploiting Linux audit records with zSecure

- Using standard reports in zSecure Audit:

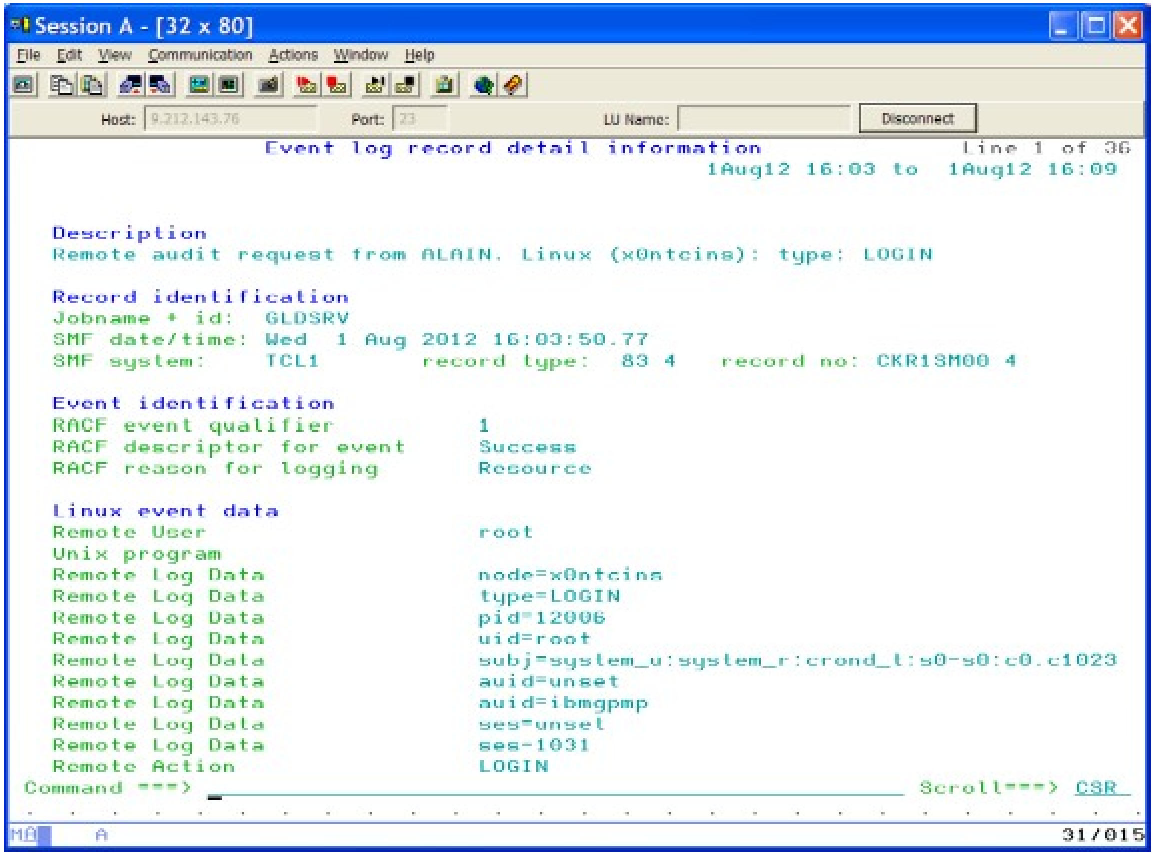


Host: 9.212.143.76 Port: 23 LU Name:

```
                  Event log record detail information             Line 1 of 28
                                                    1Aug12 16:03 to  1Aug12 16:09
   Date/time               Description
_   1Aug12 16:03:50.75 Remote audit request from ALAIN. Linux (x0ntcins): type:
_   1Aug12 16:03:50.77 Remote audit request from ALAIN. Linux (x0ntcins): type:
_   1Aug12 16:03:50.77 Remote audit request from ALAIN. Linux (x0ntcins): type:
_   1Aug12 16:03:50.77 Remote audit request from ALAIN. Linux (x0ntcins): type:
_   1Aug12 16:03:50.78 Remote audit request from ALAIN. Linux (x0ntcins): type:
_   1Aug12 16:03:50.80 Remote audit request from ALAIN. Linux (x0ntcins): type:
_   1Aug12 16:05:15.60 Remote audit request from ALAIN. Linux (x0ntcins): type:
_   1Aug12 16:05:15.60 Remote audit request from ALAIN. Linux (x0ntcins): type:
_   1Aug12 16:05:16.01 Remote audit request from ALAIN. Linux (x0ntcins): type:
_   1Aug12 16:05:16.25 Remote audit request from ALAIN. Linux (x0ntcins): type:
_   1Aug12 16:08:50.78 Remote audit request from ALAIN. Linux (x0ntcins): type:
_   1Aug12 16:08:50.79 Remote audit request from ALAIN. Linux (x0ntcins): type:
_   1Aug12 16:08:50.79 Remote audit request from ALAIN. Linux (x0ntcins): type:
_   1Aug12 16:08:50.81 Remote audit request from ALAIN. Linux (x0ntcins): type:
_   1Aug12 16:08:50.82 Remote audit request from ALAIN. Linux (x0ntcins): type:
_   1Aug12 16:08:50.82 Remote audit request from ALAIN. Linux (x0ntcins): type:
_   1Aug12 16:08:50.82 Remote audit request from ALAIN. Linux (x0ntcins): type:
_   1Aug12 16:08:50.88 Remote audit request from ALAIN. Linux (x0ntcins): type:
_   1Aug12 16:08:50.89 Remote audit request from ALAIN. Linux (x0ntcins): type:
_   1Aug12 16:08:50.89 Remote audit request from ALAIN. Linux (x0ntcins): type:
_   1Aug12 16:08:50.89 Remote audit request from ALAIN. Linux (x0ntcins): type:
_   1Aug12 16:08:50.91 Remote audit request from ALAIN. Linux (x0ntcins): type:
_   1Aug12 16:09:50.82 Remote audit request from ALAIN. Linux (x0ntcins): type:
_   1Aug12 16:09:50.82 Remote audit request from ALAIN. Linux (x0ntcins): type:
_   1Aug12 16:09:50.82 Remote audit request from ALAIN. Linux (x0ntcins): type:
_   1Aug12 16:09:50.82 Remote audit request from ALAIN. Linux (x0ntcins): type:
_   1Aug12 16:09:50.90 Remote audit request from ALAIN. Linux (x0ntcins): type:
Command ===> ____________________________________________________ Scroll===> CSR
MA      A                                                                 04/002
```

Exploiting Linux audit records with zSecure

- Example for a LOGIN event:



```
                  Event log record detail information              Line 1 of 36
                                                 1Aug12 16:03 to  1Aug12 16:09


   Description
   Remote audit request from ALAIN. Linux (x0ntcins): type: LOGIN

   Record identification
   Jobname + id:   GLDSRV
   SMF date/time: Wed  1 Aug 2012 16:03:50.77
   SMF system:    TCL1       record type:   83 4   record no: CKR1SM00 4

   Event identification
   RACF event qualifier          1
   RACF descriptor for event     Success
   RACF reason for logging       Resource

   Linux event data
   Remote User                   root
   Unix program
   Remote Log Data               node=x0ntcins
   Remote Log Data               type=LOGIN
   Remote Log Data               pid=12006
   Remote Log Data               uid=root
   Remote Log Data               subj=system_u:system_r:crond_t:s0-s0:c0.c1023
   Remote Log Data               auid=unset
   Remote Log Data               auid=ibmgpmp
   Remote Log Data               ses=unset
   Remote Log Data               ses=1031
   Remote Action                 LOGIN
 Command ===> _________________________________________________ Scroll===> CSR
MA    A                                                                  31/015
```

THANK YOU