Spårbarhet i operativsystem

Mer om spårbarhet än du någonsin velat veta

Martin Englund
Code Nursery AB

Agenda

- Bakgrund
- Terminologi
- Implementering av spårbarhet
- Diverse problem
- Analys av spårbarhetsloggar

Bakgrund

- Varför spårbarhet?
- Svarar på frågan:
  - Vem gjorde vad med vem och när?
- Härstammar från Orange Book (ur DOD Rainbow series)
  - Ett krav för att uppnå C2
  - Common Criteria

Bakgrund

- Detta verkar komplicerat är det inte lättare att använda något av
  - logging shell
  - honey pot
  - keylogger
  - etc
- Nej, inte om du bryr dig om **full** spårbarhet!
  - Hur får ovanstående program reda på vad ett program jag kör gör?

Bakgrund

- Loggning av händelser
  - Systemanrop
    - `execve(2), accept(2), open(2), write(2)`
  - Administrativa händelser
    - su login, reboot
  - Applikationshändelser
    - inetd ratelimit

Bakgrund

- Vilka operativsystem har den spårbarhet som beskrivs i denna presentation?
  - Solaris (2.3 och senare)
  - Mac OS X (10.3.6 och senare)
  - FreeBSD (7 beta1 och förmodligen 6.3)
  - Linux (RH & Fedora) m.h.a. OpenBSM
- ~~BSM~~
  - Basic Security Module

Agenda

- Bakgrund
- Terminologi
- Implementering av spårbarhet
- Diverse problem
- Analys av spårbarhetsloggar

Terminologi

- Subjekt
  - En enhet som agerar på ett objekt, vanligtvis en användare, d.v.s. en process
- Objekt
  - Det som ett subjekt har som mål för sina handlingar
  - T.ex. en fil, en enhet, en nätverks-socket

Mer terminologi

- Audit Trail
  - Resultatet av de aktiviteter som ett
  - en serie av *audit record*s
- Audit Record
  - En sekvens av *audit token*
- Audit Token
  - Definerar specifik information för ett *audit record*
  - T.ex. en sökväg, processinformation, ett subjekt

Ännu mer termininologi

- Audit Event
  - En enskild spårbarhetshändelse
  - T.ex. exekvering av program (`execve(2)`), öppning av en fil (`open(2)`)
- Audit Class
  - En gruppering av *audit event*
  - T.ex. filläsningshändelser, processhändelser

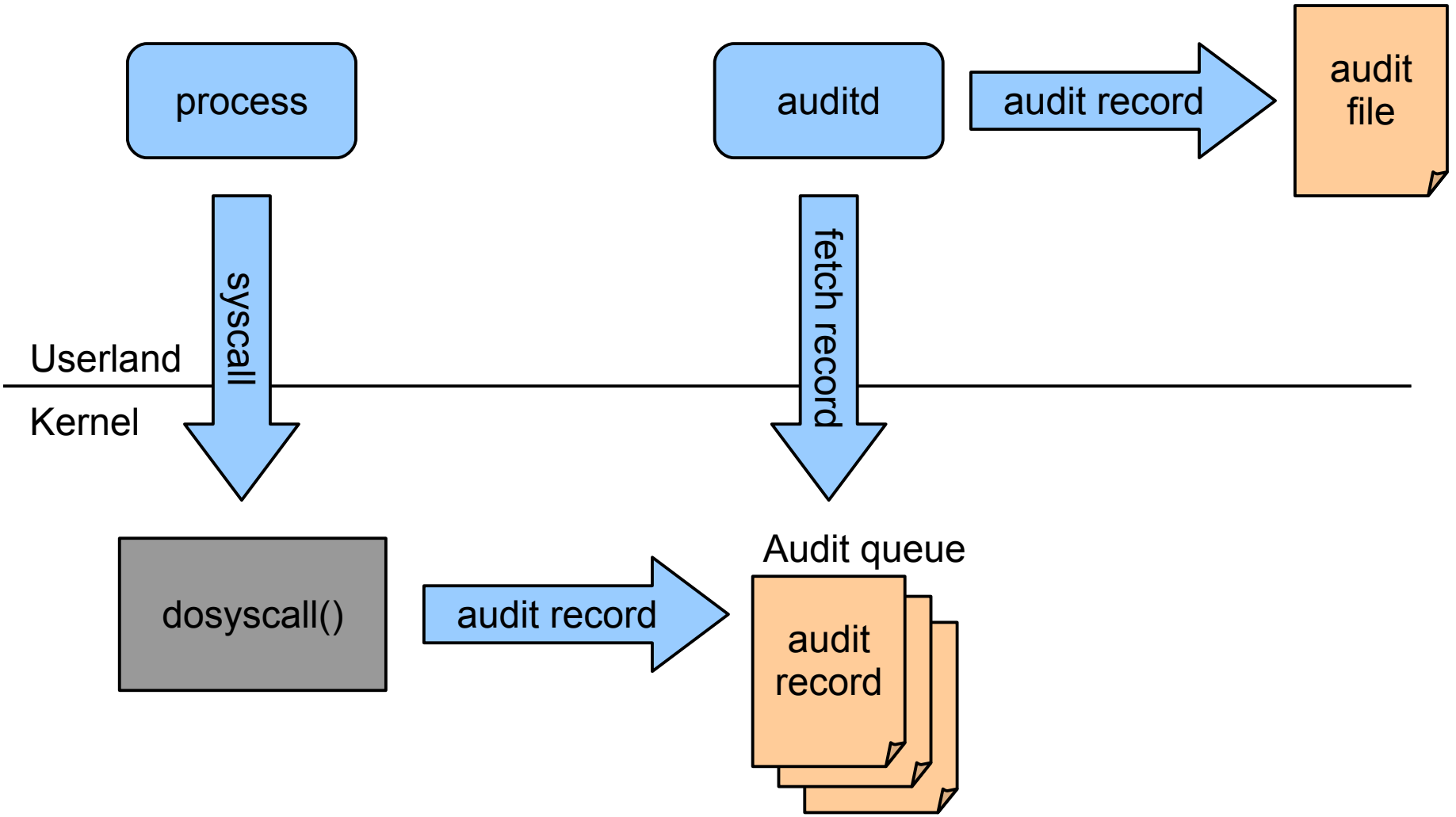
Ytterligare terminologi

- Audit Policy
  - Bestämmer vad som skall loggas
- Preselection mask
  - Avgör vilka *audit class*er som kommer att generera *audit record*s
  - En mask per process samt en för kärnan

Unik identifiering

- Hur särskiljer man flera inloggningar till samma
- användare?
- Varje *audit record* märks med
  - Audit Id
    - Följer användaren från inloggning till utloggning
    - Ändras inte vid “`su -`” eller exekvering av ett setuid-program
  - Session Id
    - Varje inloggning ges ett nytt id

Under huven


process
auditd
audit record
audit file
syscall
fetch record
Userland
Kernel
dosyscall()
audit record
Audit queue
audit record

Djupare under huven

```
+ dosyscall()
 |
 + syscall_entry()
 | |
 | + pre_syscall() (if t_pre_sys set)
 |   |
 |   + audit_start() (if audit_active set)
 |     |
 |     + au_init
 |     | |
 |     | + aui_*()
 |     + auditme() (to audit or not to audit)
 |     + au_start
 |       |
 |       + aus_*()
 |
...
 |
 |
 + syscall_exit()
 | |
 | + post_syscall()
 |   |
 |   + audit_finish() (if audit_active set)
 |     |
 |     + au_finish
 |       |
 |       + auf_*()
...
```

Agenda

- Bakgrund
- Terminologi
- Implementering av spårbarhet
- Diverse problem
- Analys av spårbarhetsloggar

Implementering av spårbarhet

- Börja med din policy
  - Vad är målet med spårbarheten?
  - Vad ska registreras?
  - Får det finnas luckor i spårbarheten?
  - Hur länge ska loggarna sparas?
  - Hur ska det efterbehandlas?
  - Vem skall analysera?

Luckor i spårbarheten

- Om maskinen är hårt lastad kan det hända att kön blir full
- Synkrona händelser
  - Blockera processen tills dess att resurser finns tillgängliga
  - Blockera inte processen, men räkna hur många audit records som kastas
- Asynkrona händelser
  - Ignorera händelsen
  - Stoppa systemet

Solaris

- För att starta
  - Kör programmet `/etc/security/bsmconv`
  - Starta om
- För att stänga av
  - `/etc/security/bsmunconv`

Mac OS X

- Kallas
  - Common Criteria Tools
- Kräver extra programvara
  - Laddas ner gratis från Apple (se länklista)
- Efter installation
  - Öppna `/etc/hostconfig` i en editor
  - Lägg till en rad med `AUDIT=-yes-`
  - Starta om

Filer

- audit_config
- audit_user
- audit_class
- audit_event
- audit_warn
- audit_startup

audit_class

- Innehåller definitionerna av spårbarhetsklasserna
  - `0x00000000:no:invalid class`
  - `0x00000001:fr:file read`
  - `0x00000002:fw:file write`
  - `0x00000004:fa:file attribute access`
  - `...`
  - `0x00100000:ps:process start/stop`
  - `0x00200000:pm:process modify`
  - `0x00300000:pc:process (meta-class)`

audit_event

- Inehåller mappningen av spårbarhetshändelser till spårbethetsklasser
  - `23:AUE_EXECVE:execve(2):ps,ex`
  - `33:AUE_ACCEPT:accept(2):nt`
  - `72:AUE_OPEN_R:open(2) - read:fr`
  - `73:AUE_OPEN_RC:open(2) - read,creat:fc,fr`
  - `76:AUE_OPEN_W:open(2) - write:fw`

audit_control

- Innehåller den övergripande konfigureringsinformationen
  - `dir: /var/audit`
  - `minfree: 20`
  - `flags: lo,ex,ad`
  - `naflags: na`

Att välja klasser

-
- cl
- +cl
- -cl
- ^cl
- ^+cl
- ^-cl

audit_user

- Inehåller konfigurering per användare
  - `user:always audit:never audit`
  - `martin:ex,lo,nt:-fr`

audit_warn

- Script som avgör vad som skall hända vid olika
  - `soft`
  - `allsoft`
  - `hard`
  - `allhard`

audit_startup

- Sätter policy vid boot
  - `auditconfig -setpolicy +argv`
  - `auditconfig -setpolicy +cnt`
  - `auditconfig -setpolicy +seq`
  - `auditconfig -setpolicy +zonename`

Program

- audit
  - `-n` (skapar ny loggfil)
  - `-s` (läser om konfigurationsfilerna)
  - `-v` *`audit_control`* (kontrollerar syntaxen)
- auditconfig
  - svc:/system/auditd:default
- bsmrecord
  - visar *audit record* formatet

Program

- auditconfig
  - -getcond

```
root@x2200# auditconfig -getcond
audit condition = auditing
```

  - -getpinfo pid

```
root@x2200# auditconfig -getpinfo $$
audit id = martin(300)
process preselection mask = ex,pm,aa,lo,nt,fd,fm,fw(0x4028112a,0x4028112a)
terminal id (maj,min,host) = 13360,71168,mbp(83.183.210.61)
audit session id = 2066870292
```

  - -getkmask

```
root@x2200# auditconfig -getkmask
audit flags for non-attributable events = lo,na(0x1400,0x1400)
```

  - -getpolicy

```
root@x2200# auditconfig -getpolicy
audit policies = arge,argv,cnt,group,zonename
```

Syslog plugin

- Gäller endast Solairs
  - Bara 1023 bytes långa meddelanden p.g.a. begränsningen i syslog/udp
- audit_config

```
plugin:
  name=audit_syslog.so;p_flags=lo,ex
```

- syslog.conf

```
audit.notice:     @loghost
```

Agenda

- Bakgrund
- Terminologi
- Implementering av spårbarhet
- Diverse problem
- Analys av spårbarhetsloggar

Loggar för lite eller för mycket

- Kontrollera följande filer:
  - `/etc/security/audit_control`

    `flags: ...`
  - `/etc/security/audit_user`

    `user: ...`
  - `/etc/security/audit_startup`

    `auditconfig -setpolicy ...`
  - `/var/spool/cron/crontabs/*.au`
    - rad två och tre

Loggar för lite eller för mycket

- Kör följande kommando

```
-auditconfig -getpinfo $$
audit id = martin(300)
process preselection mask =
ex,pm,aa,lo,nt,fd,fm,fw(0x4028112a,0x4028112a)
terminal id (maj,min,host) =
12922,5632,unknown(217.28.34.132)
audit session id = 3816290814

-auditconfig -getkmask
audit flags for non-attributable events =
lo,na(0x1400,0x1400)

-auditconfig -getpolicy
audit policies = arge,argv,cnt,group,zonename
```

Enorma filer

- Rotera loggen m.h.a.
  - `audit -n`
- En gång per vecka, dag eller timme beroende på aktivitet
- Kör från `cron` eller `logadm`

Fortfarande enorma filer

- Loggarna innehåller en hel del repeterad information
- Bra kompressionsfaktor
  - Ca 95% med gzip
- Ett alternativ är zfs med kompression
  - Mindre effektiv p.g.a. zljb är optimerad för snabbhet
  - Fortfarande ca 50% kompression

Filändringar över NFS loggas ej

- Om maskinen är NFS server så loggas ej klienternas förändringar av filer
  - Servern saknar tillräcklig information för att kunna skapa en fullständig händelse
- Logga på klienten istället
- Använd `nfslogd(1M)` om du inte kan lita på klienten

Panic p.g.a. “ahlt” policy

- Panic sträng `”non-attributable halt. should dump core”`

```
# mdb -k unix.0 vmcore.0
Loading modules: [ unix krtld genunix specfs ufs ip sctp usba fctl
lofs audiosup nfs random crypto ptm ]
> au_zone_key::print zone_key_t
0x4
> zone0::zsd 0x4 | ::print 'struct au_kcontext'
{
    auk_valid = 0x5a5a5a5a
    auk_zid = 0
    auk_hostaddr_valid = 1 (B_TRUE)
    auk_sequence = 0
    auk_auditstate = 0x1
    auk_output_active = 0
    auk_current_vp = 0
    auk_policy = 0x2001
    auk_queue = {
        head = 0xffffffff80a3eee0
        tail = 0xffffffff80a3ed18
        cnt = 0x2
```

Agenda

- Bakgrund
- Terminologi
- Implementering av spårbarhet
- Diverse problem
- Analys av spårbarhetsloggar

Analys av spårbarhetsloggar

- Som att hitta en nål i en höstack
- Två hjälpmedel
  - `auditreduce`
  - `praudit`
- Perl
- AuditAnalyzer

auditreduce

- Hackar och sorterar loggfilerna
- Otaliga filtreringsoptioner

`-a` *`tidpunkt`*	efter denna tidpunkt
`-b` *`tidpunkt`*	före denna tidpunkt
`-d` *`datum`*	bara denna dag
`-c` *`klass`*	bara händelser i denna klass
`-m` *`händelse`*	bara denna typ av händelse

auditreduce

- Fler optioner

`-M` *`maskin`*	bara denna maskin
`-e` *`användare`*	händelser med detta euid
`-r` *`användare`*	händelser med detta uid
`-u` *`användare`*	händelser med detta audit id
`-j` *`subjekt`*	händelser med detta subjekt id
`-z` *`zon`*	händelser i denna zon

auditreduce

- Ytterligare optioner
  - `-M` *`objekt=värde`* bara objekt som matchar
    - `file=sökväg`
    - `fileowner=användare`
    - `pid=process id`
    - `procowner=användare`
    - `sock=nätverksport`

auditreduce

- Exempel

```
auditreduce -a 200710271345 -c lo -u
  martin -z prod
auditreduce -d 20071027 -e root -o
  file=/etc/shadow
```

praudit

- Omvandlar binärformatet till två typer av textformat
  - “mänskligt läsbart “
  - XML
- OBS! Se till att ha en homogen konfiguration
  - `/etc/passwd, /etc/group`
  - `/etc/hosts`
  - `/etc/security/audit_class, /etc/security/audit_event`

praudit

- Exempel

```
auditreduce ... | praudit > audit.txt
auditreduce ... | praudit -x > audit.xml
```

- Formatet definierat i
  - `audit.log(4)`

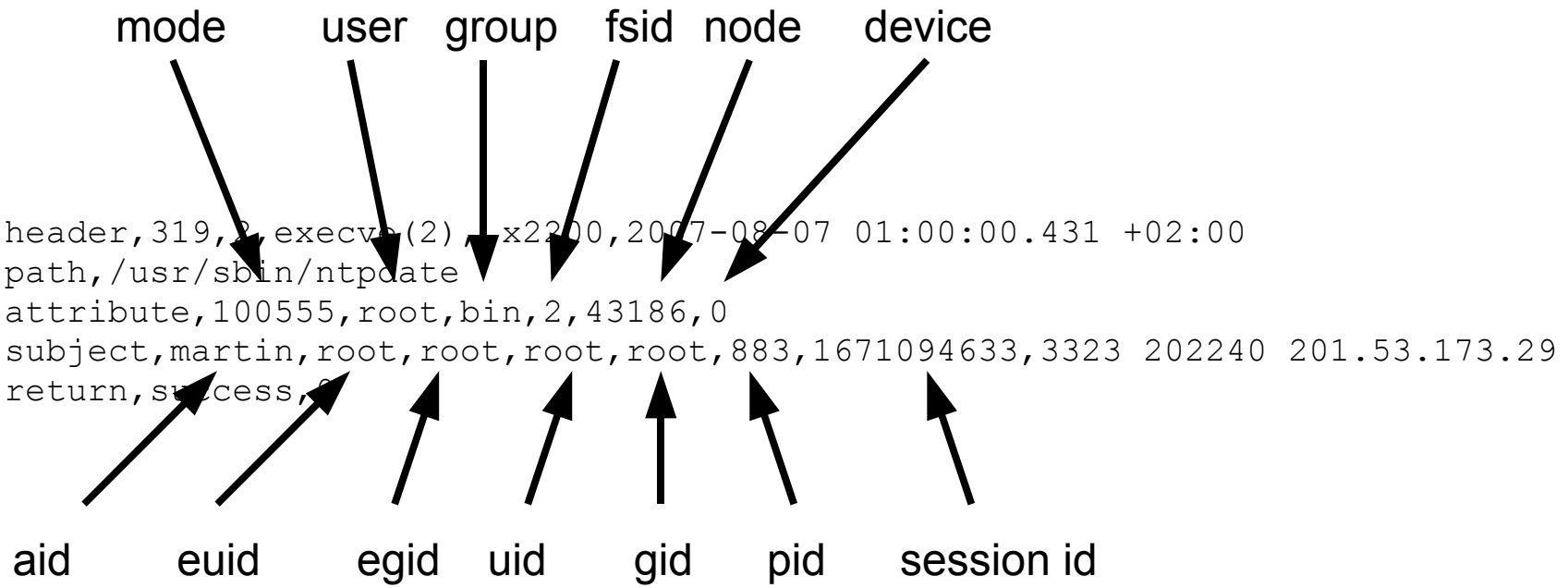
Exempel på loggar

```
header,319,2,execve(2),,x2200,2007-08-07 01:00:00.431 +02:00
path,/usr/sbin/ntpdate
attribute,100555,root,bin,2,43186,0
subject,martin,root,root,root,root,883,1671094633,3323 202240 201.53.173.29
return,success,0

header,182,2,sendto(2),,x2200,2007-08-07 01:00:00.540 +02:00
argument,1,0x3,so
argument,3,0x0,flags
socket,0x0200,0x0100,0x007b,0.0.0.0,0x007b,62.119.40.98
subject,martin,root,root,root,root,883,1671094633,3323 202240 201.53.173.29
return,success,48

header,191,2,open(2) - read,write,,x2200,2007-08-07 03:10:00.070 +02:00
path,/etc/logadm.conf
attribute,100644,root,sys,2,2114,0
subject,martin,root,root,root,root,978,1671094633,3323 202240 201.53.173.29
return,success,3
```

Exempel på loggar



Exempel på loggar

```
<record version="2" event="execve(2)" host="x2200"
        iso8601="2007-08-16 03:10:00.313 +02:00">
  <path>/usr/bin/chmod</path>
  <attribute mode="100555" uid="root" gid="bin" fsid="2" nodeid="297" device="0"/>
  <exec_args>
            <arg>/bin/chmod</arg><arg>644</arg><arg>/var/adm/messages</arg>
  </exec_args>
  <subject audit-uid="martin" uid="root" gid="root" ruid="root" rgid="root"
          pid="10341" sid="1671094633" tid="3323 202240 201.53.173.29"/>
  <return errval="success" retval="0"/>
</record>

<record version="2" event="chmod(2)" host="x2200"
        iso8601="2007-08-16 03:10:00.314 +02:00">
  <argument arg-num="2" value="0x1a4" desc="new file mode"/>
  <path>/var/adm/messages</path>
  <attribute mode="100666" uid="root" gid="root" fsid="2" nodeid="44733"
            device="0"/>
  <subject audit-uid="martin" uid="root" gid="root" ruid="root" rgid="root"
          pid="10341" sid="1671094633" tid="3323 202240 201.53.173.29"/>
  <return errval=</record>
```

Att arbeta med stora filer

- Eftersom data skrivs sekventiellt så är det svårt att följa en specifik användare
- Dela upp stora filer i flera mindre
  - Per zon
  - Per användare (audit id)
  - Per session (session id)

Vilka händelser är intressanta?

- Händelser som har
  - `aid != uid`
  - `aid != euid`
- Händelser som misslyckas med
  - return token, error value är
    - `EPERM`
    - `EACCES`

Vilka händelser är ointressanta?

- Lyckade händelser som har
  - `aid == uid == euid`
- Misslyckade händelser som har
  - return token, error value
    - `ENOENT`

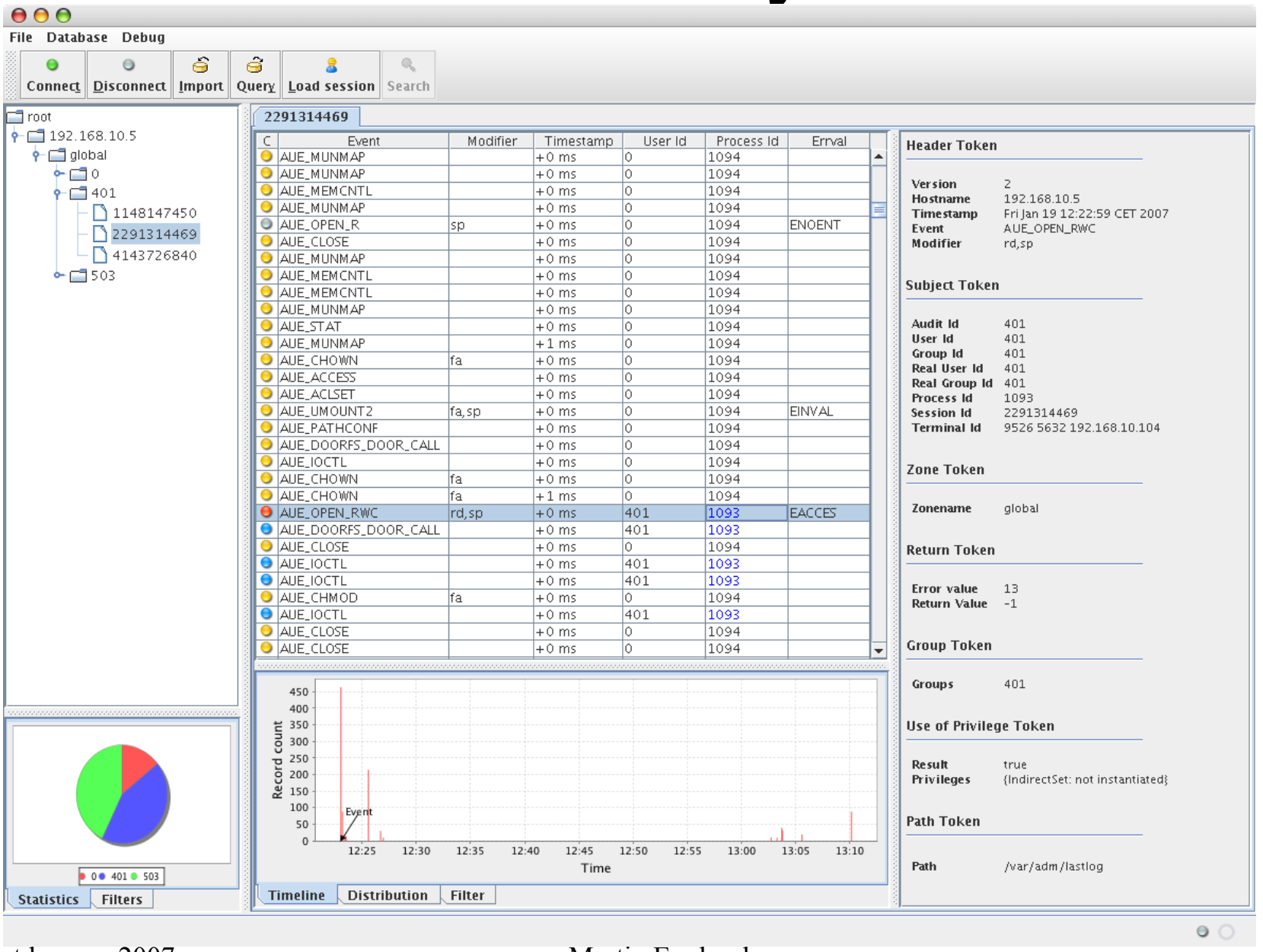
AuditAnalyzer

- Ett verktyg som är under utveckling
- `auditreduce` och `praudit` är inte skalbara
  - Oanvändbara i min vardag
    - 340 servrar
    - 170 GB audit-data per dag (bara `lo,ex`)
    - Sparas i **minst** ett år
    - ca 21 TB per år
- För att klara storskalig hantering händelser krävs databaslagring

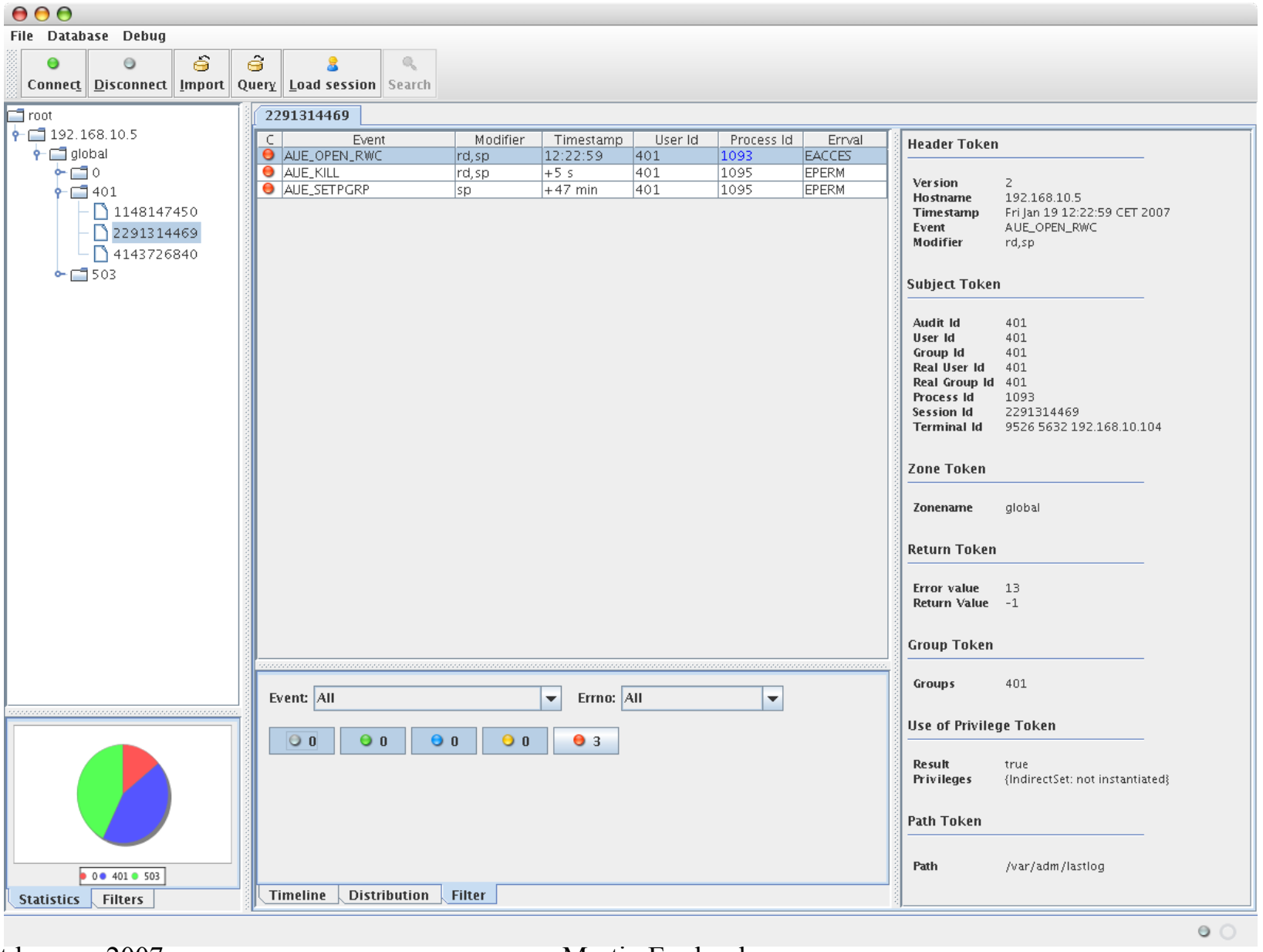
AuditAnalyzer

- Bygger på två arbetssätt
  - Session auditing
  - Forensics analysis

AuditAnalyzer



AuditAnalyzer



Frågor?

“The question is not if you are paranoid,
it is if you are paranoid enough.”

Länkar

- http://blogs.sun.com/martin/
- http://auditanalyzer.com/
- http://opensolaris.org/os/project/audit/
- http://docs.sun.com/app/docs/doc/816-4557/auditplan-6
- http://images.apple.com/support/security/commoncriteria/
- http://openbsm.org/