SELinux - Linux Audit Subsystem

Audit Subsystem

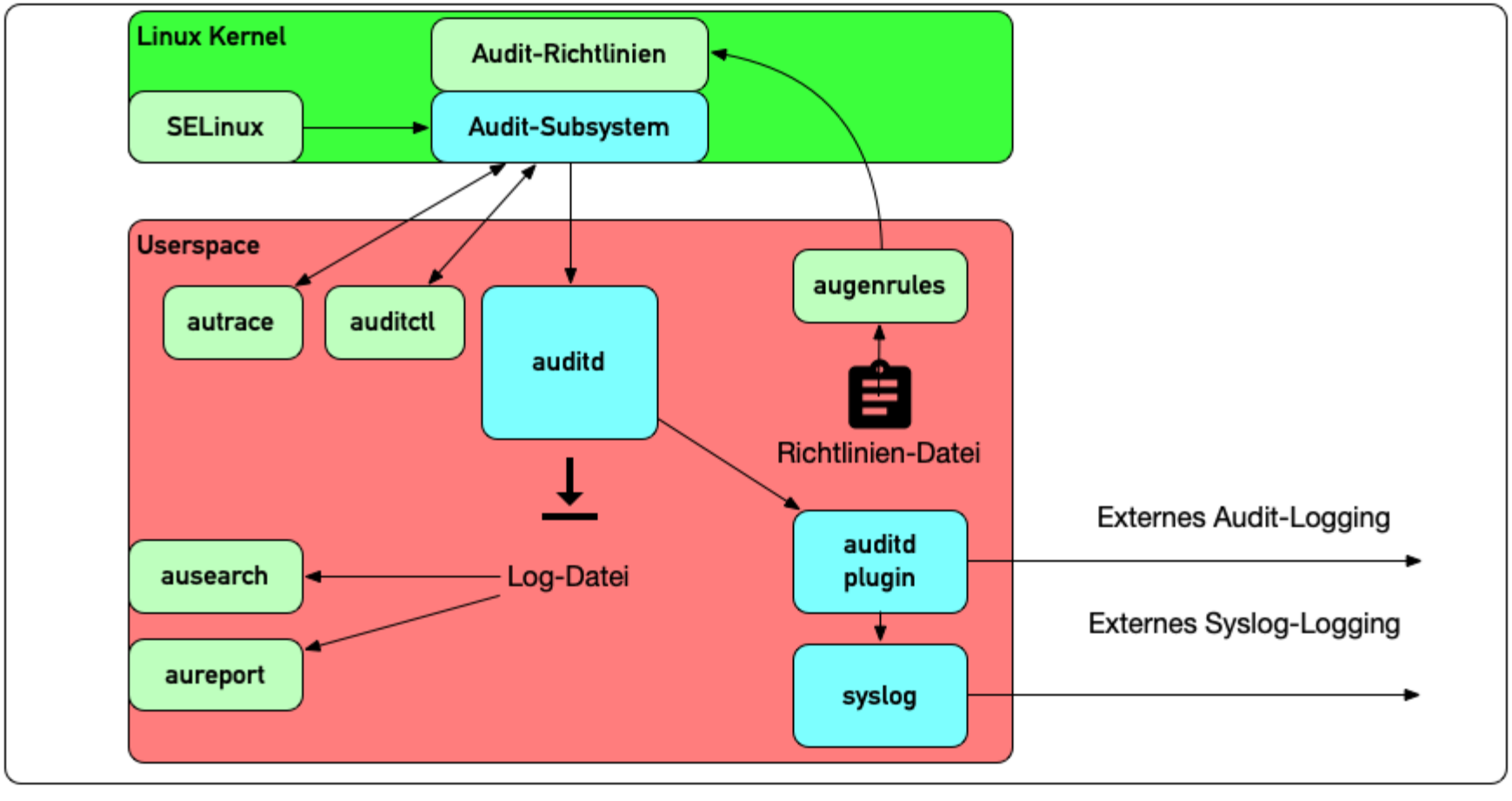
Das Linux Audit Subsystem

- Das Linux Audit Subsystem erlaubt es dem Betreiber eines Linux Systems, ein fein-granuläres Audit-Logging für System-Ereignisse festzulegen
    - Ereignisse von Linux-Security-Modulen (LSM) wie AppArmor oder SELinux
    - Ereignisse von sicherheitsrelevanten Anwendungen (SSH, Login-Programm)
    - Ereignisse welche von beliebigen Anwendungen ausgelöst werden und in der Audit-Subsystem Richtlinien-Konfiguration hinterlegt wurden (z.B. Systemcalls, Dateizugriffe, Netzwerk-Aktivitäten)

LSMs und das Audit Subsystem

- Der Betrieb eines Linux Security Moduls (LSM) ohne das Audit-Subsystem ist technisch möglich, aber oft nicht sinnvoll
  - Das LSM verhindert Zugriffe, diese werden aber nicht protokolliert so das der Betreiber des Systems keine Übersicht über die Wirksamkeit des LSM hat
  - Ein gut konfiguriertes Audit-Subsystem ist die Basis für den Einsatz von SELinux

Audit Subsystem Übersicht



Linux Audit Subsystem Konfiguration

Konfiguration des Audit-Daemon

- Die Konfigurationsdatei des Audit-Daemons unter `/etc/audit/auditd.conf` wurde vom Audit-Subsystem Projekt in den letzten 20 Jahren nicht aktualisiert und sollte daher vor der Inbetriebnahme des Audit-Subsystems (oder SELinux) auf sinnvolle Werte überprüft werden

Lokal vs. Remote Log Quelle

- Der Audit-Daemon kann Log-Events vom Linux-Kernel oder über das Netzwerk von einem anderen Linux System per Remote-Audit Logging empfangen (Remote Audit Logging ist ein anderes Protokoll als syslog)

```
local_events = yes # "no" to receive
                   # remote log-data only
```

- Die Standardeinstellung ist `yes`

Audit-Logdatei

- In der Standard-Konfiguration werden die Log-Daten in die Datei `/var/log/audit/audit.log` geschrieben. Für zentrale Audit-Logserver ist es sinnvoll wenn diese Datei auf einem separaten Dateisystem liegt

```
write_logs = yes  # set to "no" for
                  # only remote logging
log_file = /var/log/audit/audit.log
```

Gruppenrechte und Format der Audit-Logdatei

- Sollen die Log-Informationen von anderen Benutzer als dem Superuser `root` gelesen werden, so sollte die Gruppen-Zugehörigkeit der Log-Datei angepasst werden (z.B. auf die Gruppe `wheel`)

```
log_group = root
log_format = ENRICHED   # "enriched" or "raw"
```

- Beim Log-Format `enriched` werden die Log-Daten vom Audit-Daemon mit Metadaten versehen, welche die Auswertung der Log-Daten vereinfachen. Bei der Einstellung `raw` werden die Log-Daten wie vom Linux-Kernel gesendet gespeichert

Schreiben der Log-Datei

Anzahl Log-Dateien

- Der Audit-Daemon kann selbstständig die Log-Dateien *rotieren*, wenn die Log-Datei eine definierte Größe erreicht.
    - Anders als bei anderen Log-Rotate Programmen (z.B. BIND 9) werden überschüssige Log-Dateien nur beim (Neu-)Start des Audit-Daemons, oder bei einem `space-left` Event gelöscht(!)

```
max_log_file = 8   # Max Groesse der Log-Datei in MB
num_logs = 5       # Anzahl Generationen der Log-Datei
# Aktion beim Erreichen der max. Log-Groesse
max_log_file_action = ROTATE
```

Log-Host Metadaten

- Der Audit-Log-Daemon kann die Log-Informationen um den Rechnernamen des Quell-Hosts anreichern
    - Dies ist bei remote Audit-Logging hilfreich
    - `name` ist der Hostname / Domain, welche den Log-Daten hinzugefügt wird wenn bei `name_format` der Wert `user` eingetragen wurde

```
name_format = NONE
##name = mydomain
```

Log-Host Metadaten

- Mögliche Werte für `name_format`:

WERT	BESCHREIBUNG
NONE	Keine Quell-Host Metadaten
HOSTNAME	Der Hostname des Systems (gethostname)
FQDN	Voller Domain-Name durch DNS Auflösung
NUMERIC	IP(v4) Adresse des Hosts
USER	Wert des `name` Parameters

E-Mail Meldungen

- Der Audit-Daemon kann bei außergewöhnlichen Systemzuständen (Fehler im Dateisystem oder in der Storage Hardware, kein Platz auf dem Speichermedium) E-Mail Warnungen an die Administratoren versenden
  - `action_mail_acct` ist ein lokaler E-Mail Benutzer, oder eine externe E-Mail Adresse. Ein MTA Programm muss unter `/usr/lib/sendmail` installiert sein
  - `verify_email` prüft ob der Domain-Name der angegebenen E-Mail Adresse per DNS aufgelöst werden kann

```
verify_email = yes
action_mail_acct = root
```

"Space Left" Event

- Der `space_left` Wert gibt an, ab welchem Limit an freiem Speicher auf dem Speichermedium der Log-Datei der Audit-Daemon eine Warnung absetzen soll
    - In der Standard-Einstellung wird die Warnung in das `syslog` geschrieben
    - Der Wert ist in Megabyte (MB) und für modernen System wahrscheinlich zu gering (Empfehlung: 1000 MB). Der Wert kann auch in Prozent angegeben werden (Beispiel: `25%`)

```
space_left = 75
space_left_action = SYSLOG
```

"Admin Space Left" Event

- Der `admin_space_left` Wert gibt an, ab welchem Limit an freiem Speicher auf dem Speichermedium der Log-Datei der Audit-Daemon seine Arbeitsweise ändern soll. Dieser Wert sollte geringer als `space_left` sein.
    - In der Standard-Einstellung stellt der Audit-Daemon seine Arbeit ein
    - Der Wert ist in Megabyte (MB) und für modernen System wahrscheinlich zu gering (Empfehlung: 300 MB). Der Wert kann auch in Prozent angegeben werden (Beispiel: **5%**)

```
admin_space_left = 50
admin_space_left_action = SUSPEND
```

Speicher-Medium Fehler Event

- Diese Werte geben an wie der Audit-Daemon auf Fehler beim Schreiben der Log-Datei reagieren soll
    - In der Standard-Einstellung stellt der Audit-Daemon seine Arbeit ein

```
disk_full_action = SUSPEND
disk_error_action = SUSPEND
```

Event-Aktionen des Audit-Daemon

SCHLÜSSELWORT	BESCHREIBUNG
ignore	Zustand ignorieren, keine Aktion
syslog	Zustand via `syslog` melden
rotate	Log-Dateien rotieren, überflüssige Log-Dateien entfernen
exec	Ein Skript ausführen
suspend	Keine Log-Daten mehr schreiben
single	System in den Single-User-Mode versetzen, Netzwerk deaktivieren
halt	System herunterfahren (shutdown)

Weiterleiten von Event-Informationen

Audit-Daemon Plugins

- Der Audit-Daemon kann Audit-Events an andere Log-Systeme (z.B. SYSLOG oder SIEM Systeme) weiterleiten
- Hierzu werden Plugins geladen welche das jeweilige Logging-Protokoll implementieren
    - Die Plugins befinden sich unterhalb von `/etc/audit/plugins.d`
    - `max_restarts` gibt an wie oft ein Plugin nach einen Crash von Audit-Daemon neu gestartet wird

```
max_restarts = 10
plugin_dir = /etc/audit/plugins.d
```

Beispiel der Syslog-Plugin Konfiguration

- Um ein Plugin zu aktivieren wird der Wert `active` auf `yes` gesetzt und danach der Audit-Daemon neu gestartet

```
active = no
direction = out
path = /sbin/audisp-syslog
type = always
args = LOG_INFO
format = string
```

Weitere Konfigurations-Optionen

- Die weiteren Konfigurations-Optionen in der Datei `auditd.conf` werden für den Empfang von Audit-Log-Daten über das Netzwerk benötigt
- Wir werden uns diese Konfiguration später anschauen

Audit-Subsystem Status abfragen

Dienste Status

- Auf Red Hat basierten Linux-Systemen ist der Audit-Daemon in der Regel vorinstalliert und gestartet (speziell wenn SELinux installiert und aktiviert ist)

```
# systemctl status auditd
● auditd.service - Security Auditing Service
     Loaded: loaded (/usr/lib/systemd/system/auditd.service; enabled; vendor preset: enabled)
     Active: active (running) since Sun 2022-10-16 08:25:40 UTC; 12h ago
       Docs: man:auditd(8)
             https://github.com/linux-audit/audit-documentation
   Main PID: 645 (auditd)
      Tasks: 4 (limit: 2506)
     Memory: 8.1M
        CPU: 960ms
     CGroup: /system.slice/auditd.service
             ├─645 /sbin/auditd
             └─647 /usr/sbin/sedispatch

Oct 16 08:25:40 localhost augenrules[660]: enabled 1
Oct 16 08:25:40 localhost augenrules[660]: failure 1
Oct 16 08:25:40 localhost augenrules[660]: pid 645
Oct 16 08:25:40 localhost augenrules[660]: rate_limit 0
Oct 16 08:25:40 localhost augenrules[660]: backlog_limit 8192
Oct 16 08:25:40 localhost augenrules[660]: lost 0
Oct 16 08:25:40 localhost augenrules[660]: backlog 4
Oct 16 08:25:40 localhost augenrules[660]: backlog_wait_time 60000
Oct 16 08:25:40 localhost augenrules[660]: backlog_wait_time_actual 0
Oct 16 08:25:40 localhost systemd[1]: Started Security Auditing Service.
```

Ad-Hoc Auditing

Befehl autrace

- Mit dem Audit-Subsystem Trace Befehl **`autrace`** kann ein Audit-Trace eines Prozesses (hier 4 x ICMP Echo per **`ping`**) erzeugt werden
    - Der Trace ist unabhängig vom geladenen Audit-Regelwerk

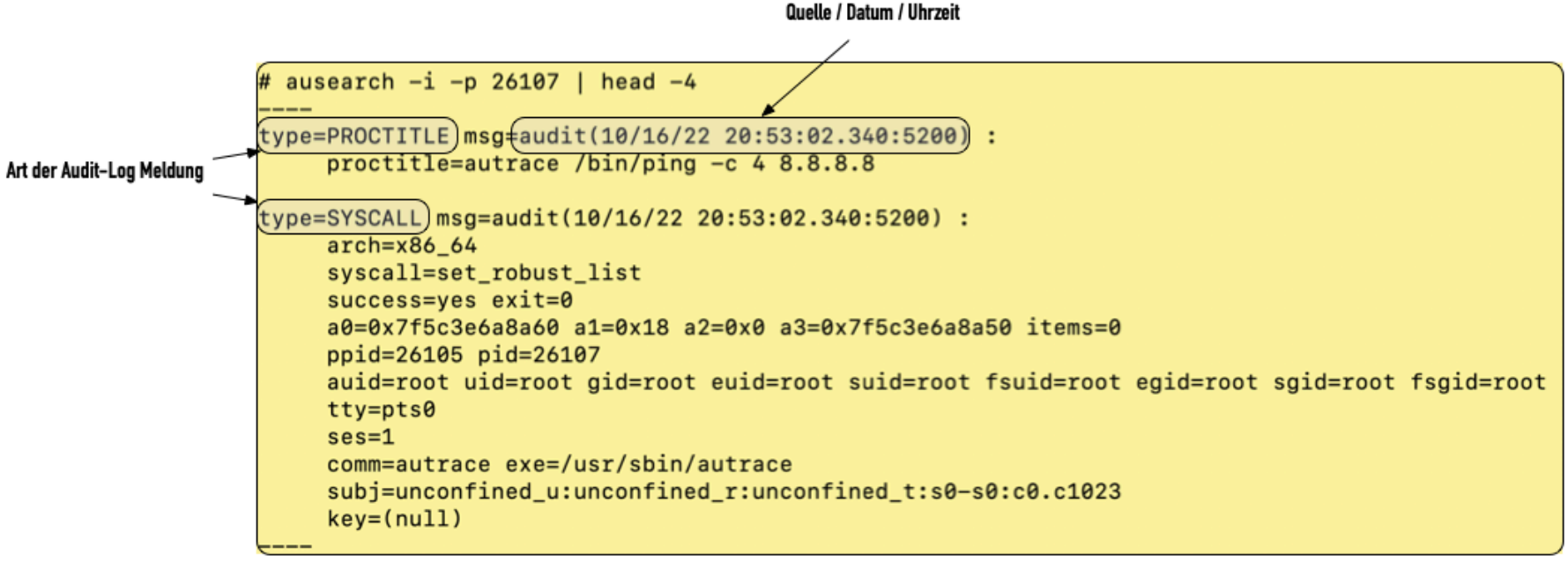
```
# autrace /bin/ping -c 4 8.8.8.8
Waiting to execute: /bin/ping
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=60 time=1.11 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=60 time=0.807 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=60 time=0.666 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=60 time=0.726 ms

--- 8.8.8.8 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3045ms
rtt min/avg/max/mdev = 0.666/0.827/1.110/0.170 ms
Cleaning up...
Trace complete. You can locate the records with 'ausearch -i -p 26107'
```

Auditlog des Trace anzeigen

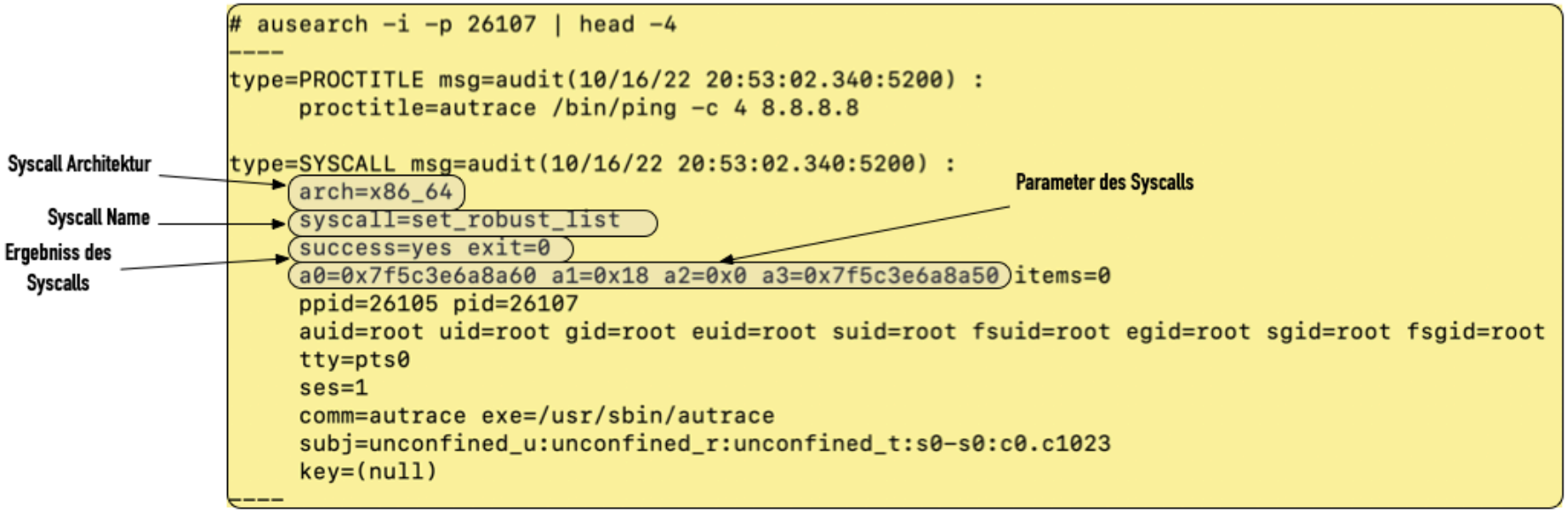
- Audit Log für einen speziellen `autrace` Aufruf anschauen

```
ausearch -i -p <audit-id>
```

Anatomie eines Audit-Log Eintrags

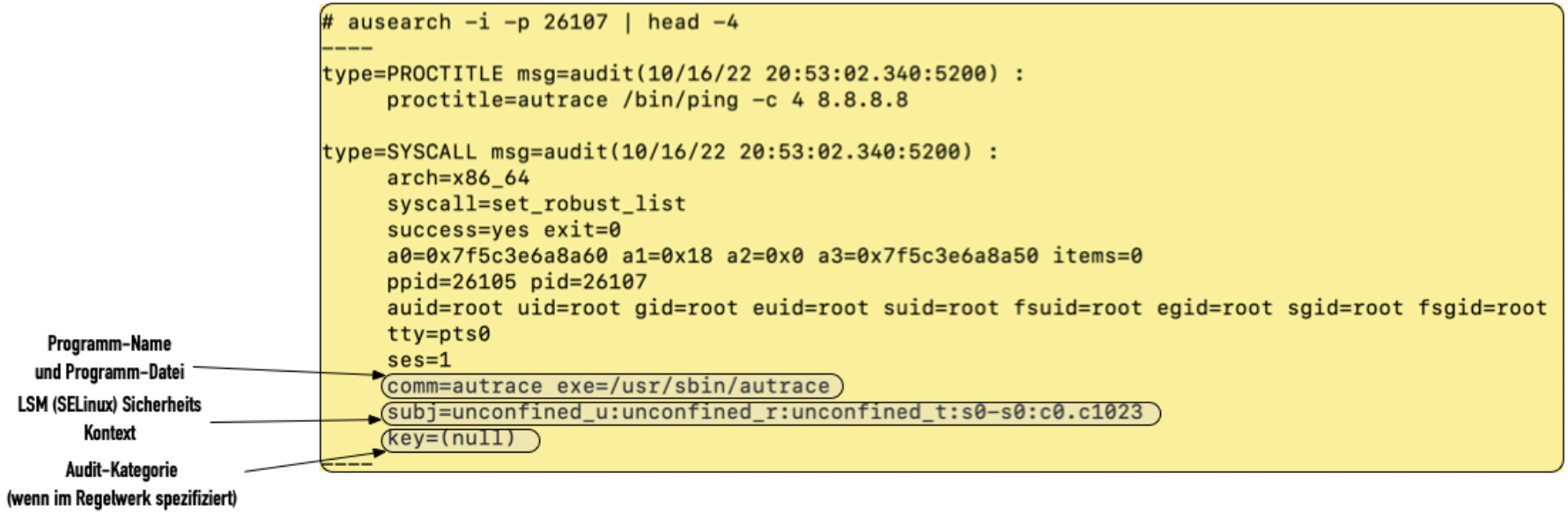


Anatomie eines Audit-Log Eintrags



Anatomie eines Audit-Log Eintrags

Anatomie eines Audit-Log Eintrags



Audit-Regelwerke

Audit Richtlinien Regelwerke

- Die Richtlinien-Regelwerke des Audit-Subsystems sind für den Einsatz von SELinux nicht notwendig, aber bieten oft eine sinnvolle Ergänzung zu den SELinux Funktionen
    - SELinux: Durchsetzen einer Richtlinie
    - Audit-Subsystem: Überwachung einer Richtlinie

Audit Richtlinien Regelwerke

- Die Richtlinien Regelwerke liegen im Verzeichnis `/etc/audit/rules.d/`
- Beim aktivieren der Regeln werden die Dateien in diesem Verzeichnis in der alphanumerischen Reihenfolge der Dateinamen zu einer Datei unter `/etc/audit/audit.rules` zusammengefasst und in den Linux-Kernel geladen
- Die Datei `/etc/audit/audit.rules` sollte nicht manuell editiert werden, Änderungen werden überschrieben

Beispiel Richtlinien Regelwerke

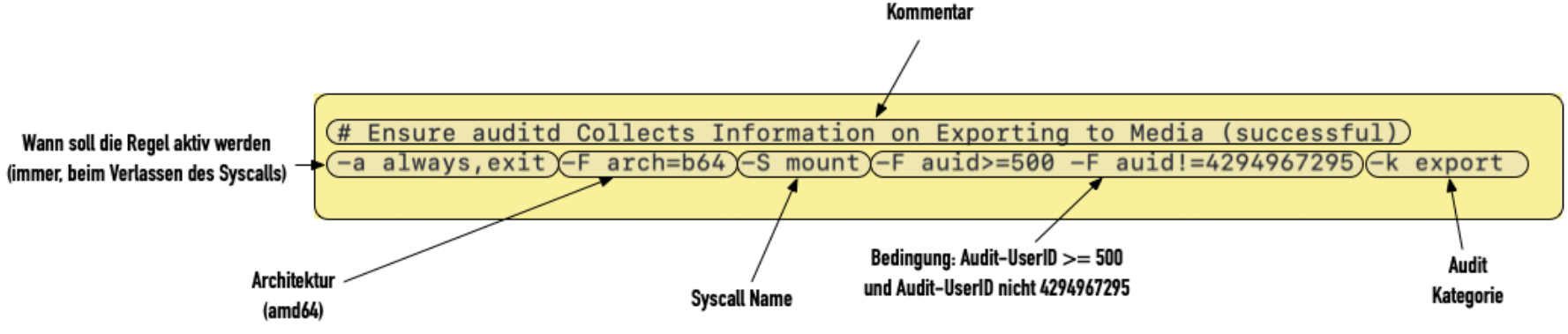
- Beispiel-Richtlinien werden unter `/usr/share/audit/sample-rules` mitgeliefert und können in `/etc/audit/rules.d/` kopiert werden.
  - Die Regelwerke sollten jeweils manuell geprüft und ggf. angepasst werden (System-Architektur, Syscall-Namen)
  - Einige Regelwerke werden für das System individuell via Shell-Skripte erzeugt und müssen ggf. nach Software-Installationen oder -Updates neu generiert werden

Inhalt der Audit-Regelwerke

- Jede Zeile der Regelwerke ist eine Regel
- Die Regeln sind die Kommandozeilen-Parameter des Programms `auditctl`
- Die Dokumentation befindet sich in der `man` Page zu `auditctl`

```
# man auditctl
```

Beispiel einer Audit-Regel



Immutable vs. Mutable

- Audit-Regelwerke können nach dem Laden in den Kernel als *unveränderbar* (immutable) markiert werden

```
# Make the auditd Configuration Immutable
-e 2
```

- Unveränderbare Regeln können nicht mehr im laufenden Betrieb des Linux-Systems geändert werden, es ist immer ein Neustart (Reboot) notwendig

Audit-Regelwerke aktivieren

augenrules

- Das Programm `augenrules` prüft die Regelwerke, fasst diese zusammen und lädt die Regeln in den Linux-Kernel

```
# augenrules --check
# augenrules --load
```

Status des Audit Subsystems

- Status des Audit-Subsystems anzeigen

```
# auditctl -s
enabled 2
failure 1
pid 12901
rate_limit 0
backlog_limit 8192
lost 0
backlog 0
backlog_wait_time 0
loginuid_immutable 0 unlocked
```

Aktuelles Regelwerk auflisten

- Aktive Regeln auflisten

```
# auditctl -l
-a always,exit -F arch=b32 -S stime,settimeofday,adjtimex -F key=time-change
-a always,exit -F arch=b64 -S adjtimex,settimeofday -F key=time-change
-a always,exit -F arch=b32 -S clock_settime -F a0=0x0 -F key=time-change
-a always,exit -F arch=b64 -S clock_settime -F a0=0x0 -F key=time-change
[...]
```

Audit-Logs auswerten

Abfragen mit "ausearch"

- Mittels des Programms `ausearch` lässt sich das lokale Audit-Log abfragen

Beispiel-Abfrage mit "ausearch"

- Alle Audit-Einträge zum Thema "sudo" zeigen

```
ausearch -i -x sudo
```

Beispiel-Abfrage mit "ausearch"

- Report über fehlgeschlagene Anmeldeversuche

```
ausearch -m USER_AUTH,USER_ACCT --success no
```

Beispiel-Abfrage mit "ausearch"

- Alle Audit-Meldungen für Benutzer UID 1000

```
ausearch -ua 1000 -i
```

Beispiel-Abfrage mit "ausearch"

- Fehlgeschlagene Syscalls seit gestern

```
ausearch --start yesterday --end now -m SYSCALL -sv no -i
```

CSV Ausgabe

- `ausearch` erlaubt die Ausgabe der Abfrage-Ergebnisse als CSV-Datei
- Diese Dateien können in Office-Programme oder Datenbanken importiert werden

```
# ausearch --start today --format csv 2>/dev/null > audit-today.csv
```

Reports

- Der Befehl `aureport` bietet viele zusammengefasste Reports auf den Audit-Log-Dateien. Hier zum Beispiel ein Report über Verstöße gegen LSM (AVC) Richtlinien:

```
# aureport -a

AVC Report
============================================================
# date time comm subj syscall class permission obj result event
============================================================
1. 10/16/22 08:25:41 chronyd system_u:system_r:chronyd_t:s0 262 lnk_file read system_u:object_r:unlabeled_t:s0 denied 22
2. 10/17/22 00:01:01 logrotate system_u:system_r:logrotate_t:s0 262 file getattr system_u:object_r:unlabeled_t:s0 denied 6127
3. 10/17/22 00:01:01 logrotate system_u:system_r:logrotate_t:s0 262 file getattr system_u:object_r:unlabeled_t:s0 denied 6128
```

Reports

- Weitere Reports

BEFEHL	REPORT
aureport -s	Syscall Report
aureport -p	Prozess Report
aureport -x	Report nach ausführbaren Dateien
aureport -f	Report nach Dateizugriffen
aureport -u	Report über Benutzeraktivitäten
aureport -l -i	Report über Logins

Audit-Hilfsprogramme

Spezielle Abfrageprogramme

- Die letzten Logins auf dem System anzeigen

```
# aulast
```

Letzte Anmeldezeiten von Benutzern

- Wann haben sich Benutzer zum letzten Mal am System angemeldet?

```
# aulastlog
```

Statistiken über das Audit-Log

- Zusammenfassung des Audit-Log

```
# aureport
```

Grafische Auswertungen

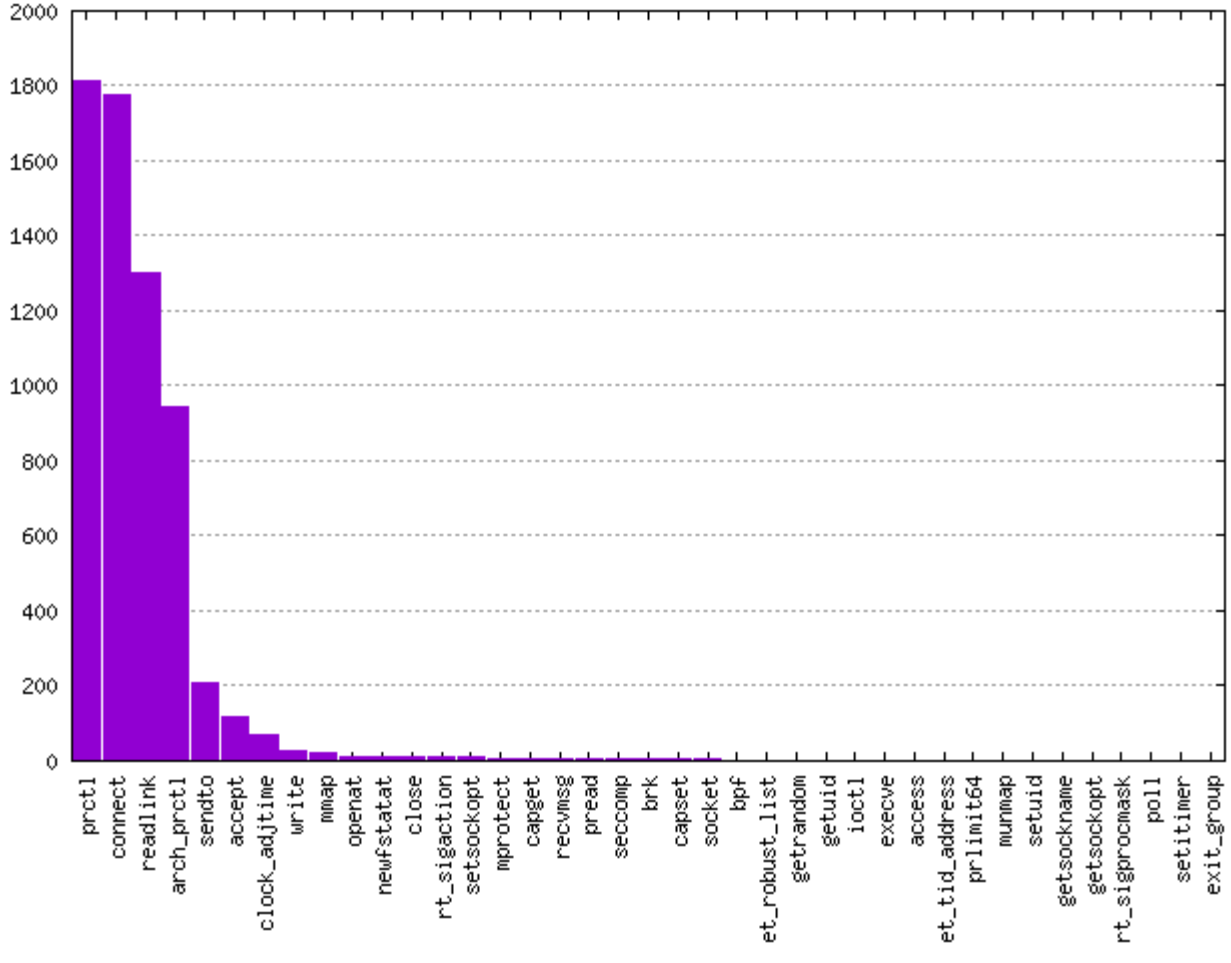
- Eine grafische Auswertung von Audit-Logs ist mit Skriptes des Linux-Audit Entwicklers Steve Grubb möglich
    - Diese Skripte wurden angepasst um mit modernen Versionen des Programms `gnuplot` zu funktionieren, und das Ausgabeformat wurde von Postscript auf PNG-Dateien geändert

```
$ sudo dnf install epel-release
$ sudo dnf install graphviz gnuplot git
$ git clone https://github.com/cstrotm/audit-visualize
$ cd audit-visualize
$ chmod +x ./mkgraph
$ chmod +x ./mkbar
```

Grafische Reports erstellen

```
$ sudo aureport -s -i --summary  | bash ./mkbar syscall
$ sudo aureport -f -i --summary --failed | bash ./mkbar failed-access
$ sudo aureport -e -i --summary | egrep -vi '(syscall|change)'
$ sudo aureport -e -i --summary | egrep -vi '(syscall|change)' | bash ./mkbar events2
```

Beispiel einer Syscall Grafik



Syscall Benutzung von Programmen

```
$ sudo aureport -s -i | awk '/^[0-9]/ { printf "%s %s\n", $6, $4 }' | sort | uniq | bash ./mkgraph
Gzipping graph...
Graph was written to gr.png
```

Welcher Benutzer führt welche Programme aus?

```
sudo aureport -u -i | awk '/^[0-9]/ { printf "%s %s\n", $4, $7 }' | sort | uniq | bash ./mkgraph
```

Wer greift auf Dateien zu?

```
sudo aureport -f -i | awk '/^[0-9]/ { printf "%s %s\n", $8, $4 }' | sort | uniq | bash ./mkgraph
```

Weitere Audit-Logauswertungs-Programme

- AuditExplorer - An R shiny app that visualizes audit data using many tools all in one app: https://github.com/stevegrubb/audit-explorer/

Vorlagen für Audit-Richtlinien

Vorlagen für Audit-Regeln

- CIS Benchmark Webseite: https://downloads.cisecurity.org/
- A Linux Auditd rule set mapped to MITRE's Attack Framework https://github.com/bfuzzy1/auditd-attack
- Enhanced AuditD (Audit Daemon) rules for Linux systems, HIPAA & HITRUST compliance https://github.com/gnxsecurity/enhanced-auditd-rules
- An auditd ruleset for monitoring linux servers https://github.com/benjaminkoffel/auditd-rules
- UnderstandingAuditdRules - This is an overview of writing auditd rules for linux https://github.com/clevelandjosh/UnderstandingAuditdRules

Ende des Kapitels "Audit-Subsystem"