



OTUS

ОНЛАЙН-ОБРАЗОВАНИЕ

Онлайн-образование



Не забыл включить запись

Меня хорошо видно && слышно?

Ставьте плюсы, если все хорошо
Напишите в чат, если есть проблемы

Правила вебинара

- Активно участвуем
- Задаем вопросы в чат или голосом
- Off-topic обсуждаем в Slack #канал группы или #general
- Вопросы вижу в чате, могу ответить не сразу

Сбор и анализ логов

Маршрут вебинара

- Логи
- rsyslog + logrotate
- journald
- abrt
- auditd
- kdump

Цели занятия

После занятия вы сможете:

1. Понять каким образом осуществляются сбор и анализ логов
2. Освоить принципы логирования с помощью rsyslog
3. Понять возможности logrotate для задач ротации логов
4. Научиться работать с journald и auditd
5. Узнать про adrttd и kdump

Зачем вам это уметь:

1. Чтобы понимать основные аспекты журналирования событий
2. Чтобы использовать более полный спектр имеющихся инструментов для анализа событий в инфраструктуре
3. Чтобы иметь выбор инструментов для решения задач сбора и анализа логов

Логи

Основные лог файлы

- Приложения
- События
- Службы
- Системные

/var/log

- /var/log/syslog или /var/log/messages - глобальный системный журнал
- /var/log/auth.log или /var/log/secure - информация об авторизации пользователей
- /var/log/dmesg - оборудование и драйверы устройств

```
root@logs ~]# dmesg -l err
[1131424.604352] logs kernel: end_request: I/O error, dev sdc, se
[1131424.604352] logs kernel: Buffer I/O error on device sdc, log
[1131424.604352] logs kernel: Buffer I/O error on device sdc, log
```

/var/log

- /var/log/anaconda.log - лог установки системы
- /var/log/audit - лог демона auditd
- /var/log/boot.log - лог загрузки системы
- /var/log/cron - лог демона crond

/var/log

Для каждого дистрибутива создается отдельный журнал менеджера пакетов:

- /var/log/yum.log – для программ установленных с помощью yum в RedHat Linux
- /var/log/emerge.log – для ebuild-ов установленных из Portage с помощью emerge в Gentoo Linux

/var/log

- /var/log/dpkg.log – для программ установленных с помощью dpkg в Debian Linux и всем семействе родственных дистрибутивов
- /var/log/mysql - логи базы данных MySQL
- /var/log/apache2 - логи веб-сервера Apache
- /var/log/nginx - логи веб-сервера NGINX

Полезные утилиты для просмотра ЛОГОВ

- tail
- tail -f
- cat
- less
- zcat
- zgrep
- zmore
- lnav

Ваши вопросы?

Маршрут вебинара

- Логи
- `rsyslog + logrotate`
- `journald`
- `abrt`
- `auditd`
- `kdump`

Локальное хранение логов

Вопрос к аудитории: "Как вы считаете, каковы минусы локального хранения логов?"

Минусы локального хранения ЛОГОВ

- Выделение дискового пространства под хранение логов локально
- Неудобство анализа логов в любых инфраструктурах
- Локальные логи проще удалить или подделать
- Нагрузка на дисковую подсистему при большом потоке логов

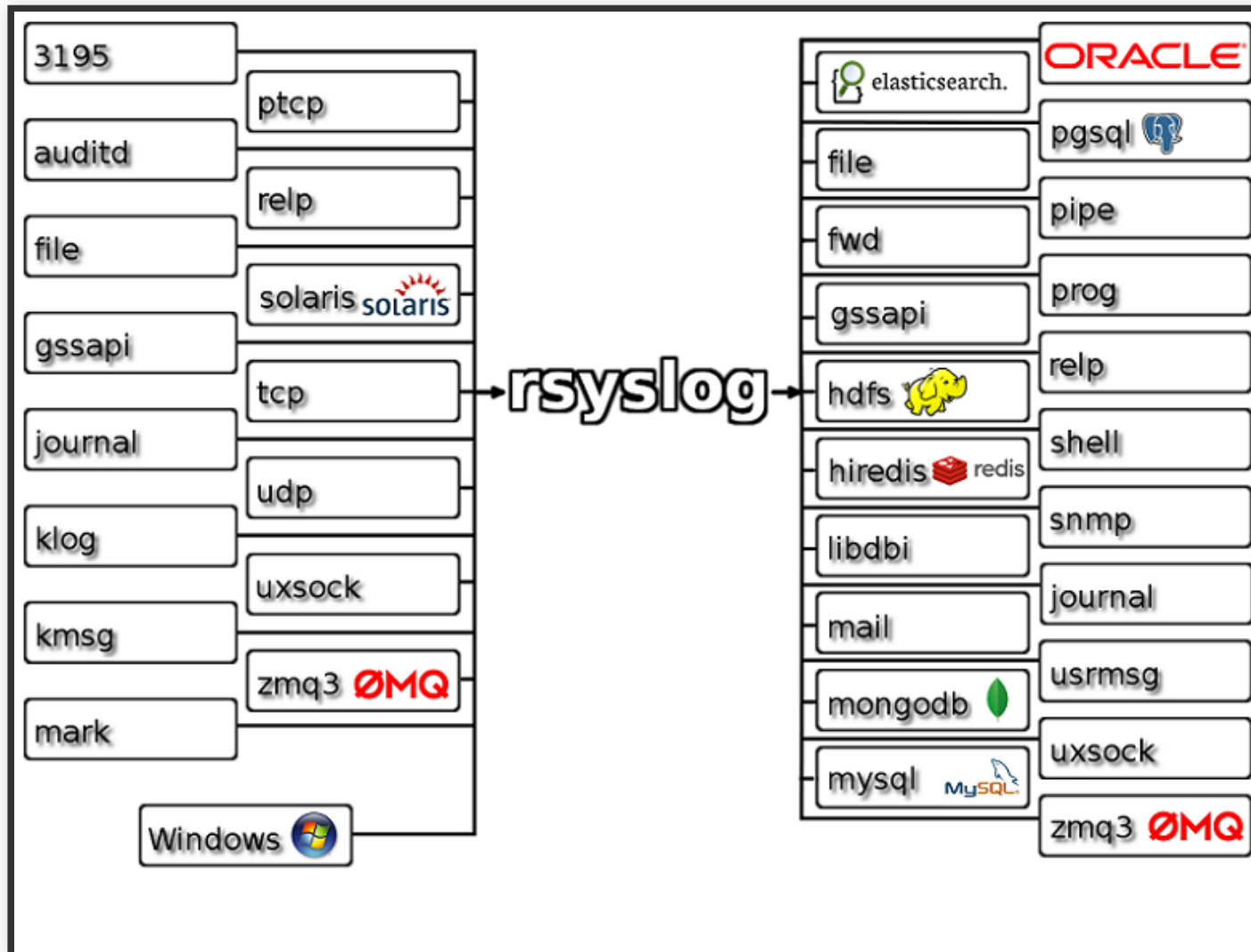
rsyslog

Rsyslog - Rocket-fast System for log processing

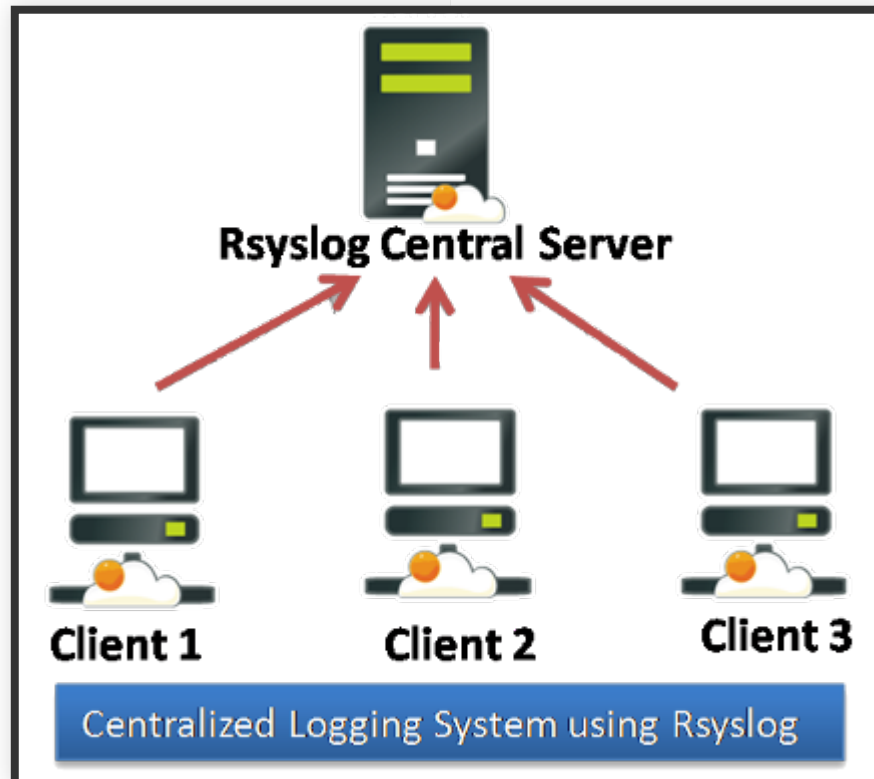
Особенности:

- многопоточный
- TCP, SSL, TLS, RELP
- сохранение логов в базы данных (MySQL, PostgreSQL, Oracle)
- фильтрация по любой части лога
- полностью настраиваемый формат вывода

rsyslog



rsyslog



rsyslog: конфигурация

Фрагмент конфига /etc/rsyslog.conf

```
# The authpriv file has restricted access.
authpriv.*                                /var/log/secure

# Log all the mail messages in one place.
mail.*                                    -/var/log/maillog

# Log cron stuff
cron.*                                    /var/log/cron

# Everybody gets emergency messages
*.emerg                                    :omusrmsg:*

# Save news errors of level crit and higher in a special file.
uucp,news.crit                            /var/log/spooler
```

rsyslog: конфигурация

Пример конфига `/etc/rsyslog.d/sftp.conf`

```
module(load="imuxsock")

# log internal-sftp activity to sftp.log
if $programname == 'internal-sftp' then /var/log/sftp.log
& stop
```

rsyslog: модули

- Модули ввода - начинаются с `im`, собирают информацию из различных источников
- Модули вывода - начинаются на `om`. Отправляют сообщения. Могут отправлять сообщения как в файл так и по сети или складывать в базу
- Модули фильтрации - начинаются с `fm`. Фильтруют сообщения по разным параметрам

rsyslog: модули

- Модули парсинга - начинаются с `pt`. Позволяют проводить синтаксический анализ
- Модули модификации сообщений - начинаются с `mm`. Меняют содержимое обрабатываемых сообщений
- Модули генерации строк - начинаются с `sm`. Позволяют генерировать строки на основе обрабатываемых сообщений

rsyslog: примеры модулей

- Input Modules (imtcp, imjournal, imudp, imrelp, ...)
- Output Modules (omelasticsearch, omfile, ommysql,)
- Parser Modules (pmciscoios, pmlastmsg ...)
- Message Modification Modules (mmcount, mmfields ...)
- String Generator Modules (smfile, smfwd, smtradfile, smfwd)

rsyslog: facility

- Facility (категория) - принимает значения от 0 до 23, им соответствуют различные категории системных служб: 0 – kernel, 2 – mail, 7 – news
- Последние 8 категорий – от local0 до local7 – определены для служб, не попадающих в predetermined categories
- Полный список категорий <https://en.wikipedia.org/wiki/Syslog#Facility>

rsyslog: severity

- Severity (важность) принимает значения от 0 (emergency, самая высокая) до 7 (debug, самая низкая)
- Полный список https://en.wikipedia.org/wiki/Syslog#Severity_level

rsyslog: наборы правил

- Набор правил (ruleset) - содержит список правил, которые состоят из фильтра и привязанных к фильтру действий (Actions)
- Можно задавать несколько наборов правил, чтобы разделить обработку различных сообщений

rsyslog: фильтры

- rsyslog позволяет фильтровать логи.
- Примеры фильтров:

```
объект.[!]операция_сравнения/уровень, действие  
:переменная, [!]операция_сравнения, "искомое_значение" действие
```

- В качестве объектов выступают категории
- В качестве уровней - уровни важности сообщений

rsyslog: фильтры

Примеры фильтров:

```
kern.* # все логи ядра
mail.crit # все критические события от объекта mail
cron.!info,!debug # все от демона crond кроме уровней info и debu
```

```
*.=crit /var/log/somefile
& root
& /var/log/criticalmessages
# фильтруем сообщения
*.* /var/log/allmsgs-incl-informational.log
:msg, contains, "informational" ~
*.* /var/log/allmsgs-no-informational.log
```

rsyslog: очереди

rsyslog: очереди

- Direct queues
- Disk queues
- In-memory queues
- Disk-Assisted Memory queues

rsyslog: шаблоны

- Templates - позволяют шаблонизировать динамические имена файлов, содержимого на входе и выходе, SQL для баз данных и т.д.
- Типы шаблонов: list, subtree, string, plugin

Logrotate

Ротация логов

Вопрос к аудитории: "Зачем по-вашему нужна ротация логов?"

Ротация логов

Logrotate - подсистема для автоматической ротации логов, с возможностью сжатия, удаления, перемещения и гибкой настройкой под каждый вид лог файлов

Ротация логов

Основные параметры конфигов logrotate. Частота проверки по условиям:

- hourly - каждый час
- daily - каждый день
- weekly - каждую неделю
- monthly - каждый месяц
- yearly - каждый год

logrotate: условия

Основные параметры конфигов logrotate:

- rotate - указывает количество старых логов, которые необходимо хранить
- dateext - добавляет дату ротации перед именем лога
- compress/delaycompress - сжатие лога и отсрочка сжатия
- copytruncate - после создания копии, обрезать исходный файл журнала вместо перемещения старого файла журнала и создания нового
- mail - отправка уведомления на почту
- missingok - не выдавать ошибки, если лог файла не существует

logrotate: условия

Основные параметры конфигов logrotate.

- `sharedscripts` - запускает сторонний скрипт только один раз
- `postrotate/endscript` - запуск скрипта или произвольной команды
- `size` - ротация лог файла произойдет при достижении файлом заданного в этом параметре размера. Это взаимоисключающий параметр с параметром интервала (`daily`, `weekly` и т.д.), то есть logrotate будет смотреть только на размер файла лога в данном конфиге

logrotate: условия

- `maxsize` - задается совместно с параметром интервала. Ротация произойдет в зависимости от того, что наступит раньше: период ротации или увеличение размера файла до указанного
- `minsize` - также задается совместно с параметром интервала, ротация происходит при достижении файлом заданного размера, но НЕ раньше, чем наступит период ротации

logrotate: пример конфигурации

```
[root@logs ~]# cat /etc/logrotate.d/nginx
/var/log/nginx/*.log {
    daily
    missingok
    rotate 52
    compress
    delaycompress
    notifempty
    create 0640 www-data adm
    sharedscripts
    prerotate
        if [ -d /etc/logrotate.d/httpd-prerotate ]; then \
            run-parts /etc/logrotate.d/httpd-prerotate; \
        fi \
    endscript
}
```

logrotate: пример конфигурации

Проверка конкретного конфига logrotate без применения:

```
logrotate -d /etc/logrotate.d/chrony
```

Применение конкретного конфига logrotate с выводом сообщений в консоль:

```
logrotate -v /etc/logrotate.d/auth
```

Запуск всех конфигов logrotate:

```
logrotate -v /etc/logrotate.conf
```

Ваши вопросы?

Маршрут вебинара

- Логи
- rsyslog + logrotate
- [journald](#)
- abrt
- auditd
- kdump

journald

journald

Вопрос к аудитории: "А вы уже используете journald?"

Journald: особенности

- бинарный формат логов (защита от подделки, возможность конвертации в другие форматы)
- не требует специальной настройки
- структурированные данные (multi-field, multi-line)
- индексированные данные
- центральное хранилище логов

journald: прием логов

Какие логи принимает journald?

- простые syslog логи
- логи ядра (kmsg)
- структурированные данные через Journal API
- логи и статусы systemd юнитов
- записи системы аудита (auditd)

journald: условия работы

- сервис - systemd-journald
- основной конфиг - /etc/systemd/journald.conf
- временное хранилище логов - /run/log/journal/*
- постоянное хранилище логов - /var/log/journal/*
- чтение из хранилища доступно только для группы systemd-journald

journald: параметры конфигурации

- Storage (volatile, persistent, auto, none) - по-умолчанию auto (пишет логи в tmpfs), чтобы писать логи на диск - нужно поставить persistent
- Compress (yes, no) - сжимает данные перед записью
- Seal (yes, no) - накладывает криптографическую печать

journald: параметры конфигурации

- SplitMode (uid, none) - позволяет разделять логи по UID пользователей
- RateLimitInterval, RateLimitBurst - регулировка скорость обработки
- MaxFileSec - период ротации
- MaxRetentionSec - как долго хранить логи
- SyncIntervalSec - интервал синхронизации журнала с файлами логов

journald: параметры конфигурации

- ForwardToSyslog, ForwardToKMsg, ForwardToConsole, ForwardToWall - опции перенаправления сообщений
- MaxLevelStore, MaxLevelSyslog, MaxLevelKMsg, MaxLevelConsole, MaxLevelWall - задаем уровни важности сообщений для разных логов
- SystemMaxUse - максимальный объем, который логи могут занимать на диске

journald: параметры конфигурации

- SystemKeepFree - объем свободного места на диске, после сохранения логов
- SystemMaxFileSize - объем файла лога, по достижении которого он должен быть удален с диска
- RuntimeMaxUse - максимальный объем, который логи могут занимать в файловой системе /run

journald: параметры конфигурации

- RuntimeKeepFree - объем свободного места на /run, после сохранения логов
- RuntimeMaxFileSize - объем файла лога, по достижении которого он должен быть удален из файловой системы /run

journalctl: варианты просмотра ЛОГОВ

Вывод сообщений в структурированном json:

```
journalctl -xe -o json-pretty
```

Вывод списка ID загрузок:

```
journalctl --list-boots
```

Вывод сообщений позапрошлой загрузки системы:

```
journalctl -b -2
```

journalctl: варианты просмотра ЛОГОВ

Просмотр сообщений за заданный период времени:

```
journalctl --since "2020-01-13 00:01" --until "2020-01-14 23:59"
```

Вывод сообщений за прошедшие 10 часов:

```
journalctl --since "10 hours ago"
```

Вывод сообщений по заданному systemd юниту в формате чтения из файла:

```
journalctl -u mysqld.service -f
```

journalctl: варианты просмотра ЛОГОВ

Вывод сообщений процессов, запущенных от имени пользователя с заданным UID:

```
journalctl _UID=1001
```

Вывод последних трех сообщений с уровнем важности crit:

```
journalctl -n 3 -p crit
```

journalctl: управление

Перенести все логи из /run в /var:

```
journalctl --flush
```

Задать максимальный размер хранящихся логов:

```
journalctl --vacuum-size=1G
```

Задать максимальное время хранения логов:

```
journalctl --vacuum-time=1years
```

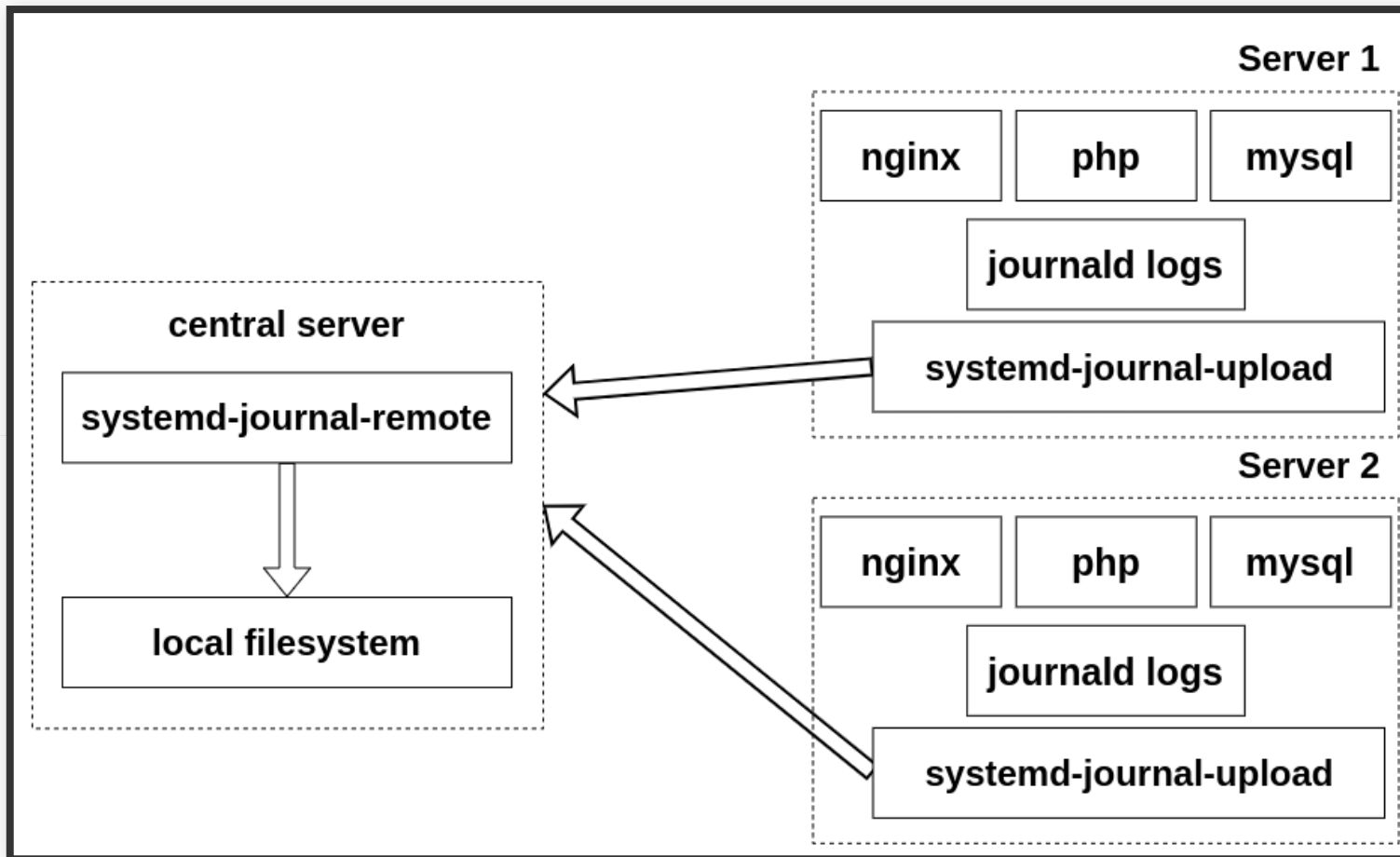
Показать занимаемый логами объем диска:

```
journalctl --disk-usage
```

journald: централизованное хранение

- `systemd-journal-remote` - демон для приема и сбора логов с удаленных хостов (работает в активном или пассивном режиме)
- `systemd-journal-gatewayd` - http-сервер для приема логов на центральный хост по протоколу http
- `systemd-journal-upload` - демон для загрузки логов с локальной машины в удаленное хранилище

journald: централизованное хранение



journald: централизованное хранение

Установка сервисов journald:

```
yum -y install systemd-journal-gateway
```

Создаем каталог для хранения логов на центральном сервере:

```
mkdir -p /var/log/journal/remote  
chown systemd-journal-remote:systemd-journal-remote /var/log/jour
```

journald: централизованное хранение

Редактируем конфиг `/lib/systemd/system/systemd-journal-remote.service`:

```
[Service]
ExecStart=/usr/lib/systemd/systemd-journal-remote \
    --listen-http=-3 \
    --output=/var/log/journal/remote
User=systemd-journal-remote
Group=systemd-journal-remote
PrivateTmp=yes
PrivateDevices=yes
PrivateNetwork=yes
WatchdogSec=10min

[Install]
Also=systemd-journal-remote.socket
```

journald: централизованное хранение

Смотрим конфиг `/lib/systemd/system/systemd-journal-remote.socket`:

```
[Unit]
Description=Journal Remote Sink Socket

[Socket]
ListenStream=19532

[Install]
WantedBy=sockets.target
```

journald: централизованное хранение

Запускаем демон systemd-journal-remote:

```
systemctl daemon-reload  
systemctl restart systemd-journal-remote
```

На удаленном сервере редактируем конфиг
`/etc/systemd/journal-upload.conf`:

```
[Upload]  
URL=http://172.16.10.110:19532  
# ServerKeyFile=/etc/ssl/private/journal-upload.pem  
# ServerCertificateFile=/etc/ssl/certs/journal-upload.pem  
# TrustedCertificateFile=/etc/ssl/ca/trusted.pem
```

journald: централизованное хранение

На удаленном сервере запускаем демон systemd-journal-upload:

```
systemctl start systemd-journal-upload
```

На центральном сервере смотрим логи с удаленного сервера:

```
journalctl -D /var/log/journal/remote --follow
```

journald: централизованное хранение

Пример запуска `systemd-journal-remote` для отправки логов на удаленный сервер:

```
[root@logs ~]# systemd-journal-remote --url https://some.host:195
```

Пример запуска `systemd-journal-remote` для экспорта локальных логов в другой каталог:

```
[root@logs ~]# journalctl -o export | systemd-journal-remote -o /
```

Ваши вопросы?

Маршрут вебинара

- Логи
- rsyslog + logrotate
- journald
- **abrt**
- auditd
- kdump

abrttd

abrttd

abrttd - инструмент для создания отчетов об ошибках (bug reports)

- Активизируется при сбоях приложений и собирает дополнительные данные для последующего анализа
- Формирует отчеты для отправки (в том числе автоматической)

В состав входят:

- abrttd - сам демон
- abrttd-applet - апплет, работающий в области уведомлений пользователей в оболочке

abrt

- abrt-gui - графический интерфейс, которые показывает собранные проблемы и позволяет оповещать о них
- abrt-cli - интерфейс командной строки

abrt

Установка:

```
yum install abrt-addon-ccpp abrt-addon-python abrt-cli abrt-tui a
```

Запуск сервисов:

```
systemctl start abrt abrt-ccpp abrt-oops  
systemctl status abrt abrt-ccpp abrt-oops
```

Проверка:

```
cat /proc/sys/kernel/core_pattern  
  
/usr/libexec/abrt-hook-ccpp %s %c %p %u %g %t e
```

abrt

Просмотр списка отчетов:

```
abrt-cli list
id d6364f5cb49aef334e957c0bc6b4331ffc34e968
reason:      bash killed by SIGSEGV
time:        Thu 06 Aug 2020 08:48:13 PM UTC
cmdline:     /bin/bash ./loop
package:     bash-4.2.46-31.el7
uid:         0 (root)
count:       1
Directory:   /var/spool/abrt/ccpp-2020-08-06-20:48:13-6128
```

abrt-d

Детальный просмотр отчета:

```
abrt-cli list -d
```

Ваши вопросы?

Маршрут вебинара

- Логи
- rsyslog + logrotate
- journald
- abrt
- auditd
- kdump

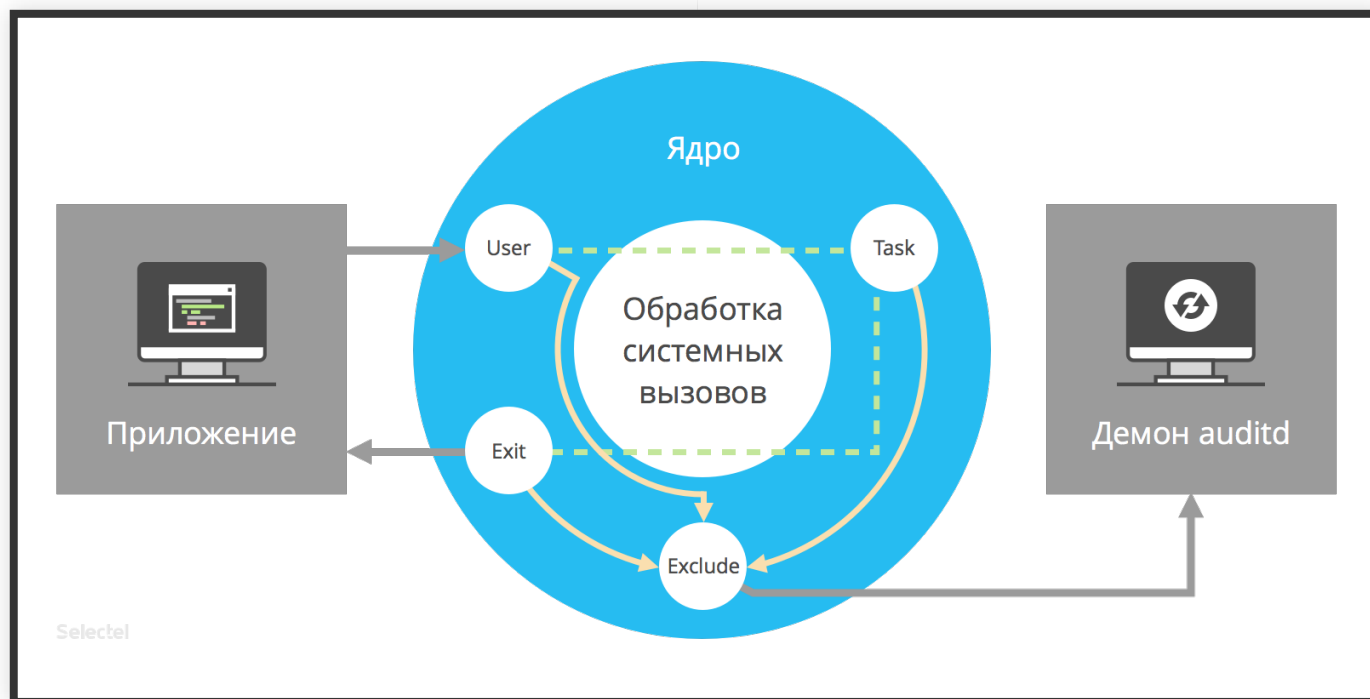
auditd

auditd

Вопрос к аудитории: "Еще помните, что такое audit.log?"

auditd

auditd - подсистема аудита в ядре Linux, отслеживает критичные с точки зрения безопасности системные события



auditd

Особенности:

- Пишет логи аудита
- Для удобства просмотра логов можно использовать `ausearch` и `aureport`
- Команда `auditctl` позволяет настраивать правила аудита вручную
- После загрузки системы правила читаются из `/etc/audit/audit.rules`
- Некоторые параметры `auditd` можно изменить в `/etc/audit/auditd.conf`

auditd: примеры добавления правил

Все системные вызовы с конкретного PID

```
auditctl -a entry,always -S all -F pid=1005
```

Все открытые конкретным пользователем файлы

```
auditctl -a exit,always -S open -F auid=510
```

Все неудачные попытки открытия файлов

```
auditctl -a exit,always -S open -F success!=0
```

auditd: примеры добавления правил

Наблюдение за каталогом

```
auditctl -w /home/user/test_dir/ -k test_watch
```

Наблюдение за вызовов insmod

```
auditctl -w /sbin/insmod -p x -k module_insertion
```

auditd: примеры просмотра лога

Выбор событий по аккаунту пользователя:

```
ausearch -u1 root
```

Выбор событий по PID процесса:

```
ausearch -p 6222
```

Выбор событий по исполняемому файлу:

```
ausearch -x '/usr/sbin/crond'
```

Выбор событий в диапазоне по дате-времени:

```
ausearch -ts 08/06/2020 20:59 -te 08/06/2020 21:59
```

Маршрут вебинара

- Логи
- rsyslog + logrotate
- journald
- abrt
- auditd
- **kdump**



kdump

kdump

- Для диагностики и анализа причин сбоев ядра в RedHat разработали kdump
- Создается два ядра: основное и аварийное
- При загрузке основного ядра под аварийное ядро выделяется определенный размер памяти
- При помощи утилиты kexec во время kernel panic основного ядра загружается аварийное и собирает дамп
- Анализ результатов сбоя производится утилитой crash

Ваши вопросы?

Рефлексия

- Назовите пожалуйста 3 момента, которые вам запомнились в процессе занятия
- Что вы будете применять в работе из сегодняшнего вебинара?

Заполните, пожалуйста,
опрос о занятии по
ссылке в чате

Приходите на следующие вебинары

Спасибо за внимание!