Лабораторная работа №7.

Аудит системных событий в Linux

*Цель работы:* Изучение возможностей инструмента *auditd* для регистрации событий безопасности в системе.

1. Установка и настройка *auditd*

Основным инструментом для сбора системных событий является *auditd* (Audit Daemon). Auditd был создан для тесного взаимодействия с ядром системы – во время своей работы инструмент отслеживает системные вызовы и может записывать события, связанные с файлами (чтение, запись, выполнение, изменение прав). Таким образом, с его помощью можно отслеживать практически любые события, происходящие в системе.

Плюсы *auditd*:

* работает на низком уровне мониторинга — отслеживает системные вызовы и действия с файлами;
* имеет неплохой набор утилит в комплекте для удобства работы;
* постоянно развивается и обновляется;
* бесплатен и легко устанавливается. Минусы *auditd*:
* большинство событий, возникающих при атаках, характерных для конкретного приложения, практически невозможно отслеживать поскольку на уровне системных вызовов и работе с файлами трудно отличить взлом от нормальной работы приложения. Такие события лучше отслеживать на уровне самих приложений;
* замедляет работу системы;
* не слишком гибок в настройке правил;
* на данный момент не лучший инструмент для работы с контейнерами.

В ОС “Альт Рабочая станция 10.2” пакет *auditd* установлен по умолчанию.

|  |  |
| --- | --- |
| Имя хоста | IP-адрес |
| alt-1 (сервер) | 192.168.10.10/24 |
| alt-2 (клиент) | 192.168.10.20/24 |

Примечание! Локальная сеть между машинами настроена по умолчанию.

Зайдем под учетной записью суперпользователя на хосте alt-2.

Запустим и добавим в автозагрузку службу *auditd*: # systemctl enable --now auditd

Основным конфигурационным файлом службы *auditd* является

*/etc/audit/auditd.conf*. Он определяет различные параметры, которые управляют поведением службы аудита, включая способ хранения журналов, их размер, уровень подробности и другие важные аспекты.

# cat /etc/audit/auditd.conf

1. Конфигурация правил аудита

Файлы конфигурации хранятся в */etc/audit/*. Правила желательно хранить в */etc/audit/rules.d/\*.rules*, по умолчанию доступ к этой директории только у суперпользователя. Обратим внимание на то, что файл с правилами в этой директории должен иметь название *\*.rules*, иначе *auditd* не прочитает его без явного указания. Если вы решили хранить правила в другом месте, то владелец файла должен быть суперпользователь. Также рекомендуется выставить группу файла *root* и права *600*, чтобы никто кроме суперпользователя не мог работать с файлом конфигурации *auditd*, т.к. зная что логируется, атакующий может избежать обнаружения. То же самое касается и файлов с правилами для других инструментов.

Для каждого из регистрируемых событий в журналах указывается следующая информация:

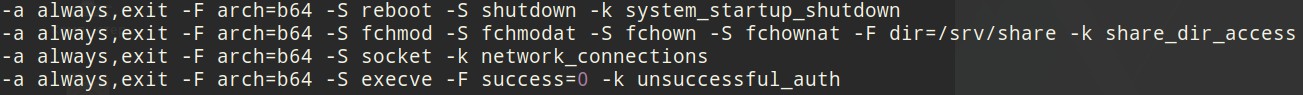
* дата и время;
* субъект, осуществляющий регистрируемое действие;
* тип события (если регистрируется запрос на доступ, то указываются объект и тип доступа);
* успешность осуществления события (обслужен запрос на доступ или нет).

Правила для логирования можно добавлять следующими способами:

1. Записать его в файл */etc/audit/rules.d/<имя файла>.rules* и перезапустить сервис.
2. Записать в файл по произвольному пути и указать его явно с помощью утилиты *auditctl* с опцией *-R*.
3. Добавить правило также можно с помощью утилиты *auditctl* с опцией

*-а*.

Создадим файл */etc/audit/rules.d/audit1.rules*:

# vim /etc/audit/rules.d/audit1.rules Добавим в файл следующие правила:

Основные параметры правил:

*-a [list,action],[action,list]* добавляет правило в конец списка правил.

Основные варианты списков *list*:

*exit* добавляет правило к списку, отвечающему за точки выхода из системных вызовов. Список применяется, когда необходимо создать событие для аудита, привязанное к точкам выхода из системных вызовов.

*exclude* добавляет правило к списку, отвечающего за фильтрацию событий определенного типа. Список используется, чтобы отфильтровывать ненужные события.

Варианты действий *action*:

*always* устанавливает контекст аудита. Всегда заполнять его во время входа в системный вызов и всегда генерировать запись во время выхода из системного вызова.

*never* не генерирует никаких записей.

Опция -F задает поле сравнения для правила. Атрибутами поля могут быть объект, операция и значение. Таким образом можно задать до 64 полей сравнения в одной команде. Каждое новое поле должно начинаться с -F. Аудит будет генерировать запись. Если произошло совпадение по всем полями сравнения. Допустимо использование таких операторов как «равно», «не равно», «меньше», «больше» и т.д.

Поскольку система ориентируется на номера (не названия) системных вызовов, а для многих системных вызовов номера отличаются для 32 и 64 разрядных систем, то необходимо указывать для какой архитектуры написано правило.

Разберем правила:

1. е правило отслеживает системные вызовы *reboot* и *shutdown*, которые используются для перезагрузки и завершения работы системы. Ключевое слово *system\_startup\_shutdown* позволяет легко идентифицировать и фильтровать эти события в журналах аудита.
2. е правило регистрирует изменения прав доступа (системные вызовы

*fchmod*, *fchmodat*, *fchown*, *fchownat*) и владельца файлов в директории

*/srv/share*. Использование ключа *share\_dir\_access* позволяет отслеживать все операции, связанные с этой директорией.

1. е правило отслеживает системный вызов *socket*, который используется для создания сетевых соединений. Ключ *network\_connections* помогает идентифицировать события, связанные с сетевой активностью.
2. е правило фиксирует события, когда системный вызов *execve* (используемый для выполнения программ) завершается неудачей (успех равен 0). Ключевое слово *unsuccessful\_auth* позволяет отслеживать неудачные попытки аутентификации или выполнения команд.

Во всех правилах используется ключ для удобного поиска в логах.

Перед загрузкой правил создадим директорию */srv/share*: # mkdir /srv/share

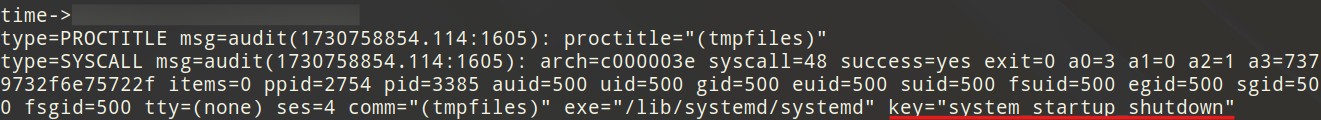
Теперь можно загружать правила из каталога */etc/audit/rules.d/*: # augenrules --load

Проверим работу правил:

1. Перезагрузим систему:

# reboot

Проверим журнал:

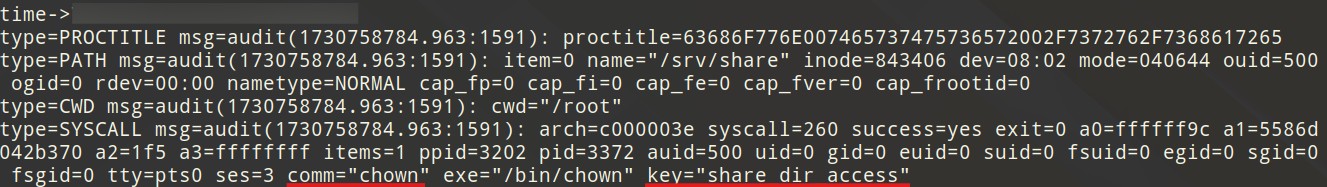
# ausearch -k system\_startup\_shutdown

1. Создадим пользователя *testuser* c паролем *P@ssw0rd*: # useradd testuser

# passwd testuser

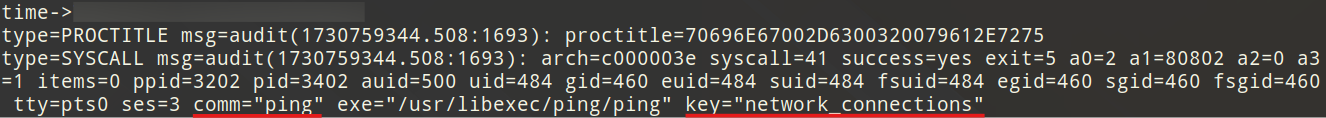
Назначим его владельцем директории */srv/share*: # chown testuser /srv/share

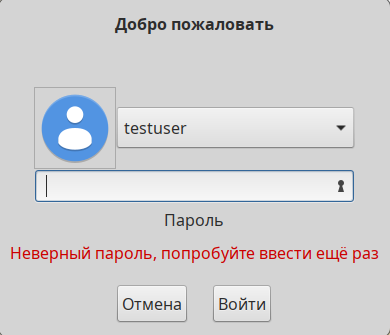
Проверим журнал:

# ausearch -k share\_dir\_access

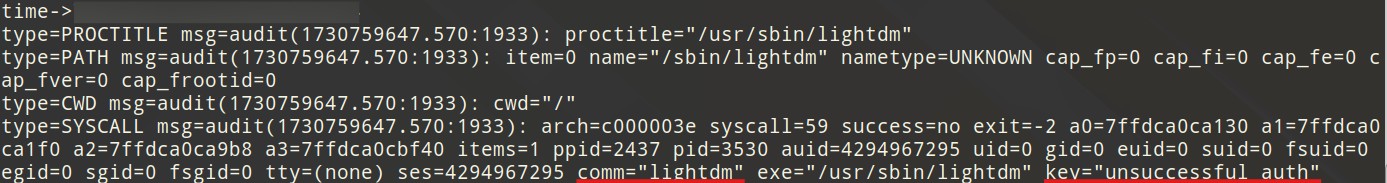
1. Отправим запросы к сайту *ya.ru*: # ping -c 4 ya.ru

Проверим журнал:

# ausearch -k network\_connections

1. Выполним попытку неудачного входа в систему из графической оболочки.

Проверим журнал:

# ausearch -k unsuccessful\_auth

# ЗАДАНИЯ

1. Напишите правило для аудита выполнения команд с повышенными привилегиями (например, *sudo*/*su*).
2. Напишите правило, которое будет фиксировать события входа и выхода пользователей из системы.
3. Напишите правило, которое будет фиксировать все попытки подключения к определенным портам (например, порт 22 для подключения по ssh).
4. Напишите правило для аудита изменений конфигурационных файлов в директории */etc*.

В отчете укажите сами правила и приведите несколько вариантов их выполнения.

Используйте ключи для удобного поиска в логах.

3. Удаленный мониторинг

События, зафиксированные с помощью *auditd*, можно передавать в

*rsyslog* для дальнейшей обработки, фильтрации и отправки на сервер.

Установим пакет *rsyslog* как на сервере, так и на клиенте: # apt-get update && apt-get install rsyslog

Запустим и добавим в автозагрузку *rsyslog*:

# systemctl enable --now rsyslog

Основным конфигурационным файлов *rsyslog* является */etc/rsyslog.conf*. В нем подключены все файлы из папки */etc/rsyslog.d/* с помощью директивы *include* в самом начале файла:

# cat /etc/rsyslog.conf

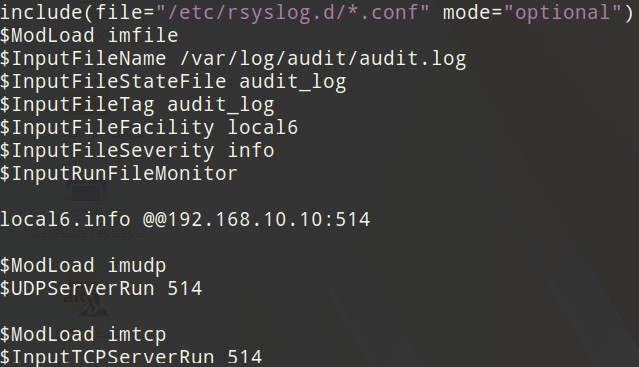
В этих файлах могут содержаться дополнительные настройки, например, аутентификация на *rsyslog* сервере.

Главный конфигурационный файл обеспечивает управление локальными логами по умолчанию, но для работы через сеть нужно добавить настройки.

Синтаксис конфигурационного файла:

$переменная значение

Все директивы начинаются со знака доллара, содержат имя переменной, а дальше связанное с ней значение. Так выглядит каждая строка конфигурационного файла. В его первой части размещены общие настройки программы и загрузка модулей, во второй – правила сортировки и фильтрации лог файлов.

Отредактируем конфигурационный файл */etc/rsyslog.conf* на хосте *alt-2*: # vim /etc/rsyslog.conf

Разберем основные части конфигурационного файла */etc/rsyslog.conf*:

$ModLoad imfile

Загружает модуль *imfile*, который позволяет *rsyslog* читать и обрабатывать содержимое файлов. Без этого модуля *rsyslog* не сможет отслеживать изменения в указанных файлах.

$InputFileName /var/log/audit/audit.log

Указывает путь к файлу, который будет отслеживаться. В данном случае это файл журнала аудита. Основной файл, который *rsyslog* будет мониторить на наличие новых записей.

$InputFileStateFile audit\_log

Указывает имя файла состояния для отслеживаемого файла. Данный файл хранит информацию о том, где *rsyslog* остановился при последнем чтении, чтобы избежать повторного считывания уже обработанных данных. Так *rsyslog* эффективно обрабатывает новые записи, не теряя или не дублируя их.

$InputFileTag audit\_log

Задает тег для записей, которые будут созданы из этого файла. Тег добавляется к каждой записи, чтобы идентифицировать источник. Так удобнее фильтровать и идентифицировать логи при их обработке или отправке.

$InputFileFacility local6

Указывает уровень службы (facility), к которой будут относиться логи из этого файла. В данном случае используется local6, который обычно резервируется для пользовательских приложений.

$InputFileSeverity info

Указывает уровень важности (severity) логов, создаваемых из этого файла. В данном случае используется уровень info. Уровни важности помогают фильтровать логи по критичности.

$InputRunFileMonitor

Запускает мониторинг файла */var/log/audit/audit.log* с учетом всех предыдущих настроек. Фактически активирует все настройки, сделанные ранее для мониторинга файла.

Для определения набора правил для обработки удаленных логов необходим следующий формат:

facility.severity\_level destination (where to store log) где:

*facility* – тип сообщения о процессе/приложении, к которому относятся *auth, cron, daemon, kernel, local0..local7*. Использование \* означает все объекты.

*severity\_level* – тип сообщения журнала: *out-0, alert-1, crit-2, err-3, warn-4, notice-5, info-6, debug-7*. Использование \* означает все уровни severity, а *none* означает ни одного уровня *severity*.

*destination* – локальный файл, либо удаленный сервер *rsyslog* в формате *IP:порт*.

local6.info @@192.168.10.10:514

Указывает, что все логи с тегом *local6* и уровнем важности *info* должны быть отправлены на удаленный сервер по адресу 192.168.10.10 через UDP.

$ModLoad imudp

Загружает модуль *imudp*, который обрабатывает все входящие сообщения по протоколу *UDP*.

$UDPServerRun 514

Настраивает входящий *UDP*-порт для получения логов. По умолчанию используется порт 514.

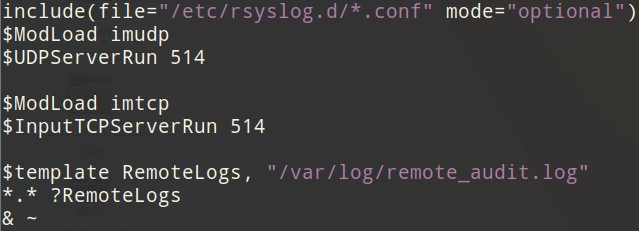
$ModLoad imtcp

Загружает модуль *imtcp*, который обрабатывает все входящие сообщения по протоколу *TCP*.

$InputTCPServerRun 514

Настраивает входящий *TCP*-порт для получения логов. По умолчанию используется порт 514.

После редактирования файла */etc/rsyslog.conf* перезапускаем rsyslog: # systemctl restart rsyslog

Зайдем в учетную запись суперпользователя на хосте *alt-1*. Отредактируем конфигурационный файл */etc/rsyslog.conf*: # vim /etc/rsyslog.conf

Разберем основные части конфигурационного файла */etc/rsyslog.conf*:

$template RemoteLogs, "/var/log/remote\_audit.log"

Директива *$template* используется для определения нового шаблона формата для записи логов. В данном случае создается шаблон с именем *RemoteLogs*. Имя шаблона *RemoteLogs* будет использоваться в последующих строках конфигурации.

\*.\* ?RemoteLogs

\*.\* – правило фильтрации, которое указывает, что будут обрабатываться все сообщения (все уровни важности и все уровни служб). Первая звездочка (\*) соответствует любому уровню службы *facility*, а вторая звездочка (\*) — любому уровню важности *severity*.

*?RemoteLogs* указывает, что сообщения, соответствующие этому правилу, должны быть записаны с использованием шаблона *RemoteLogs*,

который мы определили ранее. Т.е. все логи будут записаны в файл

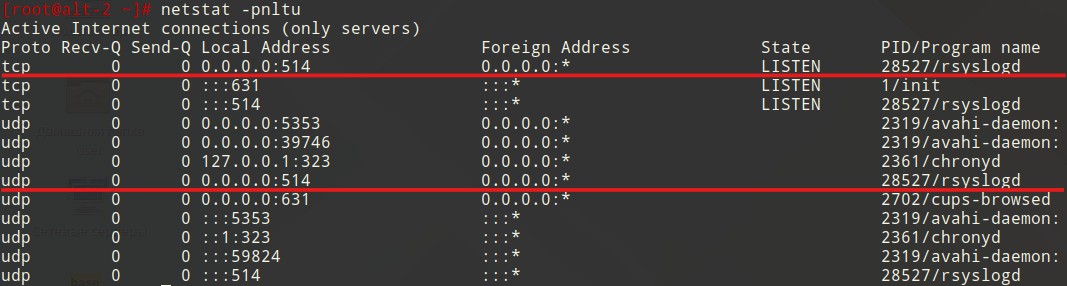
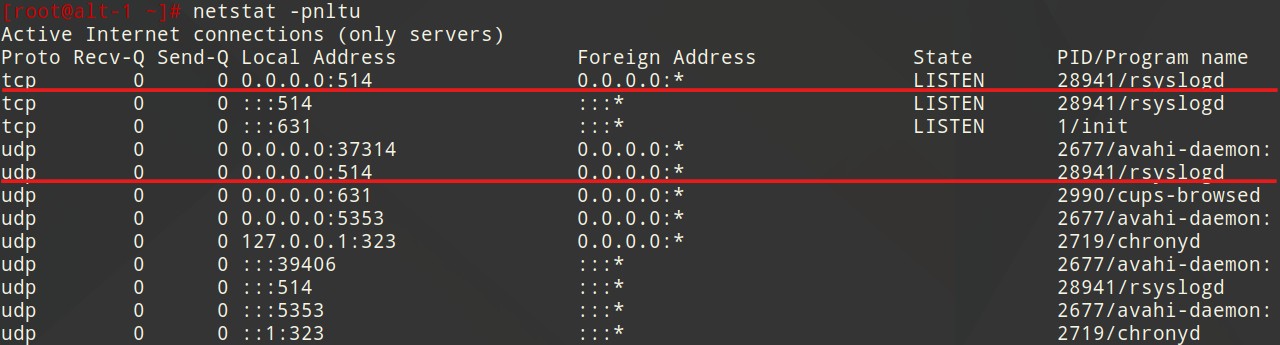
*/var/log/remote\_audit.log.*

& ~

Символ *&* используется для указания того, что обработка сообщения должна быть завершена после выполнения предыдущего правила. Т.е. сообщение не будет отправлено дальше по другим правилам конфигурации.

Символ *~* указывает на то, что сообщение должно быть отброшено после его обработки. Таким образом, после того как сообщение будет записано в файл */var/log/remote\_audit.log*, оно не будет обрабатываться другими правилами.

После редактирования файла */etc/rsyslog.conf* перезапускаем rsyslog: # systemctl restart rsyslog

Важно! Проверьте, что порты 514 открыты и на сервере, и на клиенте: # netstat -pnltu

Утилита *netstat* показывает сетевые соединения, таблицы маршрутизации, статистику интерфейсов и другую информацию о сетевых соединениях.

Опция -p (--program) показывает идентификатор процесса (PID) и имя программы, которая использует каждое соединение или прослушиваемый порт.

Опция -n (--numeric) выводит адреса и порты в числовом формате. Таким образом вместо разрешения имен хостов и сервисов будет отображаться IP-адрес и номер порта.

Опция -l (--listening) ограничивает вывод только теми соединениями, которые находятся в состоянии "прослушивания". Т.е. в выводе будут только те порты, которые ожидают входящих соединений.

Опция -t (--tcp) отображает только TCP-соединения, а опция -u (или

--udp) – UDP-соединения.

Проверим настройку *rsyslog*.

На хосте *alt-1* запустим команду для отслеживания изменений в файле

*/var/log/remote\_audit.log*:

# tail -f /var/log/remote\_audit.log

На хосте *alt-2* отправим запросы к сайту *ya.ru*:

# ping -c 4 ya.ru

В результате в файле */var/log/remote\_audit.log* должны появиться записи.

# ЗАДАНИЯ

На сервере *alt-1* создайте правило, которые логи уровня *err* для тега audit\_log записывал в файл */var/log/audit\_errors.log*, а логи уровня *info* –

*/var/log/audit\_info.log*.

Проверьте, что логи с разными уровнями важности появляются в соответствующих файлах на сервере *alt-1*.