ЛЕКЦИЯ:

Работа с логированием событий в операционных системах

ВОПРОСЫ:

1. Логирование - одно из трех модели «AAA» (Authentication, Authorization, Accounting).

2. Логирование — правильное использов и анализ.

1. Логирование - одно из трех модели «AAA» (Authentication, Authorization, Accounting).

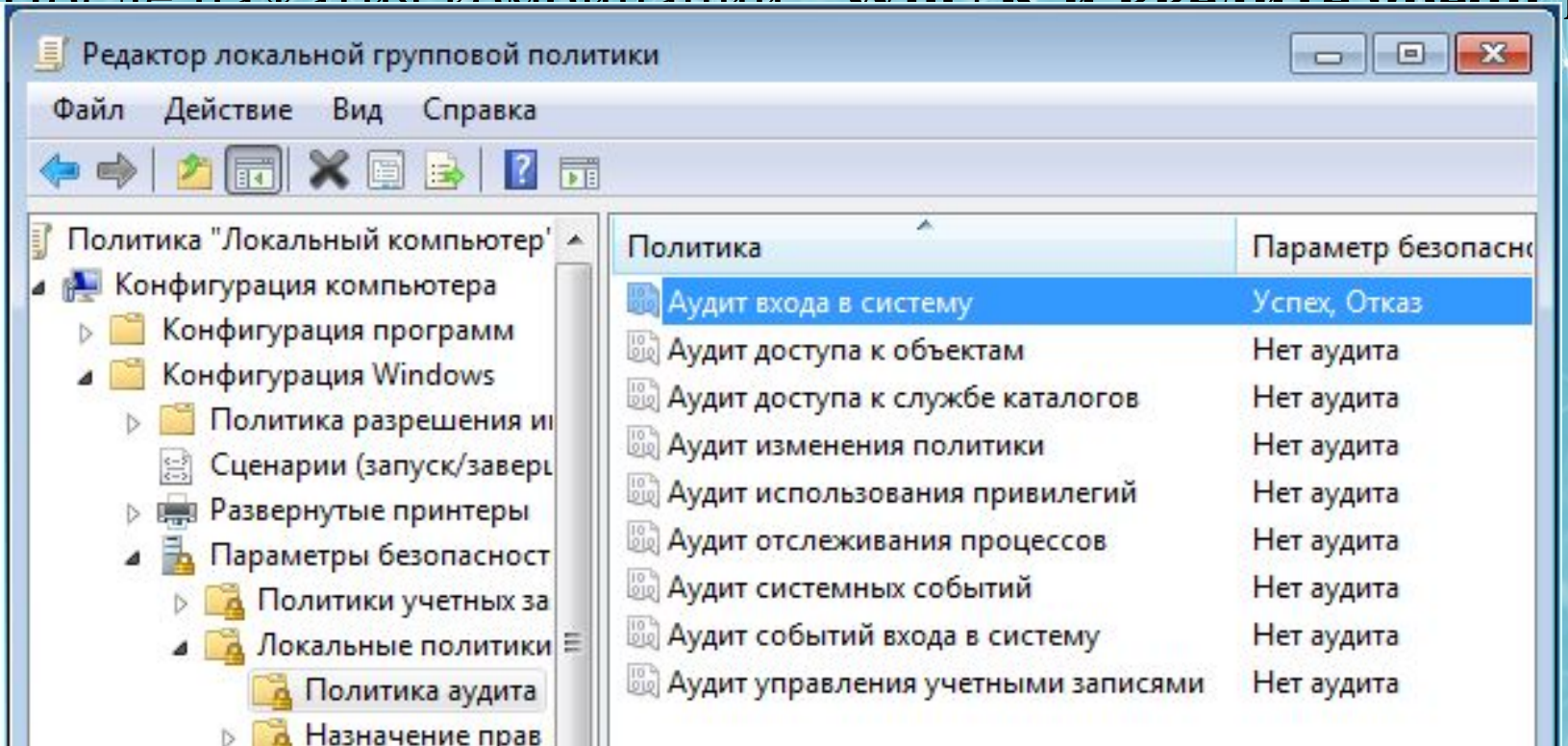
ВОПРОС:

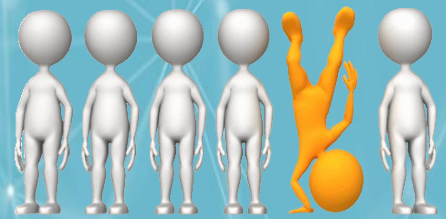
Что такое «ЛОГИРОВАНИЕ»?

Логи (лог-файлы) — **это файлы**, содержащие системную информацию работы сервера или компьютера, в которые заносятся определенные действия пользователя или программы. Иногда также употребляется русскоязычный аналог понятия — журнал. **Их предназначение — протоколирование операций**, выполняемых на машине, для дальнейшего анализа администратором. Регулярный просмотр журналов позволит определить ошибки в работе системы в целом, конкретного сервиса или сайта (особенно скрытые ошибки, которые не выводятся при просмотре в браузере), диагностировать злонамеренную активность, собрать статистику посещений.

ЛОГИРОВАНИЕ В WINDOWS

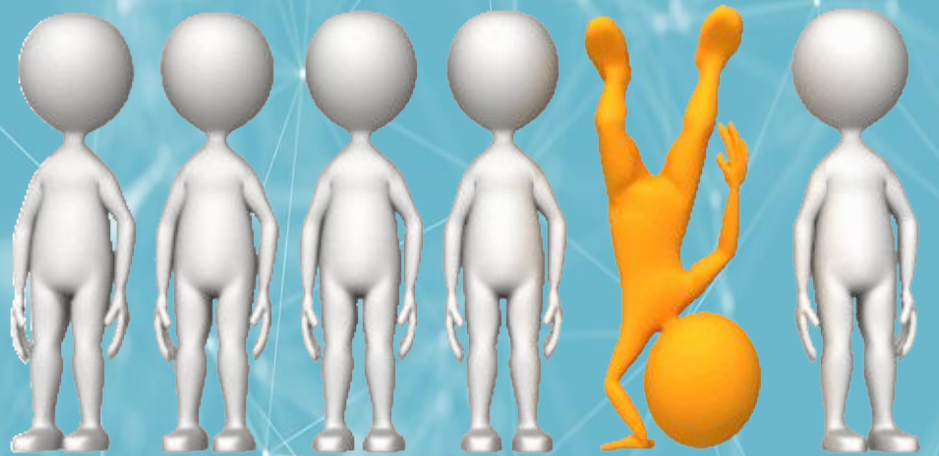
После нажатия комбинации “Win+R и введите gpedit.msc” —



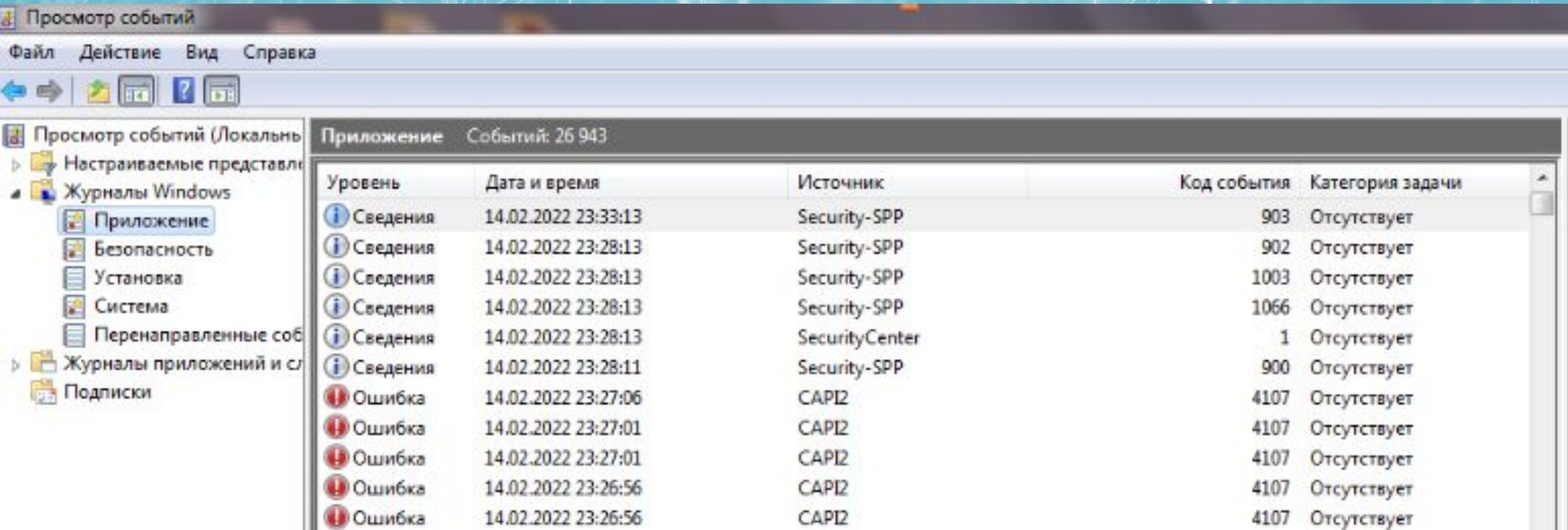


ЛОГИРОВАНИЕ В WINDOWS

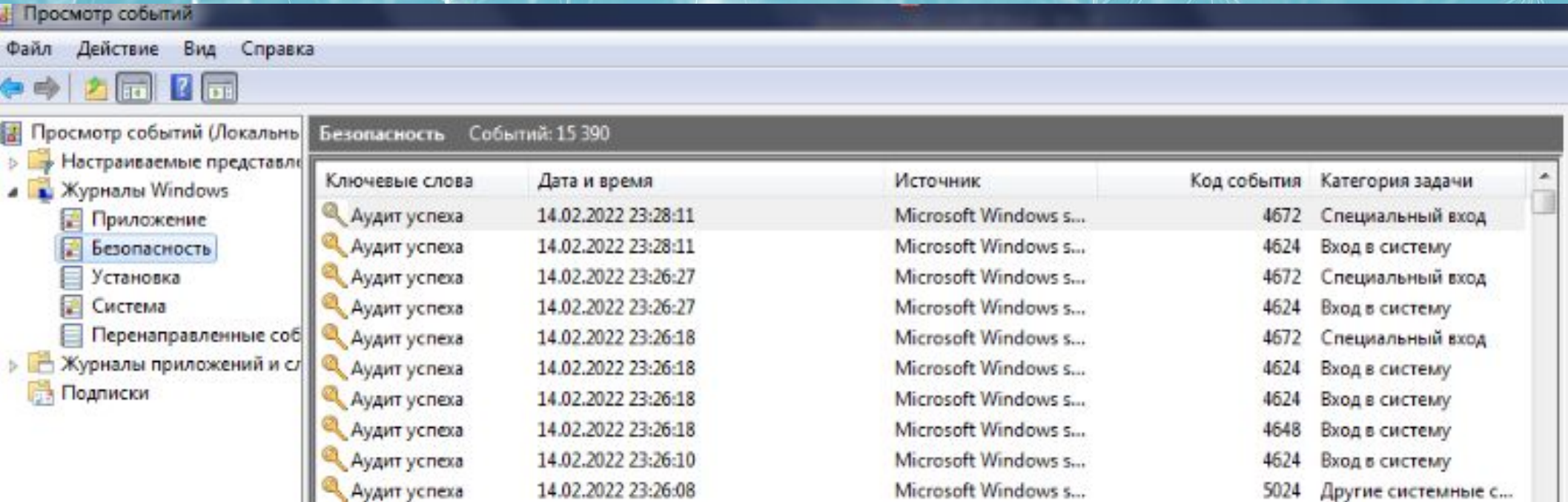
После нажатия комбинации **“Win+R и введите eventvwr.msc”** в любой системе Windows вы попадаете в просмотр событий. У вас откроется окно, где нужно развернуть Журналы Windows. В данном окне можно просмотреть все программы, которые открывались на ОС и, если была допущена ошибка, она также отобразится.



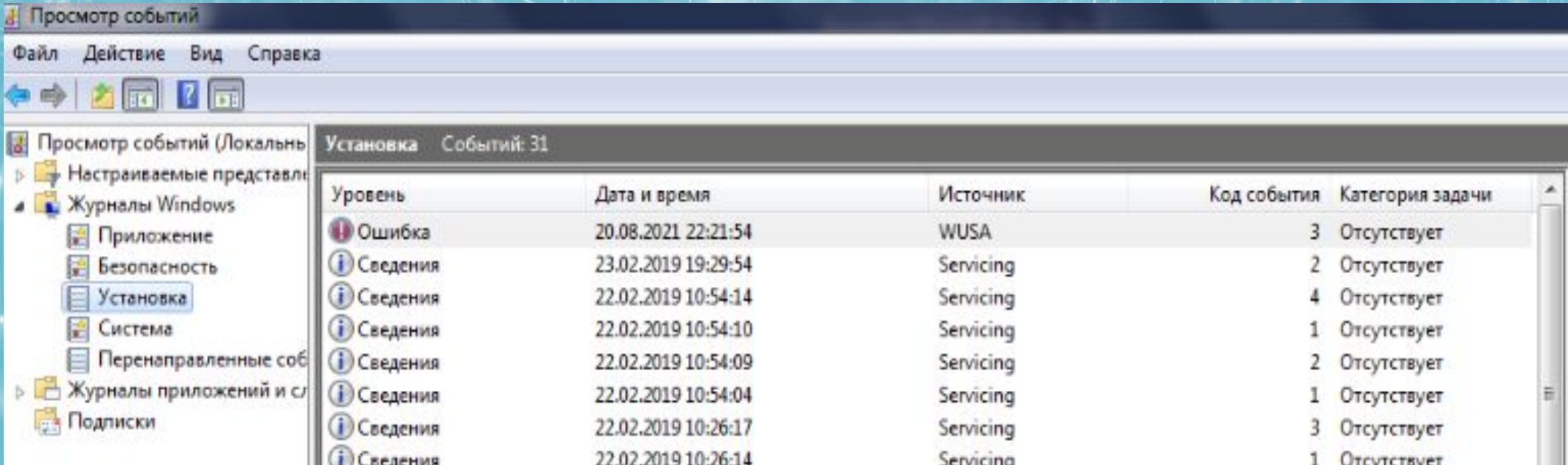
Приложение – хранит важные события, связанные с конкретным приложением. Эти данные помогут системному администратору установить причину отказа той или иной программы. Аудит журнал поможет понять, что и кто и когда делал. Также отображается информация по запросам получения доступов.



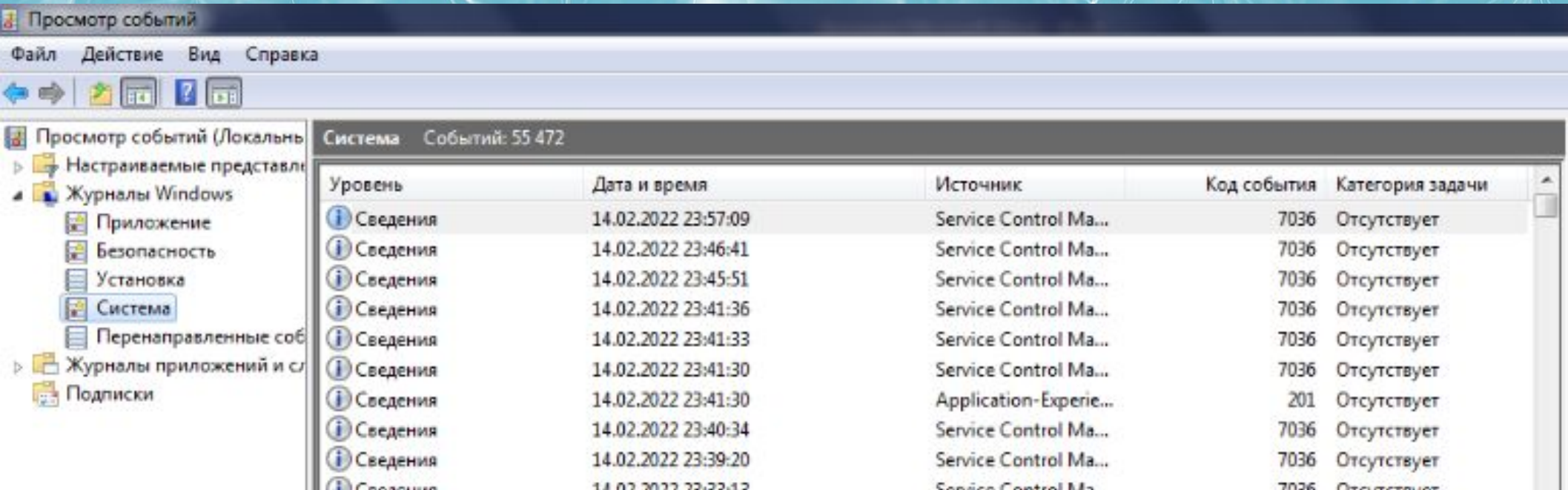
Безопасность – хранит события, связанные с безопасностью (такие как: вход/выход из системы, управление учётными записями, изменение разрешений и прав доступа к файлам и папкам).



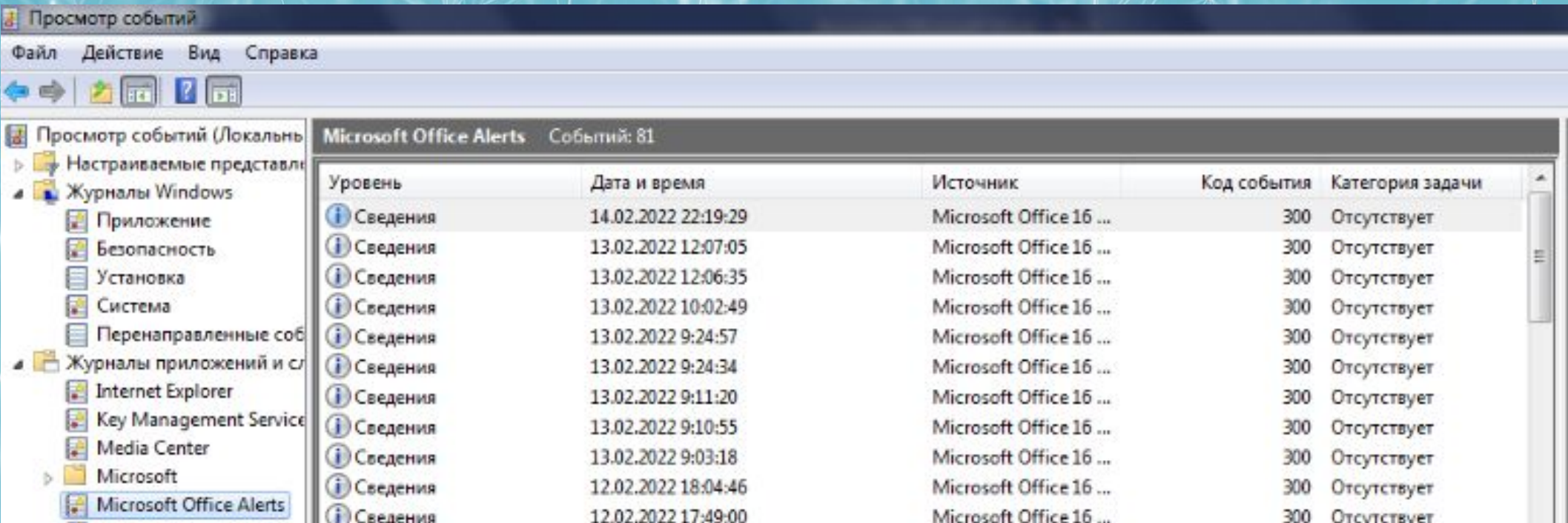
В пункте **Установка** можно посмотреть логи ОС Windows, например, программы и обновления системы.



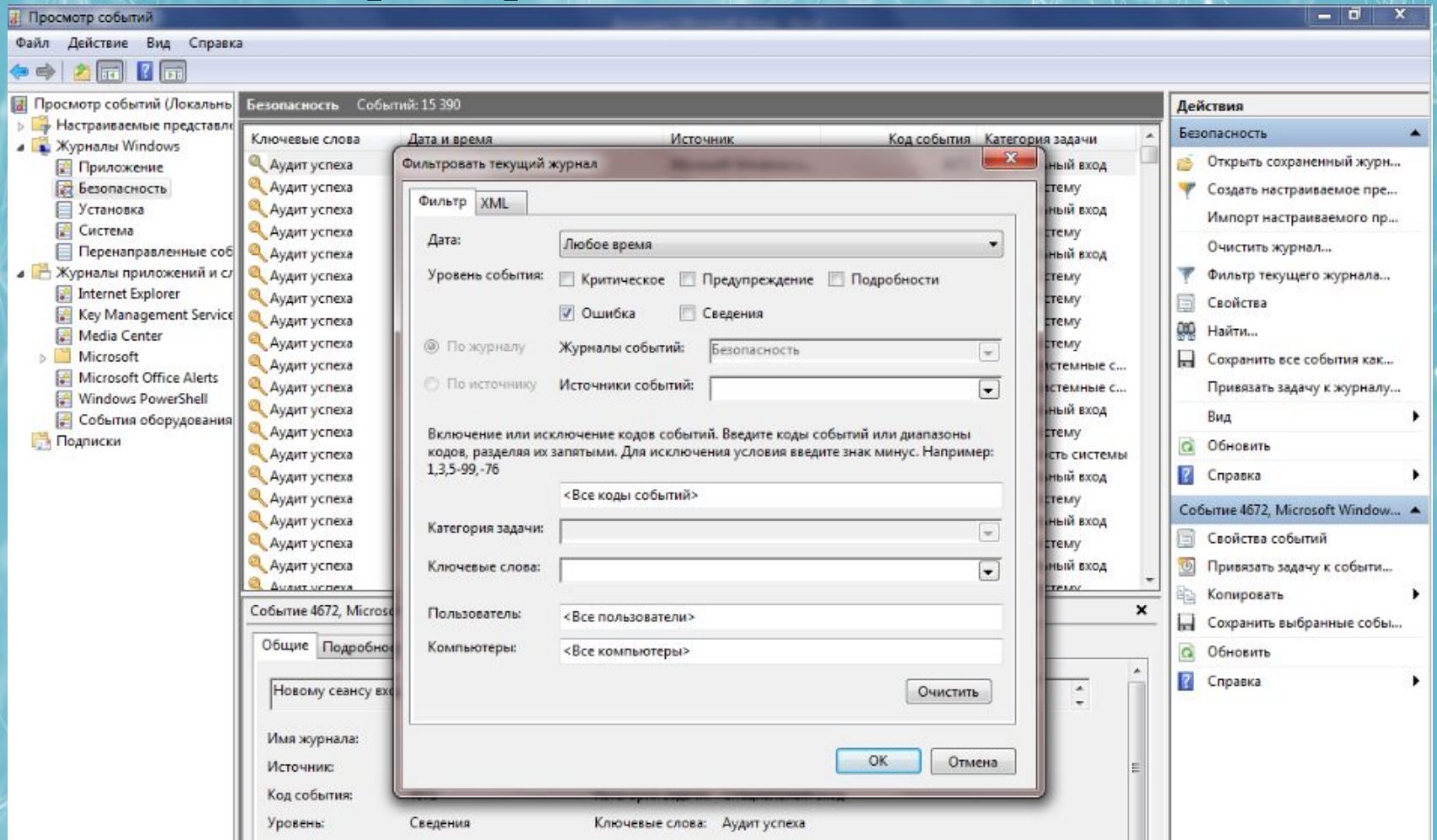
Система - наиболее важный журнал. С его помощью можно определить большинство ошибок ОС. К примеру, у вас появлялся синий экран. В данном журнале можно определить причину его появления.



Логи windows - для более специфических служб.



утилите **“Просмотр событий”** предусмотрена возможность поиска и фильтрации событий:



) **Уровени событий:**

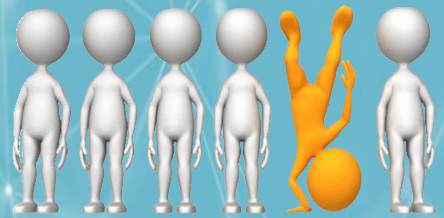
Критическое
Ошибка
Предупреждение
Сведения
Подробности

ЛОГИРОВАНИЕ В WINDOWS

Политика аудита входа пользователя в домен!

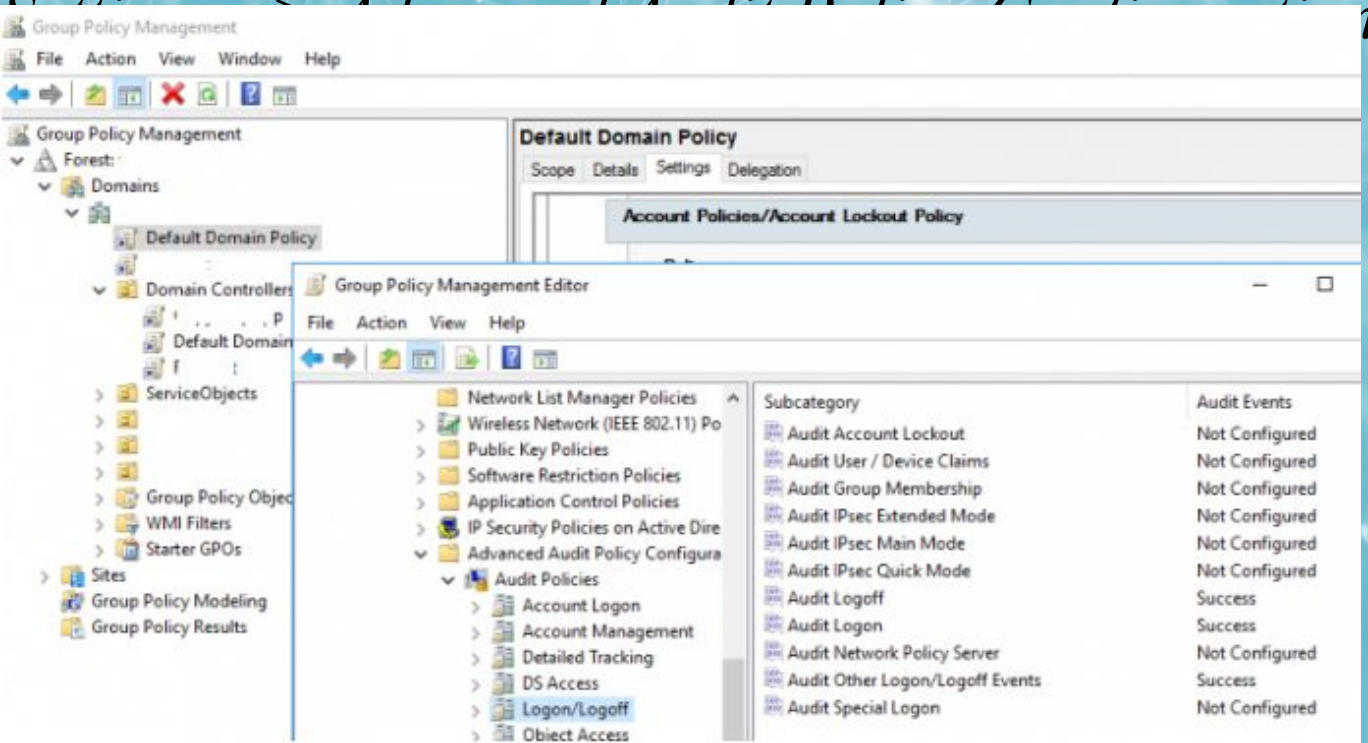
Чтобы в журналах контроллеров домена отображалась информация об успешном/неуспешном входе в систему, нужно включить политику аудита событий входа пользователей.

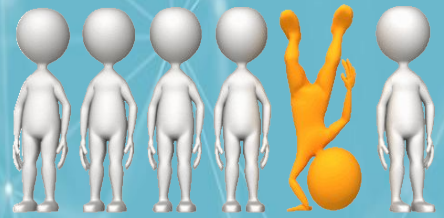
Запустите редактор доменных **GPO – GPMC.msc**



ЛОГИРОВАНИЕ В WINDOWS

Откройте настройки Default Domain Policy и перейдите в раздел *Computer Configuration -> Policies -> Windows Settings -> Security* [illegible] *-> Audit Policies ->*

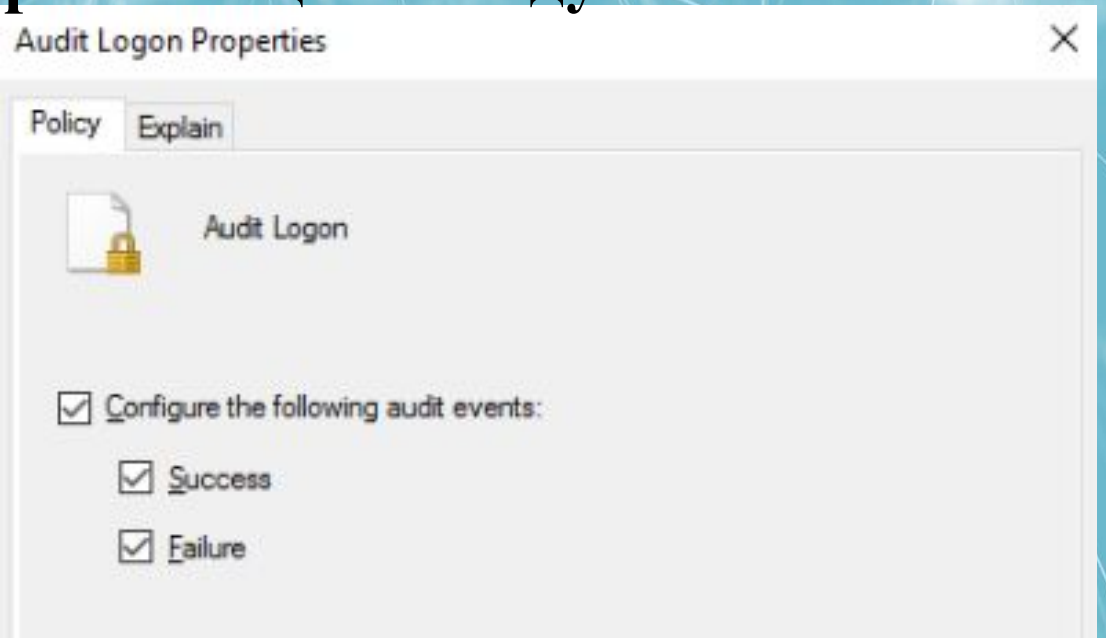


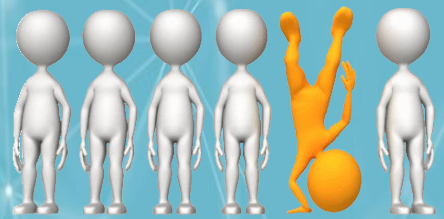


ЛОГИРОВАНИЕ В WINDOWS

Включите две политики аудита *(Audit Logon и Audit Other Logon/Logoff Events)*. Чтобы в журналах Security на DC и компьютерах регистрировались как успешные, так и неуспешные политики входа, выберите в настройках политика аудита опцию *Success* и *Failure*.

Сохраните изменения в GPO – подождите 90 минут, без учета репликации между DC!





ЛОГИРОВАНИЕ В LINUX

Большинство же лог файлов содержится в директории **/var/log.**

• **/var/log/syslog или /var/log/messages** содержит глобальный системный журнал, в котором пишутся сообщения с момента запуска системы, от ядра Linux, различных служб, обнаруженных устройствах, сетевых интерфейсов и много другого.

• **/var/log/auth.log или /var/log/secure** — информация об авторизации пользователей, включая удачные и неудачные попытки входа в систему, а также задействованные механизмы аутентификации.

• **/var/log/dmesg — драйвера устройств.** Одноименной командой можно просмотреть вывод содержимого файла. Размер журнала ограничен, когда файл достигнет своего предела, старые сообщения будут перезаписаны более новыми. Задав ключ --level= можно отфильтровать вывод по критерию значимости.

• Поддерживаемые уровни журналирования (приоритеты):

emerg - система неиспользуемая
alert - действие должно быть произведено немедленно
crit - условия критичности
err - условия ошибок
warn - условия предупреждений
notice - обычные, но значимые условия
info - информационный
debug - отладочные сообщения

/var/log/alternatives.log — Вывод программы update-alternatives, в котором находятся символические ссылки на команды или библиотеки по умолчанию.

- **/var/log/anaconda.log** — Записи, зарегистрированные во время установки системы.
- **/var/log/audit** — Записи, созданные службой аудита auditd.
- **/var/log/boot.log** — Информация, которая пишется при загрузке операционной системы.
- **/var/log/cron** — Отчет службы crond об исполняемых командах и сообщения от самих команд.
- **/var/log/cups** — Все, что связано с печатью и принтерами.
- **/var/log/faillog** — Неудачные попытки входа в систему. Очень полезно при проверке угроз в системе безопасности, хакерских атаках, попыток взлома методом перебора. Прочитать содержимое можно с помощью команды faillog.
- **var/log/kern.log** — Журнал содержит сообщения от ядра и предупреждения, которые могут быть полезны при устранении ошибок пользовательских модулей встроенных в ядро.
- **var/log/pm-powersave.log** — Сообщения службы экономии заряда батареи.
- **/var/log/samba/** — Логи файлового сервера Samba, который используется для доступа к общим папкам Windows и предоставления доступа пользователям Windows к общим папкам Linux.

/var/log/spooler — Для представителей старой школы, содержит сообщения USENET. Чаще всего бывает пустым и заброшенным.

• **/var/log/Xorg.0.log** — Логи X сервера. Чаще всего бесполезны, но если в них есть строки начинающиеся с EE, то следует обратить на них внимание.

Для каждого дистрибутива будет 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/dpkg.log** — Для программ установленных с помощью dpkg в Debian Linux и всем семействе родственных дистрибутивах.

И немного бинарных журналов учета пользовательских сессий.

• **/var/log/lastlog** — Последняя сессия пользователей. Прочитать можно командой last.

• **/var/log/tallylog** — Аудит неудачных попыток входа в систему. Вывод на экран с помощью утилиты pam_tally2.

• **/var/log/btmp** — Еже один журнал записи неудачных попыток входа в систему. Просто так, на всякий случай, если вы еще не догадались где следует искать следы активности взломщиков.

• **/var/log/utmp** — Список входов пользователей в систему на данный момент.

• **/var/log/wtmp** — Еще один журнал записи входа пользователей в систему. Вывод на экран командой utmpdump.

Так как операционная система, даже такая замечательная как Linux, сама по себе никакой ощутимой пользы не несет в себе, то скорее всего на сервере или рабочей станции будет крутится база данных, веб сервер, разнообразные приложения. Каждое приложения или служба может иметь свой собственный файл или каталог журналов событий и ошибок. Всех их естественно невозможно перечислить, лишь некоторые.

- **/var/log/mysql/** — Лог базы данных MySQL.
- **/var/log/httpd/ или /var/log/apache2/** — Лог веб сервера Apache, журнал доступа находится в access_log, а ошибки — в error_log.
- **/var/log/lighthttpd/** — Лог веб сервера lighttpd.

Инструменты при работе с логами

Команда head выводит начальные строки (по умолчанию — 10) из одного или нескольких документов. Также она может показывать данные, которые передает на вывод другая утилита.

Синтаксис у команды head следующий:

$ head опции файл

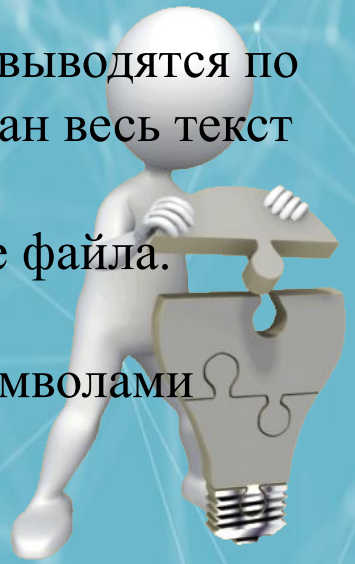
-c (--bytes) — позволяет задавать количество текста не в строках, а в байтах. При записи в виде --bytes=[-]NUM выводит на экран все содержимое файла, кроме NUM байт, расположенных в конце документа.

-n (--lines) — показывает заданное количество строк вместо 10, которые выводятся по умолчанию. Если записать эту опцию в виде --lines=[-]NUM, будет показан весь текст кроме последних NUM строк.

-q (--quiet, --silent) — выводит только текст, не добавляя к нему название файла.

-v (--verbose) — перед текстом выводит название файла.

-z (--zero-terminated) — символы перехода на новую строку заменяет символами завершения строк.



Инструменты при работе с логами

Название команды - это сокращения от слова catenate. По сути, задача команды cat очень проста - она читает данные из файла или стандартного ввода и выводит их на экран. Это все, чем занимается утилита. Но с помощью ее опций и операторов перенаправления вывода можно сделать очень многое. Сначала рассмотрим синтаксис утилиты:

$ cat опции файл1 файл2 .

Вы можете передать утилите несколько файлов и тогда их содержимое будет выведено поочередно, без разделителей. Опции позволяют очень сильно видоизменить вывод и сделать именно то, что вам нужно.

Рассмотрим основные опции:

***-b -** нумеровать только непустые строки;*

***-E -** показывать символ $ в конце каждой строки;*

***-n -** нумеровать все строки;*

***-s -** удалять пустые повторяющиеся строки;*

***-T -** отображать табуляции;*

***-h -** отобразить справку;*

***-v -** версия утилиты.*

Инструменты при работе с логами

$ tail опции файл

По умолчанию утилита выводит десять последних строк из файла, но ее поведение можно настроить с помощью опций:

- **-c** - выводить указанное количество байт с конца файла;
- **-f** - обновлять информацию по мере появления новых строк в файле;
- **-n** - выводить указанное количество строк из конца файла;
- **--pid** - используется с опцией -f, позволяет завершить работу утилиты, когда завершится указанный процесс;
- **-q** - не выводить имена файлов;
- **--retry** - повторять попытки открыть файл, если он недоступен;
- **-v** - выводить подробную информацию о файле.

Инструменты при работе с логами

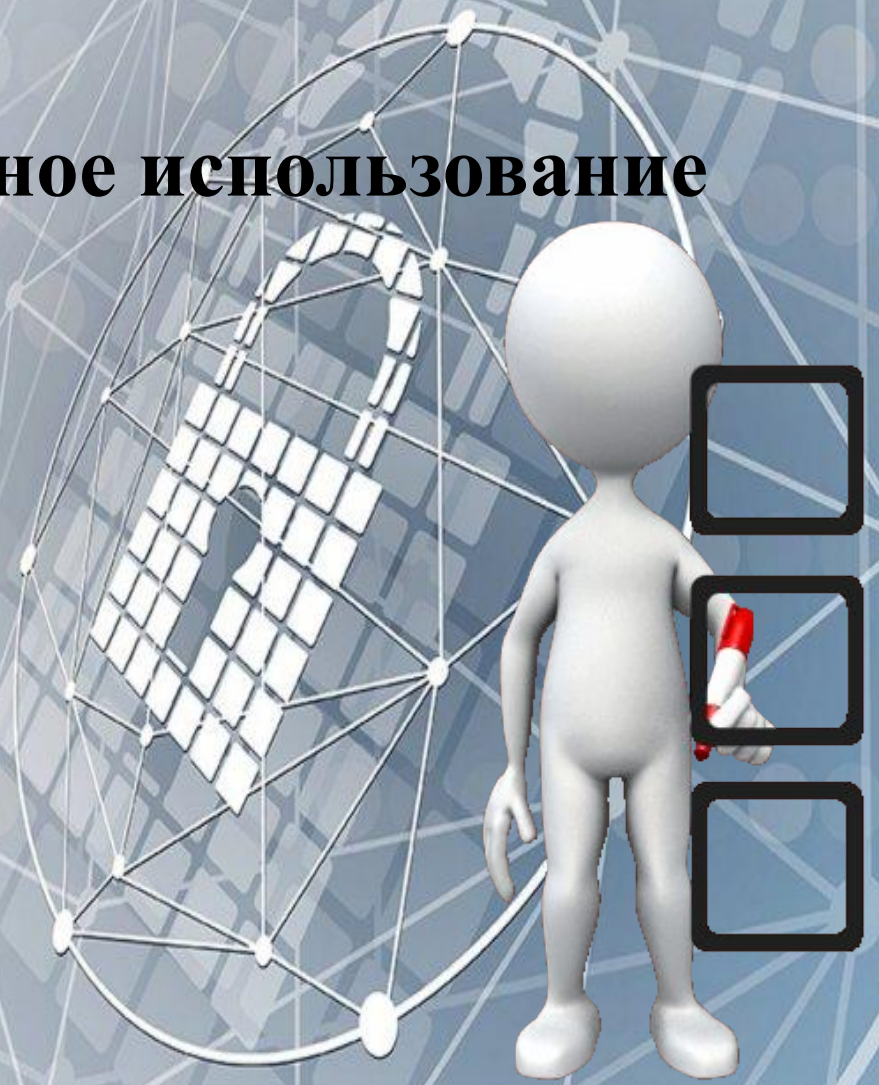
$ grep [опции] шаблон [имя файла...]

Или:

$ команда | grep [опции] шаблон

- **-b** - показывать номер блока перед строкой;
- **-c** - подсчитать количество вхождений шаблона;
- **-h** - не выводить имя файла в результатах поиска внутри файлов Linux;
- **-i** - не учитывать регистр;
- **- l** - отобразить только имена файлов, в которых найден шаблон;
- **-n** - показывать номер строки в файле;
- **-s** - не показывать сообщения об ошибках;
- **-v** - инвертировать поиск, выдавать все строки кроме тех, что содержат шаблон;
- **-w** - искать шаблон как слово, окружённое пробелами;
- **-e** - использовать регулярные выражения при поиске;
- **-An** - показать вхождение и n строк до него;
- **-Bn** - показать вхождение и n строк после него;
- **-Cn** - показать n строк до и после вхождения.

2. Логирование — правильное использование и анализ.

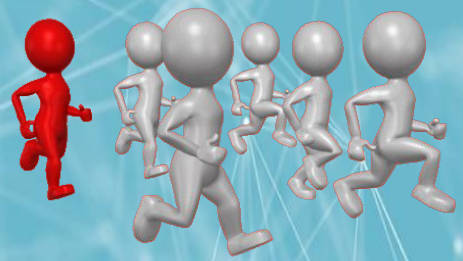


Правила работы с логами:

1. Определение источников журналов и используемых инструментов;
2. Копирование логов в другое место для их обработки и анализа;
3. Определение необходимых типов событий, с которыми будет проводиться работа;
4. Определение актуальности временного протоколирования логов;
5. Определение «аномальных» событий для инфраструктуры;
6. Составление time-line событий на основе анализа рассматриваемых журналов;
7. Формирование гипотезы исследуемого события.

Потенциальные источники логов безопасности

- Журналы операционной системы серверов и рабочих станций;
- Журналы приложений (например, веб-сервер, сервер баз данных);
- Журналы инструментов безопасности (например, антивирус, инструменты обнаружения изменений, системы обнаружения/предотвращения вторжений);
- Исходящие журналы прокси-сервера и журналы приложений конечных пользователей;
- Другие источники событий безопасности, не входящие в журналы.



Системы централизованного сбора

1 Graylog
2 Logstash
3 Fluentd
4 Flume
5 Octopussy
6 LOGalyze
7 LogPacker
8 Logwatch
9 Syslog-ng
10Inav

ELK расшифровывается как elasticsearch, logstash и kibana. Раньше это были три самостоятельных продукта, но в какой-то момент они стали принадлежать одной компании и развиваться в одном направлении. Каждый из этих инструментов (с небольшими оговорками ниже) является полноценным независимым open source продуктом, а все вместе они составляют мощное решение для широкого спектра задач сбора, хранения и анализа данных.

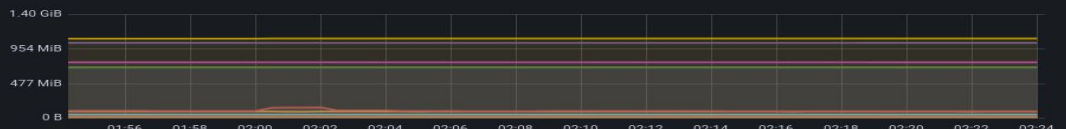
logstash – это утилита для сборки, фильтрации и последующего перенаправления в конечное хранилище данных.

elasticsearch – это решение для полнотекстового поиска, построенное поверх Apache Lucene, но с дополнительными удобствами, типа лёгкого масштабирования, репликации и прочих радостей, которые сделали elasticsearch очень удобным и хорошим решением для высоконагруженных проектов с большими объёмами данных.

kibana - красивое Angular.js приложение, позволяющее брать/искать данные по elasticsearch и строить множество красивых графиков.

namespace All pod All interval 5m

process memory



	max	avg	current
- resident	24.5 MiB	24.5 MiB	24.5 MiB
- resident	98.0 MiB	96.7 MiB	96.8 MiB

process memory deriv

	max	avg	current
- resident	0 B	0 B	
- resident	2.61 MiB	1.57 MiB	1.32

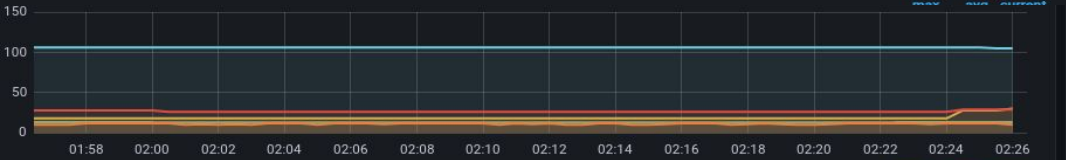
go memstats

	max	avg	current
- bytes allocated	8.78 MiB	6.21 MiB	6.05 MiB
- bytes allocated	32.4 MiB	24.1 MiB	23.0 MiB

go memstats deriv

	max	avg	current
- bytes allocated	8.11 KiB	-337 B	2
- bytes allocated	10.7 KiB	-1.7 KiB	-6

open fds



	max	avg	current
{__name__="process_open_fds", instance="goflow:8080", job="prometheus"}	13	13	13
{__name__="process_open_fds", instance="grafana:3000", job="grafana"}	30	18.7	30

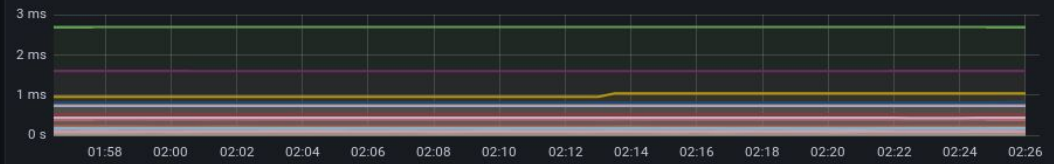
open fds deriv

	max	avg	current
{instance="goflow:8080", job="prometheus"}	0	0	0
{instance="grafana:3000", job="grafana"}	0.0524	0.00223	0.0524

Goroutines

	max	avg	current
{__name__="go_goroutines", instance="goflow:8080", job="prometheus"}	20	18.5	19
{__name__="go_goroutines", instance="grafana:3000", job="grafana"}	169	149	169

GC duration quantiles



	max	avg	current
: 0	22.7 µs	22.7 µs	22.7 µs
: 0.25	57.0 µs	55.9 µs	55.0 µs

СПАСИБО ЗА ВНИМАНИЕ!