Записки IT специалиста

Технический блог специалистов ООО"Интерфейс"

- Главная
- Устанавливаем и настраиваем AIDE - систему обнаружения вторжений для Linux

Устанавливаем и настраиваем AIDE - систему обнаружения вторжений для Linux

Безопасность - это вопрос волнующий любого администратора, так как в современном мире, фактически живущем в онлайн, скорость распространения угроз очень велика. Поэтому крайне важно вовремя заметить несанкционированную активность и принять соответствующие меры. В этом нам помогут системы обнаружения вторжений, а именно одна из самых популярных систем для Linux - AIDE (Advanced Intrusion Detection Environment). Несмотря на кажущуюся сложность вопроса, установить и использовать ее довольно просто и в данной статье мы расскажем, как это сделать.

В основе системы обнаружения вторжений лежит контроль состояния файловой системы, а в Linux, насколько мы помним, всё есть файл. AIDE считывает состояние файловой системы включая различные атрибуты, такие как права, владельцы, время создания и последнего доступа, размер и т.д. Также вычисляются контрольные суммы сразу по нескольким алгоритмам. Все это сохраняется в базу, с которой впоследствии происходит сравнение текущего состояния системы. Если какие-либо объекты были изменены, созданы или удалены - вы получите об этом отчет.

Таким образом достаточно просто можно обнаружить нежелательную активность, как внешнюю, так и внутреннюю. Последнее даже более важно, так как позволяет своевременно выявить ошибочные действия коллег, которые могут привести к существенному снижению безопасности, например, неверно выставленные права на директории. Очень часто такими вещами страдают различные веб-дизайнеры и прочие "смежники".

В нашей статье мы будем рассматривать установку и настройку системы обнаружения вторжений в среде Debian 11 и данные действия можно повторить в любой основанной на Debian системе. Для других дистрибутивов инструкция тоже подойдет с поправкой на работу пакетного менеджера и расположения конфигурационных файлов.

Все указанные ниже команды выполняются с правами суперпользователя **root** или через **sudo** если не указано иного.

Прежде всего установим программу:

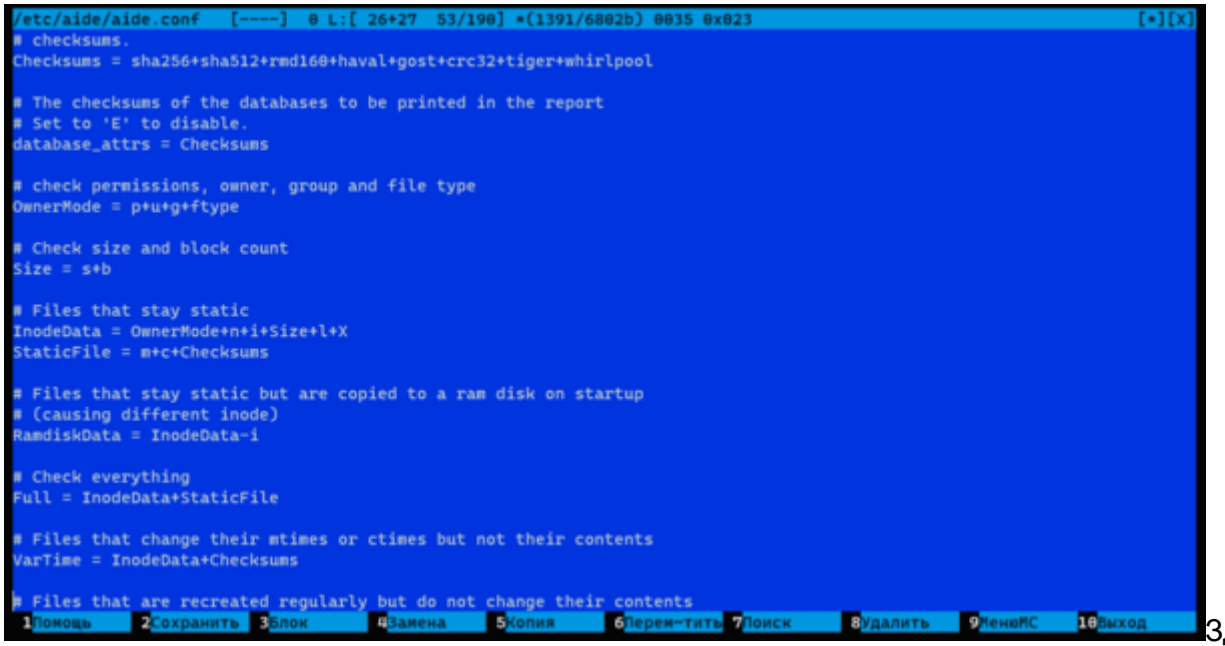
```
apt install aide
```

После чего немного осмотримся, все основные настройки хранятся в **/etc/aide**, здесь же расположен основной конфигурационный файл **/etc/aide/aide.conf**, оно достаточно короток и содержит только самые базовые настройки, например, расположение базы. Также в этом файле содержатся атрибуты, по которым производятся проверки. Атрибуты можно организовывать в группы, что позволяет получить высокую гибкость настроек и не загромождать конфигурацию, также группы можно вкладывать друг в друга.

Полный список атрибутов можно почерпнуть в документации:

Debian Manpages: AIDE.CONF
The AIDE manual

А мы пока пробежимся беглым взглядом:



Здесь как раз отлично видно, как формируются группы и как они затем вкладываются одна в другую. Скажем группа **Checksums** содержит перечисление всех используемых алгоритмов, а уже ниже она используется в составе другой группы. Но такой список алгоритмов вряд-ли нужен, а вычисления контрольных сумм достаточно ресурсоемкая операция, поэтому данный список можно подсократить по собственному усмотрению.

Также существует ряд групп по умолчанию, их, а также все поддерживаемые алгоритмы хеширования можно просмотреть командой:

```
aide --version
```

Теперь заглянем в **/etc/aide/aide.conf.d** и здесь мы увидим большое количество дополнительных конфигурационных файлов для наиболее популярных программ и ряда специфических расположений. Ведь любая работающая служба постоянно создает, изменяет, удаляет файлы и данные правила как раз предназначены исключить их сканирования нормальную активность приложений, в противном случае вы просто погрязнете в списке изменений и с большой долей вероятности пропустите действительно важные события.

Мы рекомендуем изучить конфигурационные файлы для используемых вами программ, большинство из них работает сразу, но некоторые требуют дополнительной настройки.

В этих файлах мы столкнемся с еще одним типом записи в конфигурации AIDE - правилами. Правила определяют, что именно мы проверяем и как именно. Существуют три вида правил: обычное, негативное и сравнивающее (Regular, Negative и Equals), все они работают с **регулярными выражениями**.

Начнем с обычных правил, они представляют собой конструкцию:

```
<regex> <attribute expression>
```

Каждое регулярное выражение должно начинаться с символа **/** (корень файловой системы) и представлять интересующее нас расположение. Еще раз обращаем внимание, что это не путь, а регулярка, поэтому если мы укажем:

```
/foo R
```

То это будет распространяться как на директорию **/foo**, так и **/foobar**, чтобы этого избежать явно указывайте символ окончания строки.

```
/foo$ R
```

После регулярного выражения указываем атрибуты или группу атрибутов, в нашем случае мы указали стандартную группу R. При необходимости мы можем ее прямо тут модифицировать, используя нужные атрибуты со знаками плюс или минус. Например, уберем md5-хеш и добавим вместо него gost:

```
/foo$ R-md5+gost
```

Негативное правило предназначено для исключения из проверки некоторых расположений, состоит только из регулярки, атрибуты не используются:

```
!<regex>
```

И, наконец, сравнивающее, оно отличается от обычного тем, что в базу добавляются только файлы и каталоги, соответствующие выражению, дочерние директории не добавляются.

```
=/foo R
```

Но если указать выражение с символом **/** на конце пути, то в базу будут также добавлены дочерние директории первого уровня:

```
=/foo/ R
```

Кроме того, к правилам можно применять ограничения по типам файлов, ограничив область их применения, например, только файлами или директориями. Возьмем реальный пример, мы хотим контролировать файлы конфигурации кластера 1С, но не учитывать изменения в каталогах с журналами и кешем. В этом случае исключаем каталоги и задаем правило только для файлов:

```
! /home/usr1cv8/.1cv8/1C/1cv8/reg_1541 d
/home/usr1cv8/.1cv8/1C/1cv8/reg_1541 f R
```

Во втором правиле ограничитель **f** может показаться излишним, но следует помнить, что в Linux кроме обычных файлов и директорий есть еще символические ссылки, сокеты, блочные и символьные устройства и т.д. и т.п., их обозначения смотрите в документации. Чтобы ограничиться только обычными файлами и директориями укажите:

```
=/foo d,f R
```

Также рекомендуем ознакомиться с уже существующими файлами конфигурации для используемых служб и логов, это поможет вам лучше понять принцип их составления и почерпнуть готовые приемы для использования в собственных правилах.

Собственные правила можно добавлять как в основной конфигурационный файл /etc/aide/aide.conf, так и создать отдельный файл в /etc/aide/aide.conf.d, последний вариант представляется нам предпочтительным. Но какой-бы из них вы не выбрали, помните, что конфигурационные файлы AIDE должны **завершаться пустой строкой**.

После того, как вы изучили и настроили правила можно выполнить первое сканирование и инициализировать базу данных. При этом мы явно указываем утилите какой конфигурационный файл использовать:

```
aide --config /etc/aide/aide.conf --init
```

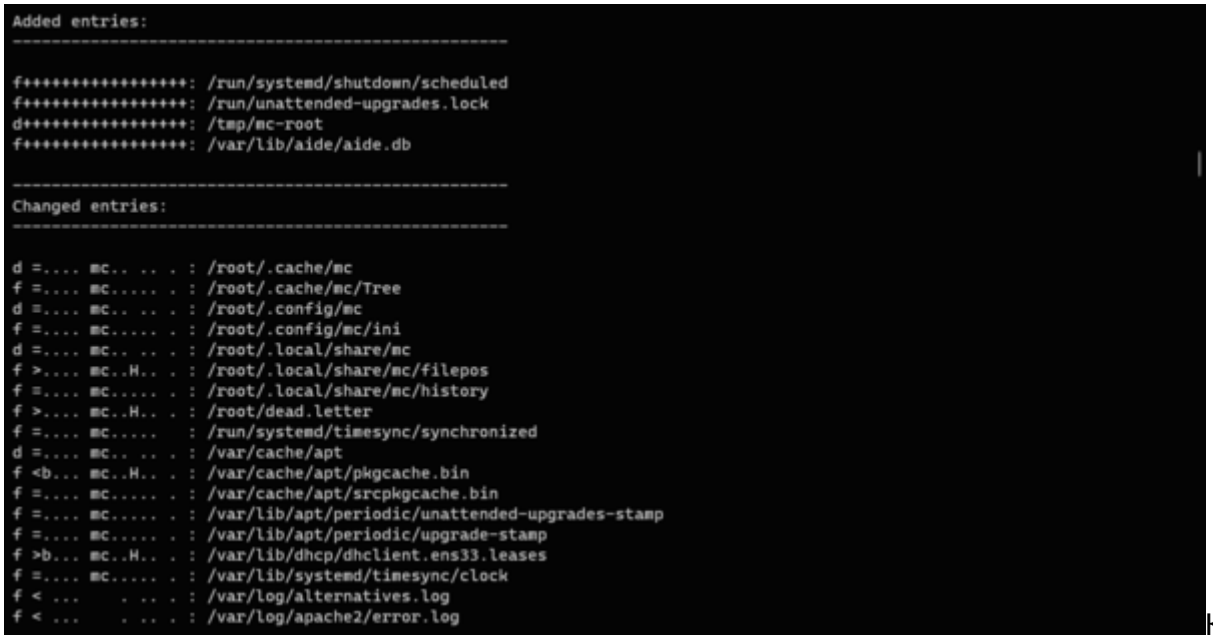
Сканирование займет некоторое время, после чего в директории **/var/lib/aide** появятся файл: **aide.db.new**. Это текущее состояние файловой системы, скопируем его в файл с названием **aide.db**, который станет эталонной базой данных, с которой мы будем производить сравнения.

```
cp /var/lib/aide/aide.db{.new,}
```

Теперь можно подождать некоторое время и запустить проверку:

```
aide --config /etc/aide/aide.conf --check
```

Будет выполнено повторное сканирование и состояние системы будет сравнено с эталонной базой данных, все расхождения будут показаны в отчете:

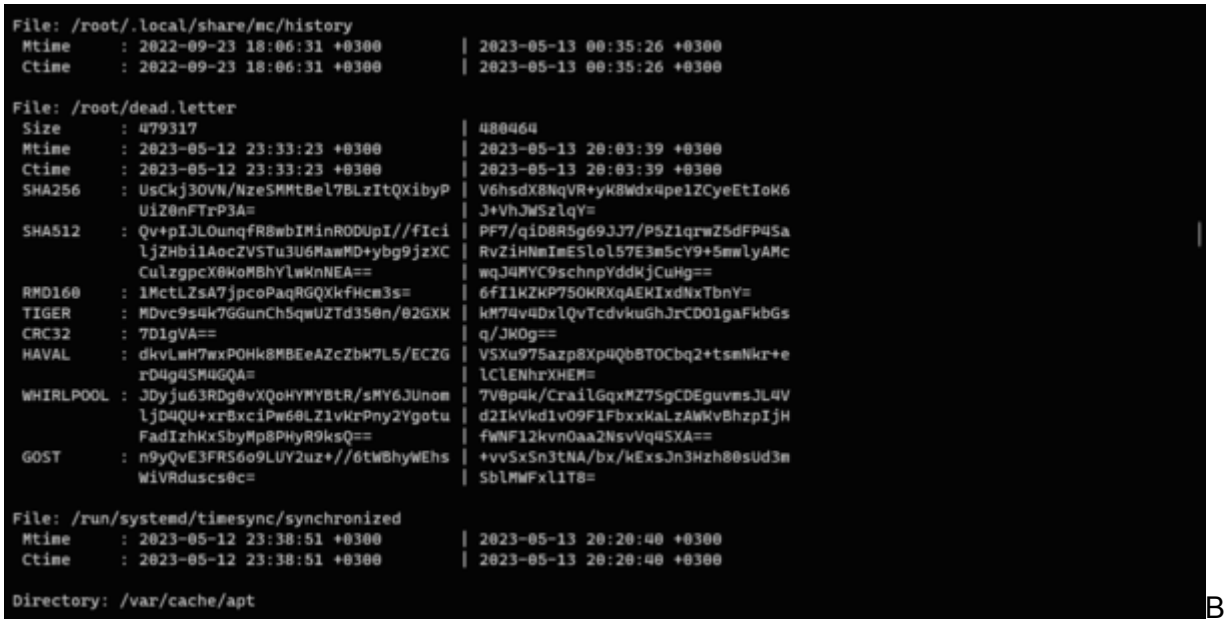


Как читать этот отчет? Очень просто. Самый первый символ в строке показывает тип файла: **f** - файл, **d** - директория. Первая секция - это добавленные элементы, тут все просто, тип файла и набор плюсов, показывающий что все остальные атрибуты были добавлены в базу. Ниже идет список изменений, знак равенства обозначает что файл не изменился, больше и меньше - увеличился или уменьшился. А буквы указывают список атрибутов, которые были изменены. Так **m** и **c** означают **mtime** и **ctime**, **H** - контрольные суммы, **b** - количество блоков. Если атрибут был добавлен - на его месте будет плюс, удален - минус, если игнорируется - двоеточие, не проверен - пробел, не изменился - точка. Сама же строка атрибутов выглядит следующим образом:

```
YlZbpugamcinHAXSEC
```

Где **Y** - это тип файла, а **Z** - знаки сравнения (=, <, >).

Если прокрутить вывод ниже, то вы получите подробный отчет по каждому файлу с приведением всех изменившихся реквизитов:



В целом информации вполне достаточно чтобы можно было как быстро просмотреть список изменений, так и выполнить детальное расследование. Со временем вы привыкните и будете считывать изменившиеся атрибуты налету, не заглядывая в документацию.

А что дальше? Отчет мы получили, изучили, никакого криминала в нем не нашли. Но если оставить все как есть, то эти файлы продолжат фигурировать во всех последующих отчетах, так как их состояние отличается от указанного в эталонной базе. Чтобы зафиксировать изменения эталонную базу следует обновить. Для этого выполним еще одно сканирование, но с ключом **--update**:

```
aide --config /etc/aide/aide.conf --update
```

Оно делает тоже самое, что и **--check**, но при этом записывает текущее состояние системы в базу данных **aide.db.new**. После чего достаточно переименовать эту базу в **aide.db** и использовать текущее состояние как новый эталон.

```
mv /var/lib/aide/aide.db.new  /var/lib/aide/aide.db
```

В дальнейшем вы можете выполнять сканирования сразу с ключом **--update**. Также следует обязательно обновлять эталонную базу после того, как вы внесли изменения в правила, особенно если добавили или исключили какие-либо расположения.

Автоматическое формирование отчетов с отсылкой на электронную почту

Ручные проверки - это хорошо, но неэффективно, поэтому все то, что можно автоматизировать должно быть автоматизировано, особенно в таком ответственном деле, как контроль безопасности. AIDE имеет штатные средства позволяющие легко настроить автоматическое ежедневное формирование отчетов с отправкой их на электронную почту.

В текущих реалиях поднимать собственный SMTP-сервер нет смысла, разве что он уже настроен на контролируемом узле, более удобно будет воспользоваться уже существующим почтовым сервером, собственным или публичным. Поэтому установим специальный почтовый клиент, который прозрачно заменит sendmail и позволит системе использовать внешние почтовые службы.

```
apt install ssmtp mailutils
```

Затем откроем **/etc/ssmtp/ssmtp.conf** и добавим учетную запись почты:

```
mailhub=mail.it-31.ru:587

 AuthUser=aide@it-31.ru
 AuthPass=Pa$$word_1
 UseTLS=YES
 UseSTARTTLS=YES
 TLS_CA_File=/etc/pki/tls/certs/ca-bundle.crt
```

Это типовой конфиг, который подходит для большинства собственных серверов и публичных почтовых служб. Каждая учетная запись начинается с опции **mailhub**, где указывается почтовый сервер и порт подключения клиента. Ниже указываем учетные данные и параметры шифрования, а также путь к собственному сертификату.

Затем найдите и приведите к следующему виду опцию:

```
FromLineOverride=YES
```

В некоторых случаях может потребоваться указать в поле **hostname** полное FQDN имя вашей системы, можно даже в несуществующем домене:

```
hostname=debian11.lab
```

Затем откроем **/etc/ssmtp/revaliases**, в который добавим строку:

```
root:aide@it-31.ru:mail.it-31.ru:587
```

Которая означает, что пользователь **root** должен использовать учетную запись **aide@it-31.ru** на сервере **mail.it-31.ru**.

Проверим отправку почты:

```
echo "Test" | ssmtp -v -s "Test" andrey@it-31.ru
```

Если все настроено правильно, то отправка будет успешной, нет - в логе будет показана ошибка.

Теперь примемся за настройку AIDE, откроем **/etc/default/aide** и раскомментируем следующую опцию:

```
CRON_DAILY_RUN=yes
```

Ниже укажем получателя почты:

```
MAILTO=andrey@it-31.ru
```

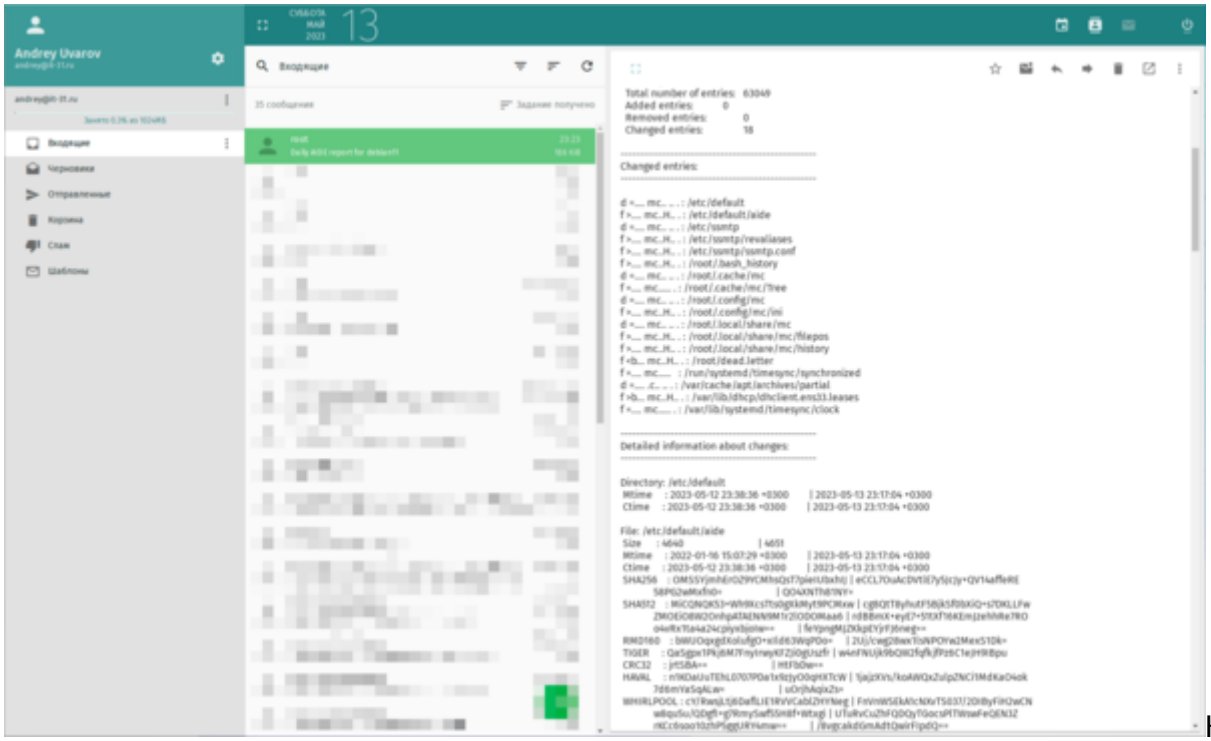
А также укажем, что после каждого сканирования следует обновлять эталонную базу данных:

```
COPYNEWDB=yes
```

Сохраним файл и попробуем проверить настройки. Скрипт, отвечающий за формирование отчетов, находится в **/etc/cron.daily**, что обеспечивает ему ежедневный запуск, для проверки работы достаточно вызвать его интерактивно:

```
/etc/cron.daily/aide
```

В случае успешной работы скрипт не выводит никаких сообщений, но на указанную почту должно прийти письмо с отчетом о сканировании.



Как видим, установить и настроить систему обнаружения вторжений несложно, но ее наличие способно значительно усилить безопасность системы и облегчить расследование возможных инцидентов, как внешних, так и внутренних.

Онлайн-курс по устройству компьютерных сетей
На углубленном курсе "Архитектура современных компьютерных сетей" вы с нуля научитесь работать с Wireshark и «под микроскопом» изучите работу сетевых протоколов. На протяжении курса надо будет выполнить более пятидесяти лабораторных работ в Wireshark.

Помогла статья? Поддержи автора и новые статьи будут выходить чаще:

Или подпишись на наш Телеграм-канал: