

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«КАЛИНИНГРАДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ»

В. В. Подтопельный

**СИСТЕМЫ ЗАЩИТЫ
ОТ УТЕЧКИ КОНФИДЕНЦИАЛЬНОЙ ИНФОРМАЦИИ**

Учебно-методическое пособие по выполнению практических работ
для студентов специальности 10.05.03
«Информационная безопасность автоматизированных систем»
специализация «Безопасность открытых информационных систем»

Калининград
Издательство ФГБОУ ВО «КГТУ»
2022

УДК 004.056(075)

Рецензент

Доцент кафедры информационной безопасности института информационных технологий ФГБОУ ВО «Калининградский государственный технический университет» А. Г. Жестовский

Подтопельный, В. В.

Системы защиты от утечки конфиденциальной информации: учебно-методическое пособие по практическим работам дисциплины для студентов специальности 10.05.03 «Информационная безопасность автоматизированных систем». – Калининград: Изд-во ФГБОУ ВО «КГТУ», 2022. – 61 с.

Учебно-методическое пособие включает в себя рассмотрение практических вопросов в области защиты информации по дисциплине «Системы защиты от утечки конфиденциальной информации». Пособие подготовлено в соответствии с требованиями утвержденной рабочей программы дисциплины.

Учебно-методическое пособие предназначено для студентов 5 курсов специальности 10.05.03 «Информационная безопасность автоматизированных систем» и смежных специальностей

Табл. 1, рис. 69, список лит. – 40 наименований

Учебно-методическое пособие рассмотрено и одобрено в качестве электронного методического материала кафедрой информационной безопасности 19 мая 2022 г., протокол № 7

Учебно-методическое пособие по изучению дисциплины рекомендовано к использованию в учебном процессе в качестве локального электронного методического материала методической комиссией института цифровых технологий ФГБОУ ВО «Калининградский государственный технический университет» 28 июня 2022 г., протокол № 4

УДК 004.056(075)

© Федеральное государственное
бюджетное образовательное
учреждение высшего образования
«Калининградский государственный
технический университет», 2022 г.
© Подтопельный В. В., 2022 г.

ОГЛАВЛЕНИЕ

Введение.....	4
Практическая работа №1. OSSEC HIDS	6
Практическая работа 2. Компоненты Wazuh.....	19
Заключение	58
Литература	59

ВВЕДЕНИЕ

Данное учебно-методическое пособие предназначено для студентов специальности 10.05.03 «Информационная безопасность автоматизированных систем», специализация: «Безопасность открытых информационных систем», изучающих дисциплину «Системы защиты от утечки конфиденциальной информации».

Цель изучения практического курса дисциплины обучить студентов выявлять и анализировать угрозы и уязвимости инфраструктур цифрового типа на предприятиях, определять особенности поведения злоумышленников в распределенных системах обработки информации.

Практикум содержит 2 работы.

Работы проводятся в лабораториях дисциплины «Информационная безопасность открытых информационных систем».

В результате выполнения лабораторных работ ожидается, что студенты сформируют навыки применения:

- направлений обеспечения защиты ресурсов вычислительных сетей и СУБД от атак вредоносных программ и злоумышленников;
- принципов функционирования современных систем аудита ресурсов ВС;
- построения систем безопасности в вычислительных сетях передачи данных;
- способов защиты трафика от изучения, разрушающих программных действий и изменений.

Программное обеспечение

1. Microsoft Desktop Education. Операционные системы: Microsoft Windows Desktop operating systems, офисные приложения (Microsoft Office), по соглашению V9002148 Open Value Subscription (срок действия: три года)

2. Программное обеспечение, распространяемое по лицензии GNU General Public License (лицензия на свободное программное обеспечение, созданная в рамках проекта GNU, по которой автор передаёт программное обеспечение в общественную собственность):

- Snort (свободная сетевая система предотвращения вторжений (IPS) и обнаружения вторжений (IDS) с открытым исходным кодом);
- Ethereal (программы перехвата и анализа сетевых пакетов), commview demo (или его аналог);
- NMAP (программа сканирование сетевых ресурсов);
- MySQL (система управления базами данных).

Типовое ПО на всех ПК:

1. Microsoft Desktop Education (операционные системы Microsoft Windows Desktop operating system, офисные приложения Microsoft Office, по соглашению V9002148 Open Value Subscription). Дата заключения контракта 05.07.2018. Номер контракта 0335100016118000073-0484577-02.

2. Антивирусное программное обеспечение Kaspersky Total Space Security Russian Edition, лицензия 17EO-171225-104659-470-270, срок использования с 2017-12-26 до 2020-03-13

Специализированное ПО:

1. VMWare Workstation, Страж-NT, Панцирь-К (по государственному контракту №10/13А от 19 апреля 2013 года), (на 2 компьютера – VMware License Purchase Information № 22033811OB);

Open Value Subscription.

Таблица. Шкала оценок уровня

Оценка			
Неудовлетворительный	Пороговый	Углубленный	Продвинутый
«2» (неудовлетворительно)	«3» (удовлетворительно)	«4» (хорошо)	«5» (отлично)
Работа выполнена в полном объеме. Отчет не оформлен и представлен. При защите отчетных материалов правильные ответы даны менее чем на 50 % включительно. Материал излагается непоследовательно, сбивчиво, не представляет определенной системы знаний по работе	Работа выполнена в полном объеме. Отчет оформлен и представлен. При защите отчетных материалов правильные ответы на 51–64 % вопросов. Допускаются нарушения в последовательности изложения. Демонстрируются поверхностные знания вопроса. Имеются затруднения с выводами. Допускаются нарушения норм литературной речи	Работа выполнена в полном объеме. Отчет оформлен и представлен. При защите отчетных материалов правильные ответы на 65–94 % вопросов. Ответы на поставленные вопросы излагаются систематизировано и последовательно. Материал излагается уверенно. Демонстрируется умение анализировать материал, однако не все выводы носят аргументированный и доказательный характер	Работа выполнена в полном объеме. Отчет оформлен и представлен. При защите отчетных материалов правильные ответы даны на 95–100 % вопросов. Ответы на поставленные в билете вопросы излагаются логично, последовательно и не требуют дополнительных пояснений. Делаются обоснованные выводы. Демонстрируются глубокие знания предмета

ПРАКТИЧЕСКАЯ РАБОТА № 1. OSSEC HIDS

Общие сведения

Цель: получения навыков обследования организации при проведении аудита информационной безопасности

Материалы, оборудование, программное обеспечение: ПК, ОС, OSSEC HIDS, текстовый редактор

1. Теоретическое введение

HIDS (хостовая система обнаружения вторжений) – это система для обнаружения вторжений, она наблюдает и анализирует события, которые происходят внутри системы (в отличие от сетевой СОВ, которая отслеживает в первую очередь сетевой трафик).

OSSEC – это хостовая система обнаружения вторжений (HIDS), свободная и с открытым исходным кодом. Она ведёт анализ системных логов, проверку целостности, наблюдение за реестром ОС Windows, обнаружение руткитов, оповещение в заданное время и если будет обнаружено какое-либо событие. Она предоставляет функцию обнаружения вторжений для большинства операционных систем, включая Linux, OpenBSD, FreeBSD, Mac OS X, Solaris и Windows.

«Система обнаружения вторжений (СОВ) (англ. Intrusion Detection System (IDS)) – программное (как в нашем случае) или аппаратное средство, предназначенное для выявления фактов неавторизованного доступа (вторжения или сетевой атаки) в компьютерную систему или сеть.

Цели, достигаемые при помощи IDS:

1. Самая главная – обнаружить вторжение или сетевую атаку.
 2. Прогнозирование возможного вторжения – злоумышленник обычно выполняет ряд действий при прощупывании вашей безопасности, это оставит след.
 3. Обеспечить централизованный контроль над безопасностью сети (особенно, большой и распределенной).
 4. Многие другие цели, которые вы можете придумать сами.
- Системы IDS делятся на network-based и host-based.

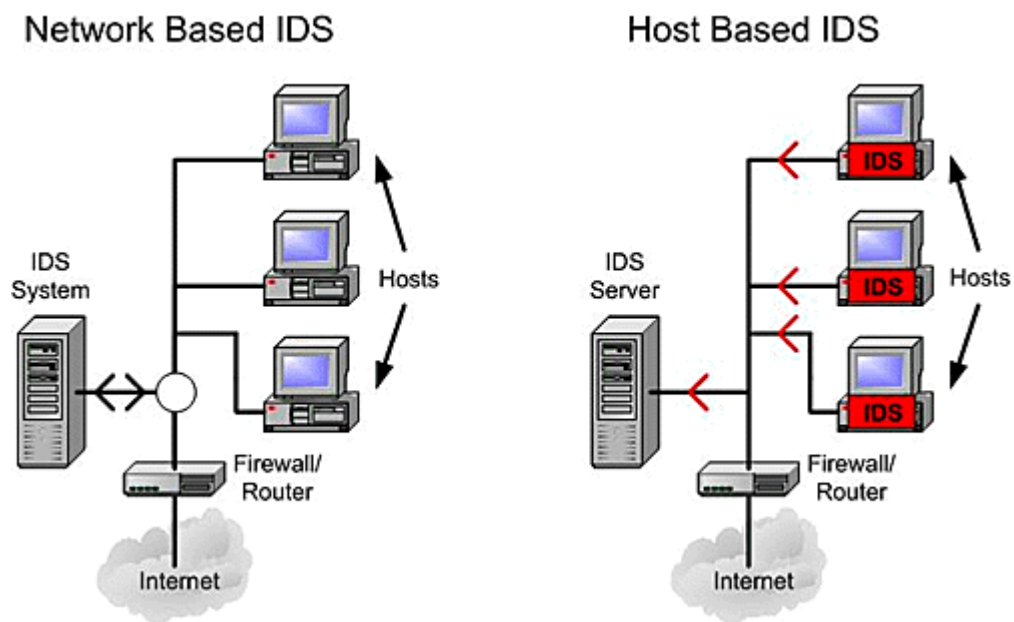


Рисунок 1.1. Устройство HIDS

NetworkBased IDS определяют атаки, захватывая и анализируя сетевые пакеты. Слушая сетевой сегмент, NIDS может просматривать сетевой трафик от нескольких хостов, которые присоединены к сетевому сегменту, и таким образом защищать эти хосты.

HostBased IDS имеют дело с информацией, собранной внутри единственного компьютера. Такое выгодное расположение позволяет HIDS анализировать деятельность с большой достоверностью и точностью, определяя только те процессы и пользователей, которые имеют отношение к конкретной атаке в ОС. Более того, в отличие от network-based IDS, host-based IDS могут «видеть» последствия предпринятой атаки, так как они могут иметь непосредственный доступ к системной информации, файлам данных и системным процессам, являющимся целью атаки» [6].

HIDS в основном используют информационные источники двух типов: результаты аудита ОС и системные логи. Результаты аудита ОС обычно создаются на уровне ядра ОС и, следовательно, являются более детальными и лучше защищенными, чем системные логи. Однако системные логи намного меньше и не столь многочисленны, как результаты аудита, и, следовательно, легче для понимания. Некоторые hostbased IDS разработаны для поддержки централизованной инфраструктуры управления и получения отчетов IDS, что может допускать единственную консоль управления для отслеживания многих хостов. Другие создают сообщения в формате, который совместим с системами сетевого управления.

OSSEC – это хостовая система обнаружения вторжений (HIDS), свободная и с открытым исходным кодом. Она ведёт анализ системных логов, проверку целостности, наблюдение за реестром ОС Windows, обнаружение руткитов, оповещение в заданное время и если будет обнаружено какое-либо событие. Она предоставляет функцию обнаружения вторжений для

большинства операционных систем, включая Linux, OpenBSD, FreeBSD, Mac OS X, Solaris и Windows.

OSSEC помогает клиентам соответствовать определенным требованиям, таким как PCI и HIPAA. Он позволяет клиентам обнаруживать и предупреждать о несанкционированных изменениях файловой системы и вредоносном поведении, встроенных в файлы журналов коммерческих продуктов, а также пользовательских приложений. Для PCI он охватывает разделы мониторинга целостности файлов (PCI 11.5, 10.5), проверки и мониторинга журналов (раздел 10), а также применения / проверки политик.

«Мультиплатформенность

OSSEC позволяет клиентам реализовать комплексную систему обнаружения вторжений на основе хоста с детализированными политиками для конкретных приложений / серверов на нескольких платформах, таких как Linux, Solaris, Windows и Mac OS X.

Настраиваемые оповещения в реальном времени

OSSEC позволяет клиентам настраивать инциденты, о которых они хотят получать уведомления, и позволяет им сосредоточиться на повышении приоритета критических инцидентов над обычным шумом в любой системе. Интеграция с smtp, sms и syslog позволяет клиентам быть в курсе предупреждений, отправляя их на устройства с поддержкой электронной почты. Также доступны варианты активного ответа для немедленного блокирования атаки.

Интеграция с существующей инфраструктурой

OSSEC будет интегрироваться с текущими инвестициями клиентов, такими как продукты SIM / SEM (Security Incident Management / Security Events Management) для централизованной отчетности и корреляции событий.

Централизованное управление

OSSEC предоставляет упрощенный централизованный сервер управления для управления политиками в нескольких операционных системах. Кроме того, он также позволяет клиентам определять специфичные для сервера переопределения для более тонких политик.

1) Агентный и безагентный мониторинг

2) OSSEC предлагает гибкость агентного и безагентного мониторинга систем и сетевых компонентов, таких как маршрутизаторы и межсетевые экраны. Безагентный мониторинг позволяет клиентам, у которых есть ограничения на установку программного обеспечения в системах (например, одобренные FDA системы или устройства), обеспечивать безопасность и соблюдение нормативных требований» [3].

3) Проверка целостности файлов

4) Все произведенные атаки на сеть и компьютеры объединяет одно: они изменяют наши системы. Необходимость проверки целостности файлов (или FIM – мониторинга целостности файлов) в том, чтобы обнаружить все изменения и предупредить нас, когда они произойдут. Это может быть атака, злоупотребление со стороны сотрудника или даже опечатка со стороны

администратора, вам будет сообщено о любом изменении файла, каталога или реестра.

5) Хакеры-преступники хотят скрыть свои действия, но, используя обнаружение руткитов, вы можете получать уведомления об изменении системы обычным для руткитов способом.

6) Активный ответ позволяет OSSEC предпринимать немедленные действия при срабатывании определенных предупреждений. Это может предотвратить распространение инцидента до того, как администратор сможет принять меры.

2. Задание к лабораторной работе:

1. Рассмотреть работу OSSEC HIDS

2. Исследовать эксплуатационные особенности OSSEC HIDS с применением Apache и Web-интерфейса.

Методические указания и порядок выполнения работы

Для реализации работы нам необходима программа VirtualBox, которая поможет нам создать виртуальную машину, одна будет с OS Ubuntu.

Установим Ubuntu на виртуальную машину.

1) Запустим VirtualBox;

2) Нажмем кнопку Создать;

3) Вводим название папки, где будет храниться созданная нами машина и выбираем тип системы, которую мы будем устанавливать;

4) Нажимаем далее и выделяем под ОС необходимый объем памяти, 20Гб будет достаточно, 128Мб оперативной памяти;

5) Затем нажимаем далее пока не выйдем из меню создания;

6) После всего этого запускаем созданную машину, указываем путь к образу Ubuntu и устанавливаем.

После установки ОС необходимо настроить сеть между виртуальной машиной и основной операционной системой. Выбираем из списка установленных ОС Ubuntu – Настройки – Сеть. В списке «Тип подключения» выбираем «Виртуальный адаптер хоста», а в «Имя» выбираем единственных из предложенных вариантов VirtualBox HostOnly Ethernet Adapter.

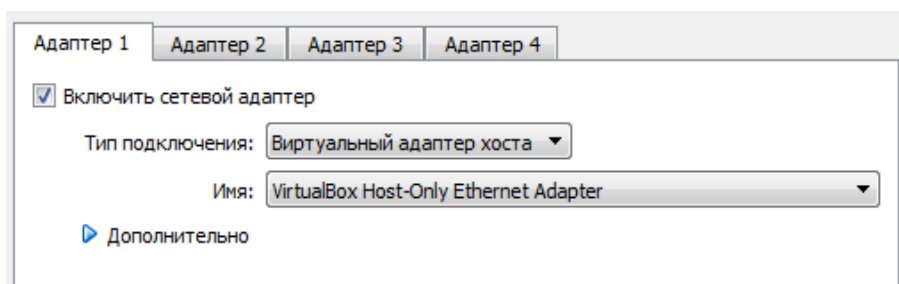


Рисунок 1.2. Настройка сети на VirtualBox

Сохраняем настройки и запускаем Ubuntu. Теперь необходимо провести настройку сети уже в самой ОС.

Запускаем через виртуальную машину ОС Ubuntu. Переходим в Сеть – Настройки – IPv4, в открывшемся окне указываем IP-адрес, маску сети, шлюз и IP-адреса DNS серверов для доступа к сети Интернет.

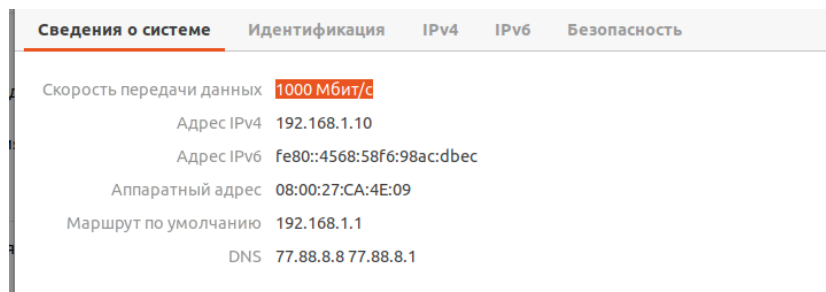


Рисунок 1.3. Настройка сети Ubuntu

Также необходимо настроить сеть основной ОС. Это обеспечит связь с OS Ubuntu.

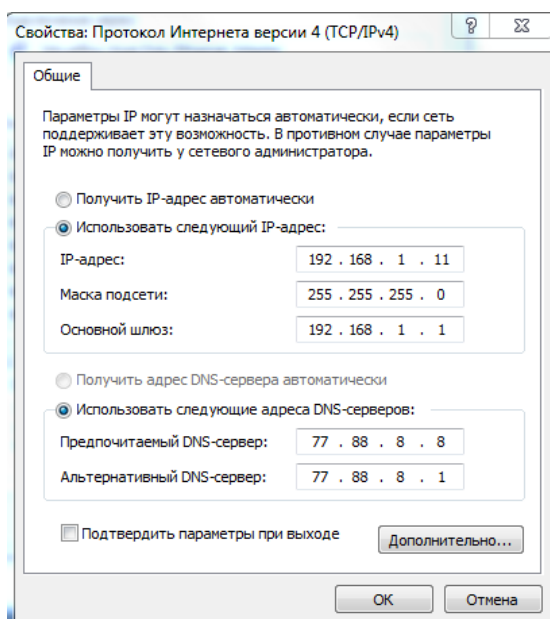


Рисунок 1.4. Настройка сети основной ОС

На этом установка лабораторной установки завершена, можно приступать к работе.

В работе мы будем использовать последнюю версию программы OSSEC, а так же Web-интерфейс для нее.

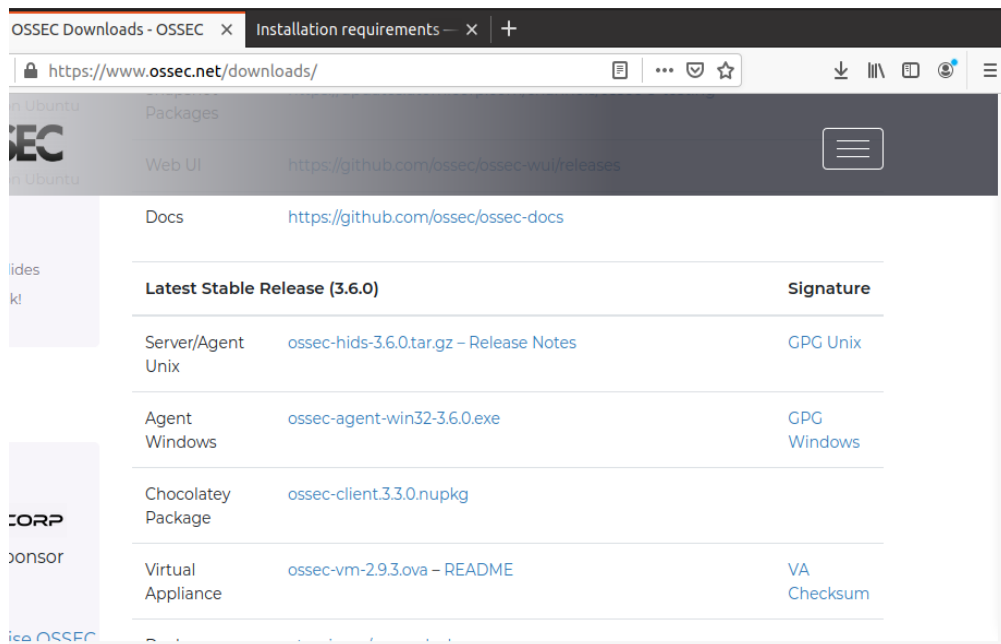


Рисунок 1.5. Официальный сайт OSSEC

Перед началом установки программы загрузите ее с официального сайта или иного репозитория, а также установите все необходимые дополнения.

1) Проведем обновление всех пакетов. Это необходимо для синхронизации файлов индекса пакетов из их источника. Будем использовать команду `sudo apt-get update`.

```
vlad@vlad-VirtualBox: ~  
vlad@vlad-VirtualBox:~$ sudo apt-get update  
[sudo] пароль для vlad:  
Пол:1 http://security.ubuntu.com/ubuntu focal-security InRelease [107 kB]  
Пол:2 http://ru.archive.ubuntu.com/ubuntu focal InRelease [265 kB]  
Пол:3 http://ru.archive.ubuntu.com/ubuntu focal-updates InRelease [111 kB]  
Пол:4 http://ru.archive.ubuntu.com/ubuntu focal-backports InRelease [98,3 kB]  
Пол:5 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [368 kB]  
Пол:6 http://ru.archive.ubuntu.com/ubuntu focal/main i386 Packages [718 kB]  
Пол:7 http://security.ubuntu.com/ubuntu focal-security/main i386 Packages [153 kB]  
Пол:8 http://security.ubuntu.com/ubuntu focal-security/main Translation-en [84,1 kB]  
Пол:9 http://ru.archive.ubuntu.com/ubuntu focal/main amd64 Packages [970 kB]  
Пол:10 http://security.ubuntu.com/ubuntu focal-security/main amd64 DEP-11 Metadata [24,3 kB]  
Пол:11 http://security.ubuntu.com/ubuntu focal-security/main DEP-11 48x48 Icons [11,0 kB]  
Пол:12 http://security.ubuntu.com/ubuntu focal-security/main DEP-11 64x64 Icons [16,5 kB]  
Пол:13 http://security.ubuntu.com/ubuntu focal-security/main amd64 c-n-f Metadata [5 396 B]  
Пол:14 http://security.ubuntu.com/ubuntu focal-security/restricted i386 Packages
```

Рисунок 1.6. Обновление пакетов

1) Теперь установим дополнительные пакеты для работы OSSEC.

```
vlad@vlad-VirtualBox: ~  
vlad@vlad-VirtualBox:~$ sudo apt-get install build-essential make zlib1g-dev lib  
pcrc2-dev libevent-dev libssl-dev  
Чтение списков пакетов... Готово  
Построение дерева зависимостей  
Чтение информации о состоянии... Готово  
Будут установлены следующие дополнительные пакеты:  
  binutils binutils-common binutils-x86-64-linux-gnu cpp-9 dpkg-dev fakeroot  
  g++ g++-9 gcc gcc-10-base gcc-9 gcc-9-base libalgorithm-diff-perl  
  libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan5 libatomic1  
  libbinutils libc-dev-bin libc6 libc6-dbg libc6-dev libcc1-0 libcrypt-dev  
  libctf-nobfd0 libctf0 libevent-core-2.1-7 libevent-extra-2.1-7  
  libevent-openssl-2.1-7 libevent-threads-2.1-7 libfakeroot libgcc-9-dev  
  libgcc-s1 libgomp1 libitm1 liblsan0 libpcrc2-16-0 libpcrc2-posix2  
  libquadmath0 libstdc++-9-dev libstdc++6 libtsan0 libubsan1 linux-libc-dev  
  manpages-dev zlib1g  
Предлагаемые пакеты:  
  binutils-doc gcc-9-locales debian-keyring g++-multilib g++-9-multilib  
  gcc-9-doc gcc-multilib autoconf automake libtool flex bison gcc-doc  
  gcc-9-multilib glibc-doc libssl-doc libstdc++-9-doc make-doc
```

Рисунок 1.7. Установка дополнительных пакетов

3) Приступаем к установке OSSEC. Для этого мы заранее распаковали архив с установочным файлом. Установку будем производить с помощью команды в терминале под правами root. Чтобы включить права root воспользуемся командой `sudo su`.

```
vlad@vlad-VirtualBox:~/Рабочий стол/ossec-hids-3.6.0$ sudo su  
[sudo] пароль для vlad:  
root@vlad-VirtualBox:/home/vlad/Рабочий стол/ossec-hids-3.6.0#
```

Рисунок 1.8. Права root

4) Запускаем установку программы командой «`./install.sh`»

```
root@vlad-VirtualBox:/home/vlad/Рабочий стол/ossec-hids-3.6.0# ./install.sh  
  
** Para instalação em português, escolha [br].  
** 要使用中文进行安装, 请选择 [cn].  
** Für eine deutsche Installation wählen Sie [de].  
** Για εγκατάσταση στα Ελληνικά, επιλέξτε [el].  
** For installation in English, choose [en].  
** Para instalar en Español , eliga [es].  
** Pour une installation en français, choisissez [fr]  
** A Magyar nyelvű telepítéshez válassza [hu].  
** Per l'installazione in Italiano, scegli [it].  
** 日本語でインストールします。選択して下さい。 [jp].  
** Voor installatie in het Nederlands, kies [nl].  
** Aby instalować w języku Polskim, wybierz [pl].  
** Для инструкций по установке на русском ,введите [ru].  
** Za instalaciju na srpskom, izaberi [sr].  
** Türkçe kurulum için seçin [tr].  
(en/br/cn/de/el/es/fr/hu/it/jp/nl/pl/ru/sr/tr) [en]:
```

Рисунок 1.9. Начало установки OSSEC

Соглашаемся со всеми вопросами и заканчиваем установку OSSEC.

Стандартно OSSEC устанавливается в каталог `var/ossec/`. Папки с бинарными файлами будут установлены – `var/ossec/bin/`. Папки с конфигурационными файлами – `var/ossec/etc/`. Папки с лог файлами – `var/ossec/log/`. В будущем, если будет необходима работа агентов с серверами будет нужен порт 1514 UDP.

5) Проведем проверку и настройку конфигурационного файла `ossec.conf`. Командой `nano /var/ossec/etc/ossec.conf` откроем нужный нам файл.

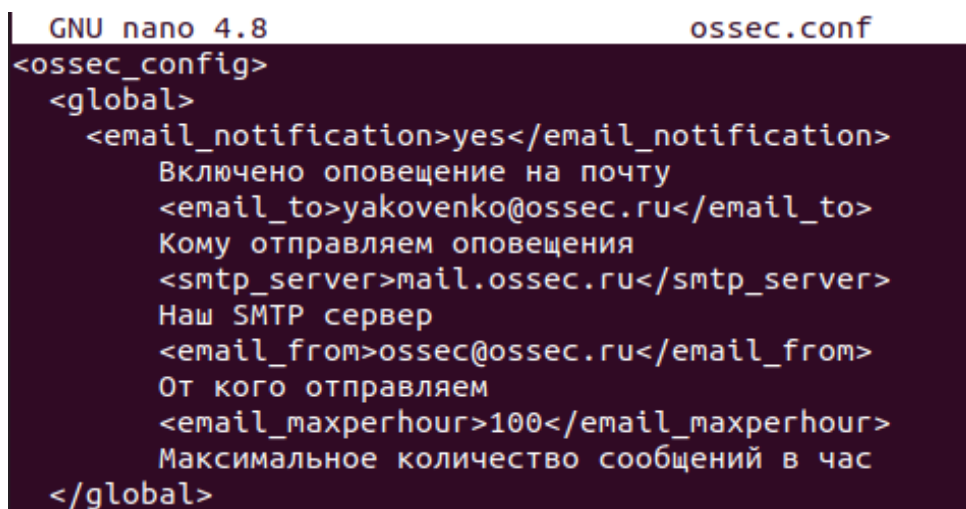


```
GNU nano 4.8 ossec.conf
<ossec_config>
<global>
  <email_notification>no</email_notification>
</global>

<rules>
  <include>rules_config.xml</include>
  <include>pam_rules.xml</include>
  <include>sshd_rules.xml</include>
  <include>telnetd_rules.xml</include>
  <include>syslog_rules.xml</include>
  <include>arpwatch_rules.xml</include>
  <include>symantec-av_rules.xml</include>
  <include>symantec-ws_rules.xml</include>
  <include>pix_rules.xml</include>
  <include>named_rules.xml</include>
  <include>smbd_rules.xml</include>
  <include>vsftpd_rules.xml</include>
  <include>pure-ftpd_rules.xml</include>
  <include>proftpd_rules.xml</include>
</rules>
[ Read 234 lines ]
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace  ^U Paste Text ^T To Spell  ^_ Go To Line
```

Рисунок 1.10 – Конфигурационный файл `ossec.conf`

Подключим email оповещения алертов нам на почту и максимальное количество сообщений в час.



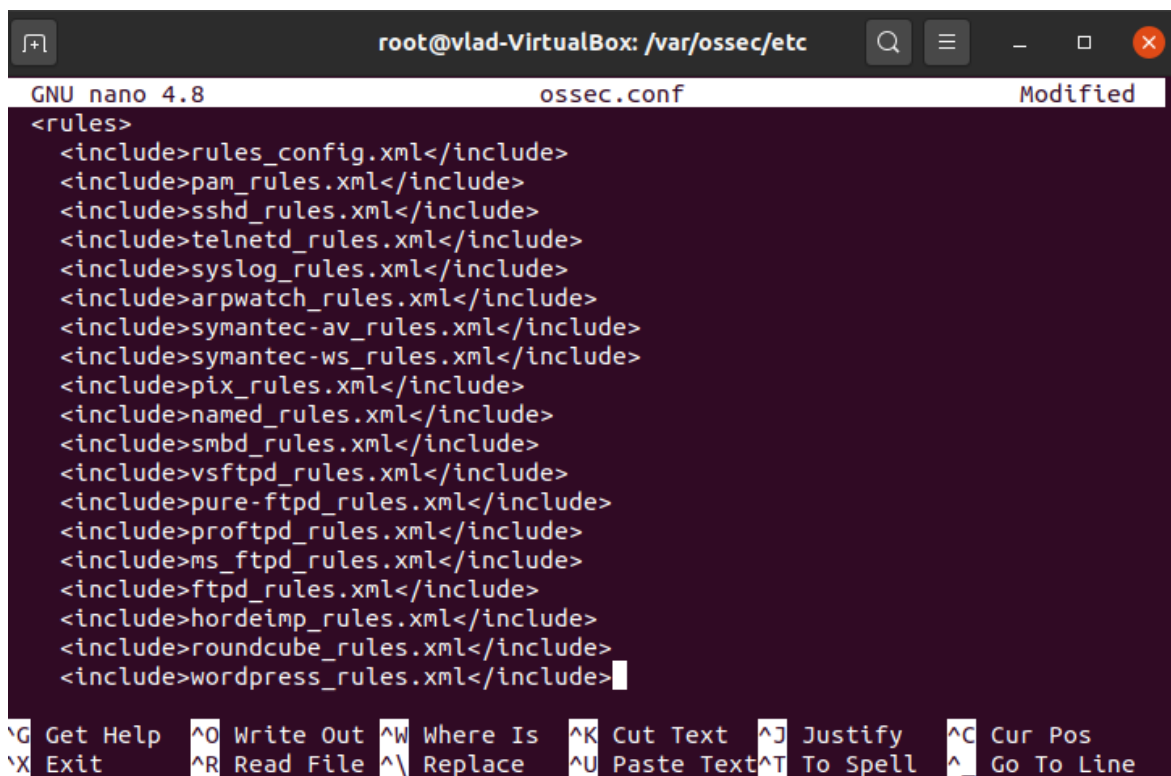
```
GNU nano 4.8 ossec.conf
<ossec_config>
<global>
  <email_notification>yes</email_notification>
  Включено оповещение на почту
  <email_to>yakovenko@ossec.ru</email_to>
  Кому отправляем оповещения
  <smtp_server>mail.ossec.ru</smtp_server>
  Наш SMTP сервер
  <email_from>ossec@ossec.ru</email_from>
  От кого отправляем
  <email_maxperhour>100</email_maxperhour>
  Максимальное количество сообщений в час
</global>
```

Рисунок 1.11. E-mail оповещения

Мы настроили email оповещения в секции global конфигурационного файла. Во всех событиях из правил есть уровень критичности, чтобы не получать оповещение каждый раз можно задать этот уровень в секции alerts.

Также можно настроить оповещения из каких-либо групп сообщений. Есть возможность получать смс оповещения.

В блоке rules находятся созданные нами правила, в которых описывается, на что и как OSSEC будет реагировать.



```
root@vlad-VirtualBox: /var/ossec/etc
GNU nano 4.8 ossec.conf Modified
<rules>
  <include>rules_config.xml</include>
  <include>pam_rules.xml</include>
  <include>sshd_rules.xml</include>
  <include>telnetd_rules.xml</include>
  <include>syslog_rules.xml</include>
  <include>arpwatch_rules.xml</include>
  <include>symantec-av_rules.xml</include>
  <include>symantec-ws_rules.xml</include>
  <include>pix_rules.xml</include>
  <include>named_rules.xml</include>
  <include>smbd_rules.xml</include>
  <include>vsftpd_rules.xml</include>
  <include>pure-ftp_rules.xml</include>
  <include>proftpd_rules.xml</include>
  <include>ms_ftp_rules.xml</include>
  <include>ftpd_rules.xml</include>
  <include>hordeimp_rules.xml</include>
  <include>roundcube_rules.xml</include>
  <include>wordpress_rules.xml</include>
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell ^_ Go To Line
```

Рисунок 1.12. Блок rules(правила)

Переходим в блок syscheck. Смысл ее заключается в том, что IDS подсчитывает хэш каждого файла в указанных директориях и периодически их сверяет. Тут мы задаем, какие директории мы будем контролировать и через сколько времени будет проводиться проверка.

```
GNU nano 4.8                               ossec.conf                               Modified

<syscheck>
<!-- Frequency that syscheck is executed - default to every 22 hours -->
<frequency>20000</frequency>

<!-- Directories to check (perform all possible verifications) -->
<directories check_all="yes">/etc,/usr/bin,/usr/sbin</directories>
<directories check_all="yes">/bin,/sbin,/boot</directories>

<!-- Files/directories to ignore -->
<ignore>/etc/mtab</ignore>
<ignore>/etc/mnttab</ignore>
<ignore>/etc/hosts.deny</ignore>
<ignore>/etc/mail/statistics</ignore>
<ignore>/etc/random-seed</ignore>
<ignore>/etc/adjtime</ignore>
<ignore>/etc/httpd/logs</ignore>
<ignore>/etc/utmpx</ignore>
<ignore>/etc/wtmpx</ignore>
<ignore>/etc/cups/certs</ignore>
```

Рисунок 1.13. Блок syscheck

Также, есть дополнительные параметры проверки целостности. Если будет необходимость проверки в определенное время, то можно использовать параметр `scan_time` или `scan_day`.

Если будет необходим постоянный контроль, каких-либо файлов, на этот случай есть параметр `realtime`. Постоянного контроля конкретных файлов использовать нельзя, необходимо указывать конкретную директорию, где хранится данный файл.

При необходимости можно включить оповещение о появлении новых файлов в папках, это параметр `alert_new_files`.

Также есть секции `rootcheck` и `localfile`. В `rootcheck` описывают файлы с характеристиками руткитов. В `localfile` хранятся лог файлы, которые `ossec` будет проверять.

б) После изменения всех настроек сохраняем и можем запустить OSSEC с помощью команды `service ./ossec-control start`.

```
root@vlad-VirtualBox:/var/ossec/bin# ./ossec-control start
Starting OSSEC HIDS v3.6.0...
Started ossec-mald...
Started ossec-execd...
Started ossec-analysisd...
Started ossec-logcollector...
Started ossec-remoted...
Started ossec-syscheckd...
Started ossec-monitord...
Completed.
root@vlad-VirtualBox:/var/ossec/bin#
```

Рисунок 1.14. Запуск OSSEC

Приступаем к установке и настройке Web-интерфейса.

1) В первую очередь устанавливаем сервер `apache`.

```
root@vlad-VirtualBox:/var/ossec/bin# apt-get install apache2 libapache2-mod-php
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapache2-mod-php7.4 libapr1
  libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.2-0 php-common
  php7.4-cli php7.4-common php7.4-json php7.4-opcache php7.4-readline
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom php-pear
```

Рисунок 1.15. Установка apache

2) Приступаем к установке Web-интерфейса.

```
root@vlad-VirtualBox:/home/vlad/Рабочий стол# mv ossec-wui-0.8 /var/www/ossec-wui
root@vlad-VirtualBox:/home/vlad/Рабочий стол# cd /var/www/ossec-wui/
root@vlad-VirtualBox:/var/www/ossec-wui# ls
CONTRIB      img          lib          README      site
css          index.php   LICENSE     README.search
htaccess_def.txt js          ossec_conf.php setup.sh
root@vlad-VirtualBox:/var/www/ossec-wui# ./setup.sh
```

Рисунок 1.16. Установка Web-интерфейса

3) Добавляем пользователя в группу apache

```
Username: vlad
New password:
Re-type new password:
Adding password for user vlad
Enter your web server user name (e.g. apache, www, nobody, www-data, ...)
apache
Enter your OSSEC install directory path (e.g. /var/ossec)

chgrp: invalid group: 'apache'
You must restart your web server after this setup is done.

Setup completed successfully.
root@vlad-VirtualBox:/var/www/ossec-wui# clear
```

Рисунок 1.17. Добавление пользователя в apache

2) Результат установки apache.

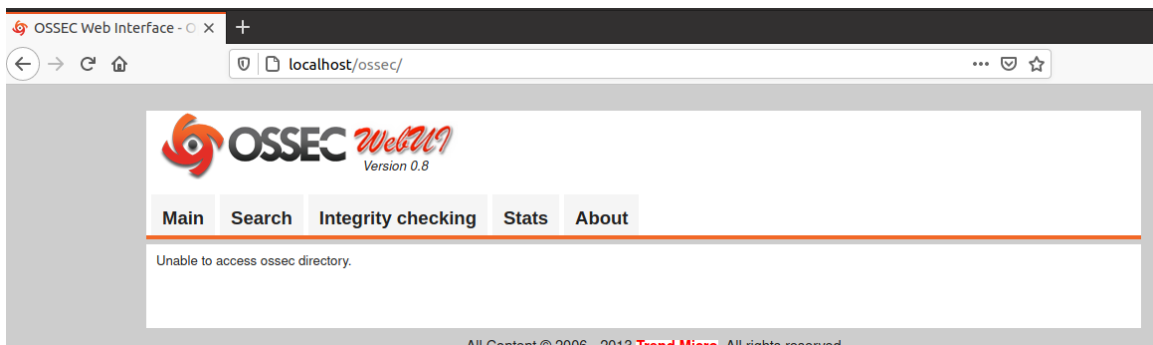


Рисунок 1.18. Web-интерфейс

На рисунке выше мы видим, что сайт уже работает, но события в него еще не записываются. Для этого необходимо выдать права папке tmp для записи в нее событий.

5) Выдаем папке tmp права

```

root@vlad-VirtualBox:/var/www/html/ossec# usermod -a -G ossec www-data
root@vlad-VirtualBox:/var/www/html/ossec# cat /etc/group |grep ossec
ossec:x:1001:ossec,ossecr,ossecm,ossece,apacheapache,www-data
root@vlad-VirtualBox:/var/www/html/ossec# chmod 770 tmp/
root@vlad-VirtualBox:/var/www/html/ossec# chgrp www-data tmp/
root@vlad-VirtualBox:/var/www/html/ossec#
  
```

Рисунок 1.19. Выдача прав папке tmp

Теперь мы можем видеть события, регистрируемые в OSSEC и пользоваться нехитрым интерфейсом.

Проверим работу сайта перезапустив OSSEC заново.

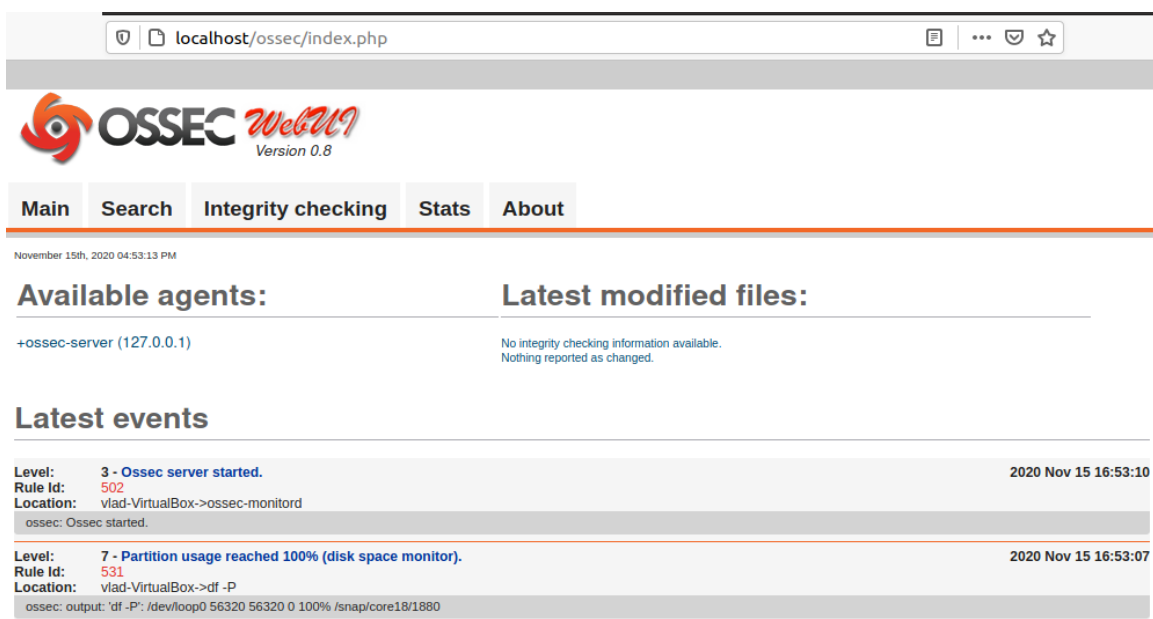


Рисунок 1.20. Результат работы сайта

На этом настройку можно считать оконченной. OSSEC очень гибкая и мощная IDS. В ней можно писать свои правила – их формат очень прост. Мы можем управлять агентами с сервера, назначать им конфигурацию, связать IDS с другой IDS (например, с NIDS Snort) и многое-многое другое.

Наша система работает в режиме SERVER (Сервер -> Агент). Помимо этого используемого нами существуют еще agent, local и hybrid.

Например, гибридный режим необходим, что бы использовать схему Агент -> Сервер -> Основной сервер. В данном случае на сервере работает как агент, так и сервер.

Существует два типа агентов: устанавливаемые агенты и агенты без агентов. Агенты устанавливаются на хостах, и они же отправляют отчеты на главный сервер через протокол сообщений. Агенты без агента не требуют установки на удаленных узлах. Это процессы, инициированные менеджером OSSEC, которые собирают информацию из удаленных систем и используют любой метод RPC.

Индивидуальное задание

Студент самостоятельно выбирает инфраструктура организации, которую он будет рассматривать при решении лабораторных задач.

Требования к отчету и защите

Защита предполагает опрос по материалу, изложенному в отчете.

Отчет должен быть оформлен: требуется наличие титульного листа, указание цели лабораторной работы, последовательное изложение разработанного материала, подписанные скриншоты хода работы, если использовалась компьютерная техника.

Проверяется знание теоретического материала с учетом знаний ответов на контрольные вопросы, приведенные в пункте «Контрольные вопросы».

Задаются вопросы по пунктам, приведенным в пункте «Методические указания и порядок выполнения работы».

Критерии к оценке защиты отчета студентом приведены в пункте «Общие сведения».

Контрольные вопросы:

1. Дайте определения безопасности информации, уязвимости информации, защищенности информации, угроз безопасности информации, защите информации. Перечислите источники угроз безопасности информации.

2. Охарактеризуйте основные принципы сбора информации об инфраструктуре изучаемого объекта.

3. Поясните суть обследование инфраструктуры ИС некоторой организации.

4. Разберите структурную схему и описание ИС.

ПРАКТИЧЕСКАЯ РАБОТА 2. КОМПОНЕНТЫ WAZUH

Общие сведения

Цель: получения навыков обследования организации при проведении аудита информационной безопасности

Материалы, оборудование, программное обеспечение: ПК, ОС, OSSEC NIDS, текстовый редактор

1. Теоретическое введение

1.1 Компоненты Wazuh

Платформа Wazuh предоставляет функции для защиты ваших облачных, контейнерных и серверных рабочих машин. К ним относятся анализ данных, обнаружение вторжений и вредоносных программ, мониторинг целостности файлов, оценка конфигурации, обнаружение уязвимостей и соответствие нормативным требованиям. Решение Wazuh основано на следующих трех компонентах:

Wazuh agent: установленный на конечных точках, таких как ноутбуки, настольные компьютеры, серверы, облачные экземпляры или виртуальные машины. Он поддерживает платформы Windows, Linux, MacOS, HP-UX, Solaris и AIX.

Wazuh server: анализирует данные, полученные от агентов, обрабатывает их с помощью декодеров и правил, использует аналитику угроз для поиска известных индикаторов компрометации (IOCs). Один сервер может анализировать данные от сотен или тысяч агентов и масштабироваться по горизонтали при настройке в виде кластера. Сервер также используется для управления агентами, их удаленной настройки и обновления при необходимости.

Elastic Stack: он индексирует и хранит оповещения, сгенерированные сервером Wazuh. Кроме того, интеграция между Wazuh и Kibana обеспечивает мощный пользовательский интерфейс для визуализации и анализа данных. Этот интерфейс также используется для управления конфигурацией Wazuh и отслеживания ее состояния.

В дополнение к возможностям мониторинга на основе агентов платформа Wazuh может отслеживать устройства без агентов, такие как брандмауэры, коммутаторы, маршрутизаторы или сетевые IDS.

На приведенном ниже рисунке представлены компоненты Wazuh и поток данных. На нем показаны три основных компонента решения: Wazuh agent, Wazuh server, Elastic Stack

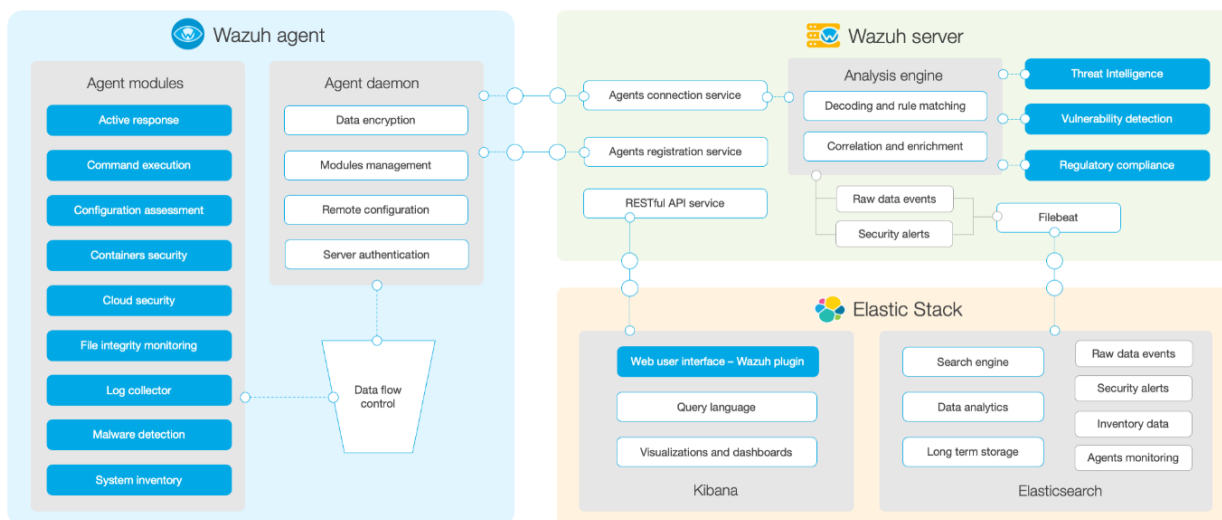


Рисунок 2.1. компоненты Wazuh

1.2 Wazuh agent

Wazuh agent работает в Linux, Windows, macOS, Solaris, AIX и других операционных системах. Его можно развернуть на ноутбуках, настольных компьютерах, серверах, облачных экземплярах, контейнерах или виртуальных машинах. Он обеспечивает защиту от угроз и обнаружения. Он также используется для сбора различных типов системных данных и данных приложений, которые он пересылает на Wazuh server по зашифрованному и аутентифицированному каналу.

1.2.1 Архитектура агента

Агент Wazuh имеет модульную архитектуру, в которой различные компоненты выполняют свои собственные задачи: мониторинг файловой системы, чтение сообщений журнала, сбор данных, сканирование конфигурации системы, поиск вредоносных программ и т. д. Пользователи могут включать или отключать модули агента с помощью изменения конфигурации, адаптируя решение к их конкретным вариантам использования.

На приведенной ниже диаграмме представлена архитектура и компоненты агента:

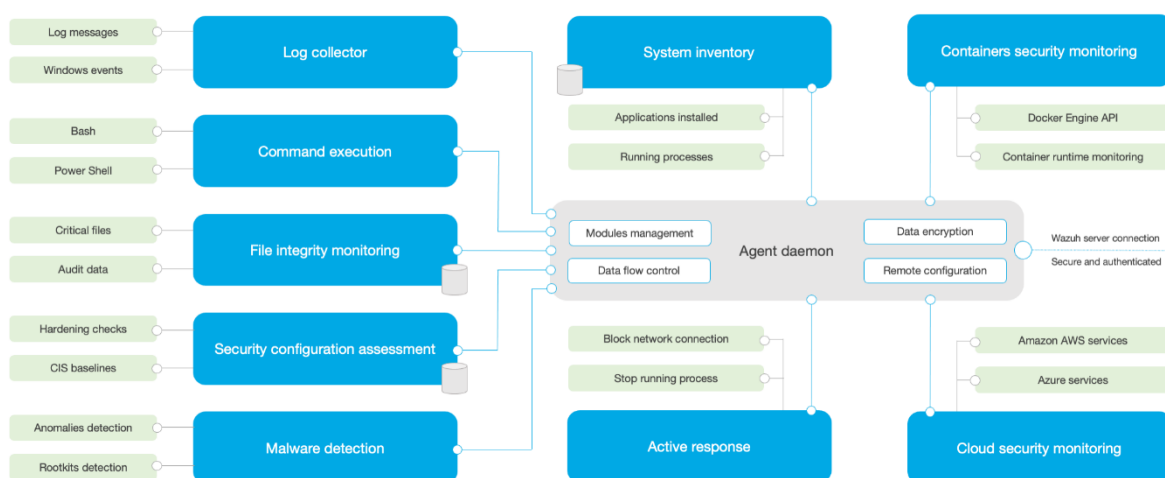


Рисунок 2.2. Архитектура Wazuh agent

1.2.2 Модули agent

Все модули агента имеют разное назначение и настройки. Вот краткое описание того, что они делают:

Логи: этот компонент агента может считывать неструктурированные файлы журналов и события Windows, собирая сообщения журналов операционной системы и приложений. Он поддерживает фильтры XPath для событий Windows и распознает многострочные форматы (например, журналы аудита Linux).

Выполнение команд: агенты могут периодически запускать авторизованные команды, собирая их выходные данные и отправляя их обратно на сервер Wazuh для дальнейшего анализа. Этот модуль можно использовать для различных целей (например, для контроля свободного места на жестком диске, получения списка последних зарегистрированных пользователей и т. д.).

File integrity monitoring (FIM): этот модуль отслеживает файловую систему, сообщая о создании, удалении или изменении файлов. Он отслеживает атрибуты файлов, разрешения, право собственности и содержимое. Когда происходит событие, в режиме реального времени фиксируются сведения о том, «кто, что и когда». Кроме того, этот модуль создает и поддерживает базу данных с состоянием отслеживаемых файлов, что позволяет удаленно выполнять запросы.

Security configuration assessment (SCA): Этот компонент обеспечивает непрерывную оценку конфигурации с использованием готовых наборов, основанных на эталонных тестах Center of Internet Security (CIS). Пользователи также могут создавать свои собственные проверки SCA для мониторинга и обеспечения соблюдения своих политик безопасности.

System inventory: этот модуль агента периодически выполняет сканирование, собирая данные, такие как версия операционной системы, сетевые интерфейсы, запущенные процессы, установленные приложения и список открытых портов. Результаты сканирования сохраняются в локальных базах данных SQLite, которые можно запрашивать удаленно.

Обнаружение вредоносных программ: Используя подход, не основанный на сигнатурах, этот компонент способен обнаруживать аномалии и возможное присутствие руткитов. Мониторинг системных вызовов, поиск скрытых процессов, скрытых файлов и скрытых портов.

Active response: этот модуль запускает автоматические действия при обнаружении угроз. Помимо прочего, он может заблокировать сетевое соединение, остановить запущенный процесс или удалить вредоносный файл. При необходимости пользователи также могут создавать собственные ответы (например, запускать двоичный файл в песочнице, перехватывать трафик сетевого подключения, сканировать файл антивирусом и т. д.).

Мониторинг безопасности контейнеров: этот модуль агента интегрирован с API Docker Engine для отслеживания изменений в контейнерной среде. Например, он обнаруживает изменения в образах контейнеров, конфигурации сети или томах данных. Кроме того, он оповещает

о контейнерах, работающих в привилегированном режиме, и о пользователях, выполняющих команды в работающем контейнере.

Security configuration assessment: этот компонент отслеживает облачных провайдеров, таких как Amazon AWS, Microsoft Azure или Google GCP. Он изначально взаимодействует с их API. Он способен обнаруживать изменения в облачной инфраструктуре (например, создание нового пользователя, изменение группы безопасности, останов облачного экземпляра и т. д.) и собирать данные журналов облачных сервисов (например, AWS Cloudtrail, AWS Macie, AWS GuardDuty). , Azure Active Directory и т. д.)

1.2.3 Связь с сервером Wazuh

Wazuh agent связывается с Wazuh server для отправки собранных данных и событий, связанных с безопасностью. Кроме того, агент отправляет данные, сообщая о своей конфигурации и состоянии. После подключения агент можно обновлять, отслеживать и настраивать удаленно с сервера Wazuh.

Связь агента Wazuh с сервером происходит по защищенному каналу (TCP или UDP), обеспечивающему шифрование и сжатие данных в режиме реального времени. Кроме того, он включает в себя механизмы управления потоком, чтобы избежать флуда, постановки событий в очередь, когда это необходимо, и защиты пропускной способности сети.

Регистрация агента Wazuh необходима перед его первым подключением к серверу. Этот процесс предоставляет агенту уникальный ключ, который используется для аутентификации и шифрования данных.

1.3 Wazuh server

Серверный компонент Wazuh отвечает за анализ данных, полученных от агентов, и запускает оповещения при обнаружении угроз или аномалий. Он также используется для удаленного управления конфигурацией агентов и отслеживания их состояния.

Сервер Wazuh использует источники информации об угрозах, чтобы улучшить свои возможности обнаружения. Он также использует нормативные требования соответствия (например, PCI DSS, HIPAA, NIST 800-53...) и платформу Mitre ATT&CK для пополнения данных предупреждений.

1.3.1 Серверная архитектура

На сервере Wazuh работает механизм анализа, Wazuh RESTful API, служба регистрации агентов, служба подключения агентов, кластера Wazuh и Filebeat.

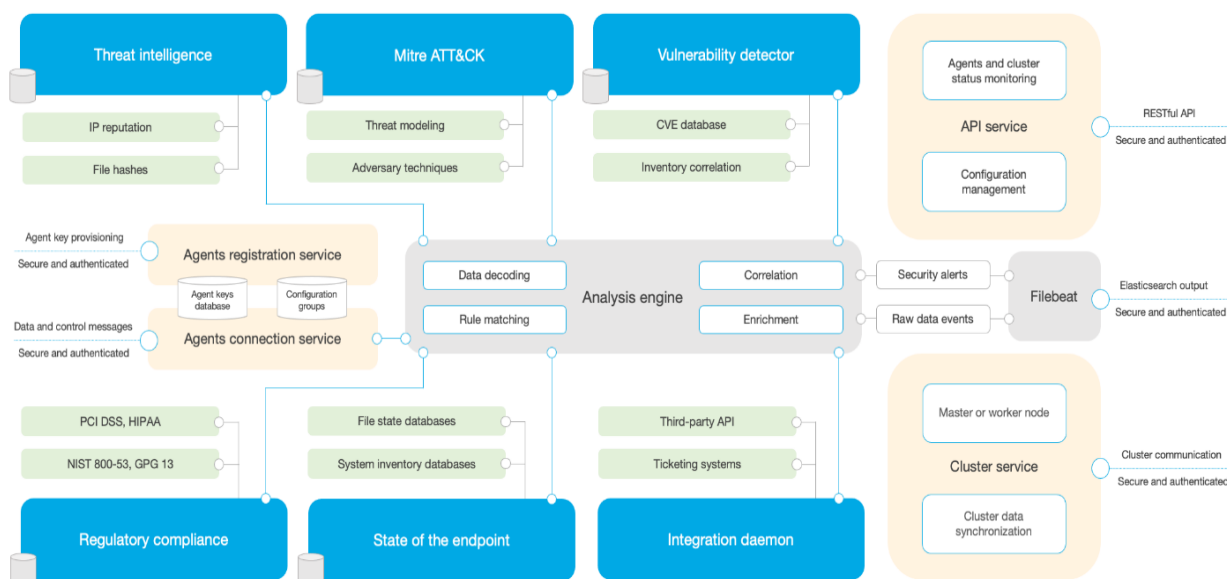


Рисунок 2.3. Архитектура и компоненты сервера

Сервер обычно работает на автономной физической машине, виртуальной машине, док-контейнере или облачном экземпляре. Он устанавливается на операционную систему Linux. Ниже приведен список основных компонентов сервера:

Служба регистрации агентов: используется для регистрации новых агентов путем подготовки и распространения ключей аутентификации, уникальных для каждого агента. Этот процесс работает как сетевая служба и поддерживает аутентификацию с помощью сертификатов TLS/SSL или путем предоставления фиксированного пароля.

Служба подключения агентов: это служба, которая получает данные от агентов. Он использует ключи для проверки личности каждого агента и шифрования связи между агентом и сервером Wazuh. Кроме того, эта служба используется для обеспечения централизованного управления конфигурацией, позволяя удаленно передавать новые настройки агента.

Механизм анализа: это процесс, выполняющий анализ данных. Он использует декодеры для определения типа обрабатываемой информации (например, событий Windows, журналов SSHD, журналов веб-сервера и т. д.) и извлечения соответствующих элементов данных из сообщений журнала (например, IP-адрес источника, идентификатор события, имя пользователя и т. д.). Затем, используя правила, он идентифицирует определенные шаблоны в декодированных событиях, которые могут вызвать оповещения и, возможно, даже вызвать автоматические контрмеры (например, блокировку IP-адреса на брандмауэре).

Wazuh RESTful API: этот сервис предоставляет интерфейс для взаимодействия с инфраструктурой Wazuh. Он используется для управления настройками конфигурации агентов и серверов, для мониторинга состояния и общего состояния инфраструктуры, для управления и редактирования декодеров и правил Wazuh, а также для запроса состояния отслеживаемых

конечных точек. Он также используется пользовательским веб-интерфейсом Wazuh, который представляет собой приложение Kibana.

Кластер Wazuh: эта служба используется для горизонтального масштабирования серверов Wazuh, развертывая их как кластер. Такая конфигурация в сочетании с балансировщиком сетевой нагрузки обеспечивает высокую доступность и балансировку нагрузки. Кластер Wazuh – это то, что серверы Wazuh используют для связи друг с другом и для синхронизации.

Filebeat: используется для отправки событий и предупреждений в Elasticsearch. Он считывает выходные данные механизма анализа Wazuh и отправляет события в режиме реального времени. Он также обеспечивает сбалансированную нагрузку при подключении к многоузловому кластеру Elasticsearch.

1.4 Elastic Stack

Elastic Stack – это унифицированный набор популярных проектов с открытым исходным кодом для управления журналами, включая Elasticsearch, Kibana, Filebeat и другие. Проекты, которые особенно важны для решения Wazuh:

Filebeat: облегченный сервер логов, используемый для передачи журналов по сети, обычно в Elasticsearch. Он используется на сервере Wazuh для отправки событий и предупреждений в Elasticsearch.

Elasticsearch: хорошо масштабируемая система полнотекстового поиска и аналитики. Elasticsearch является распределенным, то есть индексы данных разделены на сегменты. Wazuh использует разные индексы для данных предупреждений, необработанных событий и информации мониторинга состояния.

Kibana: гибкий и интуитивно понятный веб-интерфейс для добычи, анализа и визуализации данных. Он работает поверх проиндексированного контента в кластере Elasticsearch. Пользовательский веб-интерфейс Wazuh был полностью встроен в Kibana в виде плагина. Он включает готовые информационные панели для событий безопасности, соответствия нормативным требованиям (например, PCI DSS, GDPR, CIS, HIPAA, NIST 800-53), обнаруженных уязвимых приложений, данных мониторинга целостности файлов, результатов оценки конфигурации, событий мониторинга облачной инфраструктуры, и другие.

Wazuh интегрируется с Elastic Stack, предоставляя поток уже декодированных сообщений для индексации Elasticsearch, а также веб-консоль в реальном времени для анализа предупреждений и данных журнала. Кроме того, пользовательский интерфейс Wazuh, работающий поверх Kibana, используется для управления и мониторинга инфраструктуры Wazuh.

Индекс Elasticsearch – это набор документов, которые имеют несколько схожие характеристики (например, некоторые общие поля и общие требования к хранению данных). Wazuh использует целых три различных индекса, создаваемых ежедневно, для хранения различных типов событий:

wazuh-alerts: предупреждения, созданные сервером Wazuh. Они создаются каждый раз, когда событие вызывает срабатывание правила с достаточно высоким приоритетом (это пороговое значение настраивается).

wazuh-events: все события (архивных данных), полученных от агентов, вне зависимости от того, срабатывают ли правила

wazuh-monitoring: данные, связанных со статусом агентов Wazuh с течением времени. Он используется веб-интерфейсом для представления того, когда агенты были активны, неактивны или вообще не подключались.

1.4 Архитектура

Архитектура Wazuh основана на агентах, работающих на контролируемых конечных точках, которые передают данные безопасности на центральный сервер

Кластер Elasticsearch – это набор из одного или нескольких узлов, которые взаимодействуют друг с другом для выполнения операций чтения и записи. Небольшие развертывания Wazuh, не требующие обработки больших объемов данных, могут легко обрабатываться кластером с одним узлом. Кластеры с несколькими узлами рекомендуются при наличии большого количества отслеживаемых конечных точек, когда ожидается большой объем данных или когда требуется высокая доступность.

Для производственных сред рекомендуется развернуть сервер Wazuh и Elasticsearch на разных хостах. В этом сценарии Filebeat используется для безопасной пересылки предупреждений Wazuh и/или архивных событий в кластер Elasticsearch (с одним или несколькими узлами) с использованием шифрования TLS.

На приведенной ниже диаграмме представлена архитектура развертывания Wazuh. В нем показаны компоненты решения и то, как серверы Wazuh и Elasticsearch могут быть сконфигурированы как кластер, обеспечивающий сбалансированную нагрузку и высокую скорость отклика.

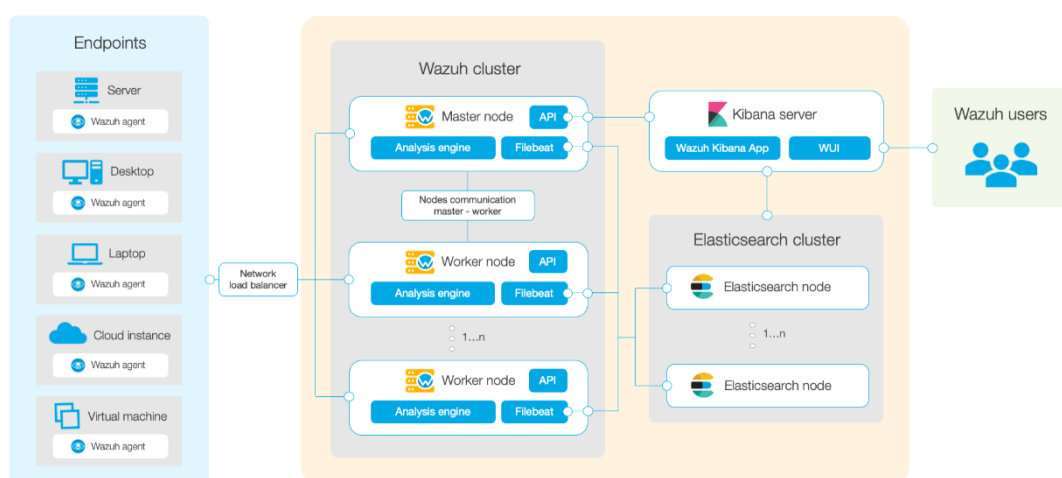


Рисунок 2.4. Архитектура Wazuh

1.4.1 Агент Wazuh – связь с сервером Wazuh

Агент Wazuh постоянно отправляет события на сервер Wazuh для анализа и обнаружения угроз. Чтобы начать их доставку, агент устанавливает соединение со службой сервера для соединения агентов, которая прослушивает порт 1514 (это настраивается). Затем сервер Wazuh декодирует и проверяет полученные события с помощью механизма анализа. События, приводящие к срабатыванию правила, дополняются данными предупреждений, такими как идентификатор и имя правила. События можно буферизовать в один или оба следующих файла, в зависимости от того, сработало ли правило:

Файл `/var/ossec/logs/archives/archives.json` содержит все события независимо от того, сработало ли правило или нет.

Файл `/var/ossec/logs/alerts/alerts.json` содержит только события, которые привели к срабатыванию правила с достаточно высоким приоритетом (порог настраивается).

Протокол сообщений Wazuh по умолчанию использует шифрование AES со 128 битами на блок и 256-битными ключами.

1.4.2 Сервер Wazuh – связь с Elastic Stack

Сервер Wazuh использует Filebeat для отправки данных предупреждений и событий на сервер Elasticsearch с использованием шифрования TLS. Filebeat считывает выходные данные сервера Wazuh и отправляет их в Elasticsearch (по умолчанию прослушивается порт 9200/TCP). Как только данные обрабатываются Elasticsearch, Kibana используется для извлечения и визуализации информации.

Пользовательский веб-интерфейс Wazuh работает внутри Kibana в виде плагина. Он запрашивает API-интерфейс Wazuh RESTful (по умолчанию прослушивается порт 55000/TCP на сервере Wazuh), чтобы отобразить информацию о конфигурации и состоянии сервера и агентов Wazuh. Он также может изменять с помощью вызовов API агенты или параметры конфигурации сервера, если это необходимо. Эта связь зашифрована с помощью TLS и аутентифицирована с помощью имени пользователя и пароля.

1.5 Wazuh server administration

Менеджер Wazuh – это система, которая анализирует данные, полученные от всех зарегистрированных агентов, и запускает оповещения, когда событие совпадает с правилом, например: обнаружено вторжение, изменен файл, конфигурация не соответствует политике, возможный руткит и другие. Менеджер также работает как агент на локальной машине, а это означает, что он имеет все функции, которые есть у агента. Кроме того, менеджер может пересылать обработанные им оповещения через системный журнал, электронную почту или интегрированные внешние API.

1.5.1 Определение порогового уровня оповещения

Каждое событие, собранное агентом Wazuh, передается менеджеру Wazuh. Менеджер назначит событию уровень серьезности в зависимости от

того, каким правилам из набора правил оно соответствует. По умолчанию он регистрирует только оповещения с уровнем серьезности 3 или выше.

Конфигурация

Пороговое значение уровня оповещения настраивается в `ossec.conf` файле с помощью `<alerts>` тега XML.

```
<ossec_config>
  <alerts>
    <log_alert_level>6</log_alert_level>
  </alerts>
</ossec_config>
```

Это установит минимальный уровень тревоги, при котором будут инициироваться оповещения, которые будут храниться в файле `alerts.log` и /или `alerts.json` файле(ах).

При изменении любого значения в `ossec.conf` файле необходимо перезапустить службу, прежде чем изменения вступят в силу.

Создание автоматических отчетов

Ежедневные отчеты представляют собой сводку предупреждений, которые срабатывали каждый день. Вы можете настроить свой собственный настраиваемый отчет, используя `report` параметр в `ossec.conf` файле.

Настройка оповещений по электронной почте и SMTP-сервера с проверкой подлинности .

```
<ossec_config>
  <reports>
    <category>syscheck</category>
    <title>Daily report: File changes</title>
    <email_to>example@test.com</email_to>
  </reports>
</ossec_config>
```

Приведенная выше конфигурация будет отправлять ежедневный отчет обо всех предупреждениях системной проверки `example@test.com`

Правила также можно фильтровать по уровню, источнику, имени пользователя, идентификатору правила и т. д.

Общие параметры электронной почты

Чтобы настроить Wazuh для отправки оповещений по электронной почте, необходимо настроить параметры электронной почты в `<global>` разделе `ossec.conf` файла:

```
<ossec_config>
  <global>
    <email_notification>yes</email_notification>
    <email_to>me@test.com</email_to>
    <smtp_server>mail.test.com</smtp_server>
```

```
<email_from>wazuh@test.com</email_from>
</global>
...
</ossec_config>
```

1.6 Регистрация агента Wazuh

Регистрация агента – это процесс регистрации агентов Wazuh в качестве авторизованных участников решения Wazuh

Способы регистрации

Существует два варианта регистрации агентов в менеджере Wazuh.

1. Регистрация через конфигурацию агента: после установки IP-адреса менеджера агент сможет автоматически запросить ключ и импортировать его. Это рекомендуемый метод регистрации.

2. Регистрация через API менеджера: пользователь запрашивает ключ из API менеджера, а затем вручную импортирует его в агент.

1.7 Сбор данных

Сбор данных журналов – это процесс в режиме реального времени осмысления записей, созданных серверами или устройствами. Этот компонент может получать журналы через текстовые файлы или журналы событий Windows. Он также может напрямую получать журналы через удаленный системный журнал, что полезно для брандмауэров и других подобных устройств.

Целью этого процесса является выявление ошибок приложений или системы, неправильных конфигураций, попыток вторжения, нарушений политик или проблем с безопасностью.

Требования агента Wazuh к памяти и ЦП незначительны, поскольку его основная обязанность – пересылать события менеджеру. Однако в диспетчере Wazuh потребление ЦП и памяти может быстро увеличиваться в зависимости от количества событий в секунду (EPS), которые должен анализировать диспетчер.

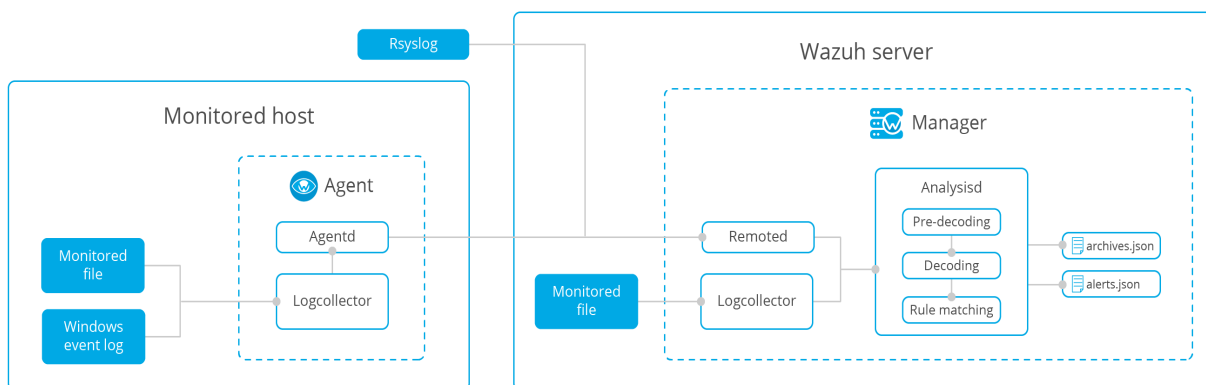


Рисунок 2.5. Сбор данных

Механизм анализа журналов можно настроить для мониторинга определенных файлов на серверах.

Linux:

```
<localfile>  
  <location>/var/log/example.log</location>  
  <log_format>syslog</log_format>  
</localfile>
```

Windows:

```
<localfile>  
  <location>C:\myapp\example.log</location>  
  <log_format>syslog</log_format>  
</localfile>
```

1.8 Мониторинг целостности файлов

Система мониторинга целостности файлов (FIM) Wazuh отслеживает выбранные файлы и выдает предупреждения при изменении этих файлов. Компонент, отвечающий за эту задачу, называется `syscheck`. Этот компонент хранит криптографическую контрольную сумму и другие атрибуты файлов или ключей реестра Windows и регулярно сравнивает их с текущими файлами, используемыми системой, отслеживая изменения.

Начиная с версии 3.12.0, Wazuh использует новый модуль FIM.

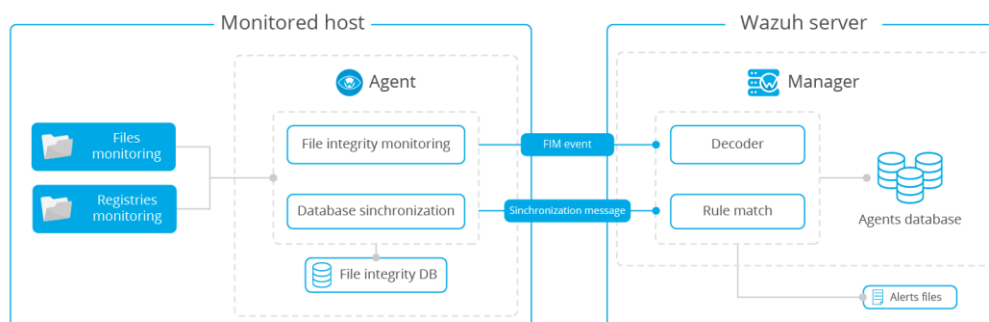


Рисунок 2.6. Мониторинг целостности файлов

Модуль FIM находится в агенте Wazuh, где выполняет периодическое сканирование системы и сохраняет контрольные суммы и атрибуты отслеживаемых файлов и ключей реестра Windows в локальной базе данных FIM. Модуль ищет модификации, сравнивая контрольные суммы новых файлов со старыми контрольными суммами. Обо всех обнаруженных изменениях сообщается менеджеру Wazuh.

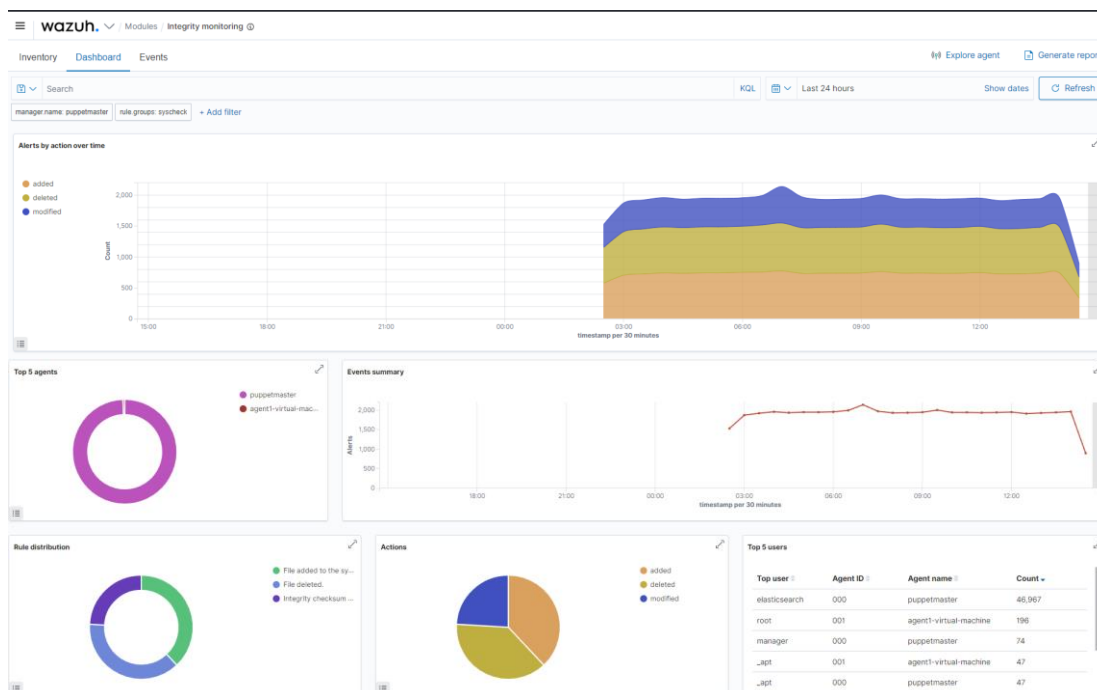


Рисунок 2.7. Integrity monitor

Чтобы сообщать о новых файлах, добавленных в систему, syscheck можно настроить с помощью параметра `alert_new_files`. По умолчанию эта функция включена в отслеживаемом агенте Wazuh, но эта опция отсутствует в разделе конфигурации syscheck:

```
<syscheck>
  <alert_new_files>yes</alert_new_files>
</syscheck>
```

1.9 Обнаружение аномалий и вредоносных программ

Обнаружение аномалий относится к действиям по поиску в системе шаблонов, которые не соответствуют ожидаемому поведению. Как только вредоносное ПО (например, руткит) установлено в системе, оно модифицирует систему, чтобы скрыть себя от пользователя. Хотя вредоносное ПО использует для этого множество методов, Wazuh использует широкий спектр подходов к поиску аномальных шаблонов, указывающих на возможных злоумышленников.

Основным компонентом, отвечающим за эту задачу, является **rootcheck**, однако важную роль играет и **Syscheck**.

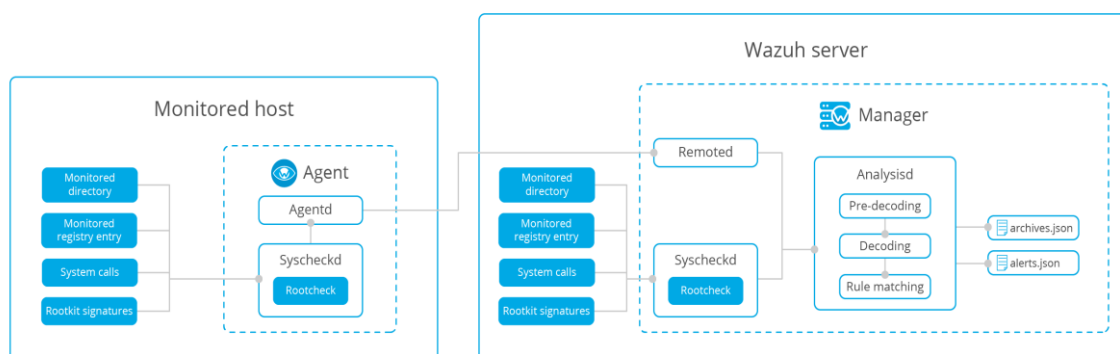


Рисунок 2.8. Обнаружение аномалий

Мониторинг целостности файлов

Вредоносное ПО может заменять файлы, каталоги и команды на хост-системе. Выполнение проверки целостности файлов в основных каталогах системы позволяет обнаружить эти действия.

Проверка запущенных процессов

Вредоносный процесс может скрыть себя в списке процессов системы. **Rootcheck** проверяет все идентификаторы процессов (PID) на наличие расхождений с различными системными вызовами (`getsid`, `getpgid`).

Проверка скрытых портов

Вредоносное ПО может использовать скрытые порты для связи с злоумышленником. **Rootcheck** проверяет каждый порт в системе с помощью `bind()`. Если он не может привязаться к порту и этот порт отсутствует в выходных данных `netstat`, возможно, присутствует вредоносное ПО.

Проверка на необычные файлы и разрешения

Wazuh сканирует всю файловую систему в поисках необычных файлов и разрешений. Проверяются все файлы, принадлежащие пользователю `root` с разрешениями на запись для других учетных записей пользователей, такие как файлы `suid`, скрытые каталоги и файлы.

Сканирование каталога `/dev`

Каталог `/dev` должен содержать только файлы, относящиеся к конкретному устройству. Любые дополнительные файлы должны быть проверены, поскольку вредоносное ПО использует этот раздел для сокрытия файлов. Если вы создаете скрытый файл на `/dev`, Wazuh должен предупредить вас, потому что в каталоге есть скрытый файл, который должен содержать только файлы, относящиеся к конкретному устройству.

Сканирование сетевых интерфейсов

Wazuh сканирует любые сетевые интерфейсы в системе с включенным неразборчивым режимом. Если интерфейс находится в неразборчивом режиме, вывод команды `ifconfig` укажет на это. Это может быть индикатором наличия вредоносного ПО.

Проверка руткитов

Rootcheck выполняет несколько проверок, используя собственную базу данных сигнатур руткитов: `rootkit_files.txt`, `rootkit_trojans.txt` и `win_malware_rcl.txt`.

Чтобы настроить параметры `syscheck` и `rootcheck`, перейдите в `ossec.conf`.

```
<rootcheck>
  <rootkit_files>etc/shared/rootkit_files.txt</rootkit_files>

<rootkit_trojans>etc/shared/rootkit_trojans.txt</rootkit_trojans>
</rootcheck>
```

1.10 SCA

Существует несколько интеграций Wazuh, которые выполняют сканирование для оценки конфигурации, включая CIS-CAT и, совсем недавно, оценку конфигурации безопасности (SCA).

У каждого агента есть собственная локальная база данных, в которой хранится текущее состояние каждой проверки: пройдено, не удалось или ошибка, что позволяет агентам отправлять только различия, обнаруженные между сканированиями. Если изменений не было, `summary` будет отправлено только событие сканирования, что позволяет избежать ненужного сетевого трафика и поддерживать диспетчер в актуальном состоянии. Затем менеджер будет использовать эти обновления для создания предупреждений, которые будут отображаться в приложении Kibana.

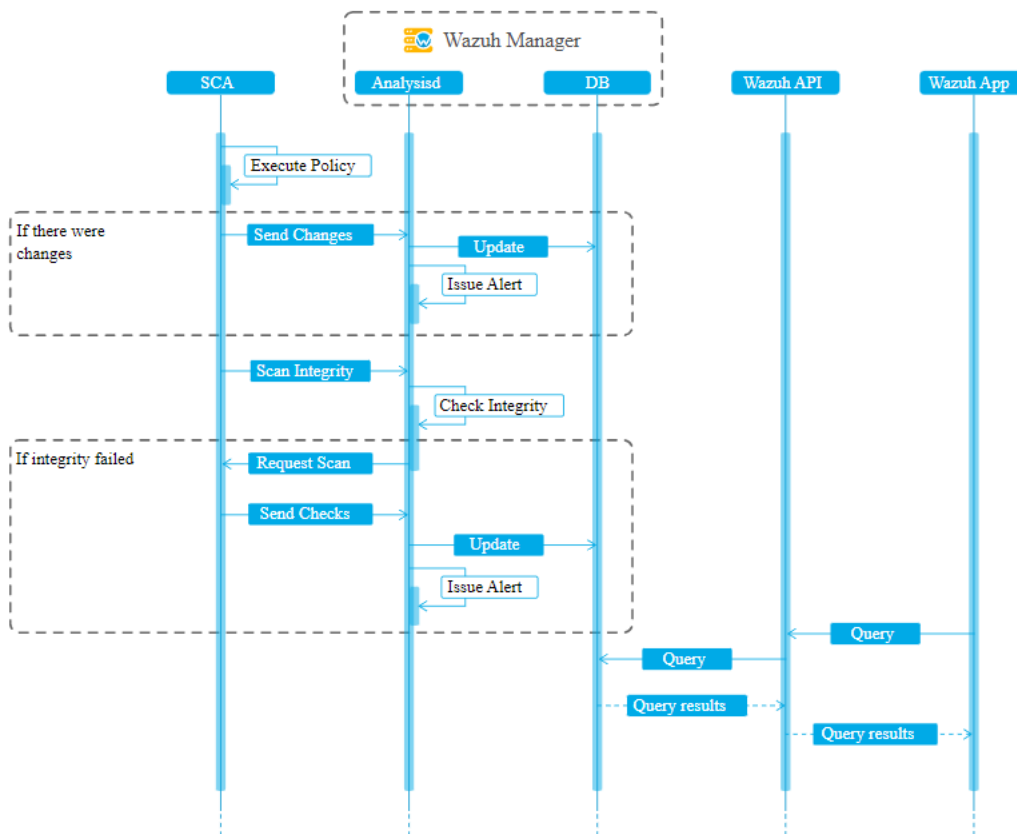


Рисунок 2.9. Работа SCA

1.11 Мониторинг команд

Бывают случаи, когда вы можете захотеть отслеживать вещи, которых нет в журналах. Чтобы решить эту проблему, Wazuh включает возможность отслеживать вывод определенных команд и обрабатывать вывод так, как если бы он был содержимым файла журнала.

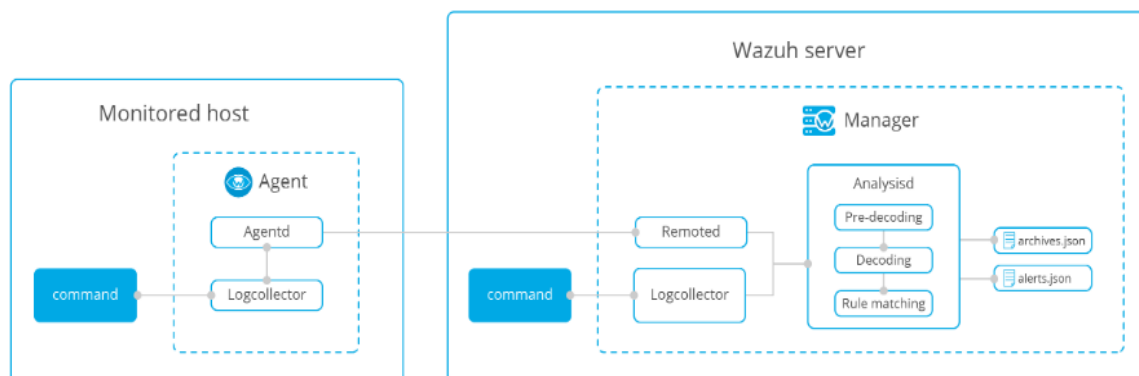


Рисунок 2.10. Мониторинг команд

Команды для запуска и мониторинга можно настроить в локальном файле **ossec.conf** отдельных агентов, однако идеальное место для этой конфигурации. Пример:

```
<localfile>
  <log_format>full_command</log_format>
  <command>.....</command>
  <frequency>120</frequency>
</localfile>
```

1.12 Обнаружение уязвимостей

Wazuh умеет выявлять уязвимости в приложениях, установленных в агентах, с помощью модуля «Vulnerability detection». Этот аудит программного обеспечения выполняется путем интеграции каналов уязвимостей, проиндексированных Canonical, Debian, Red Hat и глобальной базой данных уязвимостей.

Чтобы иметь возможность обнаруживать уязвимости, теперь агенты могут нативно собирать список установленных приложений, периодически отправляя его менеджеру (где он хранится в локальных базах данных sqlite, по одной на агента). Кроме того, менеджер создает глобальную базу данных уязвимостей из общедоступных репозиториях CVE, используя ее позже для кросс-корреляции этой информации с данными инвентаризации приложений агента.

Глобальная база данных уязвимостей может быть настроена на периодическое обновление, что гарантирует, что решение будет проверять наличие самых последних CVE.

После создания глобальной базы данных уязвимостей (с CVE) процесс обнаружения ищет уязвимые пакеты в базах данных инвентаризации (уникальных для каждого агента). Оповещения генерируются, когда CVE (распространенные уязвимости и риски) влияет на пакет, который, как известно, установлен на одном из отслеживаемых серверов. Пакет помечается как уязвимый, если его версия находится в зоне действия CVE. Результаты представлены в виде предупреждений, а также сохранены в базе данных. Таким образом, вы можете проверить последние оповещения о сканировании или запросить базу данных уязвимого программного обеспечения каждого отдельного агента.

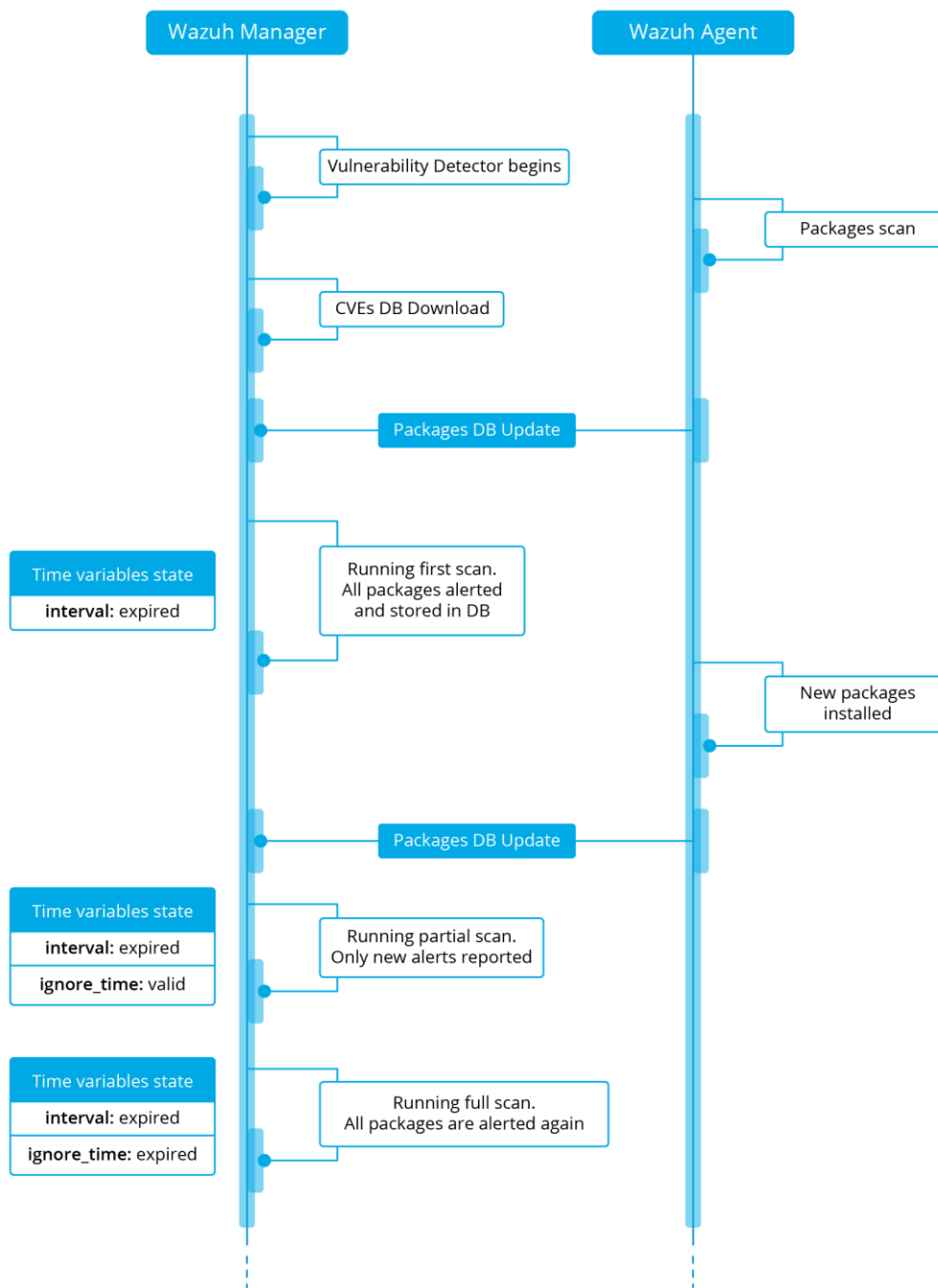


Рисунок 2.11. Поиск уязвимостей

1.13 Wazuh Kibana

Плагин Wazuh Kibana позволяет пользователям просматривать и анализировать оповещения Wazuh, хранящиеся в Elasticsearch. Пользователи могут получать статистику по каждому агенту, искать оповещения и фильтровать их с помощью различных визуализаций. Он интегрируется с Wazuh API для получения информации о конфигурации менеджера и агентов, журналах, наборе правил, группах и многом другом.

Плагин Wazuh Kibana дает быстрый обзор вашего кластера, агентов и предупреждений. Он предоставляет элегантный и простой в использовании

пользовательский интерфейс для взаимодействия с Wazuh API и менеджером Wazuh, отображая соответствующую информацию более удобным способом.

2.Задание к лабораторной работе:

1. Рассмотреть работу OSSEC HIDS
2. Исследовать эксплуатационные особенности OSSEC HIDS с применением Apache и Web-интерфейса.

2

2.1 Установка Wazuh-manager

1. Добавить GPG-ключ

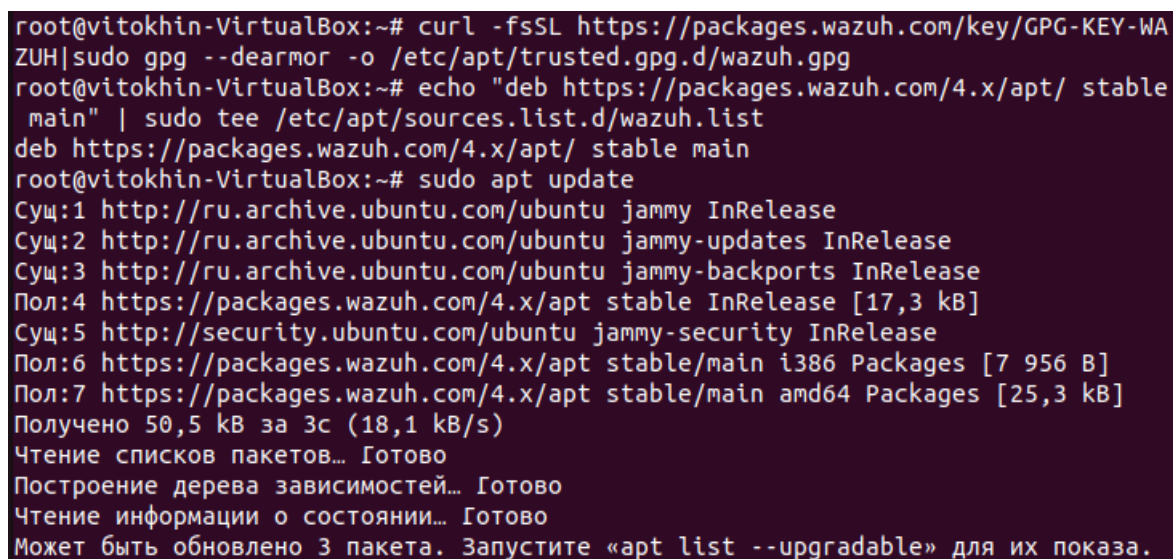
```
curl -fsSL https://packages.wazuh.com/key/GPG-KEY-WAZUH|sudo gpg --dearmor -o /etc/apt/trusted.gpg.d/wazuh.gpg
```

2. Добавьте репозиторий Wazuh

```
echo "deb https://packages.wazuh.com/4.x/apt/ stable main" | sudo tee /etc/apt/sources.list.d/wazuh.list
```

3. Обновите систему

```
sudo apt update
```



```
root@vitokhin-VirtualBox:~# curl -fsSL https://packages.wazuh.com/key/GPG-KEY-WAZUH|sudo gpg --dearmor -o /etc/apt/trusted.gpg.d/wazuh.gpg
root@vitokhin-VirtualBox:~# echo "deb https://packages.wazuh.com/4.x/apt/ stable main" | sudo tee /etc/apt/sources.list.d/wazuh.list
deb https://packages.wazuh.com/4.x/apt/ stable main
root@vitokhin-VirtualBox:~# sudo apt update
Суц:1 http://ru.archive.ubuntu.com/ubuntu jammy InRelease
Суц:2 http://ru.archive.ubuntu.com/ubuntu jammy-updates InRelease
Суц:3 http://ru.archive.ubuntu.com/ubuntu jammy-backports InRelease
Пол:4 https://packages.wazuh.com/4.x/apt stable InRelease [17,3 kB]
Суц:5 http://security.ubuntu.com/ubuntu jammy-security InRelease
Пол:6 https://packages.wazuh.com/4.x/apt stable/main i386 Packages [7 956 B]
Пол:7 https://packages.wazuh.com/4.x/apt stable/main amd64 Packages [25,3 kB]
Получено 50,5 kB за 3с (18,1 kB/s)
Чтение списков пакетов... Готово
Построение дерева зависимостей... Готово
Чтение информации о состоянии... Готово
Может быть обновлено 3 пакета. Запустите «apt list --upgradable» для их показа.
```

Рисунок 2.12. Обновление системы

4. Установите менеджер Wazuh

```
sudo apt install wazuh-manager
```

```

root@vitokhin-VirtualBox:~# sudo apt install wazuh-manager
Чтение списков пакетов... Готово
Построение дерева зависимостей... Готово
Чтение информации о состоянии... Готово
Следующие пакеты устанавливались автоматически и больше не требуются:
  libflashrom1 libftdi1-2
Для их удаления используйте «sudo apt autoremove».
Предлагаемые пакеты:
  ехрест
Следующие НОВЫЕ пакеты будут установлены:
  wazuh-manager
Обновлено 0 пакетов, установлено 1 новых пакетов, для удаления отмечено 0 пакетов, и 3 пакетов не обновлено.
Необходимо скачать 120 МВ архивов.
После данной операции объем занятого дискового пространства возрастёт на 460 МВ.
Пол:1 https://packages.wazuh.com/4.x/apt stable/main amd64 wazuh-manager amd64 4.3.9-1 [120 МВ]
Получено 120 МВ за 19с (6 395 кВ/с)
Выбор ранее не выбранного пакета wazuh-manager.
(Чтение базы данных ... на данный момент установлено 177680 файлов и каталогов.)
Подготовка к распаковке .../wazuh-manager_4.3.9-1_amd64.deb ...
Распаковывается wazuh-manager (4.3.9-1) ...
Настраивается пакет wazuh-manager (4.3.9-1) ...
root@vitokhin-VirtualBox:~# █

```

Рисунок 2.13. Установка Wazuh-manager

5. Запустите и включите службу

```

sudo systemctl daemon-reload
sudo systemctl enable --now wazuh-manager

```

```

root@vitokhin-VirtualBox:~# sudo systemctl daemon-reload
root@vitokhin-VirtualBox:~# sudo systemctl enable --now wazuh-manager
Synchronizing state of wazuh-manager.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable wazuh-manager
Created symlink /etc/systemd/system/multi-user.target.wants/wazuh-manager.service → /lib/systemd/system/wazuh-manager.service.
root@vitokhin-VirtualBox:~# █

```

Рисунок 2.14. Запуск Wazuh-manager

6. Проверьте статус менеджера Wazuh и убедитесь, что он запущен и работает.

```

systemctl status wazuh-manager

```

```
root@vitokhin-VirtualBox:~# systemctl status wazuh-manager
● wazuh-manager.service - Wazuh manager
   Loaded: loaded (/lib/systemd/system/wazuh-manager.service; enabled; vendor>
   Active: active (running) since Tue 2022-11-08 17:16:47 EET; 51s ago
   Process: 41007 ExecStart=/usr/bin/env /var/ossec/bin/wazuh-control start (c>
   Tasks: 114 (limit: 4626)
   Memory: 607.6M
   CPU: 48.782s
   CGroup: /system.slice/wazuh-manager.service
           └─41065 /var/ossec/framework/python/bin/python3 /var/ossec/api/scr>
             └─41109 /var/ossec/bin/wazuh-authd
               └─41125 /var/ossec/bin/wazuh-db
                 └─41148 /var/ossec/bin/wazuh-execd
                   └─41161 /var/ossec/bin/wazuh-analysisd
                     └─41170 /var/ossec/bin/wazuh-syscheckd
                       └─41187 /var/ossec/bin/wazuh-remoted
                         └─41248 /var/ossec/bin/wazuh-logcollector
                           └─41258 /var/ossec/bin/wazuh-monitord
                             └─41267 /var/ossec/bin/wazuh-modulesd
                               └─41361 /var/ossec/framework/python/bin/python3 /var/ossec/api/scr>
                                 └─41364 /var/ossec/framework/python/bin/python3 /var/ossec/api/scr>

ноя 08 17:16:42 vitokhin-VirtualBox env[41007]: Started wazuh-db...
ноя 08 17:16:43 vitokhin-VirtualBox env[41007]: Started wazuh-execd...
ноя 08 17:16:43 vitokhin-VirtualBox env[41007]: Started wazuh-analysisd...
ноя 08 17:16:44 vitokhin-VirtualBox env[41007]: Started wazuh-syscheckd...
ноя 08 17:16:45 vitokhin-VirtualBox env[41007]: Started wazuh-remoted...
ноя 08 17:16:45 vitokhin-VirtualBox env[41007]: Started wazuh-logcollector...
ноя 08 17:16:45 vitokhin-VirtualBox env[41007]: Started wazuh-monitord...
ноя 08 17:16:45 vitokhin-VirtualBox env[41007]: Started wazuh-modulesd...
ноя 08 17:16:47 vitokhin-VirtualBox env[41007]: Completed.
ноя 08 17:16:47 vitokhin-VirtualBox systemd[1]: Started Wazuh manager.
lines 1-31/31 (END)
```

Рисунок 2.15. Статус сервиса Wazuh-manager

2.2 Установка стека ELK

1. Запустите установку Elasticsearch

```
sudo apt install elasticsearch-oss opendistroforelasti
```

```

root@vitokhin-VirtualBox:~# sudo apt install elasticsearch-oss opendistroforelasticsearch
Чтение списков пакетов... Готово
Построение дерева зависимостей... 0%
Построение дерева зависимостей... Готово
Чтение информации о состоянии... Готово
Следующие пакеты устанавливались автоматически и больше не требуются:
  libflashrom1 libftdi1-2
Для их удаления используйте «sudo apt autoremove».
Будут установлены следующие дополнительные пакеты:
  opendistro-alerting opendistro-anomaly-detection
  opendistro-asynchronous-search opendistro-index-management
  opendistro-job-scheduler opendistro-knn opendistro-knnlib
  opendistro-performance-analyzer opendistro-reports-scheduler
  opendistro-security opendistro-sql
Следующие НОВЫЕ пакеты будут установлены:
  elasticsearch-oss opendistro-alerting opendistro-anomaly-detection
  opendistro-asynchronous-search opendistro-index-management
  opendistro-job-scheduler opendistro-knn opendistro-knnlib
  opendistro-performance-analyzer opendistro-reports-scheduler
  opendistro-security opendistro-sql opendistroforelasticsearch
Обновлено 0 пакетов, установлено 13 новых пакетов, для удаления отмечено 0 пакетов, и 3 пакета не обновлено.
Необходимо скачать 392 МВ архивов.
После данной операции объём занятого дискового пространства возрастёт на 599 МВ.
Пол:1 https://packages.wazuh.com/4.x/apt stable/main amd64 elasticsearch-oss amd64 7.10.2 [231 MB]
Пол:2 https://packages.wazuh.com/4.x/apt stable/main amd64 opendistro-alerting all 1.13.1.0-1 [13,6 MB]
Пол:3 https://packages.wazuh.com/4.x/apt stable/main amd64 opendistro-anomaly-detection all 1.13.0.0-1 [9 134 kB]
Пол:4 https://packages.wazuh.com/4.x/apt stable/main amd64 opendistro-asynchronous-search all 1.13.0.1-1 [168 kB]
Пол:5 https://packages.wazuh.com/4.x/apt stable/main amd64 opendistro-index-management all 1.13.2.0-1 [7 271 kB]
Пол:6 https://packages.wazuh.com/4.x/apt stable/main amd64 opendistro-job-scheduler all 1.13.0.0-1 [975 kB]
Пол:7 https://packages.wazuh.com/4.x/apt stable/main amd64 opendistro-knnlib amd64 1.13.0.0 [740 kB]
Пол:8 https://packages.wazuh.com/4.x/apt stable/main amd64 opendistro-knn all 1.13.0.0-1 [2 797 kB]
Пол:9 https://packages.wazuh.com/4.x/apt stable/main amd64 opendistro-performance-analyzer all 1.13.0.0-1 [65,2 MB]
Пол:10 https://packages.wazuh.com/4.x/apt stable/main amd64 opendistro-reports-scheduler all 1.13.0.0-1 [5 441 kB]

```

Рисунок 2.16. Установка Elasticsearch

2. Загрузите пользовательский файл конфигурации как показано ниже:

```

sudo curl -so /etc/elasticsearch/elasticsearch.yml
https://packages.wazuh.com/resources/4.2/opendistro/elasticsearch/7.x/elasticsearch_all_in_one.yml

```

3. Настройте роли и пользователей Kibana с помощью приведенных ниже шаблонов:

Приведенные выше команды добавляют следующих пользователей для Kibana:

1. `Wazuh_user` – будет использоваться для пользователей, которым нужен доступ только для чтения к плагину Wazuh Kibana.

2. `Wazuh_admin` – для пользователей, которым нужны права администратора

Две дополнительные роли также создаются для предоставления пользователям соответствующих разрешений.

1. `wazuh_ui_user` – предоставляет права `wazuh_user` на чтение индексов Wazuh.

2. `wazuh_ui_admin` – позволяет `wazuh_admins` выполнять чтение/запись, управление и индексирование индексов wazuh.

2.3 Установка сертификатов

1. Удалите демонстрационные сертификаты

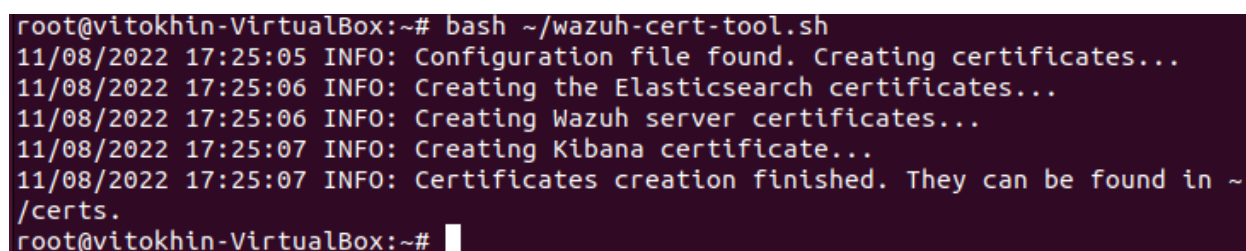
```
sudo rm -f /etc/elasticsearch/{esnode-key.pem,esnode.pem,kirk-key.pem,kirk.pem,root-ca.pem}
```

2. Загрузите `wazuh-cert-tool.sh`:

```
sudo su -
curl -sO ~/wazuh-cert-tool.sh
https://packages.wazuh.com/resources/4.2/open-distro/tools/certificate-utility/wazuh-cert-tool.sh
curl -sO ~/instances.yml
https://packages.wazuh.com/resources/4.2/open-distro/tools/certificate-utility/instances_aio.yml
```

3. Запустите `wazuh-cert-tool.sh`, чтобы создать сертификаты:

```
# bash ~/wazuh-cert-tool.sh
```



```
root@vitokhin-VirtualBox:~# bash ~/wazuh-cert-tool.sh
11/08/2022 17:25:05 INFO: Configuration file found. Creating certificates...
11/08/2022 17:25:06 INFO: Creating the Elasticsearch certificates...
11/08/2022 17:25:06 INFO: Creating Wazuh server certificates...
11/08/2022 17:25:07 INFO: Creating Kibana certificate...
11/08/2022 17:25:07 INFO: Certificates creation finished. They can be found in ~/certs.
root@vitokhin-VirtualBox:~#
```

Рисунок 2.17. Создание сертификатов

4. Переместите сертификаты Elasticsearch в соответствующее место:

```
mkdir /etc/elasticsearch/certs/
mv ~/certs/elasticsearch* /etc/elasticsearch/certs/
mv ~/certs/admin* /etc/elasticsearch/certs/
cp ~/certs/root-ca* /etc/elasticsearch/certs/
```


5. Уменьшить уязвимость Apache Log4j2 Remote Code Execution (RCE)

Добавьте следующую конфигурацию, чтобы уменьшить уязвимость Apache Log4j2 Remote Code Execution (RCE) – CVE-2021-44228 – ESA-2021-31.

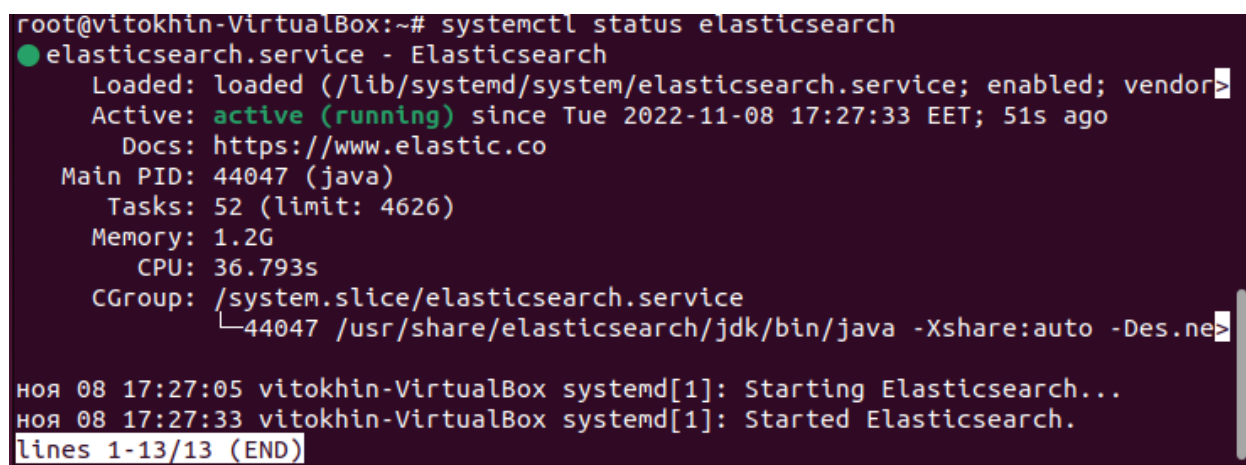
```
mkdir -p /etc/elasticsearch/jvm.options.d
echo          '-Dlog4j2.formatMsgNoLookups=true'      >
/etc/elasticsearch/jvm.options.d/disabledlog4j.options
chmod 2750 /etc/elasticsearch/jvm.options.d/disabledlog4j.options
chown                                root:elasticsearch
/etc/elasticsearch/jvm.options.d/disabledlog4j.options
```

6. Включите и запустите службу Elasticsearch:

```
systemctl daemon-reload
systemctl enable elasticsearch
systemctl start Elasticsearch
```

7. Проверьте статус сервиса Elasticsearch:

```
# systemctl status Elasticsearch
```



```
root@vitokhin-VirtualBox:~# systemctl status elasticsearch
● elasticsearch.service - Elasticsearch
   Loaded: loaded (/lib/systemd/system/elasticsearch.service; enabled; vendor
   Active: active (running) since Tue 2022-11-08 17:27:33 EET; 51s ago
     Docs: https://www.elastic.co
   Main PID: 44047 (java)
    Tasks: 52 (limit: 4626)
   Memory: 1.2G
      CPU: 36.793s
   CGroup: /system.slice/elasticsearch.service
           └─44047 /usr/share/elasticsearch/jdk/bin/java -Xshare:auto -Des.ne
ноя 08 17:27:05 vitokhin-VirtualBox systemd[1]: Starting Elasticsearch...
ноя 08 17:27:33 vitokhin-VirtualBox systemd[1]: Started Elasticsearch.
lines 1-13/13 (END)
```

Рисунок 2.18. Статус сервиса Elasticsearch

8. Запустите **securityadmin** скрипт Elasticsearch, чтобы загрузить информацию о новых сертификатах и запустить кластер:

```
export JAVA_HOME=/usr/share/elasticsearch/jdk/ &&
/usr/share/elasticsearch/plugins/opendistro_security/tools/securityadmin.sh -
cd /usr/share/elasticsearch/plugins/opendistro_security/securityconfig/ -nhnv
-cacert /etc/elasticsearch/certs/root-ca.pem -cert
/etc/elasticsearch/certs/admin.pem -key /etc/elasticsearch/certs/admin-
key.pem
```

```

root@vitokhin-VirtualBox:~# export JAVA_HOME=/usr/share/elasticsearch/jdk/ && /u
sr/share/elasticsearch/plugins/opendistro_security/tools/securityadmin.sh -cd /u
sr/share/elasticsearch/plugins/opendistro_security/securityconfig/ -nhnv -cacert
/etc/elasticsearch/certs/root-ca.pem -cert /etc/elasticsearch/certs/admin.pem -
key /etc/elasticsearch/certs/admin-key.pem
Open Distro Security Admin v7
Will connect to localhost:9300 ... done
Connected as CN=admin,OU=Docu,O=Wazuh,L=California,C=US
Elasticsearch Version: 7.10.2
Open Distro Security Version: 1.13.1.0
Contacting elasticsearch cluster 'elasticsearch' and wait for YELLOW clusterstat
e ...
Clustername: elasticsearch
Clusterstate: GREEN
Number of nodes: 1
Number of data nodes: 1
.opendistro_security index does not exists, attempt to create it ... done (0-all
replicas)
Populate config from /usr/share/elasticsearch/plugins/opendistro_security/securi
tyconfig/
Will update '_doc/config' with /usr/share/elasticsearch/plugins/opendistro_secur
ity/securityconfig/config.yml
  SUCC: Configuration for 'config' created or updated
Will update '_doc/roles' with /usr/share/elasticsearch/plugins/opendistro_securi
ty/securityconfig/roles.yml
  SUCC: Configuration for 'roles' created or updated
Will update '_doc/rolesmapping' with /usr/share/elasticsearch/plugins/opendistro
_security/securityconfig/roles_mapping.yml
  SUCC: Configuration for 'rolesmapping' created or updated
Will update '_doc/internalusers' with /usr/share/elasticsearch/plugins/opendistr
o_security/securityconfig/internal_users.yml
  SUCC: Configuration for 'internalusers' created or updated
Will update '_doc/actiongroups' with /usr/share/elasticsearch/plugins/opendistro
_security/securityconfig/action_groups.yml
  SUCC: Configuration for 'actiongroups' created or updated
Will update '_doc/tenants' with /usr/share/elasticsearch/plugins/opendistro_secu
rity/securityconfig/tenants.yml
  SUCC: Configuration for 'tenants' created or updated
Will update '_doc/nodesdn' with /usr/share/elasticsearch/plugins/opendistro_secu
rity/securityconfig/nodes_dn.yml
  SUCC: Configuration for 'nodesdn' created or updated
Will update '_doc/whitelist' with /usr/share/elasticsearch/plugins/opendistro_se
curity/securityconfig/whitelist.yml
  SUCC: Configuration for 'whitelist' created or updated
Will update '_doc/audit' with /usr/share/elasticsearch/plugins/opendistro_securi
ty/securityconfig/audit.yml
  SUCC: Configuration for 'audit' created or updated

```

Рисунок 2.19. Запуск скрипта Elasticsearch

9. Запустите команду ниже, чтобы подтвердить, что установка прошла успешно:

```
curl -XGET https://localhost:9200 -u admin:admin -k
```

```

root@vitokhin-VirtualBox:~# curl -XGET https://localhost:9200 -u admin:admin -k
{
  "name" : "node-1",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "I_fS1bo3QNSF5_hcDvJ-NA",
  "version" : {
    "number" : "7.10.2",
    "build_flavor" : "oss",
    "build_type" : "deb",
    "build_hash" : "747e1cc71def077253878a59143c1f785afa92b9",
    "build_date" : "2021-01-13T00:42:12.435326Z",
    "build_snapshot" : false,
    "lucene_version" : "8.7.0",
    "minimum_wire_compatibility_version" : "6.8.0",
    "minimum_index_compatibility_version" : "6.0.0-beta1"
  },
  "tagline" : "You Know, for Search"
}
root@vitokhin-VirtualBox:~# █

```

Рисунок 2.20. Проверка установки Elasticsearch

2.4 Установка Filebeat

1. Запустите установка Filebeat

```
sudo apt install filebeat
```

```

root@vitokhin-VirtualBox:~# sudo apt install filebeat
Чтение списков пакетов... Готово
Построение дерева зависимостей... Готово
Чтение информации о состоянии... Готово
Следующие пакеты устанавливались автоматически и больше не требуются:
 libflashrom1 libftdi1-2
Для их удаления используйте «sudo apt autoremove».
Следующие НОВЫЕ пакеты будут установлены:
 filebeat
Обновлено 0 пакетов, установлено 1 новых пакетов, для удаления отмечено 0 пакетов, и 3 пакетов не обновлено.
Необходимо скачать 22,1 МВ архивов.
После данной операции объём занятого дискового пространства возрастёт на 73,6 МВ .
Пол:1 https://packages.wazuh.com/4.x/apt/stable/main amd64 filebeat amd64 7.10.2 [22,1 МВ]
Получено 22,1 МВ за 8с (2 920 kB/s)
Выбор ранее не выбранного пакета filebeat.
(Чтение базы данных ... на данный момент установлено 197478 файлов и каталогов.)
Подготовка к распаковке .../filebeat_7.10.2_amd64.deb ...
Распаковывается filebeat (7.10.2) ...
Настраивается пакет filebeat (7.10.2) ...
root@vitokhin-VirtualBox:~# █

```

Рисунок 2.21. Установка Filebeat

2. Загрузите приведенный ниже файл конфигурации filebeat, который будет использоваться для пересылки предупреждений wazuh в Elasticsearch.

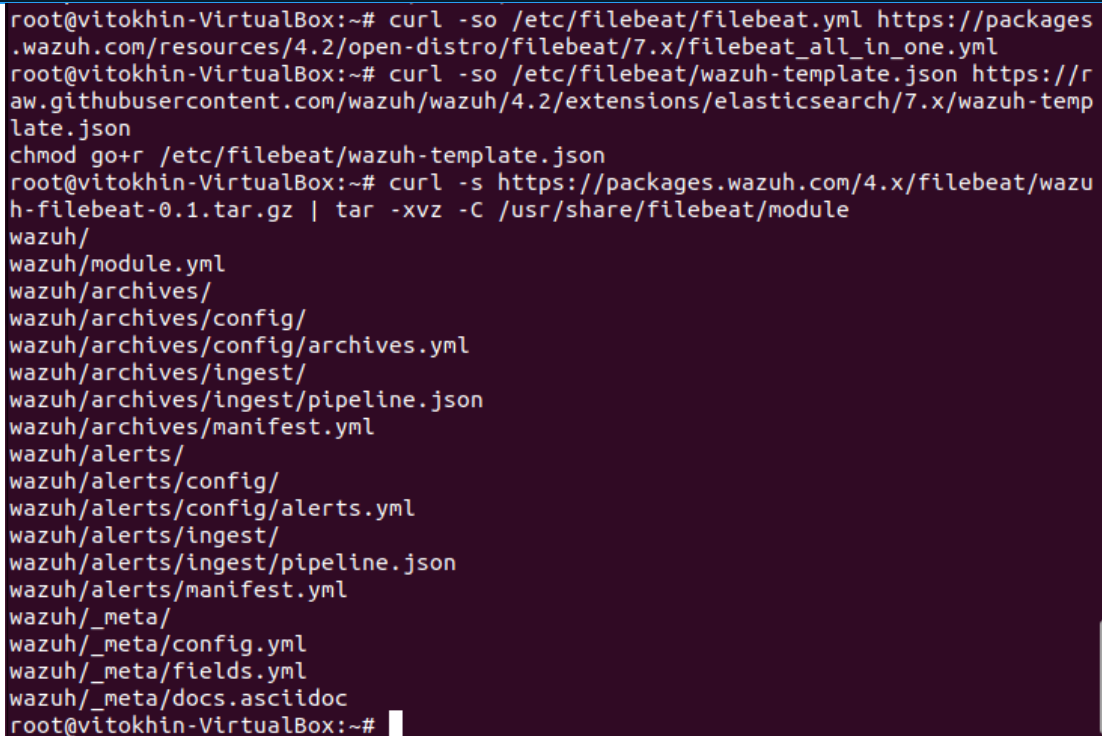
```
curl -sO https://packages.wazuh.com/resources/4.2/open-distro/filebeat/7.x/filebeat_all_in_one.yml /etc/filebeat/filebeat.yml
```

3. Загрузите шаблон предупреждений с помощью приведенной ниже команды для Elasticsearch:

```
curl -so /etc/filebeat/wazuh-template.json https://raw.githubusercontent.com/wazuh/wazuh/4.2/extensions/elasticsearch/7.x/wazuh-template.json
chmod go+r /etc/filebeat/wazuh-template.json
```

4. Загрузите модуль Wazuh Filebeat:

```
curl -s https://packages.wazuh.com/4.x/filebeat/wazuh-filebeat-0.1.tar.gz | tar -xvz -C /usr/share/filebeat/module
```



```
root@vitokhin-VirtualBox:~# curl -so /etc/filebeat/filebeat.yml https://packages.wazuh.com/resources/4.2/open-distro/filebeat/7.x/filebeat_all_in_one.yml
root@vitokhin-VirtualBox:~# curl -so /etc/filebeat/wazuh-template.json https://raw.githubusercontent.com/wazuh/wazuh/4.2/extensions/elasticsearch/7.x/wazuh-template.json
chmod go+r /etc/filebeat/wazuh-template.json
root@vitokhin-VirtualBox:~# curl -s https://packages.wazuh.com/4.x/filebeat/wazuh-filebeat-0.1.tar.gz | tar -xvz -C /usr/share/filebeat/module
wazuh/
wazuh/module.yml
wazuh/archives/
wazuh/archives/config/
wazuh/archives/config/archives.yml
wazuh/archives/ingest/
wazuh/archives/ingest/pipeline.json
wazuh/archives/manifest.yml
wazuh/alerts/
wazuh/alerts/config/
wazuh/alerts/config/alerts.yml
wazuh/alerts/ingest/
wazuh/alerts/ingest/pipeline.json
wazuh/alerts/manifest.yml
wazuh/_meta/
wazuh/_meta/config.yml
wazuh/_meta/fields.yml
wazuh/_meta/docs.asciidoc
root@vitokhin-VirtualBox:~#
```

Рисунок 2.22. Загрузка модуля Wazuh Filebeat

5. Скопируйте сертификаты Elasticsearch в/etc/filebeat/certs

```
mkdir /etc/filebeat/certs
cp ~/certs/root-ca.pem /etc/filebeat/certs/
mv ~/certs/filebeat* /etc/filebeat/certs/
```

6. Запустите и включите службу Filebeat

```
systemctl daemon-reload
systemctl enable --now filebeat
```

7. Подтвердите конфигурацию Filebeat командой ниже:

```
# filebeat test output
```

```

root@vitokhin-VirtualBox:~# mkdir /etc/filebeat/certs
root@vitokhin-VirtualBox:~# cp ~/certs/root-ca.pem /etc/filebeat/certs/
root@vitokhin-VirtualBox:~# mv ~/certs/filebeat* /etc/filebeat/certs/
root@vitokhin-VirtualBox:~# systemctl daemon-reload
root@vitokhin-VirtualBox:~# systemctl enable --now filebeat
Synchronizing state of filebeat.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable filebeat
Created symlink /etc/systemd/system/multi-user.target.wants/filebeat.service → /lib/systemd/system/filebeat.service.
root@vitokhin-VirtualBox:~# filebeat test output
elasticsearch: https://127.0.0.1:9200...
  parse url... OK
  connection...
    parse host... OK
    dns lookup... OK
    addresses: 127.0.0.1
    dial up... OK
  TLS...
    security: server's certificate chain verification is enabled
    handshake... OK
    TLS version: TLSv1.3
    dial up... OK
  talk to server... OK
  version: 7.10.2
root@vitokhin-VirtualBox:~#

```

Рисунок 2.23. Проверка конфигурации Filebeat

2.5 Установка Kibana

1. Используйте приведенную ниже команду для установки Kibana

```
apt install opendistroforelasticsearch-kibana
```

```

root@vitokhin-VirtualBox:~# apt install opendistroforelasticsearch-kibana
Чтение списков пакетов... Готово
Построение дерева зависимостей... Готово
Чтение информации о состоянии... Готово
Следующие пакеты устанавливались автоматически и больше не требуются:
 libflashrom1 libftdi1-2
Для их удаления используйте «apt autoremove».
Следующие НОВЫЕ пакеты будут установлены:
 opendistroforelasticsearch-kibana
Обновлено 0 пакетов, установлено 1 новых пакетов, для удаления отмечено 0 пакетов,
 и 3 пакетов не обновлено.
Необходимо скачать 234 МВ архивов.
После данной операции объём занятого дискового пространства возрастёт на 692 МВ.
Пол:1 https://packages.wazuh.com/4.x/apt/stable/main amd64 opendistroforelasticsearch-kibana amd64 1.13.2 [234 MB]
Получено 234 МВ за 32с (7 222 kB/s)
Выбор ранее не выбранного пакета opendistroforelasticsearch-kibana.
(Чтение базы данных ... на данный момент установлено 197797 файлов и каталогов.)
Подготовка к распаковке ../opendistroforelasticsearch-kibana_1.13.2_amd64.deb ...
Распаковывается opendistroforelasticsearch-kibana (1.13.2) ...
Настраивается пакет opendistroforelasticsearch-kibana (1.13.2) ...
show: невозможно получить доступ к '/usr/share/kibana/optimize': Нет такого файла или каталога
no optimize folder
root@vitokhin-VirtualBox:~# apt install opendistroforelasticsearch-kibana
Чтение списков пакетов... Готово
Построение дерева зависимостей... Готово
Чтение информации о состоянии... Готово
Уже установлен пакет opendistroforelasticsearch-kibana самой новой версии (1.13.2).
Следующие пакеты устанавливались автоматически и больше не требуются:
 libflashrom1 libftdi1-2
Для их удаления используйте «apt autoremove».
Обновлено 0 пакетов, установлено 0 новых пакетов, для удаления отмечено 0 пакетов,
 и 3 пакетов не обновлено.
root@vitokhin-VirtualBox:~#

```

Рисунок 2.24. Установка Kibana

2. Загрузите файл конфигурации для Kibana

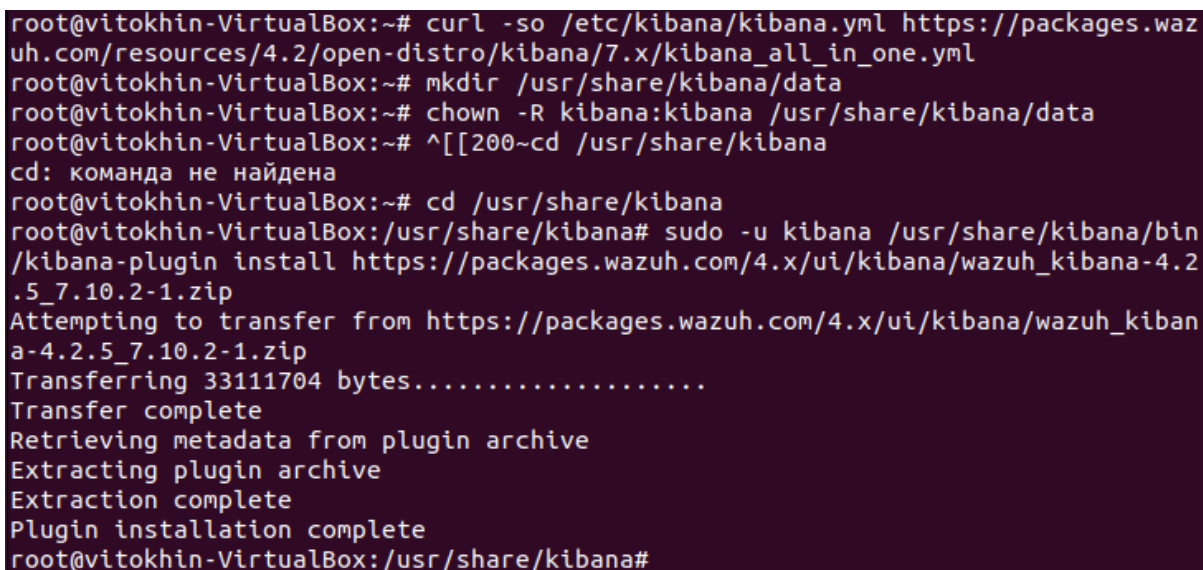
```
curl -so /etc/kibana/kibana.yml
https://packages.wazuh.com/resources/4.2/opendistro/kibana/7.x/kibana_all_in_one.yml
```

3. Создайте каталог данных для Kibana

```
mkdir /usr/share/kibana/data  
chown -R kibana:kibana /usr/share/kibana/data
```

4. Установите плагин Wazuh Kibana.

```
cd /usr/share/kibana  
sudo -u kibana /usr/share/kibana/bin/kibana-plugin install  
https://packages.wazuh.com/4.x/ui/kibana/wazuh\_kibana-4.2.5\_7.10.2-1.zip
```



```
root@vitokhin-VirtualBox:~# curl -so /etc/kibana/kibana.yml https://packages.wazuh.com/resources/4.2/open-distro/kibana/7.x/kibana_all_in_one.yml  
root@vitokhin-VirtualBox:~# mkdir /usr/share/kibana/data  
root@vitokhin-VirtualBox:~# chown -R kibana:kibana /usr/share/kibana/data  
root@vitokhin-VirtualBox:~# ^[[200~cd /usr/share/kibana  
cd: команда не найдена  
root@vitokhin-VirtualBox:~# cd /usr/share/kibana  
root@vitokhin-VirtualBox:/usr/share/kibana# sudo -u kibana /usr/share/kibana/bin/kibana-plugin install https://packages.wazuh.com/4.x/ui/kibana/wazuh_kibana-4.2.5_7.10.2-1.zip  
Attempting to transfer from https://packages.wazuh.com/4.x/ui/kibana/wazuh_kibana-4.2.5_7.10.2-1.zip  
Transferring 33111704 bytes.....  
Transfer complete  
Retrieving metadata from plugin archive  
Extracting plugin archive  
Extraction complete  
Plugin installation complete  
root@vitokhin-VirtualBox:/usr/share/kibana#
```

Рисунок 2.25. Установка плагина Wazuh Kibana

5. Скопируйте сертификаты Elasticsearch в /etc/kibana/certs:

```
mkdir /etc/kibana/certs  
cp ~/certs/root-ca.pem /etc/kibana/certs/  
mv ~/certs/kibana* /etc/kibana/certs/  
chown kibana:kibana /etc/kibana/certs/*
```

6. Привяжите сокет Kibana к привилегированному порту 443:

```
setcap 'cap_net_bind_service=+ep' /usr/share/kibana/node/bin/node
```

7. Запустите и включите службу Kibana

```
systemctl daemon-reload  
systemctl enable --now kibana
```

8. Разрешите подключение Kibana, откройте порт 443 через брандмауэр

```
sudo ufw allow 443/tcp
```

```

root@vitokhin-VirtualBox:/usr/share/kibana# mkdir /etc/kibana/certs
root@vitokhin-VirtualBox:/usr/share/kibana# cp ~/certs/root-ca.pem /etc/kibana/certs/
root@vitokhin-VirtualBox:/usr/share/kibana# mv ~/certs/kibana* /etc/kibana/certs/
root@vitokhin-VirtualBox:/usr/share/kibana# chown kibana:kibana /etc/kibana/certs/*
root@vitokhin-VirtualBox:/usr/share/kibana# setcap 'cap_net_bind_service=+ep' /usr/share/kibana/node/bin/node
root@vitokhin-VirtualBox:/usr/share/kibana# systemctl daemon-reload
root@vitokhin-VirtualBox:/usr/share/kibana# systemctl enable --now kibana
Synchronizing state of kibana.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable kibana
Created symlink /etc/systemd/system/multi-user.target.wants/kibana.service → /etc/systemd/system/kibana.service.
root@vitokhin-VirtualBox:/usr/share/kibana# sudo ufw allow 443/tcp
Правила обновлены
Правила обновлены (v6)
root@vitokhin-VirtualBox:/usr/share/kibana#

```

Рисунок 2.26. Открытие порта

9. Теперь вы можете получить доступ к интерфейсу wazuh kibana через <https://> «Ваш ip адрес (введите в терминале ‘ip add’)»
 Логин: admin
 Пароль: admin

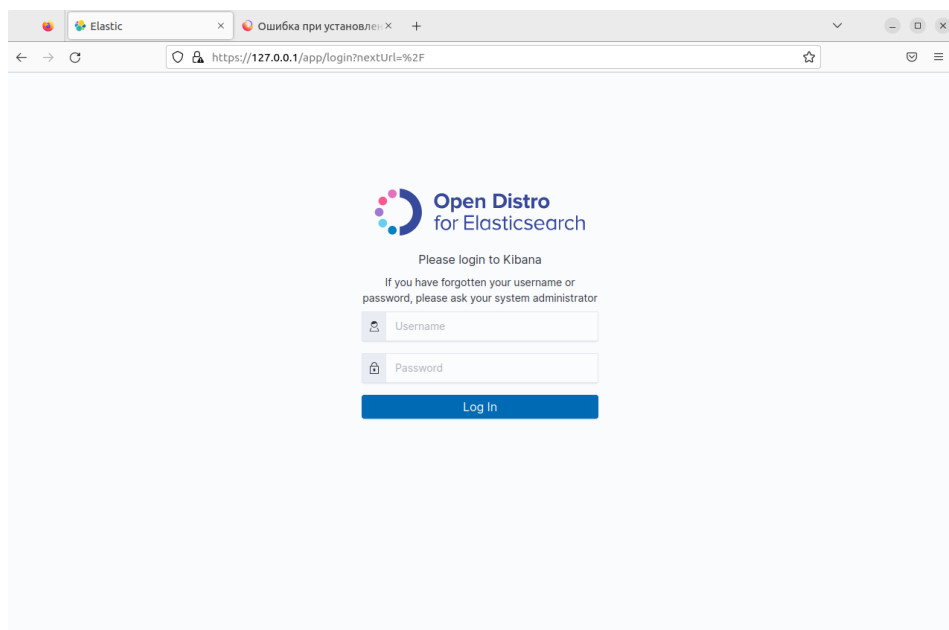


Рисунок 2.27. Окно ELK Wazuh

После авторизации, произойдет проверка установленных модулей. У некоторых возникает проблема с API version, из-за несоответствия версий Wazuh APP. Попробуйте установить другой модуль Kibana Wazuh изменив *.* на ту версию которую просит установить

```
sudo -u kibana /usr/share/kibana/bin/kibana-plugin install https://packages.wazuh.com/4.x/ui/kibana/wazuh_kibana-4.*.*_7.10.2-1.zip
```

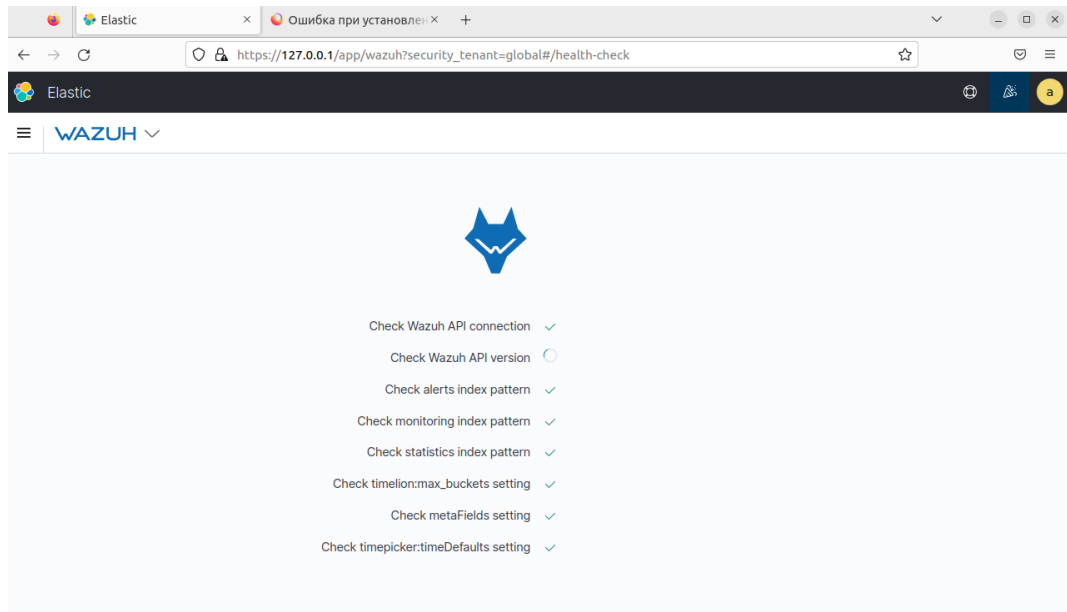


Рисунок 2.28. Проверка установленных модулей

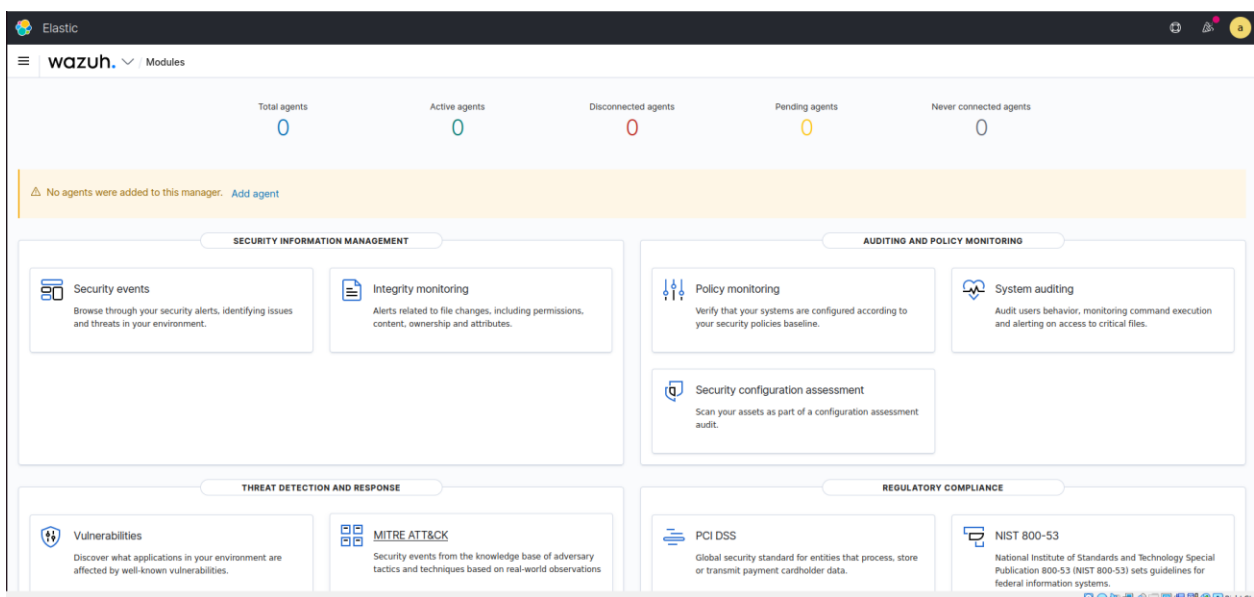


Рисунок – 2.29 Главное меню

2.5 Установка агента на Linux Ubuntu

Deploy a new agent

× Close

- 1 Choose the Operating system**
- 2 Choose the architecture**
- 3 Wazuh server address**

This is the address the agent uses to communicate with the Wazuh server. It can be an IP address or a fully qualified domain name (FQDN).
- 4 Assign the agent to a group**

Select one or more existing groups
- 5 Install and enroll the agent**

You can use this command to install and enroll the Wazuh agent in one or more hosts.

ⓘ If the installer finds another Wazuh agent in the system, it will upgrade it preserving the configuration.

```
curl -so wazuh-agent-4.3.9.deb https://packages.wazuh.com/4.x/apt/pool/main/w/wazuh-agent/wazuh-agent_4.3.9-1_amd64.deb && sudo WAZUH_MANAGER='192.168.40.133' WAZUH_AGENT_GROUP='default' dpkg -i ./wazuh-agent-4.3.9.deb
```
- 6 Start the agent**

Systemd SysV Init

```
sudo systemctl daemon-reload
sudo systemctl enable wazuh-agent
sudo systemctl start wazuh-agent
```

To verify the connection with the Wazuh server, please follow this [document](#).

Рисунок 2.30. Конфигурирование скрипта для добавления агента

```
root@agent1-virtual-machine:~# curl -so wazuh-agent-4.3.9.deb https://packages.wazuh.com/4.x/apt/pool/main/w/wazuh-agent/wazuh-agent_4.3.9-1_amd64.deb &&
sudo WAZUH_MANAGER='192.168.40.134' WAZUH_AGENT_GROUP='default' dpkg -i ./wazuh-agent-4.3.9.deb
Выбор ранее не выбранного пакета wazuh-agent.
(Чтение базы данных ... на данный момент установлено 178564 файла и каталога.)
Подготовка к распаковке ./wazuh-agent-4.3.9.deb ...
Распаковывается wazuh-agent (4.3.9-1) ...
Настраивается пакет wazuh-agent (4.3.9-1) ...
root@agent1-virtual-machine:~# sudo systemctl daemon-reload
root@agent1-virtual-machine:~# sudo systemctl enable wazuh-agent
Synchronizing state of wazuh-agent.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable wazuh-agent
Created symlink /etc/systemd/system/multi-user.target.wants/wazuh-agent.service → /lib/systemd/system/wazuh-agent.service.
root@agent1-virtual-machine:~# ^[[200-sudo systemctl daemon-reload
sudo: команда не найдена
root@agent1-virtual-machine:~# sudo systemctl enable wazuh-agent
sudo systemctl start wazuh-agent^[[201-Synchronizing state of wazuh-agent.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable wazuh-agent
^C
root@agent1-virtual-machine:~# sudo systemctl start wazuh-agent
```

Рисунок 2.31. Установка агента

```
root@agent1-virtual-machine:~# systemctl status wazuh-agent
● wazuh-agent.service - Wazuh agent
   Loaded: loaded (/lib/systemd/system/wazuh-agent.service; enabled; vendor p
   Active: active (running) since Sat 2022-11-12 01:17:28 EET; 13s ago
   Process: 7256 ExecStart=/usr/bin/env /var/ossec/bin/wazuh-control start (co
   Tasks: 28 (limit: 4584)
   Memory: 65.6M
   CPU: 6.253s
   CGroup: /system.slice/wazuh-agent.service
           └─7278 /var/ossec/bin/wazuh-execd
             └─7290 /var/ossec/bin/wazuh-agentd
               └─7303 /var/ossec/bin/wazuh-syscheckd
                 └─7314 /var/ossec/bin/wazuh-logcollector
                   └─7322 /var/ossec/bin/wazuh-modulesd

ноя 12 01:17:22 agent1-virtual-machine systemd[1]: Starting Wazuh agent...
ноя 12 01:17:22 agent1-virtual-machine env[7256]: Starting Wazuh v4.3.9...
ноя 12 01:17:23 agent1-virtual-machine env[7256]: Started wazuh-execd...
ноя 12 01:17:24 agent1-virtual-machine env[7256]: Started wazuh-agentd...
ноя 12 01:17:25 agent1-virtual-machine env[7256]: Started wazuh-syscheckd...
ноя 12 01:17:25 agent1-virtual-machine env[7256]: Started wazuh-logcollector...
ноя 12 01:17:26 agent1-virtual-machine env[7256]: Started wazuh-modulesd...
ноя 12 01:17:28 agent1-virtual-machine env[7256]: Completed.
ноя 12 01:17:28 agent1-virtual-machine systemd[1]: Started Wazuh agent.
```

Рисунок 2.32. Статус сервиса Wazuh-agent

Обновляем страницу Wazuh на сервере и видим, что агент подключен

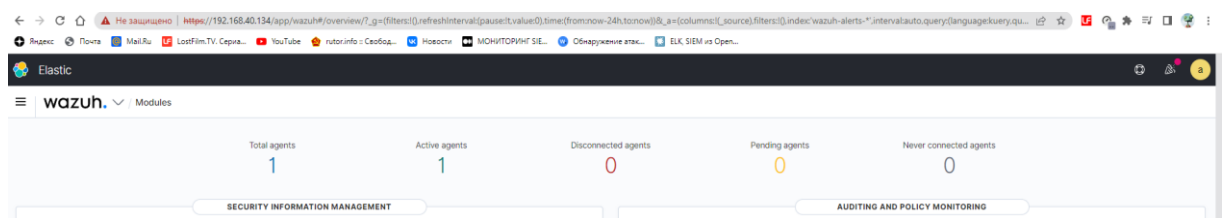


Рисунок 2.33. Подключен агент

Нажав на «Total agent» мы переходим в меню, где можем увидеть состояние всех агентов: ip-адрес, os, cluster node, дата регистрации, последняя активность, статус, имя.

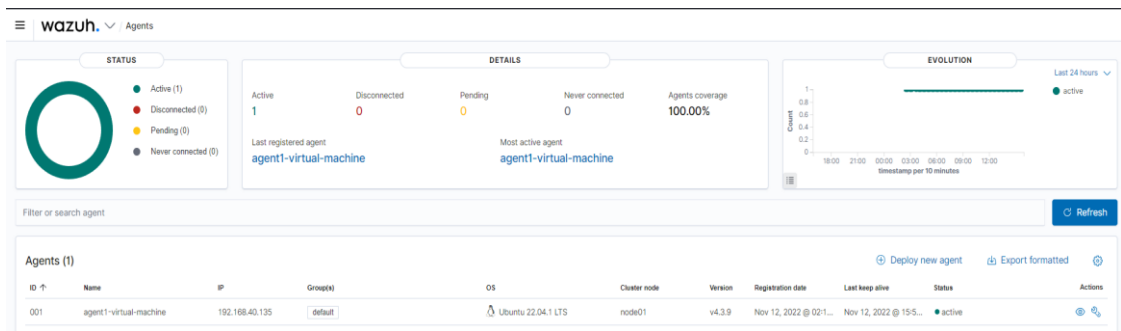


Рисунок 2.34. Состояние агентов

В этом же меню мы нажимаем на нашего агента и переходим к более детальной статистике конкретной машины:

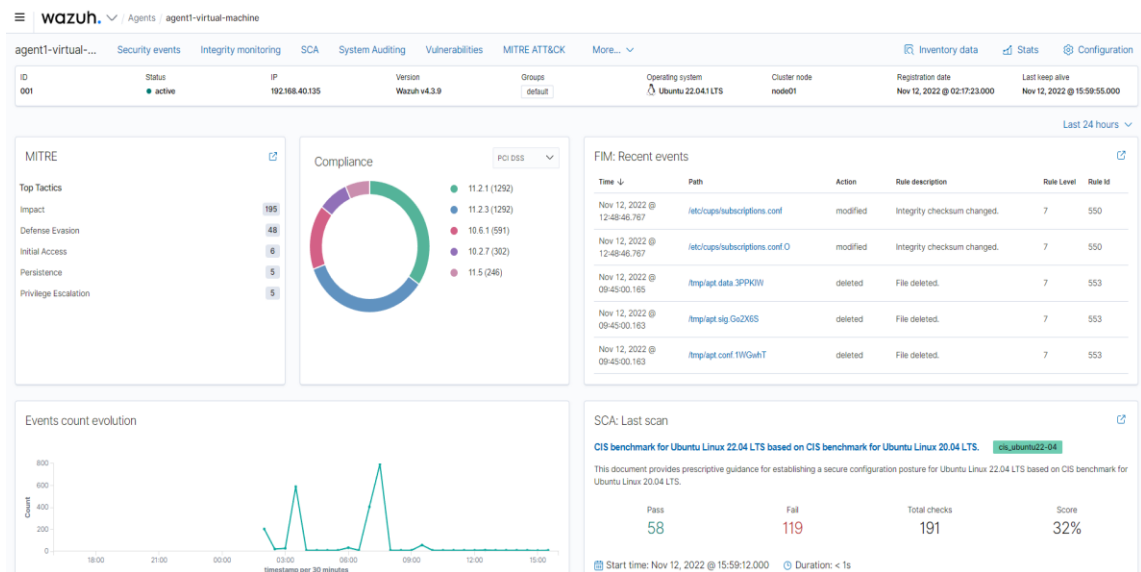


Рисунок 2.35. Статистика агента

В этом меню отображается статистика об изменениях файлов в системе агента, показывает информацию о предупреждениях, атаках (MITRE), график активности в системе, SCA и статистика проверки.

Вверху есть панель с установленными модулями такие как: Security event, Integrity monitoring, SCA, System Auditing, Vulnerabilities и т.д.

Перейдем к Security events

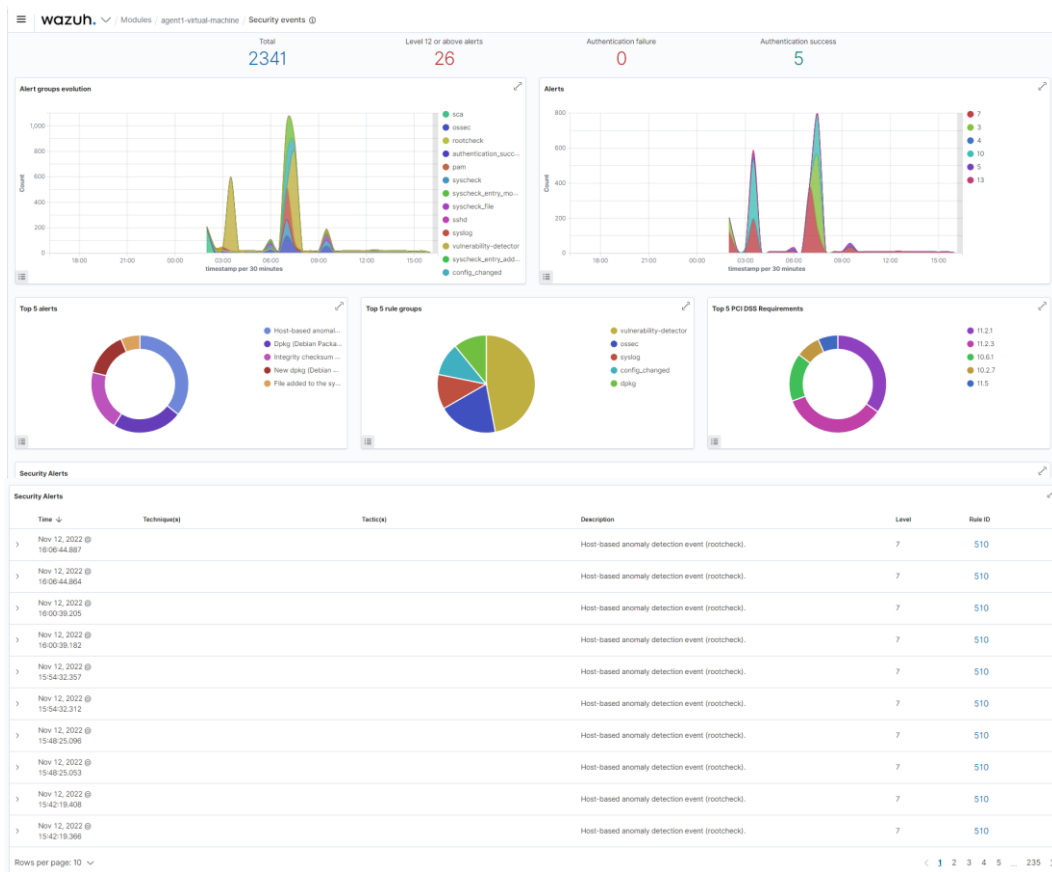


Рисунок 2.36. Security event

Здесь находятся графики тревог, отсортированные по группам «Alert groups evolution», график тревог отсортированные по критичности «Alert» и в самом низу логи этих тревог.

Table	JSON	Rule
@timestamp		2022-11-12T13:06:44.887Z
_id		caHza4QB65xwGRjKIV7z
agent.id		001
agent.ip		192.168.40.135
agent.name		agent1-virtual-machine
data.file		/usr/bin/gff
data.title		Trojaned version of file detected.
decoder.name		rootcheck
full_log		Trojaned version of file '/usr/bin/gff' detected. Signature used: 'bash 'bin/sh fwi_hjprocl_hj sevl 'n 'bin/*sh' (Generic).
id		1668258404.5383323
input.type		log
location		rootcheck
manager.name		puppetmaster
rule.description		Host-based anomaly detection event (rootcheck).
rule.firetimes		6
rule.gdpr		IV_35.7.d
rule.groups		ossec, rootcheck
rule.id		510
rule.level		7
rule.mail		false
rule.pci_dss		10.6.1
timestamp		2022-11-12T16:06:44.887-0300

Рисунок 2.37. Лог тревоги. Обнаружение трояна в файле

Переходим в Integrity monitoring

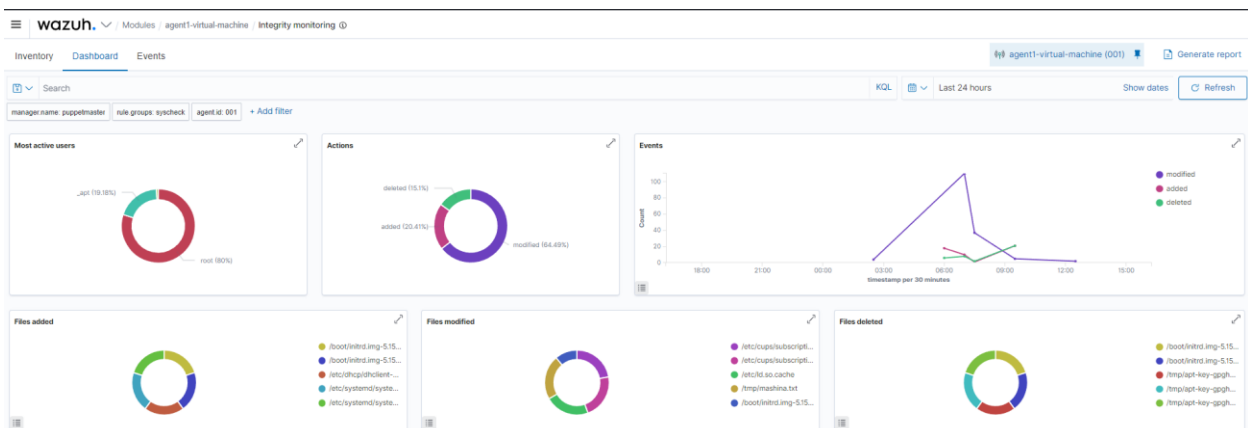


Рисунок 2.38. Integrity monitoring

Здесь собрана вся информация, связанная с целостностью файлов: изменение, удаление, открытие, пользователи, которые открывали, директории в которых была модификация.

Добавим в `ossec.conf` на агенте новое правило для проверки папки `/tmp` на целостность (код будет в приложении 2). И создадим файл и через время изменим содержимое.

`/tmp/mashina.txt`

Details

<p>Last analysis Nov 12, 2022 @ 02:42:13.000</p> <p>User ID 0</p> <p>Size 25 Bytes</p> <p>SHA1 0f80422a139f0f56b038638fec716b5fd8d892</p> <p>Permissions rw-r--r--</p>	<p>Last modified Nov 12, 2022 @ 02:42:13.000</p> <p>Group root</p> <p>Inode 817935</p> <p>SHA256 728c08122880063399298c9edb6b2c1d39f2bcef31f6dfc56a2904c30f74f060</p>	<p>User root</p> <p>Group ID 0</p> <p>MD5 c33434c9f10f603da333f0c362b60ad0</p>
--	---	--

Recent events 2 hits

Time ↓	Action	Description	Level	Rule ID
Nov 12, 2022 @ 02:42:13.438	modified	Integrity checksum changed.	7	550
Nov 12, 2022 @ 02:40:04.319	modified	Integrity checksum changed.	7	550

Rows per page: 10

Рисунок 2.39. Информация о файле собранная Integrity monitoring

Видим два изменения, открываем и посмотрим, что изменилось

```

rule.tsc                PI1.4, PI1.5, CC6.1, CC6.8, CC7.2, CC7.3
syscheck.changed_attributes  size, mtime, md5, sha1, sha256
syscheck.diff           1a2,5 > dasgadf > dgadfg > fasadfga > dfg 3,6c7,15 < < fas < df < --- > g > a > gafd > g > dfg > a > dfg > a > df
syscheck.event          modified
syscheck.gid_after      0
syscheck.gname_after    root
syscheck.inode_after    817935

```

Рисунок 2.40. Изменение в файле

В `syscheck.diff` записывается изначальное состояние файла и конечное.

Дальше переходим к SCA. После завершения сканирования агент сообщает результаты менеджеру, и они становятся доступными в пользовательском веб-интерфейсе (плагин Wazuh Kibana).



Рисунок 2.41. Sca

Взглянув на детали, мы можем увидеть различные поля для каждой проверки, включая обоснование, исправление и описание, и даже соответствие нормативным требованиям (например, PCI DSS).

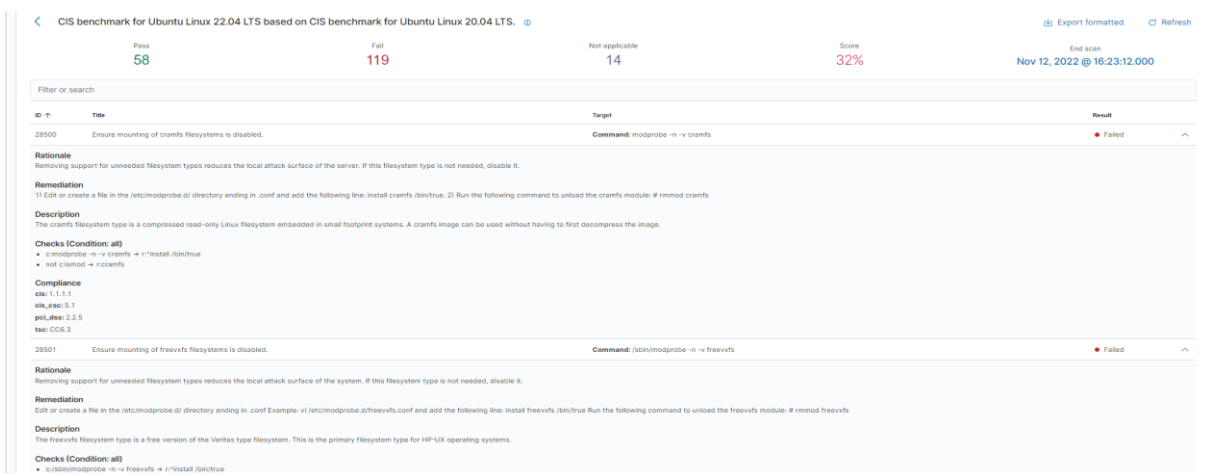


Рисунок 2.42. Детали проверки SCA

Переходим к «Vulnerabilities». Здесь мы сможем увидеть уязвимости в системе. Которые отсортированы по уровню критичности

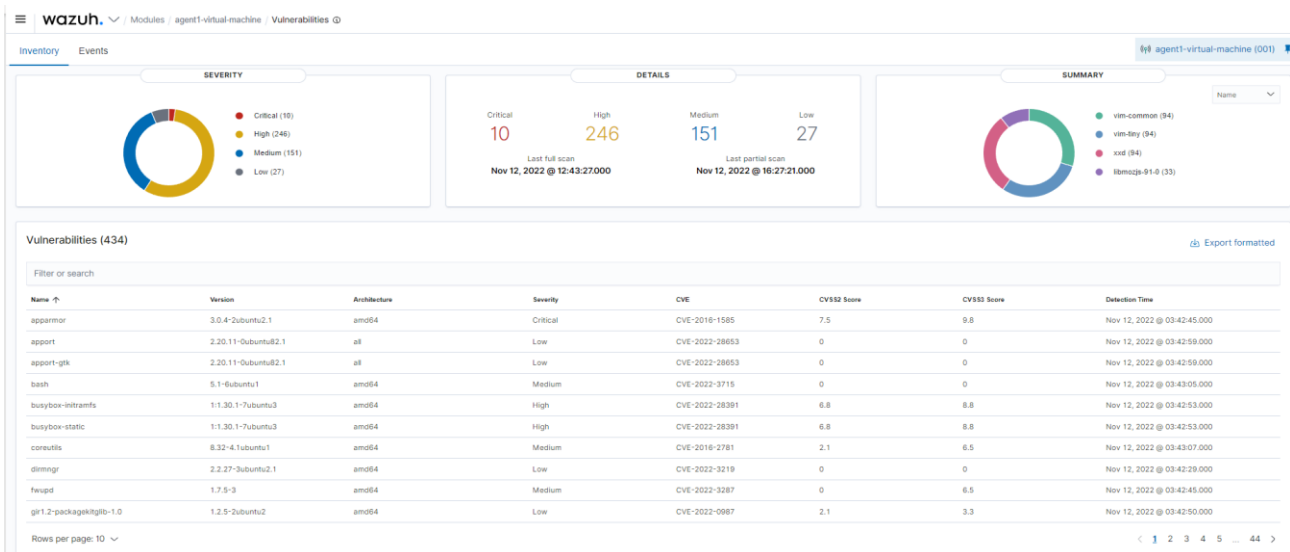


Рисунок 2.43. Vulnerabilities

Последнее что осталось рассмотреть это MITRE ATT&CK.

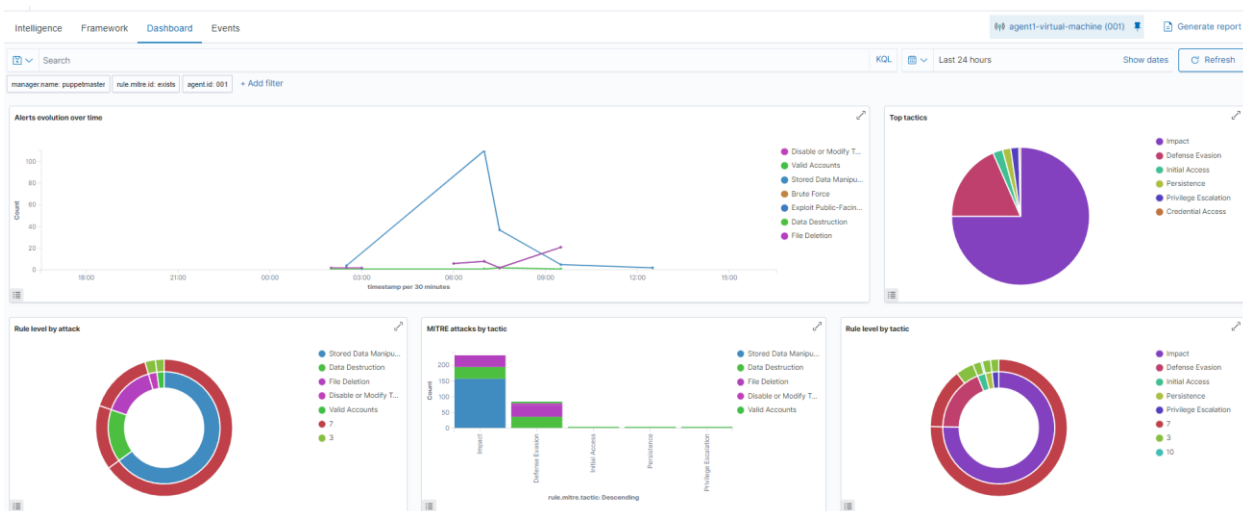


Рисунок 2.44. Предпринятые действия по защите от MITRE ATT&CK

Были попытки атаки с Kali через hping3 и nmap, DDoS на агента (192.168.40.135), но безуспешно. Разведка не смогла обнаружить открытые порты (на агенте специально открыл 80 порт), ddos по ip не дал результатов, hping3 тоже не смог.

Здесь видим графики активности тревог за последние 24 часа. Предпринятые действия (Impact, Defense Evasion, Initial Access, Persistence, Privilege Escalation, Credential Access), Тип атаки (уничтожение данных, манипуляция с исходным кодом и т.д)

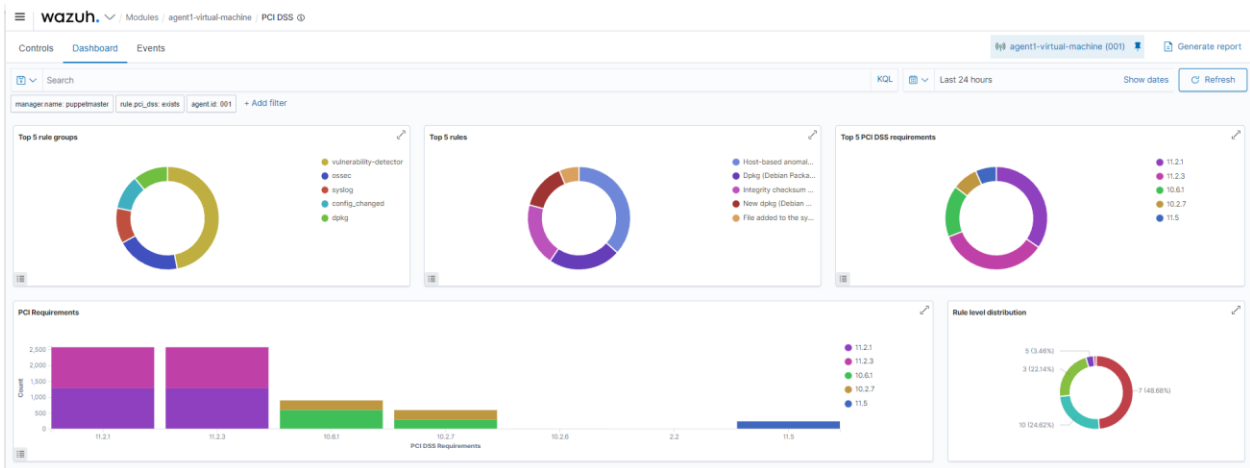


Рисунок 2.45. PCI DSS

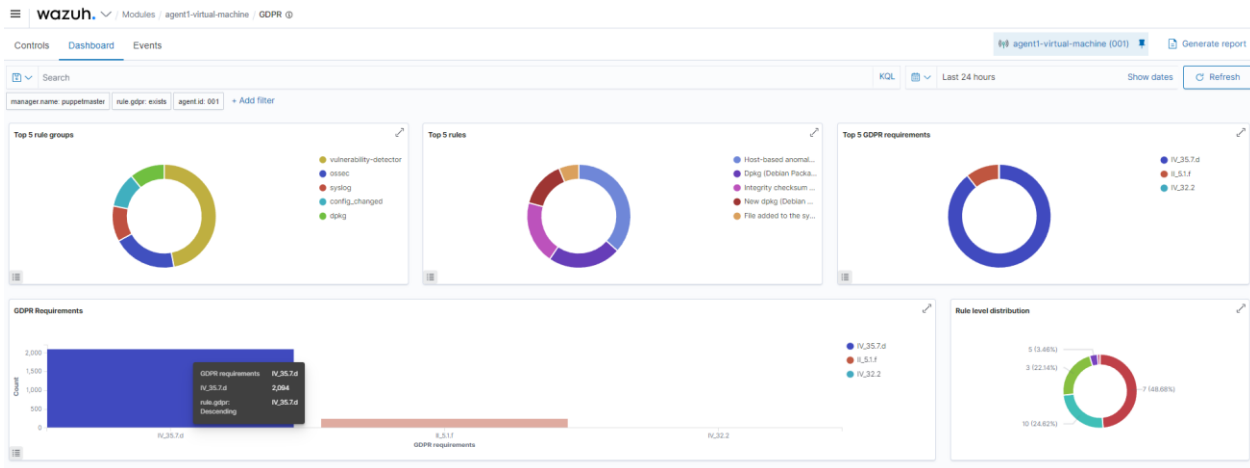


Рисунок 2.46. GDPR

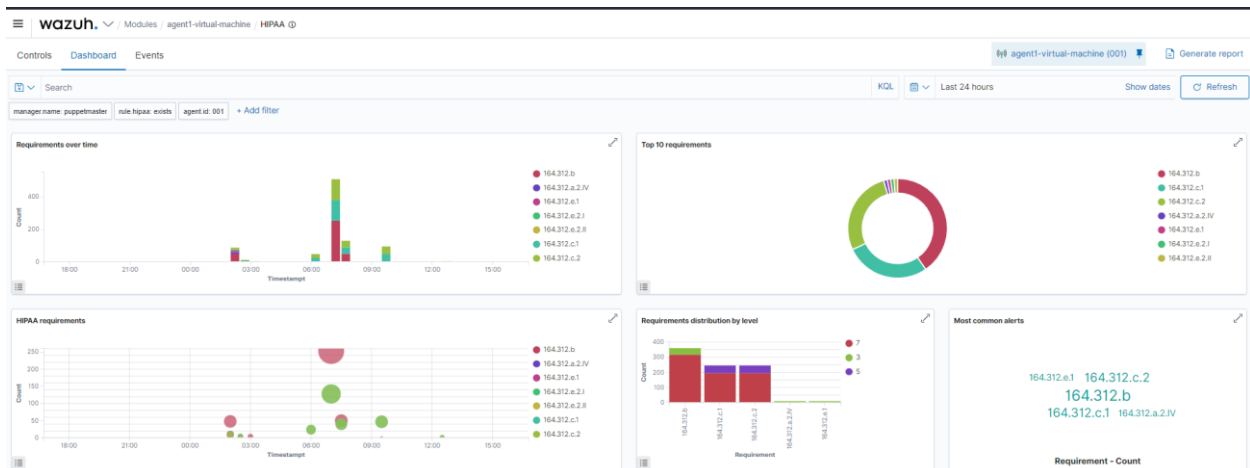


Рисунок 2.47. HIPAA

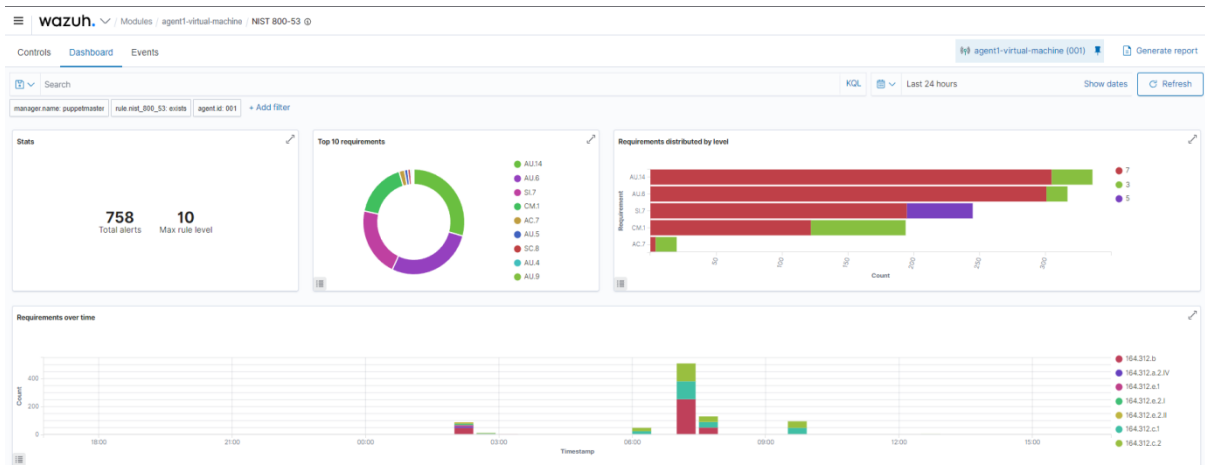


Рисунок 2.48. NIST 800-53

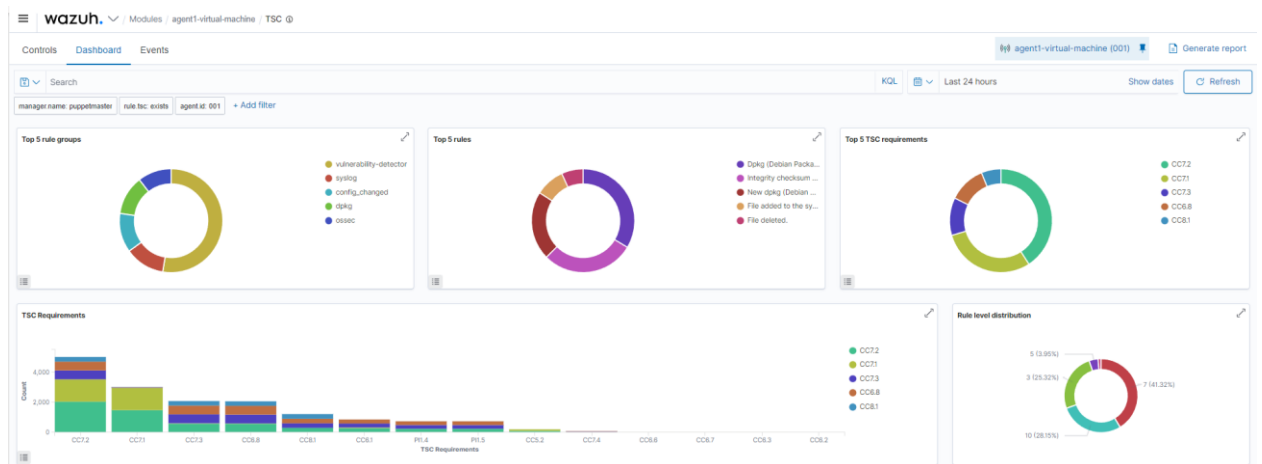


Рисунок 2.49. TSC

Контрольные вопросы

1. Принцип работы DLP-системы.
2. Шлюзовые DLP-системы
3. Агентские DLP-системы
4. Статистические методы анализа
5. Сетевой уровень контроля
6. Контроль на хостовом уровне

ЗАКЛЮЧЕНИЕ

Основным недостатком методов обнаружения вторжений является сложность, а зачастую невозможность создания правила (сигнатуры) атаки или шаблона поведения пользователя. Изменение характера атаки, делает невозможным ее обнаружение. Методы интеллектуального анализа решают эти проблемы: используя ранее накопленные знания или специально сгенерированный трафик, создание правил заметно упрощается, и потому СОВ демонстрирует высокую точность при обнаружении плохо описанных вторжений (сетевых атак).

Однако методы интеллектуального анализа достаточно сложно применить поскольку требуют высокой степени квалификации администраторов безопасности, а также постоянное сопровождение систем обучения и интеллектуального анализа СОВ разработчиками средств обнаружения вторжений. Поэтому большинство высокоинтеллектуальных систем обнаружения атак являются сетевыми сервисами или клиентско-серверными архитектурами, при этом на стороне разработчика находится сервер, который постоянно обновляет средства и базы интеллектуального анализа, поддерживая их в рабочем состоянии.

Показанные способы детектирования вредоносного трафика, устраняет указанные проблемы, позволяя обнаруживать и классифицировать не только уже известные атаки, но и определять их смежность друг с другом, что позволит администраторам безопасности еще быстрее и качественнее реагировать на любые типы атак, в том числе и сложные многокомпонентные типы атак, реализующиеся в длительном временном отрезке.

ЛИТЕРАТУРА

1. Bugcrowd's Bug Bounty List. URL: <https://bugcrowd.com/list-of-bug-bounty-programs>
2. HackerOne's Bug Bounty Programs. URL: <https://hackerone.com/programs>
3. Adobe Cross Domain Policy File Specification. URL: http://www.adobe.com/devnet/articles/crossdomain_policy_file_spec.htm
4. Google Hacking Database. URL: <http://www.exploitdb.com/google-dorks>
5. Saumil Shah. An Introduction to HTTP fingerprinting. URL: http://www.net-square.com/httpprint_paper.html
6. Qualys, Inc. TLS Renegotiation and Denial of Service Attacks. URL: <https://community.qualys.com/blogs/securitylabs/2011/10/31/tlsrenegotiation-and-denial-of-service-attacks>
7. <https://community.qualys.com/blogs/securitylabs/2011/10/31/tlsrenegotiation-and-denial-of-service-attacks>
8. Qualys SSL Labs. SSL/TLS Deployment Best Practices. URL: https://www.ssllabs.com/downloads/SSL_TLS_Deployment_Best_Practices_1.3.pdf
9. https://www.ssllabs.com/downloads/SSL_TLS_Deployment_Best_Practices_1.3.pdf
10. ЗАО Позитив Текнолоджис. В. Кочетков. Как разработать защищенное веб-приложение и не сойти при этом с ума». URL: <http://www.slideshare.net/kochetkov.vladimir/hdswasmwebinar>
11. И. Новиков. Завалить в один запрос: уязвимости вебприложение, приводящие к DoS. URL: <http://www.slideshare.net/d0znpp/ss-27695334>
12. <http://www.slideshare.net/d0znpp/ss-27695334>
13. Trustwave's SpiderLabs. Mitigating Slow HTTP DoS Attacks. URL: <http://blog.spiderlabs.com/2011/07/advanced-topic-of-theweek-mitigating-slow-http-dos-attacks.html>
14. Mitigation of 'Slow Read' Denial of Service Attack. URL: <http://blog.spiderlabs.com/2012/01/modsecurity-advanced-topicof-the-week-mitigation-of-slow-read-denial-of-service-attack.html>
15. Qualys Inc. Are you ready for slow reading? URL: <https://community.qualys.com/blogs/securitylabs/2012/01/05/slow-read>
16. Identifying Slow HTTP Attack Vulnerabilities. URL: <https://community.qualys.com/blogs/securitylabs/2012/01/05/slow-read>
17. CWE-352: Cross-Site Request Forgery (CSRF). URL: <http://cwe.mitre.org/data/definitions/352.html>
18. Barth A., Jackson C., and Mitchell J. Robust Defenses for CrossSite Request Forgery // Proc. 15th ACM Conference on Computer and Communications Security. – ACM Press, 2008. pp. 75-87.
19. Javed A. On Breaking PHP-based Cross-Site Scripting Protection Mechanisms. URL: <http://slides.com/mscashaarjaved/on-breakingphp-based-cross-site-scripting-protections-in-the-wild>
20. Д. Евтеев. SQL Injection от А до Я. URL: <http://www.ptsecurity.ru/download/PT-devteev-Advanced-SQLInjection.pdf>

21. Acunetix. Why File Upload Forms are a Major Security Threat. URL: <https://www.acunetix.com/websitesecurity/upload-formsthreat>.
22. SNORT как сервисная IPS. [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/post/123474/> (Дата обращения: 15.11.2020).
23. Система обнаружения вторжения для Чайников. Установка и Конфигурирование SNORT. [Электронный ресурс]. – Режим доступа: <https://www.securitylab.ru/analytics/216181.php> (Дата обращения: 10.11.2020).
24. OSSEC. [Электронный ресурс]. – Режим доступа: <https://www.ossec.net/> (Дата обращения: 12.11.2020).
25. Руководство пользователя. [Электронный ресурс]. – Режим доступа: <http://manual-snort-org.s3-website-us-east-1.amazonaws.com/> (Дата обращения: 14.11.2020).
26. Инструкция: внедрение HIDS. [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/post/262479/#agentconf> (Дата обращения: 15.11.2020).
27. OSSEC: Большой Брат наблюдает за тобой. [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/post/192800/> (Дата обращения: 13.11.2020).
28. Хостовая система обнаружения вторжений. [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Хостовая_система_обнаружения_вторжений (Дата обращения: 15.11.2020).
29. IPsec / URL: <https://ru.wikipedia.org/wiki/IPsec> (дата обращения
30. 22.11.2014).
31. Andrew Mason. IPSec Overview. – Cisco Press, 2002.
32. IPSecHowTo / URL: <https://help.ubuntu.com/community/IPSecHowTo> (дата обращения 22.11.2014).
33. Шаньгин В. Ф. Информационная безопасность компьютерных систем и сетей // М.: Форум, Инфра-М, 2008. – 416 с.
34. Benvenuti C. Understanding Linux Network Internals. // O'Reilly Media, 2006. – 1064 с.
35. Wazuh documentation – <https://documentation.wazuh.com/current/getting-started/index.html> – режим доступа: <https://wazuh.com/>
36. Внедряем Ossec – <https://habr.com/ru/post/262479/#config> – режим доступа: <https://habr.com/ru/>
37. Detecting Metasploit attacks – <https://wazuh.com/blog/detecting-metasploit-attacks/> – режим доступа: <https://wazuh.com/>
38. Пентест сетевого периметра с использованием Kali Linux – <https://defcon.ru/penetration-testing/11711/> – режим доступа: <https://defcon.ru/>
39. ELK, SIEM из OpenSource, Open Distro: Интеграция с WAZUH – <https://habr.com/ru/post/516332/> – режим доступа: <https://habr.com/ru/>
40. User manual – <https://documentation.wazuh.com/current/user-manual/index.html> – режим доступа: <https://wazuh.com/>

Локальный электронный методический материал

Владислав Владимирович Подтопельный

СИСТЕМЫ ЗАЩИТЫ
ОТ УТЕЧКИ КОНФИДЕНЦИАЛЬНОЙ ИНФОРМАЦИИ

Редактор С. Кондрашова
Корректор Т. Звада

Уч.-изд. л. 4,7. Печ. л. 3,8.

Федеральное государственное
бюджетное образовательное учреждение высшего образования
«Калининградский государственный технический университет»,
236022, Калининград, Советский проспект, 1