



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего
образования

РТУ МИРЭА

«МИРЭА – Российский технологический университет»

Институт кибербезопасности и цифровых технологий

Кафедра КБ-2 «Информационно-аналитические системы кибербезопасности»

Практическая работа № 6

«Матрица уязвимостей MITRE ATT&CK. Тактика: разведка. Техника: активное
сканирование»

по дисциплине «Безопасность операционных систем»

Москва

2023

1. Разведка

Разведка (Reconnaissance) набор техник сбора информации о цели, которые включают в себя активный или пассивный сбор информации:

- T1595 Активное сканирование;
- T1591 Сбор бизнес-информации об организации;
- T1597 Сбор информации из закрытых источников;
- T1593 Сбор информации из общедоступных источников;
- T1590 Сбор информации о сетевой инфраструктуре;
- T1589 Сбор информации об атакуемых пользователях;
- T1592 Сбор информации об атакуемых узлах;
- T1594 Сбор информации с сайта организации;
- T1596 Сбор технической информации из открытых источников;
- T1598 Фишинг с целью сбора сведений.

Отчет. Запишите техники в отчет и укажите к какому типу они относятся (пассивные или активные).

Для выполнения практической работы понадобятся виртуальные машины, которые мы настроили на прошлом практическом занятии:

1. Сервер SIEM-системы Wazuh
2. Kali с установленным wazuh-агентом
3. Windows с установленным wazuh-агентом

«Активное сканирование» направленное сканирование при помощи специализированного программного обеспечения подключенных к сети устройств с целью идентификации сетевых сервисов, типов и версий программного обеспечения этих сервисов, а также с целью получения конфигурационной информации компонентов систем и сетей, программного обеспечения сервисов и приложений.

Сканирование будем проводить утилитой nmap с ВМ Kali. Сканировать будем ВМ Windows. Сначала создайте сетевую папку (название папки Ваша

фамилия). Далее познакомьтесь с утилитой `nmap`, которая входит в состав ОС Kali Linux и проведите сканирование VM Windows используя параметр `-A`.

Отчет:

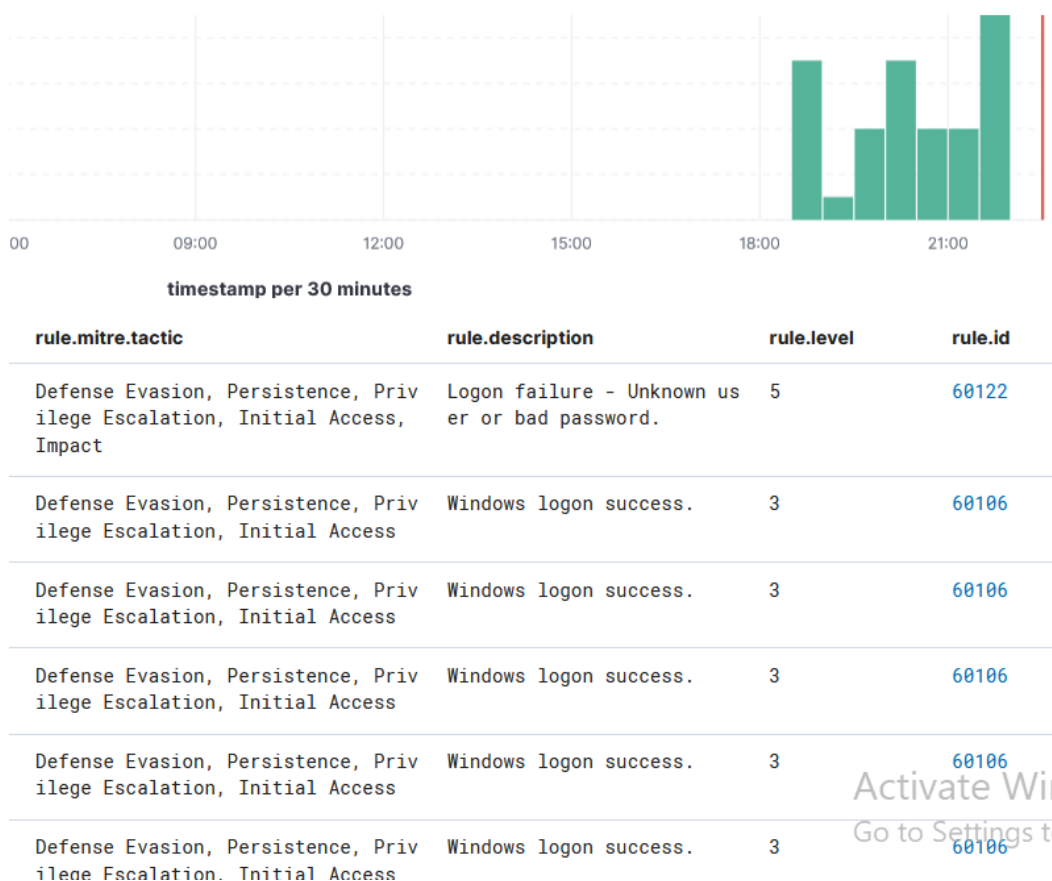
1. Покажите наличие созданной сетевой папки (команда *net share*).
2. Покажите сетевые настройки VM Kali и VM Windows.
3. Покажите наличие открытых портов на VM Windows (команда *netstat*).
4. Покажите результат сканирования в формате *xml*.

2. Обнаружение

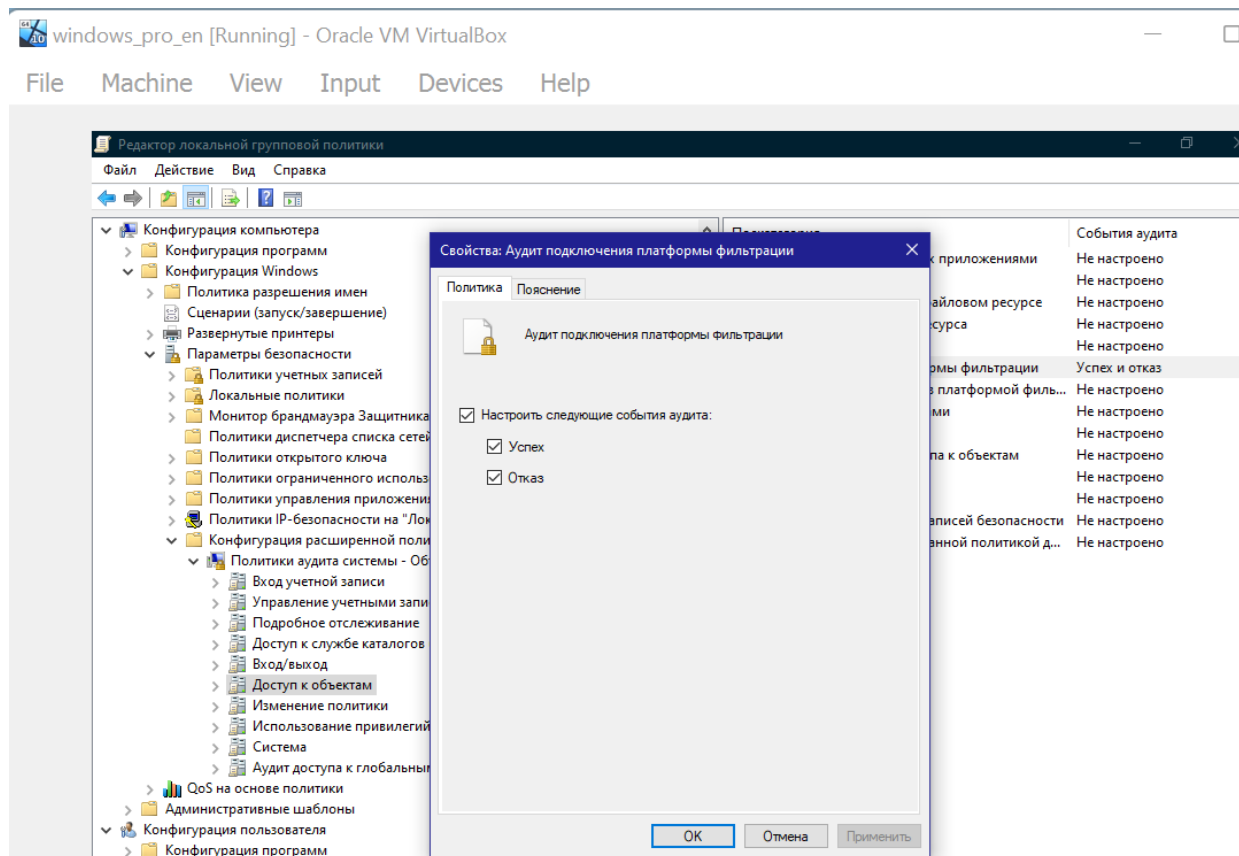
В прошлом задании мы выполнили «агрессивное» сканирование VM Windows, что является аномальным поведением в информационно-телекоммуникационных сетях и факт такого поведения должен рассматриваться как инцидент информационной безопасности. Правда сначала такое поведение необходимо обнаружить. Для обнаружения инцидентов информационной безопасности в сети применяются IDPS системы.

IDPS (Intrusion Detection and Prevention System) это системы обнаружения и предотвращения вторжения. По сути, IDPS мониторит транзитный и локальный трафик на попытки сканирования и атак, соотнося их с имеющимися сигнатурами. IDPS можно разделить на два класса NIDS (Network Intrusion Detection System) и HIDS (Host-based Intrusion Detection System). NIDS мониторят сетевой трафик. HIDS анализируют события хоста, в том числе входящий и исходящий трафик.

У нашего тестового стенда нет решений типа IDPS и наш SIEM не регистрирует данную технику. Убедитесь в этом изучив данные модуля MITRE ATT&CK на вкладке Events:



Теперь проверьте логи VM Windows, там нет событий, связанных с проведенным сканированием. Давайте включим их в оснастке управления локальными групповыми политиками, за эти события отвечает платформа фильтрации Windows (WFP). WFP подключения параметр политики обеспечивает аудит подключений, которые разрешаются или блокируются платформой фильтрации. Если этот параметр политики настроен, события аудита возникают при разрешении или блокировке подключений платформой WFP. При разрешении подключений возникают успешные события аудита, при блокировке неудачные.



Теперь проведите повторное сканирование и посмотрите, что попадет в логи VM Windows и в события безопасности wazuh-сервера.

Отчёт:

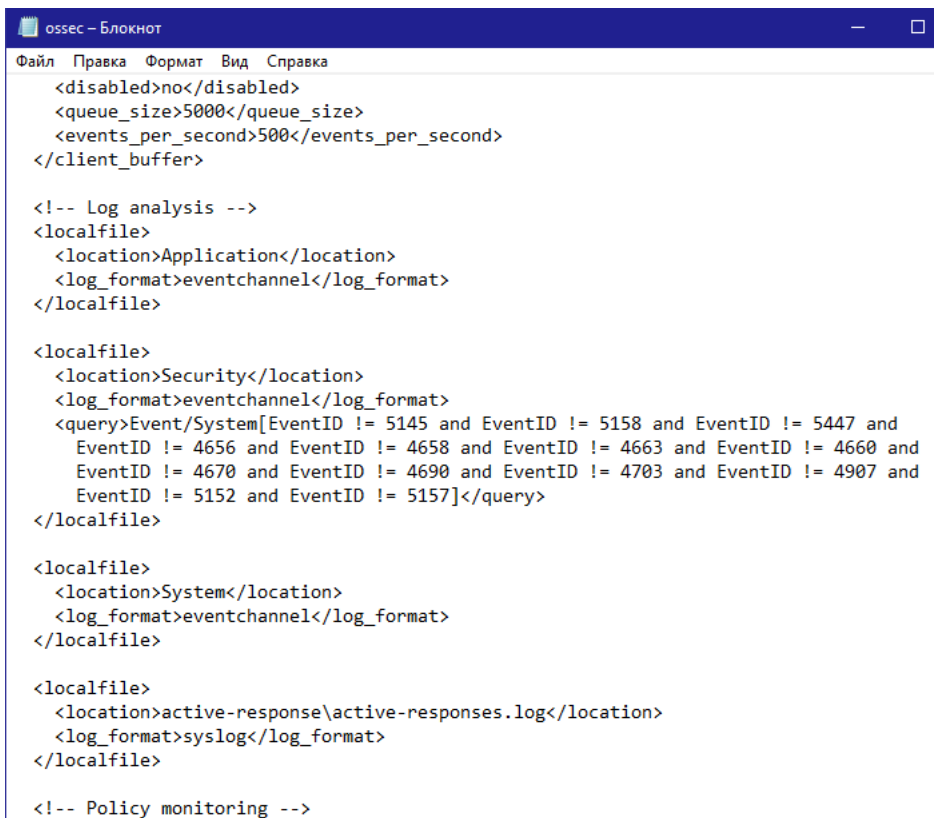
1. Запишите в отчет коды и краткое описание событий, которые попадают в логи VM Windows при попытке установить новое сетевое соединение, после включения аудита подключений WFP.

3. Подключение SIEM к обнаружению сканирования хостов в сети

Не все события, которые мы видим в логах VM Windows передаются на wazuh-сервер, это сделано для того чтобы не засорять сеть большим количеством передаваемых пакетов и не нагружать сервер обработки логов незначительными событиями. При этом в SIEM wazuh присутствует возможность гибкой настройки того за чем мы хотим наблюдать и что хотим контролировать. Приступим.

3.1 Конфигурационный файл агента на VM Windows

В конфигурационном файле агента нужно указать какие логи необходимо отправлять на сервер. Файл находится в директории *C:\Program Files (x86)\ossec-agent\ossec.conf*. Нам нужно убедиться, что логи из ветки Security настроены для отправки, они находятся в секции *<localfile>*, как показано на рисунке



```
ossec - Блокнот
Файл  Правка  Формат  Вид  Справка

<disabled>no</disabled>
<queue_size>5000</queue_size>
<events_per_second>500</events_per_second>
</client_buffer>

<!-- Log analysis -->
<localfile>
  <location>Application</location>
  <log_format>eventchannel</log_format>
</localfile>

<localfile>
  <location>Security</location>
  <log_format>eventchannel</log_format>
  <query>Event/System[EventID != 5145 and EventID != 5158 and EventID != 5447 and
    EventID != 4656 and EventID != 4658 and EventID != 4663 and EventID != 4660 and
    EventID != 4670 and EventID != 4690 and EventID != 4703 and EventID != 4907 and
    EventID != 5152 and EventID != 5157]</query>
</localfile>

<localfile>
  <location>System</location>
  <log_format>eventchannel</log_format>
</localfile>

<localfile>
  <location>active-response\active-responses.log</location>
  <log_format>syslog</log_format>
</localfile>

<!-- Policy monitoring -->
```

Важно! Обратите внимание, что по-умолчанию код события, которое нам необходимо отслеживать добавлен в исключения! Значит нужно поправить конфиг и перезагрузить агента командой *net stop wazuh; net start wazuh*.

3.2 Создание правила для обработки события на wazuh-сервере

Логи попадая на сервер обрабатываются согласно специальным правилам. Давайте проверим есть ли правило под наше событие. Правила расположены в директории `/var/ossec/ruleset/rules/`, выполним команду для поиска кода события в директории с правилами:

```
grep -Ril "5156" /var/ossec/ruleset/rules/
```

Правило для «нашего» события отсутствует, будем создавать своё. Пользовательские правила должны добавляться в файл `/var/ossec/etc/rules/local_rules.xml`. Перед внесением изменений создайте резервную копию файла. Теперь нам необходимо добавить правило, которое обрабатывает успешное событие безопасности Windows с кодом 5156. Посмотрим какие подходящие правила уже есть в папке `/var/ossec/ruleset/rules/`. Под наши требования подходит файл `0580-win-security_rules.xml`. Посмотрим что внутри

```
GNU nano 2.9.8 /var/ossec/ruleset/rules/0580-win-security_ru
<!--
  Copyright (C) 2015, Wazuh Inc.
-->

<!--
  Rules for:
    Windows security ID: 60100 - 60599
-->

<var name="MS_FREQ">8</var>

<group name="windows,windows_security,">

  <rule id="60100" level="0">
    <if_sid>60001</if_sid>
    <field name="win.system.severityValue">^INFORMATION$</field>
    <options>no_full_log</options>
    <description>Windows security informational event.</description>
    <mitre>
      <id>T1005</id>
    </mitre>
  </rule>

  <rule id="60101" level="0">
    <if_sid>60001</if_sid>
    <field name="win.system.severityValue">^WARNING$</field>
    <options>no_full_log</options>
    <description>Windows security warning event.</description>
    <mitre>
      <id>T1005</id>
    </mitre>
    <group>gpg13_4.12,</group>
  </rule>

  <rule id="60102" level="5">
    <if_sid>60001</if_sid>
    <field name="win.system.severityValue">^ERROR$</field>
    <options>no_full_log</options>
    <description>Windows security error event.</description>
    <mitre>
      <id>T1005</id>
    </mitre>
    <group>gdpr_IV_35.7.d,gpg13_4.3,system_error,</group>
```

```

GNU nano 2.9.8 /var/ossec/ruleset/rules/0580-win-security

<rule id="60101" level="0">
  <if_sid>60001</if_sid>
  <field name="win.system.severityValue">^WARNING$</field>
  <options>no_full_log</options>
  <description>Windows security warning event.</description>
  <mitre>
    <id>T1005</id>
  </mitre>
  <group>gpg13_4.12,</group>
</rule>

<rule id="60102" level="5">
  <if_sid>60001</if_sid>
  <field name="win.system.severityValue">^ERROR$</field>
  <options>no_full_log</options>
  <description>Windows security error event.</description>
  <mitre>
    <id>T1005</id>
  </mitre>
  <group>gdpr_IV_35.7.d,gpg13_4.3,system_error,</group>
</rule>

<rule id="60103" level="0">
  <if_sid>60001</if_sid>
  <field name="win.system.severityValue">^AUDIT_SUCCESS$|^success$</field>
  <options>no_full_log</options>
  <description>Windows audit success event.</description>
</rule>

<rule id="60104" level="5">
  <if_sid>60001</if_sid>
  <field name="win.system.severityValue">^AUDIT_FAILURE$|^failure$</field>
  <options>no_full_log</options>
  <description>Windows audit failure event.</description>
  <group>gdpr_IV_35.7.d,hipaa_164.312.b,nist_800_53_AU.6,pci_dss_
</rule>

<rule id="60105" level="5">
  <if_sid>60104</if_sid>
  <field name="win.system.eventID">^529$|^530$|^531$|^532$|^533$|^
  <options>no_full_log</options>
  <description>Windows logon failure.</description>

```

Как видим правила имеют вложенную структуру, это значит, что есть родительские и дочерние правила, найдем родительское правило под наши требования: *ID=60103*. Еще нам понадобится указать группу для создаваемого правила, возьмём такую же как в файле *0580-win-security_rules.xml*. Мы готовы создать свое правило. Оно будет выглядеть так:

```

<group name="windows,windows_security,">

  <rule id="100100" level="3">
    <if_sid>60103</if_sid>
    <field name="win.system.eventID">^5156$</field>
    <options>no_full_log</options>
    <description>Windows filtering platform packet OPEN.</description>
  </rule>

</group>
[root@wazuh-server ossec]#

```


Где id=100100 это id нашего правила, которое обязательно должно быть больше 100000.

Level уровень важности события.

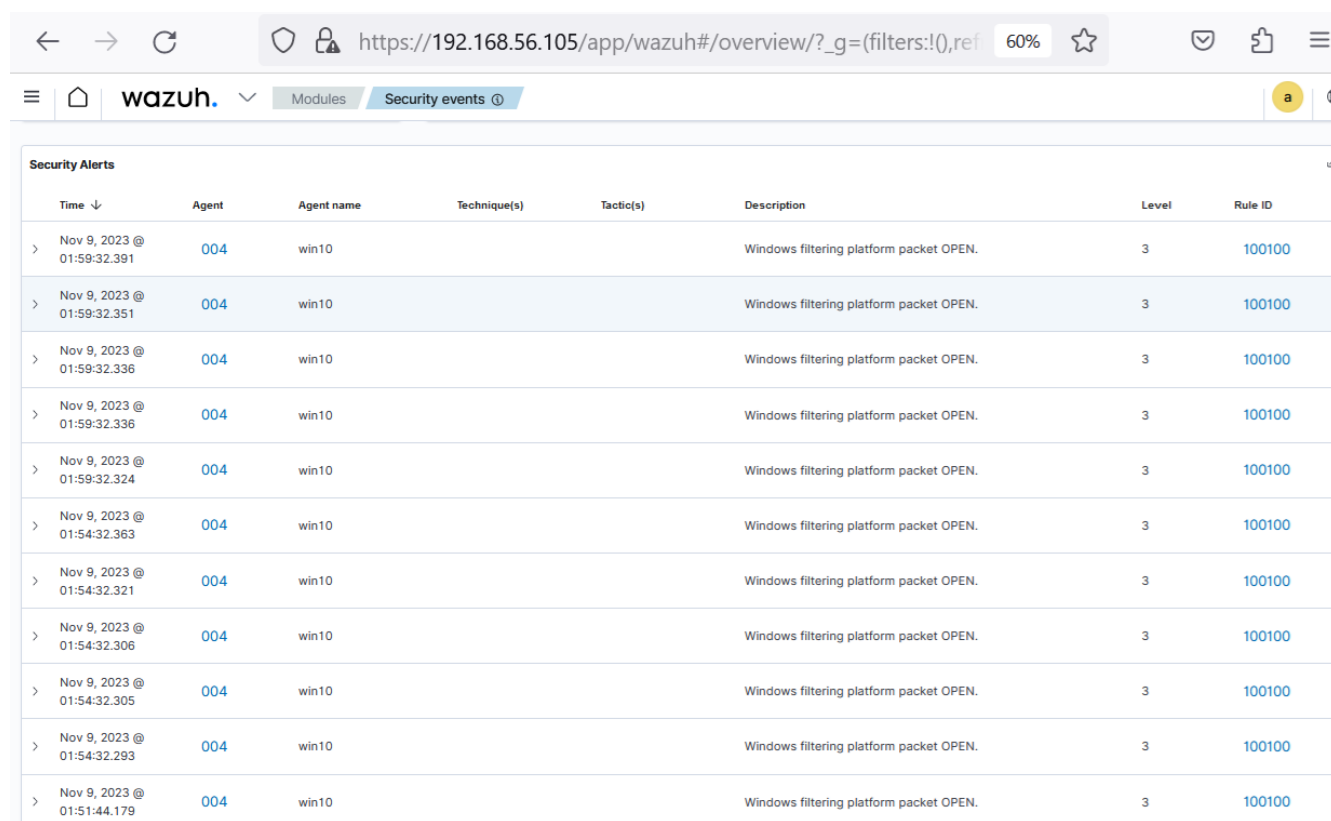
If_sid id родительского правила, которое обработало лог по своим параметрам.

Field параметр фильтрации логов, в нашем случае по коду события.

Options дополнительные параметры.

Description описание, которое будет отображаться на «дашборде».

Сохраните файл и перезапустите wazuh-manager. Повторите сканирование и найдите на «дашборде» wazuh-сервера «наше» событие.



The screenshot shows the Wazuh Security Alerts dashboard. The browser address bar displays the URL: [https://192.168.56.105/app/wazuh#/overview/?_g=\(filters:!\(\),ref](https://192.168.56.105/app/wazuh#/overview/?_g=(filters:!(),ref). The dashboard header includes the Wazuh logo, a dropdown menu, and tabs for 'Modules' and 'Security events'. The main content area is titled 'Security Alerts' and contains a table with the following columns: Time, Agent, Agent name, Technique(s), Tactic(s), Description, Level, and Rule ID. The table lists 11 alerts, all with a Level of 3 and Rule ID of 100100. The description for all alerts is 'Windows filtering platform packet OPEN.'.

Time ↓	Agent	Agent name	Technique(s)	Tactic(s)	Description	Level	Rule ID
> Nov 9, 2023 @ 01:59:32.391	004	win10			Windows filtering platform packet OPEN.	3	100100
> Nov 9, 2023 @ 01:59:32.351	004	win10			Windows filtering platform packet OPEN.	3	100100
> Nov 9, 2023 @ 01:59:32.336	004	win10			Windows filtering platform packet OPEN.	3	100100
> Nov 9, 2023 @ 01:59:32.336	004	win10			Windows filtering platform packet OPEN.	3	100100
> Nov 9, 2023 @ 01:59:32.324	004	win10			Windows filtering platform packet OPEN.	3	100100
> Nov 9, 2023 @ 01:54:32.363	004	win10			Windows filtering platform packet OPEN.	3	100100
> Nov 9, 2023 @ 01:54:32.321	004	win10			Windows filtering platform packet OPEN.	3	100100
> Nov 9, 2023 @ 01:54:32.306	004	win10			Windows filtering platform packet OPEN.	3	100100
> Nov 9, 2023 @ 01:54:32.305	004	win10			Windows filtering platform packet OPEN.	3	100100
> Nov 9, 2023 @ 01:54:32.293	004	win10			Windows filtering platform packet OPEN.	3	100100
> Nov 9, 2023 @ 01:51:44.179	004	win10			Windows filtering platform packet OPEN.	3	100100

У меня вся доска забита «нашими» событиями, причем даже если не проводить сканирование. Давайте добавим еще одно правило, которое будет дочерним по отношению к первому правилу и повысит чувствительность события.

Откроем файл `/var/ossec/etc/rules/local_rules.xml` и добавим правило

```

<group name="windows,windows_security,">

  <rule id="100100" level="3">
    <if_sid>60103</if_sid>
    <field name="win.system.eventID">^5156$</field>
    <options>no_full_log</options>
    <description>Windows filtering platform packet OPEN.</description>
  </rule>

  <rule id="100101" level="10" frequency="10" timeframe="1">
    <if_matched_sid>100100</if_matched_sid>
    <description>Possible scan detected - OPEN.</description>
    <mitre>
      <id>T1595.001</id>
    </mitre>
  </rule>
</group>
[root@wazuh-server ossec]#

```

Где:

If_matched_sid id правила, для которого мы создаем условия frequency и timeframe (оповещение наступит если правило 100100 наступит 10 раз за 1 секунду)

Mitre тег для события по классификации Mitre ATT&CK (номер техники)

Сохраним файл и перезапустим wazuh-manager. Проведите сканирование и наблюдайте за результатами на «дашборде» сервера. (есть удобная настройка обновления «дашборда» через заданное количество времени, например, каждые 30 секунд). Если для вашего стенда нужны другие настройки правил, произведите их. Теперь мы практически избавились от ложных срабатываний оповещения о сканировании на «дашборде» Mitre. Создадим правило, которое будет с высокой долей вероятности относиться именно к санированию, и выставим ему критический уровень (от 12 и выше). Править будем тот же файл:

```

<!-- Copyright (C) 2015, Wazuh Inc. -->

<!-- Example -->
<group name="local,syslog,sshd,">

  <!--
  Dec 10 01:02:02 host sshd[1234]: Failed none for root from 1.1.1.1 port 1066 ssh2
  -->
  <rule id="100001" level="5">
    <if_sid>5716</if_sid>
    <srcip>1.1.1.1</srcip>
    <description>sshd: authentication failed from IP 1.1.1.1.</description>
    <group>authentication_failed,pci_dss_10.2.4,pci_dss_10.2.5,</group>
  </rule>

</group>

<group name="windows,windows_security,">

  <rule id="100100" level="3">
    <if_sid>60103</if_sid>
    <field name="win.system.eventID">^5156$</field>
    <options>no_full_log</options>
    <description>Windows filtering platform packet OPEN.</description>
  </rule>

  <rule id="100101" level="10" frequency="10" timeframe="1">
    <if_matched_sid>100100</if_matched_sid>
    <description>Possible scan detected - OPEN.</description>
    <mitre>
      <id>T1595.001</id>
    </mitre>
  </rule>

  <rule id="100102" level="12" frequency="5" timeframe="60">
    <if_matched_sid>100101</if_matched_sid>
    <description>Critical! Scan detected - OPEN.</description>
    <mitre>
      <id>T1595.001</id>
      <id>T1595.002</id>
    </mitre>
  </rule>

</group>
[root@wazuh-server ossec]#

```

Сохраним файл и перезапустим wazuh-manager. Проведите сканирование и наблюдайте за результатами на «дашборде» сервера.

Отчёт:

1. Приложите скриншот «дашборда» Mitre.
2. Включите аудит блокировки WFP. Настройте отправку логов аудита блокировки WFP на wazuh-сервер. Создайте правила для отображения оповещений о сканировании закрытых портов.
3. Добейтесь низкого уровня ложных оповещений на «дашборде» Mitre