Задание для инженера-аналитика (v2)

Практическое задание является вторым этапом подбора кандидата на позицию
инженера-аналитика в Отдел Исследований и Аналитики (ОИА).

Задание

Информация по уязвимостям для Linux систем обычно поставляется от вендора в
формате OVAL.
В рамках данной части задания необходимо:

1. Провести частичный анализ OVAL файла от компании RHEL (https://www.redhat.com/security/data/oval/v2/RHEL8/rhel-8.oval.xml.bz2) на первых 3 уязвимостях (патчах). Определить набор объектов, из которых он строится. Понять основную логику "работы" данного формата.

   Файл содержит описание уязвимостей и критерии их выявления. Критерии содержат описание и ссылку на тест. Описания тестов также находятся в этом файле

2. Описать текстом объекты, которые были найдены и для чего они используются. (Не более 2-3 фраз по каждому объекту).

   - **oval_definitions:** этот элемент указывает, что документ соответствует схемам OVAL для определений, Unix, Linux и независимых платформ. Секция содержит схемы и пространства имен
   - **generator:** секция генератора содержит информацию о продукте, который сгенерировал этот файл, его версии, версии схемы и времени генерации файла.
   - **definitions:** Эта секция содержит определения, которые представляют информацию об уязвимостях(патчах) и критерии их выявления. Критерии содержат описание и ссылку на тест.
   - **tests:** секция содержит тесты для проверки критериев. С указанием проверяемого параметра и его эталонного значения

3. В рамках каждого определения уязвимости, есть критерии по ее выявлению: какие из критериев на ваш взгляд лишние?

   Избыточным показался критерий 'Red Hat Enterprise Linux must be installed' т.к. я выборочно проверил 5-7 уязвимостей и там дальше все равно были критерии типа: 'Red Hat Enterprise Linux 8 is installed' или 'Red Hat CoreOS 4 is installed', которые как я предполагаю уже обеспечивают выполнение этого критерия.

4. Предложить и кратко описать свой вариант по упрощению формата для описания уязвимости вместе с проверками.

   Я решил использовать в простом варианте 5 колонок:
   1. **Title:** краткое описание уязвимости(патча)
   2. **Severity:** уровень серьезности/критичности (например, низкий, умеренный, высокий)
   3. **Description:** описание, чтобы можно было узнать подробнее об уязвимости
   4. **CVEs:** список идентификаторов уязвимостей, которые относятся к этой уязвимости
   5. **Criteria:** условия, при которых уязвимость актуальна (например, специфичные версии ПО).

   Таким образом, можно будет составить общее представление об информации, которая содержится в файле, количестве наиболее критичных уязвимостей. Будет возможен поиск по конкретным CVE и версиям ПО.

5. После выполненного в предыдущей пунктах анализа, необходимо разработать приложение на языке Python, которое произведет разбор (парсинг) OVAL-файла (достаточно сделать только первые 3 и связанными с ними объекты) и преобразует его в упрощенный формат.

   Для демонстрации результатов парсинга я решил использовать сохранение результата в файл **.xlsx**.
   https://github.com/Dies013/OVALparser_for_test/
   - **OVALparser.py** - код
   - **vulnerabilities.xlsx** – результат парсинга
   - **tags.xlsx** – тэги, который извлекал для исследования файла, т.к ранее не работал с этим форматом, понадобилось предварительное изучение.