БДЗ №1. Добавление модуля анализа промышленного протокола в ПО Suricata

Suricata — сетевое средство обнаружения и предотвращения вторжений, имеющее открытый исходный код [1]. Suricata позволяет анализировать множество сетевых протоколов, однако очень малое число промышленных протоколов (например, Modbus [2]). Suricata позволяет добавлять собственноручные модули для расширения списка анализируемых протоколов, для этих целей существуют скрипты в папке /scripts:

- setup-app-layer.py — создает файлы (логгер, парсер, детектер), которые необходимо заполнить;
- setup-simple-detect.sh — создает файл детектера, который необходимо заполнить.

Также есть и другие способы добавить свой модуль для нового протокола.

Пример установки Suricata и добавления собственных модулей (ОС Ubuntu):

1. sudo su
2. apt-get update && apt-get upgrade -y
3. sudo apt-get -y git install libpcre3 libpcre3-dbg libpcre3-dev build-essential autoconf automake libtool libpcap-dev libnet1-dev libyaml-0-2 libyaml-dev zlib1g zlib1g-dev libcap-ng-dev libcap-ng0 libmagic-dev libjansson-dev libjansson4 pkg-config automake-1.15 cargo rustc
4. cargo install --force cbindgen
5. export PATH=$PATH:$HOME/.cargo/bin
6. git clone https://github.com/OISF/suricata.git
7. cd suricata
8. git submodule update --init
9. git clone https://github.com/OISF/libhtp
10. ./autogen.sh
11. ./configure --prefix=/usr --sysconfdir=/etc --localstatedir=/var
12. python3 scripts/setup-app-layer.py --detect --logger --parser *Myproto Myprotobuf*
13. make -j$(nproc)

где вместо myproto — имя протокола, для которого пишется модуль.

Далее, для протокола необходимо заполнить sigmatch_table, и описать функции Match, Parse, Setup, Free. В функции парсинга описывается разбор полей пакета протокола. Далее расширяется регулярное выражение для необходимых полей. Наконец, создается файл *.rules, где непосредственно описываются необходимые правила для Suricata.

За основу того, как заполнять указанные выше функции, можно взять примеры уже созданных модулей Suricata для различных протоколов.

Задание.

Все практические части задания необходимо делать на виртуальной машине, чтобы по итогу БДЗ был готовый образ со всем необходимым установленным ПО.

1. Изучить промышленный протокол. Описать его основные особенности. Описать структуру пакета.
2. Построить структуру клиент-сервер для исследуемого протокола. Предпочтительнее (и проще) воспользоваться уже готовыми решениями. Однако допускается разработка собственной структуры клиент-сервера (в таком случае предпочтительный язык разработки — Python). Проверить корректность пересылаемого трафика можно с помощью ПО Wireshark.
3. Разработать модуль для протокола для ПО Suricata, который будет разбирать поля пакета протокола и иметь возможность детектировать по указанным полям.
4. Написать правила для Suricata (не менее десяти различных), демонстрирующие возможность модуля парсить и детектировать протокол.
5. Написать отчет по проделанной работе, в который будет входить:
    a. Описание исследуемого протокола.
    b. Описание построенной сетевой структуры (в том числе инструкцию по установке и настройке сетевого взаимодействия).
    c. Описание созданных правил для Suricata.

Ссылки на источники информации

1. https://github.com/OISF/suricata
2. https://github.com/OISF/suricata/blob/master/src/app-layer-modbus.c
3. https://suricata.readthedocs.io/en/suricata-6.0.0/