# Лабораторная работа № 3 ПАСОИБ – весна 2021 г.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Фамилия И.О.*** | ***Гасанов Э.А.*** | | Группа | ***4851003/70801*** |
| Вариант 3.  Лабораторная выполнялась на OS Kali Linux x64*,* где была установлена Snort 2.8.15*.* Адреса и порты исследуемых веб-серверов приведены в таблице 1.  Таблица 1 – Адреса веб-серверов и их порты.   |  |  |  |  | | --- | --- | --- | --- | | Расположение | Веб-сервер | IP-адрес | Порт | | Главный офис | Main (M) | 192.168.0.1 | 80 | | Филиал №1 | dep1 (F1) | 192.168.10.1 | 80 | | Филиал №2 | dep2 (F2) | 192.168.20.1 | 8080 |   Описание назначения страниц на веб-серверах представлено в таблице 2.  Таблица 2 – Описание назначения страниц на веб-серверах.   |  |  |  | | --- | --- | --- | | Описание | Сокращение | Адрес | | Доверенные пользователи главного офиса | M trusted | 192.168.1.0/25 | | Рядовые пользователи главного офиса | M regular | 192.168.2.0/24 | | Доверенные пользователи филиала №1 | F1 trusted | 192.168.11.0/30 | | Рядовые пользователи филиала №1 | F1 regular | 192.168.12.0/28 | | Рядовые пользователи филиала №2 | F2 regular | 192.168.22.0/25 | | Внешние пользователи | Other | Все остальные |   Информация о субъектах, обращающихся к страницам, приведена в таблице 3.  Таблица 3 – Субъекты, представленные в виде сетевых хостов, обращающихся к страницам веб-серверов.   |  |  |  | | --- | --- | --- | | Описание | Сокращение | Адрес | | Доверенные пользователи главного офиса | M trusted | 192.168.1.0/25 | | Рядовые пользователи главного офиса | M regular | 192.168.2.0/24 | | Доверенные пользователи филиала №1 | F1 trusted | 192.168.11.0/30 | | Рядовые пользователи филиала №1 | F1 regular | 192.168.12.0/28 | | Рядовые пользователи филиала №2 | F2 regular | 192.168.22.0/25 | | Внешние пользователи | Other | Все остальные |   В таблице 4 красным цветом выделены ячейки, для которых согласно варианту необходимо создать правила, чтобы обнаружить атаки.  Таблица 4 – Информация о наличии прав доступа субъектов к объектам.   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | Субъект  Объект | M trusted | M regular | F1 trusted | F1 regular | F2 regular | Other | | M/confidential/\* | V | – | *1,3,11* | – | – | – | | M/internal/\* | V | V | *4,7,12* | *1,8,9* | *2,5,10* | – | | M/public/\* | V | V | V | V | V | V | | F1/confidential/\* | *3,5,6* | – | V | – | – | – | | F1/internal/\* | *2,9,11* | *4,6,10* | V | V | *7,8,12* | – | | F1/public/\* | V | V | V | V | V | V | | F2/internal/\* | *1,7,9* | *2,3,11* | *5,8,10* | *4,6,12* | V | – | | F2/public/\* | V | V | V | V | V | V | | | | | |
| ***Подпись студента*** | |  | ***Дата сдачи*** | ***10.04.2021*** |

***При необходимости использовать обратную сторону листа.***

Для конфигурации Snort использовался стандартный конфигурационный файл /usr/local/etc/snort/snort.conf. В этом файле указан препроцессор http\_inspect, одним из предназначений которого является нормализация URI. Это означает, что для обнаружения атак достаточно писать простые правила уже для нормализованных URI.

С помощью параметра http\_uri для content указывается, что совпадение с шаблоном необходимо искать только в URI. Параметр глубины depth необходим, чтобы искать совпадения только в первых N байтах URI. Для того чтобы установить свои правила, необходимо добавить в RULE\_PATH (/etc/snort/rules) свой файл с правилами. А затем подключить его с помощью директивы include в конфигурационном файле.

Далее представлен список написанных правил.

ipvar M\_IP 192.168.0.1

ipvar F1\_IP 192.168.10.1

ipvar F2\_IP 192.168.20.1

ipvar M\_TRUSTED\_NET 192.168.1.0/25

ipvar M\_REGULAR\_NET 192.168.2.0/24

ipvar F1\_TRUSTED\_NET 192.168.11.0/30

ipvar F1\_REGULAR\_NET 192.168.12.0/28

ipvar F2\_REGULAR\_NET 192.168.22.0/25

ipvar OTHER\_NET ![$M\_TRUSTED\_NET,$M\_REGULAR\_NET,$F1\_TRUSTED\_NET,$F1\_REGULAR\_NET,$F2\_REGULAR\_NET]

portvar M\_HTTP\_PORT 80

portvar F1\_HTTP\_PORT 80

portvar F2\_HTTP\_PORT 8080

######################################################################################################################################################################################################################

**Внешние к конфиденциальным**

# OTHER ->: /**confidential**/\*:

alert tcp $OTHER\_NET any -> $M\_IP $M\_HTTP\_PORT (content:"/confidential/"; http\_uri; depth:14; sid:1000016; rev:1; msg:"OTHER -> **M/confidential/\***";)

alert tcp $OTHER\_NET any -> $F1\_IP $F1\_HTTP\_PORT (content:"/confidential/"; http\_uri; depth:14; sid:1000018; rev:1; msg:"OTHER -> **F1/confidential/\***";)

############################################################################################################################################

**Рядовые к конфиденциальным**

# 2) REGULAR -> /**confidential**/\*:

alert tcp $M\_REGULAR\_NET any -> $M\_IP $M\_HTTP\_PORT (content:"/confidential/"; http\_uri; depth:14; sid:1000003; rev:1; msg:"M REGULAR -> **M/confidential/\***";)

alert tcp $M\_REGULAR\_NET any -> $F1\_IP $F1\_HTTP\_PORT (content:"/confidential/"; http\_uri; depth:14; sid:1000004; rev:1; msg:"M REGULAR -> **F1/confidential/\***";)

alert tcp $F1\_REGULAR\_NET any -> $M\_IP $M\_HTTP\_PORT (content:"/confidential/"; http\_uri; depth:14; sid:1000008; rev:1; msg:"F1 REGULAR -> **M/confidential/\***";)

alert tcp $F1\_REGULAR\_NET any -> $F1\_IP $F1\_HTTP\_PORT (content:"/confidential/"; http\_uri; depth:14; sid:1000010; rev:1; msg:"F1 REGULAR -> **F1/confidential/\***";)

alert tcp $F2\_REGULAR\_NET any -> $M\_IP $M\_HTTP\_PORT (content:"/confidential/"; http\_uri; depth:14; sid:1000012; rev:1; msg:"F2 REGULAR -> **M/confidential/\***";)

alert tcp $F2\_REGULAR\_NET any -> $F1\_IP $F1\_HTTP\_PORT (content:"/confidential/"; http\_uri; depth:14; sid:1000014; rev:1; msg:"F2 REGULAR -> **F1/confidential/\***";)

############################################################################################################################################

Доверенные к чужим конфиденциальным

**Исходя из матрицы доступа, правил, отвечающих на этот тип атаки не должно быть создано, поэтому и в таблице в этой графе 0.**

############################################################################################################################################

**Внешние к ДСП**

# OTHER -> /internal/\*:

alert tcp $OTHER\_NET any -> $M\_IP $M\_HTTP\_PORT (content:"/internal/"; http\_uri; depth:10; sid:1000017; rev:1; msg:"OTHER -> **M/internal/\***";)

alert tcp $OTHER\_NET any -> $F1\_IP $F1\_HTTP\_PORT (content:"/internal/"; http\_uri; depth:10; sid:1000019; rev:1; msg:"OTHER -> **F1/internal/\***";)

alert tcp $OTHER\_NET any -> $F2\_IP $F2\_HTTP\_PORT (content:"/internal/"; http\_uri; depth:10; sid:1000020; rev:1; msg:"OTHER -> **F2/internal/\***";)

############################################################################################################################################

**Рядовые к чужим ДСП**

# 5) REGULAR -> another /internal/\*:

alert tcp $M\_REGULAR\_NET any -> $F1\_IP $F1\_HTTP\_PORT (content:"/internal/"; http\_uri; depth:10; sid:1000005; rev:1; msg:"M REGULAR -> **F1/internal/\***";)

alert tcp $F1\_REGULAR\_NET any -> $M\_IP $M\_HTTP\_PORT (content:"/internal/"; http\_uri; depth:10; sid:1000009; rev:1; msg:"F1 REGULAR -> M/internal/\*";)

alert tcp $F1\_REGULAR\_NET any -> $F2\_IP $F2\_HTTP\_PORT (content:"/internal/"; http\_uri; depth:10; sid:1000011; rev:1; msg:"F1 REGULAR -> **F2/internal/\***";)

alert tcp $F2\_REGULAR\_NET any -> $M\_IP $M\_HTTP\_PORT (content:"/internal/"; http\_uri; depth:10; sid:1000013; rev:1; msg:"F2 REGULAR -> **M/internal/\***";)

alert tcp $F2\_REGULAR\_NET any -> $F1\_IP $F1\_HTTP\_PORT (content:"/internal/"; http\_uri; depth:10; sid:1000015; rev:1; msg:"F2 REGULAR -> **F1/internal/\***";)

############################################################################################################################################

**Доверенные к чужим ДСП**

# 1) M TRUSTED -> another /internal/\*:

alert tcp $M\_TRUSTED\_NET any -> $F1\_IP $F1\_HTTP\_PORT (content:"/internal/"; http\_uri; depth:10; sid:1000001; rev:1; msg:"M TRUSTED -> **F1/internal/\***";)

alert tcp $M\_TRUSTED\_NET any -> $F2\_IP $F2\_HTTP\_PORT (content:"/internal/"; http\_uri; depth:10; sid:1000002; rev:1; msg:"M TRUSTED -> **F2/internal/\***";)

alert tcp $F1\_TRUSTED\_NET any -> $M\_IP $M\_HTTP\_PORT (content:"/internal/"; http\_uri; depth:10; sid:1000006; rev:1; msg:"F1 TRUSTED -> **M/internal/\***";)

alert tcp $F1\_TRUSTED\_NET any -> $F2\_IP $F2\_HTTP\_PORT (content:"/internal/"; http\_uri; depth:10; sid:1000007; rev:1; msg:"F1 TRUSTED -> **F2/internal/\***";)

Количественные показатели обнаружения были посчитаны и представлены в таблице 5.

Таблица 5 – Количественные показатели обнаружения.

|  |  |
| --- | --- |
| **Тип атаки** | **Количество атак** |
| Внешние к конфиденциальным | 10 |
| Рядовые к конфиденциальным | 45 |
| Доверенные к чужим конфиденциальным | 0 |
| Внешние к ДСП | 17 |
| Рядовые к чужим ДСП | 17 |
| Доверенные к чужим ДСП | 9 |
| Всего атак | 98 |

**Контрольные вопросы**

1. Система обнаружения вторжений (СОВ) — комплекс (аппаратура и программное обеспечение), который по результатам анализа контролируемых и собираемых данных принимает решение о наличии атаки или вторжения.
2. Основные особенности СОВ Snort:

* Выявление вредоносного трафика.
* Выявление эксплоитов.
* Выявление атак на сетевые службы (Telnet, FTP, DNS, и т.д.).
* Выявление атак DoS/DDoS и переполнения буфера
* Выявление атак на базы данных.
* Возможность написания собственных правил.
* Расширение функциональности, используя возможность подключения модулей.
* Гибкая система оповещения об атаках (Log файлы, устройства вывода, БД и т.д.).

1. Последовательность действий:

* Закачка и установка с помощью менеджера пакетов.
* Внесение необходимых изменений в файл snort.conf, например, раскомментировать конфигурацию препроцессора, установка $HOME\_NET переменной
* Конфигурирование динамически подгружаемых библиотек.
* Конфигурирование плагинов вывода данных.
* Настройка своего набора правил и установка их в RULE\_PATH
* Включение директивой include собственных правил в snort.conf
* Настройка множества совместных правил.

Достоинства:

* Возможность использования встроенных правил;
* Возможность написания собственных правил;
* Многоплатформенность.
* Возможность включения собственных модулей в СОВ
* Возможность изменять конфигурацию СОВ для всех этапов обработки пакета
* ПО с открытым исходным кодом
* Возможность вывода логов в формате, читаемом tcpdump

Недостатки:

* Ложные срабатывания
* Сложность написания конфигурационного файла
* Отсутствие графического интерфейса
* Из-за вычислительной сложности используются наиболее простые алгоритмы анализа сетевых пакетов

1. СОВ считается необходимым дополнением инфраструктуры сетевой безопасности. В дополнение к межсетевым экранам, работа которых происходит на основе политики безопасности, СОВ служат механизмами мониторинга и наблюдения подозрительной активности. Они могут обнаружить атакующих, которые обошли МЭ, и выдать отчет об этом администратору, который, в свою очередь, предпримет дальнейшие шаги по предотвращению атаки. Технологии обнаружения проникновений не делают систему абсолютно безопасной. Snort же в свою очередь – облегченная система обнаружения вторжения. Snort обычно называют “обгегченным” NIDS, так как он разработан прежде всего для маленьких сетей.

**Вывод**

В ходе выполнения данной лабораторной работы была изучена концепция сетевой СОВ на примере Snort, а также рассмотрена ситуация обнаружения атак на систему электронного документооборота с помощью создания собственных правил для СОВ. Таким образом, мы научились конфигурировать SNORT и разрабатывать правила для обнаружения различных атак.