План лекции:

1 Введение.

2 Конфигурация Snort.

3 Начало работы.

4 Правила.

5 Интеграция Snort с iptables

Литература по лекции

0 Общие замечания

- При установке аккуратно ввести имя интерфейса (в моём случае enp0s3)

- После установки проверить командой snort -V

- При работе использовать sudo snort …

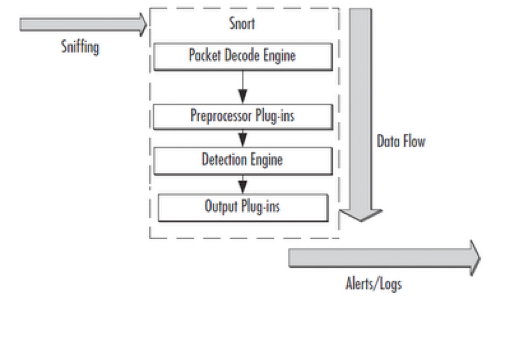
- Завершение работы по ^C

1 Введение

***www.snort.org***

Snort – разновидность сетевого анализатора NIDS (Network Intrusion Detection System ). Snort является свободно распространяемой программой с открытым исходным кодом под лицензией GPL. NIDS Snort проверяет каждый пакет, проходящий через интерфейс, в поисках известных последовательностей. С помощью Snort можно выполнять операции поиска и сопоставления над каждым пакетом, проходящим через сеть и обнаруживать множество типов атак и нелегитимного трафика в реальном времени.

Рисунок 1: Диаграмма обработки данных пакетом Snort



[***www.snort.org/downloads***](http://www.snort.org/downloads) ***(загрузка)***

[***http://securos.org.ua/kratkoe-rukovodstvo-polzovatelya-snort/***](http://securos.org.ua/kratkoe-rukovodstvo-polzovatelya-snort/)

***(краткое руководство)***

2 Конфигурация Snort

[***http://wiki.val.bmstu.ru/doku.php?id=сервис\_snort***](http://wiki.val.bmstu.ru/doku.php?id=сервис_snort) ***(краткие сведения, конфигурирование и настройка для разных систем)***

Основные данные о конфигурации Snort хранятся в файле snort.conf (большинство из приведённых ниже настроек установлены по умолчанию).

***# Раздел: Глобальные переменные для конфига***Чтобы отличить входящий трафик от исходящего, необходимо сообщить Snort узлы и IP-адреса внутренней и внешней сетей. Например:

var HOME\_NET 192.168.0.1/24

var EXTERNAL\_NET any

Информация местоположении и типах серверов содержится в переменных DNS\_SERVERS, SMTP\_SERVERS, HTTP\_SERVERS, SQL\_SERVERS, TELNET\_SERVERS и может быть задана в следующих строках файла snort.conf:

var DNS\_SERVERS $HOME\_NET

var SMTP\_SERVERS $HOME\_NET

var HTTP\_SERVERS $HOME\_NET

var SQL\_SERVERS $HOME\_NET

var TELNET\_SERVERS $HOME\_NET

var SNMP\_SERVERS $HOME\_NET

По умолчанию всем серверным переменным присвоено значение $HOME\_NET; это означает, что Snort будет контролировать все типы нападений на все системы в диапазоне HOME\_NET. Если у нас нет соответствующего сервера, то незачем отслеживать специфические для него атаки. Поэтому лишнее можно отключить.

Можно определить порты, используемые серверами для конкретных приложений. Например:

var HTTP\_PORTS 80

Это позволяет Snort не распылять ресурсы, а искать атаку целенаправленно.

***# Раздел: настройка декодеров***Задача декодеров сводится к тому, чтоб из протоколов канального уровня( Ethernet, 802.11, Token Ring…) «вытащить» данные сетевого и транспортного уровня (IP, TCP, UDP).

***# Раздел: настройка препроцессоров***  
Препроцессоры, подключаемые секции «Configure preprocessors», предназначены для обработки или предварительного анализа пакетов. Так например: sfPortscan - предназначен для выявления сканирования портов; Frag3 предназначенный для дефрагментации IP-пакетов; HTTPInspect изучает аномалии в http-трафике.

***# Раздел: конфигурация плагинов вывода***  
Настройка вывода данных. В четвертой секции «Configure output plugins» настраиваются выходные параметры. В общем случае строка параметров имеет такой вид:

output <name\_of\_plugin>: <configuration\_options>

В настоящее время Snort может использовать 10 плагинов для вывода информации (каждый из которых имеет дополнительные опции):

1. alert\_syslog – для вывода информации используется демон syslog; модуль позволяет настроить приоритеты сообщений и уровень;
2. alert\_fast – информация о возможной атаке выводится в указанный в качестве дополнительного параметра файл в сокращенном формате без подробностей;
3. alert\_full – модуль, подходящий для небольших сетей, так как сильно тормозит работу Snort; заголовок пакета выводится полностью; в лог-каталоге будет создан подкаталог, по каждому IP в который будут записываться пакеты, вызвавшие предупреждение;
4. alert\_unixsock – схож с предыдущим за исключением того, что информация в реальном времени передается в Unix-сокет, откуда может быть считана любой другой программой;
5. log\_tcpdump — записывает в указанный файл перехваченные пакеты в формате утилиты tcpdump (к имени файла будет добавляться метка времени, поэтому затереть его при перезапуске не получится);
6. database – модуль, позволяющий заносить информацию в базу данных;
7. csv – вывод в файл формата csv, который может быть использован для занесения информации в базу данных. Кроме имени файла, необходимо перечислить параметры, которые в него заносятся;
8. unified – выводит данные в специальном формате, оптимизированном для обработки внешними утилитами, которые затем будут заниматься регистрацией события;
9. alert\_prelude – доступен при конфигурировании с опцией ‘-–enable-prelude’, в этом случае Snort используется как датчик гибридной IDS Prelude([www.prelude-ids.org](http://www.prelude-ids.org/));
10. log\_null – в этом случае Snort способен реагировать на указанные предупреждения без регистрации пакетов.

Правила (сигнатуры) в соответствии с которыми система Snort определяет нарушения хранятся в папках:

var RULE\_PATH ../rules

var SO\_RULE\_PATH ../so\_rules

var PREPROC\_RULE\_PATH ../preproc\_rules

***# Раздел: подключение правил***  
В секции «Customize your rule set» необходимо убрать комментарии, указывающие на файлы с правилами:

include $RULE\_PATH/local.rules  
include $RULE\_PATH/bad-traffic.rules  
…  
# include $RULE\_PATH/multimedia.rules  
# include $RULE\_PATH/p2p.rules  
include $RULE\_PATH/experimental.rules

Названия правил говорят сами за себя. Необходимо оставить то, что действительно нужно (если есть сомнения, то лучше включи все). По умолчанию файл local.rules пуст, в него заносит свои правила сам пользователь.

3 Начало работы.

Запуск для определённого интерфейса:

snort -i <номер интерфейса>

Режимы работы:

* режим снифера;
* режим регистратора пакетов;
* режим сетевой системы выявления атак.

***Режим сниффера:***

snort -v  
- команда выводит заголовки IP- и TCP/UDP/ICMP-пакетов.

snort -vd  
- команда выводит ещё данные из пакета

snort -vde  
- еще более подробный вывод, включая заголовки канального уровня

***Режим регистрации:***

snort -v -l <file> -b

- запись в указанный каталог в формате TCPdump

snort -r <file>

- прочитать из указанного файла

***Режим выявления атак:***

snort -c /etc/snort/snort.conf -l log

- здесь необходимо указать файл конфигурации. Если не указать каталог регистрации, то регистрация пойдёт в каталог /var/log/snort/

Рабочий запуск:

snort -A fast -c /etc/snort/snort.conf

snort -A fast -r <file> -c /etc/snort/snort.conf

параметры:

-A – способ генерирования alert'ов. Возможные значения: console, fast, full, none, test. В случае fast, full создаётся файл ../log/alert.ids

-c – выявление сетевых атак, после неё указывается файл конфигурации

Сведения об обнаруженных нарушениях будут записаны в файл alert в каталоге регистрации.

4 Правила

Написание собственных правил необходимо, так как ежедневно обнаруживаются уязвимости. Для того, чтобы вовремя следить за попытками вторжения, следует научиться простому языку правил для IDS Snort.

Для указания параметров в правилах можно использовать переменные, значения которых указаны в файле "snort.conf". Например:

$HOME\_NET - обозначает домашнюю сеть, которую надо защищать и на которую будут поступать пакеты.

$EXTERNAL\_NET — внешняя сеть

$HTTP\_PORTS - номера HTTP порты перечисленные в snort.conf.

Общий синтаксис правила:

<действие\_правила> <протокол> <адрес порт> <направление> <адрес порт> ([мета\_данные] [даные\_о\_содержимом\_пакета] [данные\_в\_заголовке] [действие\_после\_обнаружения])

Например:

alert tcp any any -> 192.168.1.0/24 111 (content: »|00 01 86 a5|»;)

alert icmp any any -> 192.168.0.0/24 any (msg: ”Превышение длины ICMP данных”; dsize > 1024; )

***Действия правил***. Действия правил делятся на следующие категории:

* alert: выдать сигнал тревоги и зарегистрировать пакет;
* log: зарегистрировать пакет;
* pass: проигнорировать пакет (т.е. не выполнять никаких действий);
* activate: выдать сигнал тревоги и активизировать соответствующее динамическое (dynamic) правило;
* dynamic: не выполнять никаких действий, пока правило не будет активизировано некоторым activate-правилом; после активизации правила действовать так, как и в случае log-правила.

***Протокол.*** В данный момент поддерживаются протоколы ip, tcp, udp, icmp

***Порт.*** Номер порта n или диапазон n1:n2

***Опции.*** После указания всех параметров так называемого заголовка правила, указываются опции, по которым и будет осуществляться основной анализ пакетов:

**msg —** выводить сообщение вместе с сигналом тревоги или протоколом пакета.

msg: «<message text>»;

**ttl —** проверять значение поля TTL IP-заголовка.

ttl: <number>;

dsize – проверяет размер данных в пакете

dsize: [ < | > ] <size>;

**content —** проверять, не содержит ли пакет определенных данных.

content: [!] «<content string>»;

content: «|90C8 C0FF FFFF|/bin/sh»;

content: !»GET»;

**session —** регистрировать данные прикладного уровня.

session: [printable|all];

Аргумент printable означает, что регистрировать нужно только те данные, которые отображаются на дисплее или для устройства печати. Аргумент all дает возможность зарегистрировать данные полностью. Например, правило

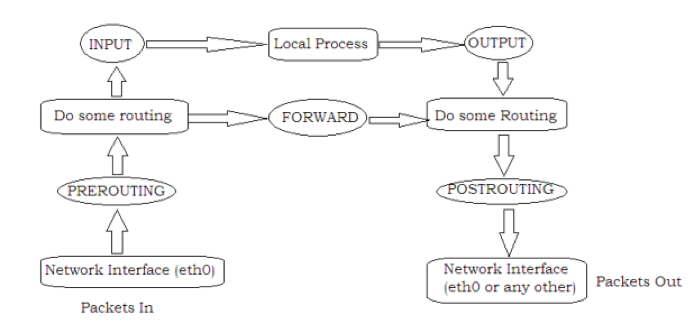
log tcp any any < > 192.168.1.0/24 23 (session: printable;)

разрешает зарегистрировать все telnet-сессии пользователей сети 192.168.1.0/24.

5 Интеграция Snort с iptables

Текст лекции

Рисунок 2: Интеграция с iptables



iptables -t nat -A PREROUTING -p tcp --dport 8080 -j NFQUEUE --queue-num 2

snort ... --daq nfq --daq-var queue=2 ...

6 Задание