УДК 004.054:004.492.3:004.056.53

А. Д. Иванов

ТЕСТИРОВАНИЕ ПРАВИЛ SURICATA МЕТОДОМ ПРОВЕДЕНИЯ АТАК

В статье рассмотрен процесс тестирования правил для системы обнаружения вторжений Suricata путем проведения атак в рамках изолированной среды. Описаны шаги атаки, которые генерируют оповещения. Для проверки достоверности правил Suricata указаны системные журналы атакуемой машины и рассмотрен файл eve.json, генерируемый используемой IDS, который содержит информацию о всех пакетах, захваченных во время работы системы.

Важность тестирования правил IDS обусловлена необходимостью постоянного повышения их точности и минимизации ложных срабатываний. Неправильно настроенные правила могут либо пропустить реальную угрозу (false negative) либо, напротив, сигнализировать об инцидентах там, где их нет (false positive). Такое поведение не только снижает уровень безопасности, но и затрудняет работу специалистов по кибербезопасности.

Система обнаружения вторжений (Intrusion detection system, IDS) – это программное или аппаратное средство, которое занимается анализом использования необходимых ресурсов и при обнаружении любых подозрительных или нетипичных событий способно оповещать о них. Также некоторые IDS способны предпринимать некоторые самостоятельные действия по обнаружению, идентификации и устранению их причин, однако данные системы способны применять лишь простейшие действия по защите [1, с. 218].

IDS используются для обнаружения вредоносной активности, которая может нарушить безопасность компьютерной системы или сети. К такой активности можно отнести внедрение вредоносного кода в сеть, сканирование портов и сетевых устройств, атаки типа «отказ в обслуживании» (DDoS), несанкционированный доступ к файлам, нацеленные на повышение привилегий атаки.

Suricata – это высокопроизводительная система обнаружения и предотвращения вторжений с открытым исходным кодом. Ее особенностями являются многопоточность, анализ множества протоколов, извлечение файлов из сетевого трафика.

Правило для Suricata состоит из:

– действия (action), определяющего, что происходит при выполнении правила;

– заголовка (header), обозначающего протокол, IP-адреса, порты и направление действия правила;

– параметров правил (rule options), устанавливающих специфику правила [2].

Сами тестируемые правила рассмотрены в таблице 1.

Таблица 1 – Тестируемые правила

Действие	Правило
1	2
Сканирование портов	alert tcp any ![22, 25, 53, 80, 88, 143, 443, 445, 465, 587, 853, 993, 1194, 8080, 51820] -> any ![22, 25, 53, 80, 88, 143, 443, 445, 465, 587, 853, 993,1194,8080,51820] (msg:"POSSBL PORT SCAN (NMAP -sT)"; flow:to_server; window:32120; flags:S; threshold:type both, track by_src, count 50, seconds 15; classtype:attempted-recon; sid:3400003; rev:3;)

Окончание таблицы 1

1	2
Telnet Brute-force	alert tcp any 23 -> any any (msg:"Telnet Multiple login failed"; content: "Login incorrect"; threshold: type both, track by_dst, count 5, seconds 60; classtype: bad-unknown; sid:1000001; rev:3;)
Reverse Shell	alert tcp any any -> any any (msg: "shellcode Reverse Shell (nc -c sh)"; content: "|6e 63 20 2d 63 20 73 68|"; fast_pattern; classtype: shellcode-detect; sid:2021002; rev:2;)
Загрузка файла на FTP-сервер	alert ftp any any -> any any (msg: "Uploading file to an FTP server"; content: "STOR "; sid:3010002; rev:1;)

Для выполнения тестирования была необходимость в создании тестовой среды. Данная среда представляет собой сеть (рисунок 1), реализованную с использованием VirtualBox. Данная сеть содержит в себе собой виртуальный коммутатор VirtualBox и три конечных точки. Первый хост представляет собой виртуальную машину с установленной Metasploitable 2, которая является специальной операционной системой с множеством уязвимостей для обучения тестированию на проникновение. На втором хосте расположена Lubuntu 24.04 с установленной Suricata и настроенным сетевым интерфейсом в режиме прослушивания трафика. Последнее устройство представляет собой атакующего с установленной Kali Linux 2024.3.

При проведении атаки сначала необходимо провести разведку целевой системы. Первым действием стоит провести сканирование портов. Проверка на наличие открытых портов будет выполняться с использованием команды: nmap -sV -p- <адрес_цели>, где -sV указывает на поиск версий сервисов, а -p- отвечает за сканирование всех возможных портов.

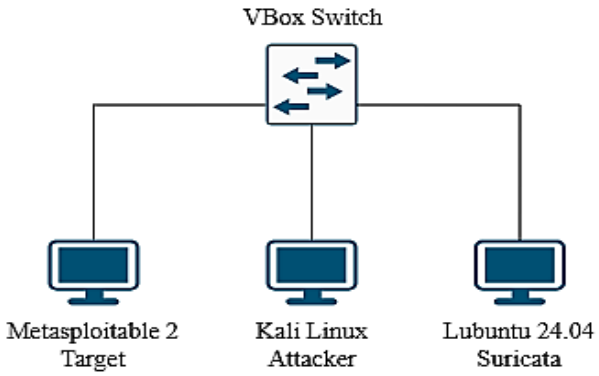

Рисунок 1 – Схема тестовой среды

После работы программы, можно увидеть наличие большого количества открытых портов (рисунок 2). Для данного сценария интерес представляет порт под номером 23 (telnet). Некоторые из оставшихся открытых портов будут рассмотрены в следующих сценариях. Telnet является протоколом, позволяющим передавать текстовые команды на удаленный сервер. Также его особенностью является отсутствие шифрования отправляемых данных.

```
┌──(kali㉿kali)-[~]
└─$ nmap -sV -p- 10.0.2.11
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-04-14 11:42 +03
Nmap scan report for 10.0.2.11
Host is up (0.0059s latency).
Not shown: 65505 closed tcp ports (conn-refused)
PORT      STATE SERVICE     VERSION
21/tcp    open  ftp         vsftpd 2.3.4
22/tcp    open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open  telnet      Linux telnetd
```

Рисунок 2 – Результат сканирования портов

В данном случае можно провести атаку подбора паролей. Эту атаку можно провести с использованием инструмента auxiliary/scanner/telnet/telnet_login, встроенного в Metasploit Framework. В этом инструменте необходимо указать адрес цели и путь к словарю, содержащему логины и пароли. После выполнения работы инструмента можно будет получить список имеющихся пользователей в системе с соответствующим паролем. В данном случае были найдены user:user и msfadmin:msfadmin.

Когда получен доступ к системе, следует закрепиться в ней, на случай если на целевой машине изменятся пароли для входа в найденных раньше пользователей. Это реализуется с помощью встроенного в Linux-системы инструмента под названием netcat.

Сначала атакующему необходимо настроить netcat на прослушивание командной: nc -nlvp <порт>.

После на скомпрометированной машине используется команда для установки соединения между текущим устройством и машиной атакующего. Это выполняется следующей командой: nc -c sh <IP-адрес> <порт>.

И последнее, что остается, это отправить интересующие данные с атакуемой машины. «Выкачать» данные можно на внешний FTP-сервер, который принадлежит атакующей стороне. Необходимо лишь со скомпрометированной машины подключиться через FTP к нужному устройству и передать на него данные. «Поднять» сервер можно используя vsftpd. Выкачивается файл командой curl -T <путь/к/файлу> ftp://<IP-адрес>/ --user логин:пароль.

В результате проводимых выше действий IDS должна сгенерировать оповещение по следующим правилам: сканирование через nmap, угадывание паролей по telnet, установка обратного shell через ncat, попытка подключения к внешнему FTP-серверу и загрузка данных на него (таблица 2). После проверки срабатывания данных правил следует проверить системные журналы для того, чтобы убедиться в правильности работы используемых в Suricata правил.

Таблица 2 – Результаты срабатывания правил для первого сценария

Действие	Оповещение
Сканирование портов	[1:3400003:3] POSSBL PORT SCAN (NMAP -sT) [Classification: Attempted Information Leak] [Priority: 2] {TCP} 10.0.2.7:60270 -> 10.0.2.11:30743
Telnet Brute-Force	[1:1000001:3] Telnet Multiple login failed [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 10.0.2.11:23 -> 10.0.2.7:40552
Reverse shell	[1:2021002:2] shellcode Reverse Shell (nc -c sh) [Classification: Executable code was detected] [Priority: 1] {TCP} 10.0.2.7:48738 -> 10.0.2.11:23
Загрузка файла на FTP-сервер	[1:3010002:1] Uploading file to an FTP server [Classification: (null)] [Priority: 3] {TCP} 10.0.2.11:37340 -> 10.0.2.7:21

Для проверки множественных попыток входа в систему необходимо просмотреть файлы /var/log/syslog для поиска подключений по telnet и /var/log/auth.log для проверки наличия неудачных входов в систему. Изучив заранее сделанные дампы этих файлов, можно найти множественные попытки подключения по telnet и неудачные попытки входа в систему в одно и то же время (рисунок 3).

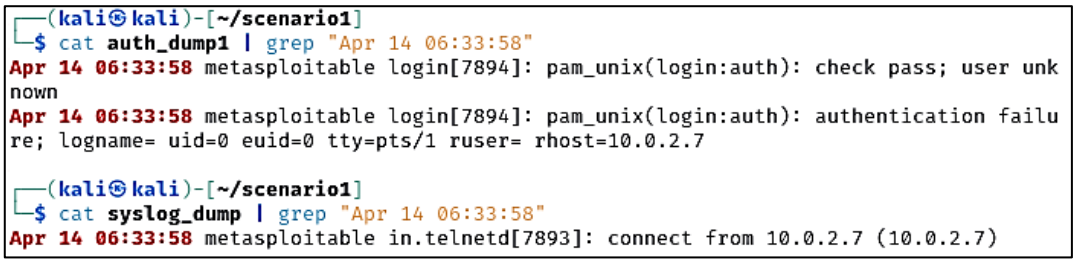

Рисунок 3 – Проверка системных журналов

Для более подробной проверки можно воспользоваться файлом eve.json, генерируемым Suricata. В данный файл попадают все сетевые события, произошедшие во время работы IDS. Отсортировав вывод по FTP запросам, можно обнаружить загрузку файла testfile.txt на сервер (рисунок 4). Исходя из этого можно подтвердить наличие подключение к FTP-серверу, т. к. без подключения не было бы возможности загрузить файл.

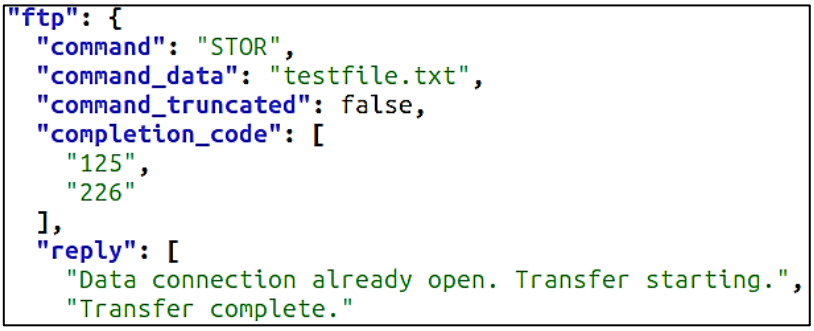

Рисунок 4 – Анализ FTP подключений в eve.json

Литература

1 Диогенес, Ю. Кибербезопасность: стратегии атак и обороны / Ю. Диогенес, Э. Озкайя ; пер. с анг. Д. А. Беликова. – М. : ДМК Пресс, 2020. – 326 с.

2 Suricata User Guide – Suricata 7.0.9 documentation : [сайт]. – URL: https://docs.suricata.io/en/suricata-7.0.9 (дата обращения: 09.04.2025).

УДК 535.3

Д. Е. Комяков

ИССЛЕДОВАНИЕ ОПТИЧЕСКИХ СВОЙСТВ И СТРУКТУРЫ ПОКРЫТИЙ, ФОРМИРУЕМЫХ В УСЛОВИЯХ ВЫСОКОГО ВАКУУМА

В статье представлены результаты комплексного исследования оптических свойств и структуры покрытий, формируемых в условиях высокого вакуума. Основное внимание уделяется анализу однослойных покрытий на основе диоксида титана (TiO_2), диоксида кремния (SiO_2) и сульфида цинка (ZnS). В работе детально изучены оптические характеристики однослойных покрытий, включая коэффициенты пропускания и отражения.

Расширение применения интерференционных диэлектрических покрытий стимулирует разработку новых методов и устройств для их получения. К этим методам относятся вакуумные методы получения интерференционных пленок, в частности получение