International Scientific Conference 20 UNITECH 26

INTERNATIONAL SCIENTIFIC CONFERENCE
19-21 November 2026
GABROVO

ПРИЛОЖЕНИЕ НА SNORT ЗА НАБЛЮДЕНИЕ И АНАЛИЗ НА МРЕЖОВИЯ ТРАФИК В ЛОКАЛНА МРЕЖА

Николай Манчев*

Технически университет – Габрово, ул. „Хаджи Димитър“ 4, Габрово, България
**кореспондиращ автор: n.manchevmobb@gmail.com*

APPLICATION OF SNORT FOR MONITORING AND ANALYSIS OF NETWORK TRAFFIC IN A LOCAL AREA NETWORK

Nikolay Manchev*

Technical University of Gabrovo, 4 Hadzhi Dimitar Str., Gabrovo, Bulgaria
**Corresponding author: n.manchevmobb@gmail.com*

Abstract
The paper presents a study of the open-source intrusion detection system Snort 3 for monitoring and analysis of network traffic in a local area network. The architecture of Snort, its operating modes and the structure of its rules are described. A laboratory environment based on switch port mirroring was built, in which controlled attacks were carried out – port scanning, ICMP flooding, SSH brute-force, SQL injection and exploitation of a known vulnerability. Detection accuracy, false positives and sensor load at different traffic rates were evaluated. The results show that the combination of community and custom rules detects all tested scenarios, while tuning of the configuration (multithreading, AF_PACKET acquisition and rule selection) reduces packet loss at 1 Gbit/s from 7.8 % to 1.6 %. Recommendations for the practical deployment of Snort in local networks are formulated.

Keywords: Snort; intrusion detection system; network traffic analysis; network security; local area network.

ВЪВЕДЕНИЕ

Нарастващият обем на мрежовия трафик и разнообразието на използваните услуги превръщат наблюдението на локалните мрежи в основна задача при осигуряването на информационната сигурност. Защитните стени контролират достъпа по адреси и портове, но не анализират съдържанието на комуникацията и не могат да разпознаят атака, пренасяна през разрешен протокол. Поради това в съвременните мрежи се използват системи за откриване на проникванията (Intrusion Detection Systems – IDS), които анализират пакетите в реално време и генерират известия при установяване на подозрителна активност [2, 3].

Snort е сред най-разпространените мрежови IDS с отворен код. Разработен е първоначално от М. Рьош като „лека“ система за откриване на проникванията [1], а понастоящем се поддържа от Cisco Talos. Третата версия на продукта (Snort 3) е с преработена архитектура, поддържа многонишкова обработка, конфигуриране чрез скриптовия език Lua и нов, по-гъвкав синтаксис на правилата [8]. Като алтернативи в практиката се използват Suricata и Zeek [4, 7], но Snort остава предпочитан в учебния процес и в малки и средни организации поради обширната си документация и наличието на безплатни набори от пра-вила.

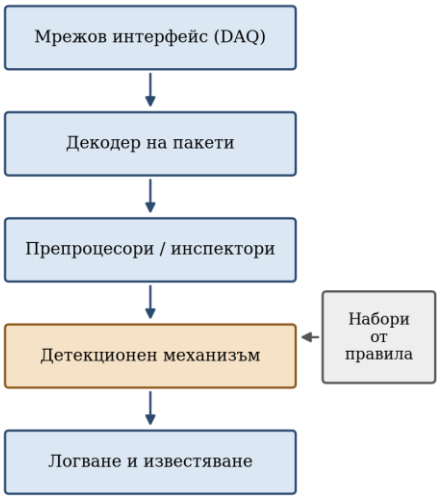
Print ISSN: 1313-230X, Online ISSN: 3033-1404

Целта на настоящата работа е да се изследва приложимостта на Snort 3 за наблюдение и анализ на трафика в локална мрежа, като се оценят точността на откриване на типични атаки и производителността на сензора при различно натоварване. За постигане на целта са решени следните задачи: анализ на архитектурата и правилата на Snort; изграждане на лабораторна среда; провеждане на контролирани атаки; оценка на резултатите и формулиране на препоръки за конфигуриране.

ИЗЛОЖЕНИЕ

Архитектура и режими на работа на Snort

Snort обработва трафика последователно в няколко функционални модула (фиг. 1). Модулът за придобиване на данни (DAQ – Data Acquisition) получава пакетите от мрежовия интерфейс чрез библиотеки като libpcap или AF_PACKET. Декодерът разпознава заглавните части на протоколите от каналния до транспортния слой и формира вътрешна структура на пакета. Препроцесорите, наричани в Snort 3 инспектори, извършват нормализация, дефрагментация на IP пакети, възстановяване на TCP потоци и анализ на приложни протоколи като HTTP, DNS, FTP и SMTP.



***Фиг. 1**. Обобщена архитектура на Snort*

Детекционният механизъм сравнява обработените данни с набора от правила, като използва алгоритми за многошаблонно търсене (напр. Aho–Corasick) за бързо откриване на съвпадения. Последният модул записва събитията и генерира известия в текстов, JSON или двоичен (unified2) формат.

Snort може да работи в три основни режима: като анализатор на пакети (sniffer), при който заглавните части се извеждат в конзолата; като регистратор на пакети (packet logger), при който трафикът се записва във файл за последващ анализ; и като система за откриване на проникванията (NIDS), при който трафикът се анализира спрямо правилата. При включване в линия (inline) Snort може да функционира и като система за предотвратяване на проникванията (IPS), като отхвърля пакетите, съвпадащи с правила с действие drop или reject.

Правилото в Snort се състои от заглавна част и опции. Заглавната част определя действието (alert, log, pass, drop, reject), протокола, адресите и портовете на източника и получателя и посоката на трафика. Опциите, оградени в скоби, задават съобщението (msg), критериите за съдържание (content, pcre), състоянието на потока (flow), праговете (detection_filter) и уникалния идентификатор (sid). В Snort 3 са въведени т.нар. „лепкави буфери“ (sticky buffers), напр. http_uri, които насочват търсенето към конкретна част от приложния протокол и повишават точността.

За да се постави Snort в контекста на другите системи с отворен код, в табл. 1 е направено кратко сравнение със Suricata и Zeek. Suricata използва сходен сигнатурен подход и в голяма степен е съвместима с правилата на Snort, докато Zeek е ориентиран към подробно журналиране на мрежовите събития и поведенчески анализ чрез собствен скриптов език [4, 7].

***Таблица 1**. Сравнение на системи за мрежов анализ*

Характеристика	Snort 3	Suricata	Zeek
Основен подход	Сигнатурен	Сигнатурен	Поведенчески
Многонишковост	Да	Да	Чрез клъстер
Режим IPS	Да	Да	Не
Конфигуриране	Lua	YAML	Скриптове
Формат на изхода	Известия, pcap	EVE JSON	Журнали

Експериментална среда и методика

Изследването е проведено в изолирана лабораторна мрежа (фиг. 2). Сензорът Snort 3 е инсталиран на сървър с Ubuntu Server 24.04 LTS, четириядрен процесор и 16 GB оперативна памет, свързан към огледален (SPAN) порт на управляем комутатор. По този начин сензорът получава копие на целия трафик между сегментите, без да влияе на комуникацията. Като атакуваща машина е използван Kali Linux, а като цел – умишлено уязвимата система Metasploitable 2. Фоновият трафик е генериран чрез tcpreplay от записи на реален легитимен трафик (HTTP, DNS, TLS), което позволява повторяемост на измерванията при зададен интензитет.



***Фиг. 2**. Топология на лабораторната среда*

Използвани са безплатният набор Snort Community Rules и набор от локални правила, разработени за целите на изследването. Активиран е инспекторът port_scan за откриване на сканиране на портове. Известията са записвани във формат alert_json, което улеснява последващата им обработка и корелация със записите от Wireshark. Основните настройки в конфигурационния файл snort.lua са:

```
HOME_NET = '192.168.10.0/24'
EXTERNAL_NET = '!$HOME_NET'
port_scan = { protos = 'all',
  scan_types = 'all', alert_all = false }
alert_json = { file = true, fields =
  'timestamp msg src_addr dst_addr sid' }
ips = { enable_builtin_rules = true,
  include = RULE_PATH .. '/local.rules' }
```

Примери за разработените локални правила са дадени по-долу. Първото правило открива наводнение с ICMP заявки, второто – многократни опити за установяване на SSH връзка от един източник, а третото – характерни ключови думи на SQL инжекция в URI на HTTP заявката:

```
alert icmp any any -> $HOME_NET any (
  msg:"ICMP flood detected"; itype:8;
  detection_filter:track by_src,
  count 100, seconds 1; sid:1000001;)
alert tcp any any -> $HOME_NET 22 (
  msg:"Possible SSH brute-force";
  flow:to_server; flags:S;
  detection_filter:track by_src,
  count 5, seconds 60; sid:1000002;)
alert tcp any any -> $HOME_NET 80 (
  msg:"SQL injection - UNION SELECT";
  flow:to_server,established; http_uri;
  content:"union",nocase; content:"select",
  nocase,distance:0; sid:1000003;)
```

Методиката включва четири етапа: (1) запис на базова линия на легитимния трафик за 24 часа с цел оценка на фалшивите положителни известия; (2) последователно провеждане на седем контролирани сценария на атака, всеки повторен по три пъти; (3) измерване на натоварването на процесора и дела на загубените пакети при интензитет на фоновия трафик от 100 до 1000 Mbit/s; (4) повторение на измерванията след оптимизация на конфигурацията. Оптимизацията включва многонишкова обработка с четири нишки (параметър -z), DAQ модул afpacket, изключване на правила за неизползвани в мрежата услуги и ограничаване на дълбочината на инспекция на HTTP отговорите.

За оценка на точността на откриване са използвани процентът на откриване DR и делът на фалшивите положителни известия FPR [3]:

$$DR = TP / (TP + FN), \quad (1)$$

$$FPR = FP / (FP + TN), \quad (2)$$

където TP, FN, FP и TN са съответно истински положителните, фалшиво отрицателните, фалшиво положителните и истински отрицателните резултати.

Резултати и анализ

Резултатите от контролираните атаки са обобщени в табл. 2. Всички седем сценария са открити от сензора, което съгласно (1) дава процент на откриване DR = 100 % за изпитания набор. Сканирането на портове се регистрира от инспектора port_scan като обобщено събитие, което значително намалява броя на известията в сравнение с подход, при който всеки сондиращ пакет се отчита поотделно.

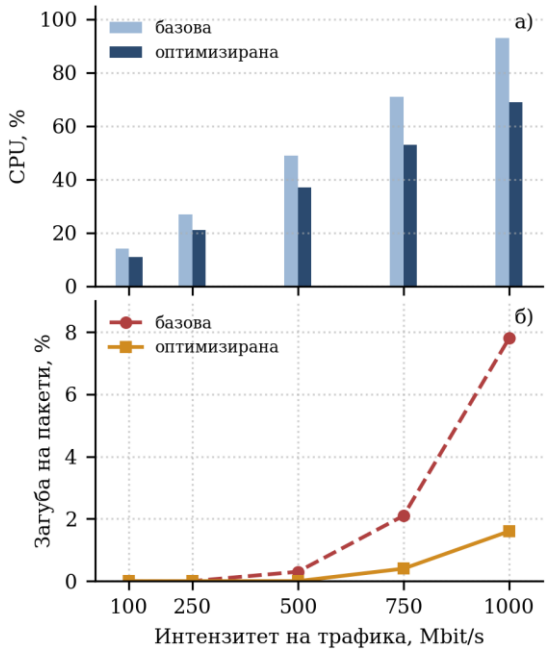
***Таблица 2**. Резултати от сценариите на атака*

Сценарий	Инструмент	Известия	Източник
TCP SYN сканиране	nmap -sS	1	port_scan
UDP сканиране	nmap -sU	1	port_scan
Определяне на ОС	nmap -O	3	Community
ICMP наводнение	hping3	12	Локално
SSH brute-force	Hydra	38	Локално
SQL инжекция	sqlmap	146	Comm. + лок.
Експлойт vsftpd 2.3.4	Metasploit	2	Community
Легитимен трафик	tcpreplay	4	Фалшиви

Атаката с груба сила срещу SSH не се открива от общностните правила, тъй като отделните опити за вход са легитимни TCP връзки; тя е разпозната едва след добавяне на локално правило с праг (detection_filter). Аналогично е положението с ICMP наводнението, кое-то изисква праг, съобразен с нормалния профил на мрежата.

Най-голям брой известия (146) е генериран при SQL инжекцията, тъй като инструментът sqlmap изпраща голям брой вариации на заявките. Това показва, че при реална експлоатация е необходимо агрегиране или прагово ограничаване на повтарящите се известия, за да не се претоварва операторът. При 24-часовия запис на легитимен трафик са регистрирани 4 фалшиви положителни известия, свързани с нестандартни DNS заявки и HTTP заглавия, генерирани от приложения за автоматично обновяване. След добавяне на две правила от тип pass и потискане (suppress) на съответните сигнатури фалшивите известия са елиминирани, т.е. FPR съгласно (2) клони към нула.

На фиг. 3 са показани натоварването на процесора (а) и делът на загубените пакети (б) в зависимост от интензитета на трафика. При базовата конфигурация (една нишка, DAQ pcap, пълен набор от правила) натоварването нараства почти линейно и достига 93 % при 1000 Mbit/s, а загубата на пакети започва над 500 Mbit/s и достига 7,8 %.



***Фиг. 3**. Натоварване на процесора (а) и загуба на пакети (б)*

Загубата на пакети е критична за IDS, тъй като пропуснатият пакет може да съдържа именно злонамерения товар. След оптимизацията натоварването при 1000 Mbit/s намалява до 69 %, а загубата на пакети – до 1,6 %. Най-голям принос има разпределението на обработката между няколко нишки, следвано от премахването на неприложимите правила.

Получените резултати са в съответствие с изводите в [4] и [5], според които производителността на Snort зависи в голяма степен от броя на активните правила и от ефективното използване на многоядрените процесори. За локални мрежи с интензитет на трафика до няколкостотин Mbit/s сензор на стандартен сървърен хардуер осигурява анализ практически без загуба на пакети, докато за гигабитови връзки е необходима внимателна настройка или разпределяне на натоварването между няколко сензора.

Трябва да се отбележат и ограниченията на изследването. Сигнатурният подход не открива неизвестни атаки (zero-day), за които няма правило, а анализът на криптиран трафик (TLS) е ограничен до метаданните на връзката. Тези ограничения могат да бъдат компенсирани чрез комбиниране на Snort със системи за анализ на поведението и аномалиите, както и чрез използване на съвременни набори от данни като CIC-IDS2017 [6] за оценка в по-широк кръг от сценарии.

Препоръки за практическо приложение

На базата на проведеното изследване могат да се формулират следните препоръки за внедряване на Snort в локална мрежа:

– сензорът да се свързва към огледален порт или мрежов разклонител (TAP) в точката с най-голяма концентрация на трафика;
– променливите HOME_NET и EXTERNAL_NET да се дефинират точно, тъй като от тях зависи коректността на голяма част от правилата;
– да се активират само правилата, съответстващи на реално използваните в мрежата услуги и операционни системи;
– праговите стойности в локалните правила да се определят след запис на базова линия на нормалния трафик;
– наборите от правила да се обновяват редовно, напр. автоматизирано чрез инструмента PulledPork;
– известията да се изпращат към централизирана система за управление на събития (SIEM) за корелация и дългосрочно съхранение;
– при трафик над 500 Mbit/s да се използва многонишкова обработка и DAQ модул afpacket.

ЗАКЛЮЧЕНИЕ

В статията е изследвано приложението на Snort 3 за наблюдение и анализ на мрежовия трафик в локална мрежа. Изградена е лабораторна среда с огледален порт и са проведени седем сценария на контролирани атаки. Установено е, че комбинацията от общностни и локално разработени правила осигурява откриване на всички изпитани атаки при нисък дял на фалшивите положителни известия, като атаките, основани на обем и честота (brute-force, наводнение), изискват правила с прагови стойности. Оптимизацията на конфигурацията чрез многонишкова обработка и подбор на правилата намалява загубата на пакети при 1 Gbit/s от 7,8 % до 1,6 %.

Предложената лабораторна среда и методика могат да се използват както за оценка на защитата на реални мрежи, така и в учебния процес по дисциплини, свързани с мрежовата сигурност. Бъдещата работа ще бъде насочена към сравнение на Snort 3 и Suricata при еднакви условия и към интегриране на методи за откриване на аномалии.

БЛАГОДАРНОСТИ

Докладът се публикува във връзка с резултатите от изпълнение на дейностите по проект № НИП2026-20/2026. „Изследване на технологии и интелигентни приложения за обработка и визуализация на данни в IoT/5G комуникационни мрежи“ към УЦНИТ при ТУ – Габрово

ЛИТЕРАТУРА

[1] Roesch, M. Snort – Lightweight Intrusion Detection for Networks. Proceedings of the 13th USENIX Conference on System Administration (LISA '99), Seattle, 1999, pp. 229–238.

[2] Scarfone, K., Mell, P. Guide to Intrusion Detection and Prevention Systems (IDPS). NIST Special Publication 800-94, National Institute of Standards and Technology, 2007.

[3] Khraisat, A., Gondal, I., Vamplew, P., Kamruzzaman, J. Survey of intrusion detection systems: techniques, datasets and challenges. Cybersecurity, vol. 2, no. 20, 2019.

[4] Albin, E., Rowe, N. C. A Realistic Experimental Comparison of the Suricata and Snort Intrusion-Detection Systems. 26th International Conference on Advanced Information Networking and Applications Workshops (WAINA), IEEE, 2012, pp. 122–127.

[5] Bul'ajoul, W., James, A., Pannu, M. Improving network intrusion detection system performance through quality of service configuration and parallel technology. Journal of Computer and System Sciences, vol. 81, no. 6, 2015, pp. 981–999.

[6] Sharafaldin, I., Lashkari, A. H., Ghorbani, A. A. Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP), 2018, pp. 108–116.

[7] Paxson, V. Bro: A System for Detecting Network Intruders in Real-Time. Computer Networks, vol. 31, no. 23–24, 1999, pp. 2435–2463.

[8] Cisco Talos. Snort 3 User Manual. Available at: https://www.snort.org/documents (accessed 13 October 2026).