УДК 004.05(045) DOI: 10.31673/2786-8362.2024.027035

Поночовний П.М., Іванченко І. С., к.т.н.

МЕТОД РАННЬОГО ВИЯВЛЕННЯ ТА ЗАХИСТУ ВІД МІНЛИВИХ DDOS АТАК НА ОСНОВІ АНАЛІЗУ ОБРОБКИ ПАКЕТНИХ ГРУП

Ponochovny P.M., Ivanchenko I.S. Early detection and defense methods for variable ddos attacks based on packet group processing analysis. This paper proposes a method for early detection and defense of variable DDoS attacks based on packet group processing analysis. The method combines real-time traffic analysis and machine learning to detect anomalies in network data behavior. This enables a quick response to changes in the attack vector and guarantees the stability and security of the information system. The effectiveness of the method is confirmed by experimental studies that demonstrate the accuracy of detection and minimization of delays in network operations. The method is based on dividing network traffic into groups of packets and analyzing them taking into account their statistical, temporal and behavioral characteristics. Particular attention is paid to the use of machine learning algorithms to detect deviations in traffic patterns characteristic of DDoS attacks. The proposed approach makes it possible to detect the signs of an attack at an early stage, before the impact of the attack becomes fatal for the infrastructure. This paper describes an algorithm for processing packet swarms that takes into account the variability of attacks and adapts to new attacker methods. The computational efficiency of the method is also discussed, which is important to ensure its practical application on heavily loaded systems. To evaluate the effectiveness of the method, a series of experiments were performed on real and synthetic datasets, achieving high attack detection accuracy (>95%) and a low level of false positives. The application of the developed method not only provides effective protection against the latest DDoS attacks, but also minimizes the risk of financial loss and reputational damage associated with their consequences. The research results can be integrated into existing defense systems to increase their adaptability and resilience to cyber threats.

Keywords: DDoS attacks, traffic analysis, packet swarm processing, network protection, machine learning, cyber security, early detection

Поночовний П.М., Іванченко І.С. Метод раннього виявлення та захисту від мінливих DDoS атак на основі аналізу обробки пакетних груп. У статті запропоновано метод раннього виявлення та захисту від мінливих DDoS-атак, заснований на аналізі обробки пакетних груп. Метод поєднує аналіз трафіку в реальному часі та машинне навчання для виявлення аномалій у поведінці мережевих даних. Це дозволяє оперативно реагувати на зміни у векторі атаки та забезпечувати стабільність і захищеність інформаційних систем. Ефективність методу підтверджено експериментальними дослідженнями, які демонструють точність виявлення та мінімізацію затримок у роботі мережі.

Метод базується на розподілі мережевого трафіку на групи пакетів, які аналізуються з урахуванням їхніх статистичних, часових та поведінкових характеристик. Особливу увагу приділено використанню алгоритмів машинного навчання для виявлення відхилень у паттернах трафіку, що характерні для DDoS-атак. Запропонований підхід дозволяє ідентифікувати ознаки атак на ранніх етапах, ще до того, як їхній вплив стане критичним для інфраструктури.

У роботі описано алгоритм обробки пакетних груп, що враховує змінність атак та адаптується до нових технік зловмисників. Окремо розглянуто питання обчислювальної ефективності методу, що є важливим для забезпечення його практичного застосування у високонавантажених системах. Для оцінки ефективності було проведено серію експериментів на реальних і синтетичних наборах даних, які продемонстрували високу точність виявлення атак (більше 95%) та низький рівень помилкових спрацьовувань.

Застосування розробленого методу дозволяє не лише забезпечити ефективний захист від сучасних DDoS-атак, але й мінімізувати ризики фінансових та репутаційних втрат, пов'язаних із їхніми наслідками. Результати дослідження можуть бути інтегровані в існуючі системи захисту для підвищення їхньої адаптивності та стійкості до кіберзагроз.

Ключові слова: DDoS-атаки, аналіз трафіку, обробка пакетних груп, захист мережі, машинне навчання, кібербезпека, раннє виявлення

Вступ

Сучасні інформаційні системи та мережі стають все більш уразливими перед новими кіберзагрозами, зокрема розподіленими атаками типу «відмова у обслуговуванні» (DDoS). Ці атаки спрямовані на вичерпання ресурсів системи, що призводить до часткового або повного порушення її функціонування. Особливу складність становлять мінливі DDoS-атаки, які

змінюють свої вектори та інтенсивність, ускладнюючи їхнє виявлення традиційними методами.

Одним із перспективних підходів до вирішення цієї проблеми є аналіз обробки груп мережевих пакетів. Використання такого підходу дозволяє виявляти аномальні зміни в трафіку та забезпечувати раннє реагування на загрози. Інтеграція сучасних технологій, зокрема методів машинного навчання, відкриває можливості для автоматизації процесів аналізу та підвищення ефективності засобів захисту від DDoS-атак.

Аналіз останніх досліджень. Аналіз останніх досліджень у галузі раннього виявлення та захисту від змінних DDoS-атак на основі аналізу обробки пакетних груп

DDoS-атаки залишаються однією з головних загроз для кібербезпеки, оскільки вони еволюціонують, стаючи складнішими та більш непередбачуваними. У відповідь на це з'являються нові методи виявлення й захисту, які поєднують технології машинного навчання, поведінкового аналізу та обробки даних.

Наукові джерела [3, 8] описують кілька підходів до класифікації DDoS-атак залежно від рівня аналізу: трафік, операційна система, сервіси. Наприклад, якісний аналіз трафіку включає моніторинг заголовків пакетів, визначення аномалій у розмірах або вмісті пакетів. Кількісний аналіз базується на обчисленні інтенсивності запитів за одиницю часу [5, 10, 11].

Дослідники [6, 8] наголошують на ефективності машинного навчання для раннього виявлення атак. Особливо успішними є нейронні мережі та дерева прийняття рішень. Їх застосовують для класифікації трафіку, виявлення аномалій у режимі реального часу. Наприклад, модель на основі дерева рішень досягла 94% точності при виявленні атак типу SYN-flood, UDP-flood [6].

Для підвищення ефективності рекомендовано використовувати комбіновані стратегії. Це включає виявлення поведінкових аномалій у трафіку, інтеграцію з хмарними платформами та застосування технологій обробки великих даних [7, 12]. Сучасні системи захисту інтегруються з алгоритмами аналізу пакетів, що дозволяє блокувати атаки ще до того, як вони досягнуть критичного рівня [8, 10].

Системи на базі штучного інтелекту дозволяють автоматизувати процес фільтрації шкідливих пакетів. Вони враховують такі параметри, як IP-адреси джерела й цілі, швидкість передачі даних, а також зміну поведінки в реальному часі. Подібні технології також використовуються для оповіщення адміністраторів про потенційні загрози [9].

Постановка завдання. В умовах стрімкого зростання обсягів мережевого трафіку та розвитку технологій кіберзахисту розподілені атаки типу "відмова у обслуговуванні" (DDoS) стають дедалі складнішими й більш руйнівними. Зокрема, мінливі DDoS-атаки, що характеризуються динамічною зміною векторів впливу, типів трафіку та інтенсивності, значно ускладнюють їх своєчасне виявлення та протидію. Традиційні методи аналізу мережевого трафіку, такі як порівняння зі статичними сигнатурами чи використання простих статистичних моделей, виявляються недостатньо ефективними для роботи в умовах таких атак.

Основною проблемою є низька швидкість та точність існуючих підходів, які не дозволяють ідентифікувати DDoS-атаки на ранніх етапах, коли вони ще не призводять до критичного перевантаження системи. Крім того, багато сучасних систем захисту не враховують змінність параметрів атак, що сприяє їхньому успішному обходу.

У зв'язку з цим постає необхідність розробки ефективних методів раннього виявлення та захисту, які базуватимуться на аналізі обробки груп пакетів і використовуватимуть сучасні алгоритми машинного навчання для адаптації до динамічних змін в атаках. Такий підхід дозволить не лише підвищити точність і швидкість виявлення, а й знизити ризики порушення працездатності критичних інформаційних систем.

Метою роботи є розробка методу раннього виявлення та захисту від мінливих DDoS-атак, який базується на аналізі обробки пакетних груп. Метод спрямований на підвищення швидкості і точності ідентифікації атак із мінімальним впливом на продуктивність системи.

Виклад основного матеріалу дослідження.

Система захисту від DDoS атак. Розглянемо детально представлені окремі модулі розробленої автором системи захисту від DDoS-атак. Та розглянемо запропоновано рішення.

Конфігурація HTTP-сервера, яка використовується для дослідження, така:

- Процесор: Intel Core i3-2120 CPU, 3.30 GHz, 3M Cache
- Частота процесора: 3,30 ГГц
- Оперативна пам'ять: 8 Гб
- Мережева карта: 3Com Typhoon (3CR990-TX-97) на MMIO 0xecf80000, 00:01:03:e6:65:e9.
- ОС: CentOS Linux версія 6.0; Linux версія 2.6.32-71.29.1.el6.i686 gcc версія 4.4.4 20100726 (Red Hat 4.4.4-13)

Захист від DDoS на GNU/Linux. Підвищення безпеки системи може бути відносно складним процесом. Зазвичай це передбачає налаштування всіх служб, які система повинна запускати в найбільш безпечний спосіб, блокування системи для запобігання локальним вторгненням. Забезпечення безпеки запущених процесів не пов'язане із захистом решти системи та її невідомих слабких місць. У системі GNU/Linux мають працювати лише ті компоненти, без яких вона не може реалізувати свою функціональність. Всі інші компоненти не повинні бути задіяні в конфігурації [14].

Доступні оновлення, які допомагають системному адміністратору повністю реалізувати захист. Одним із таких оновлень є grsecurity.

Використовуючи налаштування GNU/Linux з оновленням grsecurity, можна запобігти атакам переповнення буфера, а також створити списки контролю доступу. Після встановлення рівня безпеки в діалоговому вікні здійснюється вибір активних функцій. Усі служби вимагають реєстрації всіх подій. Ми розглянемо параметри налаштування стеку TCP/IP.

Захист від DDoS у конфігурації Apache. У цьому пункті описуються параметри Apache, спрямовані на уникнення проблем, спричинених DDoS-атаками.

TimeOut – ця директива повинна мати найменше можливе значення (на HTTP-сервері під час DDoS-атаки).

Директива KeepAliveTimeout — також слід зменшити значення та/або повністю вимкнути.

Значення різних директив часу можна представити наступним чином:

LimitRequestBody, LimitRequestFields, LimitRequestFieldSize, LimitRequestLine, LimitXMLRequestBody – потрібно правильно налаштувати, щоб обмежити споживання ресурсів, пов’язаних із запитами клієнтів.

У цьому випадку використання директиви AcceptFilter є обов’язковим. За замовчуванням він включений у конфігурацію Apache httpd, але для роботи може знадобитися повторна компіляція з новими налаштуваннями ядра операційної системи.

Директива MaxClients встановлює максимальну кількість клієнтів, які зможуть підключитися до сервера одночасно. Менше значення означає менше навантаження на сервер HTTP.

Вибір іншого модуля mpm може дозволити обробляти більше одночасних з’єднань, а не лише обмежити ефект DDoS-атаки. За замовчуванням у системах GNU/Linux встановлено mpm - prefork, який потребує найбільше системних ресурсів і, відповідно, має найменшу продуктивність.

Існує низка інших модулів Apache, які можуть обмежити певні шаблони поведінки клієнта та таким чином пом’якшити DDoS-атаку.

Необхідно автоматизувати процес блокування атакуючих зомбі-машин.

В результаті аналізу автор орієнтується на модуль Apache для організації захисту від DDoS-атак - mod_dosevasive.

Наступні правила додаються під час налаштування серверу Apache:

```
<IfModule mod_evasive20.c>
DOSHashTableSize 3097
DOSPageCount 6
DOSSiteCount 100
DOSPageInterval 2
```

DOSSiteInterval 2
DOSBlockingPeriod 600
</IfModule>

Опис налаштувань:

- DOSHashTableSize: це розмір хеш-таблиці, яка обробляє запити до веб-сервера.
- DOSPageCount: кількість запитів до однієї сторінки з тієї самої IP-адреси протягом заданого інтервалу часу.
- DOSSiteCount: кількість запитів до всіх сторінок домену, тобто якщо більше 100 запитів надійшло з однієї IP-адреси до різних сторінок домену, тоді ця IP-адреса буде заблокована.
- DOSPageInterval: інтервал для директиви DOSPageCount (у секундах).
- DOSSiteInterval: інтервал для директиви DOSSiteCount (у секундах).
- DOSBlockingPeriod: Як довго блокувати відповідний IP (у секундах).
- DOSEmailNotify: можна використовувати для сповіщення, надішле електронний лист про те, що IP-адресу заблоковано.
- DOSSystemCommand: ця директива використовується для виконання деяких команд, коли IP заблоковано. Наприклад, їх можна використовувати для додавання IP-адреси до таблиці брандмауера (наприклад: "/sbin/iptables -A INPUT -p tcp --dport 80 -s %s -j REJECT" У %s передається з IP-адреси модуль)
- DOSWhiteList: список «білих» IP-адрес, можливо, з масками (наприклад, 127.0.0.*)

Параметри були змінені таким чином:

Для apache (httpd.conf):

Timeout 20
MaxKeepAliveRequests 15
KeepAliveTimeout 2
MinSpareServers 3
MaxSpareServers 64
StartServers 1024
Максимальна кількість клієнтів 2500
MaxRequestsPerChild 100000
MaxConnPerIP 25

Для mod_php:

php_admin_flag safe_mode on
php_admin_flag allow_url_fopen off
php_admin_value doc_root /home/hst_mklimat/htdocs
php_admin_flag magic_quotes_runtime on
php_admin_value open_basedir /home/hst_mklimat/htdocs
php_admin_value upload_tmp_dir /home/hst_mklimat/htdocs/tmp
php_admin_value safe_mode_allowed_env_vars PHP_
php_admin_value upload_max_filesize 1024000
php_admin_value max_execution_time 10
php_admin_value post_max_size 1M
php_admin_value memory_limit 1M
php_admin_value admin_flag mysql.allow_persistent off
php_admin_value mysql.max_links 5
php_admin_flag pgsql.allow_persistent off
php_admin_value pgsql .max_links 5
php_admin_value disable_functions mysql_pconnect,pg_pconnect

Для mysql (/etc/my.cnf):

[mysqld]

set-variable = max_connections=15
set-variable = thread_concurrency=8

Програмний захист від DDoS. *DDoS Deflate* це безкоштовний скрипт, за допомогою якого можна зупинити «м'які» DDoS-атаки і використовується для цього в готових продуктах. Скрипт має систему сповіщень про активні DDoS-атаки та захист від ботнетів.

Він використовує аналізатор сигнатур для розпізнавання ботів.

Сценарій написаний на мові сценаріїв оболонки.

Встановіть і налаштуйте DDoS Deflate:

wget http://www.inetbase.com/scripts/ddos/install.sh

(сайт проекту http://deflate.medialayer.com/)

chmod 0700 install.sh./install.sh

Конфігураційний файл для DDoS Deflate називається ddos.conf, за замовчуванням він виглядає так:

FREQ=1
NO_OF_CONNECTIONS=50
APF_BAN=1
KILL=1
EMAIL_TO=”root”
BAN_PERIOD=600

Налаштування iptables. У цьому пункті представлено правила безпеки, складені для сервера LAMP.

Стандартна політика ланцюгів:

iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP

Скидання існуючих схем:

iptables -F
ANTI - SYN FLOOD:
iptables -A INPUT -p tcp --dport 80 --syn -m limit -- 1/s -j ACCEPT

Блокування етапу SYN (не більше 10 SYN):

iptables -A INPUT -p tcp --syn --dport 80 -m iplimit --iplimit-above 10\ -j DROP

Поставте прапорець "New not syn":

iptables -A bad_tcp_packets -p tcp --dport 80 !--syn -m state -- NEW \-j LOG --log-prefix "New not syn:"
iptables -A bad_tcp_packets -p tcp --dport 80 !--syn -m state --state NEW \-j DROP

ANTI PING OF DEAD:

iptables -A INPUT -p icmp --icmp-type echo-request -m limit\
--limit 1/s -j ACCEPT

Захист від прихованого сканування портів (ANTI - PORT SCANNER):

iptables -A INPUT -p tcp --tcp-flags SYN, ACK, FIN, RST \
-m limit --limit 1/s -j ACCEPT
iptables -A INPUT -p tcp --tcp-flags ALL SYN, ACK -j DROP

Припускаючи, що вже пройдено 400 запитів, наступні відхиляються, якщо більше 300 за секунду:

iptables –A INPUT -d $IP_web -p tcp --dport 80 -m state \
--state NEW -m limit --limit 300/c --limit-burst 400 -j DROP

Максимум 10 одночасних підключень до порту 80 з одного IP:

iptables -A INPUT -p tcp --dport 80 -m iplimit --iplimit-above 10 -j DROP

Обмеження – 12 підключень в секунду для інтерфейсу eth0, з максимально дозволеною кількістю 24:

iptables --new-chain car

```
iptables --insert OUTPUT 1 -p tcp --destination-port 80 -o eth0 --jump car
iptables --append car -m limit --limit 12/sec --limit-burst 24 --jump RETURN
iptables --append car --jump DROP
```

20 сітчастих з'єднань класу С:

```
iptables -I INPUT -p tcp --dport 80 -m iplimit --iplimit-above 20 --iplimit-mask\ 24 -j DROP
```

Щоб дозволити будь-який вихідний трафік:

```
iptables –A INPUT –m state --state ESTABLISHED, RELATED –j ACCEPT
```

Видалити підроблені пакети, позначені як Bad Guy:

```
iptables -A INPUT -m recent --rcheck --seconds 60 -m limit --limit 10/sec \
-j LOG --log-префікс "BG"
iptables -A INPUT -m recent --update --seconds 60 -j DROP
iptables -A INPUT -i $int_if -s $int_ip -m recent --set -j DROP
```

Видаліть відомі віруси та сканери портів:

```
iptables -A INPUT -i $int_if -m multiport -p tcp --dports \
53,113,135,137,139,445 -j DROP
iptables -A INPUT -i $int_if -m multiport -p udp --dports \
53,113,135,137,139,445 -j DROP
iptables -A INPUT -i $int_if -p udp --dport 1026 -j DROP
iptables -A INPUT -i $int_if -m multiport -p tcp --dports 1433,4899 -j DROP
```

На рисунку 1 представлена блок-схема алгоритму, що реалізується в модулі netfilter для забезпечення безпеки мережі від DDoS-атак, шляхом моніторингу та обмеження вхідного трафіку. Алгоритм включає кілька ключових етапів, які детально розглянемо далі.

Ініціалізація (Початок): Алгоритм розпочинається з ініціалізації всіх необхідних параметрів та змінних для подальшої роботи.

Перевірка портів (dport 80, 443): Першим кроком реалізується filtr та аналіз TCP-пакетів щодо цільових портів 80 (HTTP) та 443 (HTTPS). Ця перевірка є критично важливою, оскільки ці порти часто стають мішенню атак.

Контроль SYN-запитів (syn limit 1/s): Визначення кількості SYN-запитів, що надходять, не повинно перевищувати 1 запит за секунду. Це дозволяє уникнути перевантаження серверу в результаті атаки SYN Flood.

Перевірка обмежень IP (iplimit above 10): Пільги на кількість з'єднань з одного IP-адреси не повинні перевищувати 10, що є важливим етапом для виявлення потенційних зловмисників.

Обмеження NEW-з'єднань (NEW limit 300/s, burst 400): Додатковий контроль за встановленням нових з'єднань. Кількість нових з'єднань не повинна перевищувати 300 за секунду, з можливістю сплеску до 400. Це дозволяє підтримувати нормальний рівень трафіку та запобігти різким стрибкам.

Синхронізація IP-ліміту (iplimit above 10): Повторна перевірка IP-ліміту для підтвердження, що зловмисник не перевищує дозволений обсяг з'єднань.

Контроль маски підмережі (mask 24): Установлення маски підмережі (24) для виявлення та блокування клієнтів із фіксованих сегментів, що може зменшити ризик розповсюдження атак.

Дії щодо пакетів (DROP або ACCEPT): У залежності від результатів всіх попередніх перевірок, пакети можуть бути або прийняті (ACCEPT), або відхилені (DROP).

Завершення процесу (stop): Завершення виконання алгоритму, при цьому важливо зафіксувати результати обробки.

Дизайн даного алгоритму надає структуровану та ефективну архітектуру для обробки и фільтрації TCP-пакетів. Завдяки поступовій перевірці параметрів з'єднання, алгоритм здатний захистити сервери від широкого спектра DDoS-атак, забезпечуючи надійну і стабільну роботу мережі. Рекомендується провести подальші дослідження щодо оптимізації даного підходу, враховуючи динаміку мережевого трафіку та нові методи атаки.

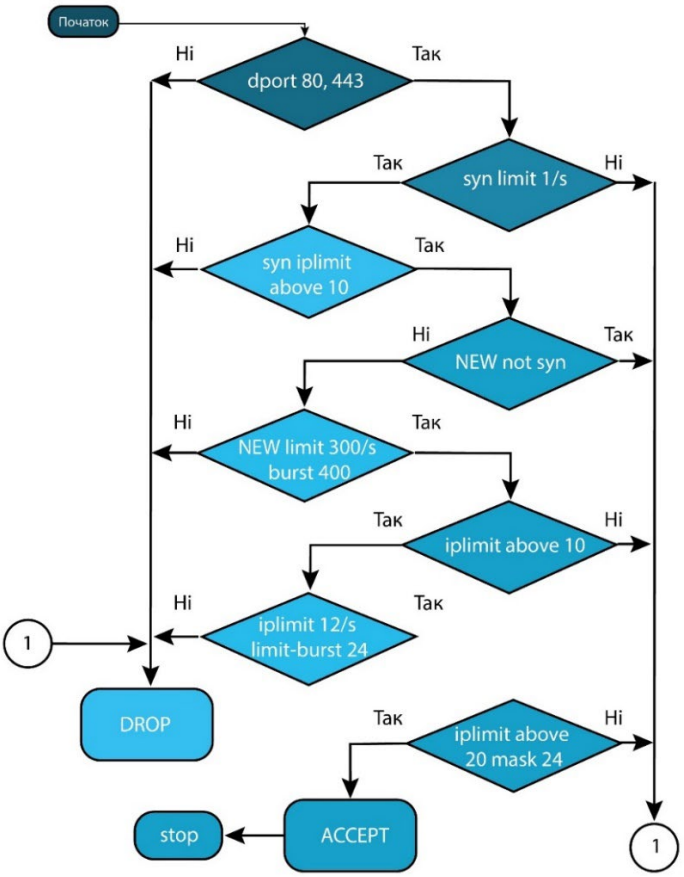

Рис 1. Блок-схема алгоритму обробки TCP-пакетів за допомогою netfilter для захисту від DDoS-атак

На рисунку 2 показані значення ресурсів, зайнятих віртуальною машиною.

```
[root@server1 cache]# vzctl exec 2 cat /proc/user_beancounters
02          Version: 2.5
03 uid  resource          held    maxheld    barrier      limit    failcnt
04 101:kmemsize        1508202    1661695   11055923   11377049          0
05   lockedpages             0          0        256        256          0
06   privvmpages          5430       7102      65536      69632          0
07   shmpages              381        381      21504      21504          0
08   dummy                   0          0          0          0          0
09   numproc                19         21        240        240          0
10   physpages            2489       2775          0 2147483647          0
11   vmguarpages             0          0      33792 2147483647          0
12   oomguarpages         2489       2775      26112 2147483647          0
13   numtcpsock              5          5        360        360          0
14   numflock                3          4        188        206          0
15   numpty                  0          1         16         16          0
16   numsiginfo              0          2        256        256          0
17   tcpsndbuf           44720          0    1720320    2703360          0
18   tcprcvbuf           81920          0    1720320    2703360          0
19   othersockbuf        13144      14356    1126080    2097152          0
20   dgramrcvbuf             0       8380     262144     262144          0
21   numothersock           11         13        120        120          0
22   dcachesize              0          0    3409920    3624960          0
23   numfile               503        531       9312       9312          0
24   dummy                   0          0          0          0          0
25   dummy                   0          0          0          0          0
26   dummy                   0          0          0          0          0
27   numiptent              10         10        128        128          0
28 [root@server1 cache]#
```

Рис. 2. Перегляд лічильників (memcached) за допомогою OpenVZ

Колона failcnt містить лише нулі. Інакше це означатиме, що встановлених ресурсів недостатньо в даний момент часу. Збільшення відповідного ресурсу здійснюється у

відповідному конфігураційному файлі (*. conf) у каталозі /etc/vz/conf, потім перезавантажте систему командою:

vzctl перезапуск 101.

Експериментальні результати. Було проведено ряд експериментів, щоб оцінити відмову в обслуговуванні з (і без) використанням системи безпеки, наслідок заповнення веб-сервера пакетами TCP SYN.

Для забезпечення чистоти експерименту автор вибрав варіант підключення без проміжних маршрутизаторів, оскільки адміністратор кожного проміжного маршрутизатора самостійно встановлює правила фільтрації для своїх мереж, що вплине на результати.

Я підключився до локальної мережі за допомогою 16-портового веб-сервера комутатора та 15 машин, які використовуються як зловмисниками, так і клієнтами в експерименті. Модель вимикача це Cisco Linksys SR2016T-EU16-портовий гігабітний комутатор 10/100/1000 (SR2016T-EU) [13].

16 портів перемикаються зі швидкістю 32 Гбіт/с. Пропускна здатність становить 23,8 млн пакетів в секунду.

Конфігурація хостів (клієнтів і зловмисників) для взаємодії з HTTP-сервером:

- Процесор: Intel® Celeron® 2,00 ГГц
- Частота процесора: 2,00 ГГц
- Оперативна пам'ять: 512 Мб
- Мережева карта: 100 Мбіт/с Ethernet
- ОС: CentOS Linux версія 6.0; Linux версія 2.6.32-71.29.1.el6.i686 gcc версія 4.4.4 20100726 (Red Hat 4.4.4-13)

На рисунку 3 показана топологія експериментальної мережі, яка складається з атакуючих хостів, веб-клієнта, веб-сервера, з'єднаних через комутатор Cisco. Вона ілюструє взаємодію елементів у процесі аналізу мережевих атак.

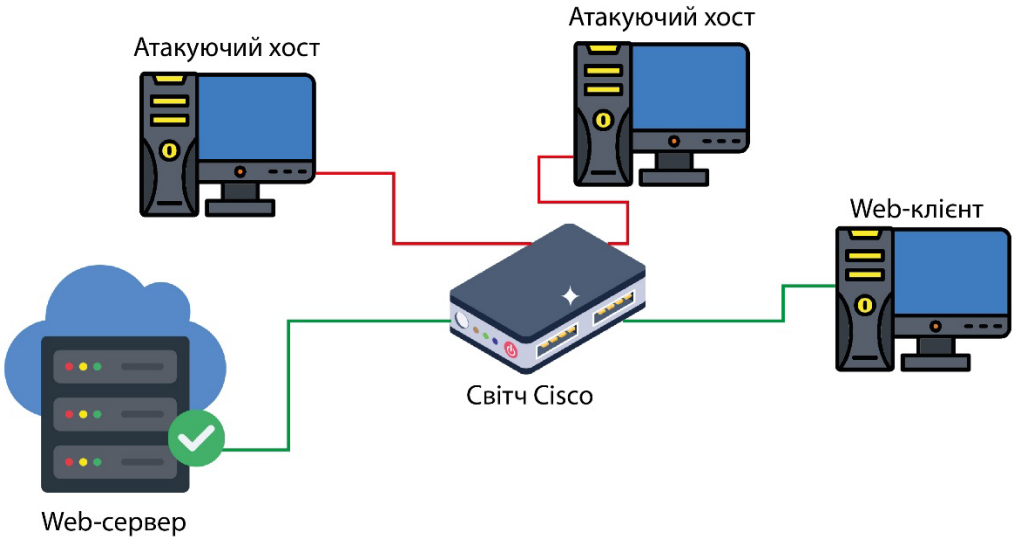

Рис. 3. Топологія експериментальної мережі

Для вимірювання трафіку використовувалася програма IPTraf, яка була запущена на веб-сервері та вимірювала вхідний і вихідний трафік у пакетах за секунду (packets/sec). Трафік, зареєстрований IPTraf, пульсує, тому в таблиці наведені середні значення.

Комп'ютери, визначені як атакуючі хости, генерують пакети TCP SYN, імітуючи SYN-флуд. Для цього використовується програма, яка без захисту системи надсилає на веб-сервер з одного хосту, а веб-сервер реєструє, що отримує 47 тисяч пакетів за секунду, встигаючи відповісти лише на 11 тисяч пакетів. в секунду. Це свідчить про те, що система має ресурс, щоб відповісти менше ніж на чверть отриманих запитів. Коли вони атакують два хости, у нього знову закінчуються ресурси — він бере 32 Кб і повертає 7 Кб. Це вказує на те, що системний

ресурс виснажений, у нас працює TCP SYN flood (DDoS-атака).

При використанні системи безпеки аналізується вплив параметра «розмір черги напіввідкритих з'єднань»:

net.ipv4.tcp_max_syn_backlog (B), який за замовчуванням приймає значення 1024B.

Для законного трафіку це значення має забезпечувати максимально якісне обслуговування для певної кількості клієнтів (наприклад, 400 за секунду).

Таблиця 1

Вплив параметра розміру черги на напіввідкриті з'єднання

net.ipv4.tcp_max_syn_backlog, B	1024	2048	4096	8191
вхід трафік, пакети/с	16000	20000	25000	25000
вихід трафік, пакети/с	4400	5000	5500	5500

При заповненні пакетами та збільшенні цього буфера система здатна обробляти більше трафіку, тобто. він має більше ресурсів, щоб протистояти атаці, і фактично вдається обслуговувати більше трафіку.

Коли у нас є той самий тип даних зі значеннями параметрів 4096 і 8192, робиться висновок, що більше немає необхідності збільшувати буфер (таблиця 1).

Тому згодом було вибрано значення:

net.ipv4.tcp_max_syn_backlog=4096.

Проаналізуємо вплив параметра net.ipv4.tcp_synack_retries при використанні захисту.

Параметр tcp_synack_retries контролює кількість повторних передач, встановлюючи час збереження напіввідкритих з'єднань у буфері. За замовчуванням 5 відліків, що означає видалення напіввідкритого з'єднання через 3 хвилини. Дані наведені в таблиці 2.

Таблиця 2

Вплив параметра tcp_synack_retries на час передачі
та повний час напіввідкритих з'єднань

tcp_synack_retries значення, кількість	**Час повторної передачі t, с**	**Загальний час зберігання напіввідкритих з'єднань у черзі, с**
1	на 3-й секунді	9 секунд
2	на 3-й і 9-й секундах	21 секунда
3	на 3, 9 і 21 сек.	45 секунд
4	3-та, 9-та, 21-ва, 45-та сек.	90 секунд
5	3-та, 9-та, 21-ва, 45-та, 90-та	180 секунд

Таблиця 3

Вплив параметра, що контролює кількість повторних передач

net.ipv4.tcp_synack_retries, с	**5**	**4**	**3**	**2**	**1**
вхід. трафік, пакети/с	21000	22000	23000	24000	25000
вихід. трафік, пакети/с	4500	4650	4800	5000	5500

Коригування зроблено так, що передача реалізується на третій секунді, а загальний час для збереження напіввідкритих з'єднань у черзі становить 9 секунд. При заповненні пакетами і зменшенні кількості повторних передач, що призводить до скорочення часу утримання напіввідкритих з'єднань в черзі, робиться висновок, що система здатна обробляти більший трафік. Тому згодом було вибрано значення net.ipv4.tcp_synack_retries=1. Дані наведені в таблиці 3.

Експериментально виявляється, що без захисту сервер стає непрацездатним при атаці з боку 7 хостів, а з використанням системи, яка залишається працездатною навіть при атаці з

боку всіх 15 машин. Дані експериментального виявлення Генерування вхідного і вихідного трафіку веб-сервера без системи захисту наведено в Таблиці 4, діаграму допустимого та вхідного трафіку ви можете розглянути на рисунку 4.

Таблиця 4

Генерування вхідного і вихідного трафіку веб-сервера без системи захисту

штурмових машин, немає	**вхід. трафік, пакети/с**	**доп. трафік, пакети/с**
1	47000	11000
2	32000	7000
3	30000	6000
4	28000	5000
5	25000	4000
6	22000	3000
7	17000	1000
8	10000	900

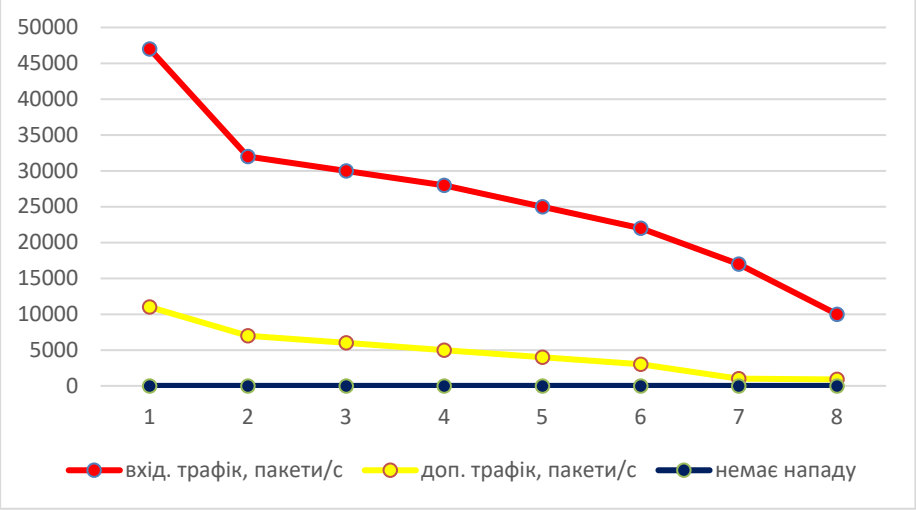

Рис. 4. Згенерований вхідний і вихідний трафік веб-сервера без системи захисту

Дані експериментального виявлення Генерування вхідного і вихідного трафіку веб-сервера з системою захисту наведено в Таблиці 5, діаграму допустимого та вхідного трафіку ви можете розглянути на рисунку 5.

Таблиця 5

Генерація вхідного і вихідного трафіку веб-сервера за допомогою системи безпеки

штурмових машин, немає	**вхід. трафік, пакети/с**	**доп. трафік, пакети/с**
1	25000	5500
2	19000	3600
3	17000	3500
4	16000	3400
5	14500	2500
6	13000	1550
7	9800	1200
8	9400	1100
9	8600	900
10	8000	850

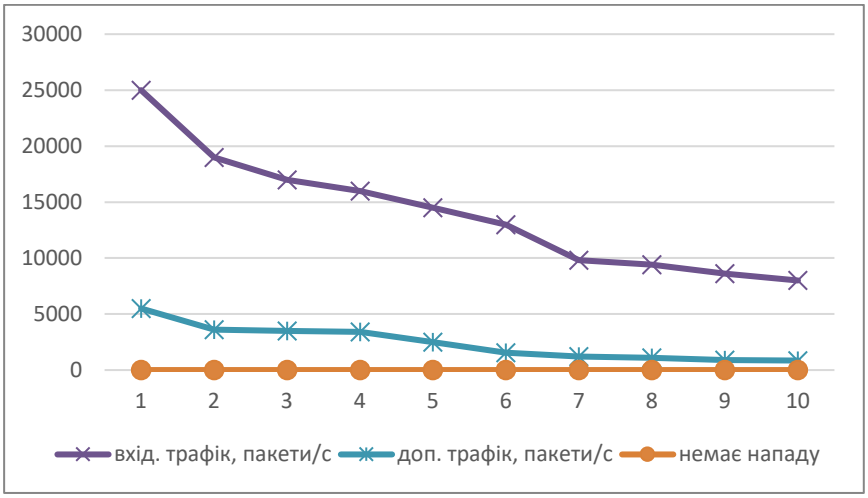

Рис. 5. Згенерований вхідний і вихідний трафік веб-сервера за допомогою Система захисту

Висновки

Розроблено та описано систему, призначену для блокування DDoS-атак.Система змінює параметри ядра ОС, основні конфігураційні файли, містить додаткові модулі. В результаті верифікації підтверджено теоретично обґрунтовані параметри, представлені в математичній моделі в розділі 3: розмір буфера та інтервал очікування встановлення з'єднання. Експериментально доведено стабільну роботу системи під час реальної DDoS-атаки; його здатність підтримувати роботу веб-сервера підтверджено. В результаті проведеного аналізу можна зробити наступні висновки:

1. Основними елементами системи безпеки є параметри ядра ОС, стек TCP/IP, скрипт iptables.
2. Сильно виражена залежність встановлених для захисту параметрів від характеристик веб-сервера (процесор, пам'ять, ОС, пропускна здатність каналу зв'язку).
3. За допомогою системи вдається підтримувати веб-сервер у робочому стані, наскільки це можливо, відносно пропускної здатності ліній зв’язку та пристроїв, які переповнюють підроблену систему. Без допомоги системи веб-сервер перестає обслуговувати клієнтів на значно нижчих рівнях потоку вхідних запитів.

Розроблено та описано систему, призначену для блокування DDoS-атак. Система змінює параметри ядра ОС, основні конфігураційні файли, містить додаткові модулі. В результаті верифікації підтверджено теоретично обґрунтовані параметри, представлені в математичній моделі в розділі 3: розмір буфера та інтервал очікування встановлення з'єднання. Експериментально доведено стабільну роботу системи під час реальної DDoS-атаки; його здатність підтримувати роботу веб-сервера підтверджено.

Список використаної літератури:

1. Using the Latest Methods of Cluster Analysis to Identify Similar Profiles in Leading Social Networks. Bohdan Zhurakovskyi, Ihor Averichev and Ivan Shakhmatov. Information Technology and Implementation (Satellite) Conference Proceedings, 21 November, 2023. URL: https://ceur-ws.org/Vol-3646/Paper_12.pdf

2. Корченко А. Методи ідентифікації аномальних станів для систем виявлення вторгнень. Монографія, Київ, ЦП «Компринт», 2019. URL: https://nubip.edu.ua/sites/default/files/u34/monografiya_korchenko_anna_1.pdf

3. С. Казмірчук, А. Корченко, Т. Паращук, «Аналіз систем виявлення вторгнень», Захист інформації, Т.20, №4, с. 259-276, 2018. URL: https://doi.org/10.18372/2225-5036.24.13431.

4. І. Терейковський, А. Корченко, Т. Паращук, Є. Педченко, «Аналіз відкритих систем виявлення вторгнень», Безпека інформації. Т.24, №3, с. 201-216, 2018. URL:

https://doi.org/10.18372/2225-5036.24.13431

5. Юдін О. К., Коновалов Е. О., Рогоза І. Є. Методи виявлення атак до інформаційних ресурсів автоматизованих систем. Ukrainian information security research journal. 2010. Т. 12, № 2 (47). URL: https://doi.org/10.18372/2410-7840.12.1940.

6. Гончаренко М.С. Кіберзахист: основи аналізу. Львів: Видавництво ЛНУ, 2020.

7. Using machine learning to classify DOS/DDOS attacks / M. S. Kavetskyi et al. Radiotekhnika. 2024. No. 217. P. 55–63. URL: https://doi.org/10.30837/rt.2024.2.217.04.

8. Savchenko V. A. Diagnosing the start of a slow HTTP DDoS attack based on two-parameter traffic correlation analysis. Telecommunication and Information Technologies. 2021. Vol. 73, no. 4. URL: https://doi.org/10.31673/2412-4338.2021.042840.

9. Chornobuk M., Dubrovin V., Deineha L. Cybersecurity: research on methods for detecting ddos attacks. Computer systems and information technologies. 2023. No. 4. p. 6–9. URL: https://doi.org/10.31891/csit-2023-4-1.

10. Lunhol O. Overview of cybersecurity methods and strategies using artificial intelligence. Cybersecurity: education, science, technique. 2024. Т. 1, № 25. С. 379–389. URL: https://doi.org/10.28925/2663-4023.2024.25.379389.

11. Гайдук С.П. Мережевий аналіз трафіку. Харків: Видавництво ХНУРЕ, 2022.

12. AWS Shield Whitepaper: Advanced DDoS Protection. Amazon Web Services, 2023.

13. Opanasenko M. I. The technology of ensuring cyber security of the cloud environment based on the Cisco Cloudlock solution. Modern information security. 2023. Vol. 53, no. 1. URL: https://doi.org/10.31673/2409-7292.2023.010010.

14. Lockhart A. Network security hacks. O'Reilly Media, Incorporated, 2006.

Автори статті

Поночовний Петро – аспірант, Державний університет інформаційно-комунікаційних технологій, Київ, Україна.

ORCID: 0009-0008-6480-6990

Іванченко Ігор – кандидат технічних наук, доцент, Державне некомерційне підприємство «Державний Університет "Київський Авіаційний Інститут"», Київ, Україна.

ORCID: 0000-0003-3415-9039

Authors of the article

Ponochovny Petro – postgraduate, State University of Information and Communication Technologies, Kyiv, Ukraine.

ORCID: 0009-0008-6480-6990

Ivanchenko Ihor – Candidate of Science (technic), Associate Professor, State Non-Commercial Enterprise State University "Kyiv Aviation Institute", Kyiv, Ukraine.

ORCID: 0000-0003-3415-9039