ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ВСТУП

Актуальність дослідження. Забезпечення безпеки та доступності веб-додатків – це важлива складова підтримки життєдіяльності будь-якого веб-додатку. З збільшенням загроз зі сторони зловмисників, необхідно докладати все більше і більше зусиль, аби підтримувати на високому рівні захищеність веб-додатків.

Все більше і більше компаній переходять на модель веб-додатків, перекладаючи на себе більше зобов’язань перед користувачами, наприклад забезпечення стабільної доступності до своїх продуктів. Кожна хвилина простою завдає розробникам репутаційні та фінансові збитки. Щоб завадити цьому компанії докладають всіх зусиль, аби покращити цей параметр.

Організації, прагнучи захистити свої цифрові активи та забезпечити безперебійний онлайн-досвід для користувачів, все частіше звертаються до передових програмних рішень. Прикладом такого рішення є програмний комплекс F5 BIG-IP.

Цифровий ландшафт 21-го століття ознаменований різноманіттям кіберзагроз, які включають в себе витоки даних, DDoS-атаки, складні схеми фішингу та вимагання. Ці загрози не тільки створюють ризики для чутливих даних, але й підривають довіру та надійність веб-сервісів. У цьому середовищі безпека веб-додатків не є просто технічною вимогою, але й бізнес-імперативом.

Оскільки веб-додатки стають складнішими та невід'ємною частиною бізнес-операцій, потреба в комплексних рішеннях безпеки стає більш вираженою. F5 BIG-IP пропонує багатогранний підхід до безпеки, вирішуючи як актуальні загрози, так і розвиваючийся ландшафт кіберзагроз. Цей пакет не просто захисний механізм проти атак; це також проактивний інструмент, який підвищує продуктивність та надійність веб-додатків, забезпечуючи їх доступність та відповідність потребам користувачів.

Об’єкт дослідження – процес забезпечення безпеки та доступності веб-додатків.

Предмет дослідження – технологія забезпечення безпеки та доступності веб-додатків на базі рішення F5 BIG-IP.

Мета роботи – дослідити варіант розгортання технології забезпечення безпеки та доступності веб-додатків на базі рішення F5 BIG-IP та розробити рекомендації щодо використання технології.

Наукові завдання:

- провести аналіз питання щодо необхідності забезпечення безпеки та доступності веб-додатків;
- проаналізувати основні загрози веб-додаткам;
- проаналізувати методи та засоби забезпечення безпеки та доступності веб-додатків;
- дослідити варіант розгортання технології забезпечення безпеки та доступності веб-додатків на базі рішення F5 BIG-IP та розробити рекомендації щодо використання технології.

Методи дослідження – опрацювання літератури за даною темою, аналіз експлуатаційної документації, міжнародних стандартів та їх порівняння, моделювання процесу забезпечення безпеки та доступності веб-додатків на базі рішення F5 BIG-IP.

1 АНАЛІЗ НЕОБХІДНОСТІ ЗАБЕЗПЕЧЕННЯ БЕЗПЕКИ ТА ДОСТУПНОСТІ ВЕБ-ДОДАТКІВ ТА МЕТОДІВ ЇХ ЗАБЕЗПЕЧЕННЯ

1.1.Аналіз потенційних загроз порушення безпеки веб-додатків

З розвитком мережевих технологій веб-додатки набули неймовірної популярності та поступово витісняють інші види застосунків. Від соціальних мереж до банківських систем, від освітніх платформ до урядових сервісів - усе це функціонує завдяки веб-додаткам. Через збільшення попиту на веб-додатки, їхня кількість зростає кожного дня. А з врахуванням ускладнення їх функціоналу, не завжди враховується такий аспект, як забезпечення захисту цих додатків від внутрішніх та зовнішніх небезпек. Це призвело до виникнення нових вразливостей, що можуть нашкодити користувачам.

Веб-додаток – це програмне забезпечення або програма, яку можна відкрити за допомогою будь-якого браузера. Зовнішній інтерфейс веб програми розробляється за допомогою таких мов програмування: HTML, CSS, Javascript, які підтримуються на будь-якому браузері (Opera, Chrome, Mozilla). У той час як для написання серверної частини (Back-end) може використовуватися будь-яка інша мова програмування або фреймворк, Python, PhP, Ruby, Java.

Основними перевагами веб-додатків є:

- веб додатки можуть застосовуватися на будь-якій операційній системі (Linux, Mac, Windows), оскільки всі вони підтримують сучасні браузери;
- у зв’язку з тим, що у веб-додатку використовується той самий код порівняно з desktop додатками їх набагато легше підтримувати;
- додаток простіше програмувати оскільки він не включає багато роботи з елементами ПК(ядро, процесор, відеокарта);
- на відміну від мобільних додатків, для веб-додатків не потрібно схвалення жодних платформ, щоб випустити свою програму;

- веб додатки це більш економний варіант для будь-якого підприємства оскільки веб-програми не вимагають підписки або покупки ліцензій, а можуть використовуватися як SaaS-сервіс, що значно дешевше.

Вразливості веб-додатків можуть стати серйозною загрозою для безпеки і конфіденційності користувачів та їхньої інформації. Ці вразливості можуть бути причиною кібератак. Згідно з OWASP Top Ten, найпоширеніші вразливості можна розділити на 10 категорій:

1. Зламаний контроль доступу. Контроль доступу забезпечує дотримання політики таким чином, щоб користувачі не могли діяти поза межами призначених дозволів. Збої зазвичай призводять до несанкціонованого розголошення інформації, модифікації чи знищення всіх даних або виконання дій за межами обмежень користувача. Поширені вразливості контролю доступу включають:

- Порушення принципу найменших доступних привілеїв, або заборони за замовчуванням, де доступ має надаватися лише для певних можливостей, ролей або користувачів, однак доступний для всіх.
- Обхід перевірок контролю доступу шляхом зміни URL-адреси (підробка параметрів або примусовий перегляд), внутрішнього стану програми чи сторінки HTML або використання інструменту атаки, що змінює запити API.
- Дозвіл на перегляд або редагування чужого облікового запису шляхом надання його унікального ідентифікатора (незахищені прямі посилання на об’єкти).
- Доступ до API із відсутніми елементами керування доступом для POST, PUT і DELETE.
- Підвищення привілеїв. Діяти як користувач без входу в систему або діяти як адміністратор, коли ви ввійшли як користувач.
- Примусово переглядати автентифіковані сторінки як неавтентифікований користувач або привілейовані сторінки як звичайний користувач.

2. Криптографічні помилки. Виникнення криптографічних збоїв, або відсутність криптографічного захисту під час транспортування та зберігання даних.

Наприклад паролі, номери кредитних карток, медичні записи, персональна інформація та комерційна таємниця вимагають додаткового захисту, найчастіше через те, що підпадають під дію законів про конфіденційність.

3. Ін`єкції. Серед розповсюджених видів ін`єкції є SQL, NoSQL, OS command, Object Relational Mapping. Принцип дії однаковий для кожного інтерпретатора. Додатки вразливі, якщо:

- Інформація введена користувачем не перевіряється або не фільтрується додатком.
- Динамічні запити або виклики без параметрів використовуються безпосередньо в інтерпретаторі.
- Ворожі дані використовуються в параметрах пошуку ORM для отримання додаткових конфіденційних записів.

4. Незахищений дизайн. Ризики, пов`язані із недоліками дизайну архітектури.

5. Неправильна конфігурація безпеки. Неправильно налаштовані або взагалі не налаштовані функції чи компоненти можуть послабити захист додатку та спричинити більше шкоди. Додаток може бути вразливим, якщо:

- Ввімкнено або встановлено непотрібні функції (наприклад непотрібні порти, служби, сторінки, облікові записи або привілеї).
- Облікові записи за замовчуванням та їхні паролі залишаються активними та не змінюються.
- Обробка помилок виявляє користувачам надто інформативні повідомлення про помилки.
- Для оновлених систем не налаштовані найновіші функції безпеки.

6. Вразливі або застарілі компоненти. Використання вразливих або застарілих компонентів може призвести до використання методів атаки на вже відомі вразливості.

7. Помилки аутентифікації та ідентифікації. Підтвердження особи користувача, аутентифікація та керування сеансами мають критичне значення для захисту від атак, пов`язаних з аутентифікацією. Аутентифікація може бути ненадійною, якщо додаток:

- Дозволяє автоматизовані атаки, як перебір облікових даних, коли зловмисник має список дійсних імен користувачів і паролі.
- Дозволяє атаку грубою силою, або інші автоматизовані атаки.
- Дозволяє стандартні, слабкі або добре відомі паролі.
- Використовує слабкі або неефективні процеси відновлення облікових даних і забутих паролів.
- Використовує неефективну багатофакторну автентифікацію, або взагалі не використовує її.
- Розкриває ідентифікатор сеансу в URL-адресі.
- Повторне використання ідентифікатора сеансу після успішного входу.

8. Порушення цілісності інформації та програмного забезпечення. Ці порушення відносяться до програмного коду та інфраструктури, які не були належним чином захищені від втручання. Прикладом цього є ситуація, коли програма використовує плагіни, бібліотеки або модулі з ненадійних джерел, сховищ і мереж доставки вмісту. Незахищений канал може призвести до несанкціонованого доступу, виконання зловмисного коду або компрометації системи. Зрештою, багато програм тепер включають функцію автоматичного оновлення, коли оновлення завантажуються без достатньої перевірки цілісності та застосовуються до попередньо довіреної програми. Зловмисники потенційно можуть завантажити власні оновлення для розповсюдження та запуску на всіх пристроях.

9. Проблеми логування та моніторингу. Без логування та моніторингу порушення не можуть бути виявлені. Недостатність логування, виявлення вразливостей, моніторингу та активних дій виникають щоразу, як:

- Події, які підлягають аудиту, такі як входи в систему, невдалі входи та транзакції на великі суми не реєструються.
- Попередження та помилки створюють неадекватні, незрозумілі або взагалі не створюють повідомлень в журналі логування.
- Журнали логування програм та API не перевіряються на наявність підозрілої активності.
- Журнали зберігаються лише локально.
- Порогові значення сповіщень та процеси реагування відсутні або не діють.
- Тестування на проникнення та сканування інструментами динамічного тестування безпеки додатків не викликають попереджень.
- Програма не може виявляти, реагувати або сповіщати про атаки в режимі реального часу, або майже в режимі реального часу.

10. Підробка запитів зі сторони сервера. Ця проблема виникає, коли веб-додаток отримує віддалений ресурс без перевірки посилання, наданого користувачем. Це дозволяє зловмиснику змусити програму надіслати створений запит до неочікуваного адресата, навіть якщо він захищений брандмауером, VPN або іншим типом списку контролю доступу до мережі (ACL). Оскільки сучасні веб-додатки надають кінцевим користувачам зручні функції, отримання URL-адреси стає звичайним сценарієм. Як наслідок, кількість випадків використання підробки запитів зі сторони серверів зростає. Крім того, серйозність цієї вразливості стає вищою через хмарні сервіси та складність архітектури.

1.2 Аналіз необхідності забезпечення доступності веб-додатків та методів її реалізації

Однією з головних переваг веб-додатків є їхня доступність з будь якого пристрою. Однак трапляються ситуації, коли додаток стає недоступним. Це призводить до фінансових та репутаційних втрат власників додатку з однієї сторони.

З іншої сторони користувач не отримає бажаного результату, що також може призвести як до фінансових втрат, так і до інших неприємних наслідків.

Отже розробник зі своєї сторони повинен прикласти всіх зусиль, аби користувач не мав проблем з доступом до додатку.

Доступність сайту означає, що веб-додаток має бути доступним щоразу, коли хтось до нього звертається через браузер або мобільний додаток. Доступність веб-програми може бути безпосередньо пов'язана з часом роботи веб-сервера та залежних серверів, якими може бути база даних, сервер додатків тощо.

Говорячи про доступність веб-додатку, зазвичай посилаються на значення відношення часу коли додаток доступний до загального часу. Це значення вимірюється в відсотках, як наприклад 99.9% доступності. Залежно від необхідної точності вимірювання можна використовувати розрахунок на основі годин, хвилин, секунд чи мілісекунд.

Веб-додаток з однією годиною простою в рік матиме значення доступності в 99.9%.

$$\big((8760 \text{ годин} - 1 \text{ година}) \div 8760 \text{ годин}\big) \cdot 100 = 99.9\%$$

99.9% часу доступності, з однією годною простою в рік є чудовим результатом для більшості додатків. Однак деякі додатки, наприклад Google, перевершує 99.999% часу доступності, з часом простою в 5.26 хвилин на рік.

Серед можливих причин виникнення недоступності веб-додатків можуть бути:

- Помилки серверу. Сервер може не відповідати, не даючи користувачу підтвердити або отримати інформацію з бази даних. Таку поведінку сервера можуть спричинити проблеми з пам'яттю, мережеві проблеми або інша помилка. Для уникнення таких інцидентів слід запровадити моніторинг мережі та налаштувати програмне забезпечення для сповіщення , які будуть слідкувати за станом життєдіяльності системи та завчасно попереджати про виниклі збої.
- Помилки програмного забезпечення. Ці помилки включають збої в операційній системі або веб серверах. Веб-додатки не можуть коректно працювати на

пошкодженій оперативній системі, а також веб-сервер не функціонуватиме, якщо він крашнувся, або потребує перезапуску або переналаштування чи встановлення заново.

- Помилки апаратного забезпечення. Ці помилки включають збої жорстких дисків чи інших накопичувачів, вихід з ладу процесорів або мережевих карт. Процес діагностики та виправлення цих проблем може зайняти багато часу, під час якого веб-додаток буде недоступним. Можливим рішенням є налаштування додаткових серверів, які будуть дублювати та в разі чого замінювати основний сервер.

Для забезпечення доступності веб додатків використовують мережевий балансер навантажень, який розподіляє мережеве навантаження між серверами додатку. Балансери навантаження це рішення яке діє як проксі-сервер та розподіляє мережевий трафік або трафік додатку між серверами. Балансери використовують для розподілу потужності під час пікового навантаження та для підвищення надійності додатків. Вони покращують загальну продуктивність, зменшуючи навантаження на окремі служби, і розподіляють попит між різними обчислювальними територіями.

Серед переваг мережевих балансерів навантаження можна виділити:

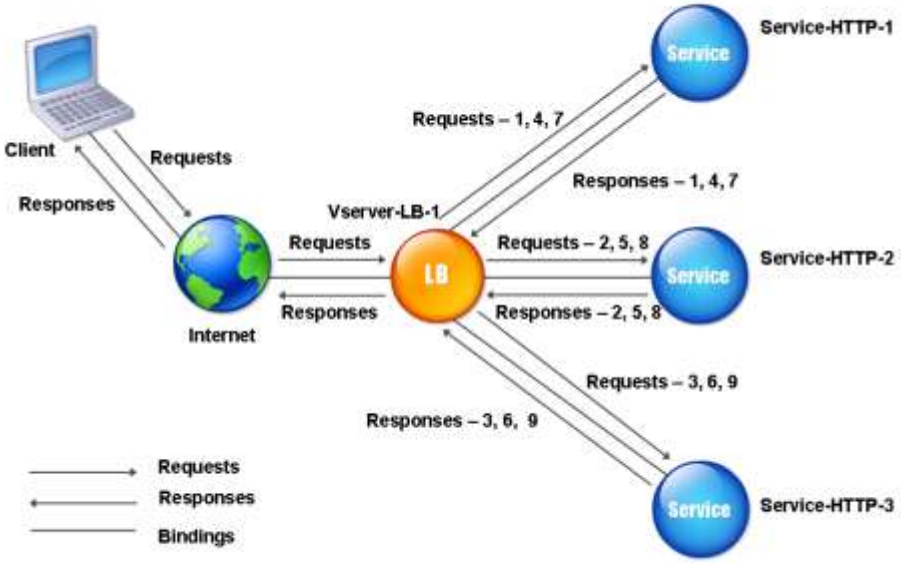
- Доступність програми: як внутрішні, так і зовнішні користувачі повинні мати можливість покладатися на доступність програми. Якщо програма чи функція не працюють, відстають або зависають, втрачається дорогоцінний час і з’являється потенційне джерело недовіри, яке може підштовхнути клієнта до конкурента.
- Масштабованість додатка: за необхідності збільшення розрахункових потужностей можна використати мережевий балансер для розподілу навантаження та збільшення пропускної спроможності додатку.
- Безпека програм: балансування навантаження також дозволяє організаціям масштабувати свої рішення безпеки. Одним із основних способів є розподіл трафіку між кількома серверними системами, що допомагає мінімізувати площу атаки та ускладнює виснаження ресурсів і перенасичення посилань. Балансери навантаження також можуть перенаправляти трафік на інші системи, якщо одна система вразлива або скомпрометована. Крім того, балансери навантаження можуть

запропонувати додатковий рівень захисту від атак DDoS атак, перенаправляючи трафік між серверами, якщо певний сервер стає вразливим.

- Продуктивність програми. Виконуючи все вищезазначене, балансер навантаження підвищує продуктивність програми. Підвищуючи безпеку, оптимізуючи час безвідмовної роботи та забезпечуючи масштабованість у разі різкого зростання попиту, балансери навантаження забезпечують роботу додатків відповідно до запланованих потреб.

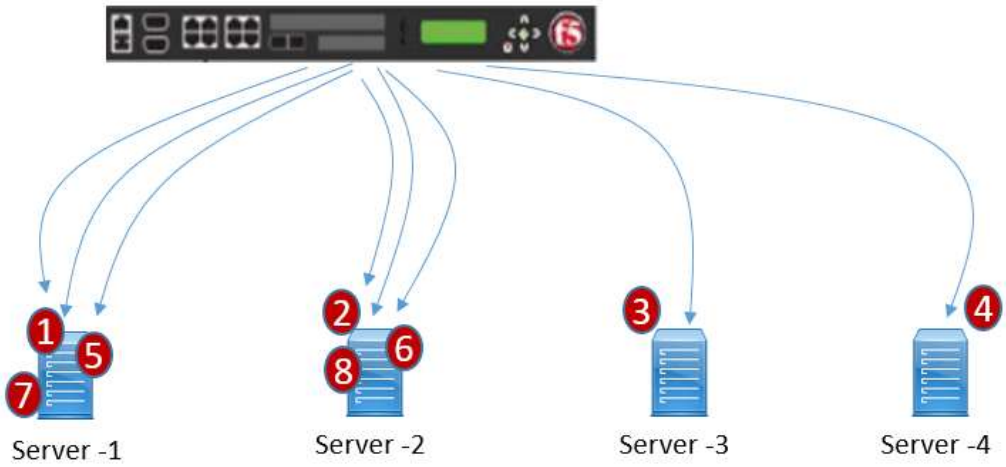
Серед можливих способів балансування трафіку можна виділити наступні:

- Round Robin: система передає кожен новий запит на з'єднання на наступний сервер у черзі, зрештою рівномірно розподіляючи з'єднання між масивом машин, навантаження на які балансується. Цей метод добре працює в більшості конфігурацій, особливо якщо обладнання, яке ви балансуєте, має приблизно однакову швидкість обробки та пам'ять.



Рис

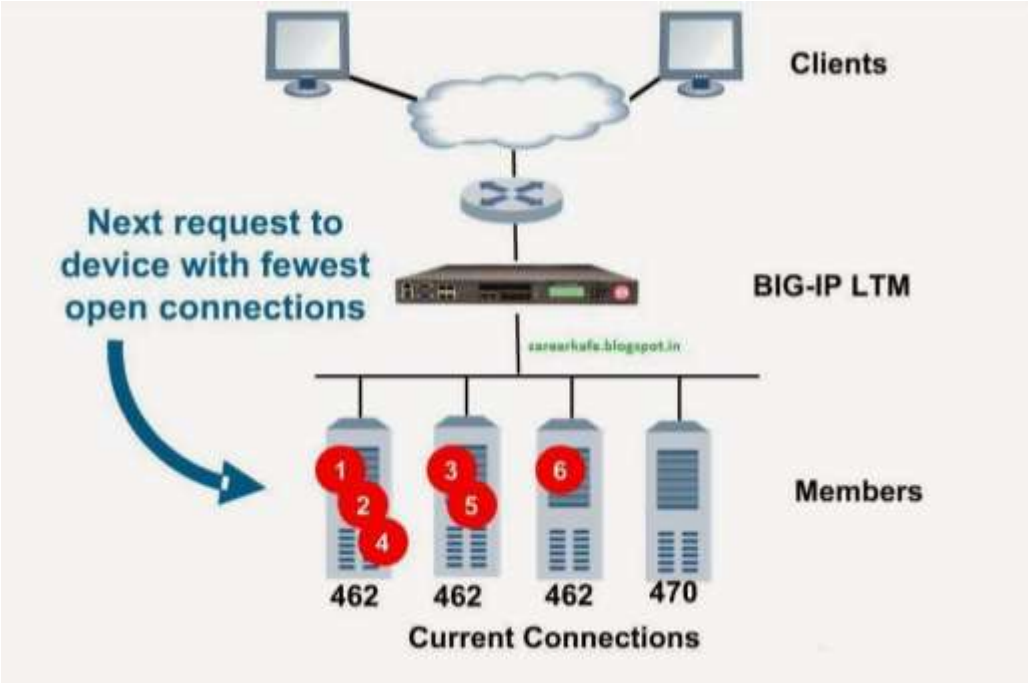
- Співвідношення: кількість підключень, які отримує кожна машина з часом, пропорційна значенню співвідношення, яке визначається для кожної машини в пулі.



Ratio			
Server -1	Server -2	Server -3	Server-4
3	3	1	1

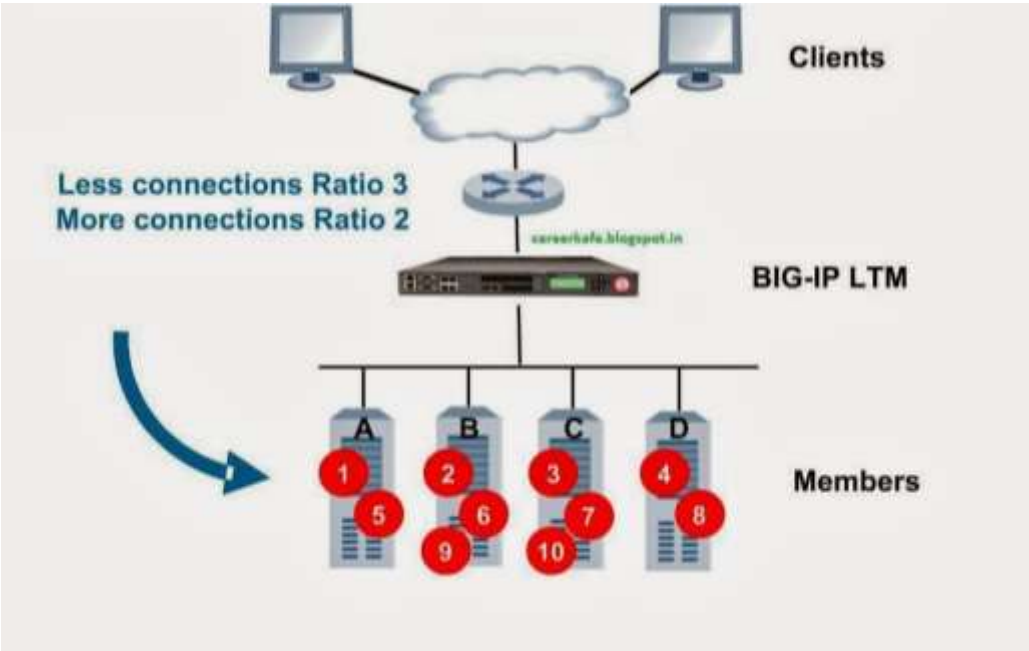
Рис

- Найменше підключень: система передає нове підключення до вузла, який має найменшу кількість поточних підключень у пулі. Цей метод найкраще працює в середовищах, де сервери чи інше обладнання, навантаження на яке балансується, має подібні можливості. Це метод динамічного балансування навантаження, який розподіляє підключення на основі різних аспектів аналізу продуктивності сервера в реальному часі, наприклад поточної кількості підключень на вузол або найшвидшого часу відповіді вузла.

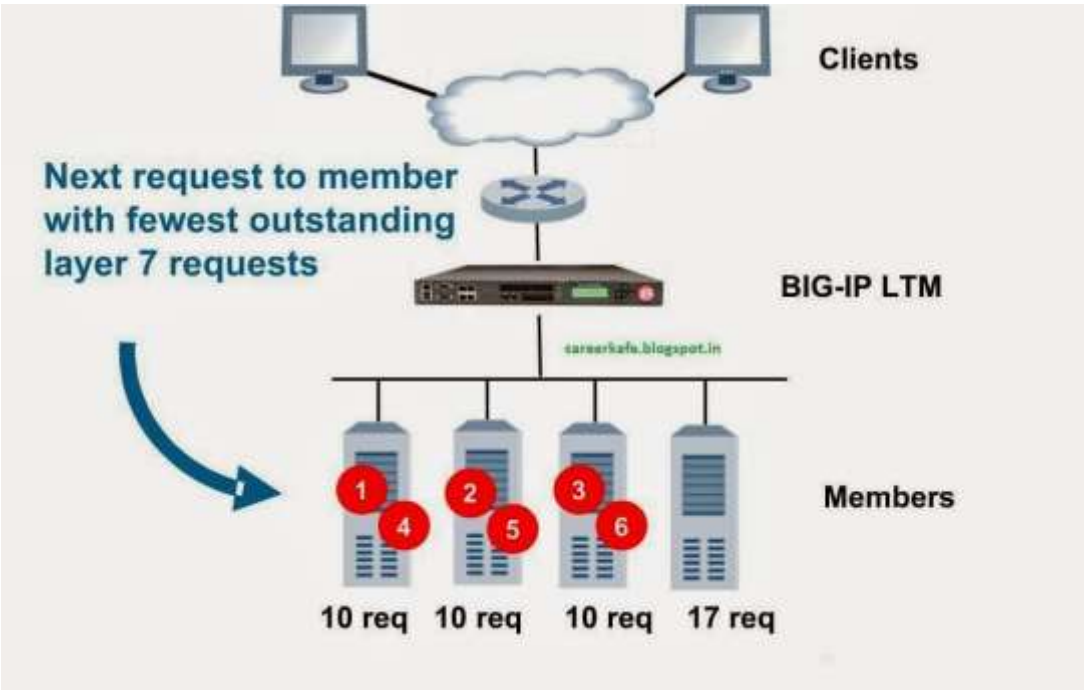


Рис

- Observed: система розбиває на ранги вузли на основі кількості з’єднань. Вузли, які мають кращий баланс найменшої кількості з’єднань, отримують більшу частку з’єднань. Цей метод відрізняється від методу найменших підключень тим, що метод найменших підключень вимірює підключення лише в момент балансування навантаження, тоді як метод спостережень відстежує кількість підключень рівня 4 до кожного вузла протягом певного часу та створює коефіцієнт для балансування навантаження. Цей метод динамічного балансування навантаження добре працює в будь-якому середовищі, але може бути особливо корисним у середовищах, де продуктивність вузлів значно відрізняється.



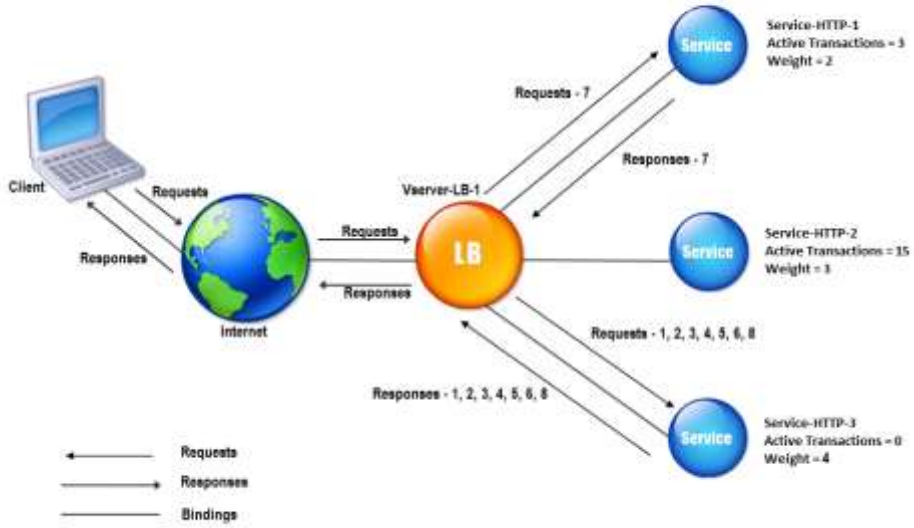
- Predictive: використовує метод ранжирування, який використовується методами спостереження, за винятком того, що система аналізує тенденцію ранжирування з часом, визначаючи, покращується чи знижується продуктивність вузла. Вузли в пулі з кращим рейтингом продуктивності, які наразі покращуються, а не знижуються, отримують більшу частку підключень. Цей метод динамічного балансування навантаження добре працює в будь-якому середовищі.
- Найшвидший: система пропускає нове з’єднання на основі найшвидшої відповіді з усіх пулів, членом яких є сервер. Цей метод може бути особливо корисним у середовищах, де вузли розподілені між різними логічними мережами.



Рис

- Динамічний коефіцієнт: цей метод схожий на режим співвідношення, за винятком того, що коефіцієнти базуються на постійному моніторингу серверів і тому постійно змінюються. Це метод динамічного балансування навантаження, який розподіляє підключення на основі різних аспектів аналізу продуктивності сервера в реальному часі, наприклад кількості поточних підключень на вузол або найшвидшого часу відповіді вузла.

- Найменша кількість сеансів: система передає нове підключення до вузла, який наразі має найменшу кількість постійних сеансів. Цей метод найкраще працює в середовищах, де сервери чи інше обладнання, яке балансується, має подібні можливості. Це метод динамічного балансування навантаження, який розподіляє підключення на основі різних аспектів аналізу продуктивності сервера в реальному часі, наприклад кількості поточних сеансів. Використання цього методу балансування навантаження вимагає, щоб віртуальний сервер посилався на тип постійного профілю, який відстежує постійні підключення.



Рис

Балансер навантаження працює шляхом статичної або динамічної відповіді на запит користувача та розповсюдження цього запиту на один із внутрішніх серверів, здатних виконати запит. Якщо один із серверів виходить з ладу, балансер навантаження перенаправляє трафік на інші онлайн-сервери.

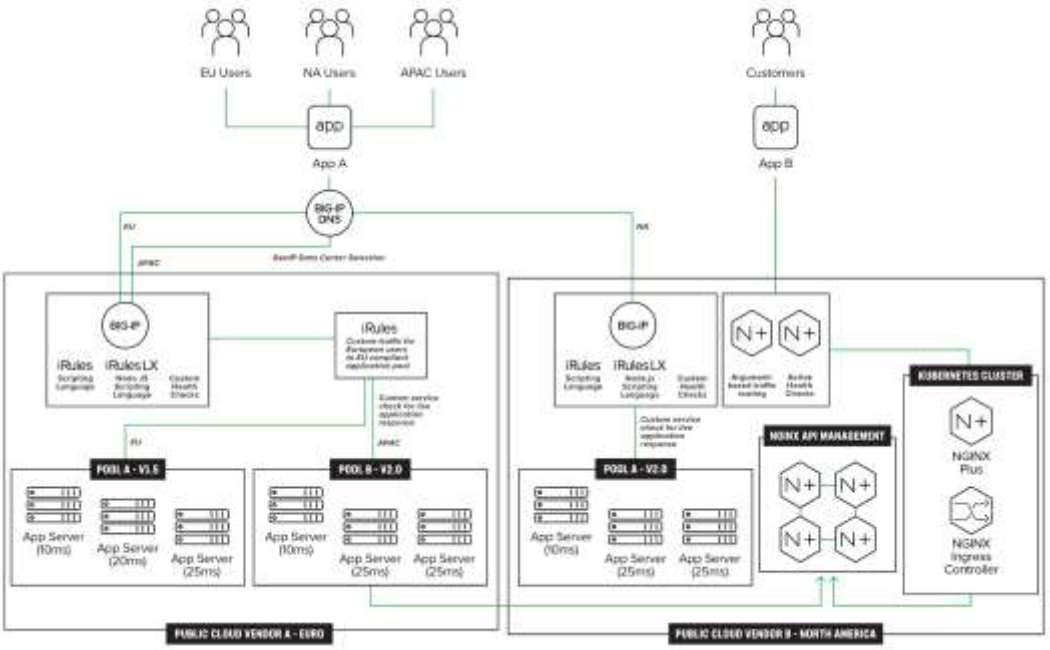


Рис 1.1.

1.3 Аналіз технології WAF для забезпечення безпеки веб-додатків

Для захисту веб-додатків від різних загроз можна використовувати Web Application Firewall (WAF).

Фаєрвол веб-додатків (WAF) — це особлива форма фаєрвола, яка фільтрує, відстежує та блокує трафік HTTP до веб-додатку та з нього. Перевіряючи HTTP-трафік, він може запобігти атакам, які використовують відомі вразливості веб-додатків, такі як впровадження SQL, міжсайтовий сценарій (XSS), включення файлів і неправильна конфігурація системи.

Головною перевагою WAF є послідовний захист готових, діючих веб-додатків на рівні програми, без необхідності змінювати сам додаток.

З однієї сторони, WAF пропонує базовий захист від відомих атак або вразливостей за принципом чорного списку. Наприклад стандарт безпеки індустрії кредитних карток (PCI DSS) передбачає використання WAF, як альтернативу регулярним перевіркам коду спеціалістом, як адекватний захід захисту веб-додатків.

Використання WAF стає особливо актуальним у випадку, наприклад, конкретних вразливостей, виявлених під час проведення тестів на проникнення або огляду вихідного коду. Навіть якби вдалося швидко виправити вразливість у додатку, модифікована версія може не бути вчасно розгорнута для користування. За допомогою WAF можна додати вразливість у білий список, тим самим виправивши її до виходу наступних програмних оновлень.

WAF є особливо важливим в захисті веб-додатків, які в свою чергу складаються з кількох компонентів, які не можуть бути швидко змінені в разі виявлення вразливості.

Існують й інші переваги WAF, зумовлені використанням його в центральній ролі. Процес визначення місця виникнення помилки значно спрощується, якщо WAF підтримує центральний збір повідомлень про помилки, на відміну від генерування індивідуальних помилок в кожному додатку. Після цього помилки можуть бути центрально оцінені в WAF.

Цей принцип стосується всіх аспектів моніторингу та звітності. В ролі центрального сервісу WAF здатен створювати завдання, які будуть виконані однаково для всіх додатків. Хорошим прикладом цього є безпечне керування сеансами для всіх додатків, що базуються на зберіганні файлів кукі.

Багато WAF також забезпечують активні механізми безпеки, такі як шифрування URL-адреси або примусове використання сайту, щоб мінімізувати область для атак з використанням якомога менших зусиль. Крім того, використання WAF підвищує стійкість до зовнішніх атак.

WAF пропонують й інші додаткові переваги в залежності від типу імплементації. Апаратне рішення, встановлене перед веб-сервером, часто може закінчувати SSl підключення, а також іноді має здатність балансувати навантаження на сервер.

У таблиці нижче наведено можливі заходи безпеки для типових загроз, вразливостей та атак, та оцінка, наскільки добре WAF може захистити додаток. В стовпці WAF наступні символи означають:

- «+» - дуже добре забезпечується WAF
- «-» - не забезпечується (або забезпечується в малій мірі) WAF
- «!» - залежить від WAF/додатку/вимог
- «=» - може частково забезпечуватись WAF

Таблиця 1.1

Можливості WAF для захисту веб-додатків.

Проблема	WAF	Протидія
Захист файлів кукі	+	Файли кукі можуть бути підписані
	+	Файли кукі можуть бути зашифровані
	!	Файли кукі можуть бути повністю приховані або переміщені
	!	Файли кукі можуть бути прив’язані до IP адреси користувача

Продовження таблиці 1.1

Витік інформації	+	Фільтр маскування, вихідні сторінки можуть бути очищені (повідомлення про помилки, коментарі, небажана інформація)
Контроль сесії	+	Шифрування URL-адреси / використання токенів
Час очікування сесії	!	Можна вказати час очікування для активних і неактивних сесій, якщо WAF може контролювати сесії. Якщо сеансами керує програма, WAF може виявити це та зупинити сесію
Запис сесії	=	Можна запобігти, якщо WAF контролює сесію
Перехоплення сесії	-	Важко запобігти, хоча WAF може сповістити про загрозу в разі порушень
Завантаження файлів	+	Перевірка на наявність вірусів
Підробка параметрів	+	Підробка параметрів може бути усунена шляхом шифрування URL-адрес та параметрів
Примусовий перегляд	+ +	Може бути усунена шляхом шифрування URL-адрес Контроль використання сайту
Обхід перевірки посилання	+ +	Може бути усунена шляхом шифрування URL-адрес Контроль використання сайту
Логування	+	Вся або вибрана інформація може логуватися
Підвищення привілеїв	-	Підвищення привілеїв не може бути перевіреним, або може бути перевіреним в малій мірі
Логічний рівень	-	Логіка додатку, що виходить за межі перевірки URL-адрес чи полів заповнення не може бути перевіреною WAF
Антиавтоматизація	=	Автоматизовані атаки можуть бути частково помічені та заблоковані
DoS програми	= =	Транзакції, IP-адреси та/або користувачі можуть бути заблоковані. Підключення та/або сесії можуть бути закінчені
SSL	+ +	WAF може примушувати SSL з попередньо встановленою силою шифрування. Можливе SSL з'єднання від WAF до додатку
Перевірка інформації	+ + !	Можливе тестування в широкому спектрі параметрів. Правила можуть генеруватися автоматично. Є залежність від додатку, деяких полів або попередньо встановлених параметрів в URL-посиланні, хоча WAF може автоматично їх розпізнавати.
Переповнення буферу	+	Те саме, що для перевірки інформації

Продовження таблиці 1.1

Атака на основі формату рядку	=	Можна виявити за допомогою перевірки даних, якщо відповідні символи чи рядки фільтруються (важко реалізувати на практиці, оскільки необхідно точне знання про додаток). Для більшості скритих полів це може бути реалізовано без знання про додаток
Міжсайтовий скриптінг	=	Використовуючи перевірку інформації, лише деякі види цих атак можуть бути виявлені та зупинені, деякі можуть бути частково зупинені за певних умов, а деякі не можуть бути виявлені взагалі
Міжсайтове стеження	+	Заборона HTTP методу для, наприклад, GET або POST
WebDAV	+	Можливе обмеження прав лише до читання для WebDAV методів
Ін`єкції в код	+	Те саме, що для перевірки інформації
Ін`єкції в командний рядок	+	Те саме, що для перевірки інформації
SQL ін`єкції	+	Те саме, що для перевірки інформації
LDAP ін`єкції	+	Те саме, що для перевірки інформації
XML/Xpath ін`єкції	+	Те саме, що для перевірки інформації
Вчасне виправлення помилок	+	Завдяки перевірці інформації, WAF здатен захистити від щойно виявлених вразливостей та/або атак
Поділ HTTP відповідей	!	Може бути виявлено лише при використанні перевірки інформації в URL-посиланні та/або параметрах, якщо %0d%0a фільтрується
Підробка HTTP запитів	+	Запобігається за умови перевірки кожного запиту на відповідність стандартам

2. МЕТОДИ ВПРОВАДЖЕННЯ ІНСТРУМЕНТІВ ЗАБЕЗПЕЧЕННЯ БЕЗПЕКИ ТА ДОСТУПНОСТІ ВЕБ ДОДАТКІВ

2.1 Методи встановлення та налаштування програмного забезпечення F5 BIG-IP WAF

Одним з можливих варіантів розгортання F5 BIG-IP LTM є розгортання на віртуальній машині. Перед встановленням та налаштуванням слід переконатися, що віртуальна машина задовольняє необхідні технічні умови для стабільної роботи системи.

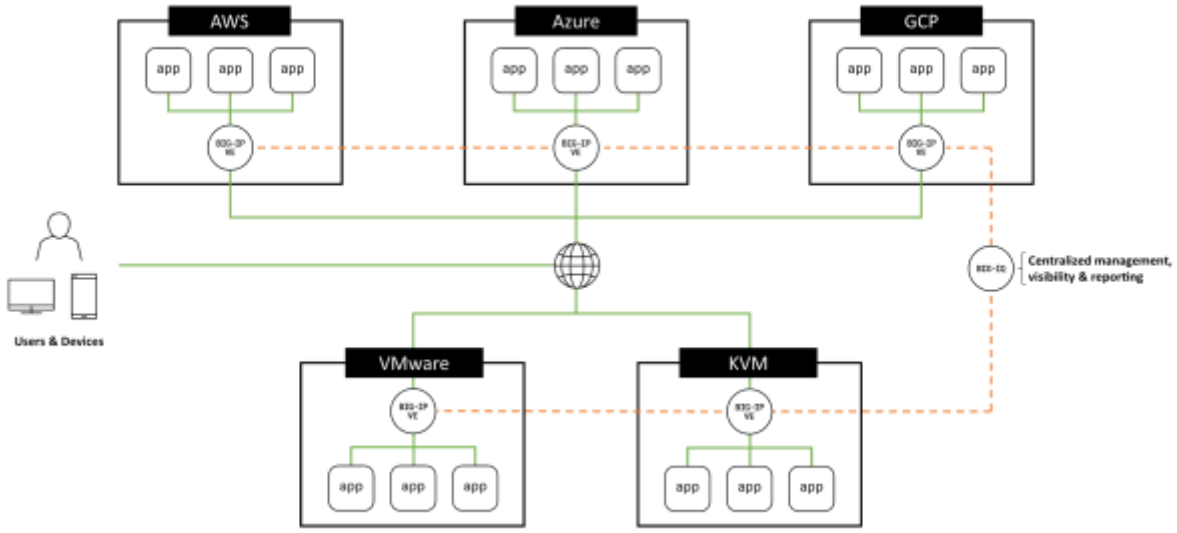


Рис 2.1.

Центральний процесор машини, на якій буде розгортатися віртуальна машина має задовольняти наступні вимоги:

- Процесор має працювати на 64-бітній архітектурі.
- Процесор має підтримувати технологію віртуалізації.
- Процесор має підтримувати роботу віртуального процесора в режимі один до одного, багато до одного.
- Якщо процесор підтримує стандарт AES-NI, процес шифрування SSL проходитиме швидше.

Від кількості оперативної пам`яті на машині, на якій розгортатиметься віртуальна машина, залежитиме кількість працюючих ядер.

Таблиця 2.1.

Залежність кількості ядер від кількості оперативної пам`яті

Кількість ядер	Необхідна кількість оперативної пам`яті
1	2 гб
2	4 гб
4	8 гб
8	16 гб

Для встановлення додаткових модулів, необхідна додаткова оперативна пам`ять.

Таблиця 2.2.

Необхідна оперативна пам`ять для додаткових модулів

Необхідна оперативна пам`ять	Кількість підтримуваних модулів	Деталі
4 гб або менше	Максимум два модулі	AAM може бути забезпеченим тільки автономно
4-8 гб	Максимум три модулі	BIG-IP DNS не враховується в ліміті модулів. Винятком є використання Application Acceleration Manager (AAM).
8 гб	Максимум три модулі	BIG-IP DNS не враховується в ліміті модулів.
12 гб або більше	Всі модулі	

В залежності від модулів, які будуть встановлені, визначається необхідна кількість пам`яті для зберігання.

Таблиця 2.3.

Необхідна кількість пам`яті для зберігання

Необхідна пам`ять	Встановлені модулі	Деталі

8 гб	Тільки Local Traffic Manager (LLT) модуль. Без місця для оновлень	Для оновлення LLT модуля або встановлення інших модулів необхідно розширити об`єм пам`яті.
38 гб	Тільки Local Traffic Manager (LLT) модуль. З місцем для встановлення оновлень	Для встановлення додаткових модулів необхідно збільшити об`єм пам`яті. Можна окремо встановити інший LLT модуль.
127 гб	Всі модулі та місце для встановлення оновлень	AAM модуль вимагає 20 гб виділеної пам`яті.

Віртуальна машина повинна мати мінімум три віртуальні порти. Більше портів необхідно, якщо сконфігурована опція з можливістю більшої доступності.

Для створення інструменту BIG-IP на віртуальній машині необхідно скачати образ та встановити його. Важливим є не змінювати конфігурацію віртуального середовища на меншу, ніж необхідні, описані далі.

Для встановлення BIG-IP слід виконати наступні кроки:

1. Скачати архів з необхідними файлами з сайту виробника.
2. Розпакувати архів.
3. В менеджері віртуальних середовищ створити нову віртуальну машину.
4. В полі ім`я ввести ім`я для нової віртуальної машини та натиснути далі.
5. В полі оперативна пам`ять ввести 4096 та натиснути далі. Для підвищення якості роботи можна виділити до 8192.
6. Для налаштування підключення вибрати керування та натиснути далі.
7. Співставити вихідну мережу HA з назвою мережі з високою доступністю, наявною в розпорядженні.
8. Вибрати опцію використовувати існуючий віртуальний накопичувач, знайти місце знаходження файлу встановлення BIG-IP, вибрати його та натиснути далі.

9. На заключній сторінці перевірити налаштування та натиснути кнопку закінчити. Тепер віртуальна машина з BIG-IP з`явиться в списку віртуальних машин.

10. В списку доступних віртуальних машин вибрати щойно додану машину та натиснути кнопку налаштування.

11. Зі списку апаратного забезпечення вибрати процесор та змінити кількість виділених логічних процесорів на 2, та змінити відсоток резерву для віртуальної машини до 100.

12. Натиснути додати апаратне забезпечення, вибрати мережевий адаптер та натиснути додати. Повторити цю дію для трьох адаптерів (або чотирьох для більш доступної конфігурації).

13. В полі керування натиснути вимкнути операційну систему. Це зроблено для того, щоб після запуску машина працювала з заданими налаштуваннями.

14. Натиснути кнопку ок та закрити вікно налаштувань.

Для створення та налаштування F5 WAF слід виконати наступні кроки:

1. Зайти на консоль F5. На домашній сторінці натиснути Захист веб-додатків та API.

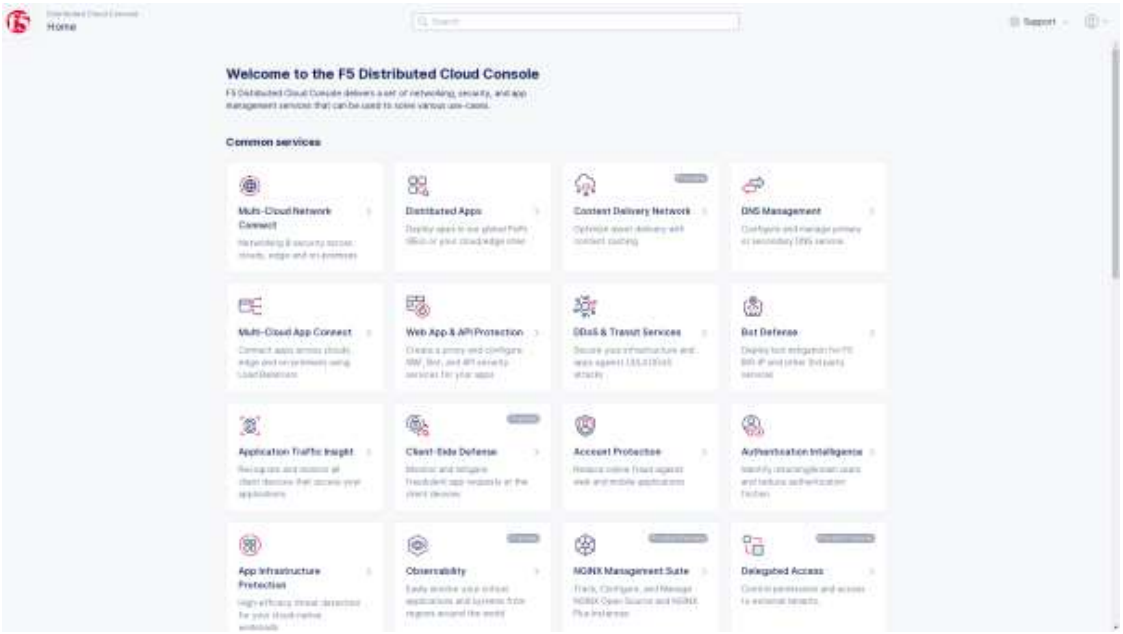


Рис 2.2.

2. В викидному меню вибрати бажаний простір імен. Також можна створити власний простір імен, для якого буде створено фаєрвол. Для цього на домашній сторінці консолі треба вибрати сервіс Адміністрування, потім вибрати Персональний менеджмент, потім Власні простори імен. Натиснути додати простір імен, ввести ім`я та натиснути зберегти.

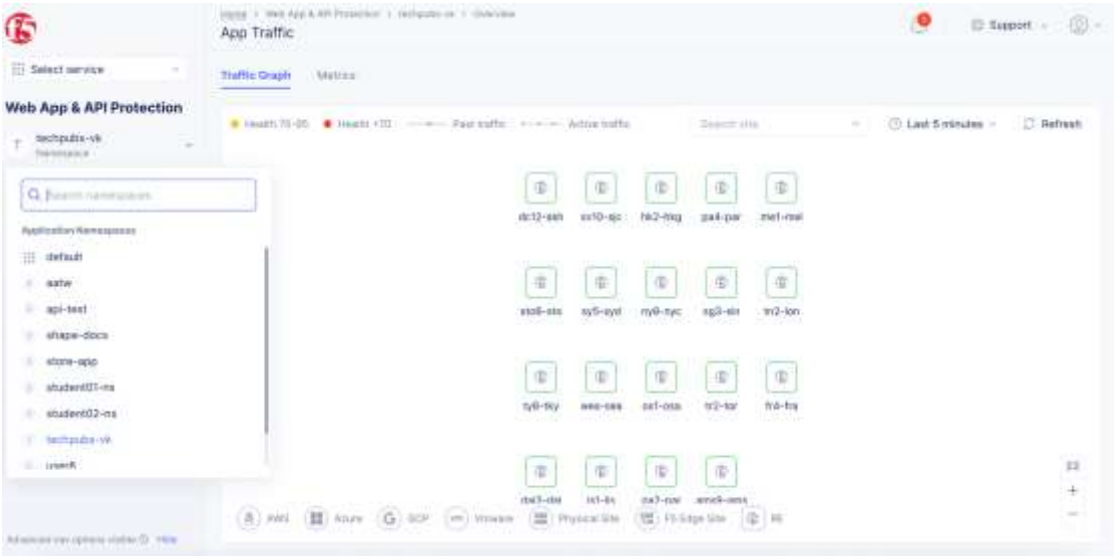


Рис 2.3.

Після вибору простору імен натиснути Керувати та Фаєрвол. Далі натиснути Додати фаєрвол для завантаження форми створення WAF.

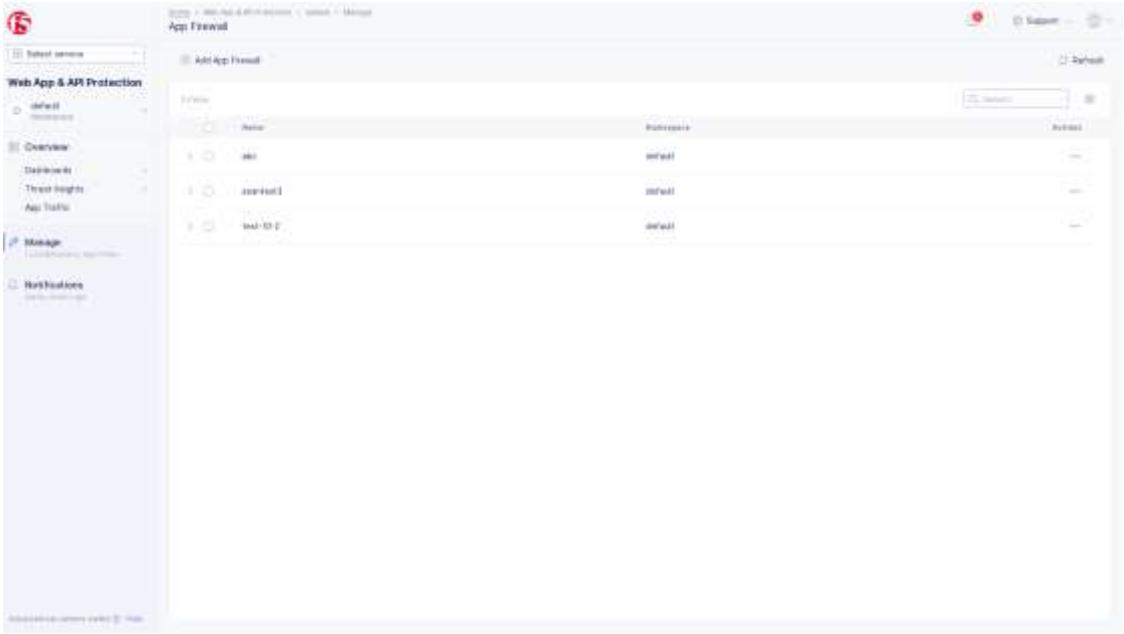


Рис 2.4.

3. В секції Мета дата слід заповнити всі необхідні поля. В полі Ім`я слід ввести назву WAF. В полі Режим роботи в викидному меню слід вибрати в якому режимі буде працювати WAF – в режимі моніторингу чи блокування.

В режимі блокування шкідливий трафік записується в лог-файли та блокується.

В режимі моніторингу трафік не блокується, але будь-який шкідливий чи підозрілий трафік генерує записи безпеки в журналі подій.

Рис 2.5.

4. В секції Параметри виявлення необхідно виконати необхідні налаштування. В викидному меню Політики безпеки треба обрати стандартні налаштування або персональні налаштування.

Стандартні налаштування являють собою широкий набір сигнатур високої та середньої точності, загроз та порушень.

Персональні налаштування дозволяють створити спеціальну конфігурацію для типів атак, вибірки сигнатур, сигнатур автоматичних атак, загроз та порушень. Можна змінювати одну або всі параметри конфігурації.

В разі вибору персональних налаштувань слід налаштувати наступні параметри:

- Типи атак. Стандартні налаштування виявляють всі типи атак. Для налаштування цього параметру та відключення виявлення конкретних типів атак треба вибрати персональні налаштування. В них можна обрати бажані типи атак. Можна обрати один або декілька типів атак.

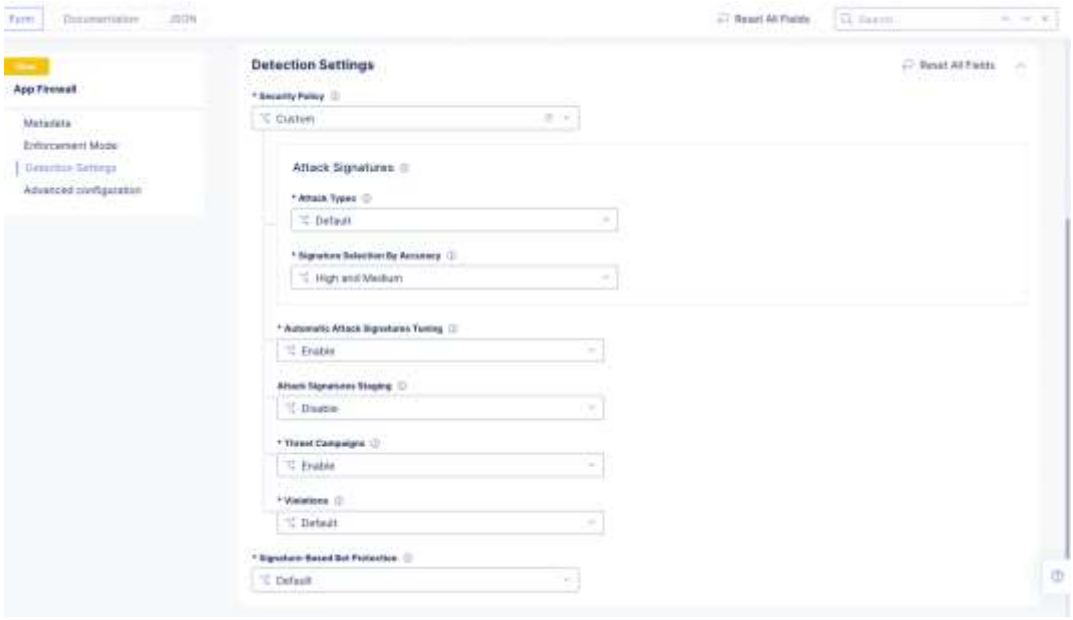


Рис 2.6.

- В меню Вибір сигнатур за точністю слід вибрати одну з запропонованих опцій точності. Висока та середня точність доступні за замовчуванням.
- В меню Автоматичне налаштування сигнатур атак можна ввімкнути або вимкнути цю опцію. Вона ввімкнена за замовчуванням.
- В меню Обробка сигнатур атак можна вибрати, чи мають нові та оновлені сигнатури оброблятися інакше від режиму роботи фаєрволу. Коли сигнатури обробляються, вони будуть оброблені в режимі моніторингу, навіть якщо фаєрвол працює в режимі блокування.

Якщо вимкнути цей параметр всі сигнатури будуть блокуватися.

Якщо ввімкнути обробку нових сигнатур атак, вони будуть оброблятися в режимі моніторингу, тобто не будуть блокуватися а лише створювати запис в журналі подій. Існуючі сигнатури, які були оновлені будуть блокуватися.

Якщо ввімкнути обробку нових та оновлених сигнатур атак, вони будуть оброблятись в режимі моніторингу, тобто не будуть блокуватися а лише створювати запис в журналі подій.

- В меню Загрози можна ввімкнути або вимкнути виявлення загроз. Якщо ввімкнути цей параметр, WAF виявлятиме специфічні загрози та виконуватиме дії залежно від режиму роботи фаєрволу.
- В меню Порушення можна вибрати опцію персональні налаштування, щоб вимкнути одну або більше порушень. В меню вимкнені порушення треба вибрати порушення, які треба вимкнути.

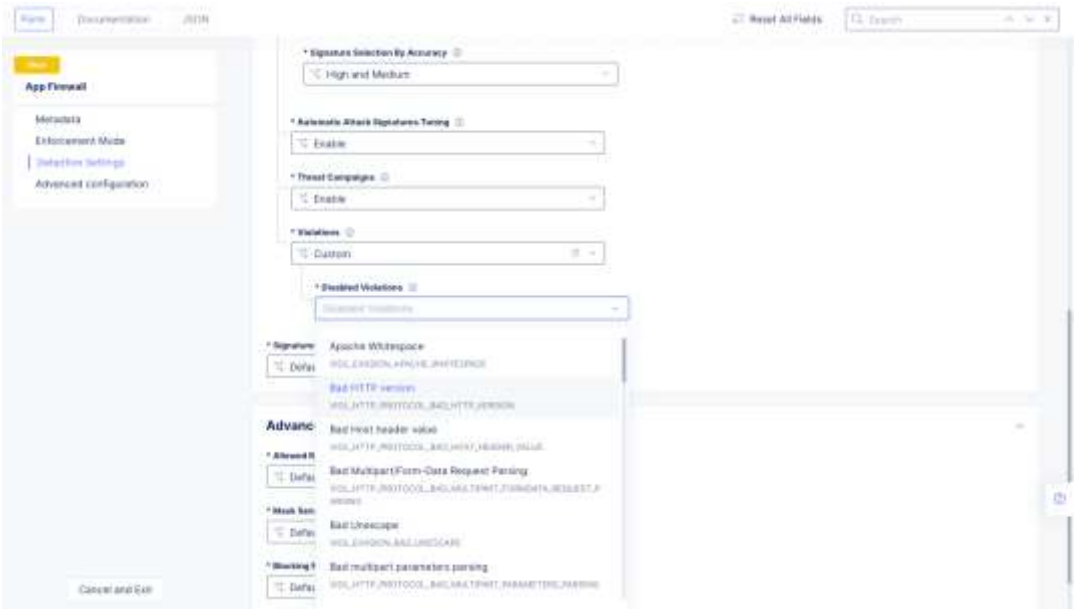


Рис 2.7.

5. В викидному меню Захист від ботів на основі сигнатур треба вибрати опцію захисту від ботів.

За замовчуванням шкідливі боти блокуються та генерують подію в журналі безпеки. Хороші та підозрілі боти тільки генерують подію в журналі безпеки, а WAF не блокує їх активність.

Власні налаштування дозволяють встановити яку дію (блокування, попередження, або ігнорування) слід виконувати WAF, коли він виявляє шкідливого бота, підозрілого бота, або хорошого бота. Слід встановити необхідні дії для кожного типу бота в відповідному полі.

Рис 2.8.

6. В секції Розширені налаштування натиснути Показати поля розширених налаштувань.

В меню Дозволені відповіді коду статусу вибрати персональні налаштування та вказати список HTTP відповідей коду статусу, які дозволено бачити клієнту. Всі інші HTTP відповіді будуть заборонені.

В меню Маскування чутливих параметрів в журналі логування за замовчуванням налаштовано, що чутливі параметри (наприклад номери кредитних карток) будуть маскуватися в журналі логування. Можна вимкнути цю функцію або налаштувати, які саме параметри будуть маскуватися. Для цього в персональних налаштуваннях треба натиснути додати пункт та вибрати HTTP заголовок, параметр запиту або файл кукі та ввести відповідне ім`я. Можна додати більше ніж один пункт.

В меню Блокування сторінки відповіді за замовчуванням повертає користувачу HTML відповідь з системними налаштуваннями. Для налаштування спеціальних сторінок відповідей треба вибрати персональні налаштування та зробити необхідні налаштування. В меню Код відповіді треба обрати відповідь, яка буде надсилатися для заблокованого запиту. В полі Тіло сторінки заблокованих відповідей написати відповідь, яку буде бачити клієнт.

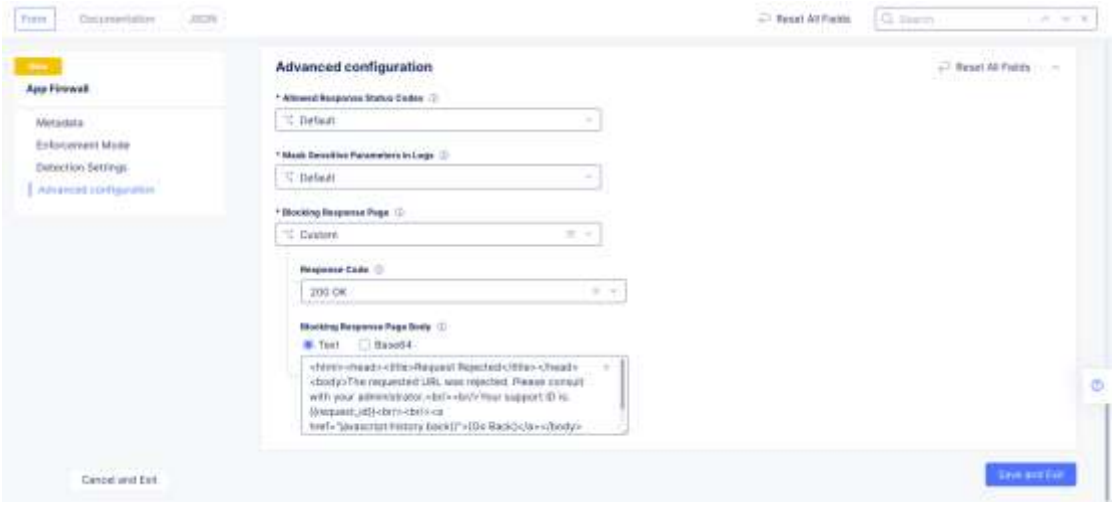


Рис 2.9.

7. Після виконаних налаштувань натиснути кнопку Зберегти та вийти.

Після створення WAF, його треба під`єднати до мережевого балансера. Для цього треба виконати наступні кроки:

1. На домашній сторінці консолі керування натиснути З`єднання багатьох хмарних додатків. Після цього обрати необхідний простір імен в викидному меню. Далі натиснути Керування, Мережевий балансер, HTTP мережевий балансер. Вибрати потрібний мережевий балансер та натиснути Керування параметрами. Потім натиснути Змінити Налаштування, щоб відкрити поля зміни.

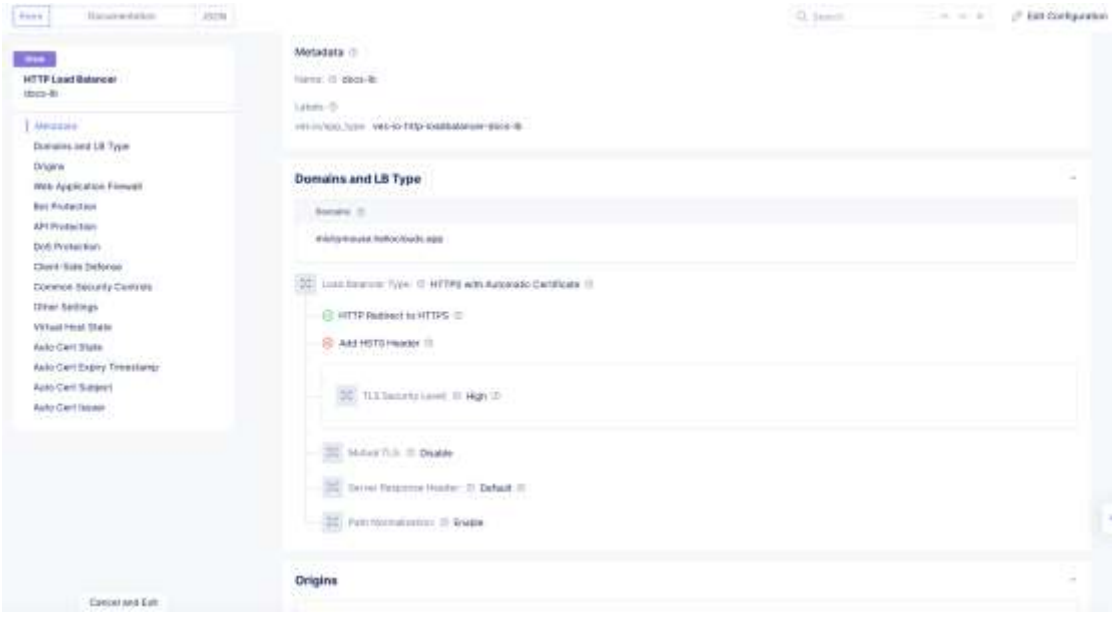


Рис 2.10.

2. В секції WAF вибрати значення Ввімкнути. В викидному меню вибрати WAF, створений до цього.

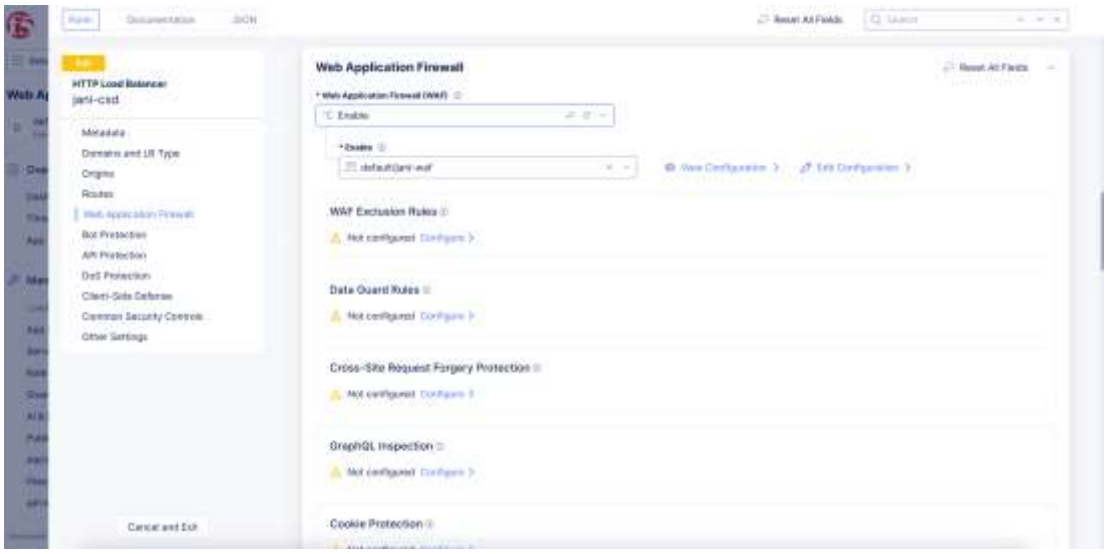


Рис. 2.11.

2.2 Методи налаштування мережевого балансувальника навантажень від F5 BIG-IP

Для створення та налаштування мережевого балансера треба зайти в консоль та натиснути Multi-Cloud App Connect.

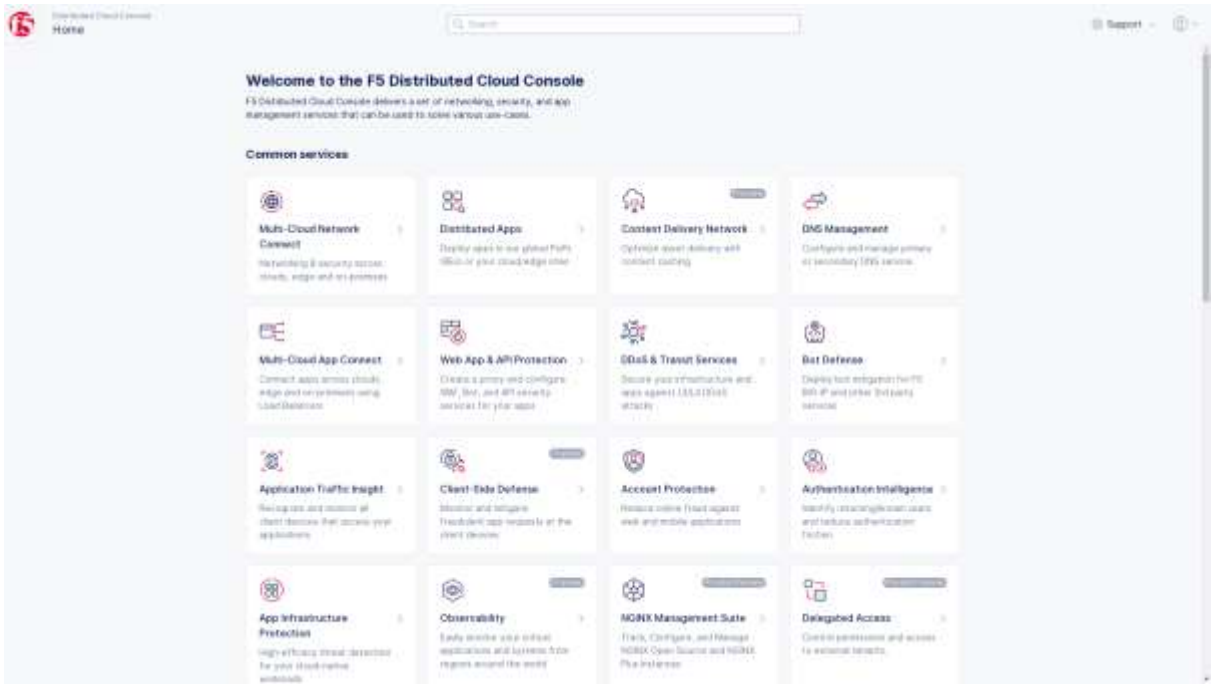


Рис 2.12.

Після цього вибрати Керування, Мережеві балансери, HTTP мережеві балансери.

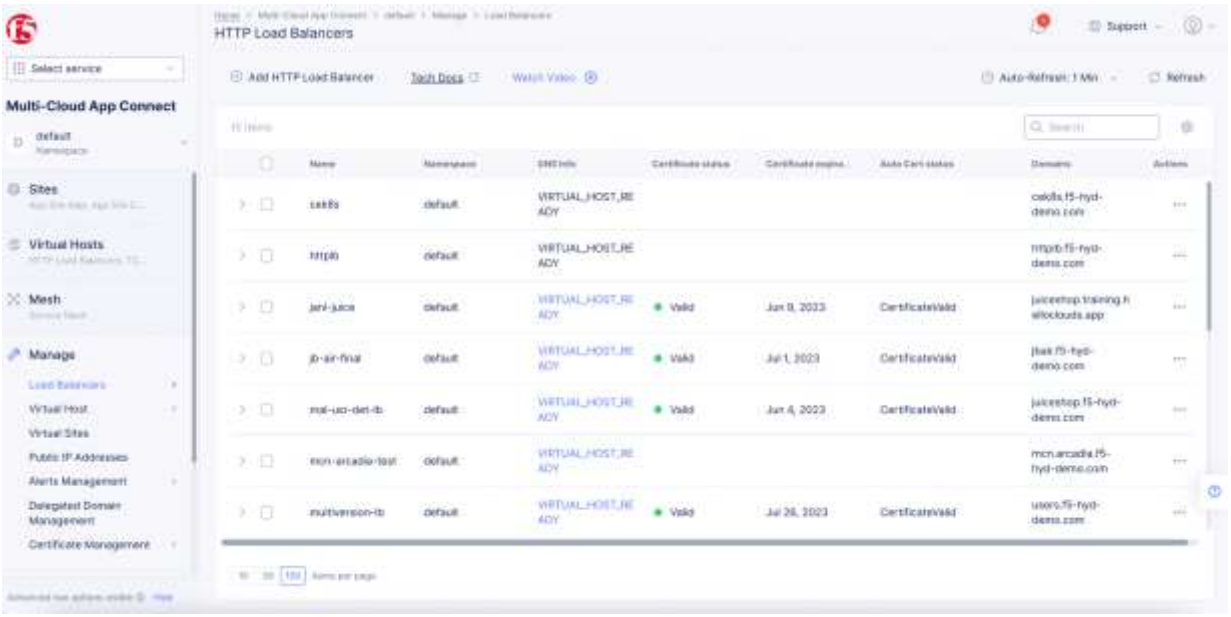


Рис 2.13.

Далі треба підтвердити вибраний простір імен та натиснути Додати HTTP мережевий балансер.

Далі необхідно виконати налаштування створеного мережевого балансера. Для цього слід виконати наступні кроки:

1. В полі Ім`я ввести назву для мережевого балансера.
2. За потреби можна ввести помітку та опис.

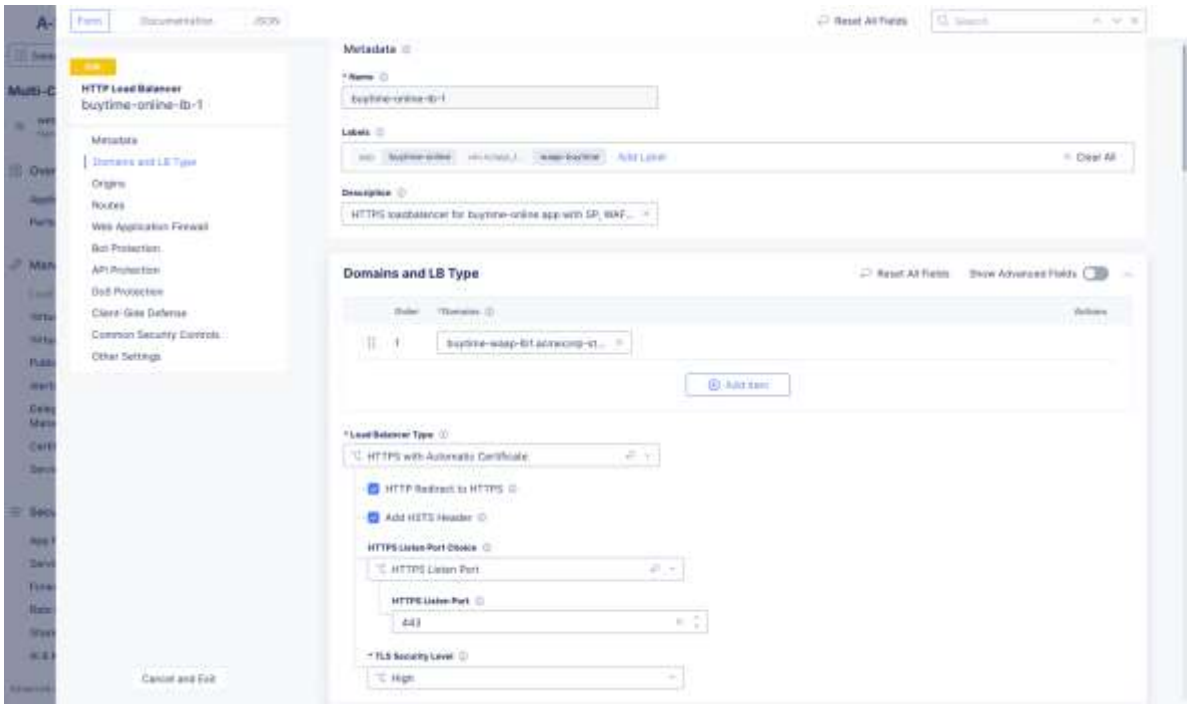


Рис 2.14.

3. В полі Домени треба ввести ім`я домену. За необхідності можна додати більше доменів.

4. В викидному меню Тип мережевого балансера треба вибрати одну з запропонованих опцій.

HTTP – для створення HTTP мережевого балансера.

HTTP з автоматичними сертифікатами – для створення HTTPS мережевого балансера з автоматичними TLS сертифікатами.

HTTP з налаштовуваними сертифікатами – для створення HTTPS мережевого балансера з власними налаштовуваними TLS сертифікатами.

Якщо вибрано опцію HTTP, треба обрати чи повинні хмарні сервіси обробляти ваші DNS записи за допомогою Автоматичної обробки DNS записів.

В іншому випадку опціонально треба обрати Перенаправлення HTTP в HTTPS та Додати HSTS заголовки.

Для всіх типів мережевого балансеру можна використовувати Спосіб обробки HTTP портів для вибору між обробкою одного порту або списку портів, та ввести ці порти.

Якщо використовується HTTP з автоматичними сертифікатами, в викидному меню TLS рівень безпеки треба обрати бажаний рівень безпеки TLS.

Якщо використовується HTTP з налаштовуваними сертифікатами, треба виконати наступні налаштування:

- В викидному меню Налаштування TLS треба обрати між використанням одного або багатьох сертифікатів та натиснути Налаштувати.
- В викидному меню Рівень безпеки TLS треба обрати бажаний рівень.
- В секції Налаштування TLS натиснути Додати предмет.
- Для шифрування URL-посилання на сертифікат треба обрати тип шифрування PEM або base64, та ввести URL-посилання на сертифікат.
- Для налаштування приватного ключа треба натиснути Налаштувати.
- В секції Secret налаштувати параметри для приватного ключа та натиснути застосувати.
- В викидному меню OSCP печатки треба обрати необхідну OSCP печатку.
- Натиснути Застосувати.
- На сторінці Параметри TLS натиснути застосувати.

5. Для налаштування пулу джерел в секції Джерела натиснути Додати джерело для створення нового пулу.

В меню Вибір методу пулу джерел треба обрати тип пулу – звичайний чи налаштовуваний.

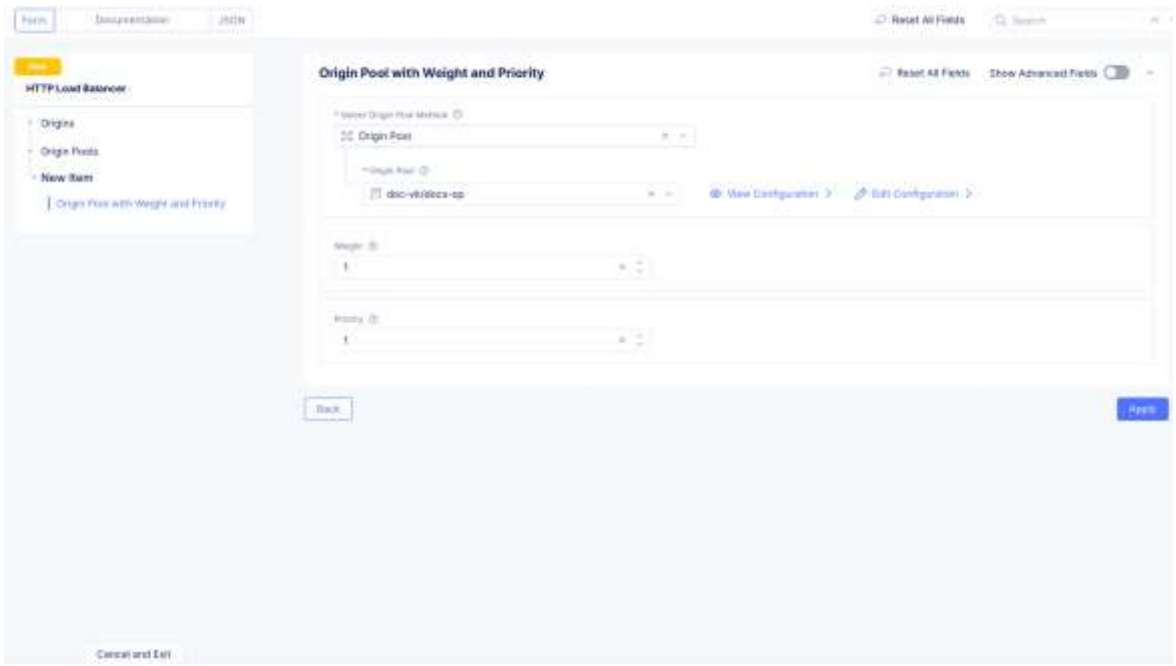
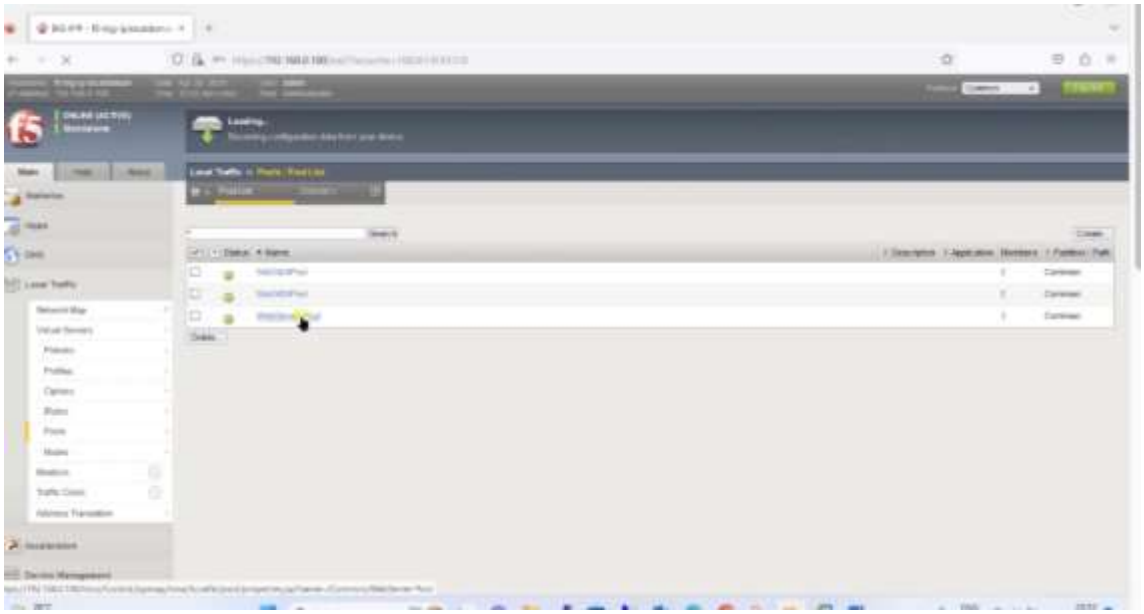


Рис 2.15.

2.3 Налаштування методів балансування навантаження та порівняння результатів балансування різних типів

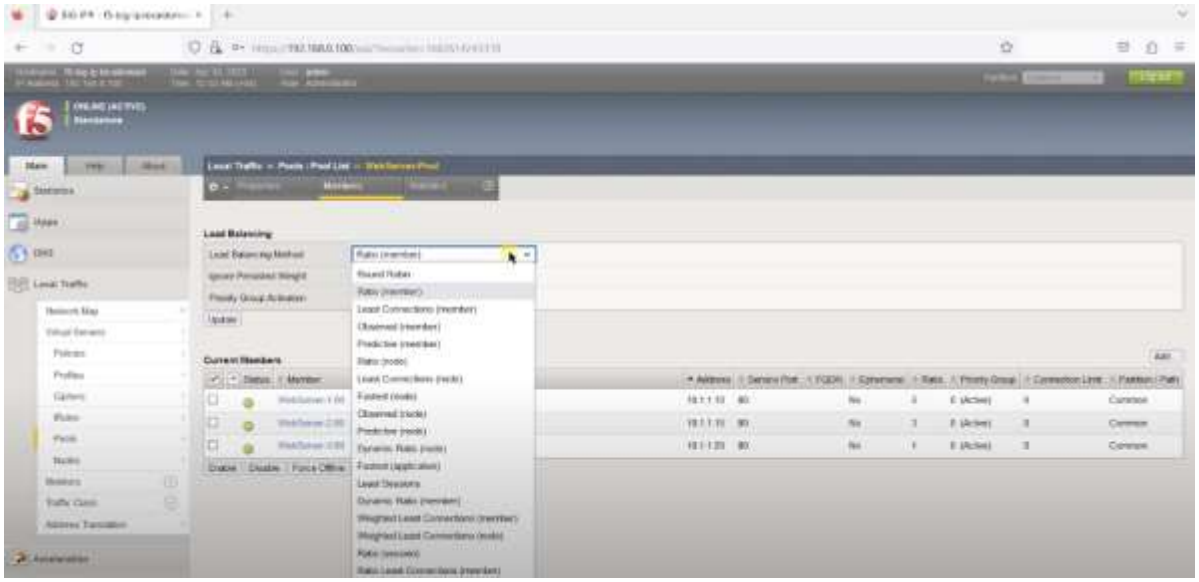
Для налаштування методу балансування навантаження в F5 Big-ip необхідно зайти в панель керування балансувальника навантаження та виконати наступні дії:

1. В головному меню вибрати Local traffic, Pools, Pool list.
2. Зі списку пулів необхідно обрати той, для якого буде проводитись налаштування.



Рис

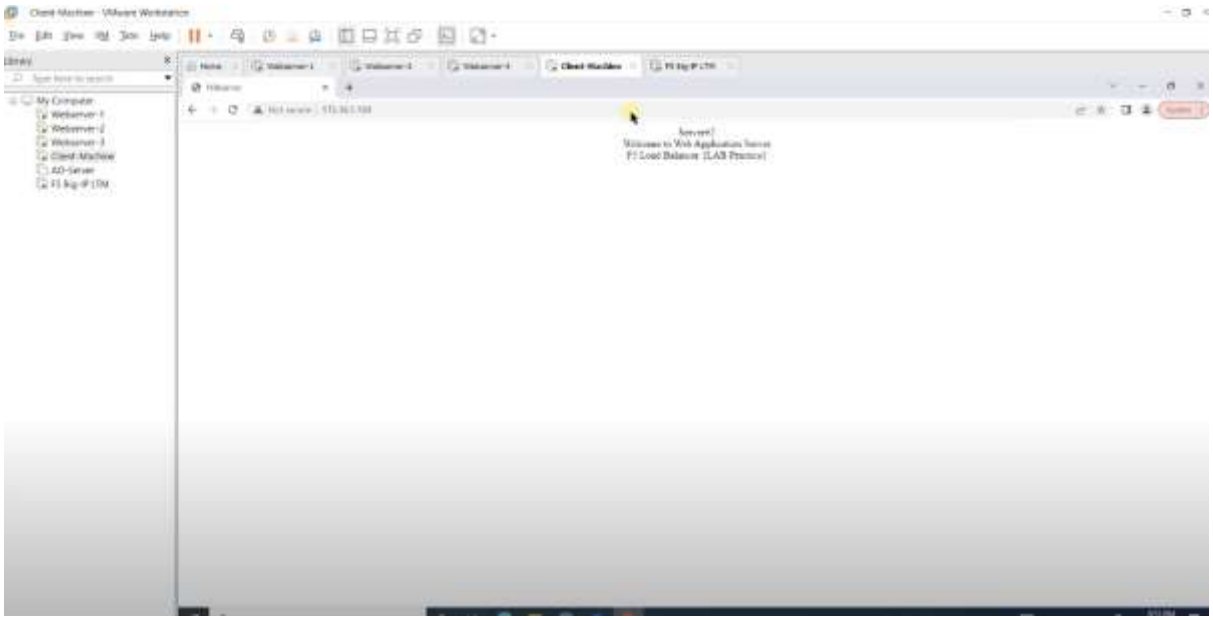
3. В полі Load Balancing Method треба вибрати метод балансування навантаження. Для прикладу обираємо Dynamic Ratio.



Рис

4. Після цього треба натиснути кнопку Update.

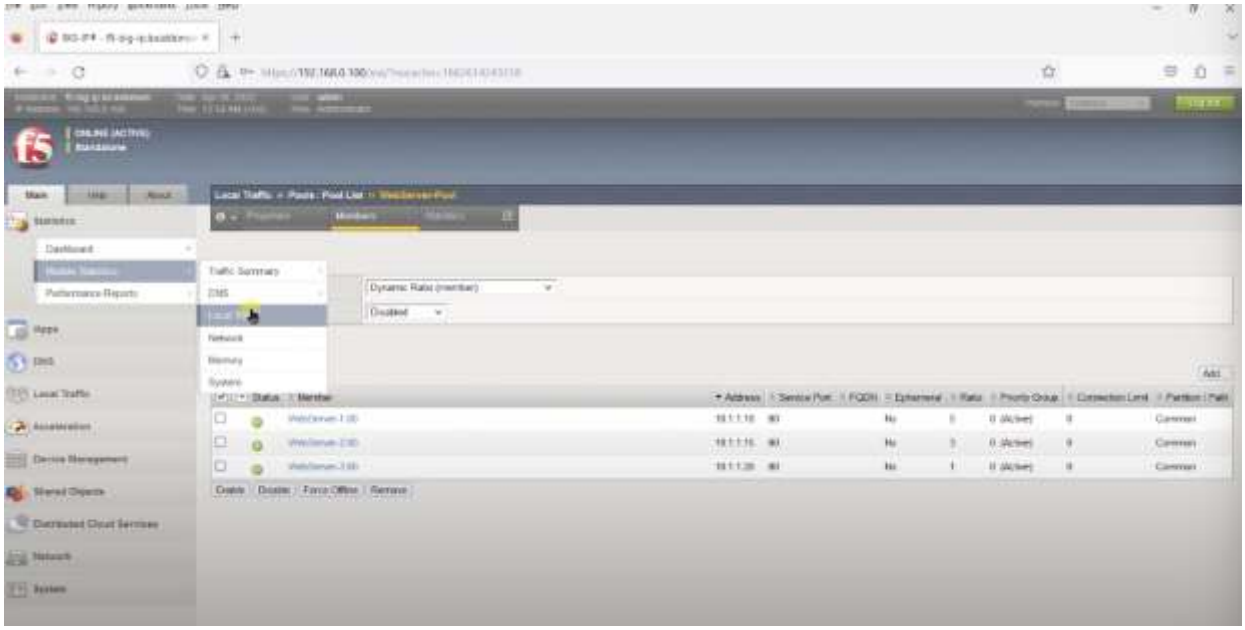
Для перевірки роботи балансувальника навантаження виконаємо декілька запитів на веб-сервер.



Рис

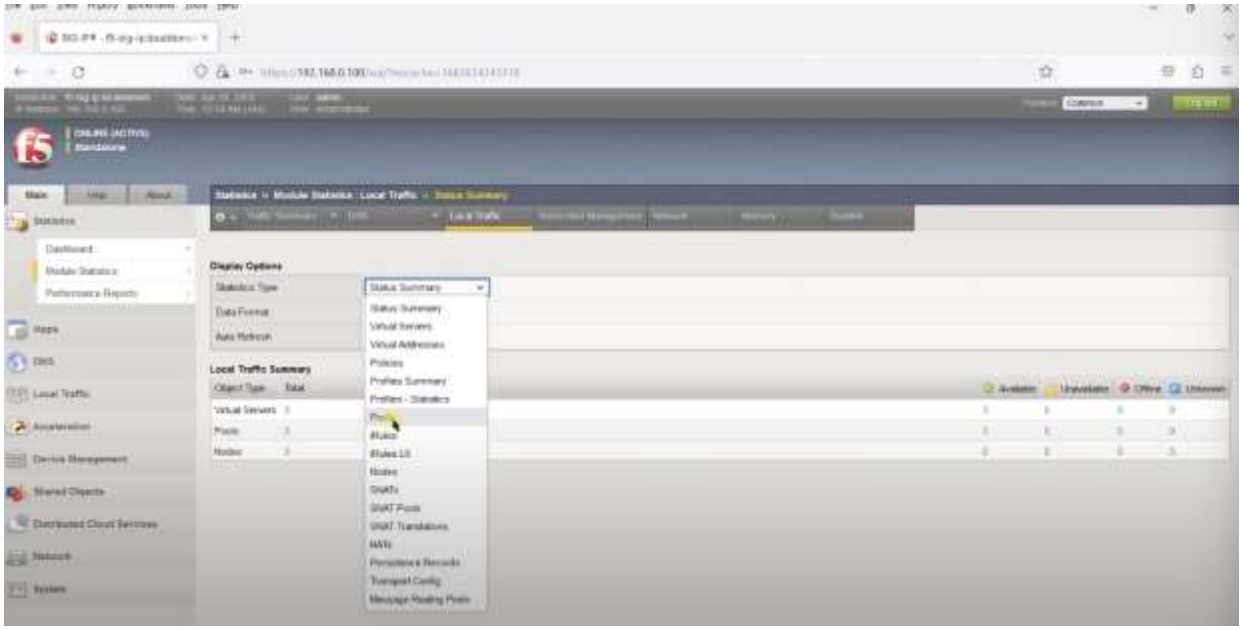
Після цього треба знову зайти в панель керування балансувальника навантаження.

В головному меню треба вибрати Statistics, Module Statistics, Local Traffic.



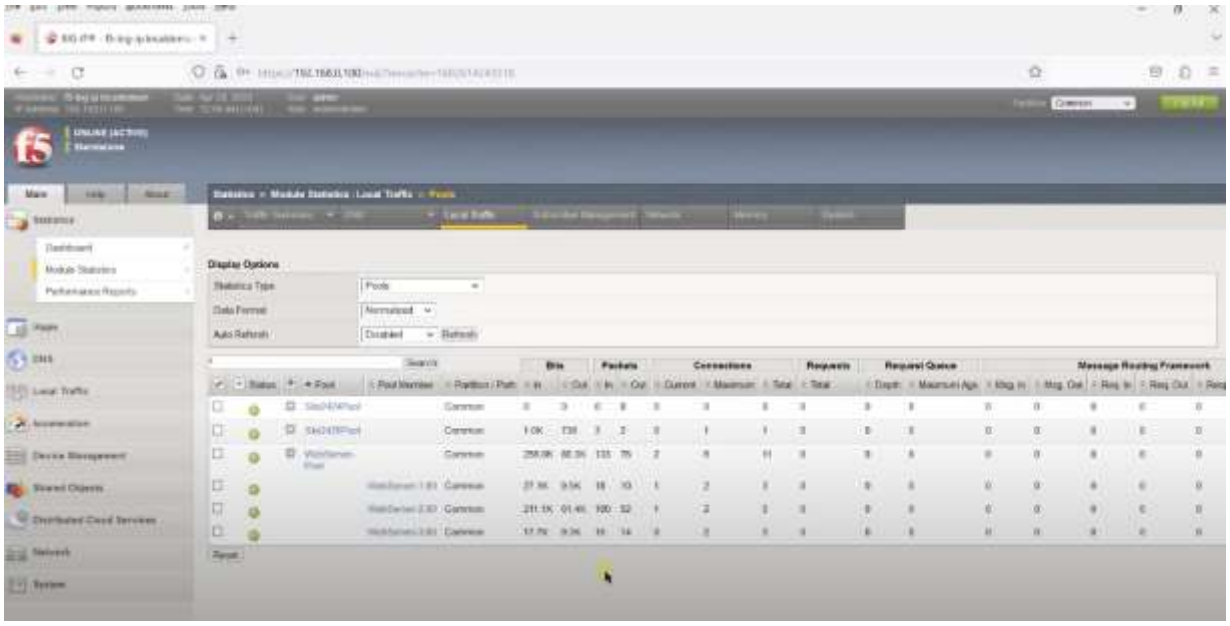
Рис

В полі Statistics Type треба обрати опцію Pools.



Рис

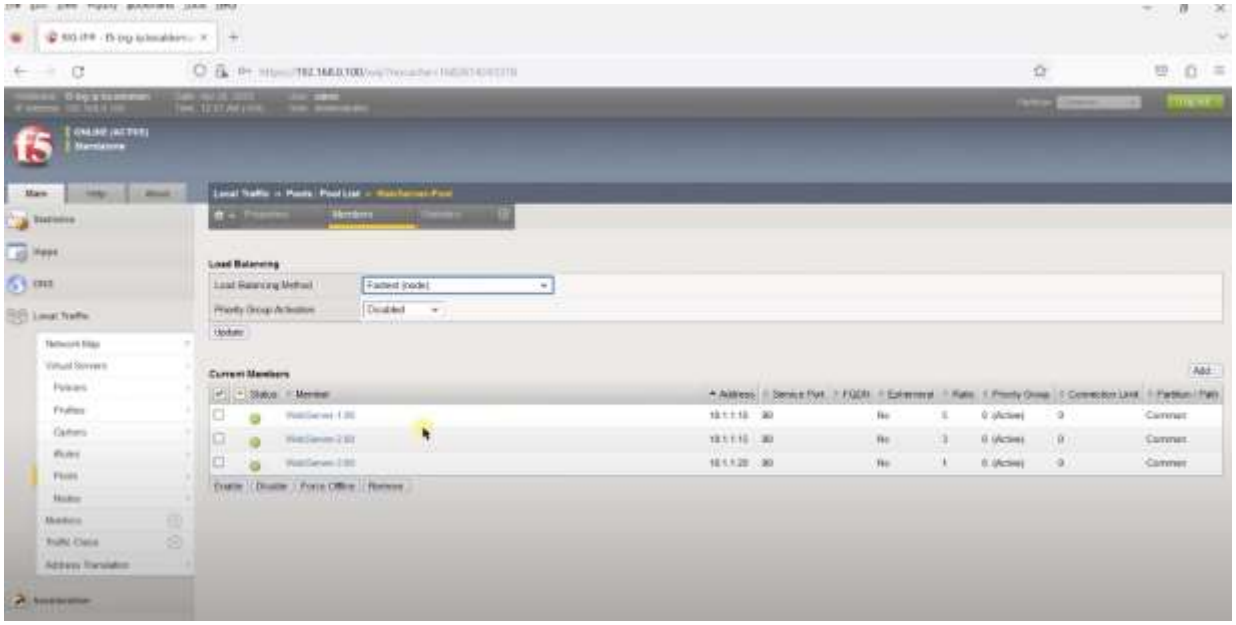
Серед запропонованих пулів треба обрати необхідний та натиснути кнопку Refresh.



Рис

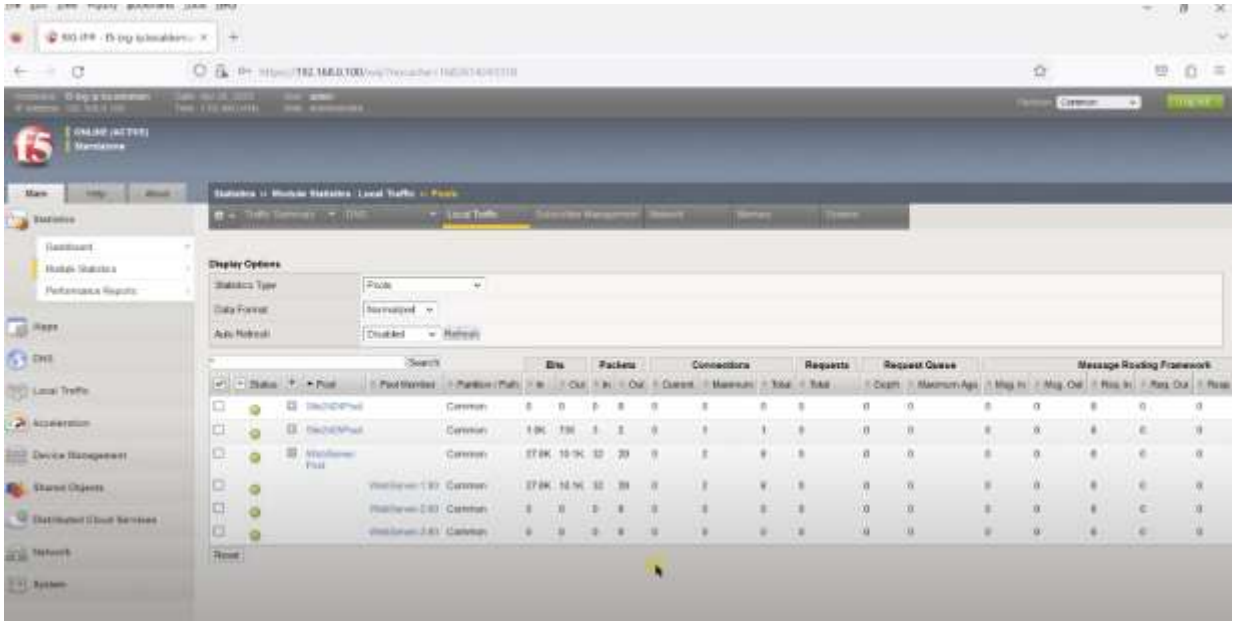
На рисунку () видно що загалом було зроблено 11 запитів до веб-сервера. З них 3 було направлено на перший сервер, 5 запитів на другий сервер та 3 запити на третій сервер.

Для зображення різниці між різними методами балансування навантаження було проведено такі самі налаштування для інших методів балансування навантаження.



Рис

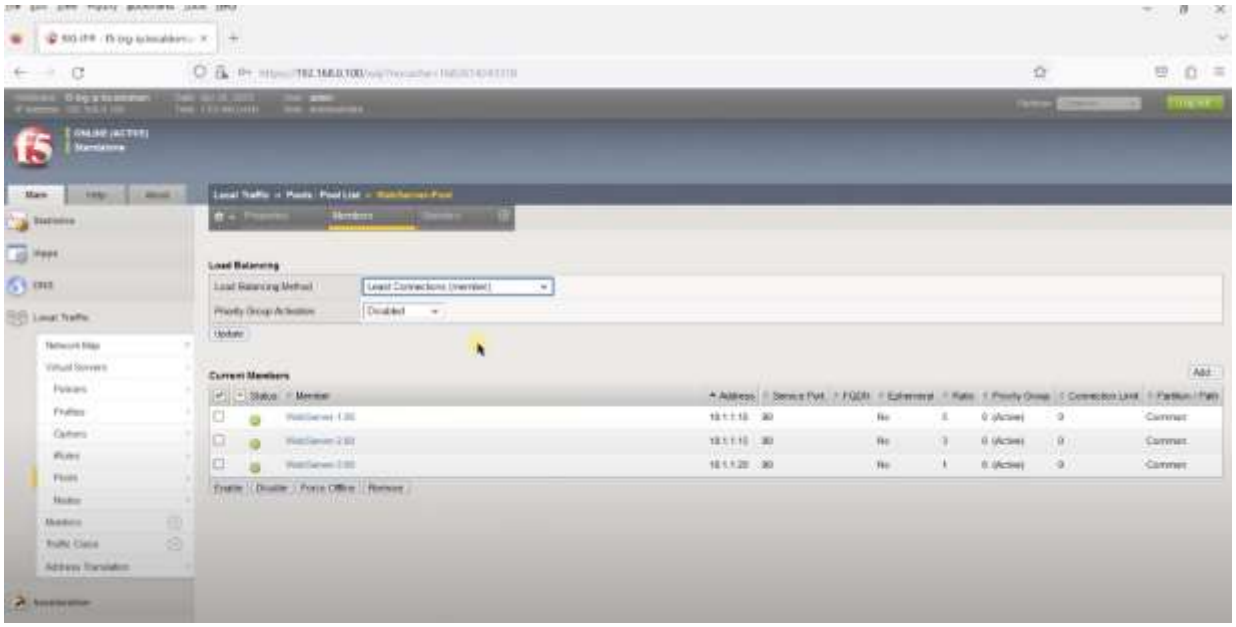
Для методу балансування навантаження Найшвидше з’єднання було отримано наступний результат:



Рис

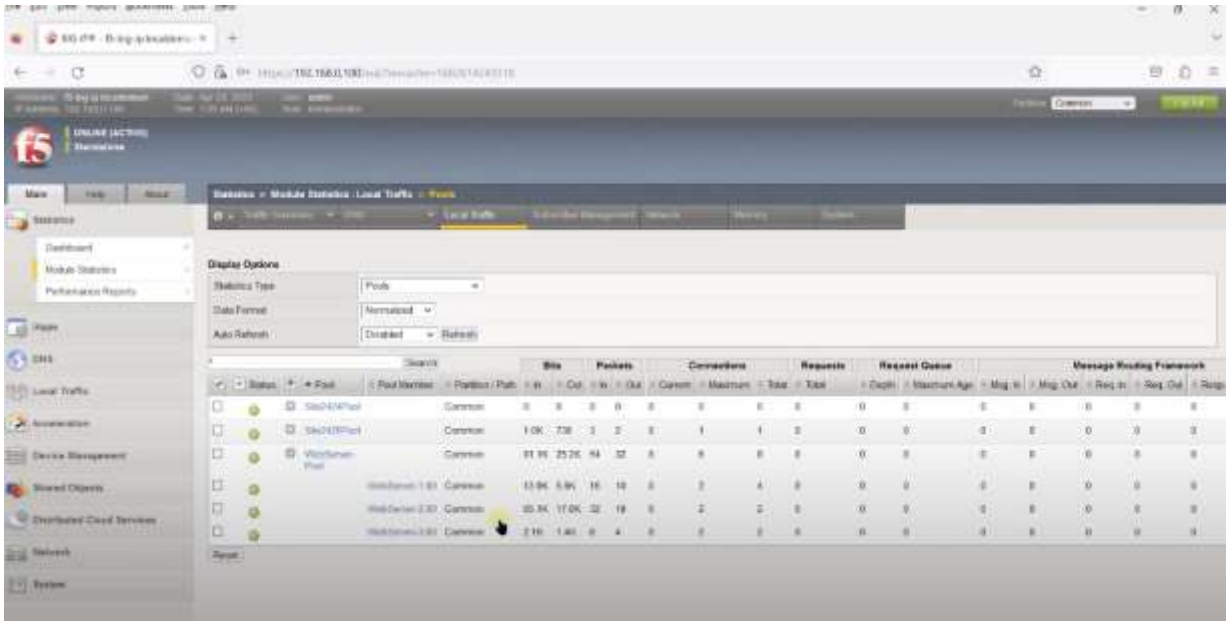
Усі запити були направлені на перший сервер, тому що з ним було найшвидше з`єднання.

Також було налаштовано метод балансування Найменша кількість з`єднань.



Рис

При такому налаштуванні було отримано наступні результати:



Рис

В цьому випадку до кожного з трьох серверів було зроблено по два запити. Це пояснюється тим, що перший запит був направлений на сервер з найменшою кількістю підключень. Оскільки спочатку всі три сервери мали нуль підключень, то перший запит був направлений, наприклад на перший сервер. Другий запит був направлений на другий чи третій сервер, оскільки до першого вже було здійснено одне підключення. Таким чином всі запити рівномірно були розподілені між всіма серверами.

3 РОЗРОБКА РЕКОМЕНДАЦІЙ ЩОДО ПОКАРАЩЕННЯ МЕТОДІВ ЗАБЕЗПЕЧЕННЯ БЕЗПЕКИ ТА ДОСТУПНОСТІ ВЕБ-ДОДАТКІВ

3.1 Рекомендації щодо впровадження додаткових систем забезпечення безпеки веб-додатків та необхідності використання комплексного підходу забезпечення безпеки

Для забезпечення кращого захисту веб-додатків варто розглянути й інші методи, окрім використання F5 Big IP. Найкращим рішенням буде використовувати його в якості одного з багатьох інструментів та заходів. Комплексне використання різних методів та засобів призведе до підвищення рівня захищеності веб-додатків, а також допоможе усунути потенційні ризики.

Для покращення роботи системи безпеки веб-додатків необхідно створити захист від атак низького рівня. Для цього можна використати системи IPS та Firewall. Firewall буде формувати доступ до портів та забезпечувати мінімальний захист, IPS буде відстежувати атаки низького рівня та реагувати на зміни потоків трафіку.

Intrusion Prevention System (система протидії вторгнень) – система, яка розпізнає ознаки вторгнення, виявляє атаки і запобігає їм. Під час аналізу використовуються різні методи виявлення атак – сигнатурний, поведінковий і ідентифікація аномалій в протоколах.

Також, всі види IPS технологій, як правило, виконують наступні функції:

- IPS зупиняє саму атаку.
- Блокує зловмисну частину, дозволяючи неураженій частині проникати до системи.
- Повідомляють адміністраторів безпеки у разі важливих подій, що спостерігаються у системі
- Реагують на інциденти, змінюючи середу безпеки для зривання атаки.
- Створюють звіти.

Загальні вимоги до IPS та типи подій, які найбільш часто виявляються:

- Дослідження і атаки прикладного рівня (наприклад, переповнення буфера, підбір пароля, передача шкідливих програм). Більшість мережевих IPS аналізують протоколи додатків.
- Дослідження і атаки транспортного рівня (наприклад, сканування портів, незвичайна фрагментація пакетів, SYN floods). Найбільш часто аналізуються протоколи транспортного рівня – TCP і UDP.
- Дослідження і атаки мережевого рівня (наприклад, підміна IP-адреси, ненормальні значення заголовка IP).
- Неочікувана робота додатків (наприклад, хости виконують несанкціоновані дії).
- Порушення політики (наприклад, використання заборонених протоколів).

Firewall (міжмережевий екран) – система мережної безпеки, яка відстежує і контролює вхідний і вихідний трафік на основі заздалегідь визначених правил безпеки. Міжмережевий екран, зазвичай, встановлює бар'єр між захищеною внутрішньою мережею і зовнішньою незахищеною мережею. Основною його метою є захист внутрішньої мережі або окремих її вузлів від несанкціонованого доступу. Міжмережевий екран контролює доступ до ресурсів мережі за допомогою позитивної моделі управління (у внутрішню мережу потрапляє тільки дозволений правилами трафік, весь інший трафік заборонений).

Загалом міжмережеві екрани діляться на дві категорії:

- Міжмережеві екрани мережного рівня дозволяють чи забороняють трафік, базуючись на адресах джерела IP і адресах чи портах призначення IP.
- Міжмережеві екрани прикладного рівня аналізують протоколи прикладного рівня, спостерігаючи за активністю протоколу по відношенню до визначеного профілю і дозволяють чи забороняють трафік, базуючись на відхиленнях від профілю.

Типові функції Firewall:

- Контроль доступу до вузлів в мережі

- Фільтрація доступу до незахищених служб
- Контроль порядку доступу до мережі
- Запобігає спробам доступу з зовнішньої і з внутрішньої мережі
- Перешкоджання отримання закритої інформації із внутрішньої захищеної мережі. Таблиця 3.1 зображує на якому рівні моделі OSI працює кожна з систем.

Таблиця 3.1 – Робота систем комплексу на моделі OSI

Рівень моделі OSI	Firewall	IPS	WAF
2	+		
3	+	+	
4	+	+	
5		+	+
6		+	+
7			+

Таким чином, в комплексі, системи будуть захищати інтернет-ресурси на всіх рівнях моделі OSI.

На рисунку 3.1 зображена схема запропонованої інтеграції систем. WAF впроваджується в систему в режимі зворотного проксі-сервера перед захищеними веб-серверами. IPS впроваджується в комплекс в режимі Transparent. Отримуючи запити, допущені Firewall, IPS аналізує протоколи і зупиняє певні види атак, надалі дані передаються до WAF, де обробляються і також блокуються атаки функціоналу WAF.

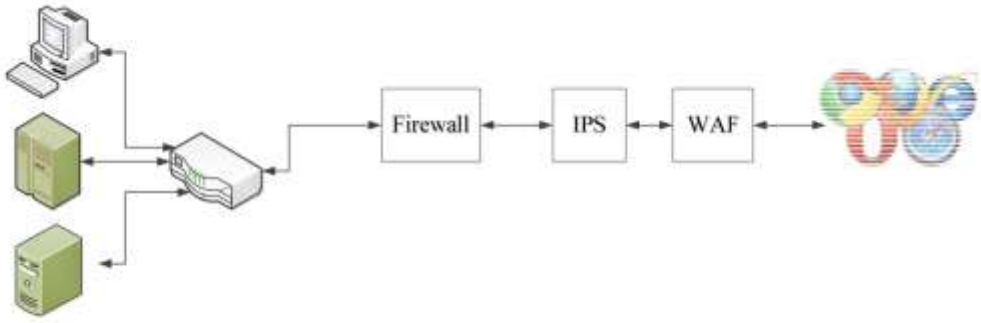


Рис 3.1. Схема інтеграції систем з Web Application Firewall

Відповіді веб-сервера знову повертаються до WAF, де перевіряються на наявність витоку даних. Після перевірки, дані ідуть до користувача.

Використання даної комплексної системи захисту веб-додатків у складі Firewall, IPS та WAF забезпечує на 30% більшу ефективність ніж використання звичайного WAF. Використання сучасних високопродуктивних Firewall та IPS дозволить блокувати потрапляння шкідливих файлів у внутрішню захищену мережу, забезпечить додаткову безпеку і зменшить ризики цілеспрямованих атак на IT-ресурси. Даний комплекс дозволить збільшити захищеність будь-якого інтернет-ресурсу, зменшити навантаження на адміністраторів IT-систем, та забезпечити більш ефективну обробку легітимних користувачів інтернет-ресурсів.

3.2 Рекомендації щодо використання DPI системи для покращення аналізу мережевого трафіку

Найефективнішою системою аналізу мережевого трафіку являється DEEP PACKET INSPECTION, яка позволяє на найвищих рівнях моделі OSI працювати з даними для захисту систем. Потрібно не тільки констатувати інциденти а і притягувати до відповідальності порушників. Ідентифікаційну інформацію, яка «залишається» під час роботи з WEB сервісами представлена у вигляді таблиці 3.2.

Таблиця 5.1 – Ідентифікаційна інформація користувача WEB

«Ідентифікатор»	Зміст ідентифікуючих даних	Спосіб анонімізації
IP-адреса	Як мінімум інформація про провайдера та країну користувача	VPN, Proxy, SSH, Tor, I2P, P2P- анонімайзери
DNS leaks	Витоки інформації від служби доменних імен; протоколювання активності клієнта виникає, якщо програмне забезпечення відправляє DNS-запити через DNS-сервер провайдера	Використання анонімних мереж; під час роботи через VPN використання примусово статичних DNS серверів, що належать VPN провайдеру
MAC-адреса	При підключені до публічної WiFi точки доступу фіксується	Зміна MAC-адреси до сеансу підключення

	MAC-адрес мережного інтерфейсу користувача	
«Профілювання»	Співставлення великого обсягу трафіку, який виходить через один вузол, із конкретним користувачем	Відмова від використання постійних схем (ланцюгів) Tor, регулярна зміна вихідних вузлів
Соціальна активність в анонімному сеансі	Розкриття особи користувача під час відвідування ним власного профілю соціальної мережі, незважаючи на засоби анонімності	Недопущення неузгодженої активності в анонімному сеансі

Ще одним випадком деанонімізації користувача є передавання програмним забезпеченням, зокрема оглядачами (браузерами), різного роду даних, що зазвичай передбачено специфікацією до програмного продукту.

Типовий оглядач містить наступні функціональні компоненти і технологічні категорії:

- cookies – це текстові файли з деякими даними, що їх зберігають прикладні програми для різних задач, наприклад, аутентифікації. Розкриття анонімного клієнта настає, якщо він спочатку відвідав ресурс через відкритий сеанс, браузер зберіг cookies, а потім користувач з’ єднався через анонімний сеанс. В результаті серверу доступно співставлення cookies і, як наслідок, деанонімізація клієнта;
- Flash, Java – плагіни, що ґрунтуються на цих технологіях, завантажуються від імені користувача як окреме програмне забезпечення та можуть працювати в обхід проксі, зберігати свої cookies й інші налаштування;
- відбиток (fingerprint) браузера – оглядач представляє серверу десятки категорій даних, що дає змогу сформувати унікальний цифровий відбиток браузера, за яким його можна ідентифікувати серед багатьох інших навіть в анонімному сеансі (найчастіше застосовується з метою цільової реклами);
- скрипти JavaScript – код, що виконується на стороні клієнта, здатен накопичувати для сервера ідентифікуючу інформацію, а також, за умови вразливості

цільового для користувача ресурсу, створює умови для проведення успішних атак на інформаційний ресурс;

- http-referrer – за допомогою цього http-заголовку цільовий для користувача веб-сайт може визначити, ким було сформовано трафік.

Вирішенням цієї проблеми є налаштування параметрів безпеки оглядача, включаючи блокування кожної із наведених категорій ідентифікації даних, та відмова під час анонімного сеансу від неперевіреного програмного забезпечення.

Система Deep Packet Inspection (DPI, також complete packet inspection і Information eXtraction або IX) — технологія накопичення статистичних даних, перевірки і фільтрації мережевих пакетів по їх вмісту. На відміну від брандмауерів, Deep Packet Inspection аналізує не лише заголовки пакетів, але і повний вміст трафіку на рівнях моделі OSI з другого і вище. Deep Packet Inspection здатна виявляти і блокувати віруси, фільтрувати інформацію, що не задовольняє заданим критеріям, виконує глибокий аналіз усіх пакетів, що проходять через неї. Система DPI здійснює так званий поведінковий аналіз трафіку, який дозволяє розпізнати додатки, що не використовують для обміну даними заздалегідь відомі заголовки і структури даних.

За допомогою DPI спецслужби можуть вести спостереження за мережевою активністю того або іншого користувача та аналізувати VPN, HTTPS трафік. Система DPI може зібрати різну інформацію, не порушуючи особистих прав користувача.

DPI може захистити від:

- Спам-ботів (виявляються на основі аналізу SMTP трафіку).
- DoS і DDoS-атак (виявляються за аномаліями трафіку).
- Зараження вірусами (виявляється за сигнатурами).

Захист від спаму реалізується шляхом блокування відправника, коли з однієї адреси генерується надмірно велика кількість SMTP -запитів.

Система DPI дозволяє захиститися від TCP SYN Flood і Fragmented UDP Flood.

Атака SYN flood викликає підвищену витрату ресурсів системи, оскільки на кожний SYN-пакет, що входить, система повинна зарезервувати певні ресурси в пам'яті або згенерувати велику кількість пакетів, що призводить до її відмови.

DPI виявляє перевищення порогу SYN-запитів, та замість сайту відповідає на них.

Fragmented UDP Flood атака здійснюється фрагментованими udp-пакетами, зазвичай невеликого розміру, на обробку і аналіз яких витрачається багато ресурсів.

DPI відкидає неактуальні для сайту протоколи або обмежує їх по смузі пропускання (для веб-сайту залишаються тільки протоколи HTTP і HTTPS).

ВИСНОВОК

Зі збільшенням кількості загроз веб-додаткам, збільшується необхідність в забезпеченні їх безпеки. Веб-додатки все більше використовуються в повсякденному житті в різних сферах життєдіяльності, від інтернет банкінгу та покупок, до розваг та спілкування. Тому забезпечення безпеки та доступності веб-додатків є ключовою задачею спеціаліста в цій сфері.

Досліджено процес забезпечення безпеки та доступності веб-додатків.

Досліджено технологію забезпечення безпеки та доступності веб-додатків на базі рішення F5 BIG-IP.

Досліджено варіанти розгортання технології забезпечення безпеки та доступності веб-додатків на базі рішення F5 BIG-IP та розроблено рекомендації щодо використання технології.

Проведено аналіз питання щодо необхідності забезпечення безпеки та доступності веб-додатків.

Проаналізовано основні загрози веб-додаткам.

Проаналізовано методи та засоби забезпечення безпеки та доступності веб-додатків.

Досліджено варіант розгортання технології забезпечення безпеки та доступності веб-додатків на базі рішення F5 BIG-IP та розробити рекомендації щодо використання технології.

Варто зазначити, що запропоновані рекомендації не дають сто відсоткової гарантії забезпечення необхідного рівня безпеки та доступності веб-додатків. Проте користуючись рекомендаціями працівник відділу інформаційної безпеки матиме розуміння про існуючі методи та засоби забезпечення безпеки та доступності веб-додатків та направлення на подальше вдосконалення безпеки.

Рекомендації у своїй роботі можуть використовувати керівники відділів інформаційної безпеки та працівники відділу інформаційної безпеки.