Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет __*Інфокомунікацій*_____
(повна назва)
Кафедра _____*Інформаційно-мережної інженерії*_____
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти _____другий (магістерський)_____

Дослідження інструментів безпеки мережевої інфраструктури

(тема)

Виконав:
студент _2_ курсу, групи __*ІМІм-22-3*___
Чалий Д.В.
(прізвище, ініціали)

Спеціальність *172. Телекомунікації та радіотехніка*
(код і повна назва спеціальності)
Тип програми _*освітньо-наукова*_____
(освітньо-професійна або освітньо-наукова)
Освітня програма *Інформаційно-мережна інженерія*
(повна назва освітньої програми)

Керівник ____*ст.викл. Калюжний М.М.*
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ *Безрук В.М.*
(підпис) (прізвище, ініціали)

2024 р.

Не містить відомостей, заборонених
до відкритого публікування

Керівник _____ /*М.М.Калюжний*

Студент _____ / *Д.В. Чалий*

Харківський національний університет радіоелектроніки

Факультет *Інфокомунікацій*

Кафедра *Інформаційно-мережної інженерії*

Рівень вищої освіти другий (магістерський)

Спеціальність *172. Телекомунікації та радіотехніка*

(код і повна назва)

Тип програми *Освітньо-наукова*

(освітньо-професійна або освітньо-наукова)

Освітня програма *Інформаційно-мережна інженерія*

(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

«_18_» *березня* 2024 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові *Чалому Дмитру Владиславовичу*

(прізвище, ім'я, по батькові)

1. Тема роботи *Дослідження інструментів безпеки мережевої інфраструктури*

затверджена наказом університету від 18 березня 2024 р. № 232 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 21 червня 2024 р.

3. Вихідні дані до роботи *Дослідити інструменти безпеки системи захисту мережі від зловмисних впливів. з використанням IDS/IPS компонент (з відкритим кодом) для випадку, коли мережа містить порядку 100-200 кінцевих вузлів. Її веб-компонентою є корпоративний сайт, що знаходиться на зовнішньому хостингу. Обгрунтувати вибір та дослідити процес налаштування інструментарію захисту периметру, внутрішньої мережі та веб-компоненти.*

4. Перелік питань, що потрібно опрацювати в роботі *Вступ*

1. Типова мережева інфраструктура підприємства

2. Побудова захисту зовнішнього контура мережі на базі міжмережевого екрану

3. Захист внутрішнього середовища на базі засобів виявлення та протидії зловмисним втручанням

4. Захист веб-компоненти мережевої інфраструктури

Висновки

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) _____
слайди презентації в форматі Power Point (назва та мета роботи, актуальні питання побудови системи комплексного захисту мережі від зловмисних впливів, використання міжмережевого екрану для захисту периметру мережі, NIDS Snort, захист вебзкомпоненти мережевої інфраструктури, висновки)

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Вступ	21.03.2024	виконано
2	Типова мережева інфраструктура підприємства	02.04.2024	виконано
3	Побудова захисту зовнішнього контура мережі на базі міжмережевого екрану	11.04.2024	виконано
4	Захист внутрішнього середовища на базі засобів виявлення та протидії зловмисним втручанням	27.04.2023	виконано
5	Захист веб-компоненти мережевої інфраструктури	18.05.2024	виконано
6	Висновки	05.06.2024	виконано
7	Оформлення пояснювальної записки	07.06.2024	виконано

Дата видачі завдання *18 березня* 2024 р.

Студент _____
(підпис)

Керівник роботи _____ *ст. викл. Калюжний М.М.*
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 78 с., 29 рис., 22 джерела, 3 додатки

SNORT, KEENETIC, NIDS-система, DDoS, ІНФОРМАЦІЙНА БЕЗПЕКА, ЗЛОВМИСНИЙ МОДУЛЬ, HTTPS

Об'єкт дослідження – методи та засоби побудови комплексного захисту мережевої інфраструктури від зловмисних впливів.

Мета роботи – дослідити ключові компоненти та механізми системи захисту мережі.

Виконується вибір засобів захисту мережевої інфраструктури на рівні периметру, на рівні внутрішнього середовища та веб-компоненти. Досліджується процес розгортання та налаштування обраних засобів. Здійснюється дослідження процедур захисту від атак відмови обслуговування, атак підбору паролю та захисту від розміщення зловмисного коду на корпоративному сайті.

THE ABSTRACT

Explanatory note: 78 p., 29 fig., 22 sources, 3 apps.

SNORT, KEENETIC, NIDS-system, SNORT, DDoS, INFORMATION SECURITY, MALICE MODULE, HTTPS

The object of research is methods and means of building complex protection of network infrastructure from malicious influences. The purpose of the work is to investigate the key components and mechanisms of the network protection system. Selection of means of protection of the network infrastructure at the level of the perimeter, at the level of the internal environment and the web component is performed. The process of deployment and adjustment of the selected means is being studied. The study of protection procedures against denial-of-service attacks, password-picking attacks, and protection against placing malicious code on a corporate website is being conducted.

ЗМІСТ

С.

ПЕРЕЛІК СКОРОЧЕНЬ

NGFW – (Next Generation Firewall) – міжмережевий екран наступного покоління;
IDS – (Intrusion Detection System) – система виявлення вторгнень;
APT – (Advanced Persistent Threat) – розвинута стійка загроза;
SSL – (Secure Sockets Layer) – протокол мережевої безпеки на базі засобів криптографії;
VPN – (Virtual Private Network) – віртуальна приватна мережа;
xDSL – (Digital Subscriber Line) – сімейство технологій доступу;
ICMP – (Internet Control Message Protocol) – протокол міжмережевих керуючих повідомлень;
HTTPS – (HyperText Transfer Protocol Secure) – захищений протокол прикладного рівня передачі гпертексту;
HTTPS – (HyperText Transfer Protocol) – захищений протокол прикладного рівня передачі гпертексту;
TCP – (Transmission Control Protocol) – протокол управління передачею;
NIDS – (Network Intrusion Detection System) – мережева система виявлення вторгнень;
SMTP – (Simple Mail Transfer Protocol) – протокол передавання пошти;
ODBC – (Open Database Connectivity) – програмний інтерфейс доступу до БД;
WAF – (Web Application Firewall) – міжмережевий екран веб-додатків;
CSRF – (Cross-Site Request Forgery) – міжсайтова підробка запиту.

ВСТУП

Питання захищеності мережевих ресурсів та корпоративних даних зараз є одними з найприорітетніших у предметній галузі, орієнтованій на забезпечення ефективного функціонування інформаційної інфраструктури підприємства.

При цьому, відповідність корпоративних даних вимогам захищеності визначається їх властивістю конфіденційності. У загальному випадку, властивість конфіденційності даних вказує на те, що доступ до них (а також його тип - читання, редагування, створення копій, видалення тощо) має обмежене коло осіб [1].

Окрім конфіденційності, дані також мають відповідати вимогам цілісності та доступності. Тут під доступністю мається на увазі властивість, що гарантує можливість отримання доступу будь-якій авторизованій особі у будь-який момент часу та у об’ємі, який визначається повноваженнями авторизованої особи.

У свою чергу, властивість цілісності вказує на те, що за будь-яких обставин дані не можуть зазнати несанкціонованої модифікації, заміни або видалення.

Зазначимо також, що попри те, що ключові характеристики даних, а саме – конфіденційність, цілісність та доступність – зазвичай розглядаються окремо – водночас, між ними існує чіткий взаємозв’язок.

По-перше, конфіденційність, або захищеність даних, створює передумови для забезпечення їх цілісності.

По-друге, доступність даних може гарантуватися в умовах, коли відсутні фактори зловмисного цілеспрямованого (або випадкового) впливу, та забезпечено технологічну коректність побудови механізмів, які відповідають за зберігання даних, та надання їх користувачеві з відповідними повноваженнями за запитом.

Тобто, як цілісність, так і доступність даних можуть розглядатися як похідні від їх захищеності, що зумовлює першочергову необхідність забезпечення захищеності даних.

Разом з тим, забезпечення захищеності даних є комплексне завдання, яке, у свою чергу, вирішується на декількох рівнях, а саме:

- організаційно-правовий;
- фізичний;
- програмно-технічний.

При цьому, на кожному з зазначених рівнів, процедура забезпечення захищеності даних зводиться до протидії зловмисним впливам з зовнішньої мережі, або з середини захищуваного мережевого простору.

Приймаючи до уваги усе вищезазначене є очевидним, що усі питання вивчення, розробки та впровадження механізмів, методів та засобів захисту мережевих ресурсів сьогодні є актуальними.

1. ТИПОВА МЕРЕЖЕВА ІНФРАСТРУКТУРА ПІДПРИЄМСТВА

1.1 Розвиток мережевої інфраструктури підприємства

Поняття «корпоративна мережа» (частіше – комп'ютерна мережа) в межах країни отримало розповсюдження ближче до кінця 90-х років XX століття [2]. Така мережа являла собою невелику кількість (не більше кількох десятків) кінцевих вузлів, поєднаних частіше за все загальною шиною (10 Base-2/5).

Дана модель не отримала широкого розповсюдження на національному рівні, а згодом її витіснила модель на базі технології 10 Base-T, що базувалася на використанні концентраторів, при цьому, у даному випадку у якості фізичного носія розглядався кабель UTP cat 3.

Зазначені мережі мали досить лімітований канал взаємодії з зовнішнім середовищем, що створювався одним з зазначених далі способів:

- на базі модемного з'єднання, яке забезпечувало комунікацію зі швидкістю не більш, ніж 56 Кбіт/с;

- з використанням xDSL-з'єднання, що, залежно від конкретної використаної технології, дозволяло досягти обміну даними з зовнішньою мережею на швидкості приблизно 10-20 Мбіт/с.

Зрозуміло, що у зазначених умовах, можливості зловмисного впливу було жорстко лімітовано реальною пропускною здатністю мережі, що стосувалося:

- можливості інтеграції зловмисних модулів усередину мережі та їх подальшого розповсюдження;

- можливості викрадення даних зловмисними модулями та їх надсилання до зовнішніх вузлів.

Інакше кажучи, в умовах досить лімітованої швидкості обміну даними вузлів мережі з зовнішніми ресурсами, важливість питання інфікування даних/додатків зловмисними модулями частково нівелювалася за рахунок обмеженої можливості витоку даних [3].

У свою чергу, у таких умовах мережева інфраструктура підприємства переважно зводилася виключно до сукупності кінцевих вузлів та серверу/серверів у складі окремої локальної мережі (рис.1.1).

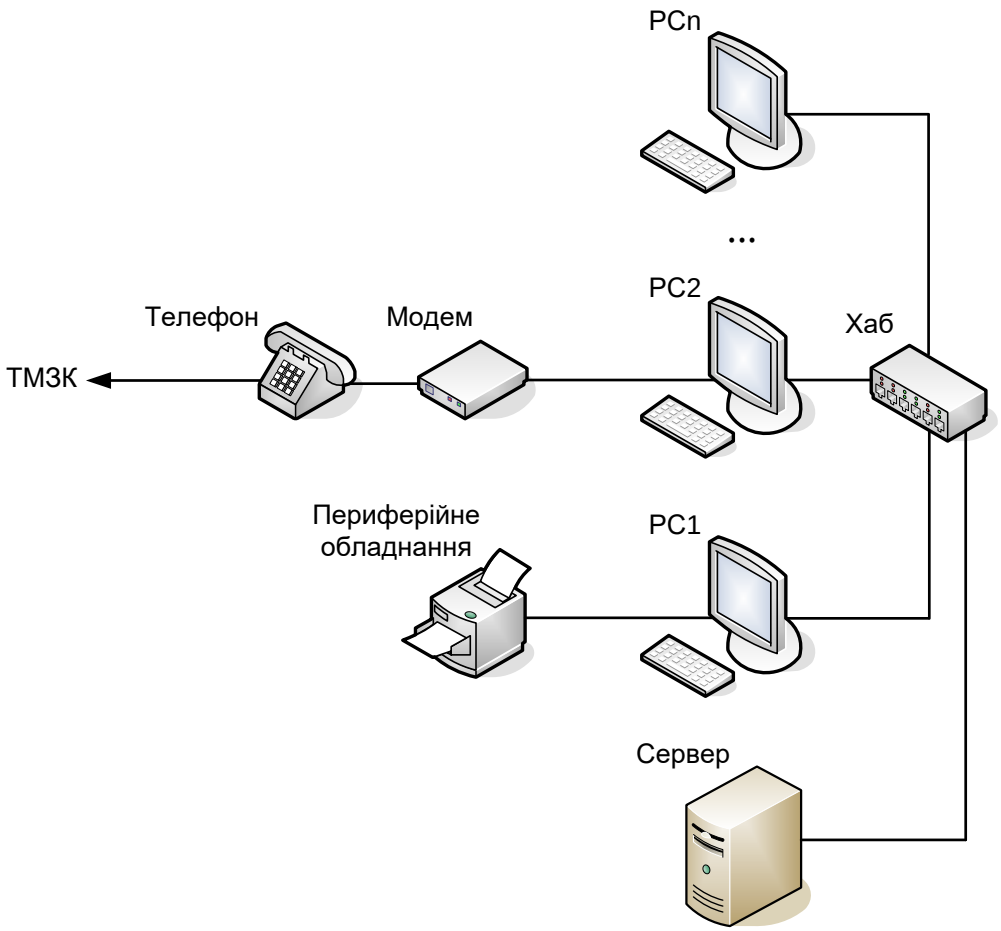


Рисунок 1.1 – Типова структура локальної мережі з виходом у зовнішнє середовище за участю dial-up модему

У зазначених умовах найбільшу потенційну небезпеку становили вірусні загрози.

При цьому, масштаби розповсюдження таких загроз обмежувалися, з одного боку, відносно низькою (порівняно з нинішньою) швидкістю обміну даних у мережі а з іншого боку – поширеним на той час регламентом роботи мережі. У рамках даного регламенту для обміну даними з зовнішнім оточенням нерідко достатньо було використовувати лише єдиний мережевий ресурс на рівні окремого кінцевого вузла.

Власне, якщо описувати регламент функціонування мережі, побудованої на зазначених принципах, включав у себе такі режими, як [4]:

- з доступом до зовнішнього середовища виключно єдиного вузла (на схемі 1.1 це відповідає PC2), який має безпосереднє з’єднання з модемом;

- з можливістю доступу до зовнішньої мережі будь-якому вузлу (для цього випадку у їхніх опціях з’єднань адреса PC2 вказується у якості шлюзу).

Це, у підсумку, зумовлювало помірні темпи інфікування мережевих ресурсів.

У свою чергу, протидію зловмисним впливам було зведено до розгортання антивірусних засобів на кожному окремо узятому кінцевому вузлі (робочій станції та/або серверу) [2, 5].

Разом з тим, як наслідок, з одного боку, розвитку технологій а з іншого боку – наслідок розвитку самих підприємств стало те, що у складі базових компонент мережевої інфраструктури підприємства з’явилися нові складові.

Таким чином, у складі мережевої інфраструктури можливо стало виокремити такі компоненти, як [5]:

- локальна компонента;
- віддалена мережева компонента;
- веб-компонента.

Тут під локальною компонентою мається на увазі сукупність ресурсів локальної мережі підприємства, ураховуючи кінцеві та мережеві вузли, периферійні пристрої а також дані, що можуть зберігатися у межах локальних серверів/робочих станцій.

Водночас, віддалена мережева компонента являє собою окремий віддалений сегмент мережі, як то:

- локальні мережі філій чи відділів підприємства, що географічно знаходяться на віддаленні від головної мережі на відстань, що перевищує комунікативні можливості технологій локальних мереж;
- окремих користувачів мережі (співробітників, які функціонують у віддаленому режимі), для яких є справедливим попереднє твердження.

У той же час, мінімізованим прикладом веб-компоненти є внутрішній або зовнішній сервер (чи їх сукупність) - власний, чи арендований, до якого користувачі звертаються з використанням протоколів FTP, HTTP/HTTPS, або повноцінний веб-ресурс підприємства (рис.1.2).

Таким чином, на сьогодні склалася ситуація, коли швидкість обміну даними як усередині локальних сегментів мережі, так і на ділянках «локальна мережа – зовнішній веб-сервер» та «локальна мережа – віддалений сегмент» чи «віддалений сегмент – веб-сервер» практично зрівнялася, як наслідок [2]:

- розвитку мережевих технологій, що виявився, у т.ч. у зростанні пропускної здатності каналів мережі;
- розвитку технологічного базису, що виявився, зокрема, у зростанні обчислювальних потужностей мережевих та кінцевих вузлів та зумовив збільшення їхніх можливостей з прийому та наступної обробки відносно великих обсягів даних в одиницю часу.

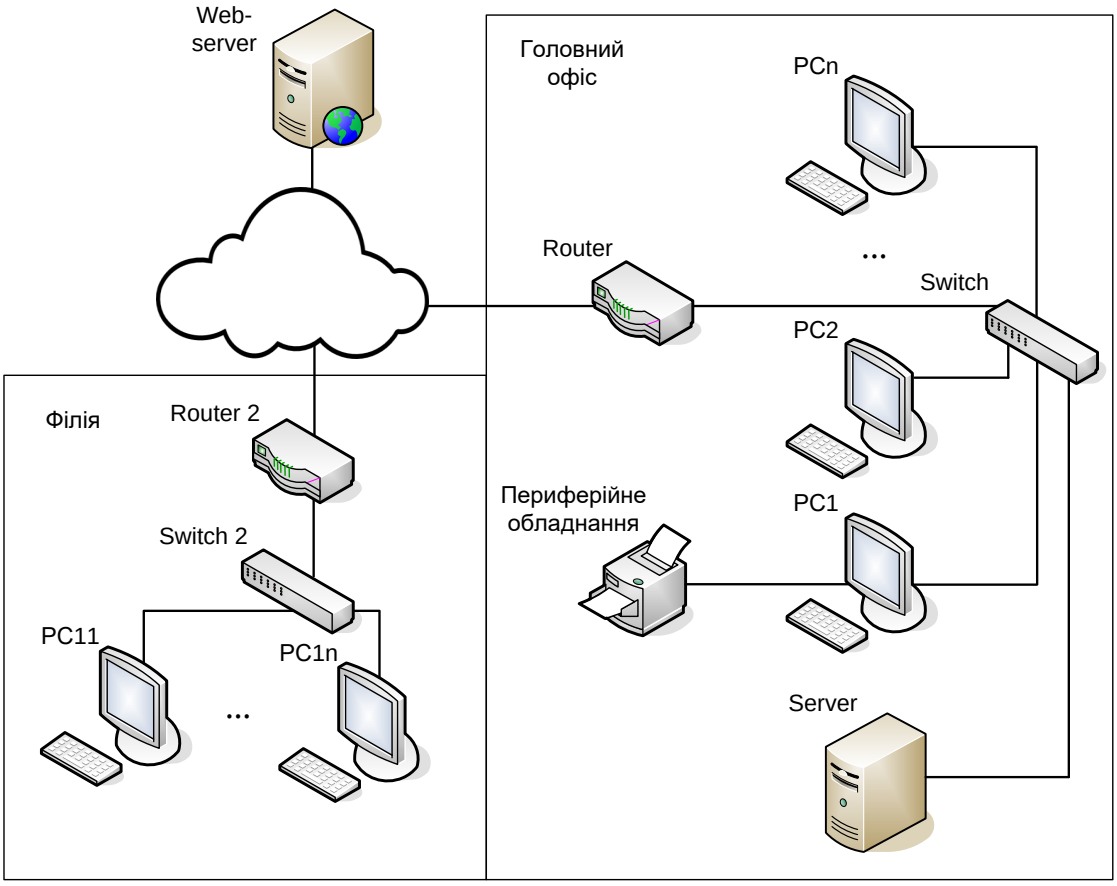


Рисунок 1.1 – Типова структура мережі з локальною та віддаленою компонентами а також веб-складовою

Отже, беручи до уваги усе вищезазначене, а також спираючись на ґрунтовні дослідження з розвитку інформаційного простору, можемо констатувати, що останні 10-15 років у галузі інфокомунікацій склалася парадоксальна ситуація, яку може бути описано двома сукупностями тверджень [2]:

1. Суттєвим чином збільшилася кількість користувачів мережевих сервісів. Це, у свою чергу, стало наслідком:
- розвитку інформаційно-комунікаційних технологій;

- зниження вартості клієнтських терміналів та послуг з доступу до мережі.

2. Значно розширився перелік як категорій поширених мережевих сервісів, так і, власне, сервісів у межах кожної з категорій, що було зумовлено:

- збільшеними технологічними можливостями у наслідок удосконалення технологічного базису;
- сформованої потреби з боку користувацької аудиторії.

3. Кардинальним чином зросла пропускна здатність каналів мереж доступу, що зумовилося впровадженням більш досконалих стандартів обміну даними.

З іншого боку [6, 7]:

1. Потенційний зловмисник отримав у розпорядження значні можливості з реалізації кібератак у наслідок:

- розширення обчислювальних можливостей окремого кінцевого вузла, який прямо чи опосередковано бере участь у реалізації тієї чи іншої атаки;
- розвитку хмарних сервісів та grid-технологій, які надають обчислювальні потужності для реалізації ресурсоємних атак.

2. Ріст пропускних здатностей мереж локального сегменту створив умови для інтенсивного обміну даними, у т.ч. для швидкого розповсюдження зловмисних модулів усередині локальної мережі.

3. Збільшення пропускної здатності на ділянці «віддалений вузол – вузол локальної мережі» створив умови для більш інтенсивного впливу потенційного чи реального зловмисника на атаковану мережеву інфраструктуру.

Тобто, у цьому випадку потенційно неможливо забезпечити захищеність даних, що зберігаються у файлосховищах та/або надсилаються мережею.

Таким чином, у зазначених умовах маємо протиріччя, яке може бути ілюстроване рисунком 1.3.

Отже, одним з нагальних питань забезпечення ефективного функціонування мережевої інфраструктури на сьогодні стає забезпечення захищеності від зловмисних впливів як окремих мережевих компонент, так і мережевої інфраструктури у цілому.

1.2 Дані, які підлягають захисту від зловмисних впливів

У загальному випадку під терміном «мережева інфраструктура» розуміється [1]:

- сукупність мережевих пристроїв у складі тієї чи іншої мережі;
- множина кінцевих вузлів (з урахуванням серверів різного формату);
- файлосховища та дані, які вони зберігають;
- мережеве програмне забезпечення;
- прикладне ПЗ, яке може використовуватися у т.ч. на мережевому рівні.

Рисунок 1.3 – Протиріччя у розвитку мережевої інфраструктури з точки зору інформаційної безпеки

При цьому, як свідчить існуюча статистика, кожна з зазначених складових може стати об’єктом зловмисного впливу.

Разом з тим, якщо розглядати різні формати зловмисного впливу, необхідно зазначити, що першочерговими об’єктами тут є [1, 4, 8, 9]:

- безпосередньо файлосховища, які містять дані, що є потенційно цінними для зловмисника;
- облікові записи користувачів.

Так, перший випадок є актуальним тоді, коли головним завданням є крадіжка, видалення, публікація у загальний доступ або модифікація конфіденційних даних.

У другому випадку вважається, що отримавши у розпорядження облікові дані користувачів, зловмисник, використовуючи їх, отримує набагато ширші можливості порівняно з попереднім випадком. Зокрема, при цьому він може отримувати змогу:

- доступу до файлосховища а відтак – усіх даних, які у ньому зберігаються;
- обходу щонайменше ряду модулів системи безпеки (а у деяких випадках – повного обходу);
- користуватися програмними засобами, у т.ч. тими, що відносяться до компоненти мережевої безпеки;
- у цілому, теоретично виконувати будь-які дії усередині атакованої мережі (водночас, слід пам’ятати, що програмне середовище, архітектура інформаційної системи та особливості її побудови можуть накладати певні обмеження).

Водночас, якщо, у першу чергу, не брати до уваги типоналежність інформаційної системи, як потенційного об’єкту атаки, слід зазначити, що найчастішими об’єктами атаки є (рис.1.4):

- платіжні системи та пов’язані з ними облікові записи;
- засоби взаємодії з веб-об’єктами (панелі керування хостингом, системи управління контентом (CMS) тощо);
- веб-сервіси та веб-додатки керування об’єктами критичної інформаційної інфраструктури (КІІ);
- облікові записи адміністраторів та користувачів мережі, її окремих сервісів;
- облікові записи адміністраторів та користувачів мережевої веб-компоненти.

У цьому випадку під обліковими записами адміністраторів, а також користувачів мережевої веб-компоненти розуміються такі, які, у свою чергу дозволяють забезпечити взаємодію з існуючими веб-ресурсами підприємства,

виконуючи при цьому їх конфігурування, адміністрування та/або контентне наповнення.

Отже, в існуючих умовах першочерговим завданням є забезпечення захищеності зазначених вище ймовірних об’єктів атаки.

Далі виконаємо аналіз найбільш поширених типів зловмисних впливів на інформаційні ресурси.

1.3 Найпоширеніші типи зловмисних впливів

Частіше за все зловмисний вплив реалізується за участю зловмисного ПЗ тієї чи іншої архітектури.

У свою чергу, зловмисне ПЗ являє собою програмний засіб, який здійснює несанкціонований доступ до даних та/або чинить вплив на дані та/або ресурси інформаційної системи.

Тобто, зловмисним ПЗ є певна самостійна сукупність програмних інструкцій, що здатна виконувати наступний перелік дій [7, 10]:

1. Приховувати факт своєї присутності в інформаційному сегменті кінцевого вузла.
2. Мати здатність до самовидалення, а також маскування з імітацією легальних програмних засобів та копіювання себя у інші довільні ділянки оперативної або зовнішньої пам’яті.
3. Виконувати процедуру модифікації (руйнування, спотворення) коду будь-яких інших програм (або програм певного типу).
4. Забезпечувати самостійне виконання *деструктивних дій, а саме* – копіювання, знищення, модифікацію, блокування інформації тощо.
5. Спотворювати, замінювати чи блокувати дані, що виводяться у зовнішній канал зв’язку та/або на зовнішній носій інформації.

Зазначимо, що зазвичай ключовими шляхами проникнення зловмисного ПЗ як до інформаційної системи взагалі, так і, зокрема, до кінцевих вузлів, є:

- мережева взаємодія;
- знімні носії інформації, як-то флеш-накопичувачі, диски тощо.

При цьому зауважимо, що інтеграція до системи зловмисних компонент може мати випадковий характер.

У свою чергу, до найбільш поширених класів зловмисного ПЗ сьогодні відносяться:

- програмні закладки;
- програмні віруси;
- мережеві черви;
- зловмисні програми інших класів, спрямовані на реалізацію несанкціонованого доступу.

1.3.1 Програмні закладки

До категорії програмних закладок сьогодні може бути віднесено як повноцінне ПЗ, так і и фрагменти коду програм, які призначаються для побудови та реалізації *недекларованих можливостей* легального програмного забезпечення [10].

У свою чергу, недекларовані можливості програмного забезпечення формально являють собою додаткові функції ПЗ, опис яких відсутній в офіційній супровідній документації.

При цьому, програмна закладка нерідко є своєрідним провідником для іншого зловмисного ПЗ та, частіше за все, на базі більшості поширених засобів антивірусного контролю не виявляється.

Окремо розглядаються закладки програмного та апаратного типів. Утім, майже усі закладки є програмними. У свою чергу, під апаратними закладками розуміються так звані прошивки.

За принципом проникнення до системи закладки діляться на [8, 10]:

- програмно-апаратні (вбудовані в програмно-апаратні засоби ПК (BIOS, прошивки периферійного обладнання);
- завантажувальні (інтегровані у ПЗ початкового завантаження, яке міститься завантажувальних секторах;
- драйверні;
- прикладні (інтегровані у прикладне ПЗ);
- виконувані (вбудовані до програмних модулей, таких, як пакетні файли);
- закладки-імітатори (мають інтерфейс такий, як і у легального ПЗ, де необхідно вводити конфіденційну інформацію.

Виявлення програмних закладок зазвичай базується на якісному підході, що зводиться до спостереження за функціонуванням системи та може проявлятися у вигляді:

- зниженні загальної швидкодії;
- зміні переліку та довжин файлів;

– частковому чи абсолютному блокуванні як окремих компонент системи так і системи у цілому;

– імітації апаратних збоїв у функціонуванні обчислювальних засобів та периферії;

– аномальної переадресації повідомлень;

– обходу засобів криптографічного перетворення даних;

– появи можливості доступу до системи з несанкціонованих пристроїв.

Існують також діагностичні методи виявлення закладок. Зокрема, антивірусне ПЗ ефективно знаходить завантажувальні закладки.

До найпоширеніших програмних закладок відноситься т.з. "троянський кінь", або Trojan, що являє собою [7, 10]:

- ПЗ, що є частиною іншого програмного засобу з відомими користувачеві функціями, але може потай від нього виконувати ряд додаткових дії, спрямованих на заподіяння йому певної шкоди;

- ПЗ з відомими його користувачеві функціями, відносно якого було здійснено певні зміни, для того, щоб забезпечити можливість потай від користувача виконувати деякі інші (руйнівні) дії.

Зараз існує велика кількість троянських програм. Розглянемо найпоширеніші серед них:

– Trojan-Notifier сповіщає зловмисника, якій ініціює атаку, про її успішне виконання. У цьому випадку зловмисникові надсилаються дані щодо комп'ютеру. Це може бути IP-адреса, номер відкритого порту, e-mail тощо.

– Trojan-PSW. Дані зловмисні модулі виконують крадіжку паролів. Вони викрадають конфіденційну інформацію з клієнтських вузлів та надсилають її зловмисникові, зазвичай використовуючи e-mail.

– Trojan-Clicker, або Інтернет-клікери. Являють собою сімейство зловмисного ПЗ, що реалізують несанкціоновані звернення до Інтернет-ресурсів. Тут можуть використовуватися різні методи. Наприклад, це може бути встановлення зловмисної сторінки у браузері, як домашньої.

– Trojan-DDoS. У результаті інфікування перетворюють комп'ютер на бот, що далі використовується для реалізації DDoS-атак на певний ресурс.

– Trojan-Proxy. Сімейство троянських проксі-серверів. Це зловмисне ПЗ, яке таємно звертається до різних Інтернет-ресурсів. Частіше за все використовуються для розсилки спаму.

– Trojan-Spy являє собою сімейство шпигунських програм, що стежать за діями користувача на інфікованому вузлі, та надсилають зібрані дані

зловмисникові. Зазвичай це паролі, аудіо та відео файли з мікрофону та/або веб-камери тощо.

– Backdoor. Один з найнебезпечніших типів троянського ПЗ. Може виконувати віддалене управління інфікованим вузлом. Має практично безмежні можливості. За результатом інтегрування Backdoor зловмисник може розсилати повідомлення від імені користувача, мати доступ до усіх даних або зруйнувати систему та повністю видалити дані.

– Trojan-Dropper. Інсталятори іншого шкідливого ПЗ. Подібні до Trojan-Downloader, але вони встановлюють те зловмисне ПЗ, що міститься усередині них.

– Rootkit. Один з найбільш важких типів троянського ПЗ з точки зору виявлення та подальшого видалення. Здатні маскуватися у системі, підмінюючи собою різні об'єкти. Можуть замінювати своїм програмним кодом вихідний код операційної системи, що не дозволяє антивірусу виявити наявність зловмисного модулю.

Будь-які програмні закладки, незалежно від методу інтеграції до системи, часу їх знаходження в оперативній пам'яті та спрямованості, мають спільну рису. Це - обов'язковий запис в оперативну або зовнішню пам'ять системи. За відсутності можливості виконати дану операцію, програмна закладка не може здійснити жодного зловмисного впливу.

1.3.2 Віруси

Вірусом (програмним або комп’ютерним) може бути програмний код, чи інтерпретований набір інструкцій, який має властивості несанкціоновано поширюватися системою та самовідтворюватися. Отримані таким чином дублікати комп'ютерного вірусу можуть не збігаються з оригіналом, проте зберігають базовий функціонал оригіналу [7, 8, 10]:

- специфічний зловмисний набір функцій;
- можливість подальшого розповсюдження;
- можливість самовідтворення.

Отже, характерною рисою програмного вірусу є створення своїх копії та інтегрування їх в обчислювальні мережі та/або файли, системні області ПК та інші об'єкти.

Життєвий цикл вірусу містить у собі ряд етапів, а саме:

– проникнення до вузла;

– активація;

- пошук об'єктів для інфікування;
- створення вірусних копій;
- застосування копій згідно закладеного специфічного функціоналу.

Вірусний код завантажувального типу, як один з поширених, дозволяє перехопити управління ПК ще на стадії ініціалізації, що передує запуску самої системи. Для цього завантажувальні віруси записуються або до boot-сектору, або до сектору, що містить системний завантажувач дискового пристрою, або виконують заміну покажчика на активний boot-сектор. Принцип дії завантажувальних вірусів заснований на алгоритмах запуску ОС у ході включення та/або перезавантаженні комп'ютера. Так, після виконання тестів інстальованого обладнання (пам'яті, дисків тощо) програма системного завантаження виконує зчитування першого фізичного сектору завантажувального диска, а далі передає керування на той дисковий пристрій, що встановлений у BIOS Setup.

У разі використання зовнішнього диска (flash-носій, мобільший SSD-модуль тощо) управління отримує boot-сектор диска, який аналізує його таблицю параметрів (BPB — BIOS Parameter Block), знаходить адреси системних файлів ОС, після чого зчитує їх у пам'ять та далі запускає на виконання. Системними файлами, наприклад, можуть бути MSDOS.SYS та IO.SYS, або IBMDOS.COM разом з IBMBIO.COM, чи інші файли, виходячи з встановленої ОС. Якщо на завантажувальному диску відсутні файли операційної системи, програма, розташована в boot-секторі диска, видає повідомлення про помилку та пропонує замінити завантажувальний диск.

Разом з тим, у випадку, коли використовується стаціонарний дисковий носій, управління отримує програма, розташована в його MBR. Дана програма виконує аналіз таблиці Disk Partition Table, знаходить адресу активного boot-сектору (за замовчуванням цим сектором є boot-сектор диска C:), після чого завантажує їх у пам'ять і передає управління. Отримавши керування, активний boot-сектор дискового пристрою виконує ті ж дії, що і boot-сектор зовнішнього носія.

У ході інфікування дисків завантажувальні віруси виконують запис свого коду замість коду будь-якої програми, що може отримувати керування у процесі завантаження системи. Тобто, принцип інфікування є аналогічним для усіх розглянутих способів. А саме: вірус змушує систему при перезапуску зчитати у пам’ять та віддати управління не оригінальному коду завантажувача, а коду вірусу [10].

У свою чергу, файлові віруси виконують інфікування безпосередньо файлів. Залежно від середовища розповсюдження, усі файлові віруси може бути розділено на ряд підгруп, а саме [8, 10]:

1. *Орієнтовані на інфікування поширених типів файлів*. Функціонують безпосередньо на рівні ресурсів ОС. Наприклад, велика кількість вірусів, завдяки невеликому розміру (1 Кб), здатні інфікувати PE-файлы таким чином, що їх розмір не змінюється. Для цього вірус знаходить у файлах незаповнені ділянки, що виникають у наслідок вирівнювання початку кожної секції файлу під кратні значення байт. Після отримання управління, вірус захоплює IFS API, далі відстежує функції звернень до файлів та інфікує файли виконуваних типів. Коли створюються ті чи інші умови, спрацьовують закладені у нього деструктивні функції. Наприклад, для віруса "Чорнобиль" деструктивна функція активується 26 квітня, та виявляється у стиранні *Flash BIOS* та початкових секторів дискових носіїв.

2. Макровіруси. Це тип зловмисного ПЗ, створене з використанням макромов, що можуть бути вбудовані до деяких прикладних систем обробки даних. Це, зазвичай, текстові редактори, електронні таблиці тощо. Найбільш розповсюдженими у даній категорії є віруси для додатків пакету Microsoft Office. Розмноження вірусів даного типу виконується з використанням можливостей макромов, власне, на їх базі виконується перенесення копій вірусу від одного документу до інших.

Найбільш характерні макровіруси для пакетів додатків Microsoft Word, Excel та Microsoft Access. У складі кожного з них містяться макромови Word Basic, Visual Basic for Applications.

При цьому, переважна більшість макровірусів *є* активними не лише при закриттю чи відкриттю файлу, але і протягом усього часу, доки сам додаток є активним. Функціонал таких вірусів являє собою стандартні макроси Word/Excel/Office.

3. *Мережеві віруси*. До них належать такі віруси, що розповсюджуються на базі протоколів як локальних, так і глобальних мереж. Ключовою відмінною рисою мережевого вірусу є можливість самовідтворення у мережевому середовищі. Окрім того, існує окрема категорія мережевих вірусів, які самостійно здатні активуватися на серверах або будь-яких віддалених кінцевих вузлах взагалі.

У свою чергу, базові деструктивні дії, які можуть реалізувати як мережеві віруси, так і черви (Worm), це:

– перевантаження мережевих каналів;
– участь в атаках «відмова в обслуговуванні (DDoS)»;
– видалення даних;
– ініціювання збоїв у роботі ПЗ;
– перевантаження ресурсів кінцевого вузла;
– крадіжка даних.

Також слід зазначити, що ряд вірусів можуть поєднувати у собі базові риси вірусів різних типів. Це можуть бути, наприклад, характерні риси файлового вірусу та вірусу-завантажувальника.

Разом з тим, деяка частина сучасних вірусів застосовує алгоритми маскування на стадії формування копій. За рахунок цього суттєво утруднюється процедура їх виявлення стандартизованим антивірусним ПЗ. До засобів маскування вірусів відносяться:

– криптографічна обробка, або шифрування. У цьому випадку вірус складається з самого тіла вірусу. А також шифратору. На цей випадок кожна створювана копія вірусу містить у собі шифратор, випадковий секретний ключ та код вірусу, попередньо зашифрований з використанням випадкового ключа.

– метаморфізм, або створення різнорідних копій вихідного вірусу, що здійснюється за рахунок:

– заміни блоків команд на можливий еквівалент;
– виконання процедури перемішування частин вихідного коду;
– використання вставок проміж значущих відрізків коду з т.з. "сміттєвих" команд, котрі, у сутності, не виконують фактично ніяких дій.

У свою чергу, поєднання вищезазначених засобів маскування вірусів є передумовою до появи специфічних вірусів наступних типів [7]:

– зашифрований вірус, що базується на використанні простого шифрування на базі випадкового ключа та статичного шифратору. Частіше за все. Віруси даного типу можуть бути відносно легко виявлені на базі сигнатури шифратору.

– метаморфний вірус, або вірус, який застосовує прийоми метаморфізму до свого тіла у процесі самовідтворення;

– вірус поліморфного типу, який може застосовувати метаморфний шифратор для шифрування базового тіла вирусу на базі випадкового ключа. У цьому випадку деяка частина даних, що

використовується для формування нових копій шифратору, також може бути шифрованою. Зокрема, може бути передбачено для застосування ряд криптографічних алгоритмів, при цьому, у ході створення наступних копій вірусу зазнаватимуть зміни як команди самого шифратору, так і використовуваний алгоритм шифрування.

1.3.3 Зловмисне ПЗ типу Worm

Worm, або черв**,** являє собою специфічний тип зловмисного ПЗ, яке може розповсюджуватися мережевими каналами, є здатним автономно долати мережеві системи захисту, окрім цього - створювати та розповсюджувати власні копії, які. як і у випадку вірусів, можуть не співпадати з оригіналом, а також реалізувати велику кількість зловмисних впливів різного типу [10].

Протягом стадії проникнення до системи, черви використовують різні методи, що дозволяє їз класифікувати за даною ознакою (дана класифікація базується на використовуваних протоколах):

– мережеві – черви, що розповсюджуються з використанням протоколів локальних та глобальних мереж. Зазвичай розповсюджуються, використовуючи некоректну обробку деякими додатками пакетів стеку tcp/ip.

– поштові – черви, які розповсюджуються у форматі повідомлень електронної пошти. Зазвичай у листі міститься тіло коду або посилання на інфікований ресурс;

– IRC-черви – клас червів, які розповсюджуються каналами IRC (Internet Relay Chat);

– P2P-черви, розповсюдженню яких сприяють пірінгові (peer-to-peer) мережі файлообміну;

– IM-черви – ще один особливий клас, що використовує для ррозповсюдження системи швидкого обміну повідомленнями (IM, Instant Messenger - ICQ, MSN Messenger, AIM тощо).

2. ПОБУДОВА ЗАХИСТУ ЗОВНІШНЬОГО КОНТУРА МЕРЕЖІ НА БАЗІ МІЖМЕРЕЖЕВОГО ЕКРАНУ

2.1 Огляд типів міжмережевих екранів

Ключова функція, яку реалізує міжмережевий екран, полягає у фільтрації трафіку, що проходить в обох напрямках на межі мережі (сегменту мережі). Для реалізації означеної процедури необхідно попередньо сформувати т.з. список доступу, у якому зазначаються усі обмеження для внутрішніх та зовнішніх вузлів. При цьому, такий список може містити [11]:

- перелік внутрішніх об’єктів (додатки, окремі вузли, групи користувачів, протоколи), яким обмежено доступ назовні;
- перелік внутрішніх об’єктів (додатки, окремі вузли, групи користувачів, протоколи), яким повністю блоковано доступ назовні;
- перелік зовнішніх об’єктів, трафік яких повністю блокується;
- перелік зовнішніх об’єктів, трафік яких блокується частково (наприклад, на рівні окремих додатків, користувачів або протоколів).

Таким чином, міжмережевий екран, отримуючи той чи інший пакет даних, зчитує інформацію щодо вузла відправника/одержувача, визначає його тип та порт призначення. Далі порівнює одержані дані з записами списку доступу, на базі чого далі приймає рішення щодо пропуску чи відхилення пакету.

В означений вище спосіб функціонують класичні firewall. На відміну від них, NGFW мають значно ширший функціонал, та додатково дозволяють виконувати [5, 11]:

- антивірусне сканування даних на межі мережі;
- протидію ряду типів атак;
- глибокий аналіз пакетів (deep packet inspection);
- дослідження підозрілих об’єктів у sandbox-оточенні тощо.

У залежності від реалізації, міжмережеві екрани можуть бути:

- програмними;
- апаратними;
- апаратно-програмними.

При цьому, перші два зазначених типи реалізації зазвичай характерні для класичних екранів.

Вибір конкретного типу міжмережевого екрану та його формату визначається умовами, у яких його буде використовуватися та специфікою завдань, які він має вирішувати.

2.2 Критерії вибору міжмережевого екрану

У загальному випадку для того, щоб обрати найбільш ефективний варіант міжмережевого екрану для захисту мережевого периметру, необхідно мати відомості відносно [11]:

- інтенсивності трафіку у напрямку «зовнішня мережа – захищуване середовище» та навпаки;
- масштабу захищуваної мережі;
- наявності інших компонент протидії зловмисним впливам;
- специфіки мережевих сервісів та додатків, які використовуються вузлами мережі.

У випадку, якщо firewall для захисту периметру мережі планується використовувати в умовах високої інтенсивності трафіку, необхідно виключити з розгляду програмні варіанти побудови екранів, оскільки продуктивність такої системи залежить від:

- продуктивності апаратного базису системи;
- інтенсивності трафіку.

При цьому, якщо інтенсивність трафіку, з одного боку, є непрогнозованою а з іншого – має місце тенденція до зміни рівня інтенсивності за пульсуючим характером, це все свідчить про недоцільність використання апаратної реалізації firewall, як наслідок того, що:

- програмна реалізація firewall передбачає використання ПК як проксі – модулю для встановлення засобу, що реалізує функціонал міжмережевого екрану;
- у загальному випадку, ПК, у першу чергу, орієнтовано на виконання завдань, які я відмінні від аналізу трафіку у реальному часі.

Отже, у зазначених умовах більш доцільним є використання апаратних, або апаратно програмних засобів. У цьому випадку програмно-апаратні засоби мають перевагу, що зумовлюється легкістю конфігурування міжмережевого екрану відповідно до певних умов використання.

Таким чином, для подальшого використання необхідно розглянути засіб, що реалізує функції firewall, ураховуючи усі вищезазначені обставини.

Отже, у нашому випадку варто розглядати засіб, орієнтований на мережу невеликого масштабу, який при цьому, дозволяє однаково ефективно обробляти трафік різних рівнів інтенсивності.

Це можливо за умови, коли роль firewall відводиться програмно-апаратному засобу [5].

При цьому, для того, щоб обрати firewall в апаратно-програмній реалізації, попередньо виконаємо порівняльний аналіз ряду найбільш популярних лінійок рішень у даній галузі, як показано табл.2.1 [12-14].

Таблиця 2.1 – Порівняння найбільш розповсюджених апаратно-програмних firewall-засобів

Розробник	Cisco	Check Point	Keenetic
Гнучкість налаштування	+++	+++	+++
Інтуітивно-зрозумілий інтерфейс	+	++	+++
Поріг входження	+++	+++	++
Необхідність знання специфічної скриптової мови	+++	+++	+
Апріорне блокування доступу ззовні	++	++	+++

Виходячи з інформації, зазначеної у таблиці 2.1, можемо дійти висновку про те, що:

- з точки зору гнучкості налаштування, кожна з розглянутих лінійок є ефективною;
- поріг входження (досвідченість адміністратору) є найнижчим для засобів Keenetic;
- за показником інтуітивної зрозумілості інтерфейсу лідером також є Keenetic;
- засоби Keenetic не вимагають від адміністратора обізнаності зі специфічних мов конфігурування програмно-апаратних засобів;
- міжмережеві екрани Keenetic на базі роутерів даного бренду за замовчуванням блокують усі зовнішні звернення до внутрішньої мережі.

Таким чином, виходячи з розглянутих даних щодо найбільш поширених на сьогодні лінійок firewall, можемо бачити, що за сукупністю показників лідером є лінійка Keenetic.

Виходячи з цього, у нашому випадку, для прикладу пропонується використати міжмережевий екран на базі роутеру Keenetic Ultra.

Інакше кажучи, єдиний пристрій поєднує у собі функції маршрутизатора та firewall.

2.3 Налаштування Keenetic Ultra у якості міжмережевого екрану

2.3.1 Первинне налаштування Keenetic Ultra

Конфігурування Keenetic Ultra передбачає взаємодію з вбудованим ПЗ засобу, що має внутрішню назву інтернет-центр Keenetic [15].

Для цього необхідно, щоб ПК, на базі якого виконуватиметься взаємодія адміністратору з інтернет-центром, було налаштовано на автоматичне отримання IP-адреси.

Далі необхідно, скориставшись браузером, звернутися до адреси my.keenetic.net, або 192.168.1, як встановлено за замовчуванням (рис.2.1).

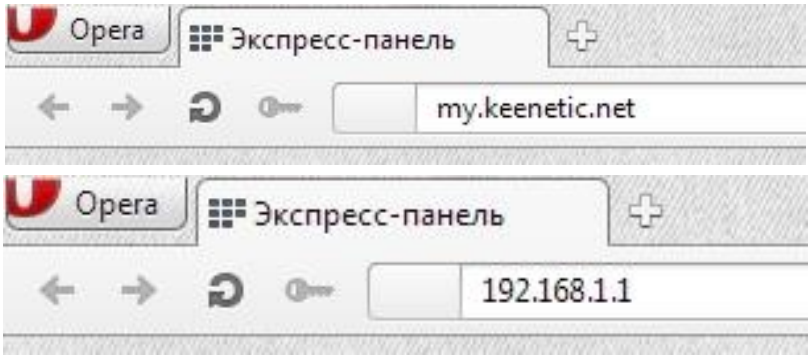


Рисунок 2.1 – Звернення до інтернет-центру Keenetic

Загальний вигляд KeeneticOS, починаючи з версії 2.12 є таким, як показано на рисунку 2.2.

При цьому, перший крок налаштування інтернет-центру управління засобом Keenetic, у сутності, полягає у встановленні паролю адміністроатора, для чого може бути використано меню Система → Користувачі, як зазначено рисунком 2.3.

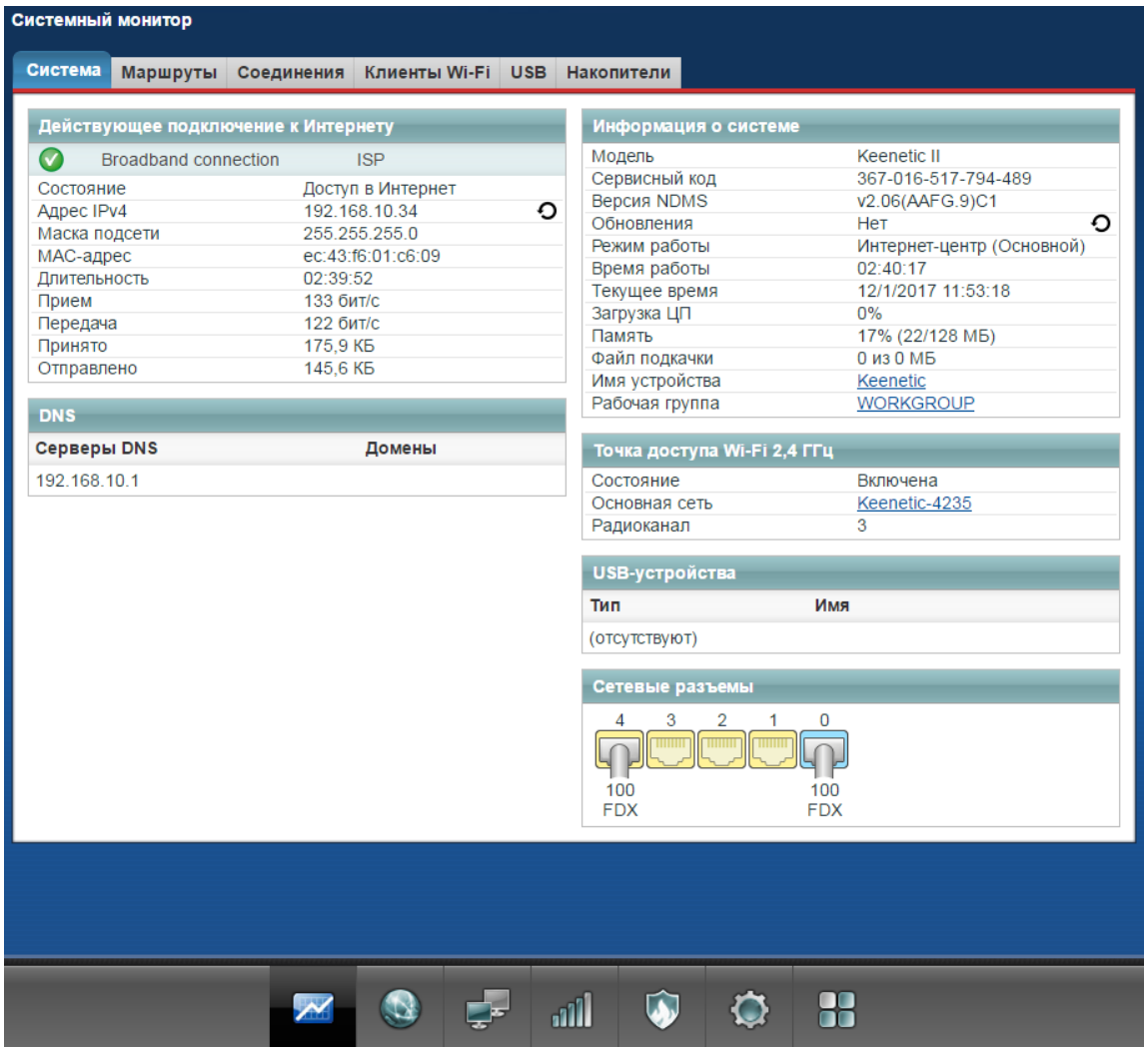


Рисунок 2.2 – Загальний вигляд вікна управління KeeneticOS

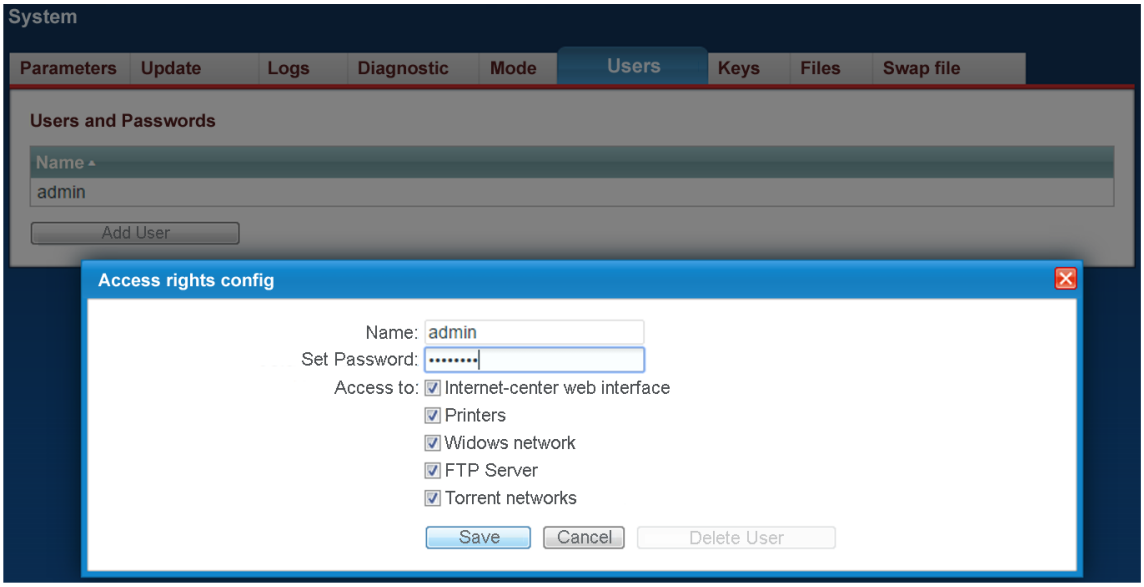


Рисунок 2.3 – Інтерфейс налаштування паролю адміністратора KeeneticOS

2.3.2 Налаштування режиму адміністрування засобу

У першу чергу, необхідно встановити необхідний з точки зору безпеки режим взаємодії з firewall, який, у загальному випадку, містить у собі такі налаштування, як [15]:

- встановлення та конфігурування режиму віддаленого використання;
- заборона доступу з зовнішньої мережі та/або з вузлів внутрішньої мережі.

При цьому, можливими режимами тут можуть бути:

- режим, у якому конфігурування дозволено у разі звернення лише з вузлів локальної мережі;
- режим, у якому конфігурування дозволено у разі звернення лише з деяких вузлівлокальної мережі;
- режим, що дозволяє доступ з будь-яких вузлів, у т.ч. зовнішніх;
- режим, у якому дозволено доступ певній множині вузлів (або одному вузлу) з довіреного переліку.

За замовчуванням доступ до веб-конфігуратору інтернет-центру ззовні є заблокованим, що забезпечує найвищий рівень захищеності як самого пристрою, так і мережі.

Разом з тим, доступ з мережі Інтернет до веб-конфігуратору виконується через браузер, для чого використовується біла IP-адреса WAN-інтерфейсу роутеру. Відомості про дану адресу може бути отримано з веб-конфігуратору на початковій сторінці "Системний монітор" у локації *Інтернет → Докладніше про з'єднання → IP-адреса.*

Водночас, у цьому випадку для того, щоб суттєво скоротити ймовірність зловмисних впливів на сам firewall, необхідно блокувати можливість доступу до веб-конфігуратору та дозволити виконання пінг-запитів для усіх зовнішніх вузлів, окрім довірених.

Для цього, у свою чергу, необхідно створити відповідне правило міжмережевого екрану, яке включає дозвіл на виконання ICMP-запитів да доступ з одного зовнішнього вузла. У нашому випадку - 93.94.95.96.

У цьому разі загальне правило створюється для інтерфейсу зовнішньої мережі. Це може бути PPPoE, PPTP, USB LTE тощо. Далі створюється правило, що дозволяє обмежений доступ ззовні (рис.2.4). Тут необхідно:

- вказати необхідну адресу (93.94.95.96) у полі «IP-адреса джерела»;
- у полі «Протокол» обрати варіант TCP/80 (HTTP).

На наступному кроці створюється аналогічне правило, яке встановлює дозвіл для протоколу ICMP (для можливості роботи утиліти ping) (рис.2.5).

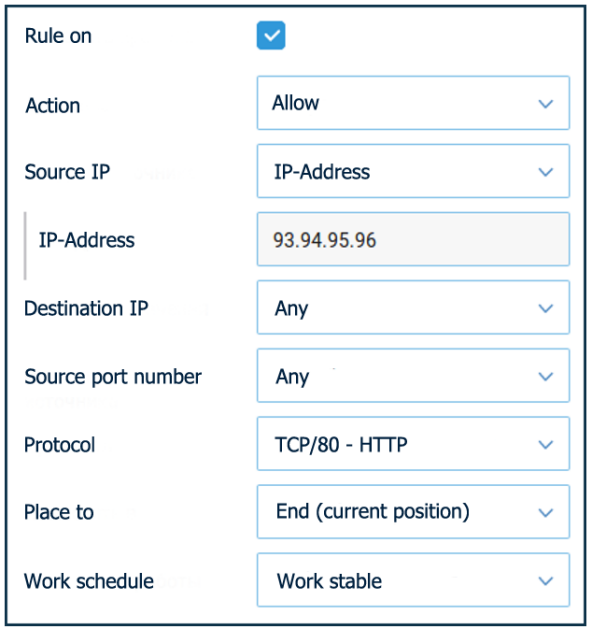


Рисунок 2.4 – Створення правила обмеженого зовнішнього доступу для конфігурування KeeneticOS

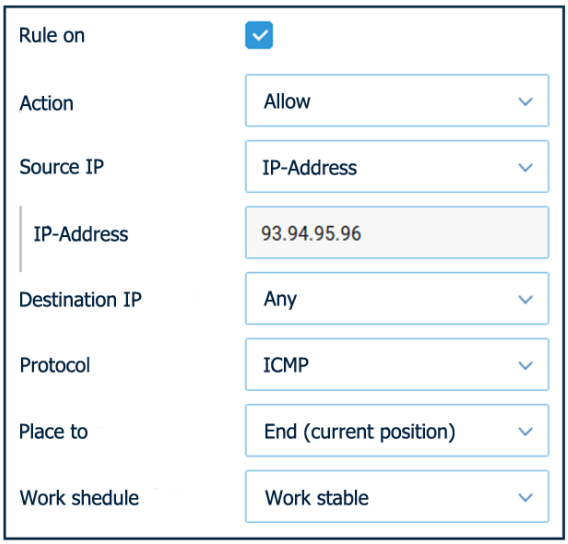


Рисунок 2.5 - Створення правила обмеженого зовнішнього доступу для ICMP

У результаті виконаних налаштувань, доступ до інтернет-центру Keenetic може бути забезпечено як для процедури пінгу (протокол ICMP), так і, безпосередньо, для віддаленого доступу до веб-конфигуратору (протокол HTTP), при цьому, виключно з зазначеної IP-адреси [15].

Далі виконаємо блокування доступу до веб-інтерфейсу міжмережевого екрану для усіх вузлів локальної мережі, окрім адміністраторського.

У першу чергу тут необхідно зазначити, що блокування слід виконувати окремо за адресами 192.168.1.1 та my.keenetic.net. Це, у свою чергу, може бути реалізовано з використанням заборонюючих правил firewall. Для цього їх необхідно створити і далі - застосувати на рівні интерфейсу локальної мережі. За замовчуванням це інтерфейс Домашня мережа.

Розглянемо приклад заборони доступу до веб-інтерфейсу роутера для вузла, що має IP-адресу 192.168.1.143. (рис.2.6).

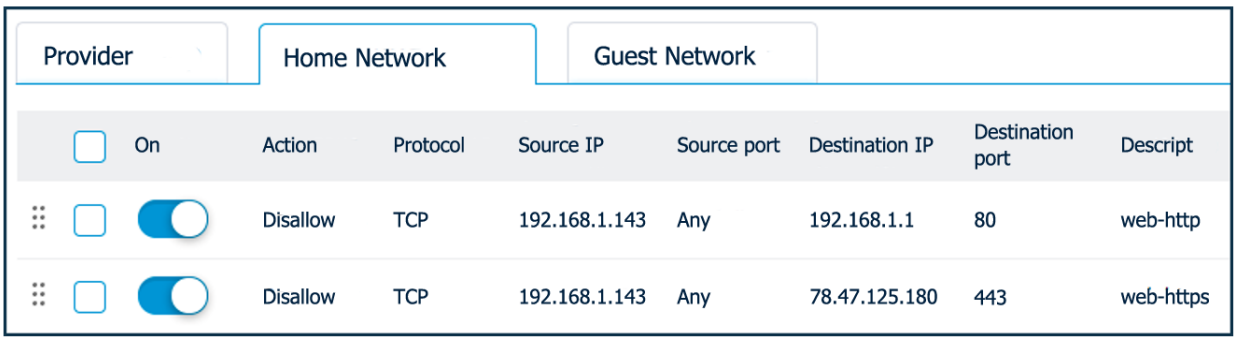


Рисунок 2.6 – Налаштування заборони доступу для вузла локальної мережі

У розглянутому прикладі створено два забороняючих правила.

При цьому, у ролі адреси джерела зазначено IP-адресу вузла з локальної мережі, якому необхідно заборонити доступ до веб-інтерфейсу роутера. Для будь-яких інших вузлів, виключаючи АРМ адміністратора, заборона доступу виконується аналогічним чином.

Разом з тим, необхідно звернути увагу на те, що у правилі для заборони доступу до адреси 192.168.1.1 порт призначення у налаштуваннях має бути зазначений як TCP/80.

Водночас у процесі формування правила для заборони доступу за адресою my.keenetic.net необхідно вказувати адресу призначення, як 78.47.125.180. Саме цю IP-адресу прив’язано до доменного імені my.keenetic.net (функціонує виключно усередині роутера), а також порт TCP/443, оскільки у ході звернення за доменним ім’ям здійснюється автоматичний редирект на протокол HTTPS [15].

Діла можемо виконувати безпосереднє конфігурування самого міжмережевого екрану.

2.4 Конфігурування міжмережевого екрану

До переліку завдань, які є типовими для міжмережевого екрану, можемо віднести [15]:

1. Блокування доступу з локальної мережі до певного сайту.
2. Встановлення доступу до Інтернету окремим вузлам локальної мережі.
3. Встановлення заборони доступу до Інтернету окремим вузлам локальної мережі.
4. Налаштування можливості доступу до окремих веб-сайтів для вузлів локальної мережі
5. Нолаштування доступу з локальної мережі назовні лише з обмеженим набором протоколів (сервіси, служби).

2.4.1 Блокування доступу з локальної мережі до певного сайту

Виконаємо налаштування заборони доступу вузлам внутрішньої мережі то того чи іншого сайту. При цьому, для прикладу візьмемо сайт wikipedia.org.

Тут слід взяти до уваги те, що під час налаштування правил firewall Keenetic не може бути використано доменні імена, а виключно IP-адреси.
Виходячи з цього, попередньо, до налаштування самих правил, у першу чергу необхідно виявити IP-адресу веб-сайту, або сайтів, які підлягають блокуванню. У цьому випадку слід брати до уваги, що один сайт может мати ряд різних IP-адрес, що зазвичай є характерним для великих ресурсів - google.com, fb.com тощо.

При цьому, найпростіший спосіб отримати відомості щодо IP-адреси сайту полягає у використанні команди nslookup <ім’я веб-сайту>, що виконується з командного рядку. У нашому випадку це (рис.2.7):

```
nslookup ru.wikipedia.org
```

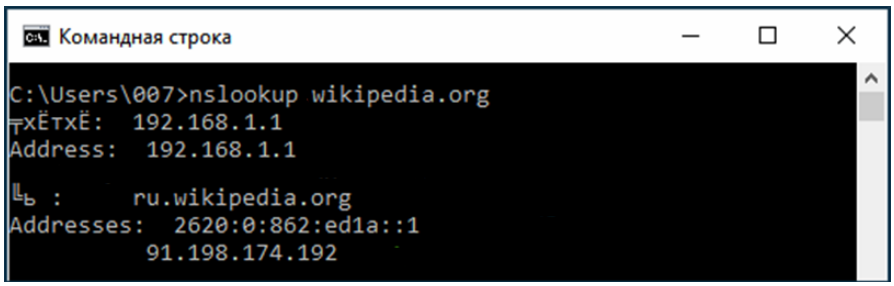


Рисунок 2.7 – результат запиту IP-адреси сайту

У свою чергу, альтернативний шлях знаходження IP-адреси сайту — скористуватися одним зі спеціалізованих онлайн-сервісів (наприклад, whois.com).

Далі, маючи необхідні IP-адреси сайтів, необхідно створити відповідні правила для міжмережевого екрану. При цьому, слід пам'ятати, що деякі сайди можуть функціонувати з використанням як протоколу HTTP, так і HTTPS одночасно, що вимагає окремих налаштувань для кожного з них.

Розглянемо приклад, коли сайт асоціюється лише з однією IP-адресою. У першу чергу для інтерфейсу «Домашня мережа» створюється два забороняючих правила, що блокують трафік за протоколом HTTP та HTTPS відповідно. Тут необхідно зазначити IP-адресу призначення (IP-адресу сайту, до якого буде заборонено доступ) та тип протоколу (HTTP або HTTPS), як показано рис. 2.8.

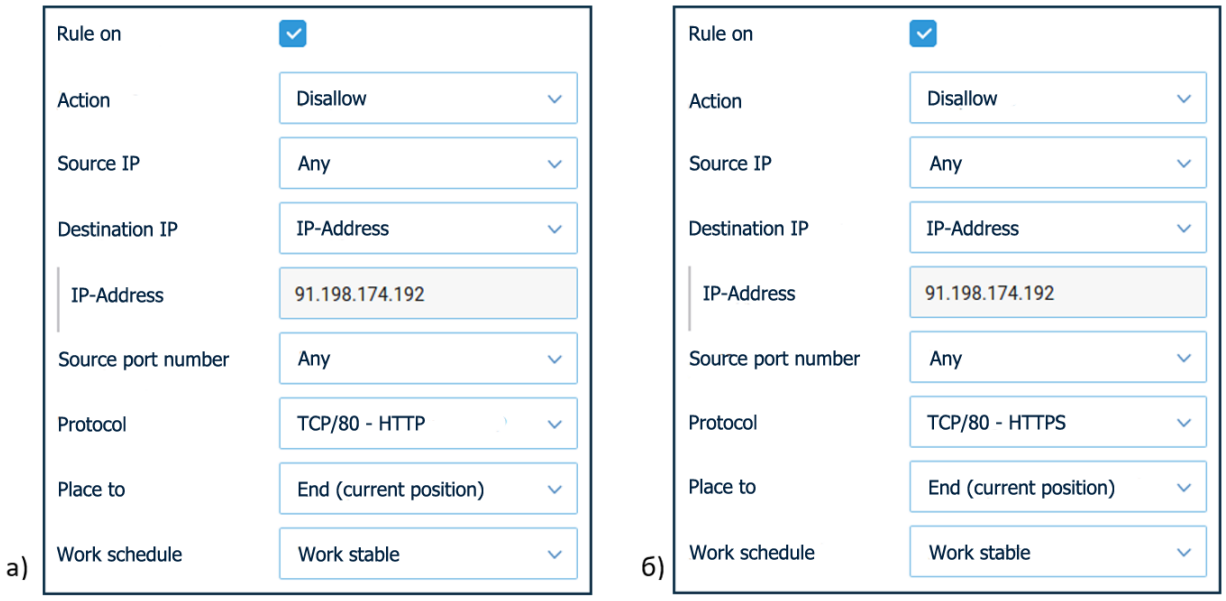


Рисунок 2.8 – Заборона доступу до сайту за протоколом HTTP а) та HTTPS б)

У зазначений вище спосіб може бути блоковано доступ до необхідного переліку сайтів.

2.4.2 Налаштування дозволу на доступ до зовнішньої мережі внутрішнім вузлам

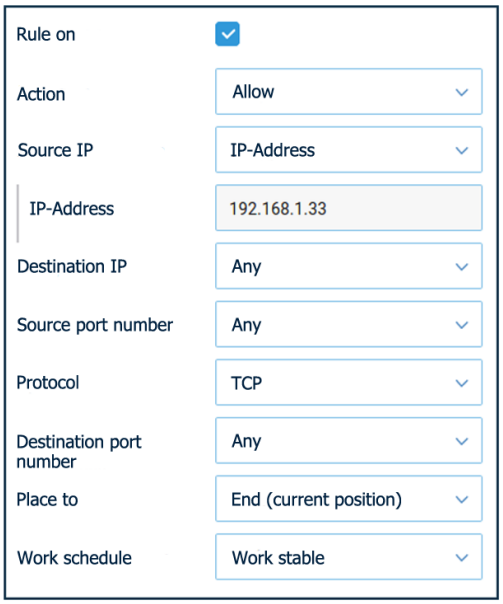
У цілому, дане завдання може розглядатися у двох аспектах, а саме [15]:

- забезпечення доступу назовні окремим вузлам;
- блокування доступу назовні окремим вузлам.

Для рішення першого завдання, як і випадку блокування доступу до сайту, також створюється два правила для інтерфейсу «Домашня мережа». При цьому, перше правило є дозволяючим. У ході його налаштування зазначається IP-адреса джерела, тобто, вузла, доступ у зовнішню мережу для якого має бути дозволено, та встановлюється тип протоколу – TCP (рис.2.9 а).

Далі створюється забороняюче правило, де у ролі IP-адреси джерела вказується підмережа (наприклад, 192.168.1.0 з маскою 255.255.255.0). Тип протоколу також вказується TCP (рис.2.9 б).

а)



б)

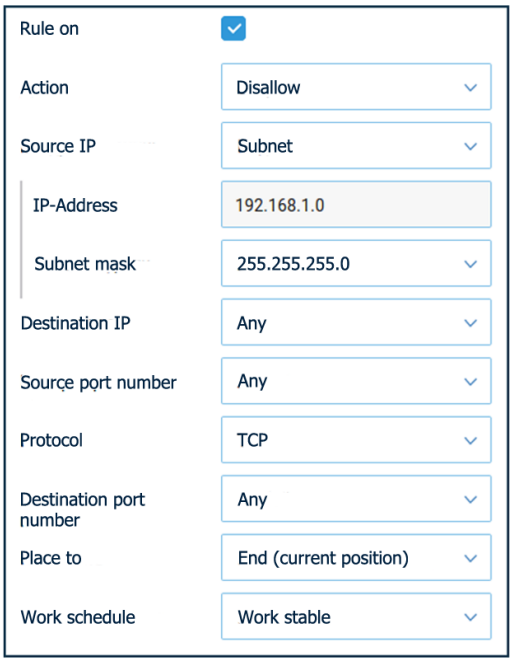


Рисунок 2.9 – Налаштування дозволяючого а) та забороняючого правил б) доступу до зовнішньої мережі

Після цього, як створено забороняюче правило, за потреби можемо побудувати дозволяючі правила для необхідної кількості вузлів внутрішньої мережі.

У свою чергу, блокування доступу у зовнішнє мережеве середовище для окремих внутрішніх вузлів реалізується простіше, ніж встановлення дозволу окремим вузлам.

У даному випадку створюється єдине забороняюче правило для інтерфейсу «Домашня мережа» (рис.2.10).

Протягом даної процедури вказується IP-адреса вузла, доступ назовні для якого буде заборонено. Тип протоколу також обирається TCP.

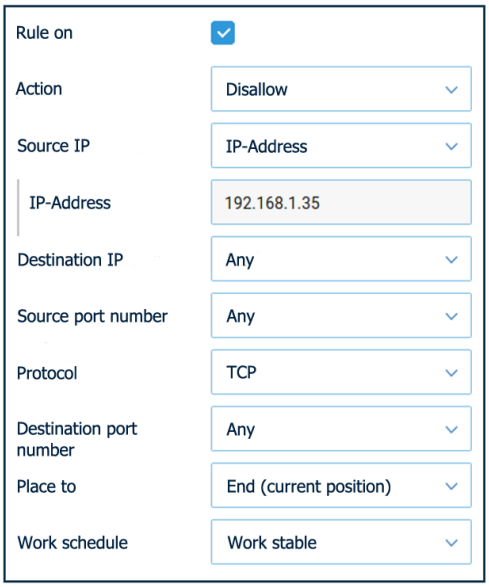


Рисунок 2.10 – Створення забороняючого правила доступу вузла у зовнішнє мережеве оточення

Отже, тут є очевидним, що кожному вузлу, якому блокується вихід назовні, відповідатиме одне забороняюче правило.

Разом з тим, реалізація багатьох специфічних бізнес-процесів у межах підприємств різних масштабів потребує хоча б частково обмеженого доступу користувачів до зовнішніх вузлів (розподілена БД, зовнішній веб-сервер тощо).

На цей випадок користувачеві може бути надано доступ до певного переліку зовнішніх вузлів.

2.4.3 Налаштування дозволу звернення до окремих вузлів Інтернет-середовища для локальних користувачів

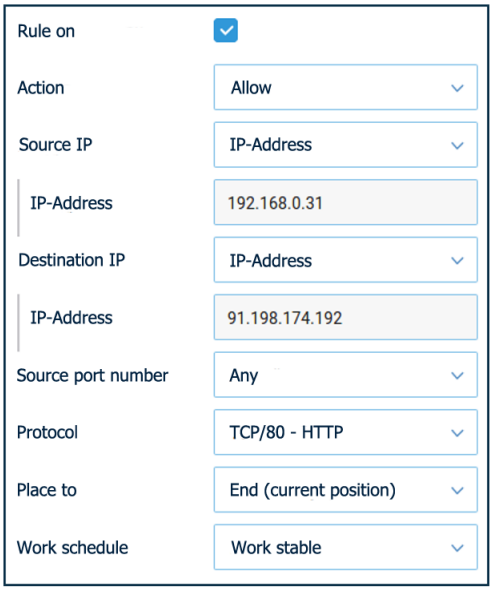
Розглянемо завдання, обернене тому, яке було вирішено у пункті 2.4.1 – а саме, встановимо дозвіл для деякого вузла локальної мережі на доступ до деякого сайту. Наприклад, аналогічно протилежному завданню, wikipedia.org.

Разом з тим, можливість доступу до інших сайтів для вузла має бути блоковано.

Раніше адресу цільового сайту було визначено (91.198.174.192), відтак можемо виконувати безпосереднє налаштування міжмережевого екрану.

При цьому, у зазначеному випадку для інтерфейсу «Домашня мережа» має бути створено три правила. У першу чергу, створюються дозволяючі правила, під час налаштування яких зазначається IP-адреса вузла, для якого встановлюється доступ (IP-адреса джерела, наприклад – 192.168.0.31), та IP-адреса призначення - тобто, адреси веб-сайту, до якого надається доступ. Окремі правила створюються для протоколів HTTP та HTTPS (рис.2.11).

а)



б)

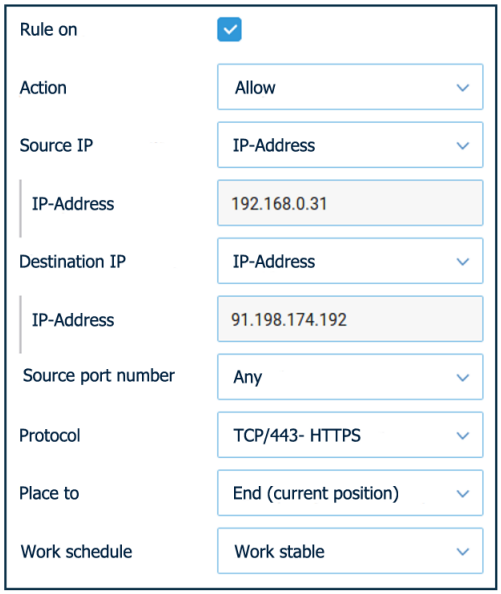


Рисунок 2.11 – Налаштування дозволяючих правил для протоколів HTTP а) та HTTPS б)

Після того, як дозволяючі правила налаштовано, необхідно створити забороняюче правило, де вказується IP-адреса вузла, для якого доступ блокується, та встановлюється TCP у якості протоколу, що необхідно для блокування Інтернету (рис.2.12).

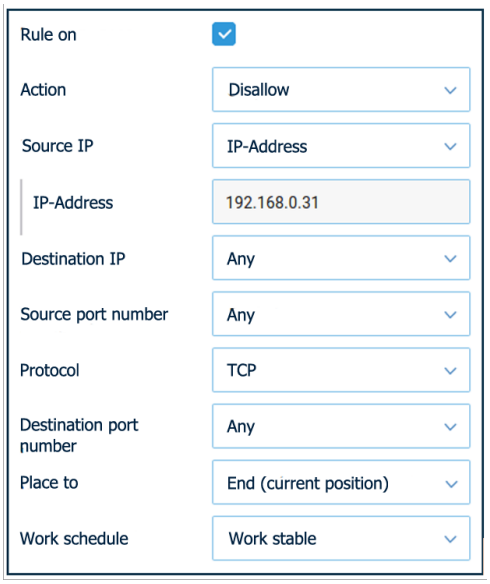


Рисунок 2.12 – Забороняюче правило для на доступ назовні

2.4.4 Надання доступу до Інтернет-мережі на базі обмеженого переліку протоколів

Нерідко з міркувань безпеки, або для оптимізації процесів внутрішньої мережі, в умовах, коли локальні вузли мають доступ до об’єктів Інтернет-середовища, необхідно обмежити доступний для використання сервісами та службами перелік протоколів.

Припустимо, що за внутрішнім регламентом безпеки підприємства, доступ до Інтернету вузли локальної мережі можуть здійснювати на базі обмеженого переліку протоколів, наприклад - FTP, HTTP, HTTPS, SMTP, IMAP, POP3 та DNS. При цьому, інший трафік має бути блоковано.

Щоб вирішити означене завдання, попередньо слід створити правила создать правила для інтерфейсу локальної мережі «Домашня мережа».

У даному випадку першочергово створюємо правила дозволяючого типу, де у полі «IP-адреса джерела» та «IP-адреса призначення» встановлюємо значення «Будь-який», а у полі «Протокол» зі списку обираємо потрібний тип (сервісу або служби).

Далі слід створити два забороняючих правила, де також у полі «IP-адреса джерела» та «IP-адреса призначення» встановлюємо значення «Будь-який». Відповідно, у полі «Протокол» обираємо значення TCP та UDP для того, щоб блокувати таким чином доступ до Інтернету.

У підсумку, для нашого випадку маємо перелік правил міжмережевого екрану, як показано рисунком 2.13.

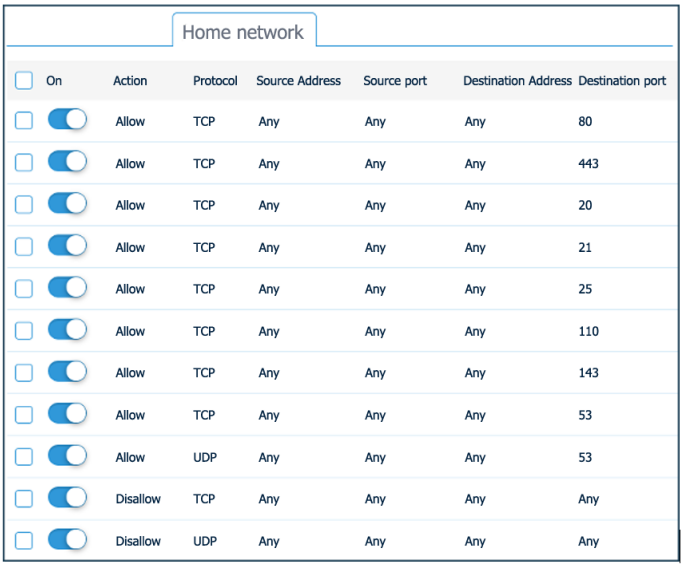


Рисунок 2.13 – Підсумковий перелік правил міжмережевого екрану

3. ЗАХИСТ ВНУТРІШНЬОГО СЕРЕДОВИЩА НА БАЗІ ЗАСОБІВ ВИЯВЛЕННЯ ТА ПРОТИДІЇ ЗЛОВМИСНИМ ВТРУЧАННЯМ

3.1 Обгрунтування необхідності захисту внутрішнього простору корпоративної мережі

У випадку, коли однією з компонент захисту мережі є класичний міжмережевий екран, його функції, за великим рахунком, зводяться до фільтрації такетів у відповідності до того, якому переліку – дозволеному чи забороненому – належать їх джерела [5, 11].

У цілому, навіть у найпростішому випадку (коли функціонал зводиться виключно до фільтрації пакетів) брендмауер сприяє збільшенню захищеності мережі, блокуючи можливість проникнення певної кількості зловмисних модулів усередину.

Разом з тим, для повноцінного захисту мережевої інфраструктури а також даних, що можуть зберігатися у внутрішніх файлосховищах, цього недостатньо, так як:

- існує висока ймовірність проникнення умережу як зловмисних модулів, так і інфікованих файлів, які надходять при цьому з довірених вузлів;

- брендмауер як класичного типу, так і NGFW-архітектури, не виконує моніторинг стану внутрішньої мережі;

- зловмисні модуль можуть потрапляти до мережі також з внутрішніх джерел (наприклад, зі знімних пристроїв) у наслідок некваліфікованих чи, навпаки, цілеспрямованих дій користувачів мережі;

- велика кількість цілеспрямованих кібератак, зокрема, на базі механізмів Code Red, Nimda, SQL Slammer, MSBlaster та подібні, мають засоби обходження міжмережевих екранів;

- протидія атакам, що реалізуються на базі довірених протоколів (наприклад, HTTP/ HTTPS), або мережевого трафіку систем Microsoft, виключно на рівні брендмауеру, є неефективною, оскільки блокування зазначених протоколів веде до аварійного функціонування мережевих служб та додатків.

З іншого боку, традиційні засоби захисту мережевих вузлів від зловмисних впливів усередині периметру, наприклад, антивірусні програми у ряді випадків демонструють низьку ефективність, так як:

- постійно створюються нові віруси, виявлення яких не гарантується до оновлення сигнатурних баз;
- для деяких складних атак (у першу чергу - APT) створюються зловмисні модулі з унікальною архітектурою, що робить недоцільним використовувати проти них як сигнатурні, так і евристичні механізми виявлення [8].

За таких умов необхідно використовувати засоби мережевої безпеки, що:

- орієнтовані на захист внутрішнього мережевого середовища;
- базуються на використанні механізмів як сигнатурних і евристичних механізмів, так і механізмів, що можуть виявляти факти зловмисного втручання за непрямими ознаками.

Такими засобами безпеки є NIDS-системи (Network Intrusion Detection System). На сьогодні існує велика кількість NIDS-системи, що належать як комерційному сектору, так і відносяться до засобів з відкритим кодом.

У нашому випадку, за умовами завдання, захист внутрішньої мережі необхідно налагодити на базі засобу, що відповідає наступним вимогам:

- безкоштовність;
- можливість розгортання на базі ПК під управлінням MS Windows.

Зазначеним вимогам повністю відповідає NIDS Snort [16].

3.2 Загальні відомості про NIDS Snort

З самого початку NIDS Snort було розповсюджувано на умовах ліцензії GNU General Public License (GPL) у середовищі Unix. Згодом було перенесено Snort (починаючи з версії 1.7) на платформу Win32. На сьогодні Snort може використовуватися майже на усіх платформах Windows, починаючи з XP Home Edition

Актуальна версія Snort забезпечує аналіз мережевого трафіку в реальному часі та реєстрацію IP-трафіку зі швидкостями Fast Ethernet та Gigabit Ethernet.

NIDS перевіряє кожен пакет, що проходить через інтерфейс, у пошуках відомих послідовностей в інформаційному наповненні, де зазвичай прихований шкідливий програмний код. За допомогою Snort можна

виконувати операції пошуку та зіставлення над кожним пакетом, що проходить через мережу організації, та виявляти велику кількість типів атак та нелегітимного трафіку у реальному часі.

До беззаперечних вимог засобу можна віднести його невибагливість до обчислювальних ресурсів.

Так, учасниками проекту Snort зазначається [16, 17], що успішне обслуговування мережі на кілька сотень користувачів з 2-3 WAN каналами може бути налаштовано на платформі FreeBSD із процесорами порядку 1 ГГц та оперативною пам'яттю об'ємом 1 Гбайт. Це э можливим завдяки ефективності вихідного тексту Snort.

3.3 Варіанти розгортання NIDS Snort

У найпростішому випадку Snort може бути використано для захисту одного вузла. При цьому, при розгортанні на ноутбукові засіб являтиме собою переносний сенсор – програма відіграє роль NIDS для будь-якого порту, до якого підключається ноутбук (рис. 3.1).

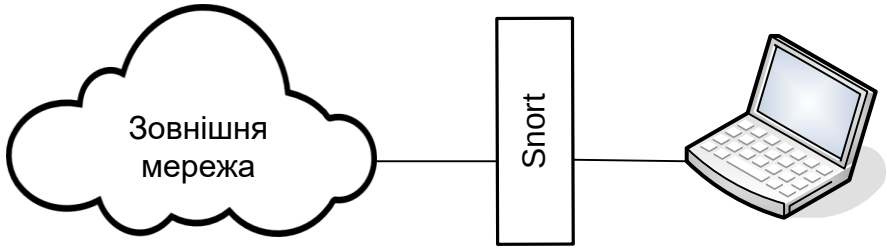


Рисунок 1 – Найпростіший спосіб використання NIDS Snort

Разом з тим, якщо ПК буде оснащено, принаймні, двома мережевими адаптерами, один з яких підключений до контрольованої мережі, а інший – має вихід назовні, це дозволить налаштувати захист мережевого простору.

На випадок мереж невеликого масштабу Snort може бути розгорнуто на сервері початкового рівня.

У будь-якому випадку, одне з головних питань, яке має бути вирішено перед розгортанням засобу, стосується конкретного місця його включення у мережі. Розглянемо деякі можливі варіанти розміщення NIDS Snort у межех контрольованої мережі.

3.3.1 Розміщення Snort на рівні периметру мережі

Даний варіант розміщення Snort передбачає його включення перед міжмережевим екраном (рис.3.2).

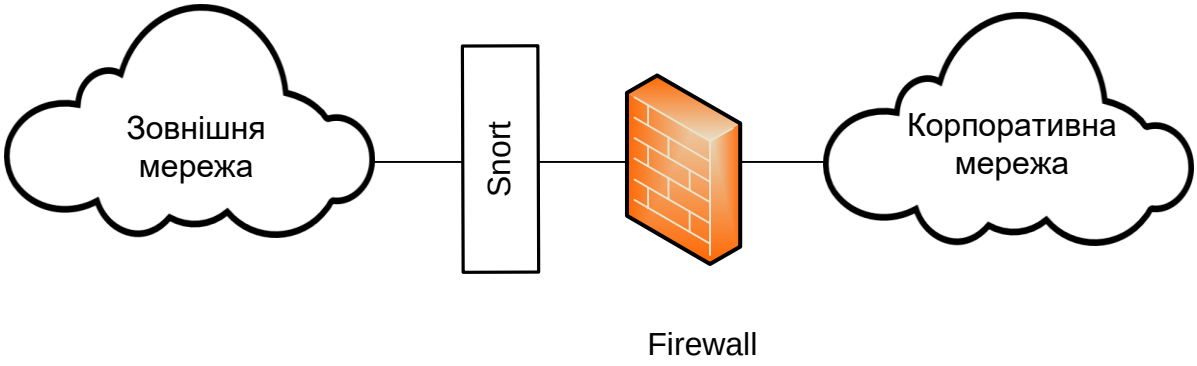


Рисунок 3.2 – Варіант розміщення Snort на рівні периметру мережі

Безумовною перевагою даного способу розміщення є виявлення великої кількості випадків зловмисних втручань.

Брендмауер, на цей випадок, буде виконувати додаткову функцію захисту.

Недоліком способу, при цьому, є велика кількість помилкових спрацювань а відтак – некоректних сповіщень про небезпеку.

Водночас, блокування більшої частини НСД і так може бути забезпечено самим міжмережевим екраном. А набагато важливішим тут є виявлення та блокування зловмисного ПЗ.

Відтак, набагато ефективнішим використання Snort для захисту мережі є розміщення його у внутрішньому просторі, після міжмережевого екрану (рис.3.2 а).

3.3.2 Включення Snort усередині периметру мережі

У загальному випадку, побудова контуру мережевого захисту з включенням NIDS у внутрішній простір мережі як одного з бар’єрних пристроїв, є суттєво раціональнішим ніж варіант його розміщення на рівні периметру мережі. Разом з тим, на цей випадок також має ураховуватися специфіка мережі, у першу чергу це стосується VPN [17].

Так, якщо за регламентом робити мережі користувачі можуть під’єднуватися до неї через з'єднання VPN, є сенс розмістити NIDS ще далі за

брандмауером, наприклад, за VPN-сервером, де пакети розшифровуються на виході з VPN-тунелю (рис. 3.2 б).

В іншому випадку NIDS не зможе протистояти шкідливим програмам, вбудованим у трафік VPN, оскільки пакети, які мають аналізуватися, будуть зашифровані. Те саме стосується також шифрованого SMTP-трафіку, шифрованих zip-файлів, які є вкладеннями у повідомлення e-mail і у цілому зашифрованих даних будь-яких інших типів.

Таким чином, як для Snort, так і для будь-яких NIDS-систем, ідеалізованим варіантом розміщення можна вважати:

- розміщення досить далеко за будь-якими компонентами, що шифрують трафік;

- розміщення досить близько до периметру мережі для можливості аналізу трафіку у максимальній кількості сегментів та підмереж.

Отже, уніфікованим та компромісним варіантом розміщення NIDS Snort у мережевій структурі може вважатися такий, за якого NIDS-модуль розміщується (у напрямку від внутрішньої мережі до зовнішнього середовища) після внутрішнього комутаційного пристрою, трафік якого спрямовується до зовнішньої мережі, за потреби проходячи актуальну низку програмно-технічних модулів [17].

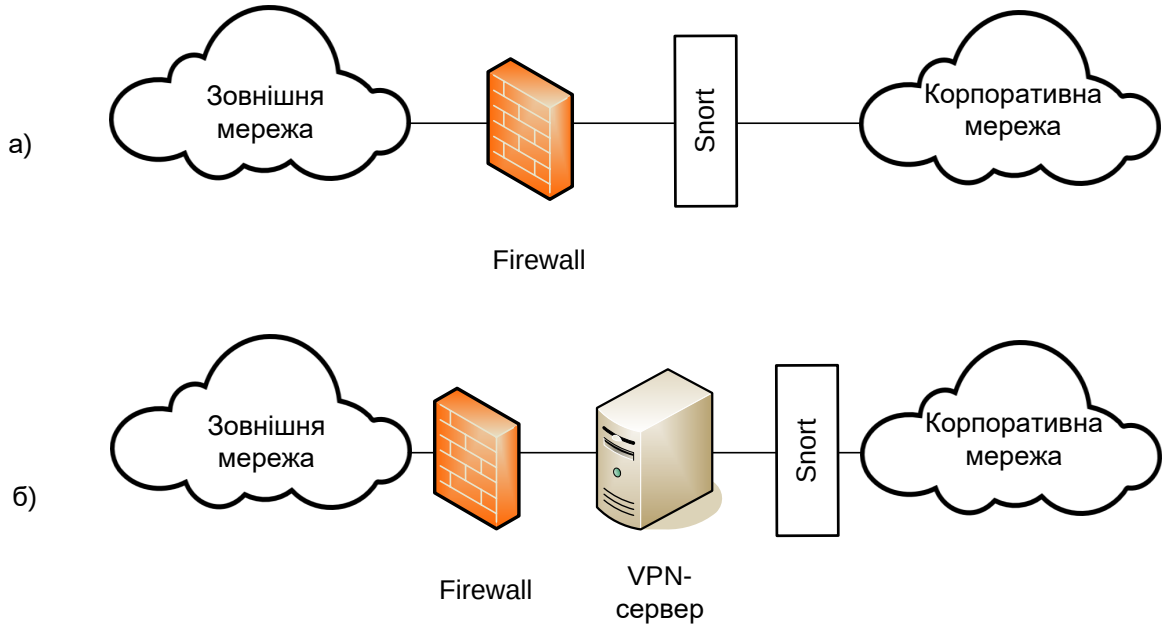


Рисунок 3.2 – Загальний принцип ефективного розміщення NIDS-модулів а), розміщення NIDS-модулів у мережах з VPN- з’єднаннями

Далі виконаємо огляд усього необхідного переліку дій для встановлення та налаштування NIDS Snort для захисту внутрішнього мережевого простору від зловмисних впливів.

3.4 Встановлення та конфігурування Snort

3.4.1 Інсталяція WinPcap

У сутності, Snort може розглядатися як мережевий аналізатор, який функціонує режимі прийому всіх пакетів (promiscuous-mode).

Виходячи з цього факту, стає зрозумілим, що засіб потребує підтримка на рівні драйверів.

Цю підтримку забезпечує WinPcap.

У свою чергу, WinPcap, було створено за рахунок переносу до середовища Windows широко поширеного серед користувачів Unix драйверу перехоплення пакетів libpcap.

При цьому, до складу WinPcap входять:

- фільтр пакетів на рівні ядра;
- низькорівнева DLL (packet.dll);
- високорівнева системно-незалежна бібліотека (wpcap.dll, з урахуванням libpcap 0.6.2).

Слід зазначити, що пакет WinPcap може бути завантажено з джерела [18]. Драйвер є сумісним з майже усими версіями MS Windows.

WinPcap також підтримує відкритий аналізатор пакетів Ethereal, який можна отримати з офіційного джерела.

Так чи інакше, процедура завантаження з мережі інсталяційного файлу WinPcap з наступною його інсталяцією зводиться до внесення інформації у кілька діалогових вікон та не викликає труднощів.

На наступному етапі виконаємо процедуру встановлення Snort.

Найновішу версію даного засобу може бути отримано з web-вузла CodeCraft Consultants або з сайту проекту Snort [16].

У ході запуску процедури встановлення, у першому діалоговому вікні необхідно вибрати режим налаштування бази даних для збереження результатів.

При цьому, якщо використовується MySQL або ODBC-сумісна база даних, можна погодитися на режим, що вибирається за замовчанням.

Разом з тим, якщо необхідно зберігати протоколи в базі даних Microsoft SQL Server, або Oracle, тоді потрібно вибрати відповідний режим, а також

переконатися, що на ПК, який планується використовувати як як сервер NIDS, є необхідна клієнтська програма. У нашому випадку було обрано режим за замовчуванням (рис. 3.3) [19].

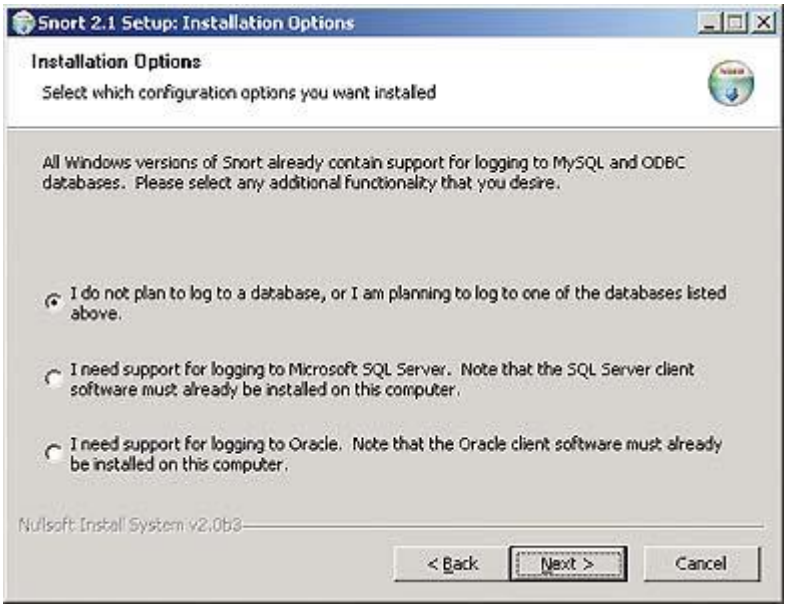


Рисунок 3.3 – Вибір БД у процесі інсталювання NIDS за замовчуванням

На наступному кроці слід визначити компоненти Snort, які необхідно інсталювати. У даному випадку стандартний набір (рис.3.4) може вважатися цілком прийнятним.

Виходячи з цього, доцільно далі його прийняти, після чого перейти до наступного діалогового вікна.

Тут, у межах діалогового вікна Choose Install Location, необхідно зазначити директорію, у межах якої на наступних етапах інсталяції буде розгорнено Snort.

Для завершення процедури тут достатньо задати ім'я директорії, після чого завершити процес інсталювання.

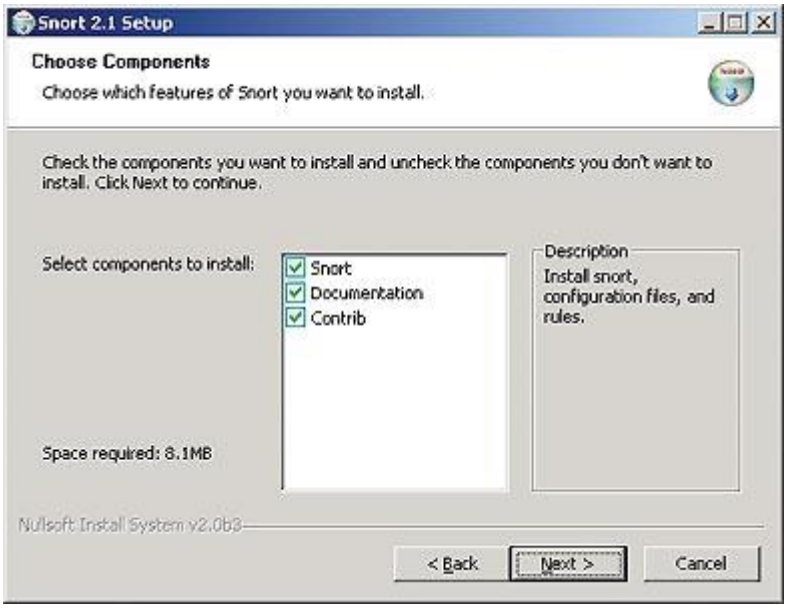


Рисунок 3.4 – Вибір компонент Snort, які буде інстальовано

3.4.2 Етап тестування установки Snort

Далі, завершивши процес установки Snort, необхідно протестувати даний NIDS-засіб. У цьому випадку, за замовчуванням, виконуваному файлові Snort необхідно повідомити дві локації, а саме [17, 19]:

- локацію, куди буде виконуватися запис журналів;
- локацію файлу конфігурації (snort.conf).

Зазначені дані надає користувач у процесі запуску NIDS Snort з командного рядку з використанням ключів -l та -c відповідно.

Наприклад, команда:

```
snort -l F:snortlog -c F:snortetcsnort.conf -A console
```

вказує програмі, що журнали слід записувати до каталогу F: snortlog, а snort.conf знаходиться у каталозі F: snortetc. Тут ключ -A задає спосіб передачі генерованих програмою попереджень.

У розглянутомі прикладі попередження виводяться на екран консолі, тобто, на цей випадок адміністратор може переконатися, що Snort працює коректно.

Тут слід звернути увагу на те, шо команда вище займає єдин й рядок. Те саме стосується й інших команд стосовно Snort, які включають у себе кілька операндів. Окрім зазначеного додамо, що багато ключів командного

рядка Snort є чутливими до регістру символів, тому вводити команди слід саме так, як вони надруковані у супровідній документації.

Разом з тим, якщо ПК, де інсталюється Snort, має кілька мережних інтерфейсів, тоді, за замовчуванням, Snort прослуховує перший виявлений інтерфейс.

Водночас, якщо порядок інтерфейсів на ПК невідомий, можна виконати команду Snort з одним ключем -W. Snort видає список імен та номерів мережевих інтерфейсів у порядку, в якому їх виявляє програма. У свою чергу, для того, щоб змусити Snort використовувати певний інтерфейс мережі, необхідно ввести ключ -i з номером інтерфейсу при запуску Snort.

Далі, після запуску NIDS Snort, можна перевірити його чутливість на сторонніх впливів. Для цього можемо спрямувати до мережі, яку може прослуховувати даний засіб, спеціально підготовлений трафік.

У даному разі одним з найпростіших способів викликати попередження про небезпеку є звернення до командного інтерпретатора (cmd.exe) на віддаленому ПК у рамках запиту HTTP URL. Це імітує типовий прийом зловмисних засобів worm Code Red і Nimda). Пр цьому, для імітації фази нападу слід звернутися до будь-якого URL і додати символи /cmd.exe у кінці запиту [19].

Наприклад, у відповідь на звернення до http://www.a-website-that-I-can-trust.com/cmd.exe, Snort повинен вивести попередження у командному вікні. Ці повідомлення далі записуються в журнал F:snortlog.

При цьому, цільові Web-вузли для тестування NIDS необхідно обирати з огляду на те, що з технічної точки зору більшість адміністраторів Web-вузлів розглядатимуть подібні дії як спробу злому. У свою чергу, така спроба не призведе до успіху (якщо тільки у конфігурації сервера не допущено серйозних помилок). Відтак, має сенс проводити тестування або лише власного серверу або стороннього довіреного серверу, адміністратори якого мають відомості про проведення випробувань.

На той випадок, коли тестування зазначеними вище способами неможливо реалізувати, може бути використано альтернативний спосіб перевірки Snort. Даний спосіб передбачає надсилання через мережу до серверу, або будь-якого іншого вузла з активною програмою Snort, максимально великого echo-пакету. У цьому разі, наприклад, можемо застосувати команду Ping, а саме:

```
ping -l 32767 ip_address
```

Тут ip_address – це IP-адреса цільового серверу або іншого Snort-вузла.

Зазначена команда виконує надсилання пакету довжиною - 32 Кбайт, що для команди Ping не є частим випадком. При цьому, Snort має виявити даний пакет.

На той випадок, якщо попередження від NIDS було отримано, далі можемо налаштовувати Snort для наших умов. У будь-якому іншому випадку необхідно повернутися до процедури інсталювання засобу та перевірити, чи не було допущено яких небудь помилок.

3.4.3 Виконання конфігурації Snort

Ключові відомості щодо конфігурації Snort зберігаються у файлі snort.conf, який за умовчанням розміщується у директорії %systemdrive%snortetc. При цьому, даний файл може як залишатися у зазначеній локації, так і може бути переміщено до іншої. Для цього необхідно вказати Snort відповідний шлях у командному рядку [17, 19].

Далі для того, щоб мати змогу відрізнити вхідний трафік від вихідного, необхідно повідомити Snort вузли та IP-адреси мережі підприємства. У свою чергу, для введення означених даних, у файлі snort.conf має бути задано змінну HOME_NET.

Також, слід знайти рядок `var HOME_NET any,` та замінити його діапазоном IP-адрес.

Окрім цього, можемо встановити один діапазон, наприклад, var HOME_NET 192.168.0.1/24, чи кілька діапазонів.

При цьому, внутрішній синтаксис вимагає при встановленні кількох діапазонів, вписувати необхідний перелік діапазонів у квадратні дужки, відокремлюючи при цьому кожен діапазон комами. Вносити пробіли між діапазонами IP-адрес є неприпустимим.

Так, якщо для прикладу візьмемо рядок, наприклад:

```
var HOME_NET [10.0.1.0/24,10.0.2.0/24,10.0.3.0/24],
```

це буде вказувати Snort на те, що підмережі 10.0.1.0/24, 10.0.2.0/24 та 10.0.3.0/24 належать мережі підприємства. За замовчуванням, Snort сприймає решту адрес, як зовнішні [19].

Також можна у явному вигляді вказати мережі, які слід вважати зовнішніми, поставивши змінну EXTERNAL_NET. Для цього у файлі snort.config потрібно наступний рядок:

```
var EXTERNAL_NET any,
```

замінити IP-адресою мережі, яку Snort має сприймати як зовнішню.

Водночас, для більшості випадків конфігурування, змінній EXTERNAL_NET доцільно встановити значенням any.

Далі, у ході наступних налаштувань, необхідно вказати типи серверів, які знаходяться у межах контрольованої мережі, та зазначити їхнє місцезнаходження. Свою чергу, інформація про сервери вноситься до змінних DNS_SERVERS, SMTP_SERVERS, HTTP_SERVERS, SQL_SERVERS та TELNET_SERVERS.

Зазначені змінні, при цьому, вносяться до файлу snort.conf. Тут спосіб присвоєння значень переліченим змінним залежить від масштабу самої контрольованої мережі. За замовчуванням формат запису є наступним, наприклад:

```
var DNS_SERVERS 194.111.8.1
```

З іншого боку, для мережі невеликого масштабу, що маємо у нашому випадку, конфігураційні змінні серверів доцільно подати у наступному вигляді [17]:

```
var DNS_SERVERS $HOME_NET
var SMTP_SERVERS $HOME_NET
var HTTP_SERVERS $HOME_NET
var SQL_SERVERS $HOME_NET
var TELNET_SERVERS $HOME_NET
var SNMP_SERVERS $HOME_NET
```

Як бачимо, тут усі внесені серверні змінні мають значення $HOME_NET. Такий формат запису вказує на те, що NIDS Snort далі буде виконувати контроль усіх типів атак на усі зазначені типи систем у діапазоні HOME_NET. Оскільки для невеликої мережі незначна кількість помилкових

попереджень NIDS-системи є допустимою, відтак дана конфігурація вважається цілком прийнятною.

З іншого боку, моніторинг інтенсивного трафіку, властивий мережам великих масштабів, потребує більш тонкого налаштування Snort для перевірки частини сигнатур для певних вузлів.

Наприклад, не має сенсу налаштовувати захист Web-серверу, що працює лише з Microsoft IIS, від атак з переповненням буфера SQL.

У свою чергу, для того, щоб визначити особливий клас вузлів, якраз і необхідно замість $HOME_NET вказувати діапазон IP-адрес (або єдину адресу) цільових серверів.

Далі можемо підвищити точність налаштування засобу за рахунок рорсткого призначення портів, які використовуються серверами для конкретних програм [19].

У рамках цього, встановимо для Web-серверу порт 8080 (HTTP/ HTTPS-трафік) замість порту 80, який встановлбється за замовчуванням. Даний порт зазвичай використовується для Web-серверів і браузерів.

Далі налаштуємо Snort на моніторинг порту 8080, для чого у файлі snort.conf знайдемо змінну HTTP_PORTS та виконаємо для неї відповідний запис:

```
var HTTP_PORTS 8080
```

На той випадок, коли у структуру мережі буде додано інші складові (наприклад, DNS-сервер), налаштування їх моніторингу виконуватиметься аналогічним чином.

3.4.4 Встановлення правил Snort

Однією з ключових компонент як Snort, так і NIDS-систем взагалі, є набір правил (сигнатур). Правила містятьть опис ознак атак та зловмисних впливів інших типів у прив’язці до різних апаратно-програмних об’єктів мережі [17]. У разі виявленні тієї чи іншої загрози система генерує відповідне попередження.

При цьому, для конфігурування правил також використовується файл snort.conf.

За підключення правил тут відповідає змінна RULE_PATH. Загальна форма запису рядка, де вказується шлях до директорії, що містить перелік правил, кожне з яких являє собою файл з розширенням *rules*, має вигляд:

```
var RULE_PATH ../rules
```

Означений вище параметр залишаємо без змін, що вказує на те, що правила будуть завантажуватися з локації за замовчуванням.

При цьому, для невеликої локальної мережі підприємства може бути використано стандартний перелік правил, без необхідності створення власних. У даному випадку важливо залишити активними лише ті з них, які для поточної конфігурації мережі є актуальними.

Зокрема, у нашому випадку Web-сервер розгорнуто на базі Apache, тому необхідно деактивувати Microsoft IIS, для якого існують окремі правила. Для цього достатньо закоментувати символом «#» рядок у файлі snort.config, у якому зазначається відповідне правило, тобто, маємо:

```
 # include $RULE_PATH/web-iis.rules
```

Натомість, знімемо коментар з рядку, який є актуальним у нашому випадку:

```
include $RULE_PATH/server-apache.rules
```

За замовчуванням, перелік активованих з самого початку правил у snort.conf є ефективним для більшості типових інформаційних систем та не потребує корегування.

Врешті решт, за потреби, правило для Snort може бути створено власноруч, відповідно до наступної структури та загального синтаксису:

```
<Дія> <Протокол> <IP-адреси джерел> <Порти джерел>
<Оператор направлення> <IP-адреси приймачів> <Порти
приймачів> (ключ_1 : значення_1; ключ_2 : значення_2;
... ключ_N : значення_N;)
```

Тут параметр «Дія» може приймати значення, як показано табл.3.1

Розглянемо приклад правила, як зазначається далі:

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS
$HTTP_PORTS (msg:"WEB-IIS cmd.exe access";
flow:to_server, established; content:
```

```
"cmd.exe";nocase;classtype:web-application-
attack;sid:1002;rev:5;).
```

У зазначеному правилі $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS вказує, що слід аналізувати лише трафік, що надходить до мережі ззовні (як визначено змінною EXTERNAL_NET).

У свою чергу, параметр content задає пошук послідовності символів cmd.exe у потоці даних. Виявивши таку послідовність, Snort генерує попередження, яке задається параметром msg.

Таблиця 3.1 – Опис параметру Дія правил Snort

Дія	Опис
alert	інформує про інцидент та вносить запис до лог-файлу
log	вносить запис до лог-файлу
pass	ігнорує пакети
activate	інформує про інцидент та активує встановлене динамічне правило
dynamic	дії відсутні до активування динамічним правилом, далі виконується лише логування
drop	блокує пакет та вносить запис до лог-файлу
sdrop	блокує пакет, не вносить запис до лог-файлу
reject	блокує пакет та вносить запис до лог-файлу, далі генерує повідомлення TSP RST, або ICMP-порт недоступний (залежно від протоколу)

Розглянемо приклад випадку, коли на рівні брендмауеру попередньо не було налаштовано блокування доступу до деяких зовнішніх вузлів, зокрема – соціальних мереж. У цьому разі можемо створити відповідне правило для Snort, як показано далі:

```
alert tcp any any -> any any (msg: "Someone accessing fb.com";
content: "fb.com";)
```

У наведеному прикладі система інформує адміністратора про факт доступу до fb.com. За потреби, параметр `alert` можемо змінити на інший потрібний.

3.4.5 Налаштування попереджень та журналів

Snort забезпечує запис інформації про інциденти у лог-файли на базі MySQL, SQL Server, Oracle або ODBC-сумісних баз даних. У свою чергу, вибір відповідного типу бази даних здійснюється у процесі інсталювання Snort [19].

При цьому, у ході запуску NIDS за допомогою команди Snort, ключ консолі A забезпечує виведення попереджень на екран.

Разом з тим, для того, щоб пересилати попередження до лог-файлу, необхідно замість ключа -A використовувати ключ -A fast, або -A full. У даному випадку параметр full виводить докладний опис загрози у кількох рядках текстового файлу з ім'ям alerts.ids у рамках директорії, шлях до якої вказує ключ -l.

Зазначений вище тип протоколювання дозволяє фіксувати вичерпні деталі. З іншого боку у випадку, коли у межах контрольованої мережі може фіксуватися велика кількість подій (інцидентів), процес їх аналізу та обробки перетворюється на досить складну задачу.

Виходячи з цього, у таких мережах рекомендується використовувати режим fast для внесення в alerts.ids однорядкових записів, які будуть містити ключові характеристики трафіку, який буде умовно маркованим, як підозрілий.

Актуальна на сьогодні версія засобу Snort забезпечує протоколювання у журналі подій Windows.

Водночас, для запису попереджень у журнал подій Application системи, у межах якої розгорнуто Snort, замість ключа -A використовується ключ -E (параметри не обов'язкові). На рисунку 3.5 продемонстровано, як виглядає подія Snort (у разі спроби звернення до cmd.exe), яка є опублікованою у журналі Application. Тут подія Windows забезпечує таку ж детальну інформацію, як екран консолі.

Тут необхідно зазначити, що використання будь-якої NIDS не має сенсу, якщо адміністратор виконує перегляд журналів подій (або зовнішніх журналів) досить рідко [4, 5, 16].

Інакше кажучи, у випадку виникнення інцидентів у мережі. адміністратор повинен дізнатися про це негайно.

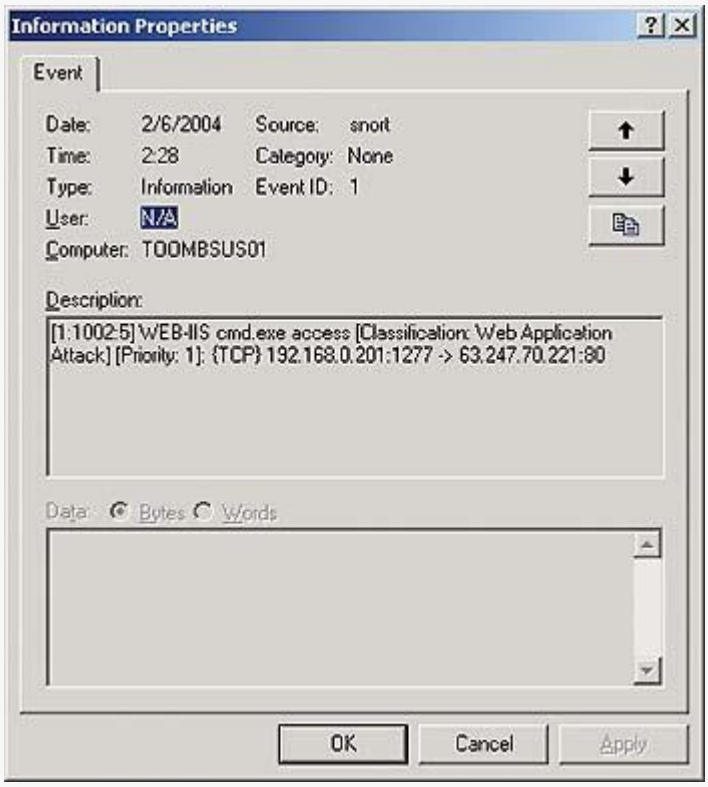


Рисунок 3.5 - Журнал подій Windows з повідомленням про атаку на cmd.exe

На той випадок, коли адміністратор може знаходитися за межами контрольованої мережі, сповіщення про інциденти може надсилати засіб EventSentry Light [20].

3.4.6 Запуск засобу Snort у вигляді служби

За умови, що попередньо було завершено усі попередні налаштування, далі засіб Snort можемо використовувати як службу [17].

При цьому, якщо Snort буде активовано з параметрами /SERVICE та /INSTALL (поряд з іншими параметрами командного рядка), надалі Snort налаштовується на роботу як служба Windows, та автоматично активується у процесі запуску ОС Windows без втручання користувача.

Отже, Snort є повнофункціональним додатком. Проте, у ряді випадків засіб потребує розширення.

Наприклад, якщо у межах різних ділянок мережі розгорнуто кілька NIDS, то, відповідно, найбільш зручно керувати Snort з графічного інтерфейсу. Дану можливості реалізовано у модулях розширення IDScenter Engage Security, а також IDS Policy Manager Activeworx.

Також, у деяких випадках буває необхідно проаналізувати інформацію, що міститься у повідомленнях (інформування адміністратора). У даному випадку для перегляду та аналізу збережених даних може бути використано модуль Analysis Console for Intrusion Databases (ACID).

4. ЗАХИСТ ВЕБ-КОМПОНЕНТИ МЕРЕЖЕВОЇ ІНФРАСТРУКТУРИ

4.1 Напрямки забезпечення захищеності веб-компоненти компанії

У загальному випадку веб-компонентою мережевої інфраструктури підприємства може вважатися веб-сайт.

При цьому, незалежно від того, де фізично розміщений веб-сервер, для нього необхідно забезпечити щонайменше:

- захищеність від DDoS-атак;
- захист від розміщення зловмисного коду;
- захист від brootforce-атак (повний перебір паролей).

Схема захисту сайту зображається на рис.4.1

При цьому, однією з найбільш істотних загроз для сайту є розміщення зловмисного коду хоча б в одному з файлів на сервері.

Навіть при тому, що зловмисний модуль буде безпечним як для серверного ПЗ, так і контенту сайту, а також для внутрішньої мережі компанії, сам факт його наявності може викликати істотне падіння пошукових рейтингів сайту або його повне видалення з пошукового каталогу. Це може статися в тому випадку, якщо хоча б один шкідливий модуль буде виявлено пошуковою системою (в т.ч. в ході індексації).

Отже, потрібно убезпечити сайт від розміщення зловмисного вмісту, що може бути реалізовано за рахунок:

- блокування можливостей розміщення шкідливого коду зловмисником:

- блокування можливостей розміщення шкідливого коду відвідувачами сайту:

- блокування можливості випадкового розміщення шкідливого коду адміністратором сайту.

4.2 Захист від DDoS-атак

Одним з зовнішніх засобів блокування атак DDoS, є відповідний інструментарій онлайн-сервісу Cloudflare. Даний сервіс, окрім проксіювання

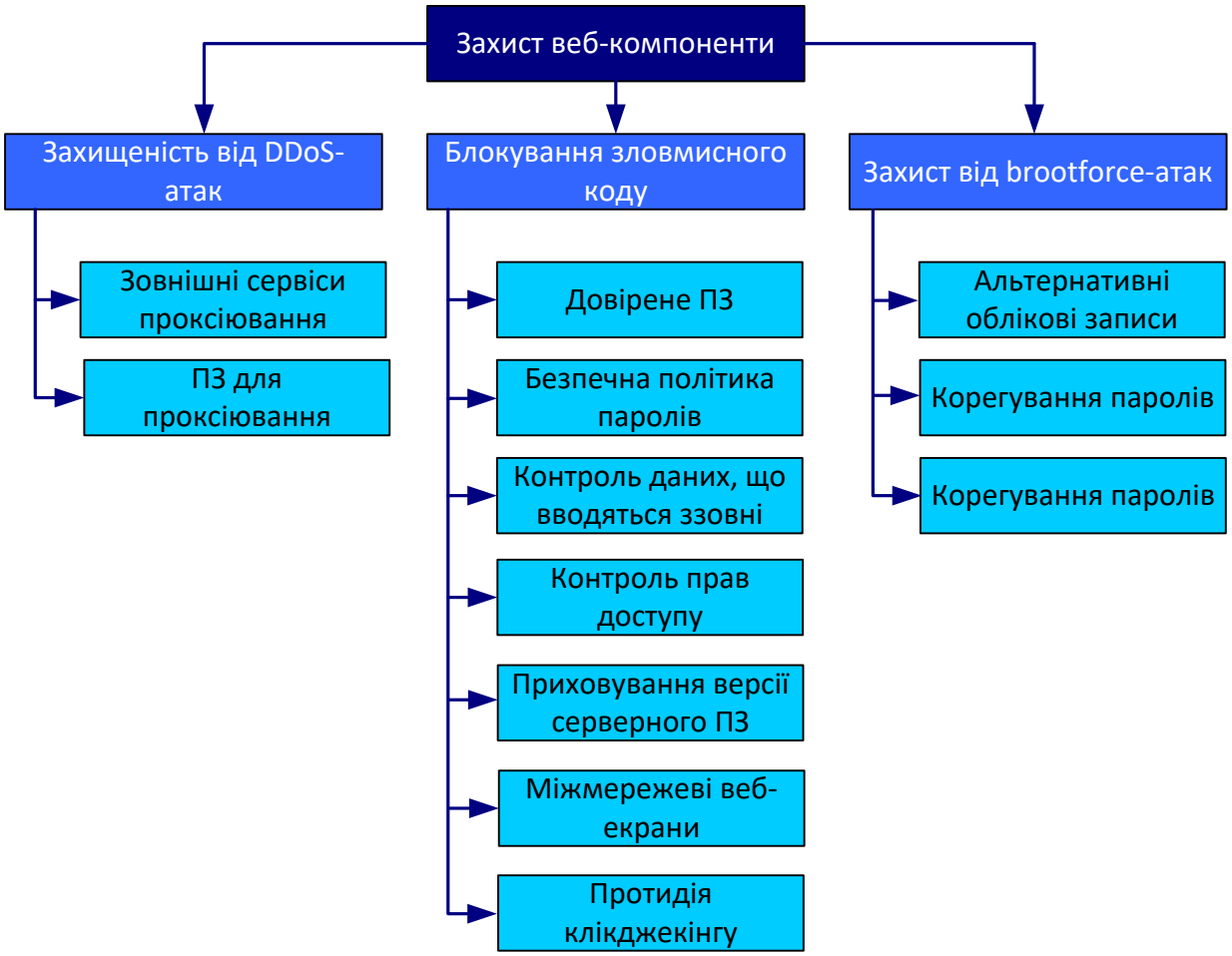


Рисунок 4.1 – Один з прикладів схем захисту веб-компоненти

трафіку (що є його ключовою можливістю), також надає користувачеві значний перелік послуг як з галузі безпеки, так і галузі оптимізації [21].

При цьому, користуючись можливостями, які надає Cloudflare, може бути побудовано захист сайту від DDoS-атак, сконфігуровано оптимізацію сайту а також кешування, налаштовано захищене з'єднання, аналіз трафіку тощо.

4.2.1 Первинне конфігурування Cloudflare

Робота з Cloudflare починається зі створення облікового запису. Для цього необхідно:

- на сайті Cloudflare звернутися до сторінки Sign Up;
- створити обліковий запис, для чого ввести пошту та пароль облікового запису, після чого можливо свторити акаунт натисненням на

кнопку «Create Account» (рис.4.2). У свою чергу, обліковий запис буде створено після підтвердження електронної пошти.

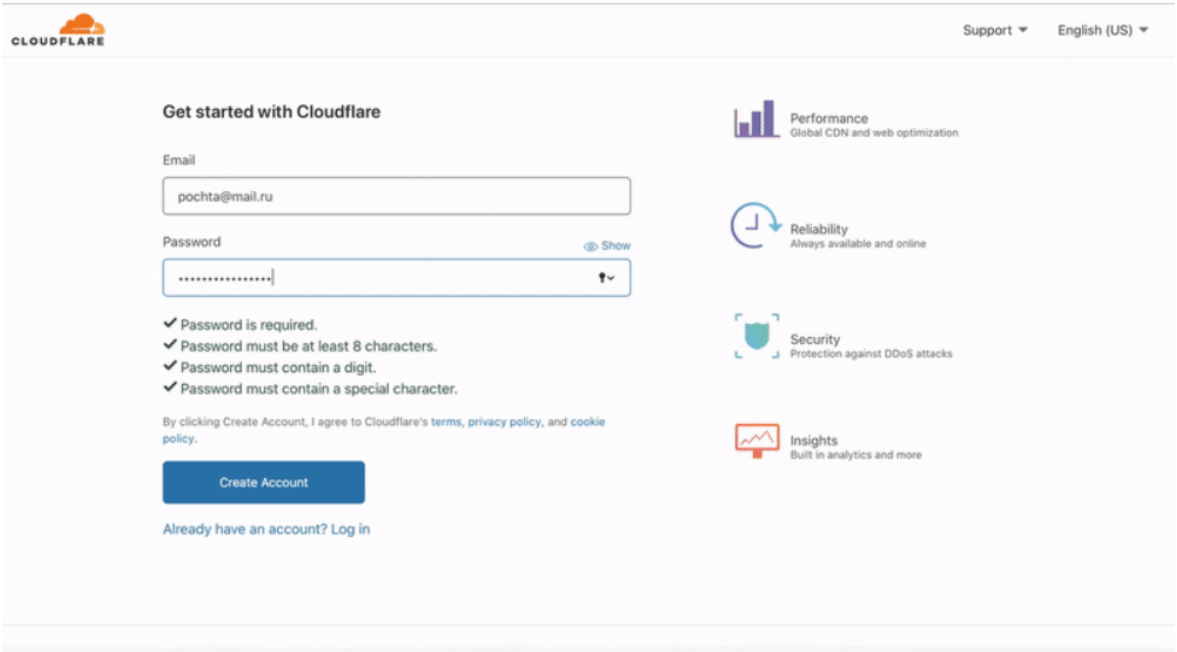


Рисунок 4.2 – Створення облікового запису Cloudflare

4.2.2 Додавання домену

Після того, як було створнено обліковий запис, необхідно ввести ім'я домену, відносно якого буде активовано сервіси Cloudflare.

На даному етапі налаштування необхідно зазначити зареєстрований домен, після чого натиснути кнопку «Add Site».

Далі слід обрати а після цього - прийняти один з запропонованих тарифних планів, та підтвердити вибір натисненням кнопки «Confirm plan» (рис. 4.3).

При цьому, у ході налаштування Cloudflare виконує сканування DNS-записів домену.

Тобто, попередньо необхідно виконати перевірку нанаштувань на предмет включення усіх DNS-записів і за необхідності додати відсутні, якщо такі є [21].

Рисунок 4.3 – Додавання домену у Cloudflare за обраним тарифом

Важливо також, що у ході даного етапу налаштування може бути активовано проксіювання для усіх записів (рис.4.4).

У даному випадку для того, щоб мати змогу застосувати усі налаштування, слід змінити сервери NS у реєстратора домену на ті, які буде запропоновано Cloudflare. Зазначимо, що платформа має у своєму розпорядження кілька сотень NS-серверів.

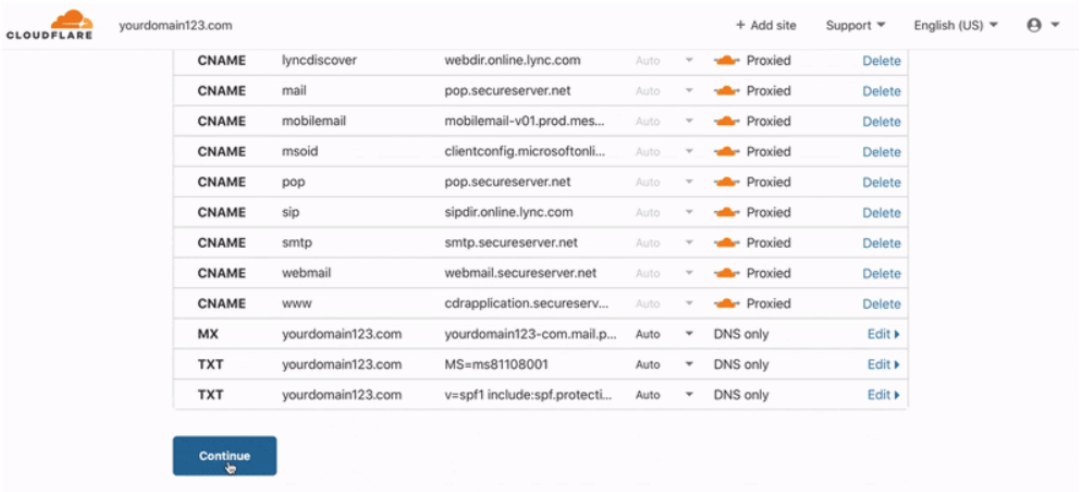


Рисунок 4.4 – Активація проксіювання облікових записів

У цьому випадку, після додавання домену до облікового запису, а також виконання зміни (за необхідності) NS-серверів, та наступного оновлення DNS-кешу, домен отримає підтвердження реєстрації та далі буде успішно взаємодіяти з зовнішнім оточенням через Cloudflare.

При цьому, факт підтвердження реєстрації фіксується відповідним повідомлення на пошту.

4.2.3 Налаштування DNS та статусу проксіювання

Виконувати процедури зі зміни, додавання та/або редагувати ресурсних записів користувач має можливість у розділі «DNS».

Водночас, для облікових записів користувач може обрати один з наступних статусів проксіювання, а саме [21]:

1. Налаштування «Proxied» (помаранчева хмара в інтерфейсі конфігурування) свідчить про те, що замість IP-адреси контрольованого сайту буде зазначено адресу Cloudflare.

Таким чином, у даному статусі усі запити зазнають проксіювання, тобто, у першу чергу надходять до одного з серверів Cloudflare, після чого далі спрямовуються до серверу, де сайт фізично розміщений.

2. DNS only (сіра хмара в інтерфейсі конфігурування) вказує на те, що у разі виконання звернення до домену, Cloudflare буде віддавати IP-адресу контрольованого сервера, який було зареєстровано.

У цьому випадку конфігурування кешування, firewall та інші опції не буде застосовано. Разом з тим для того, щоб змінити статус, необхідно натиснути на рядок облікового запису для того, щоб перейти у режим редагування (рис.4.5).

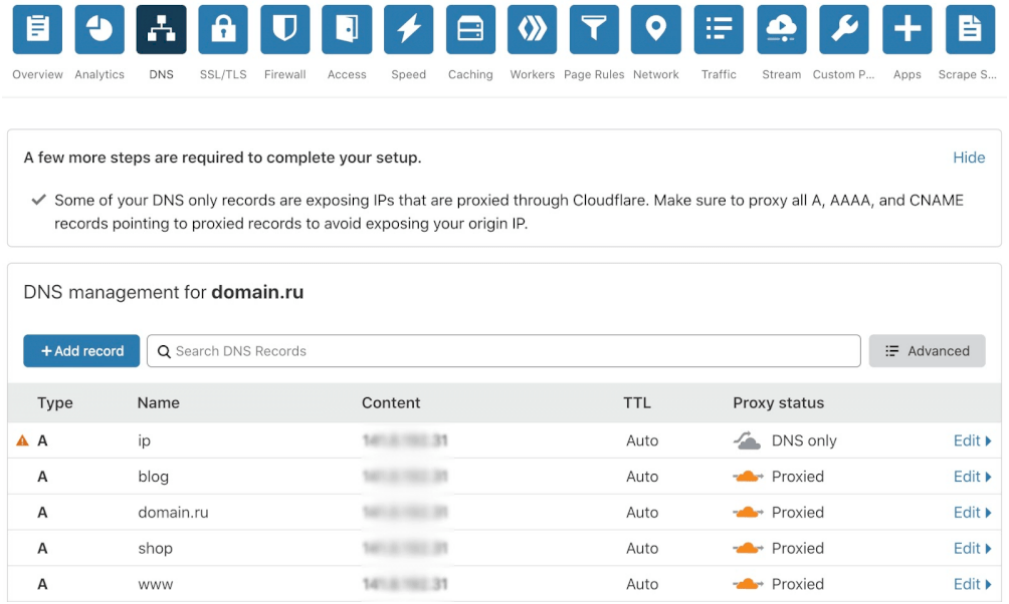


Рисунок 4.5 – Налаштування облікового запису у режимі редагування

При цьому, використання режиму проксіювання на безкоштовному тарифному плані має ряд обмежень.

Зокрема, налагодити зв'язок з доменом, що проксується, на базі FTP, SSH та поштових протоколів неможливо.

У зазначеному режимі Cloudflare дозволяє виконувати запити до контрольованого сайту виключно за 443 та 80 портах, виходячи з цього, для забезпечення можливості підключення до пошти та інших сервісів слід використовувати IP-адресу сервера, де знаходиться обліковий запис контрольоваго сайту.

Цю інформацію може бути внесено у розділі налаштувань облікового запису у локації:

Панель управління → Сайти та домени → IP-адреси.

4.2.4 Налаштування захисту від DDoS-атак

Беручи до уваги сучасний стан інформаційного середовища, є очевидним, що кількість кібератак, зокрема – DDoS, постійно збільшується, відтак, необхідно застосувати дієву систему захисту від них.

У цілому, тут може бути застосовано 2 парадигми побудови такого захисту, а саме [21]:

- використання засобів захисту від DDoS на рівні окремого ресурсу;
- зихист від DDoS на базі зовнішніх інструментів/сервісів.

У першому випадку передбачається реалізація усіх інструментів захисту на рівні серверу/хостингу.

При цьому, продуктивність таких механізмів визначається з урахуванням обмежень, які мають сервер та/або хостинг.

У другому випадку, захист як веб-сайтів, так і окремих процесів у їх складі, забезпечується глобальною хмарною мережею компанії, яка, при цьому, надає швидкістю передачі даних на рівні кількох десятків Тбіт/с. У цьому разі як розмір, так і частота атак технічно не обмежується, незважаючи на це, мережа Cloudflare блокує до сотні мільярдів загроз на день.

Разом з тим, для того, щоб мати змогу скористатися функціоналом захисту від DDoS-атак, який надає Cloudflare, необхідно увімкнути проксіювання (Proxied) для контрольованого сайту у налаштуваннях розділу

«DNS». У наслідок цього усі запити до контрольованого сайту першочергово будуть надходити до серверу Cloudflare, який, у свою чергу, виконуватиме надсилання перевірених запитів до вузла, звідки фактично працює сайт (рис.4.6).

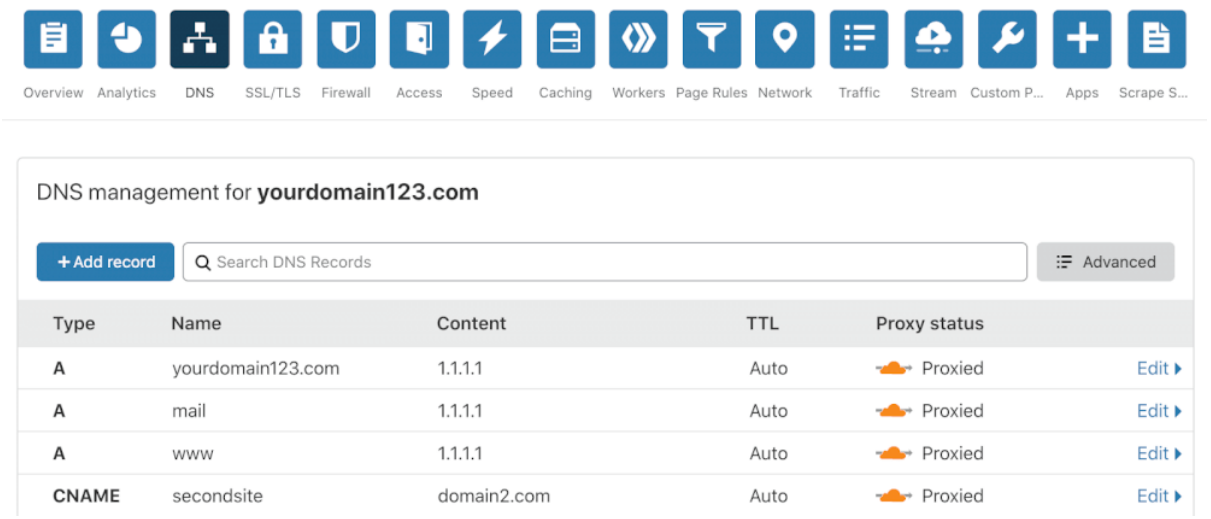


Рисунок 4.6 – Налаштування захисту від DDoS на базі Cloudflare

4.2.5 Побудова додаткового захисту сайту на базі Web Application Firewall

У загальному випадку, WAF може розглядатися, як своєрідний набір правил, застосовуючи які може бути налаштовано додаткові механізми захисту сайту від зловмисних впливів.

Налаштування зазначених правил WAF, у свою чергу, може виконуватися у розділі Firewall у межах особистого кабінету Cloudflare (рис.4.7). Такі правила, у свою чергу, дають змогу блокування запитів, отриманих від небажаних роботів, також надають доступ до сайту для довірених (перевірених) сервісів, виконувати налаштування додаткових перевірок для роботів тощо.

При цьому, кожен запит у WAF зазнає перевірки за списком правил та використовуючи механізми, що базуються на застосуванні штучного інтелекту. Водночас, у ході виконання такої перевірки підозрілі запити може бути як заблоковано так і додатково перевірено, або занесено до журналу залежно від того, якими є первинні налаштування користувача, хоча, при цьому, легітимні запити успішно надсилаютимуться до контрольованого сайту.

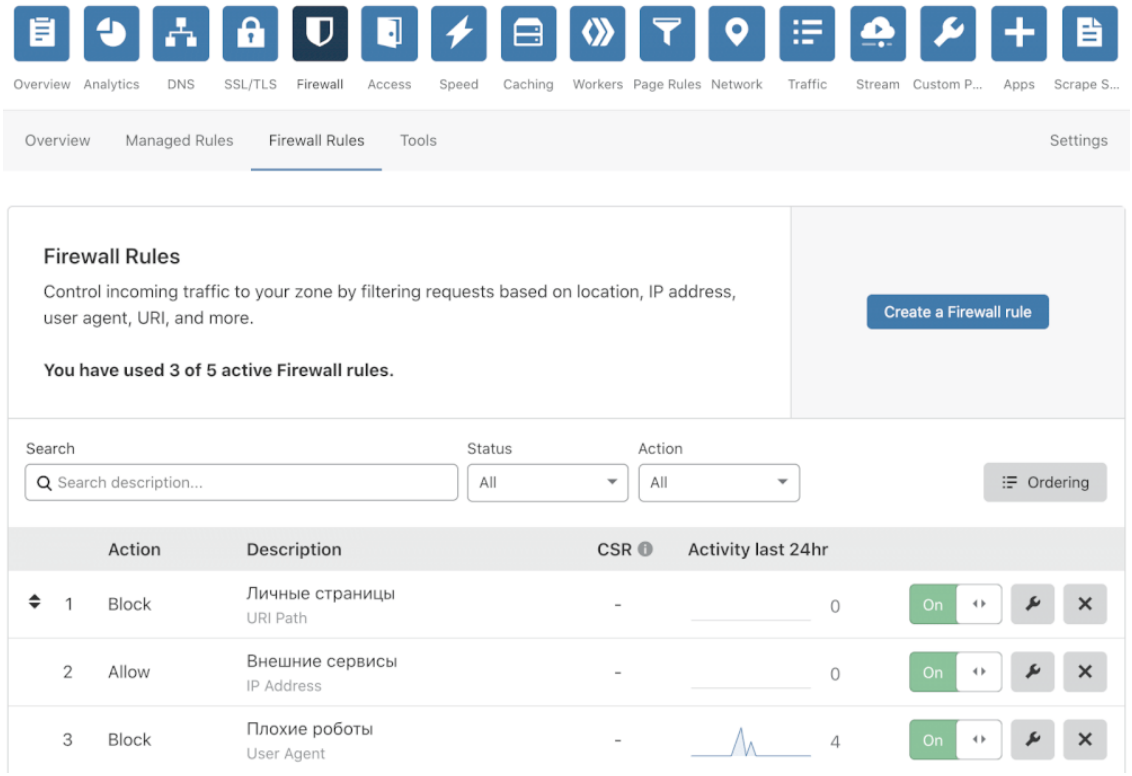


Рисунок 4.7 – Налаштування захисту Web Application Firewall

4.2.6 Побудова захищених з'єднань на базі Cloudflare

При використанні проксіювання від Cloudflare, для використання є доступним налаштування захищеного з'єднання для сайту в різних режимах, зокрема, Flexible, Full або Full (Strict) [21].

У загальному випадку, відмінності між зазначеними режимами ілюструє рисунок 4.8.

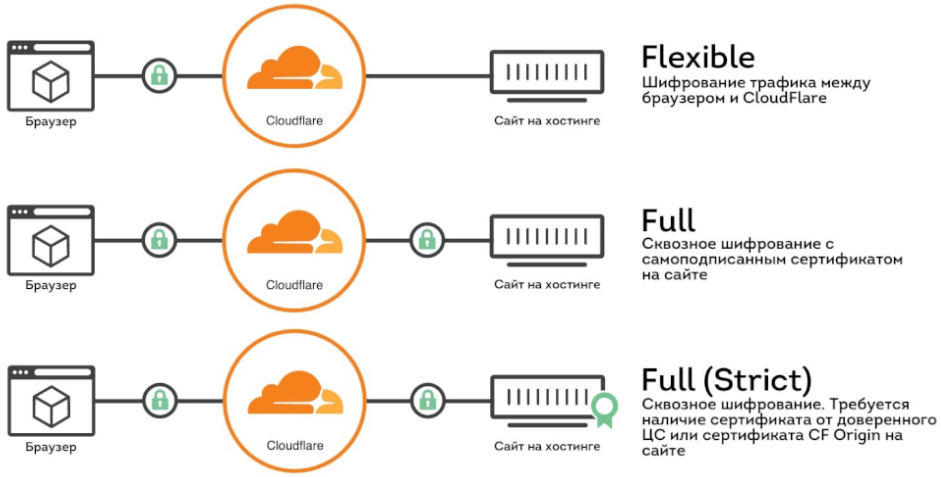


Рисунок 4.8 – Відмінності між режимами захищених з'єднань

Тут слід звернути увагу на те, що режим Flexible шифрує трафік тільки між браузером і сервером Cloudflare. На думку сервісу, таке налаштування здатне допомогти особистим сайтам та блогам у просуванні, оскільки Google та інші пошукові системи звертають увагу на HTTPS.

4.3 Захист від розміщення зловмисного коду

4.3.1 Захист від розміщення шкідливого коду зловмисником

Тут можна виділити низку організаційно-технічних превентивних заходів, як [5]:

1. Застосування довіреного ПЗ. У рамках цього рекомендується використовувати дистрибутиви веб-додатків та розширень/плагіні для CMS тільки з перевірених джерел, а також їх регулярне оновлення. Крім того, періодично потрібно виконувати аудит безпеки серверів.

У разі, якщо сайт функціонує під управлінням однієї зі стандартних CMS, після її встановлення необхідно усунути інсталяційні та налагоджувальні скрипти.

2. Використання безпечної парольної політики для веб-серверного ПЗ (FTP, SSH, адміністративні панелі хостингу та CMS). Елементами такої політики можуть бути:

- встановлення вимог до паролю за мінімальною довжиною і змістом символів (букви в різних реєстрах, цифри, спеціальні символи);
- заборона використання однакових паролів для доступу до різних сервісів;
- заміна навіть найнадійніших паролів з деякою періодичністю;
- заборона на збереження паролів у веб-браузерах, файлових менеджерах, а також FTP- та SSH- клієнтах.
- відстежування безпеки робочих станцій. У межах даного аспекту щонайменше необхідно забезпечити присутність на усіх робочих станцій, які задіяні у процесі роботи з веб-сервером (АРМ вебмайстра, адміністратора сайту, контент-менеджера, менеджера з продажів тощо) антивірусних пакетів з підтримкою регулярних оновлень. Також слід своєчасно виконувати оновлення ОС та прикладного ПЗ.

3. Контроль даних, які вводять користувачі сайту. Сюди, у свою чергу, можна віднести [1, 5]:

- фільтрацію HTML-розмітки серед даних, що вводяться користувачем, у першу чергу. серед тих, які може бути вбудовано у код сторінок сайту;

- заборону на розміщення отриманих від користувачів даних безпосередньо до виклику `eval()`, SQL-запитів чи у перетворення типів. При цьому, завжди слід перевіряти та видаляти дані. Отримані від потенційно небезпечних елементів;

- заборону на розміщення у робочій версії коду параметрів, які було задіяно для відлагодження функціоналу, експерементування з новою чи деактивованою функіцональністю.

4. Контроль прав доступу користувачів. У рамках цього, зокрема, у першу чергу необхідно реалізувати захист від міжсайтової підробки запитів (CSRF). Окрім цього, слід обмежити доступ до адміністративних панелей CMS и БД (наприклад, до phpMyAdmin), а разом з цим до [1, 5]:

- резервних копій коду;
- файлів конфігураії;
- метаданих систем контролю версій (зокрема, до каталогів `.svn` або `.git`).

5. Приховування версій серверного ПЗ (у т.ч. CMS, веб-серверу, також інтерпретатору сценаріїв та СУБД).

6. Конфігурування міжмережевих екранів у такий спосіб, що дозволяє лише ті з'єднання, які необхідні для роботы.

7. Протидія клікджекінгу. Для цього, як один з прикладів, достатньо застосувати на сторінці Javascript-конструкції вигляду:

```
if (top.location != window.location) top.location =
window.location
```

або

```
top.location = 'http://example.
```

4.3.2 Захист від розміщення зловмисного коду користувачами сайту

У тому випадку, коли користувачі сайту мають змогу завантажувати файли або текст на контрольований сайт, існує ризик потрапляння зловмисного коду разом з завантаженим контентом – випадково, або

цілеспрямовано. Для протидії цьому у загальному випадку може бути використано 2 ключових механізми, а саме:

1. Захист від ботів.

У цілому, щоб захистити веб-ресурс від роботів-зламників, може бути застосовано спеціалізовані плагіни для CMS, або постійно виконувати пошук IP-адрес відвідувачів у чорних списках.

2. Перевірка даних, які могли бути введені відвідувачами. До цього комплексу заходів у загальному випадку можемо віднести:

- блокування можливості розміщення коду JavaScript усередині конструкцій <script>, а також у тегах та посиланнях;
- заборона на пряме розміщення на сторінках сайту у межах тегів <iframe>, <object> та <embed>;
- заборона на завантаження файлів .jar, .swf та .pdf, так як з їхньою допомогою ряд тегів может генеруватися автоматично.
- впровадження та використання «білого списку» дозволених HTML-тегов, керуючись яким дозволяти, або – навпаки, забороняти, розміщення користувацького контенту;
- перевірка посилань, розміщених користувачами, використовуючи чисельні сервіси та механізми Safe Browsing, які надаються у т.ч. пошуковими системами.

4.3.3 Захист від випадкового розміщення зловмисного коду адміністратором сайту

Для того, щоб уникнути ситуацій, коли зловмисний модуль завантажено у ході сеансу адміністрування сайту, необхідно:

1. Виконувати перевірку використовуваного ПЗ для адміністрування, а саме:

- здійснювати завантаження дистрибутивів CMS, бібліотек, плагінів та віджетів виключно з офіційних сайтів розробників, або, щонайменш, з перевірених джерел;
- у разі необхідності завантаження дистрибутиву з сумнівногог джерела слід в обов’язковому порядку виконати його перевірку на предмет присутності зловмисного коду;
- виконувати детальне дослідження будь-яких додаткових сторонніх скриптів, які необхідно додати до CMS.

2. Здійснювати моніторинг рекламних блоків та супутнього коду. Сюди можемо віднести [4]:

- використання на сторінках сайту виключно тих рекламних блоків, які було надано апріорі надійними (перевіреними) рекламними системами;
- здійснення попереднього дослідження відгуків партнерської системи, а також прикладів контенту, який у ній розповсюджується;
- уникнення «унікальних пропозицій», які характеризуються підозріло високою платнею за лічильники та блоки, а також монетизацію мобільного трафіку;
- орієнтація на використання статичного контенту (посилання, тексти та зображення) на сторінках сайту. При цьому, слід уникати завантажуваних елементів script та iframe. Також компоненти Flash, Java та ActiveX необхідно приймати виключно у вигляді вихідного коду, який може бути перевірено, а далі – скомпільовано самостійно;
- уникнення використання партнерських програм, які мають приховані блоки.
- детально контролювати доступ до інтерфейсів службового типу. Тобто, можливість здійснення службового доступу до контрольованого сайту повинні мати виключно ті особи, для котрих такий доступ є виключно необхідним. Така необхідність, у свою чергу, зберігається протягом часу, коли це необхідно. Сюди входить:
- анулювання доступу фахівців, які виконують разові роботи на сайті, це саме стосується попередніх власників (якщо такі є), також осіб, щобезпосередньо не відповідають за функціонування сайту. Це, зокрема, може бути веб-дизайнер, або керівники структурних підрозділів компанії;
- залучення для роботи з сайтом сторонніх осіб, які мають рекомендації довірених джерел. Разом з тим, після виконання запланованих робіт їхні облікові записи має бути анульовано, а у випадку подальшого використання - змінено паролі;
- блокування доступу за FTP для представників партнерських програм у випадку, коли контрольований сайт є статичним. Час від часу деякі партнерські системи можуть виконувати запит на доступ за FTP для того, щоб, наприклад, виконувати самостійну заміну банерів. Водночас, тут обов’язково слід пам’ятати про те, що надавати доступ подібним чином небезпечно. Наприклад, якщо базу даних партнерської системи буде зламано,

за результатом цього зловмисник може може одержати безпосередній доступ до файлів на контрольованому сайті.

3. Орієнтуватися виключно на використання надійного, якісного та перевіреного хостингу. Тут слід брати до уваги те, що, нажаль, не всі існуючі сьогодні хостері здатні винятково якісно забезпечувати захищеність своїх серверів.

При цьому, слід також брати до уваги те, що деякі з хостерів можуть цілеспрямовано сприяти (або безпосередньо брати участь) у процедурах інфікування сайтів клієнтів.

4.4 Захист від brute force-атак

Найбільшою мірою проблематика захисту від brute force-атак актуальна для сайтів, що знаходяться під управлінням ряду поширених CMS – зокрема – WordPress.

Рішення даної проблеми може бути досягнено за ряд технологічних кроків, як то [22]:

- зміна стандартного облікового запису для входу;
- корегування адміністративного паролю;
- зміна стандартного файлу авторизації.

4.4.1 Зміна стандартного облікового запису

На даному технологічному кроці, першочергово необхідно виконати деактивацію користувача admin.

Означена процедура для версій WordPress, починаючи з 3.0, виконується наступним чином:

- створюється новий користувач;
- створеному користувачеві призначаються адміністраторські повноваження;
- видаляється старий користувач з іменем «admin».

У випадку, коли така операція виконується відносно попередній версій WordPress, її може бути реалізовано одним з настипних пари SQL-запитів:

```
update wp_users set user_login= 'ваш новий логін' where user_login= 'admin';
```

або

```
update wp_posts set post_author= 'ваш новий логін' where post_author= 'admin'
```

4.4.2 Корекція паролю адміністратора

У загальному випадку, така процедура виконується тоді, коли:

- довжина паролю є недостатньою (менш ніж 10-12 символів);

- символи, що формують пароль, являють собою (хоча б частково) осмислений вираз, частину слова чи числову послідовність;

- перелік дозволених символів для формування парольної фрази обмежено цифрово-літерним простором.

Ефективним може вважатися пароль, що являє собою хаотичну комбінація великих та малих літер, знаків і цифр не менше ніж з 10-12 символів довжиною.

4.4.3 Зміна стандартного файлу авторизації

За замовчуванням вхід до адміністративної панелі WordPress здійснюється через файл «wp-login.php», де містяться поля вводу користувацьких даних – логіну та паролю. У таких умовах зловмисник матиме ймовірну точку атаки, де спробує підібрати пароль на базі brute force-механізмів [22].

Тоді перш за все розглянемо ситуацію, коли веб-вузол має єдиного легітимного адміністратора, який використовує статичну IP-адресу. У такому разі, доцільно встановити доступ до директорії «wp-admin» виключно з власної IP-адреси адміністратора, тим самим блокуючи навіть можливість виконання авторизації для усіх інших користувачів. Для цього у директорії «wp-admin» створюється файл «.htaccess», куди вносяться наступні рядки команд:

```
order deny,allow
deny from all
allow from IP
```

Тут IP — це IP-адреса вузла, звідки дозволяється доступ.

У свою чергу, тоді, коли необхідно забезпечити можливість підключення іншим особам, що також мають статичні IP, їх може бути додано до списку, як показано фрагментом коду далі:

```
order deny,allow
deny from all
allow from IP1
allow from IP2
allow from IP3
```

де IP1, IP2 - IP3 - IP-адреси вузлів, з яких дозволено доступ до адміністративної панелі WordPress.

Після виконання зазначеної операції усі користувачі (або боти), IP-адреси яких відсутні у переліку дозволених, фактично не отримають доступу до директорії «wp-admin». Відтак реалізація brute force-атак відносно файлу «wp-login.php» для них буде неможливою. При цьому, у разі звернення до даного ресурсу, повертатиметься помилка 403.

Розглянемо інший випадок, коли провайдер призначає користувацьким вузлам (у т.ч. адміністратору також) динамічну IP-адресу, що змінюється під час кожного нового підключенням до Інтернету.

Зрозуміло, у зазначеній ситуації розглянутий вище спосіб не може бути використано, так як «.htaccess» тоді має редагуватися при кожному підключенні до мережі. Натомість, шляхом подальшого налаштування файлу «.htaccess» може бути встановлено додаткову серверну HTTP-авторизацію, для чого виконуються наступний перелік операцій:

1. У межах директорії «wp-admin» створюються файли «.htaccess» та «.htpasswd». При цьому, перший файл міститиме інструкції а другий - дозволені дані для доступу до директорії.

У свою чергу, до файлу «.htaccess» вноситься наступний запис:

```
AuthType basic
AuthName 'Access Denied'
AuthUserFile '/fullpath/.htpasswd'
Require valid-user
DirectoryIndex index.php
```

У наведених вище директивах змінна *fullpath* вказує на повний шлях до файлу «.htpasswd». Разом з тим, повний шлях до цього файлу може бути отримано або від хостинг-провайдера або, виконавши наступний php-скрипт:

```
$dir = dirname(__FILE__);
```

```
echo 'Full Path: ' . $dir . '/';
```

Або застосувавши альтернативний скрипт:

```
echo 'Повний шлях: ' .$_SERVER['DOCUMENT_ROOT'] . '/';
```

2. Виконати блокування доступу до файлу «wp-login.php» у спосіб, аналогійний розглянутому вище.

Для цього у «.htaccess» створюється наступний запис вигляду:

```
<FilesMatch"wp-login.php">
AuthName'Access Denied'
AuthType Basic
AuthUserFile'/fullpath/.htpasswd'
Require valid-user
```

При цьому, може бути використано той же самий файл «.htpasswd». Його внутрішній зміст має виглядати приблизно так, як зазначено далі:

```
username:password
```

Тут username являє собою ім'я дозволеного користувача, тоді як password – його пароль. Оскільки пароль має зберігатися у вигляді хешу, попередньо виконується його шифрування з використанням одного з відповідних онлайн-сервісів.

Отже, у випадку, коли необхідно внести записи про декількох користувачів з паролями, їх може бути внесено у «.htpasswd» таким чином, щоб в одному рядку містився запис виключного одного користувача, наприклад:

```
username1:password1
username2:password2
username3:password3
```

Таким чином, якщо після впровадження розглянутих налаштувань будуть виконуватися спроби доступу до адмінпанелі WordPress, перед авторизацією необхідно буде ввести логін і пароль доступу до директорії

«wp-login.php», тобто, да файлу авторизації. Подальший доступ до авторизаційного інтерфейсу буде можливим лише після успішного входу.

ВИСНОВКИ

У відповідності до технічного завдання, у ході виконання кваліфікаційної роботи було виконано:

- дослідження передумов, що визначають необхідність застосування захисних механізмів, методів та засобів по відношенню до мережевої інфраструктури і даних у цілому;
- визначення даних, що потребують захисту від неавторизованого доступу;
- виявлення рівнів захисту мережевої інфраструктури;
- визначення та дослідження типових кроків налаштування ключових компонент захисту мережевої інфраструктури для мереж невеликого масштабу.

Зокрема, було розглянуто загальну схему захисту мережі, у складі:

- міжмережевого екрану класичного типу з лінійки Keenetic;
- системи виявлення та протидії вторгненням, у ролі якої використовується NIDS Snort;
- мережевих сервісів, інструментів у складі CMS та панелі керування хостингом а також ряду організаційно-технічних заходів, спрямованих на захист веб-компоненти мережі підприємства.

У свою чергу, на базі використання міжмережевого екрану у розглянутій схемі забезпечується блокування проходження трафіку непривілейованих та/або підозрілих джерел.

При цьому, на NIDS-систему покладено завдання виявлення різнорідним аномаліям у контрольованій мережі, а також протидія зловмисним впливам, як джерелам таких аномалій.

Такими зловмисними впливами можуть бути вірусні модулі та інші приклади шкідливого ПЗ (черви, трояни тощо) або кібератаки.

Водночас, у ролі веб-компоненти мережевої інфраструктури, за умовою завдання, було розглянуто сайт компанії, який розміщено на хостингу у зовнішньому мережевому оточенні.

Для його захисту від зловмисних впливів було застосовано схему, що забезпечує протидію:

- спробам розміщення зловмисного коду;
- атакам DDoS;
- brute force-атакам на адміністративну панель CMS сайту.

Отже, усі пункти технічного завдання було виконано у повному обсязі.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Шаньгін В.Ф. Інформаційна безпека та захист інформації. ДМК-Прес., 2017, 702 с.

2. The Computer History Museum, SRI International, and BBN Celebrate the 40th Anniversary of First ARPANET Transmission, Precursor to Today's Internet [Електронний ресурс] – Режим доступу: https://www.sri.com/newsroom/press-releases/computer-history-museum-sri-international-and-bbn-celebrate-40th-anniversary.

3. Tananbaum A., Wetherall D. Computer networks / Andrew S. Tanenbaum, David J. Wetherall. – Pearson Education, Inc. - 2011. – 933 p. ISBN-10: 0-13-212695-8.

4. El-Monem A., El-Bawab A. Untangle Network Security. – Packt Publishing, - 2014. – 564 p. ISBN: 9781849517720.

5. Sadiqui A. Computer Network Security. - Wiley-ISTE, - 2020. - pp: 655 p. ISBN: 9781786305275.

6. Microsoft security report [Електронний ресурс] – Режим доступу: https://microsoft.com/securityinsights.

7. Antivirus and Cybersecurity Statistics & Facts 2021 [Електронний ресурс] – Режим доступу: https://wethegeek.com/antivirus-statistics-facts/

8. Stood, A. Targeted Cyber Attacks. — Elsevier Inc.. — 2014. — ISBN 978-0-12-800604-7.

9. Шаньгін В.Ф. Захист інформації в розподілених корпоративних мережах і системах [Текст]: підручник / В.Ф. Шаньгин, А.В. Соколов. – М.: ДМК Пресс, 2002. – 656 с.

10. Malware types – Viruses, Keyloggers, Worms, Trojans, etc. [Електронний ресурс] Режим доступу: https://entri.app/blog/malware-types-viruses-keyloggers-worms-trojans-etc/

11. Types of Network Firewall [Електронний ресурс] Режим доступу: https://www.geeksforgeeks.org/types-of-network-firewall/?ysclid=lvp3sapfv2538210839

12. Cisco Secure Firewall - Cisco [Електронний ресурс] Режим доступу: https://www.cisco.com/site/us/en/products/security/firewalls/index.html?ysclid=lvp406v2e951196669

13. Keenetic Firewalls [Електронний ресурс] Режим доступу: https://keenetic.com/en/products

14. Check Point Next Generation Firewalls [Електронний ресурс] Режим доступу: https://www.checkpoint.com/quantum/next-generation-firewall/

15. Firewall typical using [Електронний ресурс] Режим доступу: https://help.keenetic.com/hc/ru/articles/360000991640internet-center-connection#3

16. Snort – Network Intrusion Detection System [Електронний ресурс] Режим доступу: https://snort-org.herokuapp.com/

17. Snort - Intrusion Detection System & Prevention System | Installation & Use in Windows [Електронний ресурс] Режим доступу: https://techofide.com/blogs/snort-intrusion-detection-system-prevention-system-installation-use-in-windows/

18. WinPcap Tool [Електронний ресурс] Режим доступу: https://winpcap.polito.io

19. Snort Tutorial: How to use Snort intrusion detection resources [Електронний ресурс] Режим доступу: https://www.techtarget.com/searchitchannel/tutorial/Snort-Tutorial-How-to-use-Snort-intrusion-detection-resources

20. EventSentry Light [Електронний ресурс] Режим доступу: http://www.netikus.net/products_downloads.html

21. Connect, protect and build everywhere | Cloudflare [Електронний ресурс] Режим доступу: https://www.cloudflare.com/

22. WordPress Brute Force Protection: 4 Steps to Prevent Attacks [Електронний ресурс] Режим доступу: https://jetpack.com/blog/wordpress-brute-force-protection/