Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Інститут електроенергетики
Факультет інформаційних технологій
Кафедра безпеки інформації та телекомунікацій

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеня магістра

студента *Харитонова Івана Андрійовича*

академічної групи *125м-19-1*

спеціальності *125 Кібербезпека*

спеціалізації[1]

за освітньо-професійною програмою *Кібербезпека*

на тему ***Методика захисту інформаційно-телкомунікаційнихс истем комерційних підприємств від DoS атаки.***

Керівники	**Прізвище, ініціали**	**Оцінка за шкалою**		**Підпис**
		рейтинговою	**інституційною**	
кваліфікаційної роботи	К.ф.-м.н., доц Гусєв Олександр Юрійович			
розділів:				
спеціальний	Ст.в Саксонов Генадій Михайлович			
економічний	К.е.н., доц. Пілова Дар'я Петрівна			

Рецензент				

Нормоконтролер	Ст.в. Тимофєєв Д. С.			

Дніпро
2020

ЗАТВЕРДЖЕНО:
завідувач кафедри

ЗАВДАННЯ
на кваліфікаційну роботу ступеня магістра

студенту *Харитонову І.А.* **академічної групи** *125м-19-1*
(прізвище та ініціали) (шифр)

спеціальності *125 Кібербезпека*

спеціалізації

за освітньо-професійною програмою *Кібербезпека*

на тему ***Методика захисту інформаційно-телекомунікаціх систем комерційних підприємств в DOS атак***

Затверджену наказом ректора НТУ «Дніпровська політехніка» від 22.10.20 № 888-С

Розділ	**Зміст**	**Термін виконання**
1.	Проаналізовано ситуацію з DOS атаками	
2	Сворена методика захисту ІТС від DOS атак	
3.	Розрахована вартість захисту за цією методикою.	

Завдання видано _____ *Гусєв О.Ю*
(підпис керівника) (прізвище, ініціали)

Дата видачі завдання: 01.09.20

Дата подання до екзаменаційної комісії: 09.12.20

Прийнято до виконання _____ *Харитонов І.А.*
(підпис студента) (прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 60 с., 8 рис., 2 табл., 4 додатки, 22 джерела.

Об’єкт дослідження: Інформаційно-телекомунікаційні системи комерційних підпрємств.

Мета роботи: Створення методики захисту від DOSатак, яка використовувала для аналізу нейромережу.

Методи розробки:спостереження, порівняння, аналіз, опис, пошук, адаптація.

У першому розділі було проаналізовано стан на 2019 - 2020 роки з DOSатаками, методами цих атак, види ботів. Було проведено аналіз знайдених вразливостей, та засоби їх нейтралізувати.

У спеціальній частині був проведений аналіз мереж комерційних підприємств, їх основні типи, переваги та недоліки, був проведений аналіз найбільш популярних DOS атак, а також аналіз сучасних засобів захисту. Була створена нейромережа для аналізу трафіка, та методика.

У економічній частині були розраховані витрати на впровадження даної методики на підприємство, розробку та впровадження нейромережі, розраховані поточні витрати на обслуговування системи, Оцінено можливий збиток від DOSатаки, загальний ефект від впровадження системи інформаційної безпеки, були визначені та проаналізовані показники економічної ефективності системи інформаційної безпеки.

Практичне значення роботи полягає у зменшені часу реагування на DOS атаку.

Наукова новизна роботи полягає у створені нейромережі для швидкісного аналізу трафіку.

НЕЙРОМЕРЕЖА, МЕТОДИКА ЗАХИТСУ ВІД DOS АТАК, ВИДИ БОТІВ, ВИДИ АТАК.

РЕФЕРАТ

Explanatory note: 60 pp., 8 figs., 2 tables., 4 appendices, 22 source.

Object of research: Information and telecommunication systems of commercial enterprises

Purpose: To create a method of protection against DOS attacks, which was used to analyze the neural network.

Development methods: observation, comparison, analysis, description, search, adaptation.

The first section analyzed the situation for 2019 - 2020 with DOS attacks, methods of these attacks, types of bots. The found vulnerabilities were analyzed and the means to neutralize them.

In the special part the analysis of networks of the commercial enterprises, their basic types, advantages and lacks, the analysis of the most popular DOS attacks, and also the analysis of modern means of protection was carried out. A neural network for traffic analysis and methodology was created.

In the economic part, the costs for the implementation of this technique were calculated for the enterprise, development and implementation of the neural network, calculated current costs for system maintenance, estimated possible damage from DOS attack, the overall effect of the information security system, identified and analyzed .

The practical value of the work is to reduce the response time to a DOS attack.

The scientific novelty of the work lies in the creation of neural networks for high-speed traffic analysis.

NEURAL NETWORK, METHOD OF PROTECTION AGAINST DOS ATTACKS, TYPES OF BOTS, TYPES OF ATTACKS.

РЕФЕРТ

Пояснительная записка: 60 с., 8 рис., 4 приложения, 22 источника

Объект исследования: Информационо-телекомуникацоные системы комерческих предприятий.

Цель работы: Создание методики защиты от DOS атак, которая использовала для анализа нейросеть.

Методы разработки: наблюдение, сравнение, анализ, описание, поиск, адаптация.

В первом разделе было проанализировано состояние на 2019 - 2020 годы с DOS атаками, методами этих атак, виды ботов. Был проведен анализ найденных уязвимостей, и средства для их нейтрализации.

В специальной части был проведен анализ сетей коммерческих предприятий, их основные типы, преимущества и недостатки, был проведен анализ наиболее популярных DOS атак, а также анализ современных средств защиты. Была создана нейросеть для анализа трафика, и методика.

В экономической части были рассчитаны затраты на внедрение данной методики на предприятие, разработку и внедрение нейросети, рассчитанные текущие расходы на обслуживание системы, Оценено возможный ущерб от DOS атаки, общий эффект от внедрения системы информационной безопасности, были определены и проанализированы показатели экономической эффективности системы информационной безопасности.

Практическое значение работы состоит в уменьшении времени реагирования на DOS атаку.

Научная новизна работы заключается в создании нейросети для скоростного анализа трафика.

Нейросети, Методика ЗАХИТСУ ОТ DOS АТАК, ВИДЫ ботов, ВИДЫ АТАК.

СПИСОК УМОВНИХ СКОРОЧЕНЬ

1.DOS–розподілена атака на відмову в обслуговуванні

2.ПК – персональний комп’ютер

3.DNS – система додаткових імен

4.UDP – протоколдатаграм корисувача

5.IP – ідентифікатор мережевого рівня

6.VPN- віртуальна приватна мережа

7.WAF – фаэрвол веб додатків

8.CDN – географічно розподілена мережа

9. ACL – список управління доступом.

10.ПЗ – програмне забезпечення.

ЗМІСТ

ВСТУП

У 2020 році DOS атаки залишаються дуже актуальними, постійно розроблюються нові та вдосконалюються старі методи DOS атак, так з'явилися нові ботнети що експлуатують нещодавно знайдені вразливості як самих серверів так и засобів захисту. Наприклад ботнети MiraitaLucifer. А з появою нейромереж багато старих засобів захисту від ботнетов стали неефективними, у зв'язку з тим що нейромережу легко обучити їх обходити. Окрім того з вводом карантину багато комерційних підприємств перейшли на дистанційну роботу, що зробило їх більш вразливими перед DOS атакою. Також були знайдені нові вразливості у HTTPRange а також у механізмі делегування DNScерверов. Таким чином більшість комерційних підприємств залишаються дуже вразливими перед DOS атаками. Виходячи з цього, основною метою цього проекту є розробка методики захисту ІТС комерційних підприємств від DOS атак, яка підійшла би малим, середнім, та великим підприємствам та компаніям.

Актуальність вибраної теми дипломної роботи зумовлена тим що кількість DOS атак с кожним роком збільшується. При цьому навіть крупні компанії та компанії що надають послуги захисту від DOS атак зазнають шкодив від DOS нападів. Ця методика направлена на захист підприємств від сучасних способів DOS атак.

1 СТАН ПИТАННЯ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Стан питання

З 2019 року з'явилась велика кількість нових ботнетів:

- Cayosin-продається через "законні" платформи соціальних медіа, а не через Dark Web. Одним з перших маркетингових інструментів було відео на YouTube, яке демонструє його роботу. Обліковий запис користувача "Instagram", який називається "unholdable", містить кілька статей та відео, які пояснюють, як орендувати місце в ботнеті Cayosin, як найкраще використовувати шкідливе програмне забезпечення та як придбати вихідний код для оригінальної версії програмного забезпечення ботнету. Дослідження в соціальних мережах привели дослідників до додаткового шкідливого програмного забезпечення та ботнетів, включаючи Yowai, ботнет, описаний дослідниками Trend Micro. А облікові записи в соціальних мережах дозволяють розробнику Cayosin брати участь у дослідженні ринку та підтримці клієнтів у комерційних масштабах. Кайозін розвивається як у своїй здатності заражати нові системи, так і корисному навантаженні, яке він може розподілити. Хоча Cayosin в основному використовувався для запуску розподілених атак відмови в обслуговуванні (DOS), змінюється корисне навантаження показує, що починає сприймати дії як інструмент для розповсюдження конфіденційної інформації, крадіжки облікових даних та інших видів діяльності, які можуть мати більший економічний ефект вплив, ніж простий DOS.[3]
- Roboto-Ботнет Roboto атакует Linux-серверы через известную уязвимость в приложении для удаленного администрирования Webmin, яка отримала назву CVE-2019-15107, щоб скинути модуль завантажувача на сервери Linux, що працюють під вразливими установками інструменту адміністрування.

Модуль DOS підтримує чотири типи методів DOS-атак - ICMP Flood, HTTP Flood, TCP Flood та UDP Flood - залежно від системних дозволів. Схема праці бота наведена на рисунку 1.1.[11]

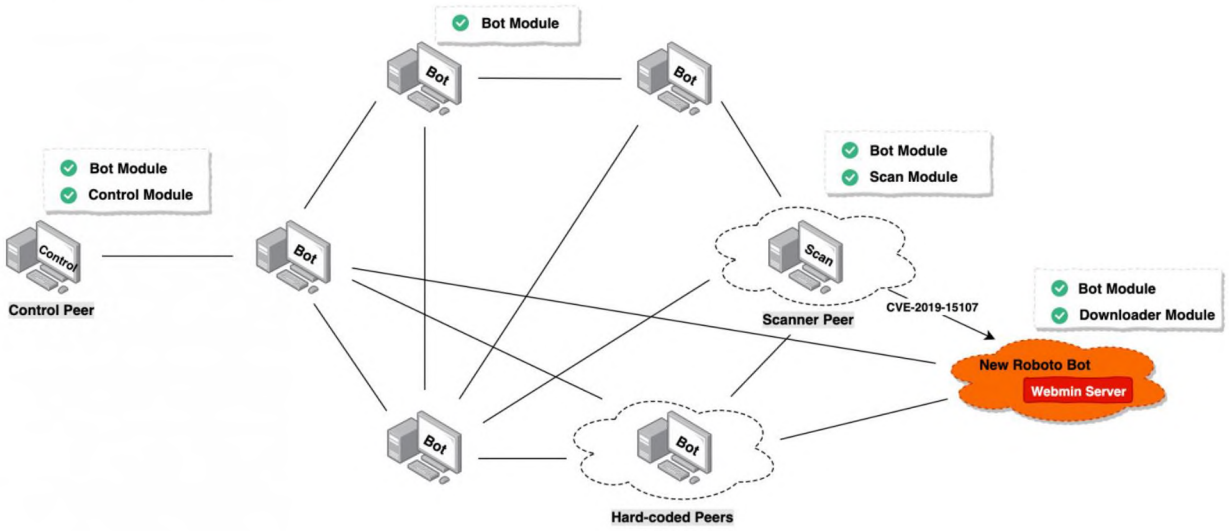

Рис.1.1

- Mozi- постійно розглядає маршутизатори: Netgear, D-Link та Huawei, на предмет слабких паролів Telnet, та беруться їм під конроль. Цей ботнет пов'язаний зі шкідливим програмним забезпеченням Gafgyf, так як повторно використовує часнину свого коду. Ботнет реалізований за допомогою спеціального розширеного протоколу розподіленої хеш-таблиці (DHT), заснованого на стандартному, який зазвичай використовується торрент-клієнтами та іншими платформами P2P для зберігання контактної інформації вузла. Шкідливе програмне забезпечення використовує telnet і експлуатує для розповсюдження на нові вразливі пристрої, увійшовши до будь-якого цільового маршрутизатора або відеореєстратора відеоспостереження, який постачається із слабким паролем, скидаючи та виконуючи корисне навантаження після успішного використання непрацюючих хостів.

  Після того, як шкідливе програмне забезпечення буде завантажено на зламаний пристрій, щойно активований бот автоматично приєднається до мережі Mozi P2P як новий вузол. На наступному етапі зараження нові бот-вузли отримують і виконують команди від майстра ботнетів, а також

здійснюють пошук та зараження інших вразливих маршрутизаторів Netgear, D-Link та Huawei для додавання до ботнету. Щоб переконатись, що їх ботнет не зайнятий іншими акторами загроз, оператори Mozi налаштовують його на автоматичну перевірку всіх команд та синхронізованих конфігурацій, що надсилаються на вузли ботнету, при цьому приймаються та виконуються лише ті, що проходять ці вбудовані перевірки за вузлами. Схема роботи бота наведена на рисунку 1.2.[12]

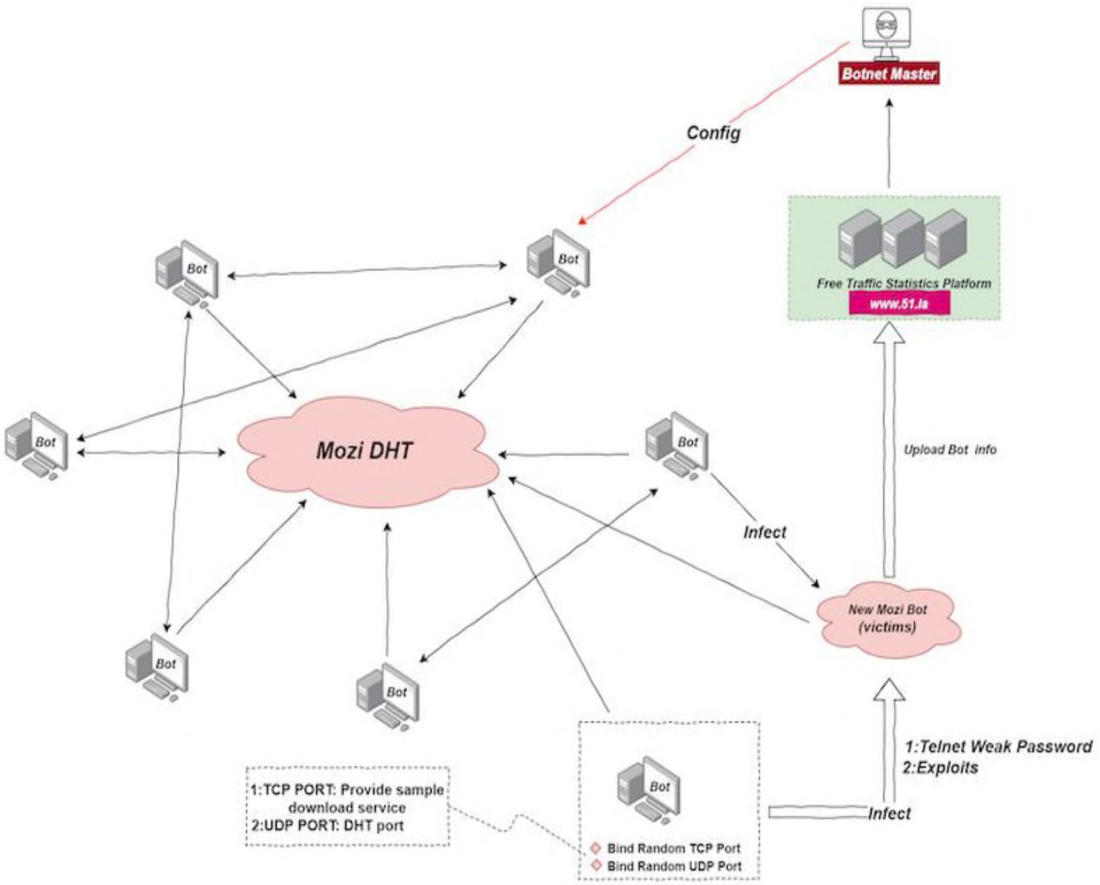

Рис1..2

- Lucifer- програмне забезпечення для криптоджекінгу з численних випадків експлуатації CVE-2019-9081. При детальному розгляді виявилося, що шкідливе програмне забезпечення, здатне проводити DOS-атаки та добре оснащене різними експлойтами проти вразливих хостів Windows.Люцифер досить потужний за своїми можливостями. Він не тільки здатний скинути XMRig для крипто-розкрадання Monero, він також здатний керувати та керувати (C2) роботою та саморозповсюдженням через використання

безлічі вразливих місць та примушування грубих даних. Крім того, він скидає та запускає EternalBlue, EternalRomance та DoublePulsar задні двері проти вразливих цілей для інтранет-інфекцій.Вичерпний перелік озброєних експлойтів включає CVE-2014-6287, CVE-2018-1000861, CVE-2017-10271, вразливості ThinkPHP RCE (CVE-2018-20062), CVE-2018-7600, CVE-2017-9791, CVE- 2019-9081, PHPStudy Backdoor RCE, CVE-2017-0144, CVE-2017-0145 та CVE-2017-8464. Ці вразливі місця мають або «високий», або «критичний» рейтинг через їх тривіальний характер для використання та величезний вплив на жертву. Після використання зловмисник може виконувати довільні команди на вразливому пристрої. У цьому випадку цілями є хости Windows як в Інтернеті, так і в інтрамережі, враховуючи, що зловмисник використовує утиліту certutil у корисному навантаженні для розповсюдження шкідливого програмного забезпечення. На щастя, патчі для цих вразливих місць легко доступні.[8]

А також отримали оновлення старі ботнети. Так ботнет Miraiотримав декілька оновлень завдяки чому він тепер заражає не тільки точки доступу, маршутизатори, мережеві камери, но і безпроводні системи презентацій та рекламні панелі.[5] Зявився ще один варіант ботнету, що експлуатує баг CVE-2020-10173 в роутерах Comtrend VR-3033, який дозволяє скомпрометувати ділянку мережі, підключений до уразливому роутера. У серпні стало відомо про варіант Mirai, атакуючому продукти сімейства BIG IP через уразливість CVE-2020-5902. У сімейство BIG IP входять брандмауери, додатки для управління навантаженням і контролю доступу, системи захисту від шахрайства і ботнетів. Уразливість дозволяє виконувати довільні команди, завантажувати і видаляти файли, відключати сервіси і запускати скрипти JavaScript. А ботнет Gafgyt, також відомий як Bashlite, отримав оновлення шкідливого програмного забезпечення та спрямоввує його на уразливості в трьох моделях бездротових маршрутизаторів. Huawei HG532 і Realtek RTL81XX були орієнтовані на попередні версії Gafgyt, але зараз він також націлений на Zyxel P660HN-T1A.[6]

Були створенні нові методиDOSатак:

Так були зафіксовани атаки, при яких посилення забезпечувалося за рахунок спуфинга зворотного IP-адреси через протокол многоадресной передачі WS-Discovery. Та дуже швидко змогли достигнути потужності атаки до 350 Гбіт/с. Протокол WSD має обмежену сферу застосування і в принципі не призначений для зв'язку машин з інтернетом; з його допомогою пристрою повинні знаходити один одного в LAN-мережах. Однак WSD досить часто використовується не цілком за призначенням в самому різному обладнанні, від IP-камер до мережевих принтерів. По даним таким чином до інтернету підключено більше 630 тис. таких пристроїв. Також було зафіксоване нова корисне навантаження, поширювана через бекдор в засобі пошуку та аналітики даних Elasticsearch. Зловредів небезпечний тим, що застосовує багатоступінчастий підхід до зараження, успішно уникає виявлення і може використовуватися для формування ботнетів з подальшим запуском масштабних DOS-атак. Trend Micro радить всім, хто користується Elasticsearch, оновити систему до останньої версії, оскільки патч проти бекдора вже випущений. Слідом за цим кіберзлочинці звернулися до служби Apple Remote Management Service (ARMS), яка входить до складу програми для віддаленого адміністрування Apple Remote Desktop (ARD). Перші атаки з використанням ARMS були зафіксовані ще в червні 2019 року, проте до початку жовтня протокол потрапив в арсенал платформ, які надають DOS як послугу, і такі атаки стали масовими. За даними порталу binaryedge.io, на початку кварталу майже 40 тис. Систем під управлінням macOS з включеною службою ARMS були доступні онлайн. Також дослідники спостерігали хвилю атак з відображенням TCP-трафіку. В основі цього методу лежить відправка запитів легітимним сервісів від імені жертви, в результаті чого її завалює Вами відповідями, а IP-адреси зловмисників не "світяться".

А у 2020 році з’явились ще 2 метода NXNSAttack и RangeAmp.

NXNSAttack - використовує спосіб роботи рекурсивних вирішувачів DNS при отриманні відповіді реферaлу NS, що містить сервери імен, але без відповідних

IP-адрес (тобто відсутні клеючі записи). Кількість повідомлень DNS, якими обмінюються в типовому процесі розв'язання, може бути набагато вищою на практиці, ніж очікується в теорії, головним чином за рахунок проактивного дозволу IP-адрес серверів імен. Ця неефективність стає вузьким місцем і може бути використана для здійснення нищівної атаки проти одного або обох рекурсивних засобів розв'язання та авторитетних серверів. NXNSAttack є більш ефективним, ніж атака NXDomain: i) Він досягає коефіцієнта посилення більше 1620x щодо кількості пакетів, що обмінюються рекурсивним розподільником. ii) Окрім негативного кешу, атака також насичує кеші роздільної здатності 'NS'.[4]

RangeAmp, це нова техніка відмови в обслуговуванні (DoS) використовує неправильні реалізації атрибута HTTP "Range Requests". Запити на діапазон HTTP є частиною стандарту HTTP і дозволяють клієнтам (як правило, браузерам) вимагати від сервера лише певну частину (діапазон) файлу. Функція була створена для призупинення та відновлення трафіку в контрольованих (дії паузи / відновлення) або неконтрольованих (перевантаження мережі або відключення) ситуаціях. Стандарт HTTP Range Request (Запити діапазону HTTP) обговорюється в Інженерній робочій групі (IETF) більше півроку десятиліть, але завдяки своїй корисності він уже впроваджений браузерами, серверами та CDN. існують дві різні атаки RangeAmp.

Перший називається атакою RangeAmp Small Byte Range (SBR). У цьому випадку [див. (A) на рисунку 3], зловмисник надсилає неправильний запит діапазону HTTP провайдеру CDN, який посилює трафік до сервера призначення, врешті-решт збіяючи цільовий сайт.

Другий називається атакою RangeAmp Overlapping Byte Ranges (OBR). У цьому випадку [див. Б) на рисунку 3] зловмисник надсилає неправильний запит діапазону HTTP постачальнику CDN, і в цьому випадку трафік спрямовується

через інші сервери CDN, трафік посилюється всередині мереж CDN, збій CDN-сервери, що робить CDN і багато інших сайтів призначення недоступними.

Протестувавши атаки RangeAmp проти 13 постачальників CDN і виявили, що всі вони були вразливі до атаки RangeAmp SBR, а шість також були вразливі до варіанту OBR при використанні в певних комбінаціях. DOSлідники заявили, що напади були дуже небезпечними і вимагали мінімум ресурсів. З двох, атаки RangeAmp SBR можуть найбільше посилити трафік.

DOSлідницька група виявила, що зловмисники можуть використовувати атаку RangeAmp SBR, щоб збільшити трафік від 724 до 43330 разів від звичайного.[9]

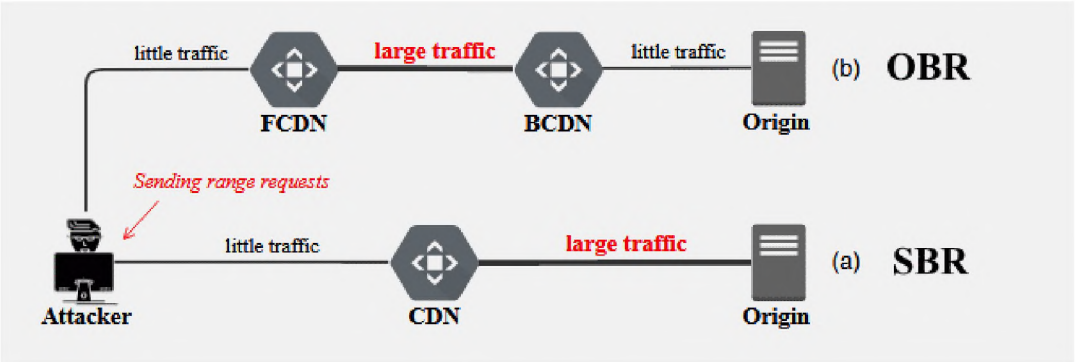

Рис.1.3

За даними лабароторії Касперського, з 2019 роком до 2020 кількість DOSатак збільшилась на 174%, а кількість розумних атак збільшилась на 258%. Найбільш популярним методом атак залишається флуд(SYN)(рисунок1. 4), а кількість ботнетов була більша на ОС Linux ніж на ОС Windows[1]. 95% на ОС Linux та 5% на ОС Windows рисунок 1.5, таблиця 1.1.

А найбільш популярна тривалість атак, атаки тривалістю від 30 до 60 хвилин и скали 83% від усіх атак.

Таблиця 1.1

Ботнети	Windows	Linux
2020	4.61%	95.39%

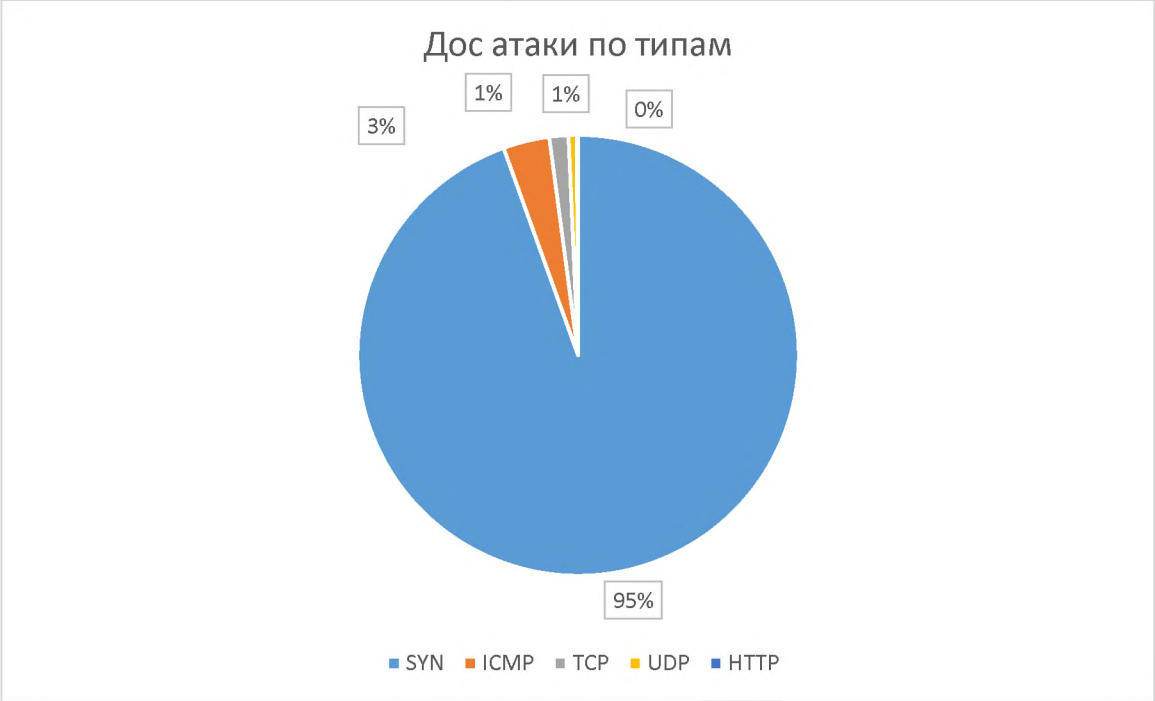

Рис.1.4

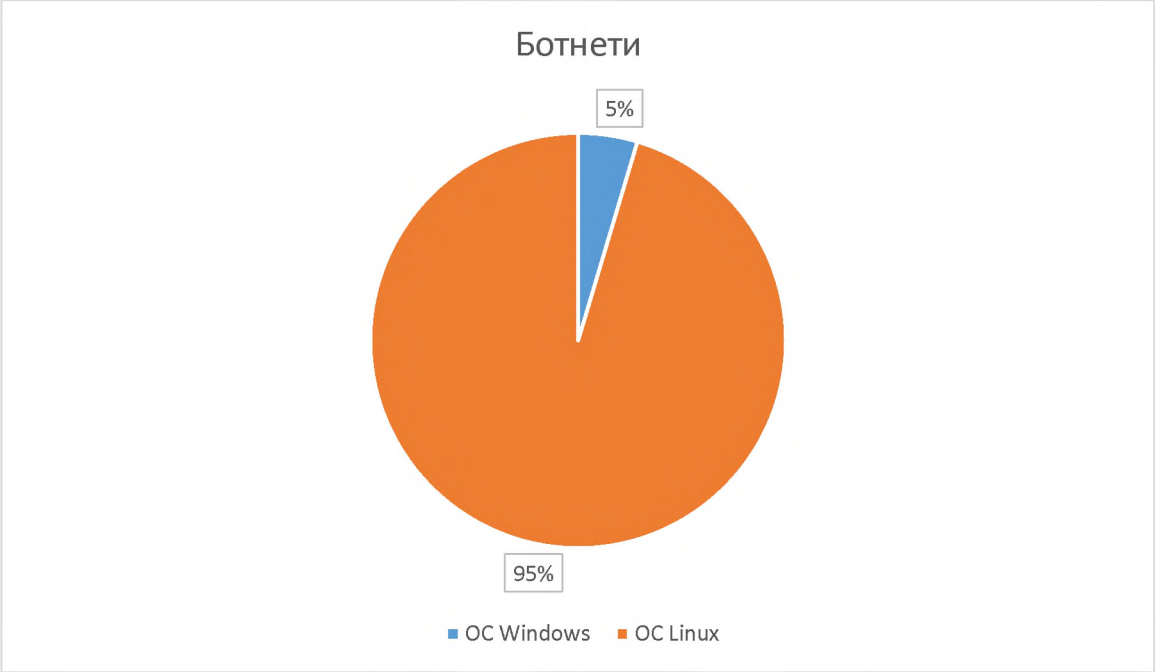

Рис1.5

А з появою епідемії короновірусу у світі багато підприємств перейшли до дистанційної форми роботи, таким чином став більш вразливими до DOSатак. Окрім того карантинні заходи призвели до зменшення доходів підприємств.

Цілью даної кваліфікаційної роботи є створення методики захисту від DOSатак, яка б використовувала нейромережу для аналізу трафіку.

1.2 У даної кваліфікаційної роботи вирішуються наступні задачі

- Аналіз мереж комерційних підприємств.
- Виявити найбільш поширені типи DOS атак.
- Дослідити сучасні засоби захисту.
- Розробити методику захисту від DOS атак.
- Створити нейромережу для швидкого аналізу трафіку.

1.3 Висновки

3 2019 року кількість атак значно збільшилась, були знайдені нові вразливості та створені більш ефективні ботнети. Окрім того з почтаку 2020 року експерти спостерігають аномальну кількість DOS атак. Швидше за все, це пов'язано з пандемією коронавируса і обмежувальними заходами, які в багатьох країнах тривали частина кварталу або на всій його довжині. Вимушене переміщення життя в інтернет збільшило кількість можливих мішеней для DOS. В іншому в другому кварталі мало що змінилося: склад TOP 10 за кількістю атак і мішеней практично той же, як і розподіл атак по тривалості. Помітно впала частка всіх видів DOS, крім SYN- і ICMP-флуду, однак говорити про яку-небудь тенденції в зв'язку з цим, ще рано.

2 СПЕЦІАЛЬНА ЧАСТИНА

2.1 Мережі комерційних підприємств:

Існує 3 типи мереж комерційних підприємств:

1. Однорангова мережа.

2.Мережа з виделиним сервером.

3. Гібридна мережа

В однорангових мережах всі комп'ютери рівні в можливостях доступу до ресурсів один одного. Кожен користувач може за своїм бажанням оголосити будь-якої ресурс свого комп'ютера розділяються, після чого інші користувачі можуть з ним працювати. У однорангових мережах на всіх комп'ютерах встановлюється така операційна система, яка надає всім комп'ютерам в мережі потенційно рівні можливості. Мережеві операційні системи такого типу називаються однорангових ОС. Очевидно, що однорангові ОС повинні включати як серверні, так і клієнтські компоненти мережевих служб.

У той же час адміністратор може закріпити за деякими комп'ютерами мережі тільки функції, пов'язані з обслуговуванням запитів від інших комп'ютерів, перетворивши їх таким чином в "чисті" сервери, за якими користувачі не працюють. У такій конфігурації однорангові мережі стають схожими на мережі з виділеними серверами, але це тільки зовнішня схожість - між цими двома типами мереж залишається істотна відмінність.

В одноронгових мережах відсутній спеціалізація ОС в залежності від того, яку роль відіграє комп'ютер - клієнта або сервера. Зміна ролі комп'ютера в одноранговій мережі досягається за рахунок того, що функції серверної або клієнтської частин просто не використовуються.

У мережах з виділеними серверами використовуються спеціальні варіанти мережевих операційних систем, які оптимізовані для роботи в ролі серверів і

називаються серверними ОС. Комп'ютери користувачів в таких мережах працюють під управлінням клієнтських ОС.

Гібрідні мережі об’єднують однорангові та мережі с виделиними сервирами.

Та поєднують у собі найкращі сторони обох попередніх типів. Гібрідні мережі найбільш розповсюджений тип локальних мереж комерційних підприємств.

Локальна мережа підприємства з’єднана з мережею інтернет за допомогою мережі доступу.

За топологією мережі поділяються на:

1. Топологія зірка.

2. Топологія кільце.

3. Топологія шина

4. Топологія дерево

Топологія зірка – топологія мережі з явно виділеним центром, до якого підключаються усі інші комп’ютери у мережі. Обмін інформацією йде через винятково через центральний комп’ютер, на який лягає більшість навантаження, тому нічим іншим крім мережі він займатися не може. Мережеве устаткування повинно бути складнішим ніж устаткування периферійних комп’ютерів. Центральний комп’ютер найпотужніший, бо на нього покладають всі функції по керуванню обмінів. Якщо казати про стійкість мережі то вихід з ладу периферійного комп’ютеру або його мережевого устаткування на функціюванні мережі не відобразиться, проте будь-яка відмова центрального комп’ютера робить мережу не працездатною.

Пропускна здатність мережі визначається потужністю вузлового комп’ютера та гарантується для кожної робочої станції. Топологія зірка є найбільш швидкодіючою з усіх топологій. Центральний вузол реалізує оптимальний

механізм захисту проти несанкційованого доступу до інформації. Уся мережа може управлятися з її центру.

Топологія кільце – робочі станції пов'язані одна з іншою по колу. Основна проблема при кільцевій топології в тому, що при виході з ладу хоча б однієї робочої станції уся мереже перестає працювати.

Топологія шина – середовище передачі інформації представлено у вигляді комунікаційного шляху до якого мають бути підключені робочі станції. Усі робочі станції можуть вступати в контакт одна за одним. Робочі станції можуть бути у будь який час підключені або відключенні від мережі.

Топологія дерево – комбінація вищеназваних топологій. Основа мережі розташовується в точці де збираються комунікаційні лінії інформації.

На сучасних підприємствах використовується Гібридна мережа з топологією дерева. Де до персональних станцій підключено мережу інтернет. Залежно від виду підприємства можуть бути сервери також підключенні до мережі інтернет. У таких мережах найбільш вразливою крапкою є корінь (сервер). Бо на ньому зазвичай храняться необхідні документи, та програми. Таким чином кіберзлочинці намаються вивсети з ладу саме його.

Окрім того є необхідність моніторити не тільки сам сервер, а й вузли що з'єднують локальні мережі відділів з сервером. У відділах підприємств зазвичай використовують однорангові мережі, в яких один із комп'ютерів виконує роль локального сервера. На підприємствах до локальної мережі зазвичай підключена оргтехніка до якої може отримати доступ будь який комп'ютер у локальній мережі відділу.

2.2Найбільш розповсюджені типи атак:

Найбільшу небезпеку являють наступні типи DOS-атак:

- DNSREFLECTEDAMPLIFICATION
- GENERATEDUDP FLOOD

- HTTP GET/POST FLOOD
- HIT-AND-RUN
- SYN FLOOD
- SLOWLORIS

2.2.1 DNS REFLECTED AMPLIFICATION:

Ця DOS-атака являє собою об'ємну розподілену атаку відмови в обслуговуванні (DOS) на основі відображення, в якій зловмисник використовує функціональність відкритих DNS-розв'язувачів, щоб перевантажити цільовий сервер або мережу посиленим обсягом трафіку, відтворюючи сервер і роблячи навколишню інфраструктуру недоступною.

Усі атаки посилення використовують нерівність у споживанні смуги пропускання між зловмисником та цільовим веб-ресурсом. Коли розбіжність у вартості збільшується для багатьох запитів, отриманий обсяг трафіку може порушити мережеву інфраструктуру. Надсилаючи невеликі запити, що призводять до великих відповідей, зловмисний користувач може отримати більше, ніж менше. Помножуючи це збільшення на те, щоб кожен бот у бот-мережі робив подібні запити, зловмисник заважає виявленню та отримує переваги значно збільшеного трафіку атак.

Окремого бота в атаці посилення DNS можна розглядати в контексті зловмисного підлітка, який зателефонував до ресторана і сказав: "У мене буде все, будь ласка, передзвони мені і розкажи мені все моє замовлення". Коли ресторан запитує номер зворотного дзвінка, вказаний номер є номером телефону цільової жертви. Потім ціль отримує дзвінок із ресторану з великою кількістю інформації, яку вони не запитували.

В результаті кожного бота, що надсилає запити на відкриття DNS-розробників із підробленою IP-адресою, яка була змінена на справжню IP-адресу вихідної адреси цільової жертви, ціль отримує відповідь від DNS-вирішувачів. Для того, щоб створити великий обсяг трафіку, зловмисник структурує запит таким чином, що

генерує якомога більшу відповідь від вирішувачів DNS. В результаті ціль отримує посилення початкового трафіку зловмисника, і їх мережа засмічується помилковим трафіком, що спричиняє відмову в обслуговуванні. Схема цього типу атак на рисунку 6.

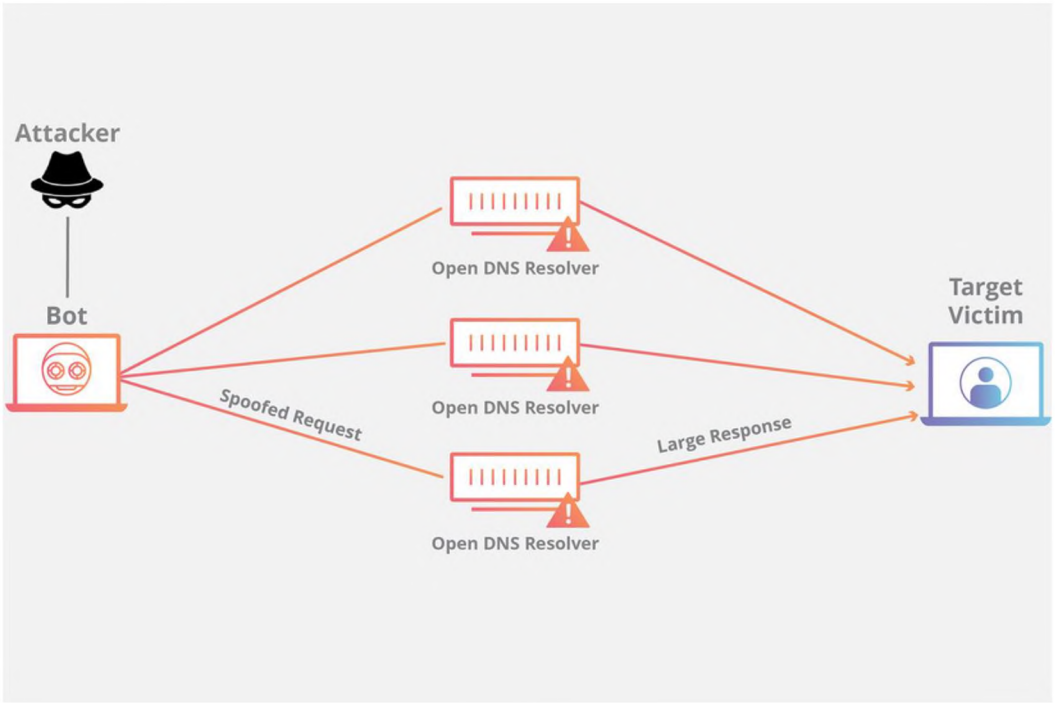

Рис.2.6

Посилення DNS можна розбити на чотири етапи:

1.Зловмисник використовує скомпрометовану кінцеву точку для надсилання пакетів UDP із підробленими IP-адресами до рекурсора DNS. Підроблена адреса на пакетах вказує на реальну IP-адресу жертви.

2.Кожен з UDP-пакетів робить запит до вирішувача DNS, часто передаючи такий аргумент, як „БУДЬ-ЯКИЙ”, щоб отримати якомога більшу відповідь.

3.Отримавши запити, вирішувач DNS, який намагається допомогти, відповідаючи, надсилає велику відповідь на підроблену IP-адресу.

4.IP-адреса цілі отримує відповідь, а навколишня мережева інфраструктура переповнюється потоком трафіку, що призводить до відмови в обслуговуванні.

5.Хоча кількох запитів недостатньо, щоб зруйнувати мережеву інфраструктуру, коли ця послідовність множиться між кількома запитами та вирішувачами DNS, посилення даних, які отримує ціль, може бути значним. 2.2.2 GENERATED UDP FLOOD

Атака потоку UDP - це тип атаки відмови в обслуговуванні. Подібно до інших поширених атак повеней, напр. ping-повінь, HTTP-повінь і SYN-флуд, зловмисник надсилає велику кількість підроблених пакетів даних до цільової системи. Мета полягає в тому, щоб перевантажити ціль до такої міри, що вона більше не може відповідати на законні запити. Як тільки ця точка досягнута, послуга зупиняється.

Поток UDP - це об'ємна атака DoS. Подібно до пінг-повені, ідея полягає в тому, щоб завалити цільову систему великим обсягом вхідних даних. Таким чином, повінь UDP відрізняється від пінгу смерті, який збиває цільову систему, використовуючи помилку пам'яті, і від потоку SYN, який зв'язує ресурси на сервері. Одне спільне між усіма згаданими раніше атаками DoS полягає в тому, що вони мають на меті перевантажити ціль і тим самим заперечити її законне використання.

Повеня UDP стала предметом суспільного інтересу внаслідок деяких вражаючих хакерських атак на міжнародні організації. Окрім Церкви Саєнтології, атакуються компанії, що займаються засобами масової інформації та фінансовим сектором. Цілеспрямовані веб-сайти та послуги зазнали краху внаслідок потоку даних, які часом були недоступні для своїх користувачів годинами. Під час цих атак потужний інструмент, який називається Іонна гармата низької орбіти (LOIC), використовувався як зброя для розв'язання потоку UDP.

Атака UDP-потоку залежить від особливостей протоколів користувальницьких датаграм (UDP), що використовуються в атаці. Якщо на сервер надходить пакет UDP, операційна система перевіряє вказаний порт для прослуховування програм. Якщо жодної програми не знайдено, сервер повинен повідомити відправника.

Оскільки UDP є безпровідним протоколом, сервер використовує протокол керування повідомленнями Інтернету (ICMP), щоб повідомити відправника про те, що пакет не може бути доставлений.

У разі нападу потоку UDP відбувається такий процес:

Зловмисник надсилає UDP-пакети з підробленою адресою IP-відправника на випадкові порти цільової системи.

На системній стороні наступну процедуру необхідно повторити для кожного вхідного пакета.

Перевірте порт, зазначений у пакеті UDP, на наявність програми прослуховування; оскільки це випадково обраний порт, це, як правило, не так.

Надіслати пакет ICMP “недоступний для призначення” передбачуваному відправнику; оскільки IP-адреса була підроблена, ці пакети, як правило, отримує якийсь випадковий спостерігач. Схема UDP флуду зображена на рисунку 7.

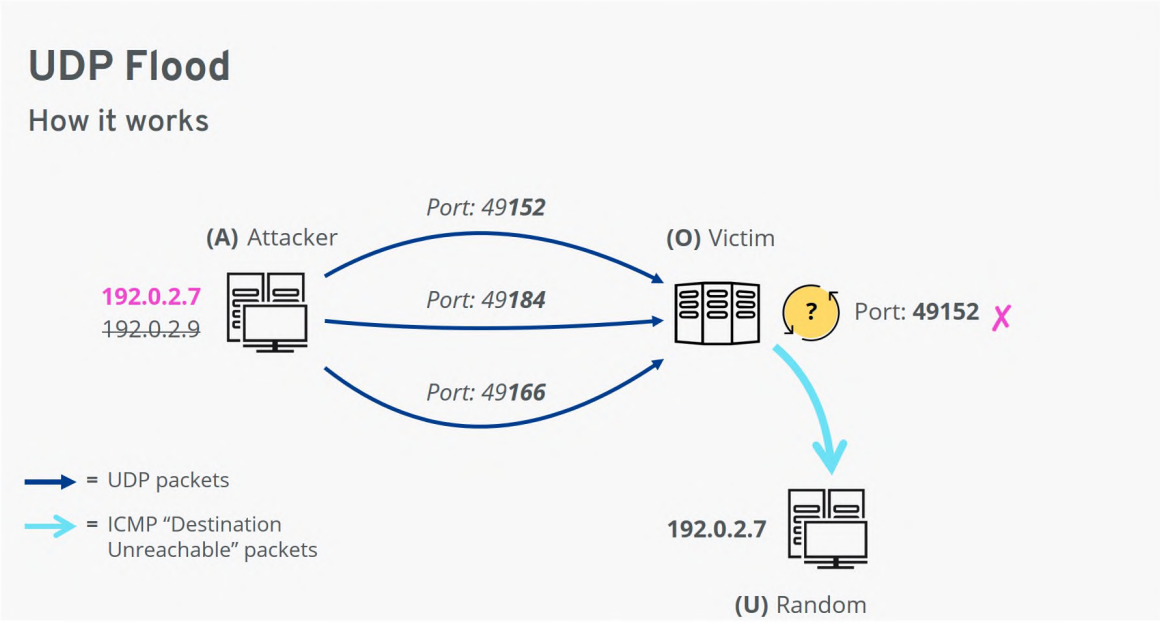

Рис.2.7

2.2.3 HTTP GET/POST FLOOD

атоплення HTTP - це тип розподіленої атаки на відмову в обслуговуванні (DOS), при якій зловмисник використовує легітимні HTTP-запити GET або POST для атаки веб-сервера або програми.

Атаки повені HTTP - це об'ємні атаки, які часто використовують бот-мережу "армія зомбі" - групу підключених до Інтернету комп'ютерів, кожна з яких була зловмисно захоплена, як правило, за допомогою шкідливих програм, таких як Троянські коні.

Складна атака рівня 7, повені HTTP не використовують неправильно сформовані пакети, підмінюючи або відбиваючи методи, і вимагають меншої пропускної здатності, ніж інші атаки, щоб збити цільовий сайт або сервер.

Таким чином, вони вимагають більш глибокого розуміння цільового сайту чи програми, і кожна атака повинна бути спеціально розроблена, щоб бути ефективною. Це значно ускладнює виявлення та блокування атак HTTP-потоків.

Коли HTTP-клієнт, такий як веб-браузер, «розмовляє» з додатком або сервером, він надсилає HTTP-запит - зазвичай один із двох типів запитів: GET або POST. Запит GET використовується для отримання стандартного статичного вмісту, такого як зображення, тоді як запити POST використовуються для доступу до динамічно генерованих ресурсів.

Атака найефективніша, коли змушує сервер або додаток розподіляти максимально можливі ресурси у відповідь на кожен окремий запит. Таким чином, зловмисник, як правило, прагне засипати сервер або додаток кількома запитами, кожен з яких є максимально інтенсивним для обробки.

З цієї причини атаки потоку HTTP із використанням запитів POST, як правило, є найбільш ефективними з точки зору зловмисника; оскільки запити POST можуть включати параметри, які запускають складну обробку на стороні сервера. З

іншого боку, атаки, засновані на HTTP GET, простіші у створенні і можуть ефективніше масштабуватися за сценарієм ботнету.

2.2.4 HIT-AND-RUN

Один з підвидів Volumetric-atak, но Hit-and-run працює особистим чином, відмінним від більшості інших атак. Це непродовжуюча загальна трафіка обсягом сотних гігабіт у секундах, тривалість 20–60 хвилин, а в ряді випадків - менше хвилин. Вони багатократно повторюються протягом тривалих періодів часу - днів і навіть неділі - з інтервалами в середньому 1–2 сутки.

Подібні атаки стали популярними з-за своєї дешевизни. Вони ефективно проти захисних рішень, що активуються вручну. Опасність Hit-and-run закладається в тому, що послідовна захист вимагає постійного моніторингу та готовності системної реабілітації.

2.2.5SYN FLOOD

Клієнт генерує SYN-пакет, запрошуючи нову сесію у сервера. Після того, як TCP сесія відкрита (алгоритм “триповерхового рукопожаття TCP” виконується), хост буде відслідковувати та обробляти каждую користувацьку сесію, коли вона не буде закрита. У часі SYN Flood атакуваний сервер з великою швидкістю отримує підручні SYN-запроси, що містять додаткові IP-адреси джерела. SYN-флуд уражає сервер, займаючи всю пам'ять таблиці з'єднань (таблиця блоку управління передачею (TCB)), зазвичай використовує для зберігання та обробки знайдених пакетів. Це виказує критичне падіння продуктивності та, як ітог, показ у роботі сервера.

Хоча SYN FLOOD застарів він залишається самим популярним типом DOS-атаки, тому його не треба недооцінювати.

2.2.6 SLOWLORIS

Slowloris - це тип інструменту відмови в обслуговуванні, який дозволяє одній машині зняти веб-сервер іншої машини з мінімальною пропускною здатністю та побічними ефектами на не пов'язані між собою служби та порти.

Slowloris намагається тримати багато з'єднань з цільовим веб-сервером відкритими і тримати їх відкритими якомога довше. Він досягає цього, відкриваючи з'єднання з цільовим веб-сервером і надсилаючи частковий запит. Періодично він надсилатиме наступні заголовки HTTP, додаючи, але ніколи не завершуючи, запит. Постраждалі сервери залишатимуть ці з'єднання відкритими, заповнюючи їхній максимально допустимий одночасний пул з'єднань, врешті-решт відмовляючи клієнтам у додаткових спробах з'єднання.

2.3Сучасні засоби захисту:

- Ще на етапі написання програмного забезпечення необхідно задуматися про безпеку сайту. Ретельно перевіряйте ПЗ на наявність помилок і вразливостей.
- Регулярно оновлюйте ПЗ, а також передбачте можливість повернутися до старої версії при виникненні проблем.
- Слідкуйте за обмеженням доступу. Служби, пов'язані з адмініструванням, повинні повністю закриватися від стороннього доступу. Захищайте адміністраторський аккаунт складними паролями і частіше їх міняйте. Своєчасно видаляйте акаунти співробітників, які звільнилися.
- DOStуп до інтерфейсу адміністратора повинен проводитися виключно з внутрішньої мережі або за допомогою VPN.
- Сканувати систему на наявність вразливостей.
- Застосовуйте брандмауер для додатків - WAF (Web Application Firewall). Він переглядає переданий трафік і стежить за легітимністю запитів.
- Використовуйте CDN (Content Delivery Network). Це мережа по доставці контента, що функціонує за допомогою розподіленої мережі. Трафік

сортується по декількох серверах, що знижує затримку при доступі відвідувачів.

- Контролюйте вхідний трафік за допомогою списків контролю доступу (ACL), де будуть вказаний список осіб, що мають доступ до об'єкта (програми, процесу чи файлу), а також їх ролі.
- Можна блокувати трафік, яких виходить від атакуючих пристроїв. Робиться це двома методами: застосування міжмережевих екранів або списків ACL. У першому випадку блокується конкретний потік, але при цьому екрани не можуть відокремити «позитивний» трафік від «негативного». А в другому - фільтруються другорядні протоколи. Тому він не принесе користі, якщо хакер застосовує першорядні запити.
- Щоб захиститися від DNS-спуфинга, потрібно періодично очищати кеш DNS.
- Використовувати захист від спам-ботів - капча (captcha), «людяні» тимчасові рамки на заповнення форм, reCaptcha (галочка «Я не робот») і т. Д.
- Зворотній атака. Весь шкідливий трафік перенаправляється на зловмисника. Він допоможе не тільки відбити напад, але і зруйнувати сервер атакуючого.
- Розміщення ресурсів на декількох незалежних серверах. При виході одного сервера з ладу, що залишилися забезпечать працездатність.
- Використання перевірених апаратних засобів захисту від DOS-атак. Наприклад, Impletec iCore або DefensePro.
- Вибирати хостинг-провайдера, який співпрацює з надійним постачальником послуг кібербезпеки. Серед критеріїв надійності фахівці виділяють: наявність гарантій якості, забезпечення захисту від максимально повного спектру загроз, цілодобова технічна, транспарентність (доступ клієнта до статистики і аналітики), а також відсутність тарифікації шкідливого трафіку.

- Переклад ресурсів в «хмару». Компанії, що спеціалізуються на хмарних технологіях і пропонують перенесення сервера в хмару, мають куди більш потужні засоби протидії хакерам, ніж підприємство малого чи середнього бізнесу.
- Нарощування обчислювальної потужності. При дійсно серйозною DOS-атаці цей метод може не спрацювати, але, якщо запитів не надто багато або атака знаходиться на самому початку, краще мати потужний сервер, який буде чинити опір як можна довше. Зрештою, за цей час можна буде застосувати інші заходи протидії.
- Спеціальне ПЗ. На ринку існує маса спеціальних пропозицій з протидії DOS-атакам. Має сенс вивчити їх усі, щоб вибрати найбільш підходящий варіант. Подібного роду ПЗ коштує недешево, але витрати окупляться при першій же критичній ситуації.
- Відведення активної IP-адреси або доменного імені від ресурсів, які можуть бути схильні до DOS-атакам.

2.4 Методика:

2.4.1Підготовка підприємства:

2.4.1.1. Необхідно встановити які ресурси та додатки є критично важними для підприємства, а які ні. Чи наявні такі ресурси у вашій мережі? Встановити з якою перервою ви готові миритись для різних інтернет ресурсів.[22]

2.4.1.2. Визначити який рівень захисту потрібен для кожного ресурсу. Для деяких будить досить магістральної захисту каналу і інфраструктурних сервісів, а десь знадобиться більш складний захист на рівні додатків, розшифровки трафіку і розбору логіки запитів-відповідей.[22]

2.4.1.3. Не варто тримати на одній ір адресі критично важливі додатки та все інше. Їх варто розділити критично важливі ресурси на одній адресі та фізичному носії, інші на іншому. Таким чином при атаці на один із носіїв ресурсів інший не постраждає.[22]

2.4.1.4. Визначити пропускну здатність мережі, наскільки її можливо збільшити у випадку атаки.[22]

2.4.1.5.Визначити межу працездатності інфраструктури при навантажені. Для цього необхідно провести тестування навантаження інфраструктури, щоб знати граничні значення трафіку, які може купірувати канал або обладнання. Тести допоможуть виявити вузькі місця, на які варто звернути увагу. А також виявити уразливості в додатках, які можуть привести до їх недоступності і зрозуміти, при яких значеннях (запитах в секунду) або обсязі запитів, продуктивність ресурсу починає деградувати.[22]

2.4.1.6. Назначити людей на наступні посади:

- Офіцер безпеки
- Мережевий інженер
- Відповідний за взаємовідносини з провайдером інтернету.
- Відповідний за CDN, якщо такий сервіс присутній на підприємстві.
- Відповідний за DNS.
- Власник додатку.

І найголовніше - необхідна одна людина, що приймає рішення в разі DOS-атаки, вона же - єдиний контакт для бізнес-підрозділів.

2.4.1.7. Створити інструкцію по реагуванню на DOS атаку.

Далі переходимо до превентивних заходів. Вони необхідні для того, щоб зменшити шкоду від атаки на мережу підприємства.

2.4.2 Превентивні заходи:

2.4.2.1. Необхідно встановити програму для моніторингу трафіку у мережі. Ця програма дозволе нам слідити за даними, що проходять у мережі, та виявити DOS атаку на початковый стадії. Таким чином у адміністратора мережі буде більше часу щоб зреагувати на атаку. Залежно від топології мережі та станцій підключених до мережі інтернет змінюється кількість станцій за якими потрібно

спостерігати. Так у топології зірка необхідно слідкувати за трафіком що йде через центральний комп'ютер мережі так, як весь трафік йде через нього, а підключення робочих станцій допоможе легко виявити через який комп'ютер йде атака. У топології кільце дуже важко виявити комп'ютер через який йде атака на мережу підприємства. Окрім того у випадку DOS атаки така мережа дуже вразлива перед нею. Тому цю топологію краще замінити на іншу. Найскладнішою для моніторингу є топологія шина бо там потрібно встановити моніторинг трафіка що проходить через усі комп'ютери що підключенні до мережі інтернет.

В топології дерево необхідно встановити моніторинг за основою та вузлами мережі.

Моніторинг мережі повинен бути цілодобовим. Краще за все буде для моніторинга окрім ручних програм для спостереження встановити нейромережу, що зробить моніторинг меш залежним від спостерігающої людини. Для цього нам потрібно навчити нейромережу розпізнавати DOSатаку та її види. Перше що зробимо це навчимо нейромережу це навчити розбирати та аналізувати лог програми моніторингу. Для цього нам знадобляться лог з «хорошими» запитами, лог з «поганими» запитами та лог який необхідно проаналізувати. Видилити з них маркери та додати їх у словник нейромережі. В логі таких маркерів небагато:

- Сам запит, попарсенний на тип запиту (HEAD / GET / POST / etc), url і http_version. Url Парс на протокол, ім'я хоста, шлях і всі ключі від query_string
- Referer, попарсенний аналогічно url в запиті.
- status, тільки в разі кодів 503/404/403. Взагалі, під час DOS'a сервер любить відповідати 500/502, тому враховувати будемо тільки вищеописані коди.
- User-Agent, попарсенний чарівної вуличної магією, бо його формат досить сильно варіюється від браузера до браузеру.

Це і є наш словник. Словник потрібен для того, щоб з будь-якого можливого запиту створити feature-vector. Бінарний (що складається з нулів і одиниць) M-

мірний вектор (де M - довжина словника), який відображає присутність кожної ознаки зі словника в запиті.

Дуже добре, якщо словник з точки зору структури даних буде hash-таблицею, бо до нього буде безліч звернень типу if word in dictionary.

Доброю практикою є поділ dataset'a на кілька частин. Краще за все буде розбити на 3 частини в співвідношенні 60/20/20: Training set, Test set і Cross validation.

2.4.2.2. Якщо в компанії наявні пристрої під'єднані за допомогою протоколуWSD то обов'язково заблокуйте порт 3702, на усіх пристроях що мають доступ до мережі інтернет. Таким чином можливо захиститись від ботнетів що використовують вразливість підключення через WS-Discovery.

2.4.2.3. Завжди оновлюйте програми при виході нових версій, не забувайте робити резервні копії старих версій при оновленні. При знаходжені вразливості розробники програми за наступного патчу майже завжди намагаються її закрити.[18]

2.4.2.4. Делегування прав на операції. Тобто забезпечити декілька рівнів прав. Своєчасно оновлювати список осіб, що мають доступ до сервера.

2.4.2.5. Зробіть кластери своїх серверів це збільшить опір системи DOS атакам.

Схема кластеру зображена на риснку 8.

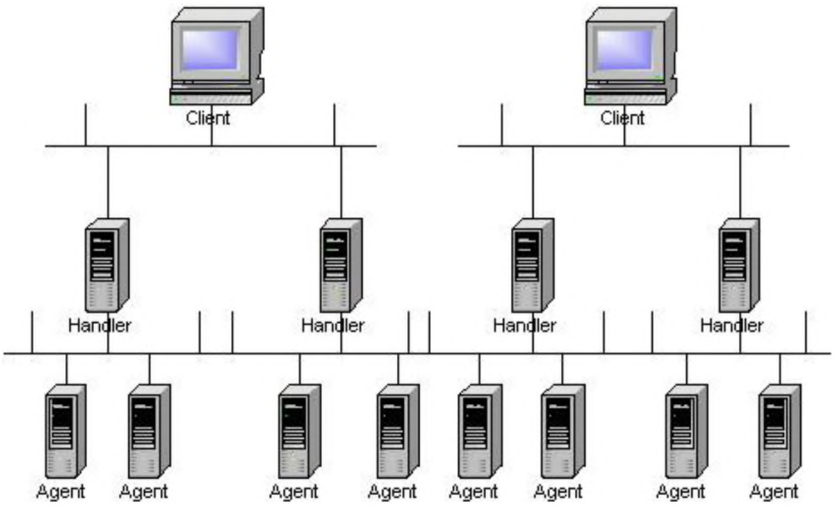

Рис.2.8

Таким чином можно виграти декілька годин на протидію атаки на відмову.

Коли почнеться DOS атаканеобхідно зробити дамп трафіка, далі проаналізувати його, використовуючи інструменти для аналізу. Таким чином

визначаємо переважаючий тип трафіку у мережі. Таким чином визначаємо тип DOSатаки. Якщо сервер не справляється з навантаженням то застосовуємо кластер який ми підготували раніше. Прописуємо один із серверів кластера як резервний, що дає більше часу на протидію.

2.4.2.6. Проводьте навчання співробітників, приблизно раз на 2 місяці, це зменшить час реагування під час реальної DOS атаки.[22]

2.4.3План дій при DOS атаці:

Якщо почалася DOS атака необхідно:

2.4.3.1. Переконатися в тому що це саме ддос атака, а ні загальні причини перебою роботи, включаючи неправильну конфігурацію DNS, проблеми з маршрутизацією і людський фактор.[19]

2.4.3.2. За допомогою технічних фахівців визначте, які ресурси піддалися атаці.

2.4.3.3.Встановіть пріоритети важливості додатків, для того, щоб зберегти найбільш пріоритетні. В умовах інтенсивної DOS-атаки і обмежених ресурсів необхідно зосередитися на додатках, що забезпечують основні джерела прибутку.[19]

2.4.3.4.Захистіть віддалених користувачів. Забезпечте роботу вашого бізнесу: занесіть в білий список IP-адреси довірених віддалених користувачів, яким необхідний доступ, і зробіть цей список основних. Розповсюдьте це список в мережі і відправте його постачальникам послуг доступу.

2.4.3.5. Визначте клас атаки. С яким типом атаки ви зіткнулися: Об'ємна? Малопотужна і повільна?[20]

2.4.3.6.Оцініть варіанти боротьби з адресами джерел атак. Заблокуйте невеликі списки атакуючих IP-адрес у вашому межсетевом екрані. Більші атаки можна блокувати на основі даних про геопозиционирования.

2.4.3.7. Заблокуйте атаки на рівні додатку. Визначте шкідливий трафік і перевірте, чи утворюється він відомим інструментом. Певні атаки на рівні додатку можна блокувати для кожного конкретного випадку за допомогою контрзаходів.[19]

2.4.3.8. Підсильте свій периметр захисту. Можливо, ви зіткнулися з асиметричною атакою DOS 7 рівня. Зосередьтеся на захист на рівні додатків: використовуйте системи логінів, систему розпізнавання людей або технологію Real Browser Enforcement.

2.4.3.9. Обмежте мережеві ресурси. Якщо попередні заходи не допомогли, то необхідно обмежити ресурси - таким чином буде обмежений «поганий» і «хороший» трафік.

Потужна DOS-атака може зайняти всю ємність інтернет-каналу «жертви», тому на стороні атакується проблему не вирішити: ефективний захист може бути забезпечена тільки на рівні оператора зв'язку. Через зниження вартості організації потужних DOS-атак розраховувати на свої сили в захисті від них можуть дозволити собі тільки компанії, що володіють широкосмуговим доступом в Інтернет і резервуванням каналу підключення, таких компаній в Україні небагато. Тому дуже важливо мати канал зв’язку постачальником інтренет послуг.

2.4.4.Контрзаходи проти найбільш розповсюджених атак:

2.4.4.1.DNSREFLECTEDAMPLIFICATION атака

Данна атака ефективна проти старого непропатченого або неправильно налаштовунаго DNS сервера. Боротись з такою атакою досить легко, необхіднообновити DNS сервер до останьої версії та правильно налаштувати сервер щоб він не відповідав на такого роду запроси. Окрім того необхідно

використовувати DNS Response Rate Limiting. Таким чином ми не тільки зупиняємо атаку, а й усуваємо нашу мережу в участі атак на інші мережі.[14]

2.4.4.2. GENERATED UDP FLOOD атака:

Для боротьби з цим типом атак необхідно обмежити частоту відповіді пактів ICMP, для того щоб розірвати DOS атаку, що вимагає відповіді ICMP. Якщо обсяг потоку UDP має достатньо високий рівень, щоб наситити таблицю стану брандмауера цільового сервера, будь-яке пом'якшення, яке відбувається на рівні сервера, буде недостатнім, оскільки вузьке місце виникне вище від цільового пристрою. Кращим варіантом буде розподілити між багатьма центрами обробки даних. Таким чином навантаження на сервер знизиться, що допоможе подолати цю DOS атаку.

2.4.4.3.HTTP GET/POST FLOOD

Цей тип атаки один за найскладніших у відбиванні. Проте його можливо відбити якщо використовувати Capcha, завдяки цьому можливо відбити більшість атак. Ще одним шляхом буде використання firewall (WAF), виявлення ip з якого йде атака і блокування цього ip. Але краще використовувати Capcha так, як ніщо не заважає зловмиснику змінити ip та продовжити атаку.[15]

2.4.4.4. HIT-AND-RUN

Дуже неприємний тип атак так, як він ефективний проти ручного захисту від DOS атак. Необхідно вивчити період атак, та не втрачати пильності. Це допоможе відбити їх. Краще рішення захисту буде створити кластер який буде працювати як резерв для основного сервера, таким чином ми створюємо автоматичну систему захисту, що під час нападу збільшує обчислювальні можливості мережі.А для моніторінгу краще використовувати нейромережу, що навчена розпізнавати DOS атаку.

2.4.4.5. SYN FLOOD

При атаці SYN FLOOD необхідно збільшити чергу полувідкритих з'єднань, зменшити час очікування з'єднання, додати обмеження числа SYN пакетів в одиницю часу. Таким чином ми зменшуємо навантаження на сервер.[16]

2.4.4.6. SLOWLORIS

Щоб захиститись від цього типу атак краще використовувати нейромережу я класифікатор, в іншому разі нам знадобляться декілька днів щоб відбити мережу. Но для того щоб її використати необхідно необхідно навчити нейромережу розпізнавати «хороші» та «погані» запроси. Таким чином нейромережа виявить усі «погані» ір адреси набагато швидше ніж людина. А потім за допомогою firewall (WAF) блокуємо усі ці ір адреси.

2.4.4.7. Атака з комп'ютера локальною мережі.

За допомогою програми моніторингу трафіку знаходимо персональну станцію з якої йде атака та відключаємо її від мережі.

2.5 Створення нейромережі для автоматичного аналізу трафіку:

Програма створюється для автаматичного аналізу лог файлів програми для моніторингу мережі та виявлення злочиного трафіку. Таким чином у випадку коли починається атака на мережу підприємства,зменшує час реагування. Для цього вона розкладає лог фали на составні частини, порівнює із написаними до її "словнику" ключовими словами, та видає результат порівняння у вигляді тексту у яком вказано чи є трафік шкідливим та його компонети у вигляді кількості запитів від ір адреси, типу трафіку, User Agent, та відповід сервера. Для поповнення словникового запасу програми її необхідно навчити. Для цьго використвується лог файл, в якому заздалегіть відомо що трафік поступавший у мережу поганий чио хороший. У співвідношенні 70/30 таким чином проход первісне навчання мережі. Після проходження первісного навчання нейромережі даємо їй лог файли 60/20/20. Таким чином знижуэмо шанс помилки. Результати перевырки трафыку записуються у логфоайл нейромережі. Для початку створюємо лог файл. Потім нормалізуємо запит у верктор. Таким чином виділяемо необхідні данні з ньго.

Функція map використовується для застосування функції lamda до кожного элементу вектору. Використання функціїї map замість циклу for зумовленно былшою бистродією функції порівняно з циклом. Наступною функією програми є нормалізація уніфікованого локатору ресурсів(адресу ресурсів). Якщо URL більший за 128 то буде видано що адреса надто довга. Якщо менший то він розширується вектор та додаються дані до нього. Наступна створена функція повертає ключі аргументу рядка запиту. При кількості ключів більше за 8 фуекція видаєзабагато ключів. Наступна функція використовує нормалізованану адресу для посилання.

Після цього програма кешує адресу,запит та user agent, та перевіряє на коди 503, 404, 403. Функція повертає запит, посилання, User agend, код. Ці данні додаються до словника програми. Наступною функцією йде додавання зразків програмі для навчання. Ппісля цього в програмі створено 2 паети даних для поганих на хороших результатів. Дані перевіряються та додаються до тренировочних сетів good file та bad file. Після того створюється сама нейромережа, Та починається її навчання.Після навчання сворена функція для аналізу трафіку та його класифікаціїї BOT чи звичайний користувач. Таким чином нейромережа бере лог файл програми для моніторингу мережі розкладає його на вітдільні елементи аналізує їх та визначае чи йде трафік від боту або це звичайний користувач. Треба помітити що ця нейромережа не є саомодостаточною та потребує скриптової обв'язки.Эфективныстъ даноъ нейромережі залежить від кількості даних, отриманих під час навчання. Дана програма виовде нані через лог.

файл stop_DOS.py:

#Імпорт модулів(підключаємо бібліотеки)

import re

import logging

```
# Робимо доступними функції напряму з модулів
from itertools import chain
from collections import namedtuple

from pybrain.datasets            import ClassificationDataSet
from pybrain.utilities           import percentError
from pybrain.tools.shortcuts     import buildNetwork
from pybrain.supervised.trainers import BackpropTrainer
from pybrain.structure.modules   import SoftmaxLayer, SigmoidLayer, LinearLayer

from pybrain.tools.xml.networkwriter import NetworkWriter

import numpy as np
from itertools import permutations

from backports import lfu_cache
from urlparse import urlparse, parse_qs

from cPickle import dump, load
#
LogEntry = namedtuple('LogEntry', 'ip url code size refer useragent')
log = logging.getLogger('')
```

```
log.setLevel(logging.DEBUG)

# Нормалізування запитув вектор та захист на випадок пустого запиті та помилок
#в запиті. За допомого функції mapпри застовуємо функцію lambda для кожного
#елемента запита. Використання функції map замість циклу for зумовлено
#більшою швидкодією функції map.

def normalize_request(req):
        vectors = []
        if req == '-':
                return ['__EMPTY_REQ__']
         try:
                method, url, http = req.split()
               vectors.append('__METHOD_' + method)
               vectors.extend(map(lambda x: "__URL_" + x, normalize_url(url)))
                vectors.append('__HTTP_VER_' + http)
        except Exception as e:
                log.debug("Broken request: {0}. Exc: {1}".format(req, e))
               return ['__BROKEN_REQ__']
         return map(lambda x: "__REQ_" + x, vectors)


@lfu_cache(maxsize=20000)

#Нормалізуємо уніфікований локатор ресурсів (адресу ресурсу), якщо він більший
за 128 то буде записано в лог що адресса була надто довгою. Якщо адресу було
проаналізовано то розримуемо врректор з додаванням даних до нього.
```

```
def normalize_url(url):
    vectors = []
    parsed_url = urlparse(url)
    vectors.append('__SCHEME_' + parsed_url.scheme)
    vectors.append('__NETLOC_' + parsed_url.netloc)
    if len(parsed_url.path) > 128:
        log.debug("url too long: {0}".format(parsed_url.path))
        vectors.append('__PATH_TOO_LONG')
    else:
        vectors.append('__PATH_' + parsed_url.path)
    if parsed_url.query:
        vectors.extend(map(lambda x: "__QS_" + x, normalize_qs(parsed_url.query)))
    return vectors


# функція повертає ключи аргументу рядка запиту, при ключів більше 8 функція
#повертає строку забагато ключів.
def normalize_qs(qs):
    """Just return query string argument keys"""
    qs_keys = parse_qs(qs).keys()
    if len(qs_keys) < 8:
        return parse_qs(qs).keys()
    else:
        log.debug("Too many keys in qs: {0}".format(qs))
```

```
        return ['TOO_MANY_KEYS']


#Функція застосовує нормалізовану адресу для посилання
@lfu_cache(maxsize=20000)
def normalize_refer(refer):
    """Apply normalize_url to refer"""
    if refer == '-':
        return [ '__NO_REFER__' ]
    return map(lambda x: "__REFER_" + x, normalize_url(refer))


# Функція розбирає та нормалізує Агнет користувача(браузер)
@lfu_cache(maxsize=20000)
def normalize_ua(ua):
    """Parse and normalize User-Agent"""
    ua_regexp                    =                    re.compile(r'(?P<comp>[^
]+)(?:\s+\((?P<os>[^)]+)\)(?:\s+(?P<version>.*))?)?')
    if ua == '-':
        return ['__UA_EMPTY']
    try:
        parsed_ua = ua_regexp.match(ua).groups()
    except Exception as e:
        log.debug("Broken UA: {0}".format(e))
        return ['__UA_BROKEN']
```

```
    try:
        base, os, version = parsed_ua
        vector = set()
        vector.add('__BASE_' + base)
        if not all([os, version]):
            return ['__UA_SIMPLE', '__UA_ONLY_BASE_' + base]
        if not version:
            vector.add('__NO_VERSION')
        vector |= set(map(lambda x: '__OS_' + x.strip(), os.split(';')))
        vector |= set(map(lambda x: '__VER_' + x.strip(' ()'), version.split()))
        #vector |= set("__".join(combined) for combined in permutations(vector, 2))
        return map(lambda x: "__UA_" + x, vector)
    except Exception as e:
        log.debug("Failed to parse UA: {0}".format(e))
        return ['__UA_BROKEN']


#
@lfu_cache(maxsize=200000)
def features_from_entry(entry):
    request = set(normalize_request(entry.url))
    refer = set(normalize_refer(entry.refer))
    useragent = set(normalize_ua(entry.useragent))
```

```
    try:
        code = set()
        if int(entry.code) in [503, 404, 403]:
            code.add('__CODE_' + entry.code)
    except Exception as e:
        log.debug("Failed to parse HTTP return code: {0}".format(e))
        pass
    return request | refer | useragent | code


#Додаємо ключовий елемент до словника.
def vector_from_entry(dictionary, entry):
    return np.array([item in features_from_entry(entry) for item in dictionary])


#Додаємо зразки нашій нейромережі для навчання
def add_samples_to_training_set(training_set, file_name, label):
    with open(file_name) as file_:
        for line in file_:
            try:
                entry = LogEntry(*nginx_log_re.match(line).groups())
                training_set.addSample(list(vector_from_entry(dictionary, entry)), label)
            except Exception:
                log.error('Failed to parse line: {0}'.format(line), exc_info=True)
```

```
#Головна частина коду нейромережі.
if __name__ == '__main__':
    from optparse import OptionParser
#Створюємо функцію для визначення типу даних добре, погано та лог для
#аналізу.
    parser = OptionParser()
    parser.add_option("-g", "--good", dest="good_file",
                      help="nginx combined access log with good clients. For example access log before DOS", metavar="FILE")
    parser.add_option("-b", "--bad", dest="bad_file",
                      help="nginx combined access log with bots' requests.", metavar="FILE")
    parser.add_option("-l", "--log", dest="log_file",
                      help="nginx combined access log for classification.", metavar="FILE")
    (options, args) = parser.parse_args()
#Компілюємо дані для збільшення бистродії програмного коду
    nginx_log_re = re.compile(r'(?P<ip>[0-9.:a-f]+) [^ ]+ [^ ]+ \[.*\] "(?P<url>.*)" (?P<code>[0-9]+) (?P<size>[0-9]+) "(?P<refer>.*)" "(?P<useragent>.*)"$')
#Свотрюємо словник та выкриваэмо файли для зчитування та запису даних до
#них.
    log.warning('Preparing dictionary')
    dictionary = set()
```

```
    with open(options.good_file) as good_file:
        with open(options.bad_file) as bad_file:
            for line in chain(good_file, bad_file):
                try:
                    entry = LogEntry(*nginx_log_re.match(line).groups())
                    dictionary |= features_from_entry(entry)
                except Exception:
                    log.error('Failed to parse line: {0}'.format(line), exc_info=True)
    log.warning('Feature vector size: {0}'.format(len(dictionary)))
    dump(dictionary, open('dictionary.p', 'wb'))
# Додаємо зразки для навчання нейромережі.
    log.warning('Adding Samples')
    alldata = ClassificationDataSet(len(dictionary), 1, nb_classes=2,
class_labels=['good','bad'])
    np.random.shuffle(alldata)
    add_samples_to_training_set(alldata, options.good_file, 0)
    add_samples_to_training_set(alldata, options.bad_file, 1)

    log.warning('Preparing data...')
    trndata, tstdata = alldata.splitWithProportion(0.70)

    for data in [trndata, tstdata]:
        data._convertToOneOfMany()
```

```
    # TODO(SaveTheRbtz@): Move to OptionParser
    tries = 10
    epochs = 10
    verbose = True
    fast = False
    bias = True
# Створюємо саму нейромережу
    previous_error = 100
    for _ in xrange(tries):
        log.warning('Constructing NeuralNetwork...')
        try_fnn = buildNetwork(trndata.indim, trndata.indim*2, trndata.outdim,
hiddenclass=SigmoidLayer, outclass=SoftmaxLayer, bias=bias, fast=fast)
# Починаємо навчати нашу мережу
        log.warning('Training NeuralNetwork...')
        trainer = BackpropTrainer(try_fnn, dataset=trndata, momentum=0.1,
verbose=verbose, weightdecay=0.01)
        trainer.trainEpochs(epochs)

        log.warning('Computing train and test errors...')
        trnresult = percentError(trainer.testOnClassData(), trndata['class'])
        tstresult = percentError(trainer.testOnClassData(dataset=tstdata ), tstdata['class'])
        print "epoch: %4d" % trainer.totalepochs, \
```

```
            "  train error: %5.2f%%" % trnresult, \
            "  test error: %5.2f%%" % tstresult
        if tstresult < previous_error:
            fnn = try_fnn
            previous_error = tstresult

    NetworkWriter.writeToFile(fnn, 'nn.xml')
# Нейрона мережа визначае чи йде атака
    log.warning('Activating NeuralNetwork...')
    nginx_log = ClassificationDataSet(len(dictionary), 1, nb_classes=2)
    add_samples_to_training_set(nginx_log, options.log_file, 0)
    nginx_log._convertToOneOfMany()  # це все ще потрібно, щоб fnn почувався
#комфортно

    out = fnn.activateOnDataset(nginx_log)
    out = out.argmax(axis=1)

    with open(options.log_file) as log_file:
        cnt = 0
        for line in log_file:
            try:
                entry = LogEntry(*nginx_log_re.match(line).groups())
                if out[cnt]:
```

```
                print "BOT:  ",
            else:
                print "GOOD: ",
            print "{0}".format(entry)
            cnt += 1
        except Exception:
            log.error('Failed to parse line: {0}'.format(line), exc_info=True)
```

backport.py

```
importcollections
importfunctools
from itertools import ifilterfalse
from heapq import nsmallest
from operator import itemgetter

class Counter(dict):
    'Mapping where default values are zero'
    def __missing__(self, key):
        return 0

def lru_cache(maxsize=100):
    '''Least-recently-used cache decorator.
```

```
Arguments to the cached function must be hashable.

Cache performance statistics stored in f.hits and f.misses.

Clear the cache with f.clear().

http://en.wikipedia.org/wiki/Cache_algorithms#Least_Recently_Used

'''

maxqueue = maxsize * 10

def decorating_function(user_function,

        len=len, iter=iter, tuple=tuple, sorted=sorted, KeyError=KeyError):

    cache = {}                  # mapping of args to results

    queue = collections.deque() # order that keys have been used

    refcount = Counter()        # times each key is in the queue

    sentinel = object()         # marker for looping around the queue

    kwd_mark = object()         # separate positional and keyword args


    # lookup optimizations (ugly but fast)

    queue_append, queue_popleft = queue.append, queue.popleft

    queue_appendleft, queue_pop = queue.appendleft, queue.pop


    @functools.wraps(user_function)

    def wrapper(*args, **kwds):

        # cache key records both positional and keyword args

        key = args
```

```
        if kwds:
            key += (kwd_mark,) + tuple(sorted(kwds.items()))

        # record recent use of this key
        queue_append(key)
        refcount[key] += 1

        # get cache entry or compute if not found
        try:
            result = cache[key]
wrapper.hits += 1
        except KeyError:
            result = user_function(*args, **kwds)
            cache[key] = result
wrapper.misses += 1

            # purge least recently used cache entry
            if len(cache) > maxsize:
                key = queue_popleft()
                refcount[key] -= 1
                while refcount[key]:
                    key = queue_popleft()
```

```
                refcount[key] -= 1
            del cache[key], refcount[key]

        # periodically compact the queue by eliminating duplicate keys
        # while preserving order of most recent access
        if len(queue) > maxqueue:
refcount.clear()
            queue_appendleft(sentinel)
            for key in ifilterfalse(refcount.__contains__,
iter(queue_pop, sentinel)):
                queue_appendleft(key)
                refcount[key] = 1


        return result

    def clear():
cache.clear()
queue.clear()
refcount.clear()
wrapper.hits = wrapper.misses = 0
```

```
wrapper.hits = wrapper.misses = 0
wrapper.clear = clear
        return wrapper
    return decorating_function



def lfu_cache(maxsize=100):
    '''Least-frequenty-used cache decorator.
    Arguments to the cached function must be hashable.
    Cache performance statistics stored in f.hits and f.misses.
    Clear the cache with f.clear().
    http://en.wikipedia.org/wiki/Least_Frequently_Used
    '''
    def decorating_function(user_function):
        cache = {}                  # mapping of args to results
        use_count = Counter()       # times each key has been accessed
        kwd_mark = object()         # separate positional and keyword args

        @functools.wraps(user_function)
        def wrapper(*args, **kwds):
            key = args
            if kwds:
```

```
            key += (kwd_mark,) + tuple(sorted(kwds.items()))
        use_count[key] += 1

        # get cache entry or compute if not found
        try:
            result = cache[key]
wrapper.hits += 1
        except KeyError:
            result = user_function(*args, **kwds)
            cache[key] = result
wrapper.misses += 1

            # purge least frequently used cache entry
            if len(cache) > maxsize:
                for key, _ in nsmallest(maxsize // 10,
                                        use_count.iteritems(),
                                        key=itemgetter(1)):
                    del cache[key], use_count[key]

        return result

    def clear():
```

```
cache.clear()
            use_count.clear()
wrapper.hits = wrapper.misses = 0

wrapper.hits = wrapper.misses = 0
wrapper.clear = clear
        return wrapper
    return decorating_function



if __name__ == '__main__':

    @lru_cache(maxsize=20)
    def f(x, y):
        return 3*x+y

    domain = range(5)
    from random import choice
    for i in range(1000):
        r = f(choice(domain), choice(domain))

print(f.hits, f.misses)
```

```
@lfu_cache(maxsize=20)
def f(x, y):
    return 3*x+y

domain = range(5)
from random import choice
for i in range(1000):
    r = f(choice(domain), choice(domain))

print(f.hits, f.misses)
```

2.4.6.Висновки

Не існує універсального методу захисту від DOSатак, кожен випадок унікальний і залежить від типу атаки, та її потужності. У той час як існує велика кількість програм для запобігання DOS атак, більшість з них не може бути застосовано для невеликих мереж. В кінцевому рахунку, ви самі повинні захиститися від DOS. Це означає, що ви повинні чітко знати, як реагувати на напад - ідентифікуючи трафік, розробляючи і здійснюючи фільтри. Необхідно проводити навчання співробітників. Підготовка та планування, безумовно, кращі методи для того, щоб пом'якшити майбутні DOS нападу. Окрім того невелика пропускна здатність каналу інтренет мережі на підприємствах, зменшує шанс ефективної боротьби з атакою на рівні підприємства, а тому потрібно звертатись до постачальника інтрнет послуг. Таким чином DOS атаки являють велику небезпеку для малих та середніх підприємств.

3 ЕКОНОМІЧНА ЧАСТИНА

3.1 Цілі та задачі, що вирішуються в економічній частині.

В Кваліфікаційній роботі була розроблена методика захисту від DOS атак, для підприємств. Результатом вирішення цієї задачі, буде методика захисту комерційних підприємств, що захищатиме їх на гідному рівні. Цілю даного розділу є розрахунок затрат.

3.1.1 Розрахунок капітальних витрат на придбання та налагодження складових системи захисту від DOS атак, та програмного забезпечення.

3.1.2.Розрахунок річних експлуатаційних витрат на утримання і обслуговування об'єкта проектування.

3.1.3.Визначення річного економічного ефекту від впровадження об'єкта проектування.

3.1.4.Визначення та аналіз показників економічної ефективності запропонованого у дипломному проекті проектного рішення.

3.1.5. Висновок про економічну доцільність проектного рішення.

3.2 Розрахунок капітальних витрат на придбання та налагодження складових системи захисту від DOS атак, та програмного забезпечення:

3.2.1 Вартість розробки проекту інформаційної безпеки та витрати на залучення зовнішніх консультантів склали 30000 грн

3.2.2 Витрати на закупівлю необхідного обладнання:

Комп'ютер 3 шт(2 під сервери, 1 для моніторингу) – 55680грн

Монітор 21.5" Samsung S22F350F 3шт. – 7800 грн.

Каз = 55680+7800 = 63 480 грн

3.2.3 Вартість на закупівлю ліцензійного основного та додаткового програмного забезпечення.

Microsoftwindows 10 professional – 6 983 грн

Microsoftsever – 25216 грн

NagiosLogServer – 111860 грн

Кзпз = 144059 грн

3.2.4 Витрати на інтеграцію системи безпеки від DOS у вже існуючу корпоративну систему:

Робота проводиться працівником самого підприємства з окладом в 15000 грн в місяць. Працівник затратить на інтеграцію обладнання 2 робочі дні. Таким чином витрати на інтеграцію складуть:

$$\frac{15000 * 2}{22} + \frac{1500 * 2 * 0{,}22}{22} = 1664 \text{ грн}$$

Кн = 1664 грн

3.2.5. Розрахунок витрат на створення нейромережі

3.2.5.1 Визначення трудомісткості розробки та опрацювання програмного продукту.

t=ttз+tв+ta+tпр+toпр+tд

Умовна кількість операторів у програмі:

$$Q = q * c * (1 + p) = 222 * 1{,}25 * 1{,}006 = 279$$

Тривалість вивчення технічного завдання, з урахуванням уточнення ТЗ та кваліфікації програміста.

$$t\text{в} = \frac{Q * B}{75 * k} = \frac{279 * 1{,}2}{75 * 0{,}8} = 6 \text{ годин}$$

де В – коефіцієнт збільшення тривалості етапу внаслідок недостатнього опису завдання, В = 1,2…1,5

k – коефіцієнт, що враховує кваліфікацію програміста і визначається стажем роботи за фахом k = 0,8.

Тривалість розробки блок-схеми алгоритму:

$$t\text{а} = \frac{Q}{20 * k} = \frac{279}{20 * 0{,}8} = 17 \text{ годи.}$$

Тривалість складання програми за блок схемою:

$$t\text{пр} = \frac{Q}{20 * k} = \frac{279}{20 * 0{,}8} = 17 \text{ годин}$$

Тривалість опрацювання програми на ПК:

$$t\text{опр} = \frac{1{,}5 * Q}{5 * k} = \frac{1{,}5 * 279}{5 * 0{,}8} = 105 \text{ годин}$$

Тривалість підготовки технічної документації на ПЗ:

$$t\text{д} = \frac{Q}{20 * k} + \frac{Q}{20} * 0{,}75 = \frac{279}{20 * 0{,}8} + \frac{279}{20} * 0{,}75 = 28 \text{ годин}$$

Трудомісткість розробки та опрацювання програмного продукту:

t=10+6+17+17+105+28=183 години

3.2.5.2 Розрахунок витрат на створення програмного продукту:

Витрати на створення програмного продукту Кпз складаються з витрат на заробітну плату виконавця програмного забезпечення Ззп та вартості витрат машинного часу, що необхідний для опрацювання програми на ПК Змч.

Кпз=Ззп+Змч

Заробітна праця виконавця враховує основну та додаткову заробітну плату, а також відрахування на соціальні потреби та визначається за формолую:

Ззп=t*Зпр, грн,

де t– загальна тривалість створення ПЗ годин,

Зпр – средньогодинна заробітна плата програміста з нарахуваннями грн/годину. У 2020 році середня заробітна плата програміста склала 366 грн/год.

Ззп = 183*366 = 66978 грн

Вартість машинного часу для налагодження програми на ПК визначається за формолую:

Змч=tопр*Смч*tд, грн,

де tопр – трудомісткість налагодження налагодження програми на ПК, годин,

tд – трудомісткість підготовки документації на ПК, годин,

Смч – вартість 1 години машиного часу ПК, грн/год.

Смч розраховується за формулою:

$$\text{Смч} = \text{Р} * t\text{нал} * \text{Се} + \frac{\text{Фзал} * \text{На}}{F\text{р}} + \frac{\text{Клпз} * \text{Напз}}{F\text{р}}$$

де Р – встановлена потужність ПК, кВт

Се – тариф на електричну енергію

Фзал - Залишкова вартість ПК на поточний рік

На – річна норма амортизації ПК

Напз – річна норма амортизації на ліцензійне програмне забезпечення

Клпз – вартість ліцензійного програмного забезпечення

Fр – річний фонд робочого часу 1920

$$\text{Смч} = 0{,}412 * 279 * 2{,}74 + \frac{20000 * 0{,}15}{1920} + \frac{8274{,}5 * 0{,}2}{1920} = 317{,}1$$

Зм=105*317.1+28 = 33295,5 грн

Кпз = 66978+33295,5= 100273,5 грн

Таким чином капітальні витрати на проектування та впровадження проектної методики складає:

К = 30000+63 480+144059+1664+100273,5 = 339476,5 грн

3.3 Розрахунок поточних(експлуатаційних) затрат:

3.3.1 Вартість відновлення й модернізації системи(Св);

Св = 20000 грн/рік

3.3.2 Витрати на керування системи в цілому(Ск)

3.3.3 Витрати викликані активністю користувачів системи інформаційної безпеки(Сак)

Під «витратами на керування системою» маються на увазі витрати, пов'язані з керуванням та адмініструванням серверів та інших компонентів системи. До цієї статті витрат відносять наступні витрати:

- Навчання адміністративного персоналу та кінцевих користувачей
  Сн = 20000
- Амортизаційні відрахування від вартості обладнання та ПЗ.
  31130,85
- Заробітна плата обслуговуючого персоналу.
  Сз = 144000+12960 = 156960 грн
- Аутсоринг С0= 0
- Навчальні курси та сертифікація обслуговуючого персоналу
- Технічне й адміністративне адміністрування та сервіс
  Сток = 6789,53

Витрати на електроенергію.

Се=1,236*2112*2,74=7152,58 грн

Ск= 20000+31130+156960+0+6789,53+7152,58+34531,2=256663,31

Сак = 390000 грн

Поточні(експлуатаційні) витрати:

$$C = 256663,31+390000+20000=666663,31$$

3.4 Оцінка можливого збитку відDOS атаки:

tп – час простою вузла або сегменту корпоративної мережі

tв – час відновлення після атаки персоналом, що обслуговує корпоративну мережу

Зо – Заробітна плата обслуговуючого персоналу

Зс – заробітна плата співробітників атакованого вузла або сегмента корпоративної мережі

Чо – Чисельність обслуговуючого персоналу

Чс – Чисельність співробітників атакованого вузла або сегменту корпоративної мережі

О – обсяг продажів атакованого вузла або сегменту корпоративної мережі на рік

I – число атакованих вузлів або сегментів корпоративної мережі на рік

N – середнє число атак на рік

Упущена вигода від простою атакованого вузла або сегмента корпоративної мережі становить:

$$U=Пп+Пв+V$$

де Пп – оплачувані втрати робочого часу та простої співробітників атакованого вузла або сегмента корпоративної мережі.

Пв – вартість відновлення працездатності вузла або сегмента корпоративної мережі.

V – втрати від зниження обсягу продаж

$$Пп = \frac{15000 + 15000 + 15000 + 12000 + 24000 + 15000 + 15000}{176} * 8 = 5045,45$$

$$\text{Пв} = \frac{20000 + 15000 + 15000}{176} * 4 = 1136{,}36$$

$$V = \frac{5000000}{2080} * (8 + 4 + 10) = 52884{,}61$$

$$U = 5045{,}45 + 1136{,}36 + 52884{,}61 = 59066{,}42$$

$$B = 5 * 10 * 59066{,}42 = 2953321$$

3.5 Загальний ефект від впровадження системи інформаційної безпеки.

$$E = B * R - C = 2953321 * 0{,}3 - 666663{,}31 = 219332{,}99$$

3.6 Визначення та аналіз показників економічної ефективності системи інформаційної безпеки.

3.6.1 Коефіцієнт повернення інвестицій, у сфері інформаційної безпеки.

$$ROSI = \frac{E}{K} = \frac{219332{,}99}{339476{,}5} = 0{,}65$$

ROSI>(8,58-4,1)\100

ROSI>0,0448

3.6.2 термін окупності підприємства:

$$To = \frac{K}{E} = \frac{1}{ROSI} = \frac{1}{0.65} = 1{,}54$$

Таким чином проект окупить себе за 1,54 роки.

3.7 Висновки:

Капітальні інвестиції окупаються досить швидко вже за півтора роки, досить великі експлуатаційні витрати менші ніж збиток від можливих DOSатак. Одним із найбільших експлуатаційних витрат є заробітна плата обслуговуючого персоналу. Вибір NagiosLogServer обумовлений її функціоналом, надійністю та

універсальністю порівняно з конкурентами. Окрім того ліцензія цієї програми купується 1 раз і в надалі не потребує оновлення кожний рік.

4 ВИСНОВКИ

З момента своєї появи та по сучасний день DOSатаки залишаються одним з найбільших викликів для підприємств. Виникали та зникали засоби захисту, змінювались та вдосконалювались методи атак. З розвитком мережі інтернет кількість атак постійно збільшується. А з 2019 року кількість атак значно збільшилась, були знайдені нові вразливості та створені більш ефективні ботнети. Окрім того з почтаку 2020 року експерти спостерігають аномальну кількість DOS атак. Швидше за все, це пов'язано з пандемією коронавируса і обмежувальними заходами, які в багатьох країнах тривали частина кварталу або на всій його довжині. Вимушене переміщення життя в інтернет збільшило кількість можливих мішеней для DOS. В іншому в другому кварталі мало що змінилося: склад TOP 10 за кількістю атак і мішеней практично той же, як і розподіл атак по тривалості. Помітно впала частка всіх видів DOS, крім SYN- і ICMP-флуду, однак говорити про яку-небудь тенденції в зв'язку з цим, ще рано. Проте можливо сказати що ринок DOS атак стабілізувався після буму у 2019 році. А підприємства вимушені шукати нові методи та методики захисту від DOSатак.

Не існує універсального методу захисту від DOS атак, кожен випадок унікальний і залежить від типу атаки, та її потужності. У той час як існує велика кількість програм для запобігання DOS атак, більшість з них не може бути застосовано для невеликих мереж. В кінцевому рахунку, ви самі повинні захиститися від DOS. Це означає, що ви повинні чітко знати, як реагувати на напад - ідентифікуючи трафік, розробляючи і здійснюючи фільтри. Необхідно проводити навчання співробітників. Підготовка та планування, безумовно, кращі методи для того, щоб пом'якшити майбутні DOS нападу. Окрім того невелика пропускна здатність каналу інтренет мережі на підприємствах, зменшує шанс ефективної боротьби з атакою на рівні підприємства, а тому потрібно звертатись до постачальника інтрнет послуг. Таким чином DOS атаки

являють велику небезпеку для малих та середніх підприємств. Саме тому було вирішено створити власну методику захисту де аналізом трафіку займалась не людина, а нейромережа. Таким чином можливо значно знизити час необхідний для реагування на атаку. Окрім того ця методика показала себе дуже гарно в плані окупності. Так капітальні інвестиції окупаються досить швидко вже за півтора роки, досить великі експлуатаційні витрати менші ніж збиток від можливих DOS атак. Одним із найбільших експлуатаційних витрат є заробітна плата обслуговуючого персоналу. Вибір Nagios Log Server обумовлений її функціоналом, надійністю та універсальністю порівняно з конкурентами. Окрім того ліцензія цієї програми купується 1 раз і в надалі не потребує оновлення кожний рік.

ПЕРЕЛІК ПОСИЛАНЬ

1.DOS-атаки во втором квартале 2020 годаURLhttps://securelist.ru/DOS-attacks-in-q2-2020/97701/

2.DOS-атаки в первом квартале 2019 годаURLhttps://securelist.ru/DOS-report-q1-2019/93890/

3.New Botnet Shows Evolution of Tech and Criminal Culture URL https://www.darkreading.com/attacks-breaches/new-botnet-shows-evolution-of-tech-and-criminal-culture/d/d-id/1333792?__cf_chl_captcha_tk__=73a449861a2414a69881c1e709b8b08708992877-1607945077-0-AfN6s1zCI-IgAIxNpJQN-wDT18N-VK1bCGFv9eieVBQsZ6UuCxfpmWTfmq23bQd7dPVZuOe81wX3u3JPBomvXlnj6qKOhfHnZlpYi4zo-laqhzKU7TstLY-NzsWhWNdGdiQQ8DjBR0wpHuMfqSaumu57jxEO8R3_j-kIb_U6F_xxZcRFz7ISc7wIknm6ahquvqKuQcAMaZdtZXAXl1xb16X_4WvTk_2luMq9XJd6IRxKDgqoejdKnpNtlabrIodX_WwHDk74LDiR2RThmIWRYYmo8dBd7dqdaKhjr9EatUgaFpGoZXYJ1ocZyJkNDQI4MXO1FPNsn9Tjrch829YsqxvWfoKvLHhi38robPhyFcpkxu2ernuwrsvafQGyYebnujLIWHgRrTd3C94K0TEO3-zhiKwL2nJAXe747JOUgumPFjJrxnHUaC7NHcoiY46tiXgwk14CFl_x93GyPg10-BA-DuuMGe43nBEakKcQQgjY31Rjx7Ch3wX0_H2ZL7bRSKEKWtYlj7DI3ynMQlMqJXdlcILA_6dm56Xt_DOj5IUj8FanibO11O6MUSWXUDgysSp-08QXi_gMtvRrgPg6qyyzVvFjy2B5SuALFq43353j3fd6eIFb0DNcDJS9FNNJQw

4.NXNSAttack URLhttps://cyber-security-group.cs.tau.ac.il/

5.The Mirai botnet exploits a new vulnerability affecting companies around the world URLhttps://www.pandasecurity.com/en/mediacenter/business/mirai-botnet-exploits-new-vulnerability/

6. New Mirai Variant Expands, Exploits CVE-2020-1017 URLhttps://www.trendmicro.com/en_us/research/20/g/new-mirai-variant-expands-arsenal-exploits-cve-2020-10173.html

7.Lucifer's SpawnURLhttps://www.netscout.com/blog/asert/lucifers-spawn

8. Lucifer: New Cryptojacking and DOS Hybrid Malware Exploiting High and Critical Vulnerabilities to Infect Windows Devices URLhttps://unit42.paloaltonetworks.com/lucifer-new-cryptojacking-and-DOS-hybrid-malware/

9. RangeAmp attacks can take down websites and CDN servers URLhttps://www.zdnet.com/article/rangeamp-attacks-can-take-down-websites-and-cdn-servers/

10. macOS systems abused in DOS attacks URLhttps://www.zdnet.com/article/macos-systems-abused-in-DOS-attacks/

11. Linux Webmin Servers Being Attacked by New P2P Roboto Botnet URL

https://www.bleepingcomputer.com/news/security/linux-webmin-servers-being-attacked-by-new-p2p-roboto-botnet/

12. New Mozi P2P Botnet Takes Over Netgear, D-Link, Huawei RoutersURL

https://www.bleepingcomputer.com/news/security/new-mozi-p2p-botnet-takes-over-netgear-d-link-huawei-routers/

13.This aggressive IoT malware is forcing Wi-Fi routers to join its botnet armyURL

https://www.zdnet.com/article/this-aggressive-iot-malware-is-forcing-wi-fi-routers-to-join-its-botnet-army/

14. DNS Amplification Attack URLhttps://www.cloudflare.com/learning/DOS/dns-amplification-DOS-attack/

15.HTTP FloodURLhttps://www.imperva.com/learn/DOS/http-flood/

16.SYN Flood AttackURLhttps://www.cloudflare.com/learning/DOS/syn-flood-DOS-attack/

17. Методы борьбы с DOS-атакамиURLhttps://habr.com/ru/post/129181/

18. Методы защиты от DOS нападений URLhttps://www.securitylab.ru/analytics/216251.php

19. Способы защиты от DOS-атаки URLhttps://timeweb.com/ru/community/articles/sposoby-zashchity-ot-DOS-ataki-1

20. DOS: механизм атаки и методы защиты URLhttp://www.panasenko.ru/Articles/12/12.html

21. Выбираем и проверяем технологии защиты от DOS-атакURLhttps://www.anti-malware.ru/analytics/Technology_Analysis/Choosing-anti-DOS-technology

22.Как не потратить деньги на защиту от DOS впустуюURLhttps://safe.cnews.ru/articles/2020-10-26_kak_ne_potratit_dengi_na_zashchitu

ДОДАТОК А. Відомість матеріалів кваліфікаційної роботи

№	Формат	Найменування	Кількість листів	Пимітки
Документація				
1	А4	Реферат	3	
2	А4	Список умовних скорочень	1	
3	А4	Зміст	2	
4	А4	Втуп	1	
5	А4	Стан питання та постанова задачі	8	
6	А4	Спеціальна частина	36	
7	А4	Економічна частина	8	
8	А4	Висновки	2	
9	А4	Перелик посилань	2	
10	А4	Додаток А	1	
11	А4	Дотаток Б	1	
12	А4	Додаток В	2	

ДОДАТОК Б. Перелік документів на оптичному носії.

1. Пояснювальна записка.docx

2.Демонстраційний матеріал.ppt

ДОДАТОК В. Відгук керівника кваліфікаційної роботи

В І Д Г У К

на кваліфікаційну роботу студента групи 125м-19-1

Харитонова Івана Андрійовича

на тему: «методика захисту ІТС комерційних підприємств від DOS атак»

Пояснювальна записка складається зі вступу, трьох розділів і висновків, викладених на 60 сторінках.

Метою кваліфікаційної роботи є створення методики захисту від DOSатак, в якій для аналізу трафіку та виявлення атаки використовувалась нейромережа.

Тема кваліфікаційної роботи безпосередньо пов'язана з об'єктом діяльності магістра спеціальності 125 «Кібербезпека». Для досягнення поставленої мети в кваліфікаційній роботі вирішуються наступні задачі: аналіз мереж комерційних підприємств, виявлення найбільш поширенихтипів DOS атак, дослідження сучасних засобів захисту, розробити методику захисту від DOS атак, створити нейромережу для швидкого аналізу трафіку.Розроблено рекомендації для проведення ідентифікації інформаційних активів.

Практичне значення результатів кваліфікаційної роботи полягає у зменшені часу реагування на DOS атаку.

Оформлення пояснювальної записки до кваліфікаційної роботи виконано з незначними відхиленнями від стандартів.

За час дипломування Хартинов І.А. проявив себе фахівцем, здатним самостійно вирішувати поставлені задачі та заслуговує присвоєння кваліфікації

магістра за спеціальністю 125 Кібербезпека, освітньо-професійна програма«Кібербезпека» .

Рівень запозичень у кваліфікаційній роботі не перевищує вимог“Положення про систему виявлення та запобігання плагіату”.

Кваліфікаційна робота заслуговує оцінки « ».

Керівник кваліфікаційної роботи

Керівник спец. розділу