

Міністерство освіти і науки України
Державний університет «Житомирська політехніка»
Факультет інформаційно-комп'ютерних технологій
Кафедра комп'ютерної інженерії та кібербезпеки

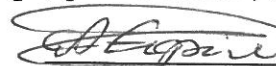
ПОЯСНЮВАЛЬНА ЗАПИСКА

до випускної кваліфікаційної роботи бакалавра
на тему: «Проект підсистеми захисту сервісів DNS
гетерогенної мережі організації.»

Виконав студент 4-го курсу, групи КБ-1
спеціальності 125 «Кібербезпека»

 Б.С. Лещенко

Керівник, завідувач кафедри КІ та КБ, к.т.н.

 А.А. Єфіменко

Рецензент, доцент кафедри КІ та КБ, к.т.н., доцент

 Ю.М. Россінський

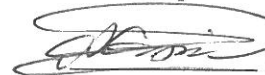
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА КІБЕРБЕЗПЕКИ
СПЕЦІАЛЬНІСТЬ 125 «КІБЕРБЕЗПЕКА»

ЗАТВЕРДЖУЮ

Завідувач кафедри

комп'ютерної інженерії

та кібербезпеки

 Єфіменко А. А.

«12» квітня 2021 р.

ЗАВДАННЯ
на випускню кваліфікаційну роботу

Студента Лещенка Богдана Сергійовича

Тема роботи: Проект підсистеми захисту сервісів DNS гетерогенної мережі організації

Затверджена Наказом університету від «12» квітня 2021 р. № 134/с

Термін здачі студентом закінченої роботи «15» червня 2021р.

Вихідні дані роботи (зазначається предмет і об'єкт дослідження):

1. Налаштуваний DNS сервер за допомогою вбудованих функцій Cisco
2. Налаштувані маршрутизатори Cisco в ролі серверів DNS
3. Сканована мережа за допомогою програмного додатку Nmap
4. Клонована веб сторінка на соціальній мережі
5. Налаштуваний веб серверу Apache2 з підробною сторінкою авторизації
6. Здійснена атака DNS Spoofing за допомогою Ettercap

Консультанти з випускної кваліфікаційної роботи із зазначенням розділів, що їх стосуються

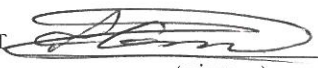
Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

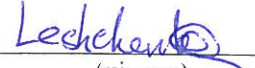
Керівник


(підпис)

Календарний план

№ з/п	Назва етапів випускної кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Постановка завдання	12 квітня 2021 – 16 квітня 2021	Виконано
2	Дослідження теоретичних відомостей, дослідження принципів роботи та принципів проведення атак на систему доменних імен	17 квітня 2021 – 19 квітня 2021	Виконано
3	Розробка схеми комплексної атаки на DNS та опис етапів та програмного забезпечення для проведення атак	20 квітня 2021 – 24 квітня 2021	Виконано
4	Налаштування DNS серверу за допомогою вбудованих функцій Cisco Packet Tracer та налаштування маршрутизатора Cisco в ролі серверу DNS	25 квітня 2021 – 30 квітня 2021	Виконано
5	Налаштування ОС Windows Server в ролі DNS серверу	1 травня 2021 – 4 травня 2021	Виконано
6	Сканування мережі за допомогою програмного додатку Nmap	5 травня 2021 – 9 травня 2021	Виконано
7	Здійснення клонування веб сторінки на прикладі сторінки соціальної мережі	10 травня 2021 – 13 травня 2021	Виконано
8	Налаштування веб серверу Apache2 з підробною сторінкою авторизації	14 травня 2021 – 18 травня 2021	Виконано
9	Здійснення атаки DNS Spoofing за допомогою Ettercap	19 травня 2021 – 22 травня 2021	Виконано
10	Дослідження методів та засобів для покращення безпеки dns	19 травня 2021 – 22 травня 2021	Виконано
11	Дослідження DNS over HTTPS та Oblivious DNS over HTTPS	23 травня 2021 – 24 травня 2021	Виконано
12	Аналіз результатів	23 травня 2021 – 24 травня 2021	Виконано
13	Написання пояснювальної записки	25 травня 2021 – 29 травня 2021	Виконано
14	Підготовка презентації та виступу	30 травня 2021 – 31 травня 2021	Виконано
15	Отримання відгуку керівника та рецензії на кваліфікаційну роботу	4 червня 2021 – 8 червня 2021	Виконано
16	Захист	18 червня 2021	Виконано

Студент  (підпис)

Керівник  (підпис)

РЕФЕРАТ

Кваліфікаційна робота складається з проекту підсистеми захисту сервісів DNS гетерогенної мережі організації та пояснювальної записки до нього. Пояснювальна записка до роботи містить 90 сторінок, 84 ілюстрацій та 2 таблиць.

Метою роботи є розробка проекту підсистеми захисту сервісів DNS гетерогенної мережі організації.

В роботі було досліджено теоретичні відомості, принципи роботи та принципи проведення атак на систему доменних імен. Досліджено загальні відомості про систему DNS, структуру системи доменних імен, структуру пакету та типи записів доменної системи імен. Досліджено роботу протоколу DNS за допомогою додатків Nslookup, dig, WireShark та tcpdump. Розроблено схему комплексної атаки на DNS та опис етапів та програмного забезпечення для проведення атак.

Налаштувано DNS сервер за допомогою вбудованих функцій Cisco Packet Tracer та налаштувано маршрутизатори Cisco в ролі серверу DNS. Налаштувано ОС Windows Server в ролі DNS серверу. Сканувано мережі за допомогою програмного додатку Nmap. Здійснено клонування веб сторінки на прикладі сторінки соціальної мережі та налаштувано веб сервер Apache2 з підробною сторінкою авторизації. Здійснено атаку DNS Spoofing за допомогою Ettercap.

Досліджено важливість безпеки DNS, типові проблеми безпеки DNS та розглянено DoS атаки на DNS. Досліджено DNS over HTTPS та Oblivious DNS over HTTPS, технології DNS – Security Extensions та DNSCrypt.

КЛЮЧОВІ СЛОВА: ЗАХИСТ, МЕРЕЖА, DNS, ДОМЕННА СИСТЕМА ІМЕН.

Зм.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Лещенко Б.С.		16.08.21	Проект підсистеми захисту сервісів DNS гетерогенної мережі організації			
Керівник		Ефіменко А. А.		16.08.21				
Рецензент.		Росінський Ю.М.		16.08.21				
Зав. каф.		Ефіменко А. А.		16.08.21				
					Літ.	Арк.	Аркушів	
					Державний університет «Житомирська політехніка», група КБ-1			

ABSTRACT

The qualification work consists of a project of the DNS service protection subsystem of the heterogeneous network of the organization and an explanatory note to it. The explanatory note to the work contains 90 pages, 84 illustrations and 2 tables.

The purpose of the work is to develop a project of the DNS protection subsystem of the heterogeneous network of the organization.

Theoretical information, principles of work and principles of attacks on the domain name system were investigated in the work. The general information about the DNS system, the structure of the domain name system, the structure of the package and the types of records of the domain name system are studied. The operation of the DNS protocol was studied using the applications Nslookup, dig, WireShark and tcpdump. The scheme of a complex attack on DNS and the description of stages and the software for carrying out attacks is developed.

The DNS server is configured using the built-in Cisco Packet Tracer features, and the Cisco routers are configured as a DNS server. Windows Server OC is configured as a DNS server. Networks were scanned using the Nmap software application. The web page was cloned on the example of a social network page and the Apache2 web server with a fake authorization page was configured. DNS Spoofing attack with Ettercap.

The importance of DNS security, typical DNS security issues and DOS attacks on DNS are investigated. DNS over HTTPS and Oblivious DNS over HTTPS, DNS - Security Extensions and DNSCrypt technologies have been studied.

KEY WORDS: PROTECTION, NETWORK, DNS, DOMAIN NAME SYSTEM.

ЗМІСТ

РОЗДІЛ 1. ТЕОРЕТИЧНІ ВІДОМОСТІ, ДОСЛІДЖЕННЯ ПРИНЦИПІВ РОБОТИ ТА ПРИНЦИПІВ ПРОВЕДЕННЯ АТАК НА СИСТЕМУ ДОМЕННИХ ІМЕН	8
1.1. Загальні відомості про систему DNS, структура системи доменних імен та структура пакету	8
1.2. Типи записів доменної системи імен	18
1.3. Дослідження роботи протоколу DNS за допомогою додатків Nslookup, dig, WireShark та tcpdump	24
1.4. Розробка схеми комплексної атаки на DNS та опис етапів та програмного забезпечення для проведення атак	33
РОЗДІЛ 2. ПРАКТИЧНЕ НАЛАГОДЖЕННЯ СЕРВЕРУ СИСТЕМИ ДОМЕННИХ ІМЕН ТА ЗДІЙСНЕННЯ АТАКИ З ПІДМІНИ ЗАПИСІВ ДОМЕННОЇ СИСТЕМИ ІМЕН	46
2.1. Налаштування DNS серверу за допомогою вбудованих функцій Cisco Packet Tracer та налаштування маршрутизатора Cisco в ролі серверу DNS.	46
2.2. Налаштування ОС Windows Server в ролі DNS серверу	51
2.3. Сканування мережі за допомогою програмного додатку Nmap	56
2.4. Здійснення клонування веб сторінки на прикладі сторінки соціальної мережі та налаштування веб серверу Apache2 з підробною сторінкою авторизації.	60
2.5. Здійснення атаки DNS Spoofing за допомогою Ettercap	68
РОЗДІЛ 3. ДОСЛІДЖЕННЯ МЕТОДІВ ТА ЗАСОБІВ ДЛЯ ПОКРАЩЕННЯ БЕЗПЕКИ DNS	74
3.1. Важливість безпеки DNS, типові проблеми безпеки DNS та розгляд DoS атак на DNS	74
3.2. DNS over HTTPS та Oblivious DNS over HTTPS	77
3.3. Технологія DNS – Security Extensions та DNSCrypt	79
3.4. Засоби запобігання ARP spoofing	86
ВИСНОВКИ	89
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	91
ДОДАТКИ	93

		Лещенко Б.С.				Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		5

ВСТУП

Система доменних імен існувала, ще до початку створення всесвітньої мережі Інтернет, ця система є фундаментальною та однією з найважливіших в роботі сучасних комп'ютерних мереж. Кожен системний адміністратор, спеціаліст з інформаційної безпеки, спеціаліст з кібербезпеки та будь-яка людина, яка має справу з комп'ютерними мережами, повинні розумітися в принципах роботи DNS. Тому темою дипломного проекту було обрано тему – «Проект підсистеми захисту сервісів DNS гетерогенної мережі організації».

DNS є основною складовою повсякденного життя кожного в Інтернеті, але дуже мало людей розуміє, як це працює, або наскільки крихкою може бути основна інфраструктура. Навіть фахівці з безпеки, яким доручається захистити організацію, часто не мають повного уявлення про потенційні підводні камені в DNS.

Практично всі пристрої, які підключені до мережі, використовують DNS. DNS постійно підлягає атакам з боку зловмисників. Сьогодні існує багато корпоративних рішень для захисту від таких атак, але їх вартість занадто велика для невеликих підприємств. Також сьогодні все більше людей починають працювати з дому та з власних комп'ютерних систем, що вносить додаткову небезпеку з точки зору інформаційної безпеки. Дуже часто звичайні користувачі та працівники підприємств не беруть до уваги ризики своєї кібербезпеки та ніяк не турбуються про безпеку своїх даних. Використання стандартних авторизаційних даних для входу до адміністративної панелі маршрутизатору та застарілі версії операційних систем та програмного забезпечення – це речі, з якими постійно стикаються кібердослідники. Система доменних імен є ще одним вектором, завдяки якому, приватність кінцевих користувачів може бути порушена. Недооцінка важливості безпеки протоколу DNS, навіть у 2021 році, може призвести до значних грошових втрат та , навіть, техногенних катастроф. Саме тому обрана проблематика дослідження є вельми актуальною на даний момент та буде актуальною ще довгий час.

		Лещенко Б.С.				Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		6

В контексті дослідження системи доменних імен підґрунтям є документація Cisco, навчальні посібники авторів Cricket Liu, Donahue G. , Liska A. та Paul Albitz, документація, яка опублікована IETF в документах RFC 8499, 1034, 2929 та ін.

Метою дослідження є дослідити загальні відомості, принципи, терміни та практичне налагодження систем доменних імен з використанням емуляторів мережевого програмного забезпечення, такого як Cisco PacketTracer, Graphical Network Simulator-3 та системи віртуалізації операційних систем - VitrualBox. Також основною метою дослідження є демонстрація важливості підтримки безпеки DNS на підприємстві та на кінцевих пристроях.

Одним із завдань дипломного проекту є отримати в результаті налагоджену комп'ютерну мережу з працюючою системою доменних імен та звіти з програм, які надають діагностичну інформацію про цю систему. Другим завданням є оволодіння навичками роботи з програмами дослідження доменної системи імен. Третім завданням є набування навичок сканування мережі, набуття навичок клонування веб сторінок та здобуття навичок здійснення атак на DNS, а в особливості, виконання DNS Spoofing атак. Після виконання згаданих вище завдань, завданням є набуття навичок запобігання атак на DNS та навичок забезпечення безпеки доменної системи імен.

Об'єктом дослідження є безпосередньо протокол DNS та пов'язані з ним механізми захисту інформації. Виконання дослідження буде виконано на базі типових мереж малого підприємства та малої домашньої мережі.

Дослідження буде проводитись за допомогою системного підходу. В даному випадку, система є штучною, динамічною та сумативною, через те, що система існує лише в цифровому вигляді, може постійно розширюватися та компоненти системи існують та працюють у певному порядку. Опис окремих елементів вже було здійснено раніше, проте підхід об'єднання багатьох прийомів в одній комплексній системі використано вперше. Також, в ході досліджень використано сучасні підходи, станом на 2021 рік.

		Лещенко Б.С.				Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

РОЗДІЛ 1. ТЕОРЕТИЧНІ ВІДОМОСТІ, ДОСЛІДЖЕННЯ ПРИНЦИПІВ РОБОТИ ТА ПРИНЦИПІВ ПРОВЕДЕННЯ АТАК НА СИСТЕМУ ДОМЕННИХ ІМЕН

1.1. Загальні відомості про систему DNS, структура системи доменних імен та структура пакету

Акронім **DNS** розшифровується як – «*Domain Name System*», що в перекладі з англійської мови – «*Система доменних імен*», також в деяких випадках DNS можна розшифрувати як – «*Domain Name Servers*». DNS це ієрархічна, розподілена база даних, яка використовується для передачі інформації про доменні іменна. Розбіжності в скороченні демонструє труднощі з документацією DNS.

Метафора, якою найчастіше описують DNS, це дерево. DNS має корінь та різні **Домени верхнього рівня** (*Top Level Domains* або *TLDs*) схожі на гілки, що відходять від кореня. Кожна гілка має менші гілки, які є **доменами другого рівня** (*Second Level Domains* або *SLDs*), а листки - це **повні доменні імена** (*Fully Qualified Domain Name – FQDN*), які іноді називають іменами хостів.

DNS важливий не тільки для функціонування інтернету, а також важливий для функціонування компаній будь-якого розміру. Не якісно налаштований DNS сервер може вплинути на функціонування всієї організації, а погано захищений сервер може сприяти легкому проникненню в мережу організації злоумисників. Навіть якщо організація добре захищена, DNS все ще залишається вектором для вчинення шкоди для організації.

В кінці 1960х Агентство передових оборонних дослідницьких проєктів США (*U.S. Department of Defense's Advanced Research Projects Agency* або *ARPA*) почало розробку ARPAnet – експериментальної великої комп'ютерної мережі, яка з'єднувала важливі дослідницькі організації в Сполучених Штатах Америки. Початкова ціль ARPAnet була дозволити державним підрядникам ділитися, на той час дефіцитними та дорогими, обчислювальними ресурсами комп'ютерів. З самого початку, як не дивно, користувачі ARPAnet використовували мережу для сумісної колаборації. Ця співпраця починалась з простого поширення файлів, програм,

		Леценко Б.С.				Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		8

обміну повідомленнями і закінчувалась сумісною розробкою та дослідженнями на віддалених комп'ютерах.

Набір протоколів керування передачею/Інтернет протокол (*Transmission Control Protocol/Internet Protocol* або *TCP/IP*) був розроблений на початку 1980х та швидко став стандартом в ARPAnet. Включення стеку TCP/IP в популярну, на той час, операційну систему BSD Unix сприяло демократизації інтернет мереж. BSD Unix був практично безкоштовним для університетів. Це означало що з'єднання TCP/IP до ARPAnet було значно доступніше, ніж всі технології, які використовувались раніше. Багато комп'ютерів, підключених до ARPAnet, також були підключені до локальних мереж (LAN), і дуже скоро інші комп'ютери в локальних мережах також підключалися через ARPAnet.

Мережа зросла з декількох хостів до десятків тисяч хостів. Початковий ARPAnet став основою конфедерації локальних та регіональних мереж на базі TCP/IP, що зараз називається мережею Інтернет.

Сьогодні Інтернет з'єднує мільйони комп'ютерів по всьому світу. Деякі комерційні магістралі мають об'єм у кілька гігабіт в секунду, що в десятки тисяч разів перевищує пропускну здатність оригінальної ARPAnet. Десятки мільйонів людей щодня використовують мережу для спілкування та співпраці.

Впродовж 1970х мережа ARPAnet була малим, дружнім співтовариством, яке налічувало всього декілька сотень хостів. Єдиний файл, **HOSTS.TXT**, вміщував в собі відповідності ім'я-адреса для кожного комп'ютера в мережі ARPAnet.

HOSTS.TXT підтримувався Мережевим інформаційним центром Stanford Research Institute (що отримав назву "NIC") і розповсюджувався з одного хоста, SRI-NIC. Адміністратори ARPAnet зазвичай надсилали свої зміни до NIC і періодично підключалися по FTP до SRI-NIC і викачували поточний файл HOSTS.TXT. Зміни в цей файл вносились раз або два на тиждень. Зі зростом популярності ARPAnet така схема використання стала занадто перевантаженою. Розмір файлу HOSTS.TXT ріс пропорційно до збільшення кількості комп'ютерів в мережі.

		Лещенко Б.С.				Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		9

Коли ARPAnet «переїхала» на стек TCP/IP, кількість хостів в мережі виросла в значній мірі, через це в системі стало безліч проблем з HOSTS.TXT:

— *Трафік та завантаження* – Плата за мережевий трафік та за навантаження на центральний процесор на SRI-NIC стала нестерпною.

— *Колізія імен* – Два комп'ютера в мережі не могли мати однакові назви. Однак, хоча NIC міг призначати адреси таким чином, щоб гарантувати унікальність, він не мав повноважень над іменами хостів. Ніщо не заважало комусь додати хост із суперечливим іменем і порушити всю схему. Наприклад, додавання хосту з тим самим іменем, що і головний поштовий концентратор, міг порушити роботу поштової служби на більшій частині ARPAnet.

— *Узгодженість* – Підтримування узгодженості у файлі з збільшенням розміру мережі ставало все важче і важче. До того часу як файл доходив до найвіддаленіших частин мережі, у файл вже могли вноситись нові зміни.

Основною проблемою було те, що механізм HOSTS.TXT не мав належного масштабування. Як не дивно, але успіх ARPAnet як експерименту призвів до невдачі та застарівання HOSTS.TXT.

Керівні органи ARPAnet почали дослідження для розробки наступника HOSTS.TXT. Їх метою було створити систему, яка вирішувала б проблеми, властиві єдиній системі таблиць хостів.

Нова система повинна була дозволити локальне адміністрування даних, але зробити ці дані загальнодоступними. Децентралізація адміністрування дозволить усунути проблему ефекту «пляшкового горла» коли базою керує один власник та полегшити проблему пропускну здатності та трафіку загалом. А локальне управління набагато полегшило б роботу з оновлення даних. Нова система повинна використовувати ієрархічний простір імен для іменування хостів. Це забезпечило б унікальність імен.

Paul Mockapetris, який на той час працював в Інституті інформаційних наук США, відповідав за розробку архітектури нової системи. У 1984 році він випустив RFC 882 та 883, в яких описана система доменних імен. Ці RFC були замінені RFC

		Леценко Б.С.				Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

1034 та 1035, чинними специфікаціями системи доменних імен. RFC 1034 та 1035 з тих пір були доповнені багатьма іншими RFC, які описують потенційні проблеми безпеки DNS, проблеми реалізації, механізми для динамічного оновлення серверів імен та захисту даних зони тощо.

Система доменних імен – це розподілена база даних. Ця структура дозволяє локальний контроль за сегментами загальної бази даних, але дані кожного сегменту доступні по всій мережі по схемі клієнт-сервер. Надійність та адекватна продуктивність досягаються завдяки тиражуванню та кешуванню. Програми, що називаються **nameservers** (*сервери імен*), становлять серверну половину механізму клієнт/сервер DNS. Сервери імен містять інформацію про деякі сегменти бази даних і роблять цю інформацію доступною для клієнтів, що називаються **resolvers**. Резольвери часто є просто бібліотечними процедурами, які створюють запити та надсилають їх через мережу на сервер імен.

«Глибина» кореневого домена може складати максимально 127 рівнів, але навряд чи таке число рівнів коли-небудь знадобиться (див рис.1.1).

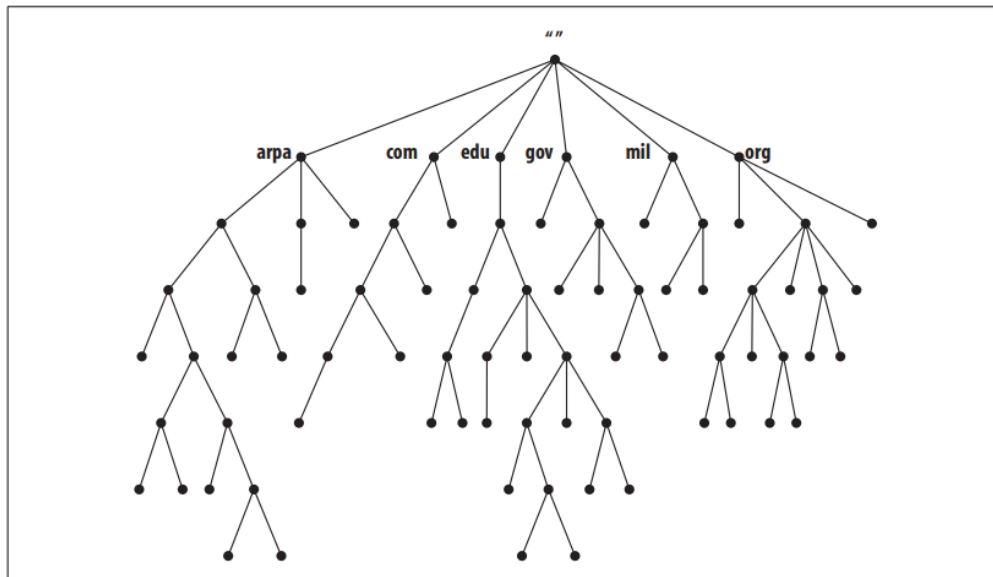


Рисунок 1.1 – Структура іменованого простору DNS

Структура бази даних DNS (див. рис. 1.2) схожа до структури Unix подібних файлових систем. Ціла база даних (або файлова система) зображена як інвертоване дерево з корневим вузлом вгорі. Кожен вузол в дереві має текстовий підпис, який ідентифікує вузол, що відноситься до батьківського. Це є приблизною аналогією до

відносного шляху в файлових системах. Перша мітка - нульова мітка або « » - зарезервована для кореневого вузла. У тексті кореневий вузол записується у вигляді однієї крапки (.). У файловій системі Unix корінь записується як коса риса (/).

Кожен вузол є також коренем нового піддерева загального дерева. Кожне з цих піддерев представляє розділ загальної бази даних – каталог(directory) у файловій системі Unix або домен(domain) у системі доменних імен. Кожен домен або каталог можна додатково розділити на додаткові розділи, які називаються *субдоменами (subdomains)* в DNS, як підкаталоги файлової системи. Субдомени, як і підкаталоги, зображуються як нащадки їх батьківських доменів.

Кожен домен має своє унікальне ім'я, як і кожен каталог. *Доменне ім'я* домену визначає його позицію в базі даних, подібно до того, як *абсолютний шлях* каталогу визначає його місце у файловій системі.

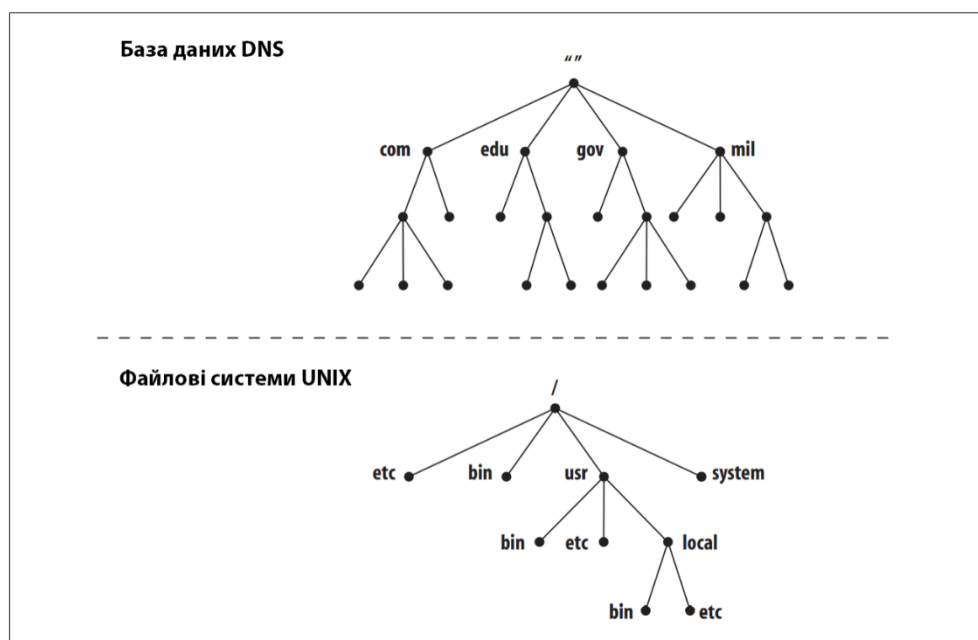


Рисунок 1.2 – Порівняння бази даних DNS і файлової системи Unix

В DNS, доменне ім'я це послідовність записів з кореневого вузла всього домену розділена крапками – «.». У файловій системі Unix абсолютним шляхом до каталогу є список відносних імен, що читаються від кореня до «листка» (навпаки як в DNS (див.рис.1.3)), використовуючи скісну риску для розділення імен.

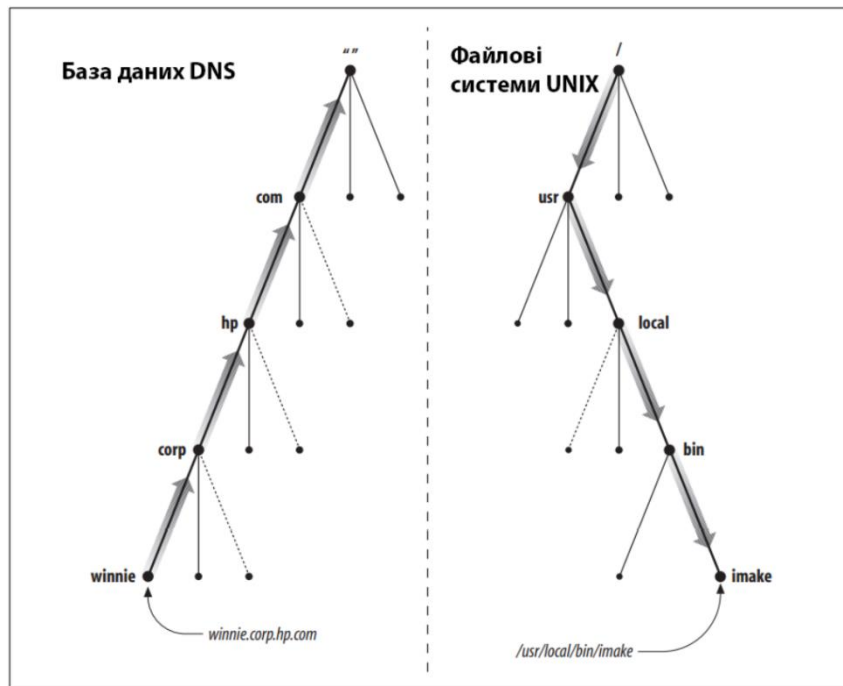


Рисунок 1.3 – Читання записів в базах даних DNS та файлових систем Unix

Також в DNS кожний домен може бути розбитий на субдомени, а відповідальність за ці субдомени може бути поділена між різними організаціями. Наприклад, організація з назвою EDUCAUSE керує *edu* (*educational* – навчальні) доменами, але відповідальність за домен *berkeley.edu* делегує Каліфорнійський університет Берклі (див. рис.1.4).

Це схоже на віддалене монтування файлової системи: певні каталоги у файловій системі насправді можуть бути файловими системами на інших хостах, змонтованих з віддалених хостів. Наприклад, адміністратор на хості *winken* відповідає за файловою системою, яка відображається на локальному хості як каталог */usr/nfs/winken*.

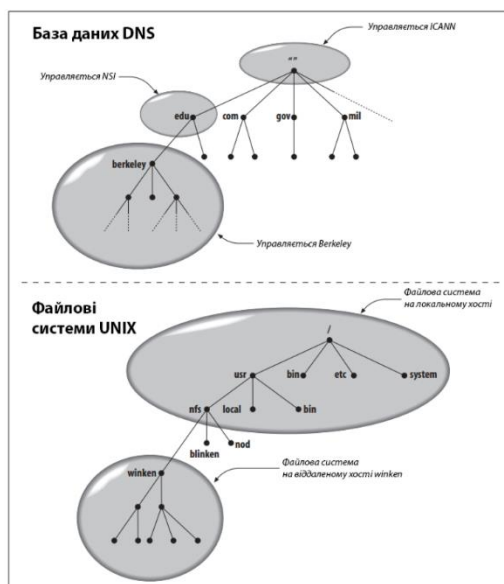


Рисунок 1.4 – Віддалене управління субдоменами та файловими системами

Делегування повноважень *berkeley.edu* до Берклі створює нову автономно-керовану зону. Зона *berkeley.edu* тепер незалежна від *edu* і вміщує всі доменні імена, що закінчуються на *berkeley.edu*. Зона *edu*, з іншого боку, вміщує лише доменні імена, що закінчуються на *edu*, але не делегують такими зонами як *berkeley.edu*. *berkeley.edu* в подальшому може бути розділений на субдомени, такі як *cs.berkeley.edu* і деякі з цих субдоменів самостійно бути окремими зонами, якщо адміністратори *berkeley.edu* проделегують відповідальність за ці зони на інші організації. Якщо *cs.berkeley.edu* це окрема зона, зона *berkeley.edu* не вміщує доменні імена що закінчуються на *cs.berkeley.edu* (див.рис.1.5).

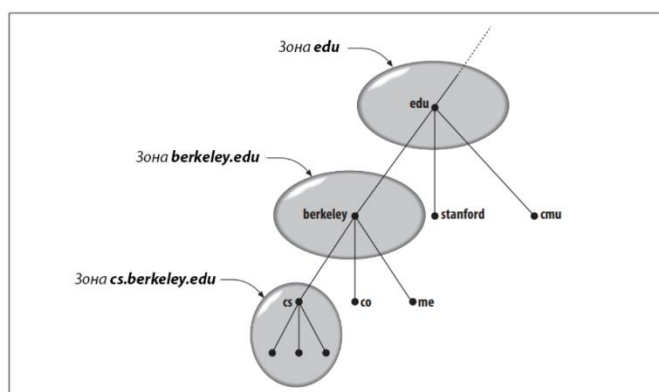


Рисунок 1.5 – Зони edu, berkeley.edu, та cs.berkeley.edu

Доменні імена використовуються як індекси в базі даних DNS. Кожний хост в мережі має доменне ім'я, яке вказує на інформацію про хост (див. рис.1.5.). Ця інформація може включати IP адреси, інформацію про маршрутизацію поштових

		Лещенко Б.С.				Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		14

сервісів та ін.. Хости також можуть мати один або більше *алиасів* (*domain name aliases*), які просто вказують з одного доменного імені до іншого (офіційного, канонічного) доменного імені. На рисунку 1.6 *mailhub.nv* є аліасом на канонічне ім'я *rincon.ba.ca*.

Структура складніша, ніж була з файлом HOSTS.TXT, але ця структура вирішує всі проблеми, що були з цим файлом. Наприклад, роблячи ієрархічну систему, ми позбуваємося проблем з колізією імен хостів. Кожний домен має унікальне доменне ім'я, отже організація що керує доменом може вільно обирати імена хостів та субдоменів в межах домену. Байдуже яке ім'я вони оберуть для хоста або субдомену, ці імена не будуть конфліктувати з іменами хостів в інших доменах.

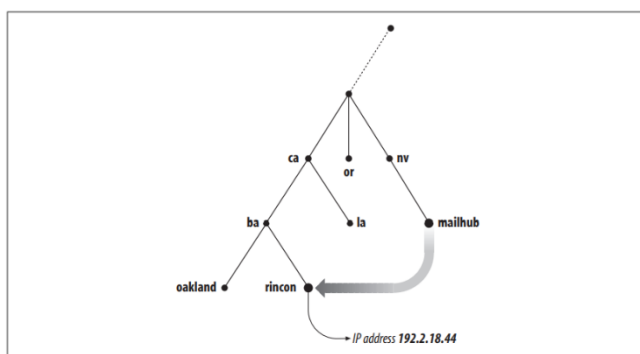


Рисунок 1.6 – DNS аліас який вказує на канонічне ім'я

Для кращого розуміння роботи доменної системи імен, необхідно розглянути структуру пакетів, за допомогою яких відбувається обмін інформацією між пристроями, які використовують технологію DNS. На рисунку 1.7 розглянуто структуру пакету DNS. А в таблиці 1.1 описано значення кожного поля.

+-----+		+-----+
	Header	Заголовок
+-----+		+-----+
	Question	Секція запитів
+-----+		+-----+
	Answer	Секція відповіді
+-----+		+-----+
	Authority	Секція відповіді про уповноважених серверах
+-----+		+-----+
	Additional	Секція відповіді додаткових записів
+-----+		+-----+

Рисунок 1.7 – Структура пакета DNS

		Лещенко Б.С.				Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		15

Таблиця 1.1 – Опис елементів пакету DNS

Поле	Значення
Header	Заголовок DNS пакета, що складається з 12 октетів.
Question	В цій секції DNS-клієнт передає запити DNS-серверу повідомляючи про те, для якого імені необхідно вирішити (зарезолвити) запис DNS, а також якого типу (NS, A, TXT і т.д.). Сервер при відповіді, копіює цю інформацію і віддає клієнту назад в цій же секції.
Answer	Сервер повідомляє клієнтові відповідь або кілька відповідей на запит, в якому повідомляє вищевказані дані.
Authority	Містить відомості про те, за допомогою яких авторитетних серверів було отримано інформацію що включена в секцію DNS-відповіді.
Additional	Додаткові записи, які відносяться до запиту, але не є строго відповідями на питання.

Записів в секціях може бути декілька, а може не бути взагалі. Все залежить від заголовку (див.рис.1.8).

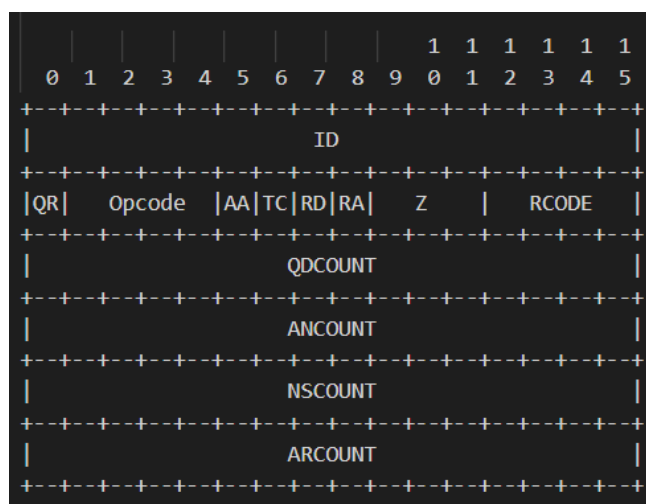


Рисунок 1.8 – Структура заголовку DNS

Таблиця 1.2 – Опис елементів заголовку DNS

Поле	Кількість біт	Значення
ID	16	Дане поле використовується як унікальний ідентифікатор транзакції. Вказує на те, що пакет належить одній і тій же сесії "запитів-відповідей" і займає 16 біт.
QR	1	Даний біт служить для ідентифікації того, чи є пакет запитом (QR = 0) або відповіддю (QR = 1).
Opcode	4	За допомогою даного коду клієнт може вказати тип запиту, де звичайне значення: 0 - стандартний запит, 1 - інверсний запит, 2 - запит статусу сервера. 3-15 - зарезервовані на майбутнє.
AA	1	Дане поле має сенс тільки в DNS-відповідях від сервера і повідомляє про те, чи є відповідь авторитетною або ні.
TC	1	Даний прапор встановлюється в пакеті відповіді в тому випадку якщо сервер не зміг помістити всю необхідну інформацію в пакет через існуючі обмеження.
RD	1	Цей однобітовий прапор встановлюється в запиті і копіюється у відповідь. Якщо прапор встановлюється в запиті - це значить, що клієнт просить сервер не повідомляти йому проміжних відповідей, а повернути тільки IP-адресу.
RA	1	Відправляється тільки у відповідях, і повідомляє про те, що сервер підтримує рекурсію
Z	3	Є зарезервованими і завжди дорівнюють нулю.
RCODE	4	Це поле служить для повідомлення клієнтів про те, чи успішно виконаний запит або з помилкою. 0 - значить запит пройшов без помилок; 1 - помилка пов'язана з тим, що сервер не зміг зрозуміти форму запиту; 2 - ця помилка з некоректною роботою сервера імен; 3 - ім'я, яке дозволяє клієнт не існує в даному домені; 4 - сервер не може виконати запит даного типу; 5 - цей код означає, що сервер не може задовольнити запит клієнта в силу адміністративних обмежень безпеки.
QDCOUNT	16	Кількість записів в секції запитів
ANCOUNT	16	Кількість записів в секції відповіді
NSCOUNT	16	Кількість записів в Authority Section
ARCOUNT	16	Кількість записів в Additional Record Section

1.2. Типи записів доменної системи імен

Файли зони (zone files) це файли, які містять в собі «простий» текст, та зберігають інформацію про домен. Оригінальний файл зберігається на головному сервері (мастер сервері) імен, а ведений сервер (slave сервер) отримує копію файлу зони і зберігає її локально. Рекурсивні сервери посилають запит на отримання інформації з файлу, але не зберігають локальні копії; рекурсивні сервери просто збирають стільки інформації, скільки авторитетні сервери імен будуть ділитися.

```
1 $ttl 43200
2 example.com. IN SOA ns1.example.com. dns.example.com. (
3 2003040401
4 10800
5 3600
6 432000
7 43200)
8 example.com. IN NS ns1.example.com.
9 ; The servers
10 www IN CNAME example.com.
11 ftp IN A example.com.
12 mail 60 IN A 10.100.0.10
13 example.com. IN A 10.100.0.12
14 ; The name servers
15 example.com. IN NS ns1.foo.com.
16 example.com. IN NS ns2.example.com.
17 ; The mail servers
18 example.com. IN MX 10 mail.example.com.
19 example.com. IN MX 50 mail.foo.com.
```

Рисунок 1.9 – Приклад вигляду файлу зони

На рис. 1.9 зображений доволі стандартний файл зони, який включає найбільш поширені сервіси: веб, поштові сервіси та FTP. Більш розширений файл зони може включати в себе список робочих станцій або додаткових серверів. Деякі файли зон можуть бути довжиною в тисячі рядків записів.

Перша лінія в файлі зон відповідає стандартному **часу для «життя»** (*Time to Live – TTL*) в секундах для зони. Цей час буде використано для всіх записів в файлі, якщо в самому записі не вказано інше. Стандартний час «життя» дорівнює 12 годинам.

		Леценко Б.С.				Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		18

Друга лінія насправді є «серцем» файлу зони, і засобом, який дозволяє різним серверам імен спілкуватися один з одним – це запис **початку повноважень** (*Start of Authority – SOA*). Запис початку повноважень вказує рекурсивним серверам імен як довго будуть зберігатися в хеші, це полегшує зв'язок між головним і веденим серверами імен.

Domain Name Class Resource Record Origin Contact

Рисунок 1.10 – Друга лінія в файлі зони вміщує однакову інформацію для всіх доменів

Перше поле в цій лінії це є сам домен, друге поле це клас (завжди IN – скорочення від Internet), третє поле це запис ресурсу (Resource Records – завжди SOA). Наступне поле це Origin Server (являється головним сервером для цього домену – сервер з якого поступає інформація). Інформація в цьому полі є однаковою як на головному сервері так і на підлеглому серверах, і вона повинна посилатися на канонічне ім'я, а не на CNAME чи IP-адресу. Останнє поле є контактною інформацією людей або команди відповідальних за обслуговування цього файлу зони. В записі контакту вказується адреса електронної пошти, але замість «@» записується «.», тому при читанні контактної інформації важливо пам'ятати, що замість першої крапки потрібно підставляти знак «@».

Рядки з 3 по 6 призначені для використання веденими серверами. Третій рядок є серійним номером, це поле повинно бути оновлене кожний раз коли на головному сервері в файлі зони виконуються зміни. Стандартний формат запису – YYYYMMDDNN, другі формати також підходять, але правило наступне: кожного разу після внесення змін це число має бути більше за попереднє.

Четвертий рядок це час оновлення виражений в секундах. Це поле говорить веденим серверам як часто перевіряти оновлення інформації з головного серверу. Коли цей час приходить, ведені сервери надсилають запит на головний сервер і перевіряють чи збільшився серійний номер у файлі. В загальному, це значення

		Лещенко Б.С.				Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		19

цього поля встановлюється як найменше, для того, щоб всі сервери мали релевантну інформацію в записах.

П'ятий рядок містить поле «повтору». Це кількість часу, виражена в секундах, це час, який ведені сервери буду очікувати перед відправкою нового запиту до головного серверу, в разі відсутності відповіді минулого разу. Варто пам'ятати, що протокол який DNS використовує по замовчуванню – це UDP. UDP це протокол ненадійної передачі, ведені сервери надсилають запит та не знають, чи отримав його головний сервер, а головний сервер не знає, чи ведені сервери отримують інформацію, надіслану їм. Поле повторної спроби компенсує цей недолік UDP, надсилаючи періодично запити.

Шостий рядок відповідає за час, який ведений сервер буде надсилати повторні запити в разі відсутності відповіді від головного серверу. Після сплину цього часу ведений сервер зітре всю інформацію про доменне ім'я. Головним завданням цього поля є запобігання накопичення старих та неправильних записів в файлі.

Сьомий рядок відноситься лише до рекурсивних серверів. Інформація, що зберігається в даному полі це TTL по замовчуванню. Це значення повинно бути таким же як і у першому рядку.

Вся інша частина файлу зони складається з записів ресурсів DNS.

Записи DNS, або **записи ресурсів** (англ. *Resource Records, RR*) — одиниці зберігання і передачі інформації в DNS. Ім'я та IP-адреса не тотожні — одна IP-адреса може мати безліч імен, що дозволяє підтримувати на одному комп'ютері безліч веб-сайтів (це називається віртуальний хостинг). Зворотне теж справедливо — одному імені може бути зіставлено безліч IP-адрес: це дозволяє створювати балансування навантаження. Для підвищення стійкості системи використовується безліч серверів, що містять ідентичну інформацію, а в протоколі є засоби, що дозволяють підтримувати синхронність інформації, розташованої на різних серверах. Існує 13 корневих серверів, їх адреси практично не змінюються.

		Лещенко Б.С.				Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		20

Протокол DNS використовує для роботи **TCP-або UDP-порт 53** для відповідей на запити. Традиційно запити та відповіді відправляються у вигляді однієї UDP дейтаграми. TCP використовується для AXFR-запитів.

Кожний ресурсний запис складається з таких полів:

— *ім'я (NAME)* — доменне ім'я, до якого прив'язана або яким «належить» даний ресурсний запис;

— *TTL (Time To Live)* — допустимий час зберігання даного ресурсного запису в кеші не відповідального DNS-сервера (вимірюється в секундах);

— *тип (TYPE)* ресурсного запису — визначає формат і призначення даного ресурсного запису;

— *клас (CLASS)* ресурсного запису; теоретично вважається, що DNS може використовуватися не тільки з TCP / IP, але і з іншими типами мереж, код в поле клас визначає тип мережі;

— *довжина поля даних (RDLEN)*;

— *поле даних (RDATA)*, формат та зміст якого залежить від типу запису.

`mail.example.com. 3600 IN A 10.10.100.102`

Рисунок 1.11 – Приклад типового A запису DNS

Найпоширенішим типом записів є **Address** або **A** запис. **A** запис містить відповідність *FQDN (Fully Qualified Domain Name)* до IP адреси. Кожна хост адреса в межах домену повинна мати A запис або Canonical Name Record (розглянемо пізніше).

Варто пам'ятати, що поле TTL не є обов'язковим для заповнення, якщо ресурсний запис може використовувати стандартний TTL для зони, тоді нема потреби додавати його в ресурсний запис. **A** запис є основним з записів ресурсів, і, безсумнівно, найбільш використовуваним. Навіть простий файл зони має зазвичай найменше 5-6 записів.

Якщо FQDN вказує одразу на декілька IP адрес, сервер імен поверне циклічно. Тобто, перший сервер який запитує цей **A** запис отримає перший запис,

		Лещенко Б.С.				Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		21

другий сервер отримає другий і т.д.. Коли сервер імен поверне всі записи, то він почне з самого початку.

Також є розроблений **A** запис спеціально для адрес **IPv6**. Ці записи називаються **AAAA** та мають такий же формат як **A** записи, тільки вміщують в собі адресу в **IPv6** форматі.

```
ipv6.example.com. 3600 IN AAAA 2001:468:504:1:210:5aff:fela:11e
```

Рисунок 1.12 – Приклад типового AAAA запису DNS

Наступний вид запису ресурсу є **записи канонічних імен** (*Canonical Name – CNAME*). Ці записи є аліасами (посиланнями) з одного FQDN на інший FQDN. Замість того, щоб прописувати FQDN до IP адреси є набагато простішим прописати його до іншого хоста. Це особливо корисно коли багато FQDN вказують на туж саму IP адресу. Коли вносяться зміни до первинної IP адреси, всі інші адреси оновлюються автоматично.

```
www.example.com. 3600 IN CNAME mail.example.com.  
mail.example.com. 3600 IN A 10.10.100.102
```

Рисунок 1.13 – Приклад типового CNAME запису DNS

В свою чергу, щоб **CNAME** працював він має вести на валідний **A** запис. **CNAME** не обов'язково повинен вказувати на FQDN в межах одного файлу зони, він також може вказувати на інші FQDN. Але, знову таки, на цьому FQDN повинен бути валідний **A** запис. Інша важлива річ про запис **CNAME**, це те що вони не можуть бути використані іншими записами ресурсів (RR). Ні **MX**, ні **NS** записи не можуть вказувати на **CNAME** запис.

Також існує такий тип записів, як **записи для обміну електронною поштою** (*Mail Exchanger – MX*). Ці записи використовуються для визначення хостів, куди слід надсилати пошту для домену. Так само як **CNAME** записи, **MX** записи не обов'язково повинні вказувати на FQDN в межах домену, **MX** записи можуть

		Леценко Б.С.				Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		22

вказувати на будь-який хост, всередині або зовні домену. ***MX*** записи повинні вказувати на FQDN, що представлені ***A*** записом.

```
example.com. 3600 IN MX 10 mail.example.com.
example.com. 3600 IN MX 20 mail.foo.com.
```

Рисунок 1.14 – Приклад типового MX запису DNS

Числове значення біля запису використовується лише в MX записах. Це значення називають «вагою» запису і це значення впливає на пріоритетність відправки листів. В прикладі (див. рис. 1.14) пошта mail.example.com має перевагу над mail.foo.com. Коли поштовий клієнт намагається відправити лист на скриньку user@example.com, то агент спочатку буде намагатися надіслати лист через mail.example.com спочатку, а тільки потім, якщо лист не вдалось відправити, намагається відправити на mail.foo.com.

У додачу до реєстрації хостів в файлі зони, сервери імен мають бути записані до файлу зони, як **записи серверів імен** (*Name Server – NS*). Записи ***NS*** використовуються для переліку авторитетних серверів імен для домену. Записи ***NS***, як правило, є першими RR після ***SOA***.

```
example.com. IN NS ns1.example.com.
example.com. IN NS ns1.foo.com.
example.com. IN NS ns2.example.com.
```

Рисунок 1.15 – Приклад типового NS запису DNS

Так само як з ***MX*** та ***CNAME*** записами, ***NS*** записи не повинні вказувати на повне доменне ім'я, яке існує у файлі хосту, запис ***NS*** може вказувати на будь-який хост, що має достовірну інформацію про домен. Записи ***NS*** також слугують іншій меті. Для демонів DNS, які підтримуються, головний сервер імен буде надсилати оновлення файлу зони на сервери імен, перелічені в цьому файлі зони. Це, звичайно, сильно залежить від існуючої інфраструктури DNS.

Запис ***PTR*** (pointer) або запис покажчика зв'язує IP хоста з його канонічним ім'ям. Запит в домені in-addr.arpa на IP хоста в reverse формі поверне ім'я (FQDN) даного хост. Наприклад, (на момент написання), для IP адреси 192.0.34.164: запит

		Лещенко Б.С.				Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		23

запису **PTR** 164.34.0.192.in-addr.arpa поверне його канонічне ім'я referrals.icann.org. З метою зменшення обсягу небажаної кореспонденції (спаму) багато серверів-одержувачів електронної пошти можуть перевіряти наявність **PTR** запису для хоста, з якого відбувається відправлення. У цьому випадку **PTR** запис для IP адреси повинна відповідати імені відправляючого поштового сервера, яким він представляється в процесі SMTP-сесії.

SRV-запис (server selection) вказує на сервери для сервісів, використовується зокрема для Jabber і Active Directory.

```
allan    IN    TXT  "Hello World!"
```

Рисунок 1.16 – Приклад типового TXT запису DNS

Також є інший тип записів – **текстові записи** (*Text – TXT*). Вони не використовуються часто та містять інформацію для подальшого читання людьми. Одна область, де записи TXT все ще активно використовуються, - це сфера шкідливого програмного забезпечення на основі DNS. Зокрема, шкідливе програмне забезпечення, яке використовує DNS як шлях exfil, дані та команди часто вбудовуються в запити DNS як записи TXT.

Ще один тип – **CAA** (*Certification Authority Authorization*) – це новий тип DNS записів, призначений для визначення центрів сертифікації, яким дозволений випуск SSL/TLS сертифікатів для відповідного доменного імені або субдомену. Найбільші та найбільш популярні центри сертифікації домовились, що починаючи з 8 вересня 2017 року в обов'язковому порядку слідувати інструкціям, вказаним в **CAA** записах доменного імені або субдомену для якого запитується випуск сертифікату. Використання **CAA** записів дозволить підвищити рівень безпеки в мережі і зменшити випадки неавторизованого отримання сертифікатів для сторонніх доменних імен.

1.3. Дослідження роботи протоколу DNS за допомогою додатків Nslookup, dig, WireShark та tcpdump

		Лещенко Б.С.				Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		24

Програмний компонент NSlookup (див. рис.1.17) зазвичай є вбудований в сучасні операційні системи, тому що входить в пакет TCP/IP. NSlookup виводить інформацію, яку можна використовувати для діагностування інфраструктури NSlookup. Для правильного прочитання інформації необхідне розуміння роботи протоколу DNS. Розроблена у складі пакету BIND (для UNIX-систем). Утиліта портована на Windows безпосередньо фірмою Microsoft і поставляється разом з операційною системою.

```
PS C:\Users\zogyy> nslookup
Default Server: UnKnown
Address: 192.168.0.1

> help
Commands: (identifiers are shown in uppercase, [] means optional)
NAME - print info about the host/domain NAME using default server
NAME1 NAME2 - as above, but use NAME2 as server
help or ? - print info on common commands
set OPTION - set an option
    all - print options, current server and host
    [no]debug - print debugging information
    [no]d2 - print exhaustive debugging information
    [no]defname - append domain name to each query
    [no]recurse - ask for recursive answer to query
    [no]search - use domain search list
    [no]vc - always use a virtual circuit
    domain=NAME - set default domain name to NAME
    srchlist=N1[/N2/.../N6] - set domain to N1 and search list to N1,N2, etc.
    root=NAME - set root server to NAME
    retry=X - set number of retries to X
    timeout=X - set initial time-out interval to X seconds
    type=X - set query type (ex. A,AAAA,A+AAAA,ANY,CNAME,MX,NS,PTR,SOA,SRV)
    querytype=X - same as type
    class=X - set query class (ex. IN (Internet), ANY)
    [no]mxfr - use MS fast zone transfer
    ixfrver=X - current version to use in IXFR transfer request
server NAME - set default server to NAME, using current default server
lserver NAME - set default server to NAME, using initial server
root - set current default server to the root
ls [opt] DOMAIN [> FILE] - list addresses in DOMAIN (optional: output to FILE)
    -a - list canonical names and aliases
    -d - list all records
    -t TYPE - list records of the given RFC record type (ex. A,CNAME,MX,NS,PTR etc.)
view FILE - sort an 'ls' output file and view it with pg
exit - exit the program
```

Рис. 1.17 – Довідка Nslookup в ОС Windows 10

Інструмент NSlookup може працювати в двох різних режимах: інтерактивному та неінтерактивному.

Якщо потрібно виконати одноразову перевірку інформації, то рекомендовано користуватися неінтерактивним режимом (див.рис. 1.18). Першим параметром в цій команді передається назва або IP адреса хоста про якого нам потрібно отримати інформацію, а другим параметром передається назва або IP адреса DNS сервера. Якщо не використовувати другий аргумент, то буде використано стандартний DNS, який налаштовано в мережевих налаштуваннях.

		Леценко Б.С.				Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		25

```
PS C:\Users\zogyy> nslookup ztu.edu.ua
Server: UnKnown
Address: 192.168.0.1

Non-authoritative answer:
Name: ztu.edu.ua
Address: 195.69.223.138
```

Рисунок 1.18 – Неінтерактивний режим NsLookup

В разі якщо потрібно виконати перевірку багатьох записів рекомендовано використовувати інтерактивний режим (див.рис. 1.19). Для запуску інтерактивного режиму з вибором необхідного серверу імен необхідно першим параметром написати дефіс (-), а другим сам сервер імен, який потрібно. Якщо не писати параметри, то команда nslookup запуститься в інтерактивному режимі зі стандартними серверами імен.

```
PS C:\Users\zogyy> nslookup
Default Server: UnKnown
Address: 192.168.0.1

> ztu.edu.ua
Server: UnKnown
Address: 192.168.0.1

Non-authoritative answer:
Name: ztu.edu.ua
Address: 195.69.223.138

> set type=MX
> ztu.edu.ua
Server: UnKnown
Address: 192.168.0.1

Non-authoritative answer:
ztu.edu.ua MX preference = 5, mail exchanger = ALT2.ASPMX.L.GOOGLE.COM
ztu.edu.ua MX preference = 1, mail exchanger = ASPMX.L.GOOGLE.COM
ztu.edu.ua MX preference = 10, mail exchanger = ALT3.ASPMX.L.GOOGLE.COM
ztu.edu.ua MX preference = 10, mail exchanger = ALT4.ASPMX.L.GOOGLE.COM
ztu.edu.ua MX preference = 5, mail exchanger = ALT1.ASPMX.L.GOOGLE.COM
>
```

Рисунок 1.19 – Інтерактивний режим NsLookup

Окрім параметрів вибору серверу імен з якого буде виконано перевірку, також корисним параметром команди nslookup є параметр **set**, завдяки якому можна, в тому числі, обирати тип необхідних для перевірки записів. На рисунку 1.19 зображено вибір для перевірки MX записів та їх дослідження, на якому видно поштові сервери для вказаного домену.

Варто звернути увагу на те, що команда Nslookup є регістрозалежною.

dig — утиліта Unix-систем, що надає користувачеві інтерфейс командного рядка для звернення до системи DNS (DNS-клієнт). Dig англійською значить

		Леценко Б.С.				Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		26

«копати», але формально — це скорочення від «domain information groper». Дозволяє задавати різні типи запитів і запитувати довільно вказувані сервера. Якщо в аргументах не вказано DNS-сервер, то буде використано перший сервер з файлу `/etc/resolv.conf`. Є аналогом **host** (див.рис. 1.20) та **nslookup**, але має більше можливостей ніж подібні утиліти.

```
bodia@DESKTOP-24L2040:/mnt/c/Users/zogyy$ host -v ztu.edu.ua
Trying "ztu.edu.ua"
;; -->HEADER<-- opcode: QUERY, status: NOERROR, id: 63987
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
ztu.edu.ua.                IN      A

;; ANSWER SECTION:
ztu.edu.ua.                2820    IN      A      195.69.223.138

Received 44 bytes from 192.168.0.1#53 in 13 ms
Trying "ztu.edu.ua"
;; -->HEADER<-- opcode: QUERY, status: NOERROR, id: 64814
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 0

;; QUESTION SECTION:
ztu.edu.ua.                IN      AAAA

;; AUTHORITY SECTION:
ztu.edu.ua.                2513    IN      SOA      ns.ztu.edu.ua. vkm.ztu.edu.ua. 2019083026 10800 3600 604800 3600

Received 71 bytes from 192.168.0.1#53 in 11 ms
Trying "ztu.edu.ua"
;; -->HEADER<-- opcode: QUERY, status: NOERROR, id: 20325
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
ztu.edu.ua.                IN      MX

;; ANSWER SECTION:
ztu.edu.ua.                2616    IN      MX      1 ASPMX.L.GOOGLE.COM.
ztu.edu.ua.                2616    IN      MX      10 ALT1.ASPMX.L.GOOGLE.COM.
ztu.edu.ua.                2616    IN      MX      10 ALT0.ASPMX.L.GOOGLE.COM.
ztu.edu.ua.                2616    IN      MX      5 ALT1.ASPMX.L.GOOGLE.COM.
ztu.edu.ua.                2616    IN      MX      5 ALT2.ASPMX.L.GOOGLE.COM.

Received 146 bytes from 192.168.0.1#53 in 11 ms
```

Рисунок 1.20 – Результат виконання команди `host -v`

Команда **dig** має значно більше параметрів та стає справжнім інструментом для системних адміністраторів. Вивід команди із стандартними параметрами зображено на рисунку 1.21.

```
bodia@DESKTOP-24L2040:/mnt/c/Users/zogyy$ dig ztu.edu.ua

; <<>> DiG 9.16.1-Ubuntu <<>> ztu.edu.ua
;; global options: +cmd
;; Got answer:
;; -->HEADER<-- opcode: QUERY, status: NOERROR, id: 44569
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
EDNS: version: 0, flags:;; udp: 4096
;; QUESTION SECTION:
ztu.edu.ua.                IN      A

;; ANSWER SECTION:
ztu.edu.ua.                2561    IN      A      195.69.223.138

;; Query time: 9 msec
;; SERVER: 192.168.0.1#53(192.168.0.1)
;; WHEN: Sat Jan 16 13:44:25 EET 2021
;; MSG SIZE rcvd: 55

bodia@DESKTOP-24L2040:/mnt/c/Users/zogyy$
```

Рисунок 1.21– Результат виконання команди `dig`

Розглянемо запит (див.рис.1.22. а) та відповідь (див.рис.1.22. б) детальніше.

		Леценко Б.С.			Арк.
Змн.	Арк.	№ докум.	Підпис	Дата	27

a)

```
;; QUESTION SECTION:
ztu.edu.ua.                IN      A
```

б)

```
;; ANSWER SECTION:
ztu.edu.ua.                3199    IN      A      195.69.223.138

;; Query time: 10 msec
;; SERVER: 192.168.0.1#53(192.168.0.1)
;; WHEN: Sat Jan 16 14:30:39 EET 2021
;; MSG SIZE rcvd: 55
```

Рисунок 1.22 – а) Запит команди dig б) Відповідь

В запиті ми вказуємо хост до якого хочемо зробити запит та тип запису, інформацію про який хочемо отримати. Стандартно **dig** робить запит до **A** записів.

В відповіді отримуємо схожий запис, але з додатковими даними. Числовим значенням після назви хоста є TTL (час життя цього запису – стільки часу можна тримати значення в кеші до повторної перевірки). Слово **IN** означає **Internet**. Так склалось історично, це потрібно для розділення типів мереж. В наступній частині відповіді говориться про час відклику серверу, яка у серверу IP адреса (192.168.0.1), на котрий порт був відісланий запит (53, DNS порт по замовчуванню), коли запит був завершений і скільки байтів було в відповіді. Навіть при звичайному DNS запиті відбувається купа речей. Кожного разу, коли ви відкриваєте веб ресурс, браузер починає робити десятки таких запитів, в тому числі для завантаження всіх зовнішніх ресурсів, таких як картинки або скрипти. Кожний ресурс відповідає за мінімум один новий DNS запит, і якщо б DNS не був розрахований на сильне кешування, то генерація трафіку в мережі була б колосальною. На прикладі вище не зображено, що сервер 192.168.0.1 зв'язався з купою інших серверів. Для подібних цілей команда **dig** має параметр **+trace** , який дає можливість дізнатися всю послідовність запитів, якщо б інформація про хост не була б вже закешована.

		Лещенко Б.С.				Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		28

```

;; global options: +cmd
45042 IN NS e.root-servers.net.
45042 IN NS f.root-servers.net.
45042 IN NS g.root-servers.net.
45042 IN NS h.root-servers.net.
45042 IN NS a.root-servers.net.
45042 IN NS i.root-servers.net.
45042 IN NS j.root-servers.net.
45042 IN NS k.root-servers.net.
45042 IN NS l.root-servers.net.
45042 IN NS m.root-servers.net.
45042 IN NS b.root-servers.net.
45042 IN NS c.root-servers.net.
45042 IN NS d.root-servers.net.
45042 IN RRSIG NS 0 518400 20210128210000 20210115200000 42351 . p/PK3VbiXTxpSJS4SHMEbM2qukiV8tB8TAdGsItFD0P7B5FBUcVRBVGO FRjomBCy1sRpkH6
;; Received 525 bytes from 192.168.0.1#53(192.168.0.1) in 9 ms

ua. 172800 IN NS cd1.ns.ua.
ua. 172800 IN NS hol.ns.ua.
ua. 172800 IN NS in1.ns.ua.
ua. 172800 IN NS pch.ns.ua.
ua. 86400 IN DS 48309 13 2 D0456Df0EAB0D07D242284110722F577207B2F1FD41D487E956A54F1 039053C8
ua. 86400 IN RRSIG DS 8 1 86400 20210129090000 20210116040000 42351 . JqMwD.VBcXIADVy2D0dxx79L8rJ/vjkr9ZQncq8ThZs8kjZ6VEjicSMU QzboSmI+ppclQAZL
;; Received 625 bytes from 198.97.190.53#53(h.root-servers.net) in 40 ms

edu.ua. 36036 IN NS ba1.ns.ua.
edu.ua. 36036 IN NS hol.ns.edu.ua.
edu.ua. 36036 IN NS nix.ns.ua.
edu.ua. 36036 IN NS deneb.dfn.de.
edu.ua. 36036 IN DS 25028 13 2 D8268D6DCD6FC2D01A7889A2080B875D6D2FC992D316D634B96ACBC2 D22C9466
edu.ua. 36036 IN RRSIG DS 13 2 36036 20210215113100 20210116113100 35976 ua. fBh1JIA191Q/53ttHR6h+NMROFmMBxy/E1ED36LZtC4TckKxIvYx656 UuBLFe2XKw7A7
;; Received 795 bytes from 74.123.224.40#53(in1.ns.ua) in 179 ms

ztu.edu.ua. 28800 IN NS ns2.trifile.net.
ztu.edu.ua. 28800 IN NS ns.ztu.edu.ua.
DT4LB7GQVGBVICT4650T9MTKCBFU79H.edu.ua. 3600 IN NSEC3 1 1 10 F7D3F97033CB8549 1UWLS0AN6403B5T4SLHNSP986Q7WVL NS SOA RRSIG DNSKEY NSEC3PARAM
DT4LB7GQVGBVICT4650T9MTKCBFU79H.edu.ua. 3600 IN RRSIG NSEC3 13 3 3600 20210210150500 20210108150500 43634 edu.ua. Mc3ADN8b8F95xzQNDGts93dGmcCyHd3fweY3LhhaQPePs7Fw+s1tr
0751KFRRAE027DPSASKPI31MKCEMMPGR.edu.ua. 3600 IN NSEC3 1 1 10 F7D3F97033CB8549 1BUHQ8KEQK9950PRVU79EDD0314LkPMU TXT RRSIG
0751KFRRAE027DPSASKPI31MKCEMMPGR.edu.ua. 3600 IN RRSIG NSEC3 13 3 3600 20210219150500 20210109150500 43634 edu.ua. oXlup5LB0g05CnaSPxpv3ChKZMRPAhVdmV2PvcH9R0wZVsR+z058ss
;; Received 507 bytes from 195.47.253.5#53(hol.ns.edu.ua) in 8 ms

ztu.edu.ua. 600 IN A 195.69.223.138
ztu.edu.ua. 600 IN NS ns.ztu.edu.ua.
ztu.edu.ua. 600 IN NS ns2.trifile.net.
;; Received 116 bytes from 195.69.223.138#53(ns.ztu.edu.ua) in 9 ms

```

Рис. 1.23 – Результат виконання команди `dig ztu.edu.ua +trace`

Інформація виводиться в ієрархічній послідовності. З зображення вище наглядно видно важливість крапки вкінці dns ієрархії. Коріневі DNS сервери обслуговуються різними компаніями та державами по всьому світі. З самого початку їх було менше, а наразі, з ростом інтернету, їх стало 13 штук, але у кожного з серверів є десятки або сотні фізичних машин, які ховаються за однією IP адресою.

З самого верху «трейсу» знаходяться кореневі сервери, кожний визначений за допомогою NS запису. NS запис пов'язує доменне ім'я (в даному випадку, кореневий домен) з DNS сервером. Коли реєструється доменне ім'я в реєстратору імен, вони створюють NS записи.

В наступному блоці видно, що **dig** обрав випадковий кореневий сервер, і запросив у нього A запис для **ztu.edu.ua**. Видно IP адресу кореневого серверу (198.97.190.53), порт (53) та доменне ім'я (h.root-servers.net). Перевірити за допомогою PTR записів можна все тим же **dig** (див.рис. 1.24).

```

;; QUESTION SECTION:
;53.190.97.198.in-addr.arpa. IN PTR

;; ANSWER SECTION:
53.190.97.198.in-addr.arpa. 3600 IN PTR h.root-servers.net.

;; Query time: 370 msec
;; SERVER: 192.168.0.1#53(192.168.0.1)
;; WHEN: Sat Jan 16 15:45:28 EET 2021
;; MSG SIZE rcvd: 87

```

Рис. 1.24 – Виконання зворотного перетворення з IP адреси до доменного імені за допомогою команди `dig -x`

		Лещенко Б.С.			Арк.
Змн.	Арк.	№ докум.	Підпис	Дата	29

```

;; Global options: +cmd
45942 IN NS e.root-servers.net.
45942 IN NS f.root-servers.net.
45942 IN NS g.root-servers.net.
45942 IN NS h.root-servers.net.
45942 IN NS a.root-servers.net.
45942 IN NS i.root-servers.net.
45942 IN NS j.root-servers.net.
45942 IN NS k.root-servers.net.
45942 IN NS l.root-servers.net.
45942 IN NS m.root-servers.net.
45942 IN NS b.root-servers.net.
45942 IN NS c.root-servers.net.
45942 IN NS d.root-servers.net.
45942 IN RRSIG NS 8 0 518400 20210128210000 20210115200000 42351 . p/P3VbiXtApS3545MEEbZqzkiV8tB8TAdgaItFD0P705BUCvRBVG0 FrJeeBCy1sRpKHk
;; Received 525 bytes from 192.168.0.1#53(192.168.0.1) in 9 ms

ua. 172800 IN NS cdi.ns.ua.
ua. 172800 IN NS hol.ns.ua.
ua. 172800 IN NS inl.ns.ua.
ua. 172800 IN NS pch.ns.ua.
ua. 86400 IN DS 48349 13 2 D8456DF6EAD0D07D2422B411072F257D7821FD4D487E56A54F1 039053C8
ua. 86400 IN RRSIG DS 8 1 86400 20210129050000 20210116040000 42351 . JqMMDJVBCiVxJ9ZD0dxX9L0e/JvJkr9Zqc8qThZs8k/jZ0VEjIoSMU Qzbo5u1+ppclQAZV
;; Received 625 bytes from 198.97.190.53#53(hol.ns.ua) in 40 ms

edu.ua. 36936 IN NS bal.ns.ua.
edu.ua. 36936 IN NS hol.ns.ua.ua.
edu.ua. 36936 IN NS nix.ns.ua.
edu.ua. 36936 IN NS deneb.dfn.de.
edu.ua. 36936 IN DS 25028 13 2 D8268D60CD6FC2D01A7B89A2000BF75D62FC929D2316D634B96ABC2 D22C9466
edu.ua. 36936 IN RRSIG DS 13 2 36936 20210215113100 20210116113100 35976 ua. fBh1JIAI9J/53tthR6h+NMROfmBky/E1ED361ZtCtCkXbXivx656 Uu8LFE2XhW7AT
;; Received 795 bytes from 74.123.224.40#53(inl.ns.ua) in 179 ms

ztu.edu.ua. 28800 IN NS ns2.trifle.net.
ztu.edu.ua. 28800 IN NS ns.ztu.edu.ua.
DTILt87QVGBV4CtU650T9MTkBFUJ79H.edu.ua. 3600 IN NS3C1 1 10 F7D3F97823CB8549 IYNLSSDAK604385145LHKSP986Q7VWML NS SOA RRSIG DNSKEY NSEC3PARAM
DTILt87QVGBV4CtU650T9MTkBFUJ79H.edu.ua. 3600 IN RRSIG NS3C1 1 3 3600 20210119150500 20210105150500 43634 edu.ua. PcAdhNs8F95XcQNDGy593GhmcQyH3fXreY3LhhaQpP3F4P+s1tr
0751HFRRAE027DPSASWP131MC6MPGR.edu.ua. 3600 IN NS3C1 1 10 F7D3F97823CB8549 18uHqCQWSP9869VPE8ED0311uJLWU TAT RRSIG
0751HFRRAE027DPSASWP131MC6MPGR.edu.ua. 3600 IN RRSIG NS3C1 1 3 3600 20210119150500 20210105150500 43634 edu.ua. oXlup150Q05C4aSPXv93KChZMRPAhDhVd2PvcH9R8uZVs+r2058ss
;; Received 507 bytes from 195.47.253.5#53(hol.ns.ua) in 8 ms

ztu.edu.ua. 600 IN A 195.69.223.138
ztu.edu.ua. 600 IN NS ns.ztu.edu.ua.
ztu.edu.ua. 600 IN NS ns2.trifle.net.
;; Received 116 bytes from 195.69.223.138#53(ns.ztu.edu.ua) in 9 ms

```

Також за допомогою dig можливо зробити запит до публічного DNS серверу, замість стандартного (див.рис.1.25).

```
body@DESKTOP-24L2040:/mnt/c/Users/zogyy$ dig ztu.edu.ua @1.1.1.1 +trace
; <<> DiG 9.16.1-Ubuntu <<> ztu.edu.ua @1.1.1.1 +trace
; global options: +cmd

111375 IN NS a.root-servers.net.
111375 IN NS b.root-servers.net.
111375 IN NS c.root-servers.net.
111375 IN NS d.root-servers.net.
111375 IN NS e.root-servers.net.
111375 IN NS f.root-servers.net.
111375 IN NS g.root-servers.net.
111375 IN NS h.root-servers.net.
111375 IN NS i.root-servers.net.
111375 IN NS j.root-servers.net.
111375 IN NS k.root-servers.net.
111375 IN NS l.root-servers.net.
111375 IN NS m.root-servers.net.
111375 IN RRSIG NS 8 518400 2021012109050800 20210116040800 42351 .LYNa9+P+TAi+XgKc7SxLc3ElyeHRQWUWtGvR
4wc9VUW4z4Ag7lYRaFMSJdh jkeBxtosIv+ZuRrWzkuZz5v0dlyszfKpJVSYPWnhD+ZaFiZEKRP8tYk mQ7XULu2L5tFiK8spDp0lb14thxmfF6UeG6jxd8XBx+Hd4zZyW2MR9Y
sb w1J3G9vSke/6tQdVLWpQGNMazh46Lmw1o7B6ys28HV7/TCX5ZFzhYKHxk EzkAKmLoAEHxk1MPztf8jvN1AnoyoJlU/quivr8G1AJhc0ykJCFCWGuHn AKuqV9s4kvG06xpW
UteOxNrlr5FTTxyGtegnUjJcq2/wdY6S1JzSvAF qdBu1g==
; Received 525 bytes from 1.1.1.1#53(1.1.1.1) in 7 ms

ua. 172800 IN NS in1.ns.ua.
ua. 172800 IN NS cd1.ns.ua.
ua. 172800 IN NS hcl1.ns.ua.
ua. 172800 IN NS pch1.ns.ua.
ua. 86400 IN DS 493409 13 2 D8456DF6EA80BD7D2422B4118722F577D20BF1FD410487E956A54F1 039D53C8
ua. 86400 IN RRSIG DS 8 1 86400 20210129050800 20210116040800 42351 .JqMJDJVBXCtADVy2Ddxxv79l8rJ/vjkr9ZQ
ncq8ThZs8kVj26VEjoseMU Qzob5m1ppcLQAZLYjwmp5yCyoxZAXzFKTKA+kkhY+Lh1Axy8AVuLi 3XqS8b7lv4tNnARkLwnBqns3ydmrYCFEGVSL34EgvtJvJoNd+X+7ny75G
j LbFv34Vv1Kd08tqdxz9L9ydrFTIEyrcr+2cw5x7TJjsmZPf1ZJ98v7g 1XTABvhyfVrTzIpT0p8famVD16rSpHtnY2lmsJcXGqt8oSwQSE0wZns fbYnQQQFPpoNYTBRYd
JEY5f1Fk348tth81FCRLmBmq8Xe19G239pfHe nlnZNw==
; Received 387 bytes from 192.36.148.17#53(in root-servers.net) in 11 ms

edu.ua. 36936 IN NS ba1.ns.ua.
edu.ua. 36936 IN NS hcl1.ns.edu.ua.
edu.ua. 36936 IN NS n3x.ns.ua.
edu.ua. 36936 IN NS denab.dfn.de.
edu.ua. 36936 IN DS 25028 13 2 D8268D6CD6FC2D081A7889A208D8F75D6D2FC992D316D634B96ACB2 D22C9466
edu.ua. 36936 IN RRSIG DS 13 2 36936 20210215133100 20210116133100 35976 ua. iXbeqokKsW5zRkjhH8iGcyo0BXLqdb
6MXPLQjpspCdcBa+/WA/9ksq q90S1XM4en+y8ytQvQVb/MnctUNA==
; Received 795 bytes from 74.123.224.40#53(in1.ns.ua) in 177 ms
```

		Лещенко Б.С.				Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

З результату на зображенні вище можна спостерігати, що тепер запит йде не на кореневий сервер **h.root-servers.net.** , а на кореневий сервер **i.root-servers.net.** .

На рис. 1.25 та 1.26 в відповідях з серверів, окрім **NS** записів, є також **RRSIG** записи. RRSIG (RRset Signature) – в цих записах зберігається DNSSEC підпис для групи записів (одного або більше записів з однаковим іменем та типом). Обробники можуть перевірити підпис за допомогою відкритого ключа, що зберігається в записі **DNSKEY**.

В операційних системах Linux є вбудований сканер пакетів tcp та udp під назвою tcpdump.

Для початку запусимо браузер та зробимо запит без ввімкненого DNS over HTTPS та запусимо сканування пакетів на 53 порті.

```
ubuntu@ubuntu:~$ sudo tcpdump -n -i lo udp port 53 -xx -XX -vv
[sudo] password for ubuntu:
tcpdump: listening on lo, link-type EN10MB (Ethernet), capture size 262144 bytes
05:25:24.596157 IP (tos 0x0, ttl 64, id 56994, offset 0, flags [DF], proto UDP (17)
, length 75)
  127.0.0.1.54941 > 127.0.0.53.53: [bad udp cksum 0xfe7e -> 0xe914!] 60497+ [1au]
A? cabinet.ztu.edu.ua. ar: . OPT UDPsize=1200 (47)
  0x0000:  0000 0000 0000 0000 0000 0000 0800 4500  .....E.
  0x0010:  004b dea2 4000 4011 5dc9 7f00 0001 7f00  .K..@.].
  0x0020:  0035 d69d 0035 0037 fe7e ec51 0120 0001  .5...5.7.~.Q....
  0x0030:  0000 0000 0001 0763 6162 696e 6574 037a  .....cabinet.z
  0x0040:  7475 0365 6475 0275 6100 0001 0001 0000  tu.edu.ua.....
  0x0050:  2904 b000 0000 0000 00    ).....
05:25:24.596166 IP (tos 0x0, ttl 64, id 56995, offset 0, flags [DF], proto UDP (17)
, length 75)
  127.0.0.1.54941 > 127.0.0.53.53: [bad udp cksum 0xfe7e -> 0x68ed!] 27742+ [1au]
AAAA? cabinet.ztu.edu.ua. ar: . OPT UDPsize=1200 (47)
  0x0000:  0000 0000 0000 0000 0000 0000 0800 4500  .....E.
  0x0010:  004b dea3 4000 4011 5dc8 7f00 0001 7f00  .K..@.].
  0x0020:  0035 d69d 0035 0037 fe7e 6c5e 0120 0001  .5...5.7.~l^....
  0x0030:  0000 0000 0001 0763 6162 696e 6574 037a  .....cabinet.z
  0x0040:  7475 0365 6475 0275 6100 001c 0001 0000  tu.edu.ua.....
  0x0050:  2904 b000 0000 0000 00    ).....
05:25:24.597093 IP (tos 0x0, ttl 64, id 56996, offset 0, flags [DF], proto UDP (17)
, length 75)
```

Рисунок 1.27 – Сканування DNS за допомогою tcpdump.

Аналізуючи рис. 1.27 можна сказати, що DNS передає запити на 53 порті у відкритому вигляді, що не є безпечним.

Для того, щоб увімкнути DNS over HTTPS в браузері Mozilla Firefox необхідно перейти в Menu>Preferences>General>Network Settings>Settings... та знайти пункт «Enable DNS over HTTPS», поставити галочку та обрати один із доступних провайдерів (див.рис. 1.28).

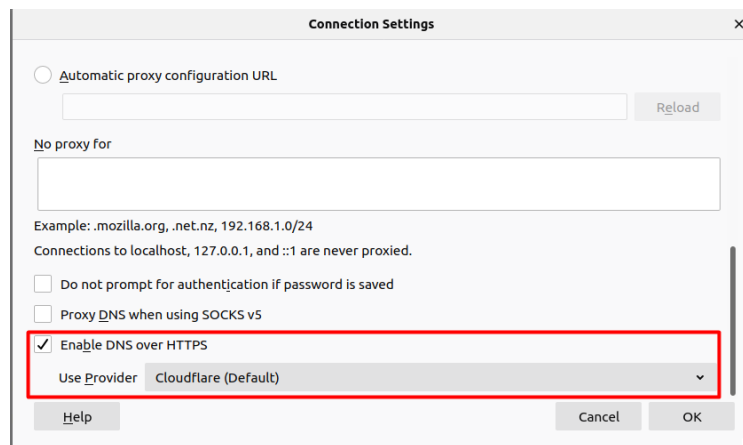


Рисунок 1.28 – Ввімкнення DoH в Mozilla Firefox.

Після ввімкнення DoH повторюємо сканування.

```
ubuntu@ubuntu:~$ sudo tcpdump -n -i lo udp port 53 -xx -XX -vv
tcpdump: listening on lo, link-type EN10MB (Ethernet), capture size 262144 bytes
05:36:25.725906 IP (tos 0x0, ttl 64, id 54557, offset 0, flags [DF], proto UDP (17), length 83)
127.0.0.1.49969 > 127.0.0.53.53: [bad udp cksum 0xfe86 -> 0x8a78!] 51031+ [1au] A? mozilla.cloudflare-dn
s.com. ar: . OPT UDPsize=1200 (55)
0x0000: 0000 0000 0000 0000 0000 0000 0800 4500 .....E.
0x0010: 0053 d51d 4000 4011 6746 7f00 0001 7f00 .S..@.gF.....
0x0020: 0035 c331 0035 003f fe86 c757 0120 0001 .5.1.S.?...W....
0x0030: 0000 0000 0001 076d 6f7a 696c 6c61 0e63 .....mozilla.c
0x0040: 6c6f 7564 666c 6172 652d 646e 7303 636f loudflare-dns.co
0x0050: 6d00 0001 0001 0000 2904 b000 0000 0000 m.....).....
0x0060: 00
05:36:25.726727 IP (tos 0x0, ttl 64, id 54558, offset 0, flags [DF], proto UDP (17), length 83)
127.0.0.1.49969 > 127.0.0.53.53: [bad udp cksum 0xfe86 -> 0x385c!] 6489+ [1au] AAAA? mozilla.cloudflare-
dns.com. ar: . OPT UDPsize=1200 (55)
0x0000: 0000 0000 0000 0000 0000 0000 0800 4500 .....E.
0x0010: 0053 d51e 4000 4011 6745 7f00 0001 7f00 .S..@.gE.....
0x0020: 0035 c331 0035 003f fe86 1959 0120 0001 .5.1.S.?...Y....
0x0030: 0000 0000 0001 076d 6f7a 696c 6c61 0e63 .....mozilla.c
0x0040: 6c6f 7564 666c 6172 652d 646e 7303 636f loudflare-dns.co
0x0050: 6d00 001c 0001 0000 2904 b000 0000 0000 m.....).....
0x0060: 00
```

Рисунок 1.29 – Сканування DNS за допомогою tcpdump після ввімкнення DoH.

Після ввімкнення DoH браузер почав відправляти DNS запити до обраного раніше провайдеру, але інформація про потрібний нам хост в цих запитах знаходиться в шифрованому вигляді.

Проведемо такі ж дослідження, але використовуємо Wireshark.

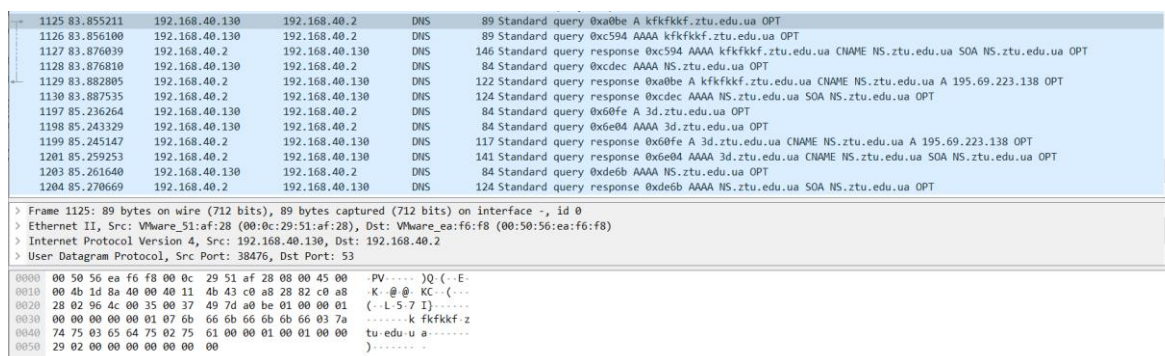


Рисунок 1.30 – Дослідження черги DNS запитів.

З вимкненим DoH запити надсилаються у відкритому вигляді.

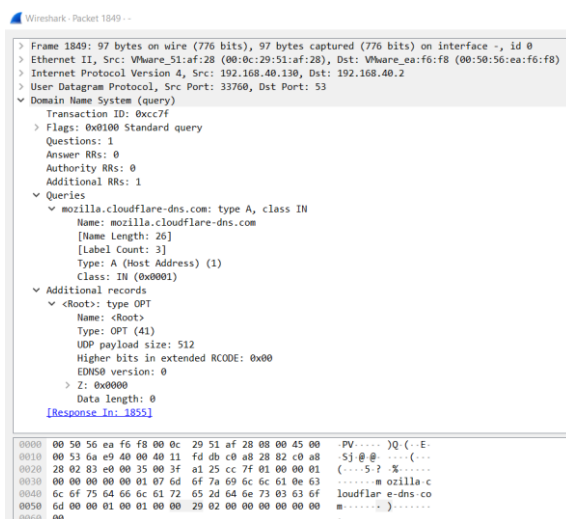


Рисунок 1.31 – Дослідження пакету DNS у ПЗ WireShark.

Отримана інформація аналогічна інформації TCPdump, але подана у більш наглядному вигляді.

З рис. 1.31 можна зробити висновок, що запити від клієнту надсилаються у шифрованому вигляді.

1.4. Розробка схеми комплексної атаки на DNS та опис етапів та програмного забезпечення для проведення атак

Перед виконанням будь яких дій по захисту комп'ютерних мереж, операційних систем, програмного забезпечення та всього іншого, що в своїй основі керується комп'ютерною логікою, необхідно визначити периметр для захисту та постаратися передбачити вектори всіх потенційних атак, які якимось чином можуть стосуватися вас або вашої організації.

Для максимального охопту всіх можливих дій зловмисників варто дослідити відомі сценарії експлуатації вразливостей.

Згідно інформації наданої компанією EfficientIP (компанія, яка займається автоматизацією та безпекою та спеціалізується на безпеці DNS), близько 79% компаній постачальників послуг в 2020 році стикалися з тими чи іншими атаками пов'язаними з DNS.

В звіті під назвою «IDC 2020 Global DNS Threat Report» сказано, що обізнаність щодо безпеки DNS стабільно зростає з року в рік, але середня кількість

		Лещенко Б.С.				Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		33

атак зростає, так само як і грошові втрати компаній від цих атак. Значний розвиток хмарних технологій безпосередньо вплинув на важливість справнопрацюючого DNS.

Збої в роботі доменної системи імен завдають шкоди, без перебільшення, всім сферам людської життєдіяльності починаючи сферою телекомунікацій і фінансовими сервісами і закінчуючи сферами освіти та охорони здоров'я. Окрім високої частоти атак, телекомунікаційні провайдери також стикаються з більш втратними атаками – близько 8% організацій заявили, що вони сумарно постраждали від збитків у розмірі понад 5 мільйонів доларів внаслідок атак на DNS.

Найбільш поширеними типами атак на DNS є фішинг (37%), шкідливе програмне забезпечення націлене на DNS (34%), DDoS атаки (27%), атаки з блокування доменів, які «виснажують» ресурси серверів доменної системи імен (22%) та атаки ампліфікації (підсилення), які в результаті можуть спричинити відмову роботи мережі компанії, тим самим викликати серйозні економічні втрати (21%) (див. рис. 1.32).

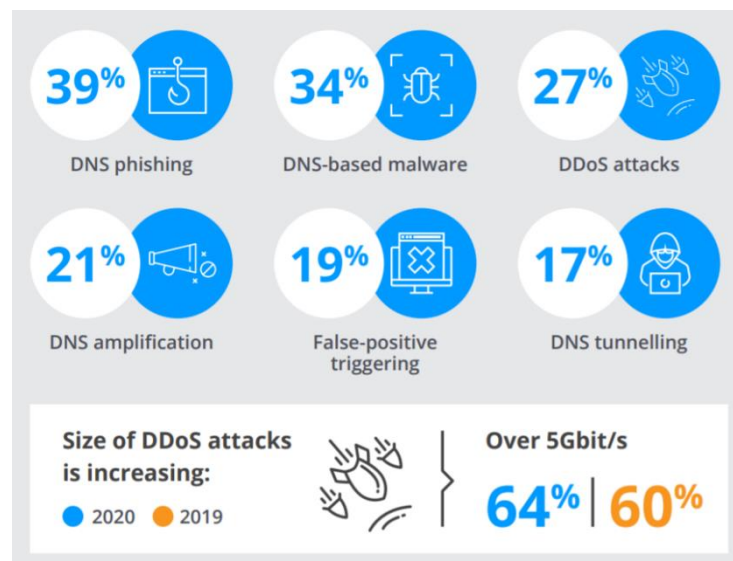


Рисунок 1.32 – Рейтинг популярності експлуатації типів атак пов’язаних з DNS

Огляд видів атак варто почати з атак з «отруєння» кешу (*DNS Cache Poisoning Attack*). Вона є однією з найчастіших атак, і її головна ціль це перенаправлення користувачів на підробні вебсайти. Користувач буде переходити по правильній адресі, але його все одно буде спрямовано на шахрайську сторінку.

		Леценко Б.С.				Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		34

Потенційні сторінки для підміни можуть бути абсолютно різними, починаючи від новинних ресурсів і до сайтів банківських установ. Все залежить від фантазії зловмисників, але, звичайно, найвигідніше підмінити ресурси, які пов'язані з фінансами та соціальні мережі з метою знаходження компромату.

Наступний вид атак це, так звані, **розподілені атаки з відмовою в роботі з «відбиттям»** (*Distributed Reflection Denial of Service*) (див. рис. 1.2). Суть цих атак полягає в тому, щоб зменшити доступність серверів жертв або взагалі призвести до повної відмови в роботі. З великої кількості керованих хостів надсилаються запити на сервер жертви. Використовуватись можуть різноманітні протоколи.

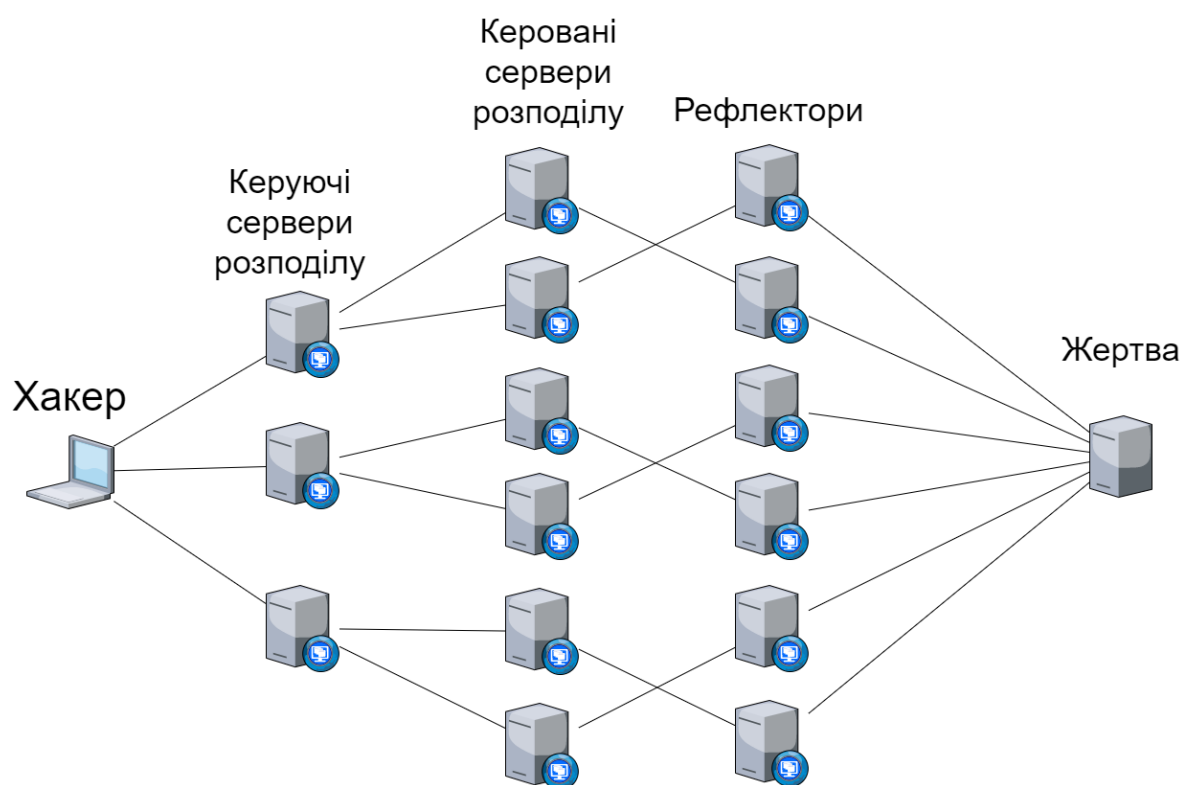


Рисунок 1.33 – Принцип проведення DRDoS атаки

Схожий тип атаки називається **атака «фантомними» доменами** (*Phantom domain attack*). В ході такої атаки зловмисник направляє на DNS сервер запити з доменами, яких не існує. Сервер пересилає ці запити далі та очікує відповіді. Якщо таких запитів багато, то це значно знижує продуктивність DNS серверу.

Ще один подібний вид DoS атаки – **TCP SYN Flood атака** (див.рис.1.3). Ця атака може бути застосована на будь якому обладнанні, що використовує протокол

		Лещенко Б.С.				Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		35

TCP. Як відомо, протокол TCP працює по системі потрійного рукоштовкання: клієнт передає SYN запит, потім сервер передає клієнту SYN/ACK відповідь, далі клієнт передає ACK на сервер. Суть TCP SYN Flood атаки полягає в тому, що злоумисник направиляє багато SYN пакетів на сервер, а ACK пакети не надсилає. В цьому випадку сервер очікує на ACK пакет та залишає порти в напіввідкритому стані. Якщо таких фіктивних SYN запитів безліч, то може статись така ситуація, що сервер не буде мати вільних портів для легітимних користувачів.

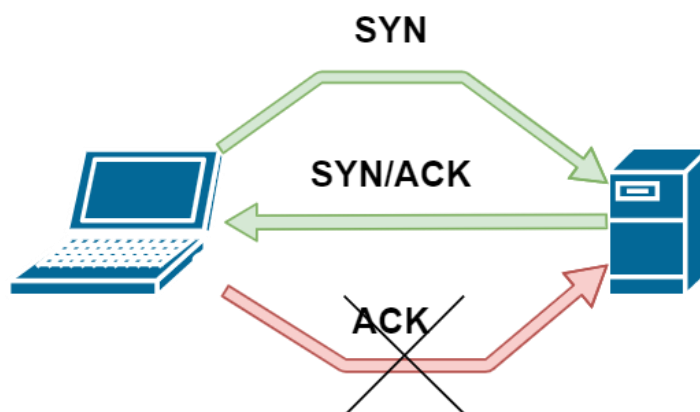


Рисунок 1.34 – Принцип проведення TCP SYN атаки

Досить широка сфера злочинності в кіберпросторі **фішинг** (*phishing*). Фішинг – шахрайські дії, спрямовані на виманювання реквізитів картки, логінів, паролів або інших персональних даних у власника. Зазвичай жертва сама добровільно повідомляє шахраям потрібну інформацію або ненавмисно надає цю інформацію.

В ході дипломного проекту буде розглянуто поєднання два найпопулярніших типів – фішингу та DNS spoofing. Для того, щоб краще візуалізувати схему всього процесу складено графічну покрокову схему (див. рис. 1.33).

Перед початком опису всіх процесів варто зазначити, що всі дії, які виконуються в курсовій роботі, є в дослідницьких цілях та не виконуються з метою нашкодити або перешкодити роботі систем або мереж. Всі виконані дії не в лабораторних умовах підпадають під статті Кримінального Кодексу України, а саме:

— несанкціоноване втручання в роботу електронно-обчислювальних машин (комп'ютерів), автоматизованих систем, комп'ютерних мереж чи мереж електрозв'язку (стаття 361 КК України);

— створення з метою використання, розповсюдження або збуту шкідливих програмних чи технічних засобів, а також їх розповсюдження або збут (стаття 361 КК України);

— несанкціоновані збут або розповсюдження інформації з обмеженим доступом, яка зберігається в електронно-обчислювальних машинах (комп'ютерах), автоматизованих системах, комп'ютерних мережах або на носіях такої інформації (стаття 361 Кримінального кодексу України);

— несанкціоновані дії з інформацією, яка оброблюється в електронно-обчислювальних машинах (комп'ютерах), автоматизованих системах, комп'ютерних мережах або зберігається на носіях такої інформації, вчинені особою, яка має право доступу до неї (стаття 362 Кримінального кодексу України);

— перешкоджання роботі електронно-обчислювальних машин (комп'ютерів), автоматизованих систем, комп'ютерних мереж чи мереж електрозв'язку шляхом масового розповсюдження повідомлень електрозв'язку (стаття 363 Кримінального кодексу України).

Першим кроком проведення будь якої атаки є розвідка. В нашому випадку необхідно зробити повне сканування мережі організації та визначити IP адреси основного шлюзу та IP адрес майбутніх жертв атаки.

Наступним кроком є вибір цільового сайту з якого жертви будуть переправлятися на підробний та, безпосередньо, проведення його копіювання.

Третім кроком буде запуск вебсерверу з підробним вебсайтом.

Четвертим та фінальним кроком є проведення DNS Spoofing атаки.

Детальний опис кроків проведення атаки описано в наступних підрозділах.

		Леценко Б.С.				Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		37



Рисунок 1.34 – Схема виконання атаки на DNS

Першим етапом в проведенні нашої атаки є здійснення **сканування мережі** (див. рис.1.35). Сканування мережі це процедура з ідентифікації активних пристроїв в мережі з задіянням функціональності (або функціональностей) в мережевих протоколах по відправці сигналів активним пристроям та очікування від них відповідей. Більшість сканувань мереж сьогодні застосовується для моніторингу та управління, але сканування також використовується частини мережі або користувачів мереж для здійснення атак. Для сканування використовуються різні протоколи, але зазвичай надсилається просте повідомлення (повідомлення ping, наприклад) на всі можливі IP адреси в визначеному діапазоні, а вже тоді використовуються інші протоколи для одержання більше відомостей про пристрої.

Сканування мережі



Рисунок 1.35 – Типи сканування мережі

		Леценко Б.С.				Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		38

Коли сканування відбувається в цілях моніторингу та управління системами, сканування використовується для ідентифікації поточних користувачів мережі, для визначення стану систем та пристроїв, і для обліку та інвентаризації елементів мережі. Часто отриманий список елементів мережі застосовується для порівняння з очікуваним списком пристроїв для оцінки наявності всіх комплектуючих. Це всі легітимні функції управління що зазвичай використовуються адміністраторами систем.

Сканування, яке використовують зловмисники, опирається на ті ж самі методи, програми та протоколи, що і для звичайного моніторингу та управління. Зловмисник спочатку отримає діапазон адрес, що належить організації. Потім адреси в межах визначеного діапазону будуть проскановані в пошуках серверів, їх операційних систем, системної архітектури, та сервісів, що на них працюють. Зловмисник може вдатися до спроб порушення безпеки цільових систем і додатків.

При **клонуванні вебсайтів** по суті копіюється front-end частина сайту (зовнішній вигляд), а сайт розміщують у домені, який запроектовано, щоб виглядати легітимно. Наприклад, замість посилання privatbank.ua зловмисники можуть використовувати посилання вигляду: privatbank.bankthatyoucantrust.com. Для недосвідчених користувачів різниця може бути не очевидною, або ж багато користувачів навіть не звернуть уваги на таке поле з посиланням у браузері.

Коли ж клонований сайт створено, і відвідувач, який не підозрює, що він на нелегітимному сайті, вирішить відвідати особистий кабінет та введе свої авторизаційні дані, то отримає лише повідомлення про невірно вказані дані. Після вводу даних вони одразу відправились шахраям, а користувача було переправлено на легітимний сайт. Користувач може і сам не помітити як перетворився в жертву.

Якщо коротко, то **веб хостинг** це процес оренди або купівлі місця для розміщення веб-сайтів в глобальній мережі. Контент веб-сайту, так як HTML, CSS, картинки та ін., повинен розміщуватися на сервері, щоб бути доступним для перегляду онлайн.

		Лещенко Б.С.				Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		39

Локальний хостинг веб-сайту це форма запуску власного веб-сайту або додатку самостійно налаштовуючи сервер та мережу.

Вибір підходящого та надійного веб хостинг провайдеру є важливим кроком до переходу в онлайн, але в рамках цього дипломного проекту для демонстрації важливості безпеки DNS нам буде достатньо розмістити сервер з сайтом локально. Проте, не варто забувати, що зловмисник з більшою вірогідністю, все таки, буде працювати віддалено в реальному житті.

Перед виконанням атаки «людина посередині» (MITM – Man in the middle) необхідно розібрати декілька концепцій. Перший – **sniffing** – це процес захоплення всього трафіку, що проходить в межах дротового та бездротового середовища мережі. Існує величезна кількість програмних додатків здатних проводити цей процес. Один із найвідоміших – WireShark, а також tcpdump, dsniff та ін. Для того, щоб переглядати трафік в мережі спочатку потрібно увімкнути «нерозбірливий» режим мережевої карти (режим монітору в безпроводних мережах). Цей режим дозволяє «ловити» весь мережевий трафік, а не тільки трафік призначений для власної MAC або IP адреси.

Друга концепція, яку потрібно розібрати – **spoofing**. На відміну від мережевих хабів, мережеві комутатори були вигадані для зменшення кількості трафіку в мережі. Якщо хаби надсилають трафік всім пристроям в мережі, то комутатори надсилають інформацію тільки для необхідних клієнтів мережі. Для здійснення цього в комутаторах знаходиться CAM (Content Addressable Memory) таблиця з відповідностями IP адрес до MAC адрес. Якщо змінити записи в цій таблиці, то можна успішно отримувати чийсь чужий трафік. Це називається ARP підміною, тому що записи в CAM таблиці надходять з ARP, що відсилає комутатор для збору інформації з мережевих інтерфейсів.

Програмного забезпечення для проведення сканувань мережі існує величезна кількість. Зазвичай програмне забезпечення виконує не лише сканування пристроїв мережі, але ще й йде в комплексі зі сканерами вразливостей кінцевих та мережевих пристроїв. Наприклад, **Acunetix Online** включає в себе

		Лещенко Б.С.				Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		40

повністю автоматизовані засоби мережевого сканування, які знаходять та репортують про більше ніж 50 тисяч, відомих на сьогоднішній день, вразливостей та некоректних налаштувань. Також цей додаток знаходить відкриті порти та працюючі сервіси, оцінює безпеку роутерів, фаєрволів, комутаторів, та ще багато речей.

Ще один популярний сканер – **ManageEngine OpUtils**. Теж дуже потужний додаток з великим функціоналом. Найкраще підходить для системних і адміністраторів безпеки малих або середніх, приватних або державних підприємств.

Але найбільш популярним сканером по праву є **NMAP**. Він є безкоштовним додатком з відкритим кодом, яким можна користуватися як в Windows так і в Linux. Програма Nmap або Network Mapper (див.рис.1.36) була розроблена Гордоном Луоном і на даний момент використовується спеціалістами по безпеці та системними адміністраторами по всьому світі.

Ця програма допомагає системним адміністраторам дуже швидко зрозуміти які комп'ютери підключені до мережі, дізнатись їх імена, а також переглянути яке програмне забезпечення на них встановлено, яка операційна система і які типи фільтрів задіюються. Функціональність програми може бути розширена за рахунок власного скриптової мови, який дозволяє адміністраторам автоматизувати багато дій.

```
(kali@kali)-[~]
$ nmap
Nmap 7.91 ( https://nmap.org )
Usage: nmap [Scan Type(s)] [Options] {target specification}
TARGET SPECIFICATION:
  Can pass hostnames, IP addresses, networks, etc.
  Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254
  -iL <inputfilename>: Input from list of hosts/networks
  -iR <num hosts>: Choose random targets
  --exclude <host1[,host2][,host3],...>: Exclude hosts/networks
  --excludefile <exclude_file>: Exclude list from file
HOST DISCOVERY:
  -sL: List Scan - simply list targets to scan
  -sn: Ping Scan - disable port scan
  -Pn: Treat all hosts as online -- skip host discovery
  -PS/PA/PY[portlist]: TCP SYN/ACK, UDP or SCTP discovery to given ports
  -PE/PP/PM: ICMP echo, timestamp, and netmask request discovery probes
  -PO[protocol list]: IP Protocol Ping
  -n/-R: Never do DNS resolution/Always resolve [default: sometimes]
  --dns-servers <serv1[,serv2],...>: Specify custom DNS servers
  --system-dns: Use OS's DNS resolver
  --traceroute: Trace hop path to each host
SCAN TECHNIQUES:
  -sS/sT/sA/sW/sM: TCP SYN/Connect()/ACK/Window/Maimon scans
  -sU: UDP Scan
  -sN/sF/sX: TCP Null, FIN, and Xmas scans
  --scanflags <flags>: Customize TCP scan flags
  -sI <zombie host[:probeport]>: Idle scan
  -sY/sZ: SCTP INIT/COOKIE-ECHO scans
```

Рисунок 1.36 – Частина довідки NMap

		Лещенко Б.С.			Арк.
Змн.	Арк.	№ докум.	Підпис	Дата	41

Для виконання подальших дій по скануванню буде використовуватися Nmap через, те що він найкраще підходить для наших задач та він є більш гнучким і використовується саме для виконання пентестів.

Для виконання клонування веб сторінок найпоширеніша програма це **HTTrack**. HTTrack — вільний офлайн-браузер з відкритим вихідним кодом. Програма дозволяє створювати локальні копії сайтів або окремих інтернет сторінок для їх подальшого перегляду. Також надається можливість оновлювати існуючі дзеркала сайтів та відновлювати по необхідності перервані скачування файлів. HTTrack має багато параметрів налаштування та вбудовану систему допомоги. Для Windows програма має назву WinHTTrack, для Unix-подібних — WebHTTrack, на основі Qt4 — HTTraQt.

Програма має багато можливостей:

- Можливість обирати глибину перевірки.
- Збереження структури сайту під час завантаження.
- Вбудований фільтр дозволяє виключити завантаження файлів за типом.
- Завантаження відбувається в багатопоточному режимі (налаштовується окремо).
- Підтримка проксі-серверів.

Програма входить в пакет інструментів Kali Linux.

В якості веб серверу з поміж Apache2, NginX та IIS було обрано Apache2 через свою простоту в налаштуванні та через те, що Apache2 входить в пакет встановлених програм Kali Linux. **Apache** HTTP-сервер це відкритий веб-сервер Інтернет для UNIX-подібних, Microsoft Windows, Novell NetWare та інших операційних систем. Apache розроблюється та підтримується спільнотою розробників відкритого програмного забезпечення під керівництвом Apache Software Foundation. В 1996 році Apache обійшов NCSA HTTPd із того часу є найбільш популярним веб-сервером у світі. Станом на січень 2021 року Apache встановлений на 25% серверів. Apache2, як і інше серверне програмне забезпечення, може бути зламаним. Основним інструментом атаки на Apache є

		Леценко Б.С.				Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		42

Slowloris, який використовує недоліки в програмному забезпеченні Apache. Він створює безліч сокетів і утримує кожен з них працюючим і зайнятим, надсилаючи кілька байтів (відомих як «keep-alive заголовки»), що дають знати серверу, що комп'ютер все ще підключений і не має проблем із мережею. Модулі Apache `mod_limitipconn`, `mod_qos`, `mod_evasive`, `mod_security`, `mod_noloris` та `mod_antiloris` були запропоновані як засіб зменшення ймовірності успішної атаки Slowloris. Починаючи з Apache 2.2.15, Apache постачає модуль `mod_reqtimeout` як офіційне рішення, яке підтримується розробниками.

Для виконання підміни DNS буде використовуватись комплекс програмного забезпечення з графічним інтерфейсом – **Ettercap**. Ettercap це вільне програмне забезпечення для виконання MITM атак в локальних мережах. Воно використовується для аналізу інтернет протоколів та аудиту безпеки, а також воно підходить для перехоплення трафіку на мережевому сегменті, перехоплення паролів та проведення активного підслуховування ряду загальноприйнятих протоколів. Пізніше його розробниками було засновано Hacking Team.

Ettercap працює за рахунок переведення мережевого інтерфейсу до «нерозбірливого» режиму та отруєння ARP на цільових комп'ютерах. Ettercap надає можливість роботи в чотирьох режимах:

- Оснований на IP: пакети фільтруються основуючись на IP адресі джерела та призначення.
- Оснований на MAC: пакети фільтруються основуючись на MAC адресі.
- Оснований на ARP: використання отруєння ARP для перехвату комутованої локальної мережі між двома хостами (full-duplex).
- Оснований на PublicARP: використання отруєння ARP для перехвату комутованої локальної мережі з хоста жертви до всіх інших хостів (half-duplex).

Ну і звичайно необхідно потрібно розібрати дистрибутив **Kali Linux**, який згадувався вище та який буде використовуватися в якості операційної системи, за якою буде працювати зловмисник. Kali Linux – це версія операційної системи Linux, яка базується на Debian. Цей дистрибутив був спеціально спроектований для

		Лещенко Б.С.				Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		43

проведення тестів на проникнення та цифрових розслідувань. Одна з найважливіших особливостей Kali це його передвстановлене програмне забезпечення для виконання різних задач кібербезпеки. Kali Linux безкоштовний. Через те що він включає в себе більше 600 різних програм та стою безкоштовність і простоту, Kali набув значної популярності в сфері кібербезпеки.

Зазвичай Kali не встановлюють на домашні комп'ютери, а використовують в вигляді віртуальної операційної системи. З офіційного сайту можна завантажити вже готові образи систем для різного програмного забезпечення віртуалізації – VirtualBox або VMWare.

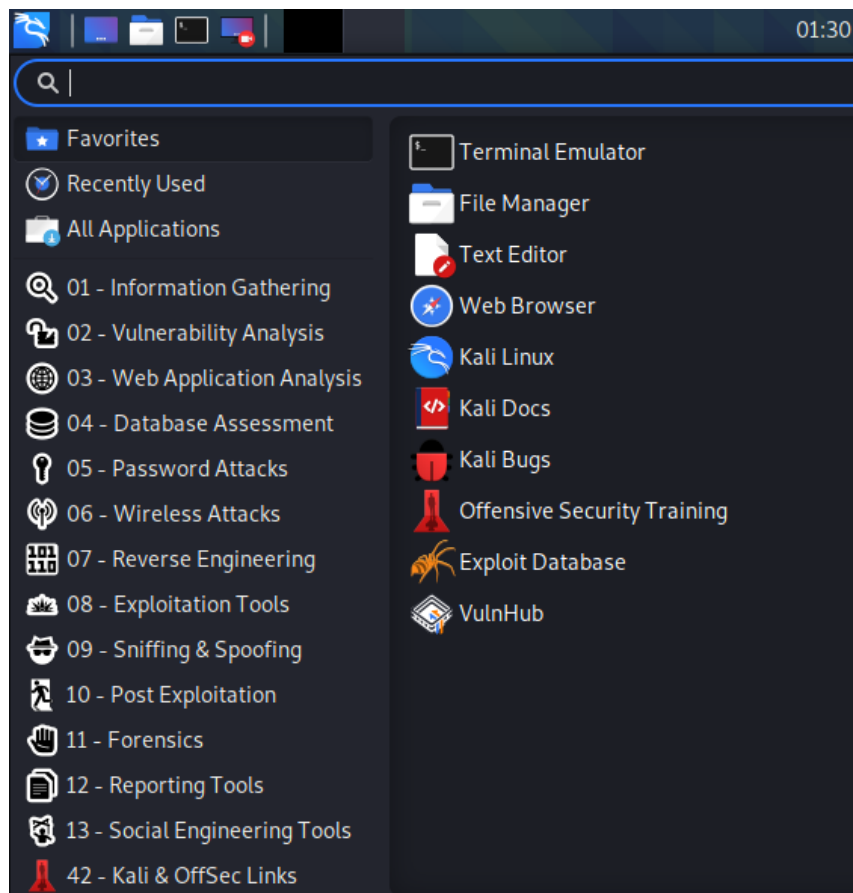


Рисунок 1.37 – Зображення інструментарію Kali Linux

Доменна система імен має давню історію, є одним із тих протоколів, що працюють майже на всіх комп'ютерних системах, що підключені до глобальної мережі Інтернет, та має доволі легкий для розуміння принцип роботи та структуру, що означає, що DNS є цілю для великої кількості зловмисників впродовж багатьох

		Леценко Б.С.				Арк.
						44
Змн.	Арк.	№ докум.	Підпис	Дата		

років. Також було досліджено, що існує велика кількість програмних застосунків для дослідження доменної системи імен.

Після розгляду теоретичних відомостей та дослідження принципів проведення атак на систему доменних імен, а саме: після розробки схеми атаки, опису всіх етапів та опису всього необхідного програмного забезпечення, можна приступати до здійснення, безпосередньо, атаки.

В наступному розділі буде використано всю інформація зібрану в розділі 1 практично.

		Леценко Б.С.				Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		45

РОЗДІЛ 2. ПРАКТИЧНЕ НАЛАГОДЖЕННЯ СЕРВЕРУ СИСТЕМИ ДОМЕННИХ ІМЕН ТА ЗДІЙСНЕННЯ АТАКИ З ПІДМІНИ ЗАПИСІВ ДОМЕННОЇ СИСТЕМИ ІМЕН

2.1. Налаштування DNS серверу за допомогою вбудованих функцій Cisco Packet Tracer та налаштування маршрутизатора Cisco в ролі серверу DNS.

Для дослідження DNS серверів вбудованих в програмне забезпечення Cisco PacketTracer будемо використовувати топологію мережі, яка зображена у Додатку А, схему підключень, яка зображена у Додатку Б, та схему адресації, яка наведена у Додатку В.

Кінцеві хости в даній мережі мають статичні IP адреси та між маршрутизаторами налаштовано протокол маршрутизації RIPv2.

Для наглядної перевірки роботи DNS в мережу Н було додано HTTP сервер. Домен Leshchenko.com буде керуватися public-dns сервером. Сайти у піддоменах site1.leshchenko.com та site2.leshchenko.com знаходяться у мережі 199.1.10.162.

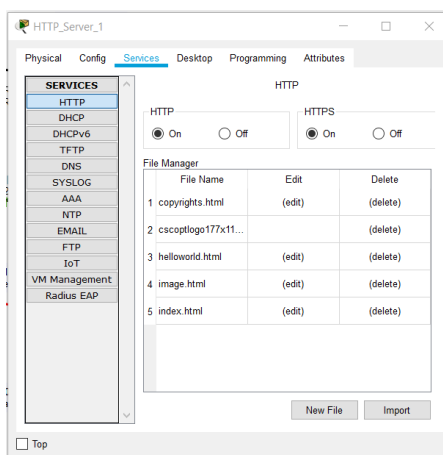


Рисунок 2.1 – Налаштування HTTP серверу

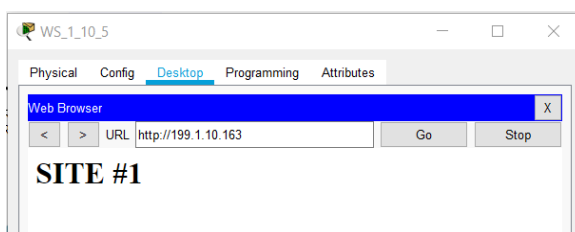


Рисунок 2.2 – Перевірка роботи серверу за IP адресою

		Лещенко Б.С.				Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		46

Додаєм **A** запис на public DNS сервер, власне, про себе самого, додаєм **NS** запис про те, що домен Leshchenko.com керується public DNS сервером, та додаєм **SOA** запис, який означає початок нової DNS зони та містить інформацію про авторитетний сервер імен, контактну інформацію і як часто DNS інформацію буде оновлено. Також потрібно додати, безпосередньо, трансляції субдоменів до IP адрес (site1.leshchenko.com та site2.leshchenko.com) у вигляді **A** записів.

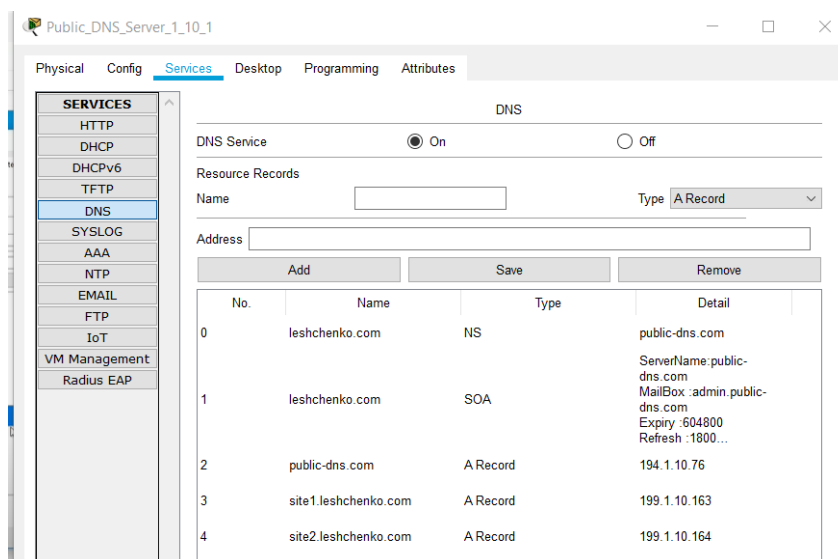


Рисунок 2.3 – Додавання нових записів на public DNS сервер

На privat DNS сервері достатньо налаштувати **A** запис з перетворенням посилання на IP адресу public сервера та **NS** запис з вказанням того, що доменом Leshchenko.com керує public сервер.

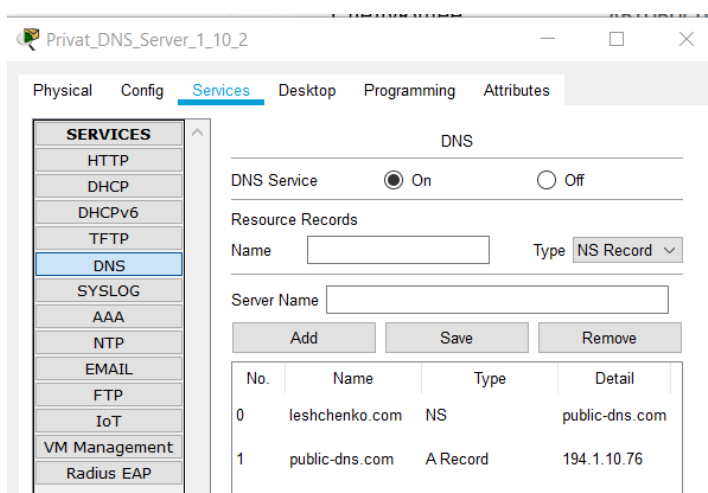


Рисунок 2.4 – DNS записи на privat сервері

В записи DNS можна вносити не лише веб-сервери, а взагалі будь-які хости, тому для простішого адміністрування мережі зробимо записи для пристроїв мережі.

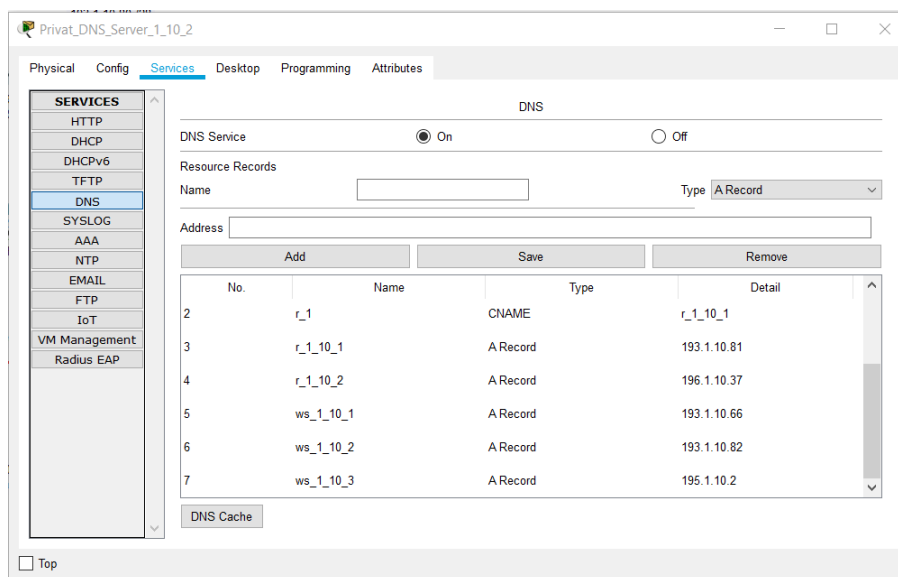


Рисунок 2.5 – Додавання записів про хости

Наряду з **A** записами, також додано **CNAME** записи, які посилаються на **A** записи.

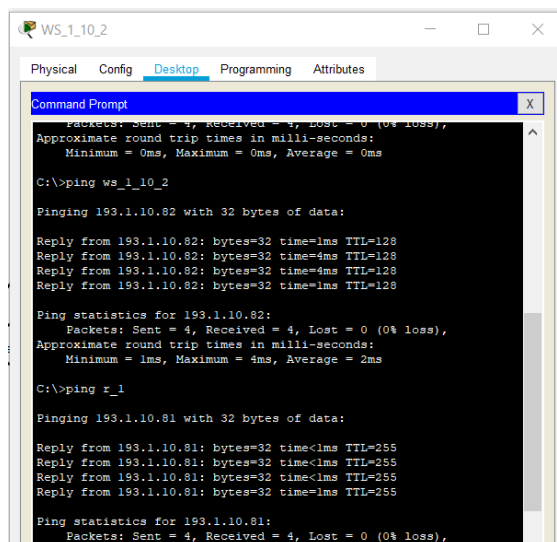


Рисунок 2.6 – Перевірка доданих записів на Privat DNS сервері

На рисунку (див.рис.2.6) зображено успішне трансліювання текстових записів до IP адрес та успішну перевірку зв'язку між пристроями.

На цьому дослідження вбудованих в Cisco PacketTracer функцій пов'язаних з DNS серверами можна вважати завершеним.

Оскільки в ролі DNS серверу також можуть виступати також маршрутизатори компанії Cisco, також розглянемо налаштування маршрутизатору в ролі DNS серверу. Топологія мережі зображена на рисунку Г в додатку Г, параметри підключень пристроїв наведено у додатку Д та параметри адресації мережі наведено у додатку Ж. На всіх кінцевих пристроях мережі налаштовано статичні IP адреси, а на всіх маршрутизаторах мережі налаштовано протокол маршрутизації RIP версії 2.

Після базового налаштування пристроїв та перевірки зв'язку за допомогою протоколу ICMP можна переходити до налаштування DNS серверів.

Першим кроком в налаштуванні DNS серверів на маршрутизаторах Cisco є введення команди *ip dns server* , яка включає DNS функціонал. **Наступним** кроком є введення команди *ip domain-lookup* для перевірки налаштувань з самого серверу. **Третім** кроком є налаштування доменного імені за допомогою команди *ip domain name <name>* , в нашому випадку *ip domain name leshchenko.com* . **Четвертим** кроком є додавання **NS** записів для Primary та Secondary DNS сервери на самих маршрутизаторах: *ip name-server 192.168.1.5 192.168.1.6* на Primary маршрутизаторі та *ip name-server 192.168.1.6 192.168.1.5* на Secondary. **П'ятим** кроком є створення записів про DNS зону, до якої належать маршрутизатори, та записів про самих себе: *ip dns primary leshchenko.com soa dns1.leshchenko.com mailbox.leshchenko.com* . **Останнім** кроком є безпосередньо створення **A** записів про хости мережі: *ip host site1.leshchenko.com 10.100.1.5* .

Після цих налаштувань (див. рис.2.7) та налаштувань DNS на хостах мережі проводимо перевірку роботи DNS в мережі (див. рис.2.8).

		Лещенко Б.С.				Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		49

```

R_1_10_1(config)#ip dns server
R_1_10_1(config)#ip domain lookup
R_1_10_1(config)#ip domain name leshchenko.com
R_1_10_1(config)#ip name-server 192.168.1.5 192.168.1.6
R_1_10_1(config)#$enko.com soa dns1.leshchenko.com mailbox.leshchenko.com
R_1_10_1(config)#ip host site1.leshchenko.com 10.100.1.5
R_1_10_1(config)#ip host dns1.leshchenko.com 192.168.1.5
R_1_10_1(config)#ip host dns2.leshchenko.com 192.168.1.6
R_1_10_1(config)#ip host main.leshchenko.com 192.168.1.1
R_1_10_1(config)#ip host host1.leshchenko.com 10.1.1.5
R_1_10_1(config)#ip host host2.leshchenko.com 10.1.1.6
R_1_10_1(config)#

```

```

R_1_10_2(config)#ip dns server
R_1_10_2(config)#ip domain lookup
R_1_10_2(config)#ip domain name leshchenko.com
R_1_10_2(config)#ip name-server 192.168.1.6 192.168.1.5
R_1_10_2(config)#ip dns primary leshchenko.com soa dns2.leshchenko.com mailbox$
R_1_10_2(config)#ip host site1.leshchenko.com 10.100.1.5
R_1_10_2(config)#ip host dns1.leshchenko.com 192.168.1.5
R_1_10_2(config)#ip host dns2.leshchenko.com 192.168.1.6
R_1_10_2(config)#ip host main.leshchenko.com 192.168.1.1
R_1_10_2(config)#ip host host1.leshchenko.com 10.1.1.5
R_1_10_2(config)#ip host host2.leshchenko.com 10.1.1.6
R_1_10_2(config)#

```

Рисунок 2.7 – Налаштування двох маршрутизаторів в ролі DNS серверів

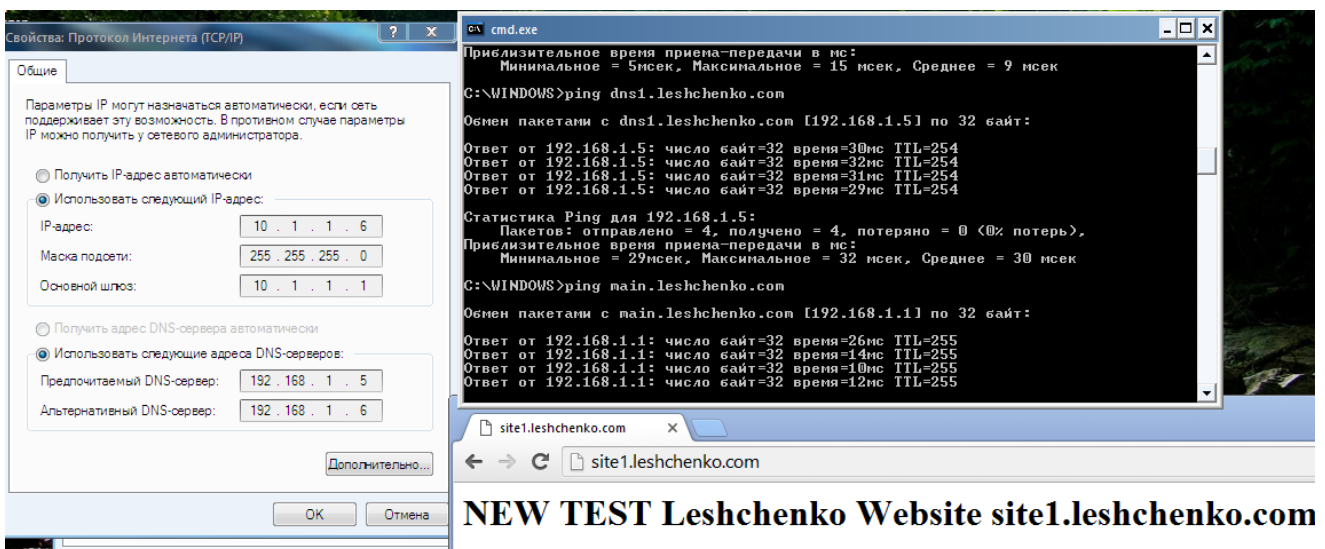


Рисунок 2.8 – Налаштування WS_1_10_1 та перевірка працездатності трансляції імен

Як видно з рисунків вище, трансляція імен виконується успішно та доступ до сайту за допомогою адреси наявний. Діагностичні команди будуть розглянуті у розділі 3.

		Лещенко Б.С.				Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		50

2.2. Налаштування ОС Windows Server в ролі DNS серверу

Програмне забезпечення Cisco PacketTracer має обмежений функціонал стосовно емуляції DNS серверів та використання маршрутизаторів, як DNS серверів, також достатньо застаріла практика. У сучасних комп'ютерних мережах в ролі DNS серверів зазвичай використовуються комп'ютери на базі операційних систем Windows Server або Unix-like систем, тому цей пункт присвячений налаштуванню компонентів операційної системи Windows Server 2019 в ролі серверу DNS.

Для дослідження було використано схему аналогічну схемі наведених у додатку Г, але замість маршрутизатору R_1_10_1 буде використано хост з ОС Windows.

Для встановлення додаткових ролей на сервер використовується Мастер додавання ролей та компонентів (Add Roles and Features), який можна знайти в Менеджері серверу (Manage). На верхній навігаційній панелі диспетчера серверу з правого боку знаходиться управління ролями та компонентами (див.рис.2.9).

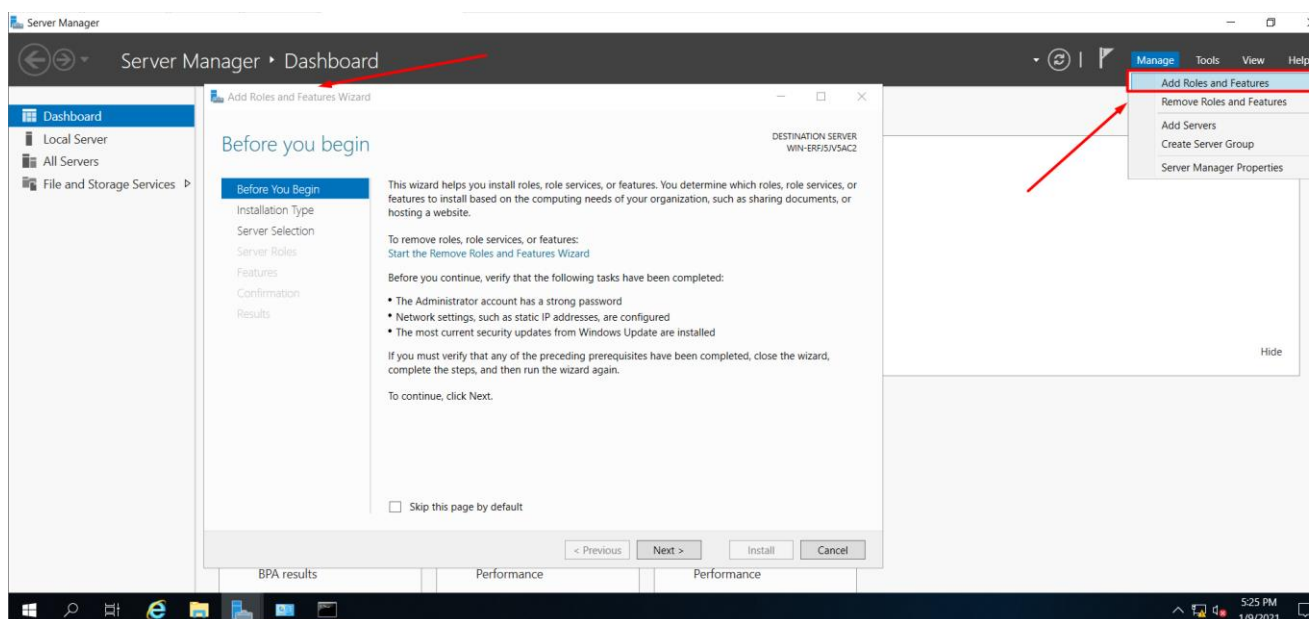


Рисунок 2.9 – Початок додавання нової ролі на Windows Server

У першому вікні майстра додавання нових ролей нам рекомендують переконатися, що обліковий запис адміністратора серверу захищений надійним паролем, налаштування мережі мають статичні адреси та що операційна система

		Леценко Б.С.			Арк.
Змн.	Арк.	№ докум.	Підпис	Дата	
					51

використовує релевантні оновлення системи безпеки. Перевіряємо чи всі пункти виконано та натискаємо «Далі».

В наступному пункті обираємо встановлення ролей та компонентів і натискаємо «Далі» (див.рис.2.10).

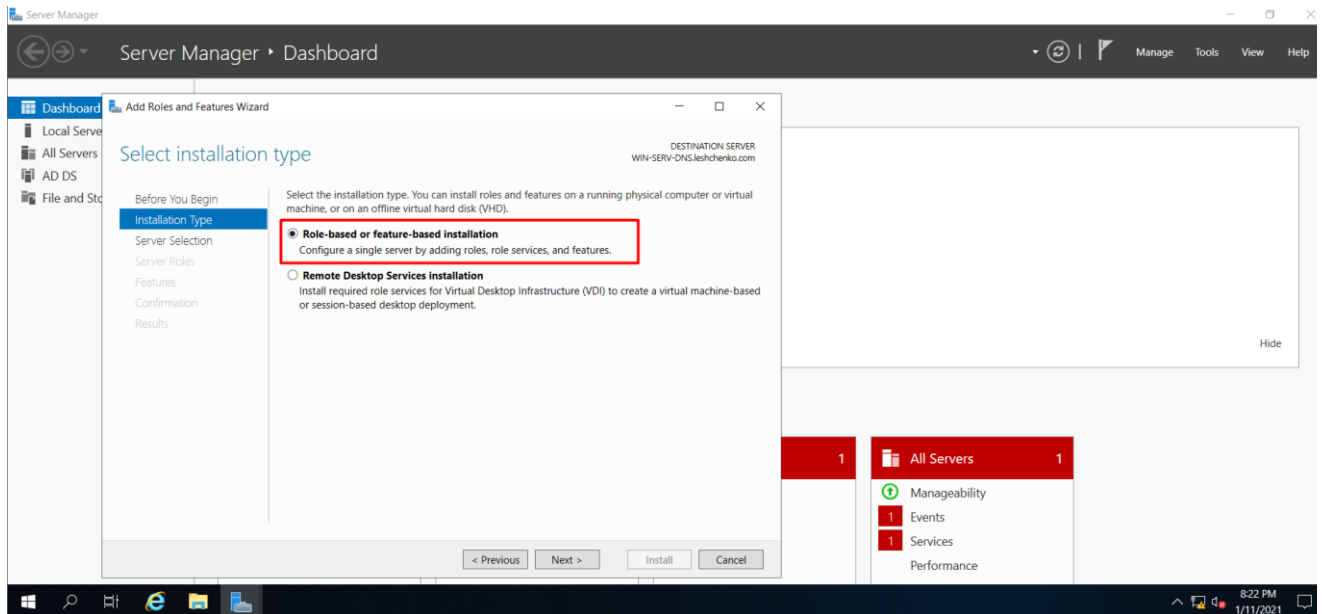


Рисунок 2.10 – Початок додавання нової ролі на Windows Server

Наступним кроком обираємо необхідний сервер з списку серверів та натискаємо «Далі».

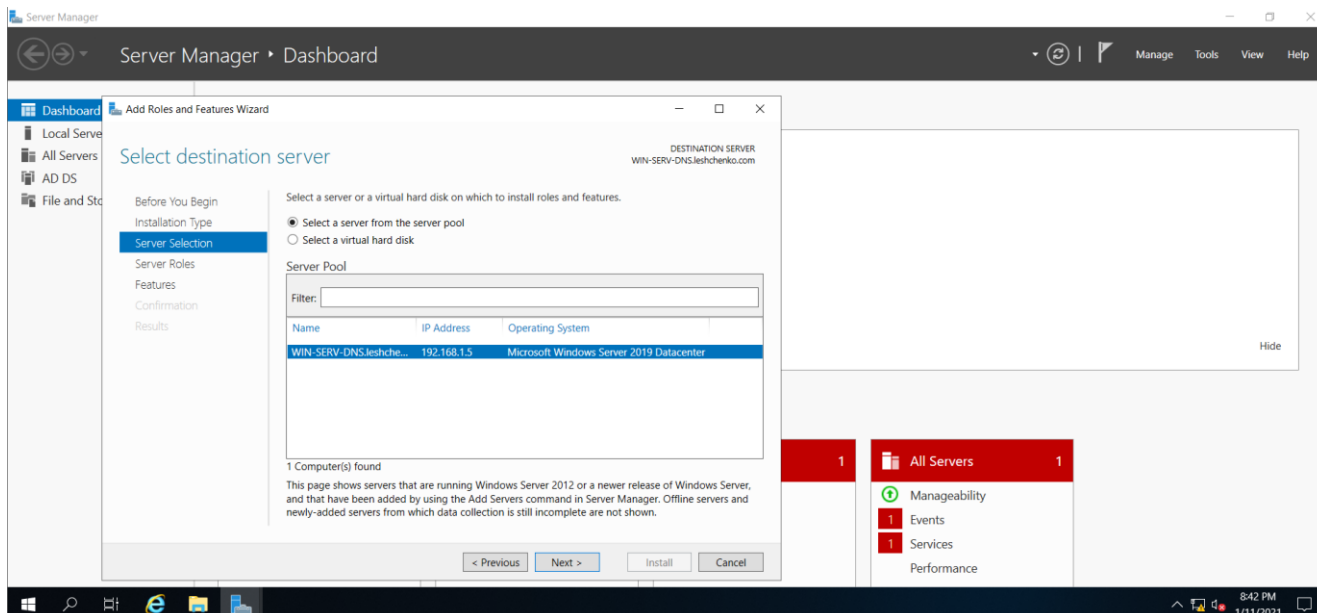


Рисунок 2.11 – Вибір необхідного серверу з пулу

Після вибору необхідного серверу, в наступному вікні маємо можливість обрати потрібні нові ролі для Windows Server. Обираємо DNS та у спливаючому

		Леценко Б.С.			Арк.
Змн.	Арк.	№ докум.	Підпис	Дата	52

вікні залишаємо стандартні налаштування та тиснемо «Додати компонент» (див.рис.2.12).

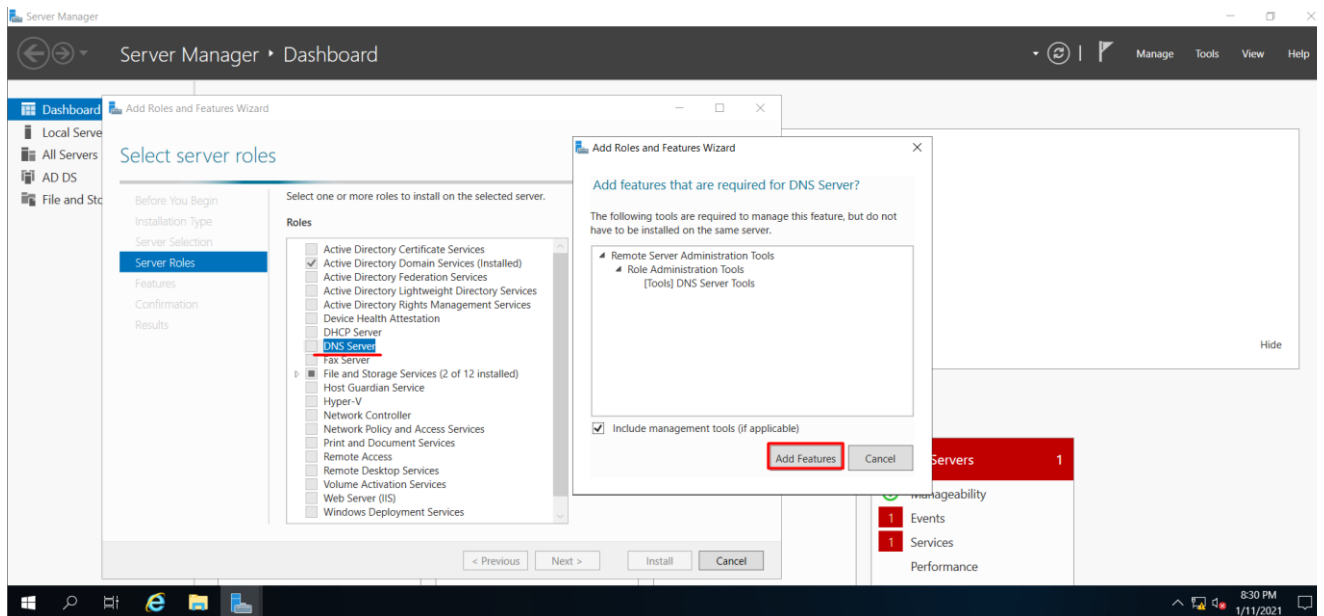


Рисунок 2.12 – Додавання нового компоненту

Всі наступні пункти не потребують змін, тому натискаємо «Далі» та встановлюємо нову роль для сервера.

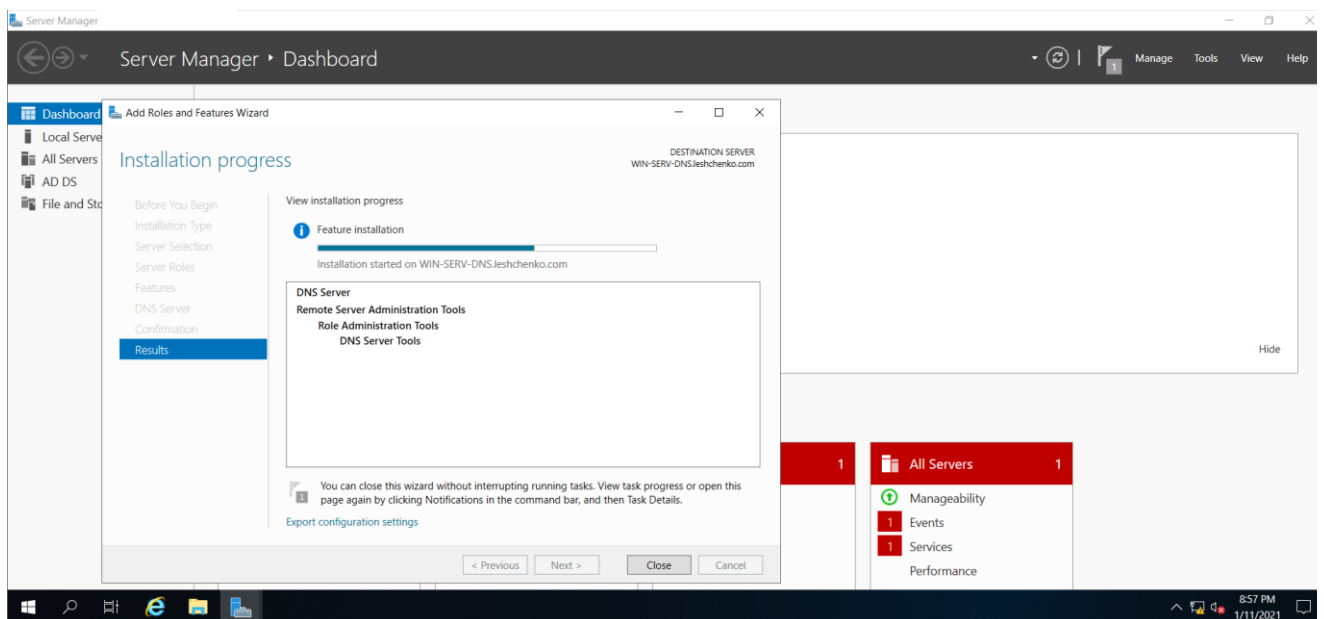


Рисунок 2.13 – Процес встановлення компонентів на Windows Server

Після успішного встановлення нової ролі в меню вибору інструментів з'явиться новий пункт «DNS». Виріб цього пункту відкриває диспетчер управління записами DNS (див.рис.2.14).

		Леценко Б.С.			Арк.
Змн.	Арк.	№ докум.	Підпис	Дата	53

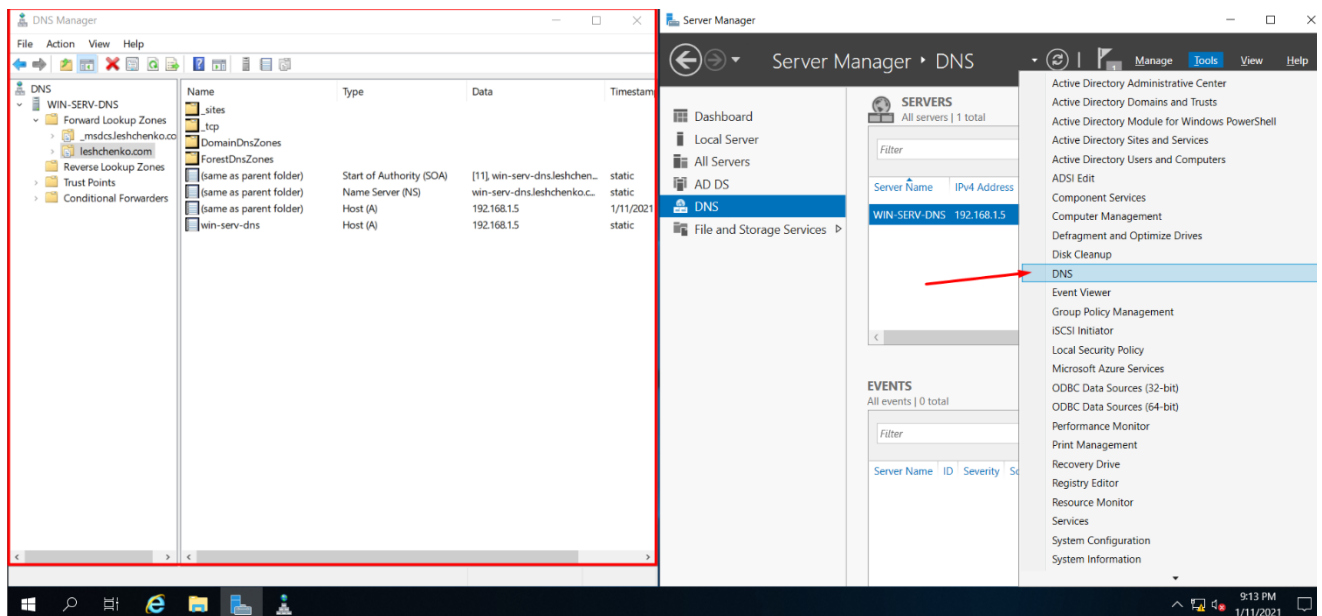


Рисунок 2.14 – Ввімкнення DNS Manager

В лівій ієрархії нашого сервера стандартно створюється дерево каталогів, кожен з яких відповідає за свої функції:

Forward Lookup Zones – *зони прямого огляду* – відповідає за трансляцію імен хостів в IP адреси.

Reverse Lookup Zones – *зони зворотного огляду* – відповідає за розпізнавання DNS сервером DNS імені хоста, тобто навпаки записам зон зворотного огляду.

Trust Points – *довірені точки* – каталог в якому містяться так звані «якорі довіри» (trust anchor). Ці «якорі» являють собою відкриті ключі шифрування для справної роботи протоколу DNSSEC.

Conditional Forwarders – *умовна переадресація* – механізм, створений для захисту від підробки джерел DNS записів, застосовується для переадресації на інший довірений DNS сервер.

Також було автоматично створено в каталозі зон прямого огляду зону для домену *leshchenko.com*, а в середині зони створено **SOA**(див.рис.2.15), **NS** та **A** записи. Додамо декілька **A** записів та перевіримо працездатність DNS серверу (див.рис.2.16).

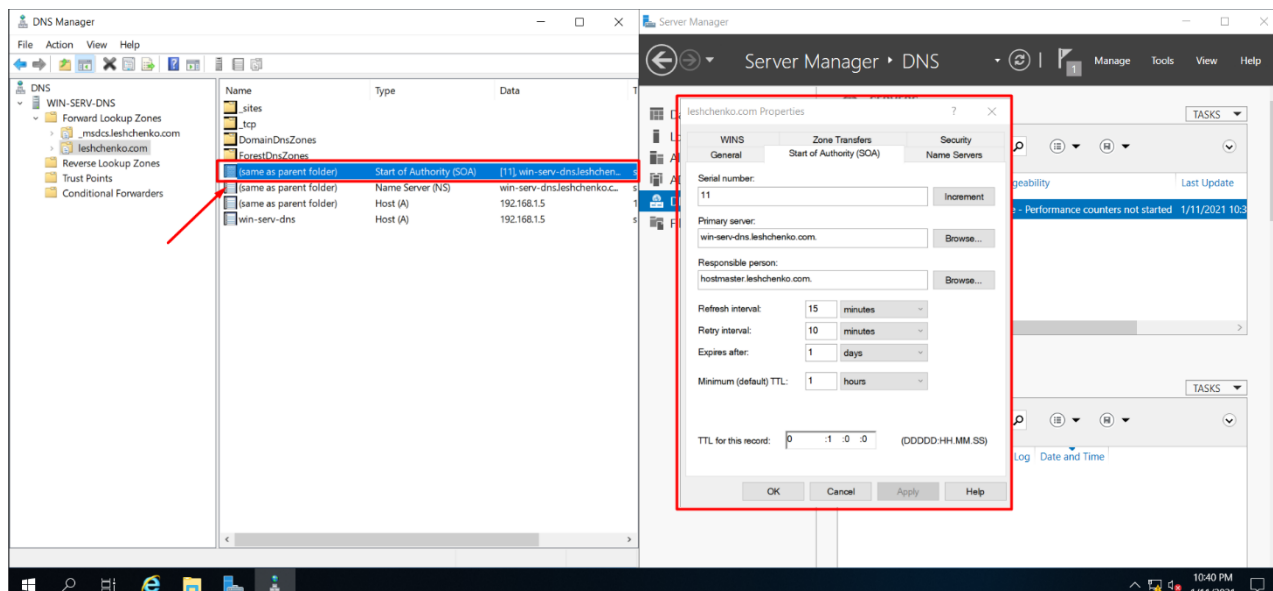


Рисунок 2.15 – Ввімкнення DNS Manager

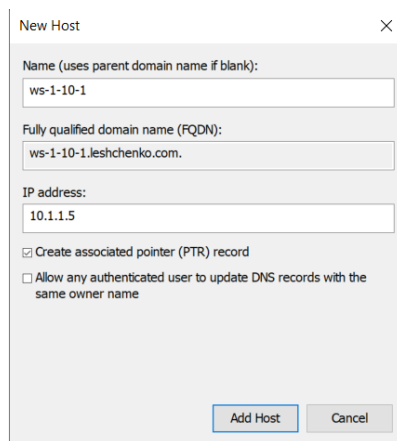


Рисунок 2.16 – Вікно створення нового А запису

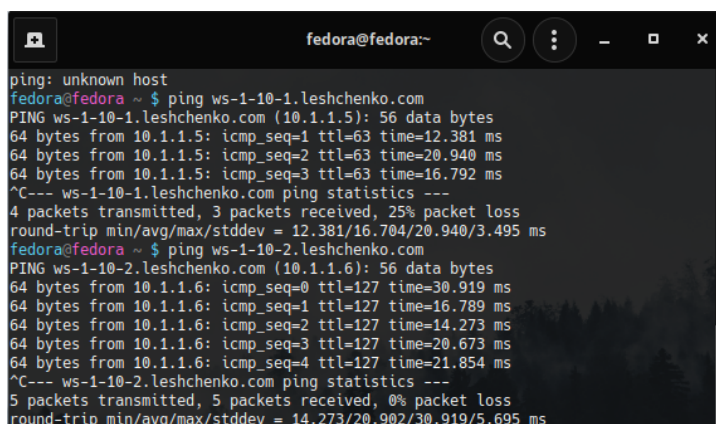


Рисунок 2.17 – Перевірка роботи DNS

		Лещенко Б.С.			Арк.
Змн.	Арк.	№ докум.	Підпис	Дата	55

Після проведеної перевірки зв'язку можна стверджувати, що DNS працює. Налаштування Windows Server в ролі DNS серверу можна вважати завершеним.

2.3. Сканування мережі за допомогою програмного додатку Nmap

Для прикладу сканування буде використано схему мережі (див. рис. 2.1), яка є простою схемою домашнього Інтернету або маленького підприємства.

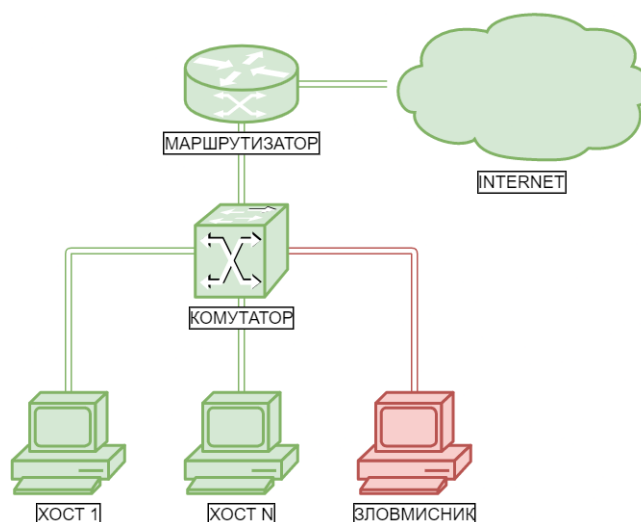


Рисунок 2.1 – Схема мережі

Після того, як зловмисник потрапив до мережі, необхідно дізнатися налаштування мережі, а саме IP адресу мережі та маску мережі. Для того, щоб отримати інформацію про мережу, в Debian-like операційних системах існує команда **ip address** (див.рис.2.2).

```

(kali@kali)-[~]
$ ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:cc:5a:6e brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.103/24 brd 192.168.0.255 scope global dynamic noprefixroute eth0
        valid_lft 7147sec preferred_lft 7147sec
    inet6 fe80::a00:27ff:fecc:5a6e/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
  
```

Рисунок 2.2 – Результат виконання команди ip address

З виводу даної команди отримуємо, що ми знаходимось в мережі 192.168.0.0 з префіксом /24 та отримуємо ще деякі відомості.

		Лещенко Б.С.			Арк.
Змн.	Арк.	№ докум.	Підпис	Дата	56

Також є важливим знання адреси основного шлюзу мережі, для таких цілей існує команда **ip route** (див. рис.2.3).

```
(kali㉿kali)-[~]  
$ ip route  
default via 192.168.0.1 dev eth0 proto dhcp metric 100  
192.168.0.0/24 dev eth0 proto kernel scope link src 192.168.0.103 metric 100
```

Рисунок 2.3 – Результат виконання команди ip route

З виводу цієї команди дізнаємось, що адреса шлюзу – 192.168.0.1.

Переходимо до роботи з Nmap. Nmap налічує десятки різних режимів, команд та підкоманд, але розглянемо основні.

Найпростіший режим – сканування за допомогою пакетів ping (див. рис.2.4) – відбувається за допомогою команди **nmap -sP 192.168.0.0/24**.

```
(kali㉿kali)-[~]  
$ nmap -sP 192.168.0.0/24  
Starting Nmap 7.91 ( https://nmap.org ) at 2021-05-06 14:32 EDT  
Nmap scan report for 192.168.0.1  
Host is up (0.019s latency).  
Nmap scan report for 192.168.0.102  
Host is up (0.092s latency).  
Nmap scan report for 192.168.0.103  
Host is up (0.00099s latency).  
Nmap done: 256 IP addresses (3 hosts up) scanned in 3.42 seconds
```

Рисунок 2.4 – Результат виконання команди nmap -sp 192.168.0.0/24

Програма знайшла три працюючих хости, які відповіли на ping запити.

Наступний режим, який вартий уваги, це режим сканування одного хост за допомогою команди **nmap 192.168.0.1** (див. рис. 2.5).

```
(kali㉿kali)-[~]  
$ nmap 192.168.0.1  
Starting Nmap 7.91 ( https://nmap.org ) at 2021-05-06 14:37 EDT  
Nmap scan report for 192.168.0.1  
Host is up (0.0056s latency).  
Not shown: 997 closed ports  
PORT      STATE SERVICE  
80/tcp    open  http  
1900/tcp   open  upnp  
49152/tcp  open  unknown  
  
Nmap done: 1 IP address (1 host up) scanned in 0.57 seconds
```

Рисунок 2.5 – Результат виконання команди nmap -sp 192.168.0.0/24

		Леценко Б.С.				Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		57

В цьому режимі програма сканує хост на наявність відкритих, серед 1000 найвідоміших, портів. Виконано сканування шлюзу та виявлено відкриті 80, 1900 та 49152 порти.

«Скритне сканування» (Stealth scan) виконується за рахунок надсилення SYN пакетів і аналізування відповідей. Якщо SYN/ACK отримано, це означає, що порт відкритий, та є можливість відкрити TCP з'єднання. Однак, «скритне сканування» ніколи не завершує потрібне рукоштовування, отже, в такому випадку цілі важко ідентифікувати сканування системи. Для виконання «скритного сканування» використовується команда ***nmap -sS 192.168.0.1*** (див. рис. 2.6).

```
(kali@kali)-[~]
└─$ sudo nmap -sS 192.168.0.1
Starting Nmap 7.91 ( https://nmap.org ) at 2021-05-06 14:48 EDT
Nmap scan report for 192.168.0.1
Host is up (0.011s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
80/tcp    open  http
1900/tcp   open  upnp
49152/tcp  open  unknown
MAC Address: 64:66:B3:48:D3:3E (Tp-link Technologies)

Nmap done: 1 IP address (1 host up) scanned in 0.43 seconds
```

Рисунок 2.6 – Результат виконання команди ***nmap -sS 192.168.0.1***

Вивід цієї команди відрізняється від звичайного сканування, що при «скритному скануванні» виводиться також і MAC адреса хосту. Варто пам'ятати, що «скритне сканування» є повільним та не настільки агресивним, ніж інші типи сканування, отже можливо буде необхідно деякий час очікувати на відповідь.

Наступним важливим видом сканування є сканування для знаходження версій сервісів. Це робить набагато простішим знаходження та експлуатація відомих вразливостей. Для виконання перевірки на версії до команди дописується прапорець «-sV» (див. рис. 2.7). Варто також пам'ятати, що ця команда може надавати не на 100% точну інформацію.

		Леценко Б.С.				Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		58

```

(kali@kali)-[~]
$ sudo nmap -sV 192.168.0.1
Starting Nmap 7.91 ( https://nmap.org ) at 2021-05-06 14:58 EDT
Nmap scan report for 192.168.0.1
Host is up (0.0026s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE VERSION
80/tcp    open  http    TP-LINK WR841N WAP http config
1900/tcp  open  upnp    ipOS upnpd (TP-LINK TL-WR841N WAP 8.0; UPnP 1.0)
49152/tcp open  http    Huawei HG8245T modem http config
MAC Address: 64:66:B3:48:D3:3E (Tp-link Technologies)
Service Info: OS: ipOS 7.0; Devices: WAP, broadband router; CPE: cpe:/h:tp-link:wr841n, cpe:/o:tp-link:tl-wr841n, cpe:/o:ubicom:ipos:7.0, cpe:/h:huawei:hg8245t

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 85.11 seconds

```

Рисунок 2.7 – Результат виконання команди `nmap -sV 192.168.0.1` для сканування шлюзу

У результаті виводу команди з'явилися дані про версії запущених сервісів.

```

(kali@kali)-[~]
$ sudo nmap -sV 192.168.0.101
Starting Nmap 7.91 ( https://nmap.org ) at 2021-05-06 15:05 EDT
Nmap scan report for 192.168.0.101
Host is up (0.00054s latency).
Not shown: 998 filtered ports
PORT      STATE SERVICE VERSION
445/tcp   open  microsoft-ds Microsoft Windows 7 - 10 microsoft-ds (workgroup: ISM-ECOMPANY)
2103/tcp  open  msrpc    Microsoft Windows RPC
MAC Address: B8:08:CF:88: (Intel Corporate)
Service Info: Host: DEV-; OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1206.81 seconds

```

Рисунок 2.8 – Результат виконання команди `nmap -sV 192.168.0.101` для сканування жертви

Раніше в склад Kali linux входив додаток Zenmap, який є графічним інтерфейсом для Nmap. Через те, що Zenmap був написаний на Python2, а підтримка Python2 завершилась в 2020 році, Zenmap не був переписаний на Python3 та вже не входить в склад Kali linux.

Nmap є, безсумнівно, «швейцарським ножом» для мережеских адміністраторів завдяки своєму набору універсальних команд. Nmap дозволяє швидко сканувати та досліджувати всю необхідну інформацію про мережу, хости, порти, фаєрволи та операційні системи. Nmap має безліч налаштувань, прапорців, що допомагають адміністраторам аналізувати мережу в деталях.

		Леценко Б.С.				Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		59

2.4. Здійснення клонування веб сторінки на прикладі сторінки соціальної мережі та налаштування веб серверу Apache2 з підробною сторінкою авторизації.

Як було досліджено в попередній пунктах, для виконання копіювання веб сторінок буде використано додаток HTTrack. HTTrack – це безкоштовна, відкрита, проста в використанні програма для оффлайн перегляду веб сайтів. Вона дозволяє завантажувати веб сайти з Інтернету в локальний каталог, повторюючи структуру тек, отримуючи HTML, зображення та інші файли з серверу на ваш комп'ютер. HTTrack організує оригінальну структуру відносних посилань. Існують версії з графічним інтерфейсом та без нього, також версії для Windows та Linux.

HTTrack має велику кількість гнучких налаштувань (див. рис.2.9) та може працювати в двох режимах: інтерактивному та неінтерактивному. Далі всі дії будуть виконуватися в інтерактивному режимі.

```
(kali@kali)-[~]
$ httrack -h

HTTrack version 3.49-2
usage: httrack <URLs> [-option] [+<URL_FILTER>] [-<URL_FILTER>] [+<mime:MIME_FILTER>] [-<mime:MIME_FILTER>]
with options listed below: (* is the default value)

General options:
  O path for mirror/logfiles+cache (-O path_mirror[,path_cache_and_logfiles]) (--path <param>)

Action options:
  w *mirror web sites (--mirror)
  W mirror web sites, semi-automatic (asks questions) (--mirror-wizard)
  g just get files (saved in the current directory) (--get-files)
  i continue an interrupted mirror using the cache (--continue)
  Y mirror ALL links located in the first level pages (mirror links) (--mirrorlinks)

Proxy options:
  P proxy use (-P proxy:port or -P user:pass@proxy:port) (--proxy <param>)
  %f *use proxy for ftp (f0 don't use) (--httpproxy-ftp[=N])
  %b use this local hostname to make/send requests (-%b hostname) (--bind <param>)

Limits options:
  rN set the mirror depth to N (* r9999) (--depth[=N])
  %eN set the external links depth to N (* %e0) (--ext-depth[=N])
  mN maximum file length for a non-html file (--max-files[=N])
  mN,N2 maximum file length for non html (N) and html (N2)
  MN maximum overall size that can be uploaded/scanned (--max-size[=N])
  EN maximum mirror time in seconds (60=1 minute, 3600=1 hour) (--max-time[=N])
  AN maximum transfer rate in bytes/seconds (1000=1KB/s max) (--max-rate[=N])
  %cN maximum number of connections/seconds (*%c10) (--connection-per-second[=N])
  GN pause transfer if N bytes reached, and wait until lock file is deleted (--max-pause[=N])

Flow control:
  cN number of multiple connections (*c8) (--sockets[=N])
  TN timeout, number of seconds after a non-responding link is shutdown (--timeout[=N])
  RN number of retries, in case of timeout or non-fatal errors (*R1) (--retries[=N])
  JN traffic jam control, minimum transfert rate (bytes/seconds) tolerated for a link (--min-rate[=N])
  HN host is abandoned if: 0=never, 1=timeout, 2=slow, 3=timeout or slow (--host-control[=N])
```

Рисунок 2.9 – Частина допоміжної довідки HTTrack

В плані тестування на проникнення HTTrack може бути корисна для:

— Дослідження структури сайтів (підкаталоги, сторінки сайту);

		Леценко Б.С.				Арк.
						60
Змн.	Арк.	№ докум.	Підпис	Дата		

- пошук файлів на сайті (документи, зображення);
- пошук по документах і метаданим файлів з сайту;
- клонування сторінок входу з метою наступного використання для фішингу.

Саме для останнього і буде використано цей додаток.

Після запуску програми, програма просить ввести назву нового проекту.

```
(kali㉿kali)-[~]
$ httrack

Welcome to HTTrack Website Copier (Offline Browser) 3.49-2+libhtsjava.so.2
Copyright (C) 1998-2017 Xavier Roche and other contributors
To see the option list, enter a blank line or try httrack --help

Enter project name :fake-facebook
```

Рисунок 2.10 – Запуск httrack

Далі додаток просить обрати шлях до зберігання проекту. За замовчуванням це «/home/<username>/websites». Залишаємо стандартний шлях натисканням клавіші Enter.

```
(kali㉿kali)-[~]
$ httrack

Welcome to HTTrack Website Copier (Offline Browser) 3.49-2+libhtsjava.so.2
Copyright (C) 1998-2017 Xavier Roche and other contributors
To see the option list, enter a blank line or try httrack --help

Enter project name :fake-facebook

Base path (return=/home/kali/websites/) :
```

Рисунок 2.11 – Вибір стандартного шляху для збереження сайтів

Наступним кроком є вибір, безпосередньо, посилань, які необхідно клонувати. В даному випадку було обрано сторінку вводу авторизаційних даних на веб сайті соціальної мережі Facebook.

Facebook це найпопулярніша соціальна мережа в світі та однойменна компанія Facebook, Inc, яка нею володіє. Наразі вона існує вже більше 17 років. Facebook входить в п'ятірку веб сайтів з найбільшим відвідуванням в світі. Станом на 2020 рік, кількість активних місячних користувачів перевищує 2.6 мільярди. Facebook з самого свого початку існування привертав увагу хакерів, і, навіть, на

		Лещенко Б.С.				Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		61

сьогоднішній день час від часу стає відомо про «злами» сторінок користувачів та «витоки» даних з серверів.

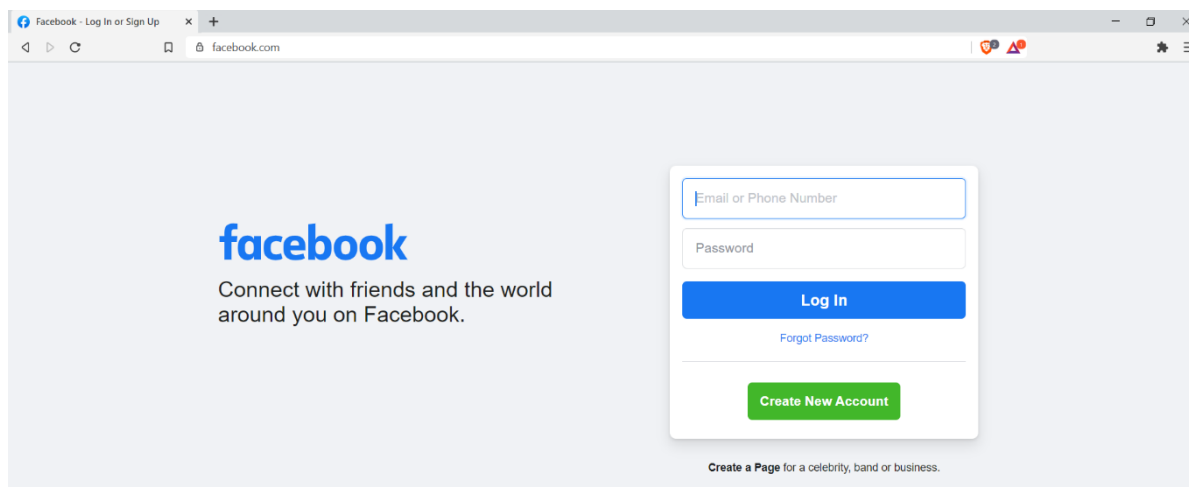


Рисунок 2.12 – Зовнішній вигляд сторінки входу в Facebook

Для входу на сайт із своїми обліковими даними користувачеві необхідно ввести логін в вигляді електронної пошти або номеру мобільного телефону та пароль. Веб сторінка входу не має динамічнопідвантажуваного контенту та коду тому створення локальної копії сайту не викликає труднощів.

В HTTrack вводимо посилання на сторінку входу (див. рис.2.12).

Після вводу адреси сайту обираємо зі списку режим копіювання всіх вкладених посилань також (пункт 4) (див.рис.2.13). Це необхідно для того, щоб коли користувач вирішив не одразу заходити в особистий кабінет, а спочатку перейти за іншим посиланням, щоб він не помітив підміни.

```
(kali@kali)-[~]
$ httrack

Welcome to HTTrack Website Copier (Offline Browser) 3.49-2+libhtsjava.so.2
Copyright (C) 1998-2017 Xavier Roche and other contributors
To see the option list, enter a blank line or try httrack --help

Enter project name :fake-facebook

Base path (return=/home/kali/websites/) :

Enter URLs (separated by commas or blank spaces) :https://www.facebook.com/

Action:
(enter) 1      Mirror Web Site(s)
        2      Mirror Web Site(s) with Wizard
        3      Just Get Files Indicated
        4      Mirror ALL links in URLs (Multiple Mirror)
        5      Test Links In URLs (Bookmark Test)
        0      Quit
: 4

Proxy (return=none) :
```

Рисунок 2.13 – Запуск httrack

		Леценко Б.С.				Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		62

Зловмисник буде завжди використовувати режим Проху. Проху сервер це сервер посередник, який дозволяє клієнтам виконувати непрямі запити. Використання проху значно збільшує анонімність виконуваних дій.

Далі програма запитує, які файли потрібно викачувати. Обираємо всі (натискаємо Enter).

Після цього потрібно ввести «глибину» клонування або ввести додаткові параметри. Вводимо «-r5», що означає клонування на 5 рівнів в середину. Чим «глибше» клонувати, там кращий буде результат, але це займе значно більше часу та місця.

```
You can define wildcards, like: -*.gif +www.*.com/*.zip -*img*.zip
Wildcards (return=none) :

You can define additional options, such as recurse level (-r<number>), separated by blank spaces
To see the option list, type help
Additional options (return=none) :-r5
```

Рисунок 2.14 – Вибір додаткових параметрів в HTTrack

Після цього додаток формує виведену команду та питає дозволу на початок проведення клонування – обираємо «yes».

```
Ready to launch the mirror? (Y/n) :y

Mirror launched on Mon, 10 May 2021 12:25:03 by HTTrack Website Copier/3.49-2+libhtsjava.so.2 [XR6CO'2014]
mirroring https://www.privat24.ua/ with the wizard help..
Done.
Thanks for using HTTrack!
*
```

Рисунок 2.15 – Успішне завершення клонування

Після завершення всіх необхідних процедур по клонуванні веб сайту програма вивела відповідне повідомлення. В каталозі /home/kali/websites з'явився каталог з файлами сайту (див.2.16).

```
(kali@kali)-[~]
└─$ cd websites/fake-facebook

(kali@kali)-[~/websites/fake-facebook]
└─$ ll
total 116
drwxr-xr-x 24 kali kali 4096 May 10 12:33 ar-ar.facebook.com
-rw-r--r-- 1 kali kali 4243 May 10 12:31 backblue.gif
drwxr-xr-x 3 kali kali 4096 May 10 12:32 connect.facebook.net
drwxr-xr-x 3 kali kali 4096 May 10 12:51 developers.facebook.com
drwxr-xr-x 24 kali kali 4096 May 10 12:35 es-la.facebook.com
-rw-r--r-- 1 kali kali 828 May 10 12:31 fade.gif
drwxr-xr-x 24 kali kali 4096 May 10 12:34 fr-fr.facebook.com
drwxr-xr-x 3 kali kali 4096 May 10 12:31 hts-cache
-rw-r--r-- 1 kali kali 264 May 10 12:31 hts-in_progress.lock
-rw-r--r-- 1 kali kali 6774 May 10 13:02 hts-log.txt
drwxr-xr-x 24 kali kali 4096 May 10 12:33 hu-hu.facebook.com
-rw-r--r-- 1 kali kali 5220 May 10 12:32 index.html
drwxr-xr-x 2 kali kali 4096 May 10 13:02 l.facebook.com
drwxr-xr-x 2 kali kali 4096 May 10 12:35 messenger.com
drwxr-xr-x 19 kali kali 4096 May 10 12:41 pay.facebook.com
drwxr-xr-x 24 kali kali 4096 May 10 12:35 pl-pl.facebook.com
drwxr-xr-x 34 kali kali 4096 May 10 12:43 portal.facebook.com
```

		Леценко Б.С.				Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		63

Рисунок 2.16 – Вміст каталогу з клонованим сайтом <https://www.facebook.com/>

Якщо відкрити в браузері файл `index.html` , то виведеться клонована сторінка (див. рис.2.17).

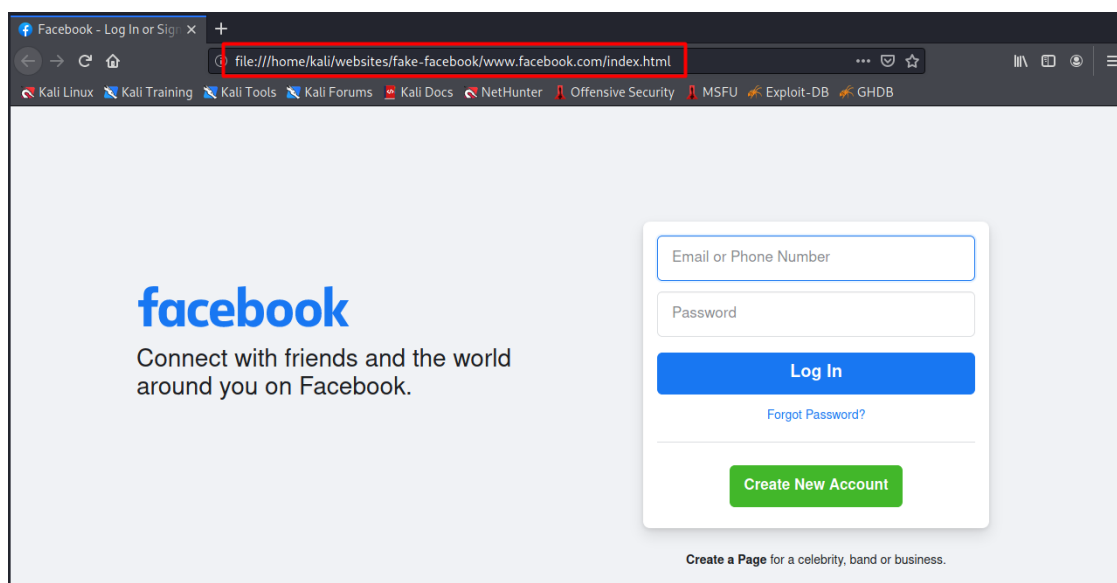


Рисунок 2.17 – Вигляд локальної зклонованої сторінки головного входу

Маніпуляцій для того, щоб користувач не помітив підробки, достатньо. Для повної роботи методу з «фішингом» необхідно змінити програмний код сторінки, щоб при натисканні кнопки входу, дані користувача надсилались на підробний сервер, але, в рамках цього дипломного проекту, реалізація цього не є доцільним. Можна допустити, що програмний код змінено та дані надсилаються одразу зловмиснику.

Недоліками даного методу клонування є:

- сучасні веб сайти можуть використовувати захист від «павуків» та подібних програмних додатків по клонуванню;
- на багатьох веб сайтах веб сторінки підвантажуються динамічно після встановлення з'єднання, а не завантажуються одразу повністю;
- веб сторінки можуть завантажитися в такому вигляді, що подальше редагування функціоналу кнопок не є можливим;
- все більше та більше веб сайтів використовують «безпарольні» методи авторизації (відправляють SMS або PUSH повідомлення на мобільні пристрої

		Лещенко Б.С.				Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		64

користувачів – це вважається безпечнішим, але спільнота все ще пропонує кращі методи).

На цьому розгляд клонування веб сторінок можна вважати завершеним.

Після успішного створення локальної копії веб сайту можна переходити до розміщення сайту на веб сервері. Для початку роботи з програмним забезпеченням Apache2 необхідно провести його встановлення. В даному випадку сервером, де буде розміщуватися веб сайт, є ,безпосередньо, комп'ютер зломисника, який знаходиться в локальній мережі.

Apache доступний в репозиторіях програмного забезпечення Kali за замовчуванням, тому його можна інсталювати за допомогою стандартних інструментів управління пакетами. Для початку вивантажмо показник локальних пакетів, щоб відобразити останні зміни, та, після цього, встановити пакет apache2 за допомогою команди *sudo apt install apache2* (див. рис.2.18).

```
(kali@kali)-[~]
$ sudo apt update
Hit:1 http://kali.koyanet.lv/kali kali-rolling InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
10 packages can be upgraded. Run 'apt list --upgradable' to see them.

(kali@kali)-[~]
$ sudo apt install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
apache2 is already the newest version (2.4.46-4).
apache2 set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 10 not upgraded.
```

Рисунок 2.18 – Встановлення Apache2 веб серверу

Після підтвердження встановлення apt виконало встановлення Apache і всіх необхідних залежностей.

При використанні веб серверу Apache можна використовувати віртуальні хости (аналогічні серверним блокам в Nginx) для інкапсуляції даних конфігурації та розміщення на одному сервері декількох доменів. За замовчуванням ввімкнено один серверний блок, який налаштовано на обслуговування документів директорії */var/www/html*. Хоча це чудово працює для окремого сайту, при хостингу декількох

		Лещенко Б.С.				Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		65

сайтів це не є зручним. Замість зміни `/var/www/html` перенесемо в `/var/www` структуру каталогів для нашого скопійованого сайту та змінимо власника каталогів за допомогою змінної середовища `$USER` (див. рис. 2.19).

```
(kali@kali)-[~]
$ sudo mv /home/kali/websites/fake-facebook /var/www/fake-facebook

(kali@kali)-[~]
$ sudo chown -R $USER:$USER /var/www/fake-facebook
```

Рисунок 2.19 – Переміщення та зміна прав каталогу з підробним сайтом

Для обслуговування даного контенту Apache необхідно створити файл віртуального хосту з правильними директивами. Замість зміни файлу конфігурації за замовчуванням, що знаходиться в `/etc/apache2/sites-available/000-default.conf`, створим новий файл в `/etc/apache2/sites-available/fake-facebook.conf`.

```
kali@kali: ~
File Actions Edit View Help
GNU nano 5.4 /etc/apache2/sites-available/fake-facebook.conf *
<VirtualHost *:80>
  ServerAdmin webmaster@localhost
  ServerName facebook.com
  ServerAlias www.facebook.com
  DocumentRoot /var/www/fake-facebook
  ErrorLog ${APACHE_LOG_DIR}/error.log
  CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Рисунок 2.20 – Створення нового віртуального хосту

Активуємо файл за допомогою інструменту `a2ensite`.

```
(kali@kali)-[~]
$ sudo a2ensite fake-facebook.conf
Enabling site fake-facebook.
To activate the new configuration, you need to run:
systemctl reload apache2
```

Рисунок 2.21 – Активація нового файлу

Відключаємо сайт за замовчуванням `000-default.conf` за допомогою інструменту `a2dissite`.

```
(kali㉿kali)-[~]
└─$ sudo a2dissite 000-default.conf

Site 000-default disabled.
To activate the new configuration, you need to run:
systemctl reload apache2
```

Рисунок 2.22 – Деактивація старого файлу

Перевіримо налаштування на помилки та перезапустимо сервіс Apache.

```
(kali㉿kali)-[~]
└─$ sudo apache2ctl configtest

AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'ServerName' directive globally to suppress this message
Syntax OK

(kali㉿kali)-[~]
└─$ sudo systemctl restart apache2
```

Рисунок 2.23 – Перевірка та перезапуск конфігурацій

Тепер Apache повинен обслуговувати доменне ім'я.

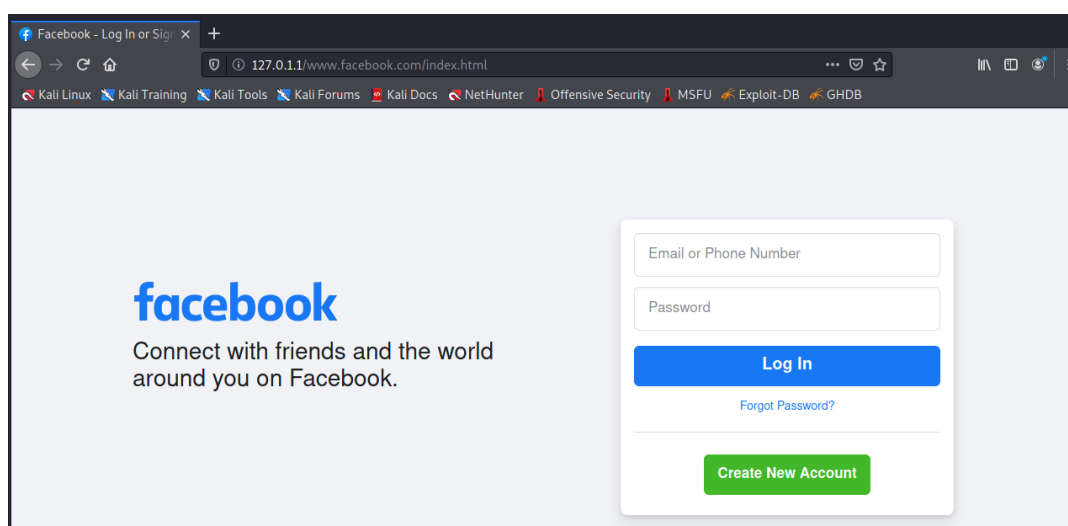


Рисунок 2.24 – Перевірка роботи клонованого сайту запущеного за допомогою Apache

Додатково перевіримо статус Apache за допомогою команди `sudo systemctl status apache2`.

		Лещенко Б.С.				Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		67

```

● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; disabled; vendor preset: disabled)
   Active: active (running) since Mon 2021-05-10 15:00:36 EDT; 5min ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 22948 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 22952 (apache2)
    Tasks: 11 (limit: 4692)
   Memory: 17.5M
      CPU: 168ms
   CGroup: /system.slice/apache2.service
           └─22952 /usr/sbin/apache2 -k start
             └─22955 /usr/sbin/apache2 -k start
               └─22956 /usr/sbin/apache2 -k start
                 └─22957 /usr/sbin/apache2 -k start
                   └─22959 /usr/sbin/apache2 -k start
                     └─23020 /usr/sbin/apache2 -k start
                       └─23022 /usr/sbin/apache2 -k start
                         └─23023 /usr/sbin/apache2 -k start
                           └─23024 /usr/sbin/apache2 -k start
                             └─23025 /usr/sbin/apache2 -k start
                               └─23026 /usr/sbin/apache2 -k start

May 10 15:00:36 kali systemd[1]: Starting The Apache HTTP Server ...
May 10 15:00:36 kali apachectl[22951]: AH00558: apache2: Could not reliably determine the server's f
May 10 15:00:36 kali systemd[1]: Started The Apache HTTP Server.
lines 1-25/25 (END)

```

Рисунок 2.25 – Результат виконання команди `sudo systemctl status apache2`

Після виконаних перевірок можна стверджувати, що налаштування було виконано успішно. Після цих дій переходимо до наступного етапу.

2.5. Здійснення атаки DNS Spoofing за допомогою Ettercap

Фінальна та най головна частина здійснення атаки – це, безпосередньо, виконання DNS підміни. Як говорилося в попередніх розділах, для виконання DNS spoofing буде використано програмне забезпечення Ettercap з графічним інтерфейсом (див.рис.2.26). Для запуску у графічному режимі необхідно написати команду `sudo Ettercap -G`.

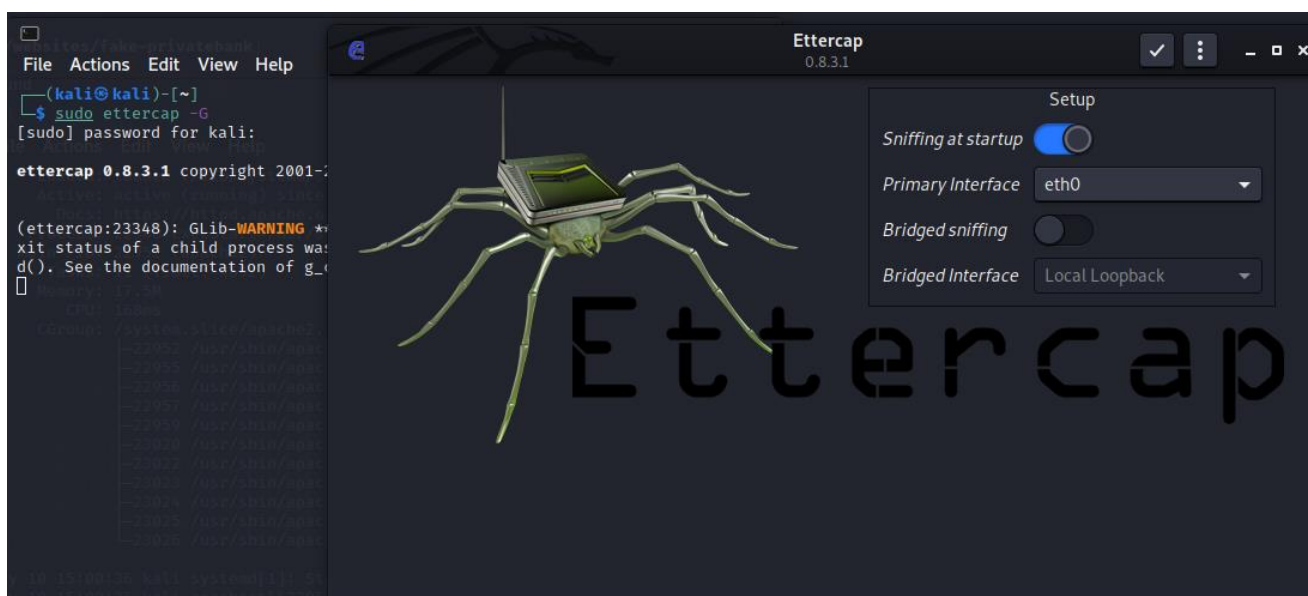


Рисунок 2.26 – Запуск та головне меню програми Ettercap

		Леценко Б.С.			Арк.
Змн.	Арк.	№ докум.	Підпис	Дата	68

Обираємо необхідний інтерфейс для виконання підміни та натискаємо галочку.

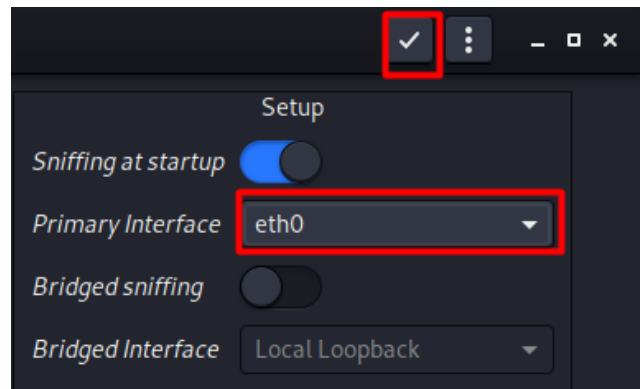


Рисунок 2.27 – Вибір потрібного інтерфейсу

Далі натискаємо кнопку «Hosts list» (відкриється вкладка зі списком всіх доступних хостів для атаки), а потім кнопку «Scan for hosts», щоб провести нове сканування.

З попередніх сканувань ми дізнались, що адресу основного шлюзу – 192.168.0.1 , а адреса необхідної нам жертви – 192.168.0.105.

Для здійснення атаки необхідно спочатку обрати зі списку IP адресу жерти, потім натиснути на кнопку «Add to Target1», а далі вже натиснути на IP адресу шлюзу та натиснути на кнопку «Add to Target2».

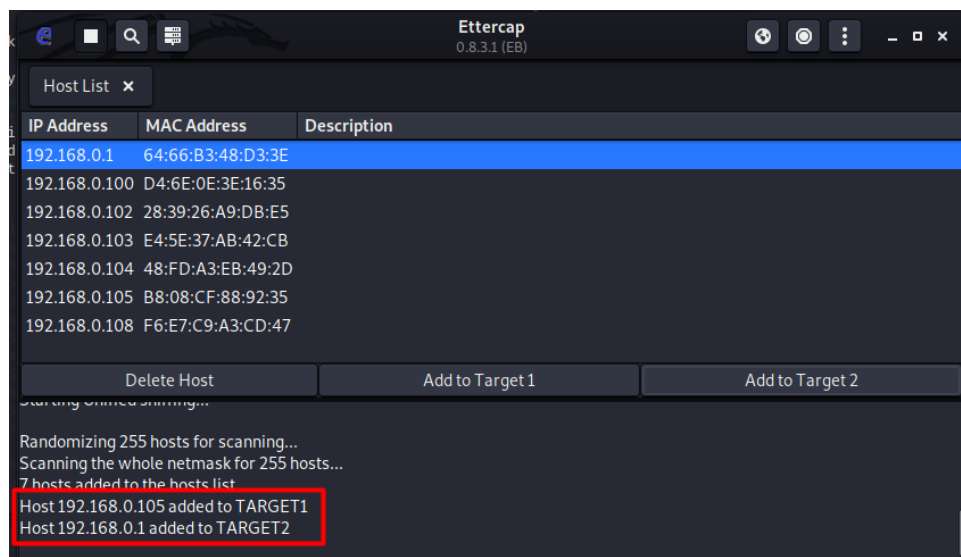


Рисунок 2.28 – Виконання дій по вибору жертв

Після цього необхідно перейти до «MITM Menu» та обрати «ARP poisoning...» (див.рис.2.29).

		Лещенко Б.С.				Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		69

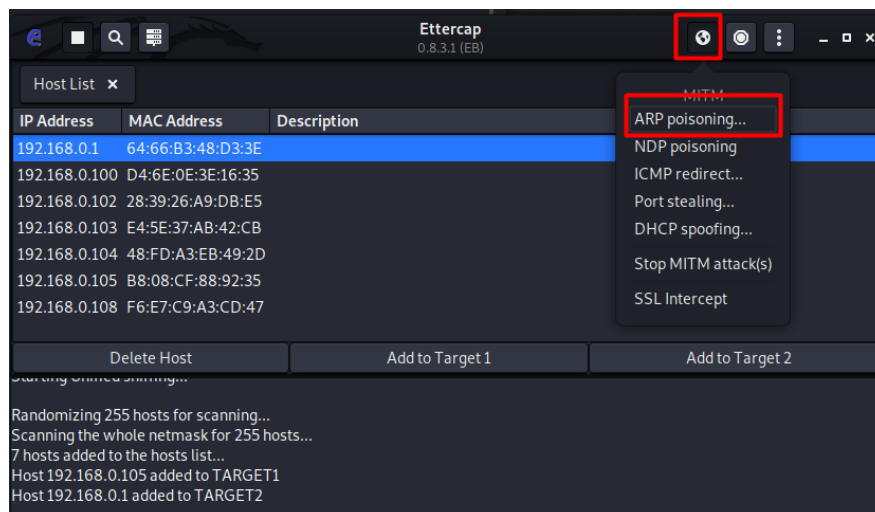


Рисунок 2.29 – Вибір необхідних меню

Після вибору необхідний пункт відкриється вікно (див.рис.2.30). Для подальшої роботи ставимо позначку напроти пункту «Sniff remote connection.» та натискаємо «ОК».

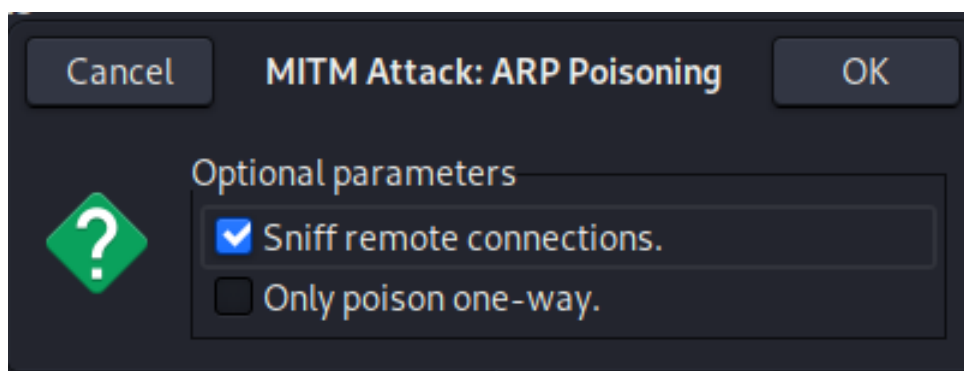


Рисунок 2.30 – Ввімкнення перехоплення віддалених підключень

Далі необхідно налаштувати, власне, правила перехоплення та перенаправлення трафіку в мережі. Конфігураційні файли для підміни DNS в Ettercap знаходяться за шляхом: `/etc/ettercap/etter.dns`. Відриваємо цей файл та робимо зміни. Синтаксис цього файлу збігається з синтаксисом файлу `hosts`. В файлі наявна довідка по використанню з прикладами, саме після цієї довідки додаємо нові рядки з метою перенаправлення жертв з легітимного сайту `facebook.com` на підробний сервер (в нашому випадку на IP адресу) (див.рис.2.31).

		Леценко Б.С.				Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		70

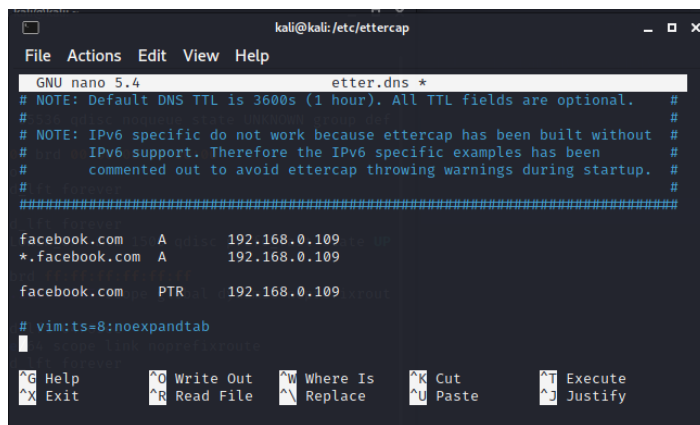


Рисунок 2.31 – Додавання нових записів в конфігураційні файли

В програмному забезпеченні наявні багато додаткових модулів. Переходимо до «Ettercap MENU» та обираємо пункт «Plugins», а далі «Manage plugins».

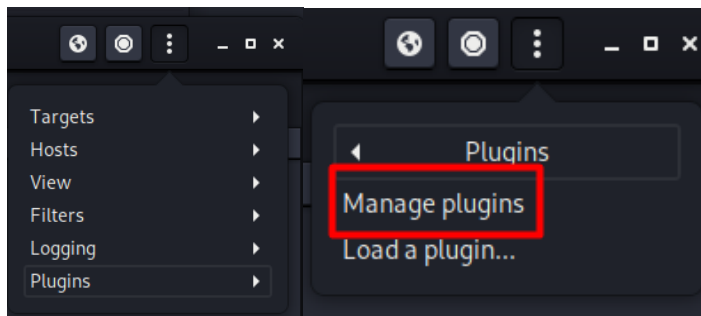


Рисунок 2.32 – Відкриття меню з плагінами

Після виконання дій описаних вище обираємо, необхідний нам, модуль «dns_spoof» та два рази натискаємо по ньому.

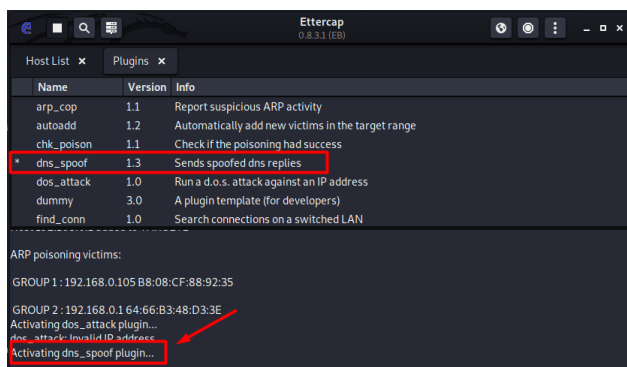


Рисунок 2.33 – Запуск плагіну dns_spoof

Біля назви плагіну з'явилась зірочка, що означає, що модуль працює. Якщо все добре, то на цьому етапі атака вже запущена та залишилось чекати поки жертва перейде на facebook.com. А так як сайт підробний, то жертву перенаправить на сторінку з введенням паролю та логіну.

		Леценко Б.С.			Арк.
Змн.	Арк.	№ докум.	Підпис	Дата	71

З хосту жертви переходимо на facebook.com (див.рис.2.34). З рисунку можна помітити дві речі: сайт виглядає на 100% ідентично оригінальному, в адресному рядку адреса відображається як <http://facebook.com/www.facebook.com/index.html> , а не як <https://facebook.com/index.html>. Таке відображення є через специфічне налаштування Apache2 веб серверу. Звичайно, це можна змінити і в реальних умовах навіть підробити SSL сертифікати для того, щоб сайт відображався ідентично на 100%. В межах дипломного проекту нам достатнього цього результату.

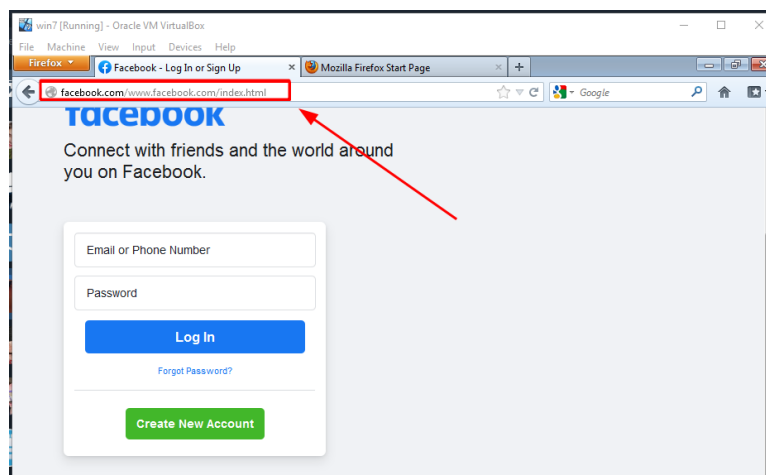


Рисунок 2.34 – Перехід на сайт з комп'ютеру жертви

В програмному додатку Ettercap з'явилися повідомлення відповідного характеру (див.рис.2.35). В цих повідомленнях знаходиться інформація про модуль, який спрацював, та до яких хостів відносилась робота модулю.

```
Activating dns_spoof plugin...
dns_spoof: A [facebook.com] spoofed to [192.168.0.105] TTL [3600 s]
DHCP: [08:00:27:A3:72:DE] REQUEST 192.168.0.107
DHCP: [08:00:27:A3:72:DE] REQUEST 192.168.0.107
DHCP: [08:00:27:A3:72:DE] REQUEST 192.168.0.107
dns_spoof: A [facebook.com] spoofed to [192.168.0.105] TTL [3600 s]
dns_spoof: A [www.facebook.com] spoofed to [192.168.0.105] TTL [3600 s]
dns_spoof: A [www.facebook.com] spoofed to [192.168.0.105] TTL [3600 s]
dns_spoof: A [www.facebook.com] spoofed to [192.168.0.105] TTL [3600 s]
dns_spoof: A [facebook.com] spoofed to [192.168.0.105] TTL [3600 s]
```

Рисунок 2.35 – Перехід на сайт з комп'ютеру жертви

		Лещенко Б.С.				Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		72

Жертва перейшла по правильному посиланню, а її було перенаправлено на підробний веб-сайт. На цьому, реалізацію атаки з підміною DNS можна вважати успішно завершеною. Методи, технології та способи захисту від побідних атак буде розглянуто в наступному розділі.

		Леценко Б.С.				Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		73

РОЗДІЛ 3. ДОСЛІДЖЕННЯ МЕТОДІВ ТА ЗАСОБІВ ДЛЯ ПОКРАЩЕННЯ БЕЗПЕКИ DNS

3.1. Важливість безпеки DNS, типові проблеми безпеки DNS та розгляд DoS атак на DNS

Частина проблем з безпекою впливає з того, що DNS - це такий сервіс, що часто «встановлюють і забувають». Той факт, що DNS настільки стійкий, в поєднанні з тим, що реєстрація доменів відбувається не часто, і те, що не багато команд по безпеці систем мають досвід роботи з DNS, і занадто мало адміністраторів DNS, які мають досвід у безпеці – це все створює певні проблеми у захисті інфраструктури DNS. Поєднання внутрішній проблем із зовнішніми загрозами безпеки DNS, з якими стикається організація: DDoS атаки на основі DNS, «отруєння кешу», шкідливе програмне забезпечення, яке використовує DNS для цілей управління – безпека DNS стає потенційним кошмаром для будь-якої команди.

Для того, щоб краще розібратися з потенційними загрозами, потрібно дослідити вже існуючі випадки використання вразливостей DNS. Звичайно, розглянути всі випадки неможливо, але найбільші та найцікавіші випадки розглянути варто.

Так в 1996 році Eugene Kashpureff використав «отруєння» кешу DNS, щоб перенаправити трафік з сайту InterNIC на свій власний сайт. «Отруєння» тривало декілька днів поки Eugene Kashpureff не повернув все як було.

В січні 2001 року зловмисник перервав доступ до всіх серверів Microsoft. Адміністратори Microsoft зробили реалізацію цієї атаки простою – вони розмістили всі свої сервери імен в один сегмент мережі, що значно полегшило здійснення атаки.

В червні 2008 року групою турецьких хакерів, які називали себе NetDevilz, використовуючи соціальну інженерію, обманом отримали доступ до реєстраторів доменів ICANN та IANA – перенаправляли трафік на свої сервери. Записи було

		Лещенко Б.С.				Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		74

змінено лише на 20 хвилин, але багато користувачів перенаправлялись на невірні сайти ще добу.

Також в 2008 році Dan Kaminsky опублікував деталі про «Баг Камінського», який дозволяв зловмисникам надсилати авторитетні відповіді доменам, для яких сервер не був авторитетним. Наприклад, користувач міг відвідати "reallyfunwebgames.com", а авторитетний сервер імен "reallyfunwebgames.com" також надсилав авторитетну відповідь для "americanexpress.com". Таким чином, кожен користувач, який покладався на той самий рекурсивний сервер, що і початковий користувач, тепер переходив на неправильну сторінку, коли намагався перейти на americanexpress.com. Камінський зміг спроектувати це, поєднавши недолік, коли DNS-сервери управляли ідентифікаторами запитів та разом із технікою «отруєння» кешу. На відміну від інших атак у цьому списку, Камінський є відповідальним дослідником і повідомив про помилку відповідальних людей, щоб її можна було виправити, перш ніж він опублікував подробиці дослідження для широкої громадськості.

Усі ці атаки служать для ілюстрування ряду моментів. Перший полягає в тому, що в деяких випадках атак можна було запобігти, якщо була б посилена політика безпеки DNS. Другий момент, і, певним чином, найважливіший з них, полягає в тому, що великі компанії, підковані в галузі безпеки та навіть компанії з великим досвідом роботи з DNS все ще можуть бути вразливими до атак. Третій момент, який слід зазначити на цих прикладах, - це еволюція атак з часом.

Перед тим як команда по безпеці зможе ефективно захищати DNS інфраструктуру, вона повинна спочатку визначити, які ризики існують для інфраструктури. Проводячи оцінку ризиків DNS інфраструктури важливим є зробити широкий огляд ризиків безпеки. Ціллю складання плану по безпеці DNS є переконатися, що інфраструктура DNS доступна максимально можливу кількість часу і що відправляється коректна інформація до машин, які роблять запити.

Одна з основоположних причин детального розгляду безпеки DNS є той факт, що у фреймворку DNS існує купа вразливих точок. Наряду з традиційними

		Лещенко Б.С.				Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		75

вразливостями пов'язаними з поломкою фізичного обладнання, неавторизованим доступом до серверів, DDoS атаки, існують також і унікальні виключно для DNS проблеми – адміністративні проблеми реєстратору доменів, млявий маркетинг та інші. Розподілений характер DNS автоматично вимагає іншого рівню хвилювання про безпеку та додає складності плануванню безпеки.

Цілеспрямована DNS атака може приймати різні форми, в залежності від наміру нападника.

Тому що DNS використовує UDP для комунікації, це робить цей протокол особливо вразливим до атак. Одним із найпоширеніших методів нападу є насичення атакованого комп'ютера або мережевого устаткування великою кількістю зовнішніх запитів (часто безглузвих або неправильно сформульованих) таким чином атаковане устаткування не може відповісти користувачам, або відповідає настільки повільно, що стає фактично недоступним. Взагалі відмова сервісу здійснюється:

— примусом атакованого устаткування до зупинки роботи програмного забезпечення/устаткування або до витрат наявних ресурсів, внаслідок чого устаткування не може продовжувати роботу;

— заняттям комунікаційних каналів між користувачами і атакованим устаткуванням, внаслідок чого якість сполучення перестає відповідати вимогам.

Якщо атака відбувається одночасно з великої кількості IP-адрес, то її називають розподіленою (англ. Distributed Denial-of-Service — DDoS).

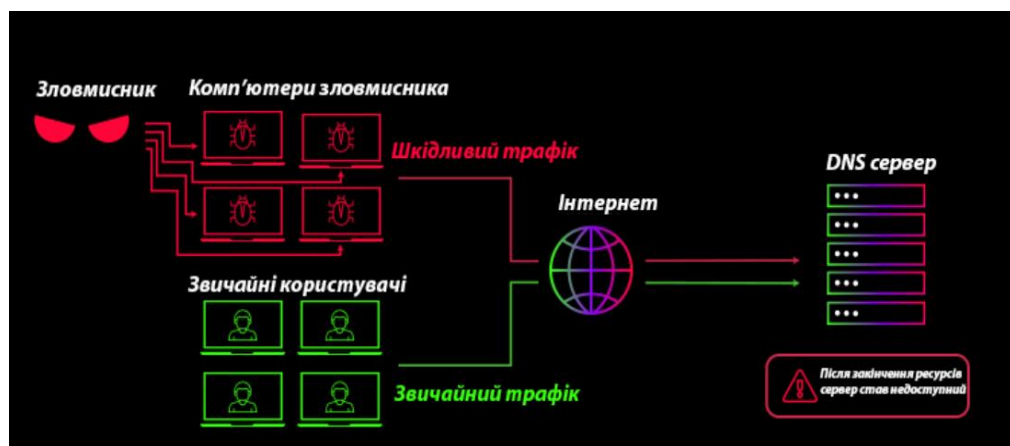


Рисунок 3.1 – Схема виконання DDoS атаки.

3.2. DNS over HTTPS та Oblivious DNS over HTTPS

DNS over HTTPS (DoH) — протокол для виконання запитів DNS по протоколу HTTPS (див.рис.3.2). Метою цього методу є збільшення конфіденціальності та безпеки користувачів шляхом запобігання перехоплення і маніпулювання даними DNS за допомогою атак типу «Man in the middle». Опубліковано IETF в RFC 8484 в жовтні 2018 року. На додачу до підвищення безпеки, ще одна ціль DNS over HTTPS – підвищити продуктивність роботи: тестування DNS резольверів ISP показало, що в багатьох випадках вони мають неочікувано повільний час відгуку.

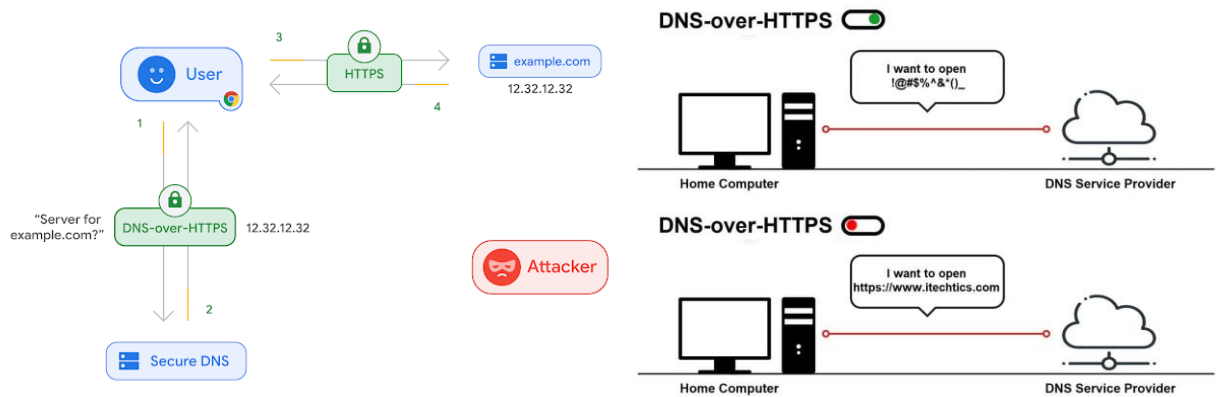


Рисунок 3.2 – Схема роботи DNS over HTTPS.

В публічно реалізованій версії цього протоколу Google використовує HTTP GET запити (через HTTPS) для доступу до інформації DNS з використанням кодування DNS запиту і параметрів результату, представлених в нотації JSON.

В липні 2019 року спеціалісти компанії Qihoo 360 виявили перший в світі зразок шкідливого програмного забезпечення, який використовує протокол DNS over HTTPS.

На перший погляд, масове впровадження DNS-over-HTTPS в програмне забезпечення, яке працює в Інтернеті викликає тільки позитивну реакцію, але даний протокол має і негативні сторони.

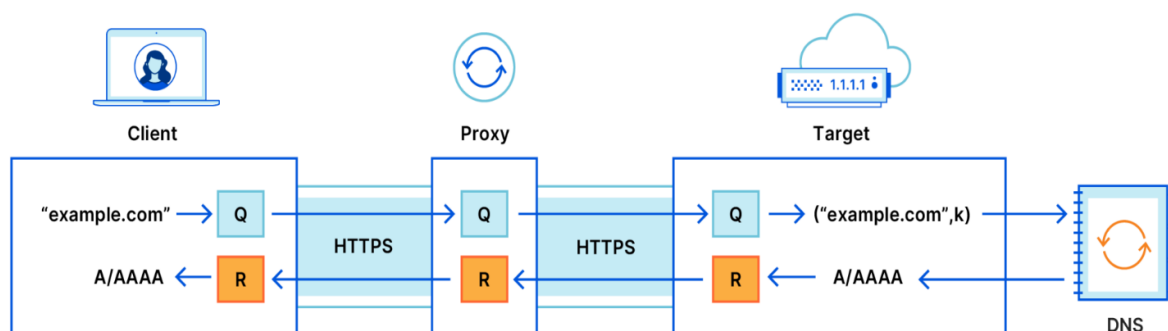
Першою проблемою, яка обмежує сферу масового використання DoH, є його орієнтація виключно на веб трафік. Дійсно, протокол HTTP і його актуальна

редакція HTTP/2, на якій базується DoH, є основою WWW, але Інтернет це не лише веб. Існує багато популярних сервісів, таких, як електронна пошта, всеможливі месенджери, системи передачі файлів, стрімінг мультимедіа і т.д., які не використовують HTTP. Таким чином, не дивлячись сприйняття DoH як панацеї, його неможливо застосовувати ніде, крім браузерних технологій.

Другою проблемою, яка потенційно ще більш значуща, ніж перша, є фактичне відмовлення від головної парадигми DNS – децентралізації. Зокрема, Mozilla пропонує використовувати сервіс від Cloudflare. Подібний сервіс запустили також і інші помітні фігури Інтернет, зокрема Google. Виходить, що впровадження DNS-over-HTTPS в тому вигляді, в якому це пропонується зараз, лише збільшує залежність кінцевих користувачів від найбільших сервісів. Не секрет, що інформація, яку може надати аналіз DNS запитів здатний збирати ще більше даних про нього, а також підвищити їх точність і актуальність.

В серпні 2020 року у співавторстві таких компаній, як Cloudflare, Apple, and Fastly, було представлено новий стандарт DNS – Oblivious DNS-over-HTTPS, який відділяє IP адреси та запити. Цей протокол зробили відкритим та кожний бажаючий може ознайомитись з його вихідним кодом на github.

ODOH - це новий протокол, який розробляється в IETF. ODOH працює шляхом додавання рівня шифрування з відкритим ключем, а також мережевого проксі між клієнтами та серверами DoH, наприклад 1.1.1.1. Поєднання цих двох доданих елементів гарантує, що лише користувач має доступ одночасно до повідомлень DNS та власної IP-адреси (див.рис.3.3).



		Леценко Б.С.				Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		78

Рисунок 3.3 – Схема роботи Oblivious DNS over HTTPS.

На рисунку вище зображено шлях DNS запиту по ODoH. **DNS сервер** дешифрує запити, які були зашифровані клієнтом, через проксі. Так само, DNS сервер шифрує відповіді і повертає їх проксі. **Проксі** працює так, як проксі і має працювати – він передає повідомлення між клієнтом та сервером. **Клієнт** поводить себе так само, як і при DNS або DoH, різниця полягає лише в шифрованих запитах для серверу та дешифруванні відповідей.

Отже, додавання шифрування та проксування надає наступні гарантії:

- Сервер бачить лише запит та IP-адресу проксі.
- Проксі не має можливості перегляду повідомлень DNS, не має можливості ідентифікувати, читати чи модифікувати запит, що надсилається клієнтом, або відповідь, яку повертає сервер.
- Тільки потрібний сервер може прочитати зміст запиту та дати відповідь.

3.3. Технологія DNS – Security Extensions та DNSCrypt

Члени Інженерної проектної групи інтернету (IETF) — організації, що відповідає за стандарти протоколу DNS — давно усвідомили проблему недостатньої надійності механізмів перевірки справжності в DNS. Праця над розробкою вирішення почалась в 1990-их роках, та її результатом став протокол DNSSEC (DNS Security Extensions — розширення безпеки системи доменних імен).

Протокол DNSSEC дозволяє підвищити надійність перевірки справжності в DNS за допомогою цифрових підписів, що ґрунтуються на криптографії відкритого ключа. При використанні DNSSEC не запити і відповіді DNS підписуються криптографічно, а самі дані DNS підписуються власником цих даних.

У кожної DNS зони є пара відкритих\закритих ключів. Власник зони використовує її закритий ключ для підпису даних DNS в цій зоні і генерації цифрових підписів для цих даних. Як можна визначити з назви, «закритий ключ» — це ключ дані про який тримаються власником зони в таємниці, а «відкритий

		Лещенко Б.С.				Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		79

ключ» публікується в ній вільно, і його отримати може кожен. Будь який рекурсивний резольвер, який виконує пошук даних в зоні, також отримує відкритий ключ цієї зони і використовує його для виконання перевірки справжності даних DNS. Резольвер перевіряє справжність цифрового підпису отриманих DNS даних. Якщо справжність підтверджується, то дані DNS вважаються справжніми і повертаються користувачу. Якщо підпис не проходить перевірку справжності, то резольвер вважає, що відбулась атака, та позбувається від даних та повідомляє користувача про помилку.

Використання DNSSEC дозволяє забезпечити дві важливі функції в DNS:

— *Перевірка справності джерела даних* дозволяє резольверу криптографічно перевірити, чи дійсно отримані дані поступили із тієї ж зони, звідки, як він вважає, вони виникли;

— *перевірка цілісності даних* дозволяє резольверу перевірити, чи не були ці дані змінені при передачі, після того як власник зони підписав їх закритим ключем цієї зони.

Кожна зона публікує свій відкритий ключ, і рекурсивний резольвер отримує його для перевірки справжності даних в цій зоні. Так само, як і всі інші дані в зоні, відкритий ключ зони також підлягає підпису. Але відкритий ключ зони підписується не її ж закритим ключем, а закритим ключем її батьківської зони. Виключенням є коренева (root) зона. — її ключ підписувати нікому.

Таким чином, відкритий ключ кореневої зони є важливою відправною точкою перевірки справжності даних DNS. Якщо резольвер довіряє відкритому ключу кореневої, то він може довіряти відкритим ключам зон верхнього рівня, які підписані її закритим ключем.

		Лещенко Б.С.				Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		80

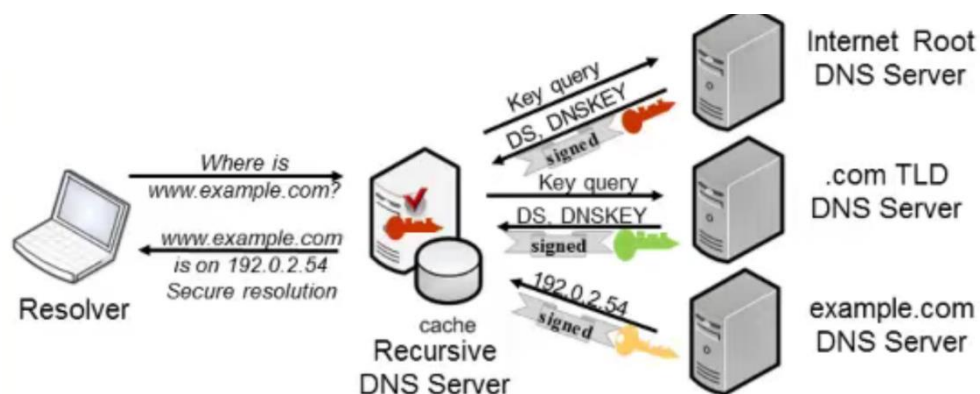


Рисунок 3.4 – Схема роботи DNSSEC

Послідовність криптографічних ключів, які використовуються для підпису інших криптографічних ключів, називається *ланцюгом довіри*. Відкритий ключ, який знаходиться на початку ланцюга довіри, називається *якорем довіри*. В резольвера є список якорів довіри — відкритих ключів різних зон, яким резольвер безумовно довіряє. Для більшості резольверів заданий лише один якор — відкритий ключ кореневої зони. Довіряючи цьому ключу, що займає найвище місце в ієрархії DNS, резольвер може вибудувати ланцюг довіри до будь якого місця в просторі імен, якщо всі зони на шляху до цього місця є підписаними.

Для забезпечення в інтернеті безпеки необхідне масштабне впровадження DNSSEC. DNSSEC не вмикається автоматично: на сьогоднішній день, мережеві оператори повинні вмикати його вручну на своїх рекурсивних резольверах, а власники домених імен — на авторитативних серверах своїх зон. Оператори резольверів і авторитативних серверів мають різні причини для ввімкнення DNSSEC на своїх системах, але якщо їх ввімкнути, то на отримання справжніх відповідей на свої запити DNS може розраховувати більша кількість користувачів. Іншими словами, користувач може бути впевнений в тому, що він в інтернеті потрапить саме туди, куди він бажає, а не на підробні веб сторінки.

Ввімкнути на рекурсивних резольверах DNSSEC-валідацію не є складно. Практично всі розповсюджені резольвери підтримують цю можливість вже багато років. Для її ввімкнення необхідно змінити всього декілька стрічок в файлі налаштувань резольвера. З цього моменту, якщо користувач запитує інформацію DNS, яка поступає з підписаних зон, і ці дані було змінено, то користувачу дані не

		Лещенко Б.С.				Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		81

повертаються (спеціально). Застосування протоколу DNSSEC захищає користувача від неправдивих даних з підписаної зони, так як дозволяє виявити атаку і заважає отриманню користувачем змінених даних.

Підписання зон за допомогою DNSSEC відбувається за декілька дій, але свою інформацію DNS підписують мільйони зон, для того, щоб гарантувати отримання правильних даних користувачам, які валідують резольверів. Майже все розповсюджене програмне забезпечення для авторитативних DNS серверів підтримує підписання зон, та багато провайдерів DNS хостингів також підтримують DNSSEC. Як правило, ввімкнути DNSSEC в зоні, де використовуються послуги хостинг-провайдерів, дуже легко.

В 2018 році ICANN (Internet Corporation for Assigned Names and Numbers — Інтернет-корпорація з присвоєння імен та номерів) вперше змінила якір довіри кореневої зони DNS, що дозволило винести дуже корисні уроки відносно DNSSEC. Завдяки цьому також підвищилась усвідомленість відносно DNSSEC серед операторів резольверів, які ввімкнули валідацію на своєму обладнанні, а світ отримав більш чітке уявлення про роботу системи DNSSEC в цілому. В найблищі роки ICANN сподівається на більш розповсюджене розгортання DNSSEC — як операторами резольверів, так і власниками зон. Це дозволить забезпечити більш надійну криптографічний захист завдяки DNSSEC та гарантувати отримання вірних відповідей на запити до DNS для більшої кількості користувачів.

«Остання миля» — канал, який з'єднує кінцеве користувацьке обладнання з вузлом зв'язку провайдера. DNSCrypt є одним із методів захисту «останньої милі» DNS трафіку і вирішенням цілого класу проблем безпеки протоколу. Все більше користувачів в світі підключаються до різних безпроводних мереж впродовж дня, необхідність в подібному інструменті очевидна.

Аналогічним чином, як SSL перетворює веб трафік HTTP в зашифрований HTTPS трафік, DNSCrypt перетворює звичайний DNS трафік в зашифрований DNS трафік, який є захищений від відстеження та MITM атак.

		Лещенко Б.С.				Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		82

DNSCrypt не вимагає якихось змін для доменних імен і способів їх роботи, а просто надає спосіб для безпечного шифрування каналу зв'язку між користувачем та власними DNS серверами в дата-центрах. Проект DNSCrypt має відкритий вихідний код та його викладено на GitHub.

DNSCrypt має потенціал найефективнішого інструменту в Інтернет безпеці нарівні з SSL, який значно покращує захист і конфіденційність користувачів Інтернету. DNSCrypt не використовує криптографічні бібліотеки та покладається на еліптичну криптографію, зокрема криптопрімітив Curve25519.

Важливо значи, що при використанні фаєрволів чи іншого програмного забезпечення для управління мережевими пакетами, бажано вмикати DNSCrypt по TCP з використанням 443 порту. В цьому випадку більшість фаєрволів будуть вважати, що мають справу з HTTPS трафіком і проігнорують його контроль.

Встановити DNSCrypt можливо на всі сучасні популярні операційні системи, в тому числі і Solaris, Android, Windows, macOS, linux та різні системи сімейства BSD.

Для встановлення служби DNSCrypt-проху в ОС Windows потрібно, для початку завантажити та встановити його з офіційного репозиторію: <https://github.com/DNSCrypt/dnscrypt-proxy/releases>.

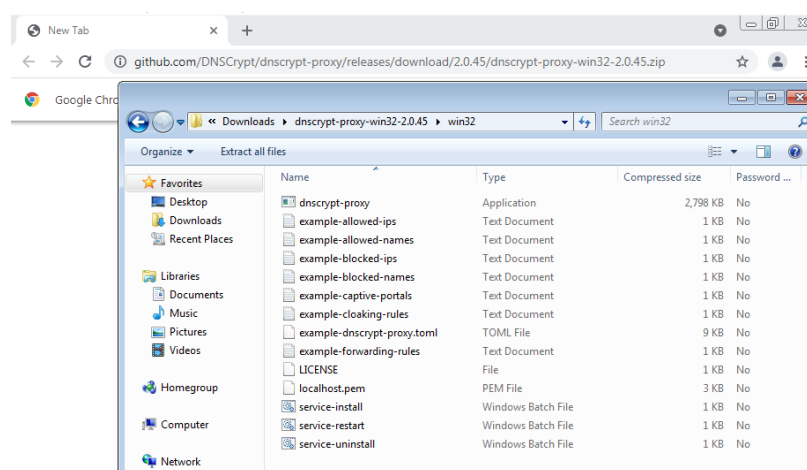


Рисунок 3.5 – Завантажений DNSCrypt

Для перевірки роботи програми, за допомогою сервісу <http://dnsleak.com/> перевіримо чи відбувається «витік» даних про IP. Переходимо по посиланню та спостерігаємо результат (див.рис.3.6).

		Лещенко Б.С.			Арк.
Змн.	Арк.	№ докум.	Підпис	Дата	
					83

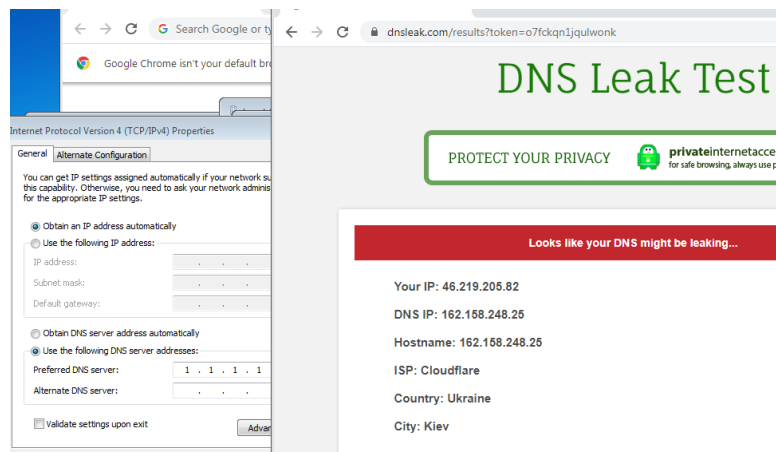


Рисунок 3.6 – Перевірка за допомогою сервісу

З інформації вище дізнаємось, що веб сайт отримав дані DNS серверу, який прописано на хості, в даному випадку — Cloudflare. Якщо DNS не прописано вручну, то буде показано інформацію про DNS сервер провайдера інтернет послуг. Тепер встановимо DNSCrypt.

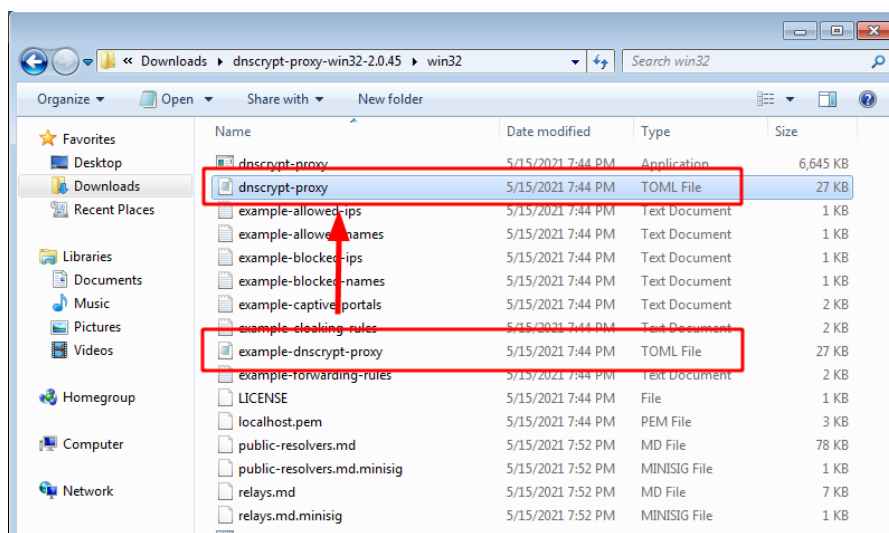


Рисунок 3.7 – Встановлення DNSCrypt

Далі копіюємо файл з прикладом конфігурації і називаємо його dnscrypt-proxy.toml . Нам достатньо стандартних налаштувань, бо DNSCrypt сервер не було налаштовано.

Далі переходимо в налаштування мережевого адаптеру та змінюємо IP адресу DNS серверу на IP адресу localhost.

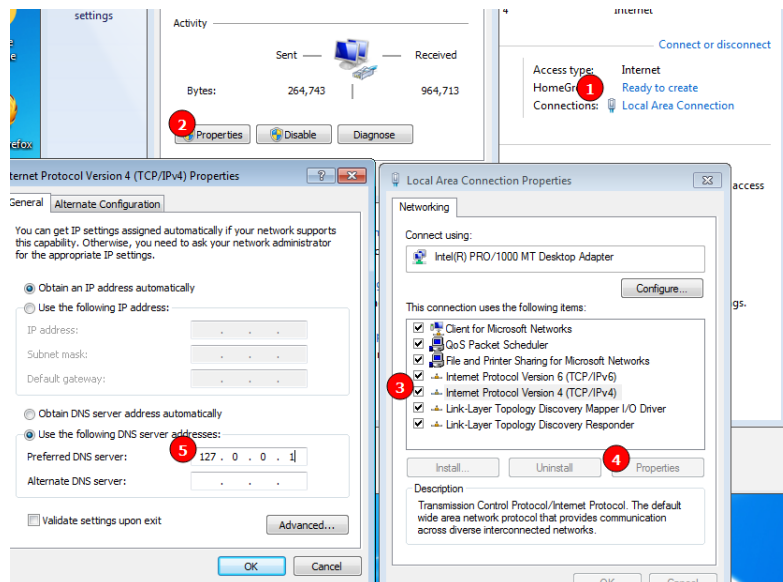


Рисунок 3.8 – Зміна адреси DNS серверу

Після виконаних дій запускаємо «service-install.bat» з правами адміністратора та виконуємо перезавантаження комп'ютеру.

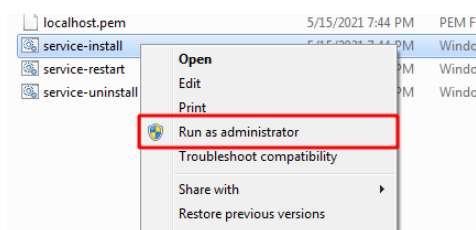


Рисунок 3.9 – Запуск service-install.bat

Перезавантажуємо комп'ютер та виконуємо ще одну перевірку на <https://dnsleak.com>.

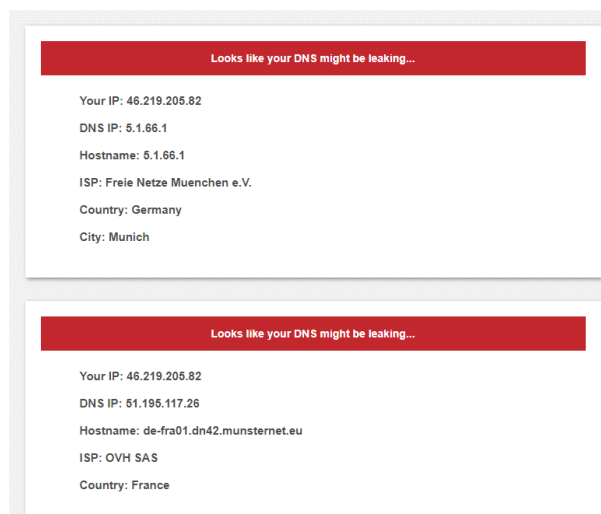


Рисунок 3.10 – Повторна перевірка з запуском DNSCrypt

		Лещенко Б.С.			Арк.
Змн.	Арк.	№ докум.	Підпис	Дата	85

Після повторної перевірки отриманий результат відрізняється від попереднього. В новому результаті відображаються інші IP адреси DNS серверів, які належать до інших країн, а не до України, як в попередньому. Отже, можна зробити висновок, що DNS трафік успішно шифрується.

3.4. Засоби запобігання ARP spoofing

Для запобігання конкретної атаки, яку було здійснено у другому розділі, необхідно дослідити механізми захисту мережі від атак канального рівня, а, конкретніше, атак на протокол ARP. Для таких цілей існує багато різного програмного забезпечення, наприклад — ArpON та arpwatch.

ArpON (ARP handler inspection) — прикладне програмне забезпечення для кінцевих хостів, що робить протокол ARP безпечнішим, та створений для уникнення можливих MITM атак через підміну ARP або отруєння кеш пам'яті ARP.

ArpON має багато особливостей. Перше, це ПЗ є безкоштовним і розповсюджується по відкритій ліцензії BSD, що означає вільну можливість зміни та використання цього ПЗ, навіть в комерційних цілях. ArpON використовується великою кількістю користувачів, в тому числі системними адміністраторами та науковими дослідниками. Через свою велику популярність є перевіреним та надійним рішенням. Також впродовж років ArpON було протестовано різноманітним програмним забезпеченням для дослідження MITM атак.

Аналогічним ПЗ є arpwatch. Arpwatch це демон, який відслідковує відповідності між IP та MAC адресами, і при виявленні аномалій, повідомляє про це в Syslog. Використовується як один з інструментів для боротьби з ARP spoofing. Демон аналізує ARP відповіді на мережевому інтерфейсі, до якого прив'язаний, та запам'ятовує відповідність IP та MAC адрес. Як тільки він помічає, що відповідність порушено, або фіксує появу нових адрес в мережі, то він повідомляє про це в системний журнал.

Ім'я інтерфейсу, на якому відбувається прослуховування, задається в якості аргументу командного рядку при запуску демона. Якщо він запускається стартовим

		Лещенко Б.С.				Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		86

скриптом, то інтерфейс вказується там, де в операційній системі прийнято вказувати ключі для демонів, що запускаються. Наприклад, в FreeBSD — в */etc/rc.conf*, а в Debian GNU/Linux — в */etc/default/arpwatch*.

```
Jan 23 16:23:54 cholpon arpwatch: new station 192.168.0.102 00:1a:4d:80:ce:78 eth0
```

Рисунок 3.11 – Приклад повідомлення arpwatch в системному журналі.

Повідомлення вище вказує на те, що в мережі з'явився новий хост.

Таблиця відповідностей знаходиться в файлі */var/lib/arpwatch/arp.dat* і виглядає як звичайний текстовий файл з MAC адресою, IP адресою, часом потрапляння в таблицю, та ім'ям інтерфейсу, через який прийшов початковий запит.

ArpON та arpwatch працюють лише в сімействі операційних систем Linux, для ОС сімейства Windows існує програмний додаток Хагр з графічним інтерфейсом.

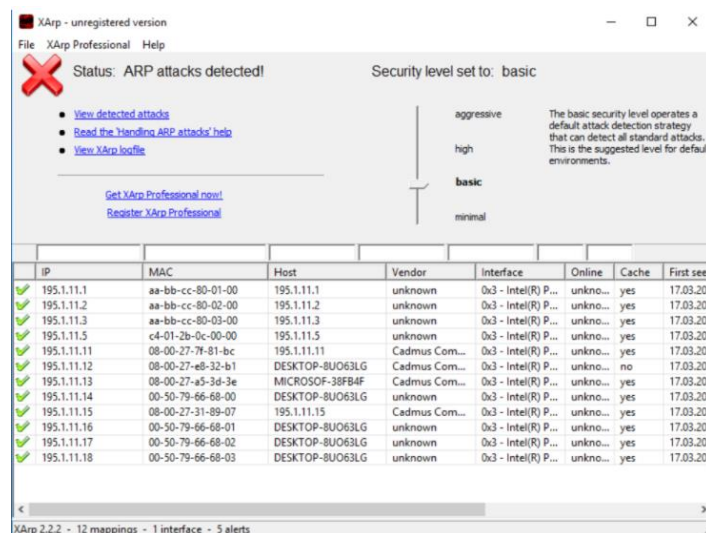


Рисунок 3.12 – Програмний додаток Хагр.

Існує також Хагр для Ubuntu, та сам Хагр є безкоштовним, але присутня і платна версія зі збільшеним функціоналом. Для демонстрації роботи нам достатньо безкоштовної версії.

Програма Хагр не лише повідомляє про якісь інциденти в локальній мережі, але й запобігає підміні ARP запитів. Має чотири режими роботи з різним рівнем можливостей.

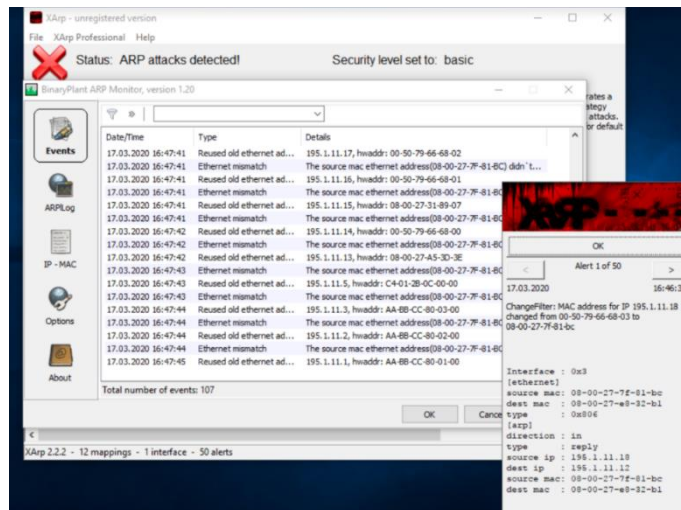


Рисунок 3.12 – Приклад виявлення ARP атаки програмним застосунком Хагр.
На цьому розгляд засобів запобігання Агр атак можна вважати завершеним.

		Леценко Б.С.				Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		88

ВИСНОВКИ

В ході дослідження системи доменних імен було, розкрито тему проекту підсистеми захисту сервісів DNS гетерогенної мережі організації. Було розглянуто значну кількість профільної літератури та створено схему для дослідження слабких місць доменної системи імен.

В контексті дослідження системи доменних підґрунтям була документація Cisco, навчальні посібники авторів Cricket Liu, Donahue G., Liska A. та Paul Albitz, документація, яка опублікована IETF в документах RFC 8499, 1034, 2929 та ін.

В першому розділі дипломного проекту було розглянуто теоретичні відомості про принципи роботи системи доменних імен, а саме: загальні відомості про DNS, структуру доменних імен, структуру пакету DNS, типи записів доменної системи імен, моніторинг за допомогою різних програмних додатків. Також було розроблено схему комплексної атаки на DNS та було описано програмне забезпечення для проведення атак.

В другому розділі було здійснено практичне налаштування серверу системи доменних імен. Було налаштовано DNS сервер за допомогою вбудованих функцій Cisco PacketTracer, маршрутизатор Cisco в ролі серверу DNS, ОС Windows Server в ролі DNS серверу. Було здійснено сканування мережі, здійснено клонування веб сторінки соціальної мережі, розміщено її на власному сервері на базі Apache2 та було здійснено атаку DNS spoofing.

В третьому розділі було досліджено методи та засоби для покращення безпеки DNS, а саме: розглянуто всі можливі атаки пов'язані з DNS, досліджено протоколи DNS over HTTPS та Oblivious DNS over HTTPS, досліджено технології Технологія DNS Security Extensions та DNSCrypt. Та, наприкінці, досліджено засоби запобігання ARP spoofing.

Було досліджено загальні відомості, принципи, терміни та практичне налагодження систем доменних імен з використанням емуляторів мережевого програмного забезпечення, такого як Cisco PacketTracer, Graphical Network Simulator-3 та системи віртуалізації операційних систем - VitrualBox. Також було

		Лещенко Б.С.				Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		89

продемонстровано важливість підтримки безпеки DNS на підприємстві та на кінцевих пристроях.

В результаті ми отримали налагоджену комп'ютерну мережу з працюючою системою доменних імен та звіти з програм, які надають діагностичну інформацію про цю систему. Другим завданням було оволодіння навичками роботи з програмами дослідження доменної системи імен. Було набуто навичок сканування мережі, набуття навичок клонування веб сторінок та здобуття навичок здійснення атак на DNS, а в особливості, виконання DNS Spoofing атак. Було набуто навичок запобігання атак на DNS та навичок забезпечення безпеки доменної системи імен.

Дослідження було проведено за допомогою системного підходу.

В ході досліджень використано сучасні методи, станом на 2021 рік.

Було розкрито тему проекту підсистеми захисту сервісів DNS гетерогенної мережі організації. Тому можна вважати завдання цього дипломного проекту виконаним.

		Лещенко Б.С.				Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		90

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Liska A. DNS Security Defending the Domain Name System / A. Liska, S. Geoffrey. – Cambridge: Elsevier Inc, 2016. – 212 с.
2. Donahue G. Network Warrior / Gary A. Donahue. – Sebastopol: O'Reilly Media, Inc, 2011. – 785 с.
3. CCIEv5 DHCP/DNS/DHCPv6 Labs [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: <https://learningnetwork.cisco.com/s/article/cciev5-dhcp-dns-dhcpv6-labs>.
4. Sidheeq S. Configure Cisco Router As DNS Servers In GNS3 – Step By Step [Електронний ресурс] / Saifudheen Sidheeq. – 2020. – Режим доступу до ресурсу: <https://getlabsdone.com/configure-cisco-router-as-dns-server-in-gns3-step-by-step/>.
5. Белов А. Настройка DNS-сервера на Windows Server 2012 и старше [Електронний ресурс] / Антон Белов. – 2019. – Режим доступу до ресурсу: <https://serverspace.ru/support/help/configuring-a-dns-server-on-windows-server-2012-or-later/>.
6. Domain Name System (DNS) Parameters [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: <https://www.iana.org/assignments/dns-parameters/dns-parameters.xhtml#dns-parameters-2>.
7. Keen P. DNS: The Good Parts [Електронний ресурс] / Pete Keen. – 2016. – Режим доступу до ресурсу: <https://www.petekeen.net/dns-the-good-parts>.
8. PETTERS J. DNS Security Guide [Електронний ресурс] / JEFF PETTERS. – 2020. – Режим доступу до ресурсу: <https://www.varonis.com/blog/dns-security/>.
9. Arends R. DNS Security Introduction and Requirements [Електронний ресурс] / R. Arends, R. Austein, M. Larson. – 2005. – Режим доступу до ресурсу: datatracker.ietf.org/doc/html/rfc4033.
10. Thomson S. Dynamic Updates in the Domain Name System (DNS UPDATE) [Електронний ресурс] / S. Thomson. – 1997. – Режим доступу до ресурсу: <https://datatracker.ietf.org/doc/html/rfc2136>

		Леценко Б.С.				Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		91

11. Karasaridis A. DNS Security: In-depth Vulnerability Analysis and Mitigation Solutions / Anestis Karasaridis., 2012. – 334 с.
12. Hermann C. DNSCrypt [Электронный ресурс] / Christian Hermann – Режим доступа до ресурсу: [https://www.dnscrypt.org/..](https://www.dnscrypt.org/)
13. Liu C. DNS and BIND / Cricket Liu., 2006. – 642 с.
14. Jeftovic M. Managing Mission - Critical Domains and DNS: Demystifying nameservers, DNS, and domain names / Mark Jeftovic., 2018. – 366 с.
15. Mockapetris P. DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION [Электронный ресурс] / P. Mockapetris. – 1987. – Режим доступа до ресурсу: <https://datatracker.ietf.org/doc/html/rfc1035>.
16. Domain Name System (DNS) Parameters [Электронный ресурс]. – 2020. – Режим доступа до ресурсу: <https://www.iana.org/assignments/dns-parameters/dns-parameters.xhtml#dns-parameters-2>.
17. Mockapetris P. Domain System Changes and Observations [Электронный ресурс] / Paul Mockapetris. – 1986. – Режим доступа до ресурсу: <https://datatracker.ietf.org/doc/html/rfc973>.
18. Таненбаум Э. Компьютерные сети. 5-е изд. / Э. Таненбаум , Д. Уэзеролл – СПб.: Питер, 2012. – 960 с.
19. Ornaghi A. THE ETTERCAP PROJECT [Электронный ресурс] / Alberto Ornaghi – Режим доступа до ресурсу: <https://www.ettercap-project.org/about.html>..
20. Keen P. DNS: The Good Parts [Электронный ресурс] / Pete Keen. – 2016. – Режим доступа до ресурсу: <https://www.petekeen.net/dns-the-good-parts>.
21. Mockapetris P. DOMAIN NAMES - CONCEPTS AND FACILITIES [Электронный ресурс] / P. Mockapetris. – 1987. – Режим доступа до ресурсу: <https://datatracker.ietf.org/doc/html/rfc1034>.
22. DNSSEC – What Is It and Why Is It Important? [Электронный ресурс] – Режим доступа до ресурсу: <https://www.icann.org/resources/pages/dnssec-what-is-it-why-important-2019-03-05-en>.

		Лещенко Б.С.				Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		92

ДОДАТКИ

		Лещенко Б.С.				Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		93

Топологія мережі для дослідження можливостей Cisco Packet Tracer

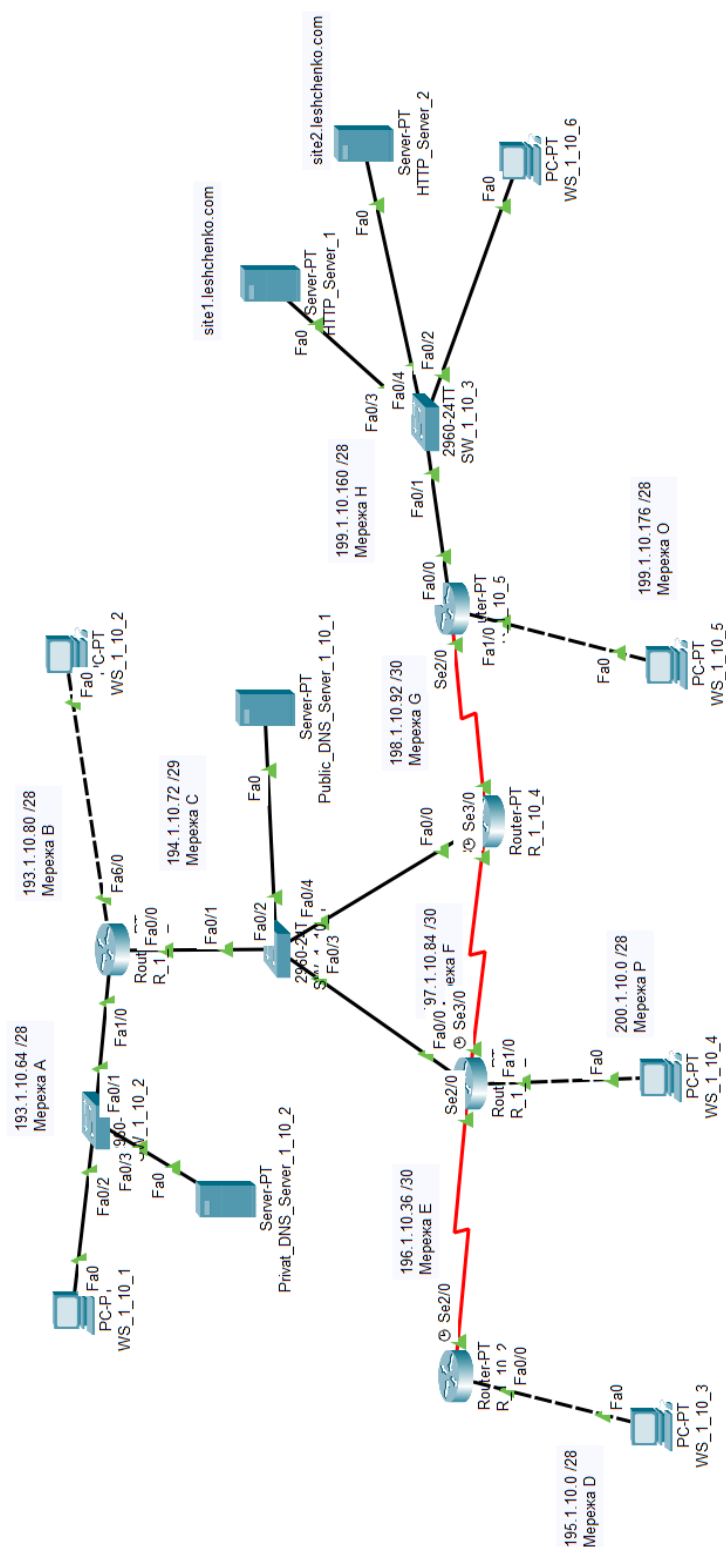


Рис. А1 – Топологія мережі для дослідження можливостей Cisco PacketTracer

		Лещенко Б.С.				Арк.
						94
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця Б.1 – Параметри адресації та підключення пристроїв мережі

Пристрій	Інтерфейс	Підключення до пристрою	Підключення до інтерфейсу
Маршрутизатор R_1_10_1	Fa0/0	SW_1_10_1	Fa0/1
	Fa1/0	SW_1_10_2	Fa0
	Gig6/0	WS_1_10_2	Fa0
Маршрутизатор R_1_10_2	Fa0/0	WS_1_10_3	Fa0
	Se2/0	R_1_10_3	Se2/0
Маршрутизатор R_1_10_3	Se2/0	R_1_10_2	Se2/0
	Fa0/0	WS_1_10_4	Fa0
	Fa1/0	SW_1_10_1	Fa0/3
	Se3/0	R_1_10_4	Se2/0
Маршрутизатор R_1_10_4	Fa0/0	SW_1_10_1	Fa0/4
	Se2/0	R_1_10_3	Se3/0
	Se3/0	R_1_10_5	Se2/0
Маршрутизатор R_1_10_5	Se2/0	R_1_10_4	Se3/0
	Fa1/0	WS_1_10_5	Fa0
	Fa0/0	WS_1_10_6	Fa0
Комутатор SW_1_10_1	Fa0/1	R_1_10_1	Fa0/1
	Fa0/2	DNS_Server_1_10_1	Fa0
	Fa0/3	R_1_10_3	Fa0/1
	Fa0/4	R_1_10_4	Fa0/0
Комутатор SW_1_10_2	Fa0/1	R_1_10_1	Fa1/0
	Fa0/2	WS_1_10_1	Fa0
	Fa0/3	DNS_Server_1_10_2	Fa0
Комутатор SW_1_10_3	Fa0/1	R_1_10_5	Fa1/0
	Fa0/2	WS_1_10_2	Fa0
	Fa0/3	HTTP_Server_1_10_1	Fa0
WS_1_10_1	Fa0	SW_1_10_2	Fa0/1
WS_1_10_2	Fa0	R_1_10_1	Gig6/0
WS_1_10_3	Fa0	R_1_10_2	Fa0/0
WS_1_10_4	Fa0	R_1_10_3	Fa0/0
WS_1_10_5	Fa0	R_1_10_5	Fa1/0
WS_1_10_6	Fa0	R_1_10_5	Fa0/0
Public_DNS_Server_1_10_1	Fa0	SW_1_10_1	Fa0/2
Privat_DNS_Server_1_10_2	Fa0	SW_1_10_2	Fa0/3
HTTP_Server_1_10_1	Fa0	SW_1_10_3	Fa0/3
HTTP_Server_1_10_2	Fa0	SW_1_10_3	Fa0/4

Таблиця В.1 – Параметри адресації мережі

Підмережа/Пристрій	Інтерфейс/Мережний адаптер/Шлюз	ІР-адреса	Маска підмережі	Префікс
Підмережа А	–	193.1.10.64	255.255.255.240	/28
Підмережа В	–	193.1.10.80	255.255.255.240	/28
Підмережа С	–	194.1.10.72	255.255.255.248	/29
Підмережа D	–	195.1.10.0	255.255.255.240	/28
Підмережа Е	–	196.1.10.36	255.255.255.252	/30
Підмережа F	–	197.1.10.84	255.255.255.252	/30
Підмережа G	–	198.1.10.92	255.255.255.252	/30
Підмережа H	–	199.1.10.160	255.255.255.240	/28
Підмережа О	–	199.1.10.176	255.255.255.240	/28
Підмережа Р	–	200.1.10.0	255.255.255.240	/28
Маршрутизатор R_1_10_1	Fa0/0	194.1.10.73	255.255.255.248	/29
	Fa1/0	193.1.10.65	255.255.255.240	/28
	Fa6/0	193.1.10.81	255.255.255.240	/28
Маршрутизатор R_1_10_2	Fa0/0	195.1.10.1	255.255.255.240	/28
	Se2/0	196.1.10.37	255.255.255.252	/30
Маршрутизатор R_1_10_3	Se2/0	196.1.10.38	255.255.255.252	/30
	Fa0/0	194.1.10.74	255.255.255.248	/29
	Fa1/0	200.1.10.1	255.255.255.240	/28
	Se3/0	197.1.10.85	255.255.255.252	/30
Маршрутизатор R_1_10_4	Fa0/0	194.1.10.75	255.255.255.248	/29
	Se2/0	197.1.10.86	255.255.255.252	/30
	Se3/0	198.1.10.93	255.255.255.252	/30
Маршрутизатор R_1_10_5	Se2/0	198.1.10.94	255.255.255.252	/30
	Fa1/0	199.1.10.177	255.255.255.240	/28
	Fa0/0	199.1.10.161	255.255.255.240	/28
Робоча станція WS_1_10_1	Мережний адаптер	193.1.10.66	255.255.255.240	/28
	Шлюз за замовчуванням	193.1.10.65	–	–
Робоча станція WS_1_10_2	Мережний адаптер	193.1.10.82	255.255.255.240	/28
	Шлюз за замовчуванням	193.1.10.81	–	–
Робоча станція WS_1_10_3	Мережний адаптер	195.1.10.2	255.255.255.240	/28
	Шлюз за замовчуванням	195.1.10.1	–	–
Робоча станція WS_1_10_4	Мережний адаптер	200.1.10.2	255.255.255.240	/28
	Шлюз за замовчуванням	200.1.10.1	–	–
Робоча станція WS_1_10_5	Мережний адаптер	199.1.10.178	255.255.255.240	/28
	Шлюз за замовчуванням	199.1.10.177	–	–
Робоча станція WS_1_10_6	Мережний адаптер	199.1.10.162	255.255.255.240	/28
	Шлюз за замовчуванням	199.1.10.161	–	–
Public_DNS_Server_1_10_1	Мережний адаптер	194.1.10.76	255.255.255.248	/29
	Шлюз за замовчуванням	194.1.10.73	–	–
Public_DNS_Server_1_10_1	Мережний адаптер	193.1.10.67	255.255.255.240	/28
	Шлюз за замовчуванням	193.1.10.65	–	–
HTTP_Server_1_10_1	Мережний адаптер	199.1.10.163	255.255.255.240	/28
	Шлюз за замовчуванням	199.1.10.161	–	–
HTTP_Server_1_10_2	Мережний адаптер	199.1.10.164	255.255.255.240	/28

		Шлюз за замовчуванням	199.1.10.161	–	–
--	--	-----------------------	--------------	---	---

Додаток Г

Топологія мережі побудована в GNS3

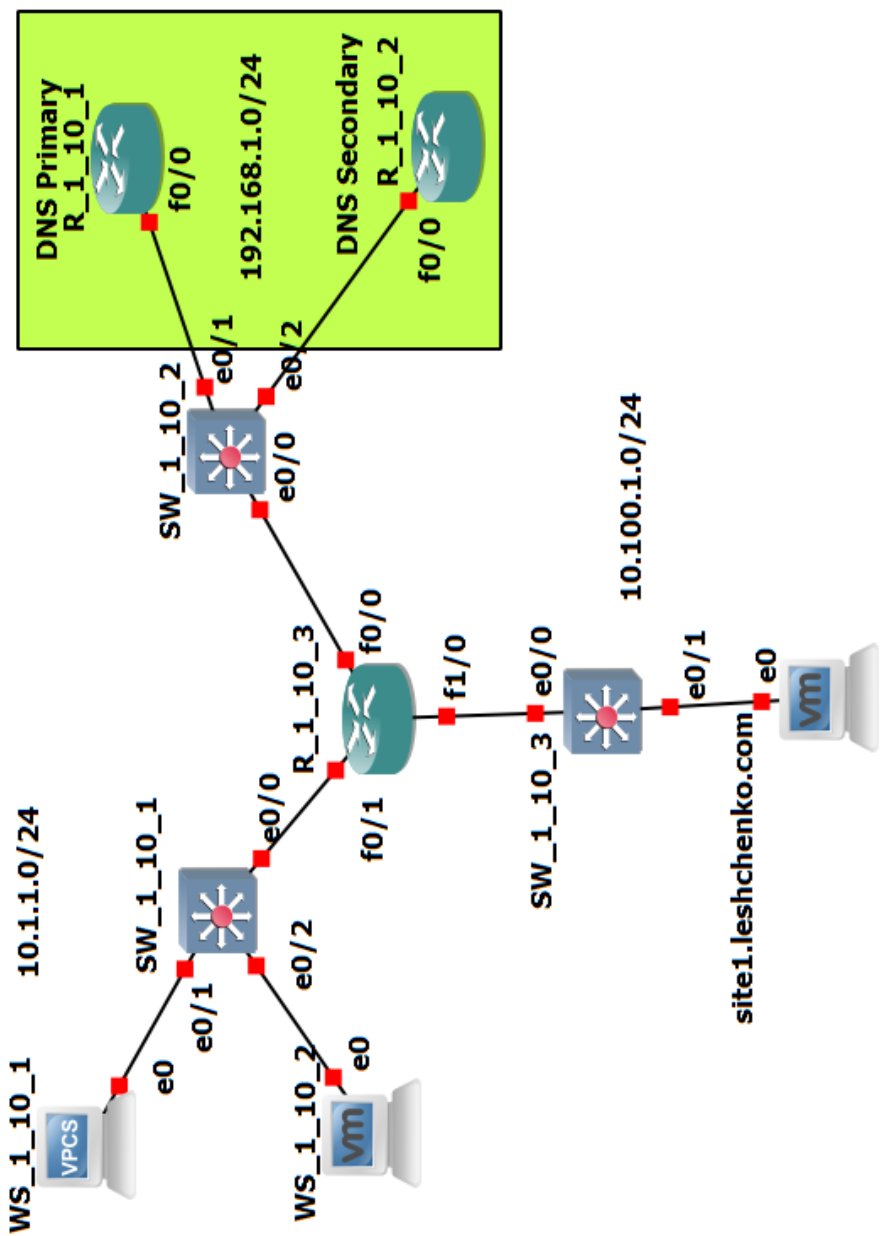


Рис.Г.1 – Топологія мережі

Таблиця Д.1 – Параметри адресації та підключення пристроїв мережі

Пристрій	Інтерфейс	Підключення до пристрою	Підключення до інтерфейсу
Маршрутизатор R_1_10_1	Fa0/0	SW_1_10_2	E0/1
Маршрутизатор R_1_10_2	Fa0/0	SW_1_10_2	E0/2
Маршрутизатор R_1_10_3	Fa0/0	SW_1_10_2	E0/0
	Fa1/0	SW_1_10_3	E0/0
	Fa0/1	SW_1_10_1	E0/0
Комутатор SW_1_10_1	E0/0	R_1_10_3	Fa0/1
	E0/1	WS_1_10_1	E0
	E0/2	WS_1_10_2	E0
Комутатор SW_1_10_2	E0/0	R_1_10_3	Fa1/0
	E0/1	R_1_10_1	Fa0/0
	E0/2	R_1_10_2	Fa0/0
Комутатор SW_1_10_3	E0/0	R_1_10_3	Fa1/0
	E0/1	Site1.leshchenko.com	E0
WS_1_10_1	E0	SW_1_10_1	E0/1
WS_1_10_2	E0	SW_1_10_1	E0/2
HTTP_Server_1_10_2	E0	SW_1_10_3	E0/1

Таблиця Д.2 – Параметри адресації та підключення пристроїв мережі

Підмережа/Пристрій	Інтерфейс/Мережний адаптер/Шлюз	IP-адреса	Маска підмережі	Префікс
Маршрутизатор R_1_10_1	Fa0/0	192.168.1.5	255.255.255.0	/24
Маршрутизатор R_1_10_2	Fa0/0	192.168.1.6	255.255.255.0	/24
Маршрутизатор R_1_10_3	Fa0/0	192.168.1.1	255.255.255.0	/24
	Fa1/0	10.1.1.1	255.255.255.0	/24
	Fa0/1	10.100.1.1	255.255.255.0	/24
Робоча станція WS_1_10_1	Мережний адаптер	10.1.1.5	255.255.255.0	/24
	Шлюз за замовчуванням	10.1.1.1	–	–
Робоча станція WS_1_10_2	Мережний адаптер	10.1.1.6	255.255.255.0	/24
	Шлюз за замовчуванням	10.1.1.1	–	–
Site1.leshchenko.com	Мережний адаптер	10.100.1.5	255.255.255.0	/24
	Шлюз за замовчуванням	10.100.1.1	–	–