КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 2 (10), 2020
ISSN 2663 - 4023
DOI 10.28925/2663-4023.2020.10.7587
УДК 004.03

Ільєнко Анна Вадимівна
Кандидат технічних наук, доцент, доцент кафедри комп'ютеризованих систем захисту інформації
Національний авіаційний університет університет,
факультет кібербезпеки комп'ютерної та програмної інженерії, Київ, Україна
ORCID: 0000-0001-8565-1117
Ilyenko.a.v@nau.edu.ua

Ільєнко Сергій Сергійович
Кандидат технічних наук, доцент, доцент кафедри автоматизації та енергоменеджменту
Національний авіаційний університет університет,
аерокосмічний факультет, Київ, Україна
ORCID: 0000-0002-0437-0995
Ilyenko.s.s@nau.edu.ua

Вертиполох Олександр Олександрович
Магістр, студент кафедри комп'ютеризованих систем захисту інформації
Національний авіаційний університет університет,
факультет кібербезпеки комп'ютерної та програмної інженерії, Київ, Україна
ORCID: 0000-0001-6228-8394
Vertipolokh@gmail.com

МЕТОД ЗАХИСТУ ТРАФІКУ ВІД ВТРУЧАННЯ DPI СИСТЕМ НА БАЗІ ВИКОРИСТАННЯ DOH ТА DOT ПРОТОКОЛІВ

Анотація. Дана стаття присвячена розгляду подальших шляхів забезпечення захисту трафіку від втручання DPI систем. У статті досліджено можливості використання мережевих протоколів та застосування DPI систем. Проведений аналіз проблеми дав змогу визначити вразливі місцями протоколу DNS, який базується на протоколі UDP. Цими вразливими місцями є - спуфінг, перехоплення та переприв'язування трафіку. Також на підставі проведеного аналізу методів захисту DNS трафіку від втручання, авторами обґрунтовано та визначено наступне: 1) усі DNS запити передаються у відкритому вигляді; 2) існуючі підходи забезпечення захисту трафіку не використовують шифрування та, в наслідок чого, не забезпечують конфіденційність інформації; 3) відбувається лише підтвердження автентичності записів. Авторами було сформовано зведену таблицю, в якій визначено надійні методи захисту DNS трафіку. Автори пропонують розробку повноцінного локального проксуючого серверу для забезпечення DNS трафіку, який може звертатися до довірених публічних DNS резолверів за допомогою протоколів doh та dot. Для розуміння принципів взаємодії протоколів було розгорнуто власну локальну реалізацію основних компонентів мережі, з якими найчастіше мають справу користувачі мережі, а саме: 1) веб сервер; 2) DNS сервер; 3) сервер забезпечення криптографічного захисту та приховування відкритих запитів. Практична цінність отриманих результатів полягає у програмній реалізації методів захисту трафіку від втручання DPI систем у середовищі Visual Studio Code за рахунок використання мови програмування Python 3.8, що дає змогу забезпечити криптографічний захист трафіку. Запропоноване рішення локального проксуючого серверу може удосконалюватись в майбутньому за рахунок впровадження локального кешування з додаванням можливості створення правил для певних доменів та їх під доменів. Реалізований тестовий doh сервер може бути розгорнуто на довіреному виділеному сервері за межами можливих точок встановлення фільтруючого обладнання. Така реалізація дасть змогу повністю контролювати власний трафік для резолвінгу доменних імен. Авторами в подальшому планується ряд науково технічних рішень розробки та впровадження ефективних методів, засобів забезпечення вимог, принципів та підходів забезпечення кібернетичної безпеки та організації захисту трафіку від втручання DPI систем в дослідних комп'ютерних системах та мережах.

Ключові слова: інтернет протокол, модель TCP/IP, TLS, DNS, HTTPS, DNS-over-HTTPS,

DNS-over-TLS, DPI.

1. ВСТУП

Постановка проблеми. Питання забезпечення інформаційної безпеки та конфіденційності користувачів в Інтернеті дуже багатогранне і складне. З одного боку глобальна мережа стає все більш захищеною: провідні IT компанії світу розробляють та активно впроваджують нові мережеві стандарти та протоколи, що робить Інтернет більш безпечним, швидким та ефективним як для звичайних середньостатистичних користувачів, так і для корпоративних клієнтів. Перші отримують змогу в будь-який момент часу швидко отримати доступ до необхідної інформації чи ресурсів для задоволення власних потреб. Корпорації ж, в свою чергу, завдяки різного роду статистичних даних мають змогу більш чітко та ефективно планувати свої бізнес-процеси, корелюючи їх з потребами клієнтів.

З іншого боку, така глобалізація і структуризація інформаційних процесів має і негативні наслідки. На основі даних, які збираються IT корпораціями та провайдерами можна повністю скомпрометувати не тільки окремо взяту людину, а й великі підприємства. Тому проблема захисту персональних даних в Інтернеті загострюється з кожним роком, адже чим більш доступною і розповсюдженою стає глобальна мережа, тим менше шансів на збереження анонімності лишається у середньостатистичних користувачів, а конфіденційні данні різного роду компаній та організацій стають більш вразливими до несанкціонованого доступу до них, порушення їх цілісності та доступності. Системи DPI все частіше використовуються для контролю і фільтрації трафіку, а також для блокування протоколів.

Аналіз останніх досліджень і публікацій. Як і у книзі [1], у даній статті розглядаються та аналізуються загальні принципи застосування DPI систем з метою контролю та фільтрації трафіку в сучасних комп'ютерних системах та мережах. Аналіз звітів і публікацій про виявлені вразливості сучасних мережевих протоколів, а особливо протоколу DNS дають зрозуміти подальші напрями удосконалення забезпечення безпеки інформаційної системи [4,5,6]. В публікаціях [2-8] представлені напрями вирішення проблем мережевого протоколу DNS з використанням процедур шифрування, але дані підходи в повній мірі не забезпечують конфіденційність інформації при її передачі по відкритих комп'ютерних системах та мережах. Це призводить до перехоплення та спуфінгу трафіку. Авторами в даній роботі будуть більш детально проаналізовані та систематизовані методи забезпечення конфіденційності інформації при її передачі в відкритому вигляді в сучасних комп'ютерних системах та мережах.

Враховуючи аналіз останніх досліджень і публікацій, актуальним є питання захисту мережевого трафіку від втручання DPI систем, принцип дії якого ґрунтується на дослідженні вразливих місць стандартів, а саме DNS-запитів з врахуванням особливостей функціонування типових мережевих протоколів, з якими має справу переважна більшість користувачів мережі Інтернет.

Мета статті. Метою даної статті є дослідження теоретичних основ застосування DPI систем, окреслення основних підходів щодо забезпечення захисту трафіку від втручання DPI систем та визначення нового методу до забезпечення захисту трафіку. Такий підхід направлений на унеможливлення або значного ускладнення можливості використання DPI систем до DNS трафіку. Кінцевим результатом роботи планується представлення власної реалізації захисту трафіку за допомогою мови програмування

Python 3.8 з подальшою можливістю її інтегрування в комп'ютерні системи та мережі. Для досягнення поставленої мети потрібно розв'язати основні задачі: провести дослідження можливостей використання мережевих протоколів та застосування DPI систем; провести дослідження методів захисту трафіку від втручання DPI систем; запропонувати власну реалізацію методу захисту трафіку від втручання DPI систем на базі використання doh та dot та локального проксуючого DNS серверу.

2. ТЕОРЕТИЧНІ ОСНОВИ ДОСЛІДЖЕННЯ

Під поняттям DPI система будемо розуміти таку систему, яка виконує, так званий, глибокий аналіз мережевих пакетів на верхніх рівнях моделі OSI. Традиційний аналіз пакетів зазвичай перевіряє інформацію в заголовках пакетів мережевого та транспортного рівнів, DPI спрямований також на поведінковий аналіз трафіку прикладного рівня у режимі реального часу, тобто такий, що дозволяє розпізнавати користувацькі програми, для яких заздалегідь не визначено відомі заголовки протоколів та структури даних [1-5].

Виділяють два виду підключення DPI:

1. Пасивний. Система, що підключена паралельно до мережі провайдера за допомогою спліттера (розділювача сигналу) або з використанням дзеркалювання трафіку (див. Рис. 1). Так як DPI зазвичай працює у режимі реального часу, то такий тип підключення не буде вузьким місцем при великому обсязі трафіку. Але такий тип підключення не має можливості одразу припинити спробу отримання доступу до заборонених ресурсів, але тільки виявити її. Хоча є можливість підроблення відповіді від сайту провайдером за допомогою підміни IP-адреси відправника і структури TCP пакету.

2. Активний. Система, що підключена до мережі провайдера напряму, як і інші мережеві пристрої (див. Рис. 2).

Провайдер або адміністратор системи налаштовує DPI систему, ніби файєрвол, на перевірку різних типів трафіку (вхідного, вихідного або обох одночасно), а система на базі різних правил приймає рішення про блокування трафіку або його пропуск.

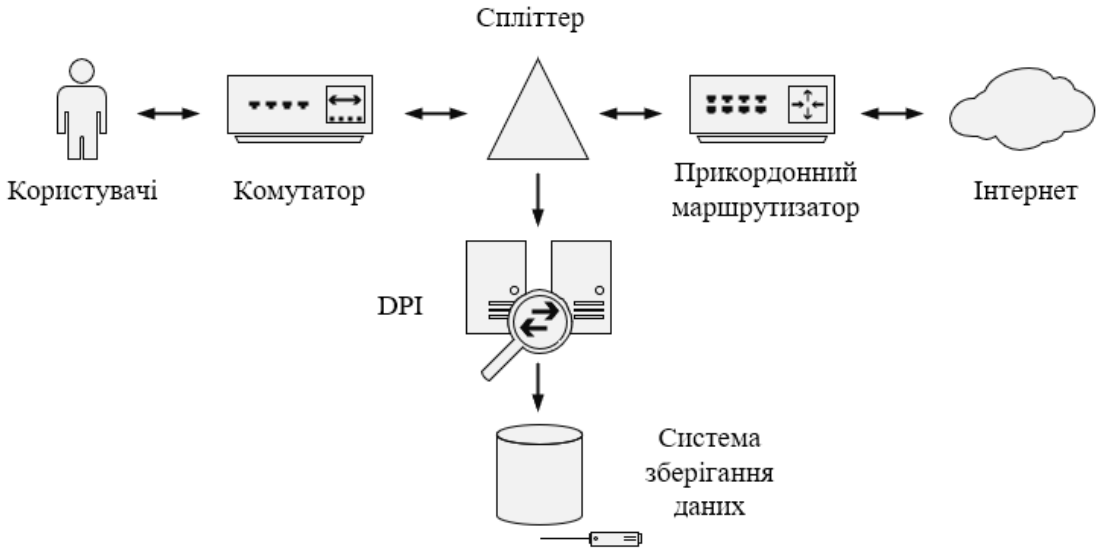

Рис. 1. Можлива схема підключення DPI в пасивному режимі

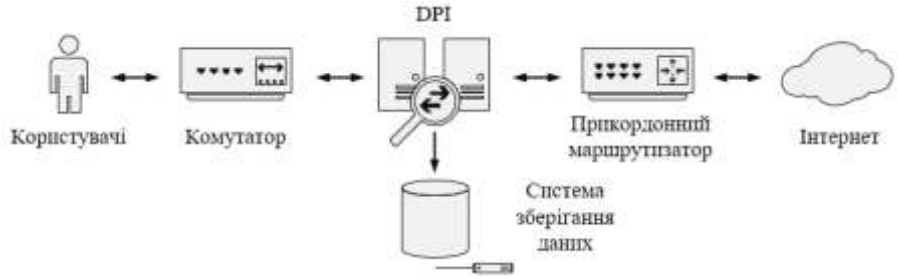

Рис. 2. Можлива схема підключення DPI в активному режимі

Напрями використання DPI систем

DPI було розроблено для використання у наступних ситуаціях:

1. Блокування мережевих ресурсів. Один з найголовніших напрямків використання цих систем з метою недопущення користувачів до користування певними ресурсами та найчастіше використовується у комплексних системах разом із системами зберігання усього трафіку.

2. Оптимізація та пріоритизація. Покращення якості обслуговування для кінцевих абонентів провайдер може досягти або шляхом збільшення власної ширини каналу, або оптимізувавши трафік, наприклад, знизивши чи збільшивши пріоритет певних протоколів під час підвищеного навантаження на мережу, яке зазвичай була прогнозоване та прив'язане до часових рамок.

3. Побудова поведінкової моделі абонента. Так як усі люди унікальні, то їх поведінка як у житті, так і у мережі відрізняється. Провайдер може збирати інформацію, коли і хто виходить в онлайн, якими додатками користується та які сайти відвідує, що дозволяє створити деякий цифровий відбиток користувача і використовувати його для оптимізації в свої цілях або продавати третім особам.

4. Захист. Так як DPI системи можуть аналізувати протоколи та сигнатури трафіку або виявляти його аномалії, це дозволяє будувати рішення, які запобігають атакам типу ddos, а також використовуватися у IPS (Intrusion Prevention System – система попередження вторгнень) для виявлення атак у реальному часі.

4. Кешування. Для більш оптимального використання інтернет трафіку і збільшення швидкості доступу до часто використовуємо інформації (контенту популярних сайтів, оновлень чи цікавих файлів) налаштовують кеш сервери.

Обладнання для DPI систем

Зазвичай обладнання для DPI систем схоже на звичайні сервери розміром 1U для монтування, де головну роль в апаратних характеристиках відіграє мережева складова, наприклад, наявність спеціального режиму для мережевих інтерфейсів, що дозволяє з'єднувати їх на найнижчому рівні OSI – фізичному та у разі проблем з системою продовжувати пропускати трафік без його повного зупинення, та швидкість оброблення (особливо важливо для активного підключення) з можливістю розпаралелювання задач. А для організації кешування чи зберігання статистики користування або взагалі усього трафіку до системи може бути підключена зовнішня система зберігання даних.

Найбільшими постачальниками апаратної реалізації DPI є Cisco, Sandvine, Procera Networks та Allot Communications, а також стрімко набирають популярність інтегровані можливості в маршрутизатори, виробниками та постачальниками такої продукції є Cisco, Ubiquiti та Juniper. Окрім апаратних рішень існують також і програмні, які надаються на безкоштовній та платних основах. Найбільш популярними та розвинутими проектами, які розповсюджуються на безоплатній основі є: - ndpi –

продовження розвитку вже непідтримуваної бібліотеки opendpi компанією ntop, є проектом з відкритим вихідним кодом та розповсюджується за ліцензією lgplv3. Підтримує більше ніж 200 протоколів та дозволяє визначати протоколи додатків прикладного рівня, аналізуючи характер мережевої активності без прив'язки до певних портів, а також дозволяє переглядати зашифрований трафік за допомогою MITM; - joy – програмний продукт компанії Cisco, який ліцензується під вільною ліцензією BSD та заснований на пакеті libpcap з можливістю захоплення трафіку у режимі реального часу, використовуючи потокоорієнтовану модель та надаючи результати у зручному форматі JSON.

При використанні DPI можуть виникнути такі проблеми:

– Вразливість до ddos атак: хоча DPI і допомогає протистояти атакам такого типу, але при неправильній конфігурації злочинці, навпаки, можуть використовувати особливості системи для створення подібних атак;

– Підвищення складності управління мережею: в додаток до налаштувань межмережевого екрану потрібно правильно налаштувати DPI, а також проводити роботи з політиками для підвищення ефективності;

– Необхідність прослуховувати трафік у обох напрямках: для детектування трафіку, зазвичай, заснованого на евристичній моделі, DPI система повинна мати доступ до обох типів трафіку для найбільш точного детектування.

Серед блокувань, що використовують DPI системи при перехопленні та фільтрації трафіку можна виділити: підміна DNS-відповідей; перенаправлення звернень до сторонніх DNS-серверів на сервери, що належать провайдеру; блокування за IP-адресою; блокування всіх піддоменів заблокованого домену; фільтрація URL на окремих або всіх IP-адресах і/або портах; підміна SSL сертифікату для прослуховування HTTPS-трафіку [10-14].

Використання особливостей протоколів для обходу DPI

Використовуючи особливості протоколів можна виділити такі способи обходу DPI блокувань: додавання пробілів або інших символів табуляції між методом HTTP (GET, POST тощо) та URI; змішування літер регістру значення заголовка хоста; видалення пробілу між назвою заголовка та значенням у заголовку хосту; фрагментація на рівня TCP для першого пакету даних; фрагментація на рівні TCP для постійних сеансів HTTP; надсилання підроблених пакетів HTTP з низьким значенням часу життя або неправильною контрольною сумою. Таким чином в даному розділі досліджено теоретичні можливості використання DPI систем до прослуховування та перехопленння DNS трафіку. Далі розгялнемо проблеми та можливі підходи щодо забезпечення захисту DNS трафіку.

3. ПРОБЛЕМИ DNS ТА СПОСОБИ ЇХ ВИРІШЕННЯ

Сучасний веб та деякі інші мережеві протоколи захищені за допомогою TLS, але DNS-запити всю свою історію передають в незашифрованому вигляді. Компанії або держави використовують це в своїх інтересах, наприклад, для збору інформації про відвідувані сайти або фільтрації трафіку або навіть проводити атаки на DNS трафік, так званий спуфінг запитів з метою перенаправлення їх на власний сервер. Навіть використання захищених VPN тунелів не дає стовідсоткову гарантію захисту, так як в рамках цієї сесії можуть використовуватися запити DNS, які можуть відправлятися за межами тунелю і призводити до витоку DNS інформації.

DNS проектувався для використання розподіленими високонавантаженими мережами доставки контенту та не забезпечує на сьогодні гідного рівня захисту трафіку

користувачів. Це призвело до появи деяких векторів атак:

- спуфінг DNS запитів: підміна справжньої IP-адреси в кеші серверу шкідливою з метою встановлення з'єднання з сервером зловмисника;

- перехоплення DNS запитів: перенаправлення на шкідливі сайти або з метою збору статистики та показу реклами;

- переприв'язування DNS: встановлення контролю над сервером, що обслуговує шкідливий домен, з подальшим завантаженням та виконанням скриптів в додатку користувача при відвідування інших сервісів.

В рамках розширення функцій DNS було запропоновано використання DNSSEC – протоколу цифрового підпису DNS записів. При використанні цього підходу до певних типів записів додається підпис, який дозволяє рекурсивним резолверам встановити, чи дійсно саме власник домену створив цей запис. Дане розширення не використовує шифрування, лише підтверджуючи автентичність запису, а усі DNS запити передаються у відкритому вигляді, як і раніше [2-6]. Таким чином, подібні розширення жодним чином не використовують шифрування даних, і, відповідно не вирішують проблему конфіденційності інформації, що передаються під час виконання DNS запитів, чим і користуються, наприклад, виробники DPI обладнання, провайдери, зловмисники і державні органи, так як подібні логи з серверів дозволяють створити деякий цифровий відбиток користувача.

Тому з метою вирішення проблеми шифрування DNS пропонуються такі протоколи [3,4,5]: dnscrypt; DNS-over-TLS (dot); DNS-over-HTTPS (doh); DNS-over-SSH (dos); DNS-over-QUIC (doq). На базі досліджених протоколів є можливість представлення зведеної таблиці (див. Табл. 1).

Dnscrypt є найдавнішою спробою імплементації захисту DNS запитів. Він автентифікує зв'язок між DNS клієнтом та DNS резолвером, запобігає деяким видам атак та використовує криптографічні підписи для перевірки відповіді. Нажаль, хоча і існує досить велика кількість серверів, що реалізують підтримку цього протоколу, але підтримка для кінцевих пристроїв так і не отримала широкого розповсюдження, для мобільних клієнтів вимагається встановлення додаткового програмного забезпечення, а для мобільних – права суперкористувача. В той же час найсучаснішим протоколом є DNS-over-QUIC, але він, разом із іншими протоколами, на які він покладається, знаходяться тільки на початковій стадії, а їх специфікацій часто оновлюються, щоб можна було доцільно імплементувати підтримку для кінцевих пристроїв. DNS-over-SSH базується на одному з найчастіше використовуваних протоколів прикладного рівня – SSH (Secure Shell – безпечна оболонка), який за вмовчуванням використовує шифрування без потреби використання TLS, але вимагає TCP. Реалізація DNS, що працює на базі SSH призводить до створення подвійного TCP, тобто TCP over TCP, що призводить до значного зниження швидкодії, а також складності синхронізації та гарантії доставки.

Таблиця 1

Порівняльна характеристика методів захисту DNS

Характеристика	**DNS Crypt**	**Dot**	**Doh**	**Dos**	**Doq**
Стандарт RFC	Ні	Так	Так	Ні	Так
Використання специфічного порту	Ні	Так	Ні	Ні	Так
Потреба стеку TCP+TLS	Част.	Так	Так	Част.	Ні
Підтримка паралелізму та пріоритизації	Так	Так	Так	Ні	Так
Велика кількість підтримуваних клієнтів	Ні	Так	Так	Ні	Ні
Висока ймовірність виявлення DPI системою	Так	Част.	Ні	Ні	Част.

ISSN 2663 - 4023

Відповідно до проведеного аналізу авторами пропонується використання поєднаних методів dot та doh, які дозволяють одразу, без впровадження нових протоколів, із забезпеченням зворотної сумісності реалізувати захищену передачу трафіку. Перший метод більшого розповсюдження набуває на мобільних пристроях, наприклад, входить до реалізації Android 9. У той же час другий метод більшого поширення набув в системах, що вже реалізують використання HTTPS, наприклад, браузери. В даному розділі було розглянуто найбільш значущі аспекти роботи мережі, основні поняття та принципи її побудови, виявлено до сих пір вразливі місця певних протоколів, а саме, DNS, який базується на протоколі UDP та не покладається на використання криптографічних алгоритмів. Цим, користуються надавачі послуг, держави і зловмисники у власних цілях, використовуючи деякі схеми атак, а також обладнання DPI. Таким чином, далі буде показано удосконалений підхід щодо приховування даних, що до сих пір передаються у відкритому вигляді в мережі Інтернет, на базі впровадження криптографічних методів.

4. ПОДАЛЬШІ НАПРЯМИ ДОСЛІДЖЕННЯ

На рис. 3 зображено класичний процес проходження трафіку від додатку до кінцевого серверу імен, після чого відповідь направляється у зворотному напрямку, і додаток може встановити з'єднання з потрібним сервером.

Як можна бачити, цей процес є рекурсивним: спочатку запит потрапляє на локальний DNS резолвер, що може обслуговуватися на робочій станції, далі, так як, зазвичай робочі станції знаходяться за NAT, тобто за маршрутизатором, то останній відповідає за перенаправлення запиту, а також може виступати у ролі локального кешуючого DNS серверу при відповідних налаштуваннях, далі маршрутизатор перенаправляє запит на деякий публічний сервер, наприклад, Інтернет провайдера або іншої компанії, що обслуговує відповідні рішення. Якщо сервер не може знайти у себе потрібну відповідь, запит буде передано наступному серверу, аж до кореневого. Як було зазначено в першому розділі, DPI обладнання зазвичай встановлюється, відповідно до поточної схеми, на третьому кроці, і, відповідно до оригінальної специфікації DNS – увесь трафік на шляху свого проходження передається у відкритому вигляді.

Відповідно до поставленої мети, автори вважають доцільним розробку такого методу доставки DNS запитів до довірених резолверів, який би забезпечив конфіденційність даних, що передаються та унеможливив або значно ускладнив виявлення цього типу трафіку DPI системами.

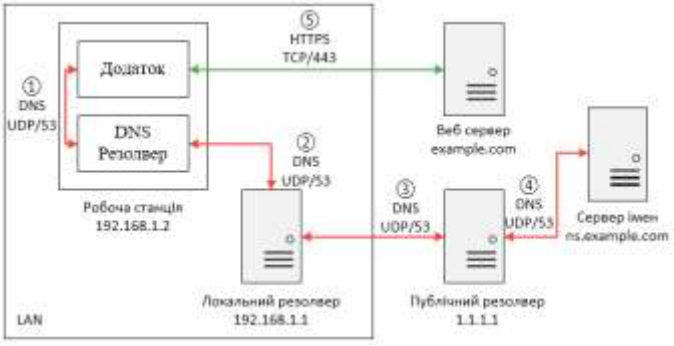

Рис. 3. Шлях DNS запиту

ISSN 2663 - 4023

Автори пропонують розробку повноцінного локального проксуючого серверу, який може звертатися до довірених публічних DNS резолверів за допомогою протоколів doh та dot. Практична цінність отриманих результатів в даній статті полягає у програмній реалізації методів захисту трафіку від втручання DPI систем у середовищі Visual Studio Code за рахунок використання мови програмування Python 3.8, що дало змогу забезпечити криптографічний захист трафіку. З практичної точки зору для розуміння принципів взаємодії запропонованих протоколів автори пропонують розгорнути спрощену локальну модель, яка складається з: веб серверу, серверу, що відповідає за приховування DNS запитів і локального DNS серверу.

На рис. 4 відображено шлях проходження трафіку при реалізації запопонованого авторами програмного модуля локального проксуючого серверу. Запропонована авторами архітектура, представлена у вигляді власного локального проксуючого DNS серверу, який може відправляти DNS трафік хоста до довіреного публічного резолвера за допомогою формування відповідних запитів та отримання відповідей з підтримкою doh та dot. Таким чином дана реалізація може бути використана для забезпечення криптографічного захисту трафіку, що зазвичай передається у відкритому вигляді, між локальним пристроєм та довіреним публічним резолвером.

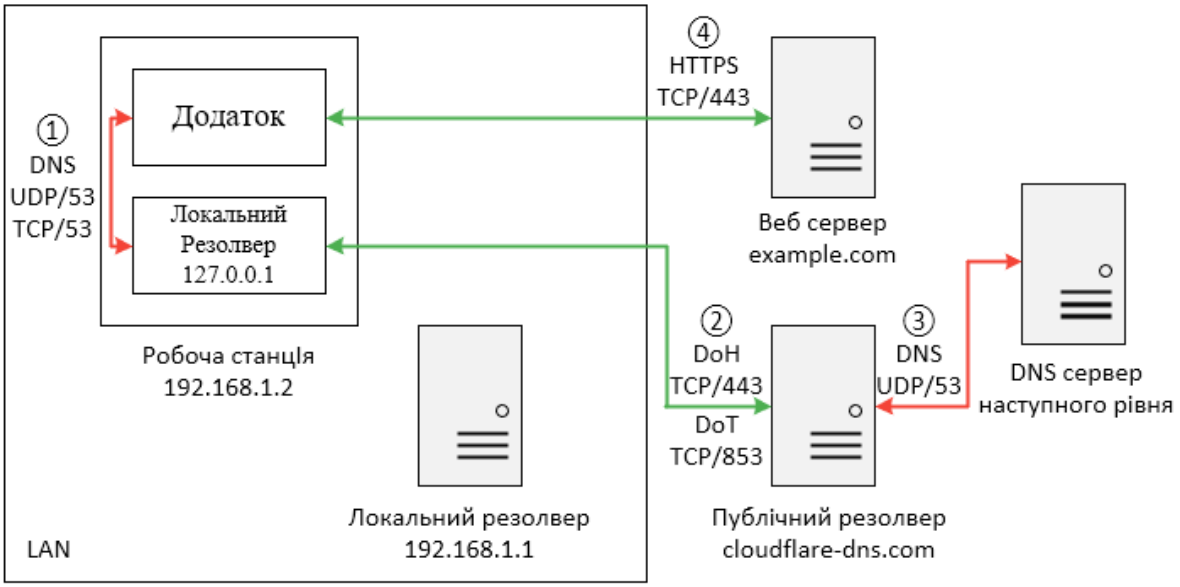

Рис. 4. Архітектура запропонованого рішення

Як приклад наводимо елементи власної реалізації методу захисту трафіку від втручання DPI систем на базі використання doh та dot та локального проксуючого DNS серверу. Приклад реалізовано на Python 3.8:

```
@staticmethod
Def doh_resolver(dns_message, sender):
    Ip, port, domain = parse_sender_dict(sender)
    Doh = proto_senders.dohsender(ip, port, domain)
    Return doh.query(dns_message)
@staticmethod
Def dot_resolver(dns_message, sender):
    Ip, port, domain = parse_sender_dict(sender)
    Dot = proto_senders.dotsender(ip, port, domain)
    Return dot.query(dns_message)
```

ISSN 2663 - 4023

```
@staticmethod
Def resolve_sender_ip(sender_address, bootstrap_ip, hosts_dict):
    If sender_address in hosts_dict:
        Return hosts_dict[sender_address]
    Dns_query = dns.message.make_query(sender_address, dns.rdatatype.A)
    Response = dns.query.udp(dns_query, bootstrap_ip)
    If response.answer:
        Return response.answer[-1].items[0].to_text()
```

Для проведення тестування коректності авторської розробки використовуються програмний інтерфейс на мові Python, утиліта curl або браузер, а для емуляції DPI використовується програма Wireshark для перегляду інтернет пакетів. Основним елементом імплементації локальної архітектури, що відображає можливість захисту DNS трафіку є впровадження серверу, який реалізує шифрування цього типу трафіку (doh та dot сервер) та виступає у ролі проксі серверу між додатком, який підтримує генерування запиту у потрібному форматі з одного боку, а з іншого – безпосередньо DNS сервером.

Авторська програма запускається за допомогою виклику файлу main.py через інтерпретатор з вказівкою параметром типу захищеного підключення до публічного резолвера (рис.5).

```
λ python main.py doh
Запити будуть оброблюватися за допомогою DOH
DNS/UDP слухається на 127.0.0.1:53
DNS/TCP слухається на 127.0.0.1:53
```

Рис. 5. Запуск локального серверу

Так як захищене з'єднання передбачає використання доменних імен, то для першого запиту резолвінгу адреси серверу, куди будуть надходити захищені запити, використовується, так звана, bootstrap адреса, тобто адреса також DNS серверу, але у вигляді IP адреси. Також програма дозволяє обирати домени, для яких не потрібно використовувати захищене з'єднання, зазвичай, це буде корисно для використання серверів синхронізації часу. Firefox та Google Chrome, які можуть використовувати системну змінну SSLKEYLOGFILE, для того щоб зберігати ключі шифрування, які можна буде використати для дешифрування трафіку. Спочатку потрібно запустити файл, що містить наступну конфігурацію:

```
@echo off
Set SSLKEYLOGFILE=keylogfile.txt
Firefox.exe
```

Таким чином, усі ключі шифрування будуть зберігатися у вказаний файл до тих пір, поки відкрито вікно браузера, у цьому випадку – Firefox. Після запису трафіку, для тестування запропонованого рішення використаємо емуляцію DPI, а саме програму Wireshark, яка буде використовувати вказані ключі для дешифрування трафіку (див. Рис. 6), що дасть змогу пересвідчитися у тому, що запити направляються на сервер cloudflare-dns.com, а також мають відповідну структуру та заголовки.

ISSN 2663 - 4023

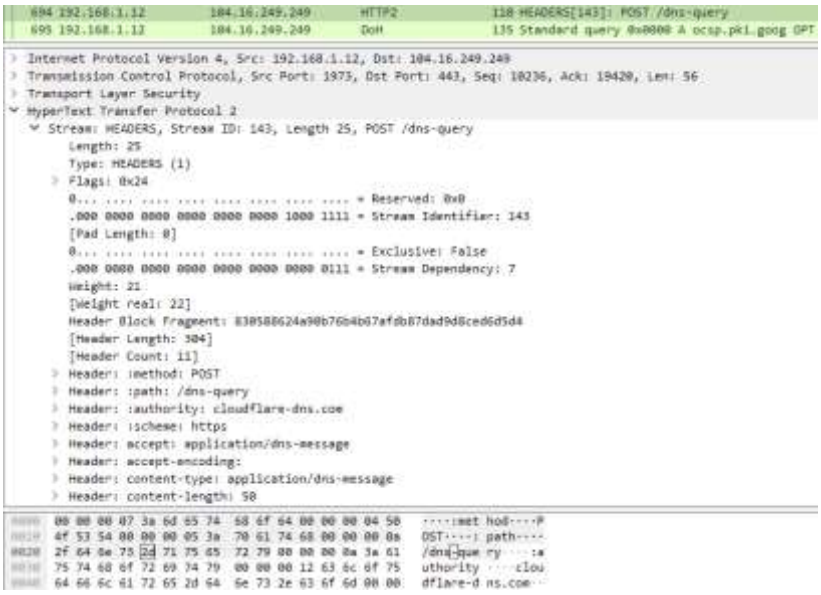

Рис. 6. Дешифрований doh трафік

Останнім тестом є перезапуск локального серверу у режим dot та захоплення трафіку програмою Wireshark (див. Рис. 7). Як можна бачити, обмін трафіком відбувається за адресою 1.1.1.1, так як на ній знаходиться сервер компанії cloudflare, що обслуговує dot з'єднання, та портом 853, що відповідає описаній специфікації протоколу.

Рис. 7. Трафік у режимі dot

Таким чином, дане тестування дозволяє стверджувати, що увесь DNS трафік буде пересилатися через захищений канал на місцях де можливо встановлення елементів DPI систем для прослуховування інформації.

5. ВИСНОВКИ

В даній статті було проведено детальньний аналіз DPI системи та їх основних компонентів, а також розглянуто способи приховування інформації від подібних систем, використовуючи особливості протоколів та нових реалізацій. Принцип дії цих протоколів ґрунтується на дослідженні до сих пір вразливих місць стандартів, а саме DNS-запитів з врахуванням особливостей функціонування типових мережевих протоколів, з якими має справу переважна більшість користувачів мережі Інтернет. На базі цих досліджень авторами досягнуто таких результатів: розглянуто та розгорнуто власну локальну реалізацію основних компонентів мережі; досліджено потік трафіку від клієнта до потрібного ресурсу; проведено аналіз взаємодії протоколів з метою

переконання в тому, що трафік між вузлами з встановленим обладнанням для фільтрації трафіку є захищеним. Також було реалізовано власний комплекс локального проксуючого серверу мовою програмування Python 3.8, та проведено його тестування на реальній системі. Цей комплекс дозволяє встановлювати захищене з'єднання з іншими довіреними серверами на базі використання протоколів doh та dot, та унеможливлює або значно ускладнює можливість використання DPI систем на межі звичайних місць їх встановлювання. Зважаючи на досягнення мети роботи практична цінність цих рішень є актуальною та необхідною для більшості користувачів та систем. Запропоноване рішення локального проксуючого серверу може бути розвинуто і далі. Наприклад, впроваджено реалізацію локального кешування або додано можливість створювати точніші правила для певних доменів та їх піддоменів, а реалізований тестовий doh сервер може бути розгорнуто на довіреному виділеному сервері за межами можливих точок встановлення фільтруючого обладнання, що дасть змогу повністю контролювати власний трафік для резолвінгу доменних імен.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

[1] Harold F., Krause M. Information Security Management Handbook, Sixth Edition. – Taylor &. Francis Group, 2007. – 3231 с.
[2] Arends R., Kosters M., Blacka D. DNS Security (DNSSEC) Opt-In, RFC 4956. – verisign, 2007. – 15 с.
[3] Dnscrypt 2 Protocol. [Електронний ресурс]. – Режим доступу: https://dnscrypt.info/protocol/
[4] Hu Z., Zhu L., Heidemann J., Mankin A., Wessels D., Hoffman P. Specification for DNS over Transport Layer Security (TLS), RFC 7858. – USC/ISI, Verisign Labs, ICANN, 2016. – 18 с.
[5] Hoffman P., mcmanus P. DNS Queries over HTTPS (doh), RFC 8484. – ICANN, Mozilla, 2018. – 21 с.
[6] Huitema C., Shore M., Mankin A., Dickinson S., Iyengar J. Specification of DNS over Dedicated QUIC Connections. – Private Octopus, Fastly, Salesforce, 2019. – 18 с.
[7] Cid C., Jacobson M.J. Selected Areas in Cryptography. 25th International Conference Calgary, 2019. – 499 с.
[8] Stallings W. Cryptography and Network Security. Principles and Practice. 7th ed. – Pearson Education Limited, 2017. – 766 с.
[9] Oppliger R. SSL and TLS Theory and Practice, Second Edition. – Artech House, London, 2016. – 280 с.
[10] Mockapetris P. Domain names - concepts and facilities, RFC 1034. – USC/Information Science Institute, 1987. – 55 с.
[11] Mockapetris P. Domain names - implementation and specification, RFC 1035. – USC/Information Science Institute, 1987. – 55 с.
[12] Pollard B. HTTP2 in Action. – Manning Publications, 2019. – 384 с.
[13] Bishop M. Hypertext Transfer Protocol Version 3 (HTTP/3). – IETF Draft, 2016. – 70 с.
[14] Iyengar J., Thomson M. QUIC: A UDP-Based Multiplexed and Secure Transport. – IETF Draft, 2016. – 182 с.

КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 2 (10), 2020
ISSN 2663 - 4023

Ilyenko Anna
Candidate of Technical Sciences, assistant professor , assistant professor of Information Security Systems Department National Aviation University of Kyiv, Faculty of Cyber Security, Computer and Software Engineering, Ukraine
ORCID: 0000-0001-8565-1117
Ilyenko.a.v@nau.edu.ua

Ilyenko Sergii
Candidate of Technical Sciences, assistant professor , assistant professor of Automation and Energy Management Department National Aviation University of Kyiv, Aerospace Faculty, Ukraine
ORCID: 0000-0002-0437-0995
Ilyenko.s.s@nau.edu.ua

Vertypolokh Oleksandr
Student Information Security Systems Department
National Aviation University of Kyiv, Faculty of Cyber Security, Computer and Software Engineering, Ukraine
ORCID: 0000-0001-6228-8394
Vertipolokh@gmail.com

METHOD FOR PROTECTION TRAFFIC FROM INTERVENTION OF DPI SYSTEMS

Abstract. This article discusses further ways to protect traffic from DPI systems. The possibilities of using network protocols and application of DPI systems are investigated in the article. The analysis of the problem made it possible to identify vulnerabilities in the DNS protocol, which is based on the UDP protocol. These vulnerabilities include spoofing, interception, and traffic tethering. Also on the basis of the analysis of methods of protection of DNS traffic from interference, the authors substantiate and define the following: 1) all DNS queries are transmitted in the open; 2) existing approaches to traffic protection do not use encryption and, consequently, do not ensure the confidentiality of information; 3) there is only confirmation of the authenticity of the records. The authors have created a summary table, which identifies reliable methods of protecting DNS traffic. The authors propose the development of a full-fledged local proxy server to provide DNS traffic that can access trusted public DNS resolvers using doh and dot protocols. To understand the principles of protocol interaction, we developed our own local implementation of the main components of the network, which are most often dealt with by network users, namely: 1) web server; 2) DNS server; 3) server providing cryptographic protection and hiding open requests. The practical value of the obtained results lies in the software implementation of methods to protect traffic from DPI systems in Visual Studio Code by using the Python 3.8 programming language, which allows to provide cryptographic protection of traffic. The proposed solution of the local proxying server can be improved in the future by introducing local caching with the addition of the ability to create rules for certain domains and their subdomains. The implemented test doh server can be deployed on a trusted dedicated server outside of possible filter equipment installation points. This implementation will allow you to fully control your own traffic for resolving domain names. The authors further plan a number of scientific and technical solutions to develop and implement effective methods, tools to meet the requirements, principles and approaches to cyber security and traffic protection from interference by DPI systems in experimental computer systems and networks.

Keywords: Internet protocol, model TCP / IP, TLS, DNS, HTTPS, DNS-over-HTTPS, DNS-over-TLS, DPI.

REFERENCES

[1] Harold F., Krause M. Information Security Management Handbook, Sixth Edition. – Taylor &. Francis Group, 2007. – 3231 c. (in English)
[2] Arends R., Kosters M., Blacka D. DNS Security (DNSSEC) Opt-In, RFC 4956. – verisign, 2007. – 15 c. (in English)
[3] Dnscrypt 2 Protocol. [Online]. – Available: https://dnscrypt.info/protocol/ (in English)

[4] Hu Z., Zhu L., Heidemann J., Mankin A., Wessels D., Hoffman P. Specification for DNS over Transport Layer Security (TLS), RFC 7858. – USC/ISI, Verisign Labs, ICANN, 2016. – 18 c. (in English)
[5] Hoffman P., mcmanus P. DNS Queries over HTTPS (doh), RFC 8484. – ICANN, Mozilla, 2018. – 21 c. (in English)
[6] Huitema C., Shore M., Mankin A., Dickinson S., Iyengar J. Specification of DNS over Dedicated QUIC Connections. – Private Octopus, Fastly, Salesforce, 2019. – 18 c. (in English)
[7] Cid C., Jacobson M.J. Selected Areas in Cryptography. 25th International Conference Calgary, 2019. – 499 c. (in English)
[8] Stallings W. Cryptography and Network Security. Principles and Practice. 7th ed. – Pearson Education Limited, 2017. – 766 c. (in English)
[9] Oppliger R. SSL and TLS Theory and Practice, Second Edition. – Artech House, London, 2016. – 280 c. (in English)
[10] Mockapetris P. Domain names - concepts and facilities, RFC 1034. – USC/Information Science Institute, 1987. – 55 c. (in English)
[11] Mockapetris P. Domain names - implementation and specification, RFC 1035. – USC/Information Science Institute, 1987. – 55 c. (in English)
[12] Pollard B. HTTP2 in Action. – Manning Publications, 2019. – 384 c. (in English)
[13] Bishop M. Hypertext Transfer Protocol Version 3 (HTTP/3). – IETF Draft, 2016. – 70 c.
[14] Iyengar J., Thomson M. QUIC: A UDP-Based Multiplexed and Secure Transport. – IETF Draft, 2016. – 182 c. (in English).