ДОДАТОК А

Графічний матеріал атестаційної роботи

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

КАФЕДРА ЕЛЕКТРОННИХ ОБЧИСЛЮВАЛЬНИХ МАШИН

Метод автоматизованої побудови VPN-ланцюгів на платформі IaaS

Атестаційна робота здобувача
вищої освіти другого (магістерського) рівня

Здобувач:
Будько А.О.
студентка гр. СПм-19-1

Керівник:
Ткачов В.М.
доцент каф. ЕОМ

МЕТА ТА ЗАДАЧІ РОБОТИ

Метою атестаційної роботи є підвищення мережної безпеки шляхом розробки алгоритму автоматизованої побудови VPN-ланцюгів на платформі IaaS для побудови високозахищеної мережі на базі технології VPN.

Задачі, що потребують вирішення, для досягнення мети атестаційної роботи

- аналіз особливості побудови VPN-тунелів з використанням хмарних технологій;
- розробка алгоритму автоматизованої побудови VPN-ланцюгів на платформі IaaS;
- розробка математичної моделі процесу динамічного VPN-тунелювання в умовах неповних даних IaaS;
- дослідження динаміки побудови маршрутів VPN-ланцюгів на базі розробленого алгоритму.

2

АКТУАЛЬНІСТЬ РОБОТИ

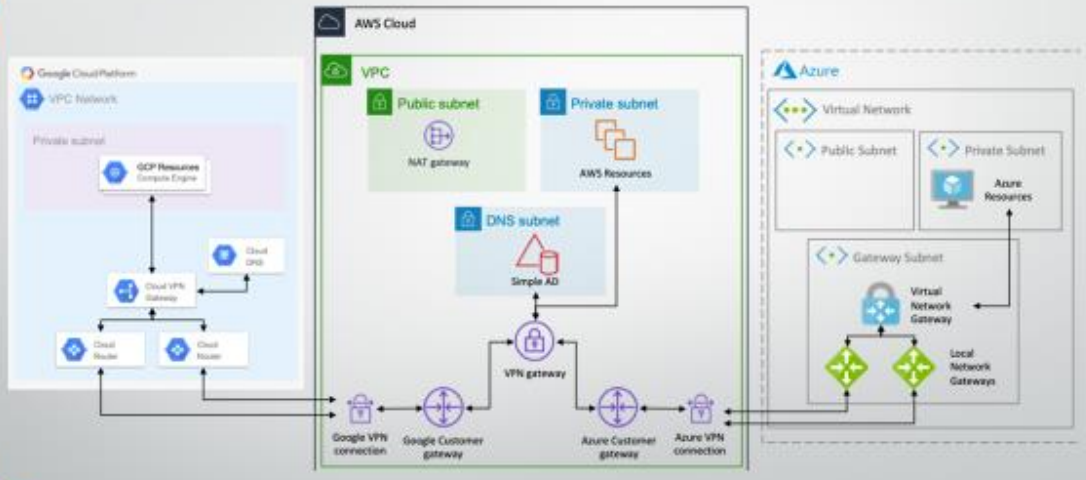
Технології хмарних обчислень займають потужну нішу майже у всіх IT-процесах. Це дозволяє виносити бізнес-майданчики саме у віртуальну площину, вибудовуючи віртуальні мережі. Сервери віртуальних машин, які розміщені в хмарній інфраструктурі, також є віртуальними, тобто все частішим є поширення концепції вкладеної віртуалізації. При цьому бізнес «на місцях» змушений підключатися до віртуальних хостів за допомогою терміналів. В даний час безпека систем термінального доступу забезпечується за рахунок стійкості протоколів обміну даними між терміналом і хостом.

Для підвищення безпеки таких з'єднань, з недавнього часу використовується концепція VPN-тунелювання. Її суть полягає у використанні проміжних VPN-серверів, взаємодію яких організовано за певним правилом. При цьому важливими завданнями є швидкість побудови такої оверлейної структури і підтримка в межах допустимих значень показників мережної затримки пакетів між кінцевими вузлами. Доволі складною є задача побудова ланцюгів з багатьох VPN-вузлів, що суттєво підвищує безпеку з'єднань.

Таким чином, актуальною є задача швидкої побудови VPN-ланцюгів з заданим рівнем безпеки і обов'язковою умовою витримки мінімально допустимої мережної затримки. При цьому використання IaaS дозволяє створювати та швидко розгортати платформонезалежні універсальні рішення при мінімальних економічних витратах.

3

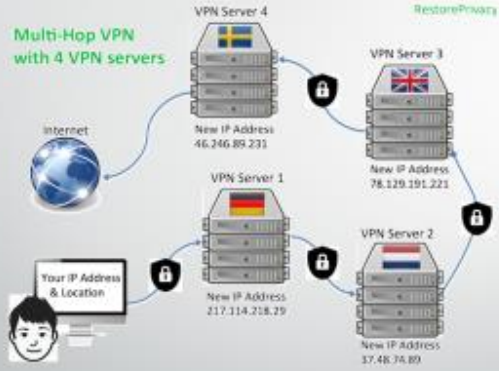
ОСОБЛИВОСТІ ЗАСТОСУВАННЯ ХМАРНИХ ТЕХНОЛОГІЙ ПРИ ПОБУДОВІ VPN-ТУНЕЛІВ



Приклад взаємодії між хмарами різних вендорів через VPN-тунелі

4

ОГЛЯД ТЕХНОЛОГІЙ ДЛЯ СТВОРЕННЯ VPN-ЛАНЦЮГІВ



- Double-hop VPNs
- Self-configurable multi-hop VPN
- «VPN всередині VPN» або «вкладений ланцюжок» VPN
- NeuroRouting

5

РОЗРОБКА МЕТОДУ АВТОМАТИЗОВАНОЇ ПОБУДОВИ VPN-ЛАНЦЮГІВ НА ПЛАТФОРМІ IAAS. Розробка математичного забезпечення

$$T_{(x\to y)} \to \min, при \begin{cases} T_{(x\to y)} \le T_{protocol} \\ T_{route} \le T_{task} \\ b = 1.2.n \end{cases}, \quad (1)$$

$$T_{route} = t_{route} + \sum t_{communication} \quad (2)$$

$$t_{route} = \begin{cases} t_{(\Theta)} + t_{(\Theta_x \to \Theta_y)} + t_{search\min}, \ if \ \ b = 1; \\ t_{(\Theta_x)} + t_{sort\min\Theta_x} + t_{search\,group\,I}, \ if \ \ b = 2; \\ t_{(\Theta)} + t_{sort\min\Theta} + t_{search\,group\,II}, \ if \ \ b = 3; \\ t_{(\Theta)} + t_{sort\min\Theta} + (n-1) \times t'_{sort\min\Theta_{b_n}} + \sum t_{(\Theta_{b_n})} + t_{search\,group\,I}, \\ якщо \ \ b = 4,6,..n, n = 2k, k \in \mathbb{N}; \\ t_{(\Theta)} + t_{sort\min\Theta} + n \times t'_{sort\min\Theta_{b_n}} + \sum t_{(\Theta_{b_n})} + t_{search\,group\,II}, \\ якщо \ b = 5,7,..n, \ n = 2k+1, \ k \in \mathbb{N}; \end{cases} \quad (3)$$

$$t_{(\Theta)} = \begin{cases} t_{(\Theta_x)}, \ if \ t_{(\Theta_x)} \ge t_{(\Theta_y)}; \\ t_{(\Theta_y)}, \ if \ t_{(\Theta_y)} \ge t_{(\Theta_x)}; \end{cases} \quad (4)$$

$$T_{protocol} - t_{(x\to b_1)} - \sum t_{b_n} \ge t_{(b_{n-1}\to b_n)} + t_{(b_n\to y)} \quad (5)$$

$$T_{protocol} - t_{(x\to b_1)} - \sum t_{b_n} - t_{(b_n\to y)} \ge t_{(b_{n-2}\to b_{n-1})} + t_{(b_{n-1}\to b_n)} \quad (6)$$

6

РОЗРОБКА МЕТОДУ АВТОМАТИЗОВАНОЇ ПОБУДОВИ VPN-ЛАНЦЮГІВ НА ПЛАТФОРМІ IAAS. Розробка алгоритмічного забезпечення (1)

Алгоритм для випадку, якщо користувачем обраний один VPN-сервер.

Крок 1. Паралельно відбувається наповнення матриць Θ_x і Θ_y. Час, що витрачається на цю операцію буде визначено, виходячи з виразу (5).

Крок 2. Матриці Θ_x і Θ_y складаються.

Крок 3. Пошук мінімального значення в складеної матриці.

Знайдений VPN-сервер буде обраний в якості рішення задачі.

Алгоритм для випадку, якщо користувачем вибрано два VPN-сервера.

Крок 1. Відбувається наповнення матриці Θ_x.

Крок 2. Сортування значень матриці Θ_x по зростанню.

Крок 3. В якості першого VPN-сервера береться той, час якого $t_{(x\to b_1)}$ мінімальний. Цей VPN-сервер виключається для подальшого пошуку другого VPN-сервера.

Крок 4. Виконується перевірка умови (6).

Крок 4.1. Перший знайдений, що задовольняє умові (6), VPN-сервер за час $t_{searchgroup1}$ зупиняє алгоритм.

Крок 4.2. Якщо задовольняючий умові (6), VPN-сервер не знайдений, то повертаємося до кроку 3 і беремо $t_{(x\to b_2)}$. Цикл може повторюватися до $t_{(x\to b_n)}$.

Визначені VPN-сервери будуть обрані в якості рішення задачі.

7

РОЗРОБКА МЕТОДУ АВТОМАТИЗОВАНОЇ ПОБУДОВИ VPN-ЛАНЦЮГІВ НА ПЛАТФОРМІ IAAS. Розробка алгоритмічного забезпечення (2)

Розглянемо послідовність кроків для випадку, якщо користувачем вибрано три VPN-сервери.

Крок 1. Паралельно відбувається наповнення матриць Θ_x і Θ_y. Час, що витрачається на цю операцію буде визначено, виходячи з виразу (5).

Крок 2. Паралельне сортування значень матриць Θ_x і Θ_y по зростанню.

Крок 3. В якості першого VPN-сервера береться той, час якого $t_{(x\to b_1)}$ чи $t_{(y\to b_1)}$ мінімальний. Якщо, наприклад, визначено першим VPN-сервер b_1, то він виключається для подальшого пошуку VPN-серверів b_2 і b_3.

Крок 4. Виконується перевірка умови (7).

Крок 4.1. Перший знайдений, що задовольняє умові (7), VPN-сервер за час $t_{searchgroup2}$ зупиняє алгоритм.

Крок 4.2. Якщо задовольняє умові (7), VPN-сервер не знайдений, то:

Крок 4.2.1. Беремо значення $t_{(x\to b_2)}$ і виконуємо перевірку умови (7). Цикл може повторюватися до $t_{(x\to b_n)}$.

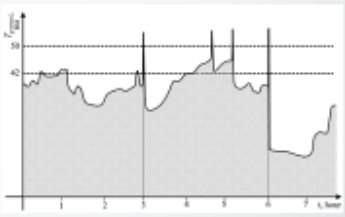
Крок 4.2.2. Якщо VPN-сервер не знайдений, то беремо значення і виконуємо перевірку умови (7). Цикл може повторюватися до $t_{(y\to b_n)}$.

Визначені таким чином VPN-сервери будуть обрані в якості рішення задачі. Якщо користувач вибирає 4 і більше VPN-серверів, то алгоритм має вигляд (4).

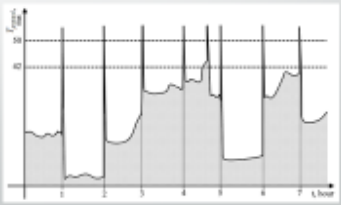
РОЗРОБКА МЕТОДУ АВТОМАТИЗОВАНОЇ ПОБУДОВИ VPN-ЛАНЦЮГІВ НА ПЛАТФОРМІ IAAS. Числовий експеримент. Результати

Вузол	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9	b_{10}	b_{11}	b_{12}	b_{13}	b_{14}	b_{15}
b_1	0	1	2	1	1	1	4	1	1	2	8	7	9	1	1
b_2	1	0	1	4	4	5	5	2	2	2	2	1	1	1	8
b_3	2	1	0	7	8	4	8	1	6	8	7	5	3	1	5
b_4	1	4	7	0	8	5	9	5	5	8	4	7	9	5	2
b_5	1	4	8	8	0	1	4	4	1	1	1	1	1	1	2
b_6	1	5	4	5	1	0	1	8	9	7	5	8	4	8	7
b_7	4	5	8	9	4	1	0	1	1	2	5	8	4	2	1
b_8	1	2	1	5	4	8	1	0	8	9	5	1	7	8	2
b_9	1	2	6	5	1	9	1	8	0	1	1	8	7	5	3
b_{10}	2	2	8	8	1	7	2	9	1	0	3	1	1	1	1
b_{11}	8	2	7	4	1	5	5	5	1	3	0	9	9	8	5
b_{12}	7	1	5	7	1	8	8	1	8	1	9	0	3	1	2
b_{13}	9	1	3	9	1	4	4	7	7	1	9	3	0	7	8
b_{14}	1	1	1	5	1	8	2	8	5	1	8	1	7	0	1
b_{15}	1	8	5	2	2	7	1	2	3	1	5	2	8	1	0
x	1	1	1	1	8	7	2	5	4	5	8	7	2	2	1
y	3	8	9	2	5	4	4	5	3	1	1	1	8	2	4

Результат роботи автоматизованої побудови VPN-ланцюга у матричному вигляді



Результат роботи Perfect Privacy



Результат роботи ПЗ з використанням запропонованого методу

9

МОДЕЛЬ ДИНАМІЧНОЇ МАРШРУТИЗАЦІЇ В VPN-ЛАНЦЮГАХ В УМОВАХ НЕПОВНИХ ДАНИХ IAAS-ІНФРАСТРУКТУРИ

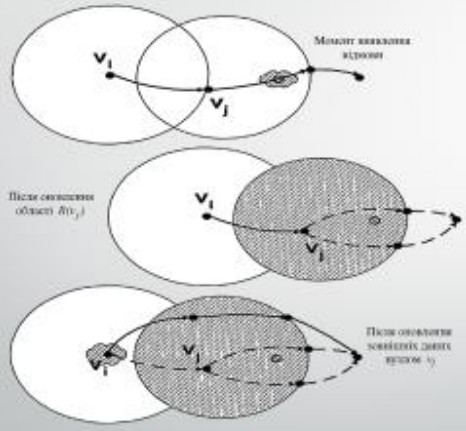
$$\sum_{u:(u,t)\in E} f^{(s,t)}_{(u,t)} - \sum_{u:(t,u)\in E} f^{(s,t)}_{(t,u)} = \begin{cases} -D(s,t), & \text{якщо } v = s, \\ D(s,t), & \text{якщо } v = t, \\ 0, & \text{в інших випадках.} \end{cases} \quad v,s,t \in V \qquad (7)$$

а) б) в)

Сценарії збору маршрутної інформації при побудові маршрутного представлення VPN-ланцюга:
а) вектор дистанції; б) стан каналу; в) область ефективної маршрутизації

10

ДОСЛІДЖЕННЯ ДИНАМІКИ ПОБУДОВИ МАРШРУТІВ VPN-ЛАНЦЮГІВ В УМОВАХ НЕПОВНИХ ДАНИХ ТА ВІДМОВ ВУЗЛІВ В IAAS-ІНФРАСТРУКТУРІ



Оновлення маршрутів після відмови вузла в VPN-ланцюзі

Схема передачі даних в VPN-ланцюгу

11

АПРОБАЦІЯ РЕЗУЛЬТАТІВ РОБОТИ

Method of Building Dynamic Multi-hop VPN Chains for Ensuring Security of Terminal Access Systems

IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T): Kharkiv 06-09 oct. 2020, Kharkiv.

12

ВИСНОВКИ

За результатами роботи мета була досягнута: отримано рішення щодо підвищення безпеки мережних з'єднань за рахунок використання розробленого алгоритму автоматизованої побудови VPN-ланцюгів на платформі IaaS для побудови високозахищеної мережі на базі технології VPN.

Основні задачі, які були вирішені:

- проаналізовано особливості побудови VPN-тунелів з використанням хмарних технологій;
- розроблено алгоритму автоматизованої побудови VPN-ланцюгів на платформі IaaS;
- розроблено математичну модель процесу динамічного VPN-тунелювання в умовах неповних даних IaaS;
- досліджено динаміку побудови маршрутів VPN-ланцюгів на базі розробленого алгоритму.

Новизна даної атестаційної роботи полягає в тому, що вперше при побудові VPN-ланцюгів, з метою зниження складності задачі (і, як наслідок, часу її рішення) було використано сукупність комбінаторних підходів, що дають логарифмічну складність завдання і дозволяють за більш менший час побудувати VPN-ланцюг для підвищення рівня захищеності середовища обміну даними в глобальній мережі між терміналом та хостом.

13

ДОДАТОК Б

Фрагменти програмного коду, що реалізують алгоритм перебудови VPN-ланцюга

```
      #!/bin/sh
      set -e
      echo
      echo 'Welcome to the corner-cutting configure script !'
      echo
      if [ ! -d "build" ]; then
        mkdir build
      fi
      if [ ! -z ${CMAKE_FLAGS+x} ]; then
      CMAKE_FLAGS="${CMAKE_FLAGS}"
      fi
      if [ ! -z ${CMAKE_INSTALL_PREFIX+x} ]; then
      CMAKE_FLAGS="-DCMAKE_INSTALL_PREFIX=${CMAKE_INSTALL_PREFIX}
${CMAKE_FLAGS}"
      fi
      if [ -z ${OPENSSL_ROOT_DIR} ]; then
      unameOut="$(uname -s)"
      if [ "$unameOut" = "Darwin" ]; then
            echo "Environment variable OPENSSL_ROOT_DIR not set, using
default Homebrew path: /usr/local/opt/openssl/"
            export OPENSSL_ROOT_DIR="/usr/local/opt/openssl/"
      fi
      fi
      if [ ! -z ${CPACK_GENERATOR+x} ]; then
      echo "CPACK_GENERATOR is set, CPack will generate ${CPACK_GENERATOR}
packages."
      CMAKE_FLAGS="-DCPACK_GENERATOR=${CPACK_GENERATOR} ${CMAKE_FLAGS}"
      elif [ -x "$(command -v rpm)" ]; then
      echo "'rpm' executable found, CPack will generate RPM packages."
      CMAKE_FLAGS="-DCPACK_GENERATOR='RPM' ${CMAKE_FLAGS}"
      else
      echo "'rpm' executable not found, CPack will generate DEB packages."
      CMAKE_FLAGS="-DCPACK_GENERATOR='DEB' ${CMAKE_FLAGS}"
      fi
      echo ""
      (cd build && cmake -DCMAKE_BUILD_TYPE=RelWithDebInfo ${CMAKE_FLAGS} ..
|| exit 1)
      echo ""
      echo "The Makefile is generated. Run 'make -C build' to build SoftEther
VPN."
```

Приклад Б.1 – Структура конфігураційного файлу configure

```
      #! / bin / bash
      set -eux
      download_libressl () {
          если [[ !   -f   " скачать-кеш / librenssl- $ {LIBRESSL_VERSION}
.tar.gz " ]] ;  тогда
              wget -P загрузка-кеш / \
                  "  https://ftp.openbsd.org/pub/OpenBSD/LibreSSL/libressl-  $
{LIBRESSL_VERSION} .tar.gz "
          фи
      }
      build_libressl () {
          если [[ " $ ( cat $ {OPENSSL_INSTALL_DIR} /.openssl-version ) "  ! =
" $ {LIBRESSL_VERSION} " ]] ;  тогда
              tar  zxf  "  download-cache  /  libressl-  $  {LIBRESSL_VERSION}
.tar.gz "
              cd  " libressl- $ {LIBRESSL_VERSION} / "
              ./configure --prefix = " $ {OPENSSL_INSTALL_DIR} "
              make -j $ ( nproc || sysctl -n hw.ncpu ||  echo 4 ) все
              сделать установку
              echo   " $ {LIBRESSL_VERSION} "   >   " $ {OPENSSL_INSTALL_DIR}
/.openssl-version "
          фи
      }
      download_libressl
      build_libressl
```

Приклад Б.2 – Структура файлу build-libressl.sh

Повна структура програмної реалізації розміщена за посиланням: https://github.com/SoftAnnEtherVPNvIaaS.