Міністерство освіти і науки України
Національний університет «Чернігівська політехніка »

МОДЕЛЮВАННЯ АНАЛІЗ ТА ІНСТРУМЕНТАЛЬНІ ЗАСОБИ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ

МЕТОДИЧНІ ВКАЗІВКИ

до виконання лабораторних робіт для здобувачів вищої освіти за освітньою програмою «Інженерія програмного забезпечення »
(освітній ступінь бакалавр)

Чернігів 2024

Зміст

Вступ

Дисципліна «Моделювання, аналіз та інструментальні засоби інформаційної безпеки »відноситься до обов'язкових дисциплін за освітньою програмою «Інженерія програмного забезпечення »(освітній рівень магістр).

Необхідною передумовою для вивчення даної дисципліни є проходження бакалаврськоі програми.

В наш час питання побудови захищеної інформаційної інфраструктури постає особливо гостро, у зв'язку із глобальною цифровізацією усіх процесів. при цьому на фактор захищеності впливає не тільки наявність окремих засобів безпеки, зокрема мережевої, а і правильно спланована архітектура вцілому. Дані методичні вказівки створені для допомоги в отриманні практичних навичок по проектуванню безпечної інформаційної інфраструктури, з точки зору мережевої безпеки. Мережева сторона інформаційної безпеки обрана в першу чергу через те, що саме комп'ютерні мережі відкривають широкий спектр загроз інформаційній безпеці.

В даних методичних вказівках розглядається питання моделювання мережевої інфраструктури різними методами, включаючи дискретно подійне моделювання та один із різновидів імітаційного - симуляцію. Такі методи моделюванні дозволяють виявити та виправити недоліки в мережевій топології ще на етапі проектування. Також лабораторні роботи включають різні методи пошуку вразливостей в мережевій інфраструктурі та застосунках, що дозволяє оцінити рівень захищеності та виправити виявлені проблеми.

Більша частина робіт запропонована в форматі проекту, коли здобувач має самостійно обрати предметну область і розробити топологію інформаційної інфраструктури, що дає можливість більш детально опрацювати підготовчі етапи та сформувати вимоги до цільової системи. Лабораторна робота №3 відрізняється від інших і передбачає пошук вразливостей веб-застосунків в форматі гри CTF. Такий формат з одного боку дає можливість застосувати знання та навички для пошуку реаль-

них вразливостей на спеціально підготовлених веб-застосунках, а з іншого дає можливість здобувачу самостійно обрати відповідний рівень складності завдань.

Лабораторна робота №1 Моделювання мереж в системі OMNeT++

Мета роботи

Мета лабораторної роботи полягає в моделюванні комп'ютерної марежі або її ділянки за допомогою засобів дискретно-подійного моделювання [1], з метою оцінки її параметрів для різних контекстів використання.

Завдання

1) Обрати предметну область та задачі для вирішення яких буде плануватись мережа;

2) Розробити дві різні топології мережі що відповідають поставленій задачі;

3) Змоделювати в середовищі OMNeT++ розроблені топології та зібрати дані по результатам моделювання;

4) Виконати аналіз результатів та порівняти характеристики розроблених топологій мереж.

Теоретичні відомості

OMNeT++ – це потужна, модульна та орієнтована на компоненти C++ бібліотека та платформа для моделювання, перш за все розроблена для

створення симуляторів мереж. Під ”мережею” тут розуміється у широкому сенсі: дротові та бездротові комунікаційні мережі, чіпові мережі, мережі масового обслуговування тощо.

Система OMNeT++ являє собою симулятор дискретних подій. Зміна стану модельованої системи відбувається в дискретні моменти часу відповідно до списка майбутніх подій (future eventlist), які відсортовано за часом. Подією може бути: початок передачі пакета, таймаут і т.п. Події відбуваються на основі виконання простих модулів (simple module). У такого модуля є функції ініціалізації, обробки повідомлення, дії і завершення роботи.

Система INET Framework - це комплект модулів з відкритим вихідним кодом, які дозволяють реалістично моделювати вузли мереж та протоколи дротових і бездротових мереж. Він включає моделі різних протоколів Інтернету: IP, IPv6, TCP, UDP, 802.11, Ethernet, PPP, MPLS з LDP і RSVP-TE signalling, OSPF і ряд інших. У комплект також входять різні реалістичні приклади використання цих протоколів.

Найвищий рівень абстракції в моделюванні IP в INET Framework представлено мережею, яка складається з IP-вузлів. Вузол може бути маршрутизатором або хостом. IP-вузол відповідає комп’ютерному поданню стека протоколів Інтернет. Модулі, з яких він складається, організовані таким чином, щоб моделювати обробку IP-дейтаграм в операційних системах. Обов’язковим є модуль, який відповідає за мережевий рівень (який реалізує 14 обробку IP) і модуль ”мережевий інтерфейс”. Додатково підключаються модулі, що реалізують протоколи транспортного рівня.

Встановлення системи

Завантажити останню версію системи OMNeT++ можна на офіційному сайті, остання на момент написання даної лабораторної роботи версія 6.1.0. Посилання на заватнаження https://omnetpp.org/download/. Для операційної системи Windows пакунок представляє собою середовище MinGW, Eclips IDE та саму платформу моделювання. Після розпаковки архіву треба запустити .bat файл, який запустить розгортання середовища, а після цього виконати компіляцію:

```
./configure
make #(можна додати -j  та к□льк□сть поток□в
 ↪  в□дпов□дно до ядер процесора, для
 ↪  пришвидшення комп□ляц□□, наприклад -j8
 ↪  -j24)
```

Після успішної компіляції модна запустити програму командою

```
omnetpp
```

Зверніть увагу

На момент написання даного документу версія 6.1.0 містить некоректне посилання на завантаження пакета INET, через що його автоматичне встановлення при першому запуску OMNET++ неможливе. Для ручного встановлення можна скористатись меню ***HELP*** *→* ***Install Simulation Models*** *та обрати одну з попередніх версій пакета і встановити його в систему, або завантажити пакет INET з офіційного веб-сайту та встановити його вручну користуючись офіційною інструкцією по встановленню що іде разом з пакетом*

Використання OMNet++

Пакет OMNet++ включає можливості моделювання широкого спектру систем, але в рамках даної лабораторної роботи зупинимось на моделюванні комп'ютерних мереж. Процес моделювання включає наступні етапи:

1) За допомогою компонентів, взятих із пакету INET або створених самостійно, відтворити потрібну топологію мережі. За необхідності внести корективи в файл вихідного коду моделі *(зазвичай виникає необхідність виправляти код* ***мережевих з'єднань****)*

2) Внести в конфігураційний файл `omnetpp.ini` налаштування мережі та програм.

3) Скомпілювати та запустити модель.

Простий урок по використанню мережевих компонентів і роботі з програмою OMNeT++ наведено за наступним посиланням:
https://youtu.be/ujQ_jaItx_Y

Приклади використання роутерів та інших мережевих компонентів наведені в офіційній документації на пакет мережевих компонентів:
https://inet.omnetpp.org/docs/tutorials/configurator/doc/step1.html
https://inet.omnetpp.org/docs/users-guide/index.html

Зверніть увагу

Моделювання в данному пакеті лише імітує відправку пакетів того чи іншого типу, і не відображає повністю мережеві пакети.

Також в демонстраційних проектах наведено різні варіанти топологій з прикладами використання, зокрема inet4.5/examples/internetcloud/ та інші.

Особливості створення моделі

При створенні нових з'єданань між доданими вузлами, необхідно перемкнутися в режим перегляду вихідного коду `.ned` файлу, оскільки створений за замовчуванням код не скомпілюється. В кінці файлу, в секції `:connections` будуть створені записи наступного плану(далі наведено абстрактний приклад, в вашому випадку назви пристроїв можуть відрізнятися):

```
standardHost.ethg[0] <--> Eth100M <-->
 ↪  switch0.ethg[0];
```

В даному випадку текст `ethg[0]` в усіх записах про з'єднання необхідно замінити на `ethg++`, тобто записи повинні мати наступний вигляд:

```
host2.ethg++ <--> Eth100M <--> switch0.ethg++;
```

Моделі програм за пакету INET

Програми INET поділяються на дві категорії:

- Специфічні програми, які реалізують дуже конкретні поведінки та генерують відповідні трафікові моделі на основі своїх параметрів. Приклади таких програм включають `TcpBasicClientApp` та `TcpGenericServerApp`.

- Загальні програми, які розділяють генерацію трафіку від використання протоколу, наприклад, `UdpApp`, `UdpSourceApp`, `UdpSinkApp` та `UdpRequestResponseApp`, `TcpSessionApp`.

Загалом бібліотека INET містить наступні основні програми [2]:

1) **`TcpBasicClientApp`** - Клієнтський застосунок для генерації запитів та отримання відповідей у стилі протоколу TCP. Можливе використання для грубого моделювання застосунків HTTP або FTP.

2) **`TcpGenericServerApp`** - Загальний серверний застосунок для моделювання взаємодії між клієнтом і сервером у стилі запит-відповідь у протоколах або застосунках, що використовують TCP. Він приймає будь-яку кількість вхідних TCP-з'єднань та очікує отримання повідомлень класу `GenericAppMsg`.

3) **`TCPEchoApp`** - Подібна до `TcpBasicApp`, але відправляє назад отримані пакети після їх прийому.

4) **`TcpSessionApp`** - TCP-застосунок з одиночним з'єднання TCP-застосунок з одиночним з'єднанням, який відкриває з'єднання, відправляє задані пакети, які складаються з пари (час, кількість байтів). Він може діяти як клієнт або сервер і сумісний з IPv4 та IPv6.

5) **`UdpApp`** - Загальний серверний застосунок UDP з композитним джерелом та стоком трафіку.

6) **`UdpRequestResponseApp`** - Загальна серверна програма UDP з композитним джерелом та стоком трафіку.

7) **`PingApp`** - Програма, яка генерує ping-запити та обчислює втрати пакетів та параметри кругового шляху відповідей.

8) **`UdpEchoApp`** - Подібна до `UdpBasicApp`, але відправляє назад отримані пакети після їх прийому.

9) **`UdpVideoStreamClient`** - Клієнт відеопотоку, який відправляє запит на відеопотік до сервера в призначений час та отримує потік від `UdpVideoStreamServer`.

10) **`UdpVideoStreamServer`** - Це сервер відеопотоку, який використовується з `UdpVideoStreamClient`. Сервер чекає на вхідні запити на відеопотік. Коли запит надходить, сервер генерує випадкову

довжину відеопотоку за допомогою параметра videoSize, відправляє UDP-пакети довжиною `packetLen` кожні `sendInterval`, доки не буде досягнуто `videoSize`.

11) **`TelnetApp`** - Модель `Telnet` сеансів.

Завдання (деталізовно)

Зверніть увагу

Завдання даної лабораторної роботи передбачає самостійний вибір предметної області, задач на вирішення яких орієнтована мережа та мережевої топології, тому далі наведені базові елементи завдання, які зорієнтують вас у правильному напрямку роботи, остаточне завдання може бути скоригованим в залежності від обраної мережі.

Основне завдання лабораторної роботи включає наступні кроки:

1) Оберіть предметну область та задачу для вирішення якої буде призначена ваша мережа. Це найбільш важливий етап лабораторної роботи, оскільки від вашого вибору залежатимуть параметри які ви будете оцінювати за допомогою моделі. Визначте основні ролі для учасників мережі та оцініть яке програмне забезпечення та які приблизні об'єми трафіка будуть призначені для кожного з учасників. В якості вузлів мережі можуть бути і клієнтські ком'ютери, і сервери, і мережеві сховища і камери відеонагляду, і відеореєстратори і звичайно ж мережеві пристрої, такі як маршрутизатори та комутатори.

2) Після цього запропонуйте принаймні два варіанта топології мережі, враховуйте, що для наглядного результату мережа має бути з достатньо складною топологією, при якій наприклад пакет має проходити в деяких випадках через кілька маршрутизаторів.

3) В середовищі OMNeT++ побудуйте моделі мереж з обраними вами топологіями та налаштуйте їх відповідно до вказаних вами задач. Пакет компонентів INET 4.5 містить багато варіантів симуляції програмного забезпечення, тому знайти готовий варіант програми для симуляції трафіка на повинно бути проблемою. Впевніться що обрані варіанти "програм" дозволяють зібрати метрики які ви збираєтесь отримати та використати для оцінки і порівняння топологій.

Параметри за якими буде проводитись оцінка можуть бути обрані на свій розсуд в залежності від параметрів обраної мережі, але всі вибори повинні бути обґрунтовані у вашому звіті.

4) Скомпілювати та запустити процес симуляції, впевнитися, що побудована модель працює саме так як і задумувалося, за необхідності внести виправлення та перекомпілювати модель.

5) Перейти до збережених результатів симуляції, де записані основні дані зібрані в процесі роботи моделі, та провести аналіз отриманих даних відомими вам методами, та отримати на основі аналізу оцінку швидкодії мережі (або іншого параметра який ви можете оцінити)

6) Зробити висновки щодо на основі отриманих оцінок та висловити думку щодо доцільності чи недоцільності використання запропонованих вами топологій мереж.

У висновках до лабораторної роботи опишіть переваги та недоліки дискретно-подійного моделювання в контексті моделювання комп'ютерних мереж та мереж зв'язку, а також наведіть короткі результати аналізу отриманих під час моделювання даних.

Орієнтовні теми для теоретичних питань

- Загальне поняття «Моделювання». Які задачі вирішує моделювання мереж.
- Основні види моделей, їх особливості.
- Дискретно подійне моделювання.
- Основні можливості пакета OMNET++ для моделювання мереж.

Лабораторна робота № 2 Імітаційне моделювання та емуляція роботи захищених комп'ютерних мереж.

Мета

Мета лабораторної роботи полягає в розробці та реалізації моделі захищеної комп'ютерної мережі, оцінці ефективності її захисту від зовнішніх загроз, а також у вивченні ефективності різних заходів захисту мережі та оцінці рівня її захищеності на основі результатів моделювання та тестування.

Завдання

1) Побудувати імітаційну модель комп'ютерної мережі, що включає програмні або апаратні засоби захисту інформаційної інфраструктури та правильно налаштувати всі мережеві комноненти.

2) Оцінити рівень захищеності за допомогою різних програмних засобів для тестування на проникнення, а також вплив наявності чи відсутності засобів безпеки на рівень захищеності.

Теоретичні відомості

Зверніть увагу

Оскільки основне програмне забезпечення для емуляції комп'ютерних мереж є достатньо ресусоємним та при цьому неефективно працює в операційній системі ***Windows****, дана лабораторна робота передбачає альтернативні варіанти виконання які можуть забезпечити більш стабільну роботу на слабких системах. Тому перед виконанням ознайомтеся з усіма альтернативними варіантами виконання і при неможливості виконати роботу в програмі GNS3 використовуйте альтернативні варіанти*

Імітаційне моделювання мереж

Для дослідження різноманітних характеристик комп'ютерних мереж використовуються методи моделювання. У сучасних умовах, коли інфраструктура мереж стає дедалі складнішою і вимоги до її ефективності зростають, використання фізичних тестів часто є занадто затратним та ризикованим. Імітаційні моделі надають можливість вивчати взаємодію мережевих компонентів, аналізувати навантаження і визначати вузькі місця без втручання у роботу реальної мережі. Це робить імітаційне моделювання важливим інструментом для попереднього оцінювання різних архітектур мереж, розробки оптимальних рішень і забезпечення надійності та безпеки комп'ютерних мереж.

Імітаційне моделювання — це метод дослідження, заснований на тому, що система, яка вивчається, замінюється імітатором і з ним проводяться експерименти з метою отримання інформації про цю систему. Експериментування з імітатором називають імітацією (імітація — це збагнення суті явища, не вдаючись до експериментів на реальному об'єкті).

Одним із варіантів імітаційного моделювання є емуляція. При емуляції модель точно відтворює роботу реальної системи так, що за більшістю параметрів це відповідає реальній системі. Ще одним схожим підвидом програмного забезпечення є програмне забезпечення для симуляції, яке може бути на перший погляд схоже на емулятор, але передбачає достатньо серйозні спрощення та обмеження. Далі розглянемо відмінності між термінами симулятор та емулятор.

Симулятор — імітує якийсь набір команд, який є незмінним і не дозволяє користувачеві вийти за цей набір. При спробі виконання непід-

тримуваної команди, ми відразу отримаємо повідомлення про помилку. Класичний приклад програми–симулятора — Cisco Packet Tracer.

Емулятори ж навпаки — дозволяють програвати (виконуючи байт трансляцію) образів (прошивки) реальних пристроїв, найчастіше без видимих обмежень. Прикладом емулятора є програмний продукт GNS3.

GNS3

GNS3 використовується сотнями тисяч інженерів мереж по всьому світу для емуляції, налаштування, тестування та усунення неполадок віртуальних і реальних мереж. GNS3 дозволяє запускати як невеликі топології, що складаються лише з кількох пристроїв на вашому ноутбуці, так і великі, з багатьма пристроями, розміщеними на декількох серверах або навіть у хмарі.

GNS3 — це безкоштовне програмне забезпечення з відкритим кодом, яке можна завантажити за адресою http://gns3.com.

Програма активно розвивається та підтримується, і має зростаючу спільноту з понад 800 000 учасників.

GNS3 підтримує кілька типів емуляторів які можуть запускати різні типи програмного забезпечення мережевих пристроїв. Розглянемо основні з них:

- **Dynamips** — це технологія, яку використовує GNS3 з моменту заснування, і емулює маршрутизатори Cisco та базову комутацію за допомогою модуля Etherswitch. Використовує оригінальні образи програмного забезпечення, але при цьому коректний запуск усіх образів не гарантується оскільки Cisco не підтримує запуск IOS програмного забезпечення на сторонніх пристроях. Підтримуються достатньо старі пристрої.

- **QEMU** — Стандартнасистема віртуалізації для Linux, підтримує запуск практично усіх образів операційних систем і добре інтегрується в GNS3. Щоб використовувати образи Cisco, такі як IOSvL2, потрібно придбати підписку Cisco VIRL. Це дозволить отримати доступ до образів, які працюють з GNS3 та схвалені командою Cisco. Образи VIRL створюються спеціально для симуляції та працюють особливо добре, якщо потрібно використовувати новіші версії операційної системи Cisco та сучасні функції. Фактично це єдиний офіційний спосіб запуску програмного забезпечення Cisco.

- **IOU** — це внутрішній спосіб Cisco запускати IOS на Unix. Це не ресурсоємна програма з точки зору ЦП та пам'яті, що робить її добрим

вибором. IOU підтримує як маршрутизатори, так і комутатори. Але при цьому дане програмне забезпечення не є офіційним і образи та ліцензії для запуску образів таким чином неможливо придбати офіційними шляхами. Стабільність роботи образів на такому емуляторі також не гарантується.

- **VMware / VirtualBox** — GNS3 підтримує інтеграцію віртуальних машин VMware та VirtualBox, але така інтеграція є достатньо поверхневою і не передбачає збереження таких віртуальних машин в проекті, тому даний варіант не можна назвати рекомендованим. Але в деяких випадках він може більш швидким аніж використання QEMU.

- **VPCS** (Симулятор віртуального ПК) — це легкий спосіб емуляції дуже базового ПК. VPCS використовує дуже мало пам'яті, тому він є добрим вибором, коли ви хочете емулювати ПК без графічного інтерфейсу та якщо вам потрібні лише прості команди, такі як ping, щоб перевірити зв'язок у ваших мережах GNS3. Для розширеного функціоналу варто використовувати інші варіанти.

- **Docker** — підтримка Docker в GNS3 надає вам можливість запускати кілька контейнерів як частину топології GNS3. Це один з найменш ресурсоємних варіантів запуску повноцінної операційної системи Linux.

Увага

GNS3 не надає ніяких образів операційних систем, за виключенням VPCS, що інтегровані в систему. при створенні шаблона пристроя GNS3 може надати посилання на скачування образа якщо обрано варіант операційної системи з відкритим вихідним кодом який доступний для використання безкоштовно. Docker образи безкоштовних проектів скачуються з репозиторія автоматично. Всі інші образи необхідно отримувати самостійно, при цьому вся відповідальність за дотримання ліцензійних угод покладена на користувача. Єдиним офіційним варіантом є оплата ліцензії у розробника відповідної операцінйої системи.

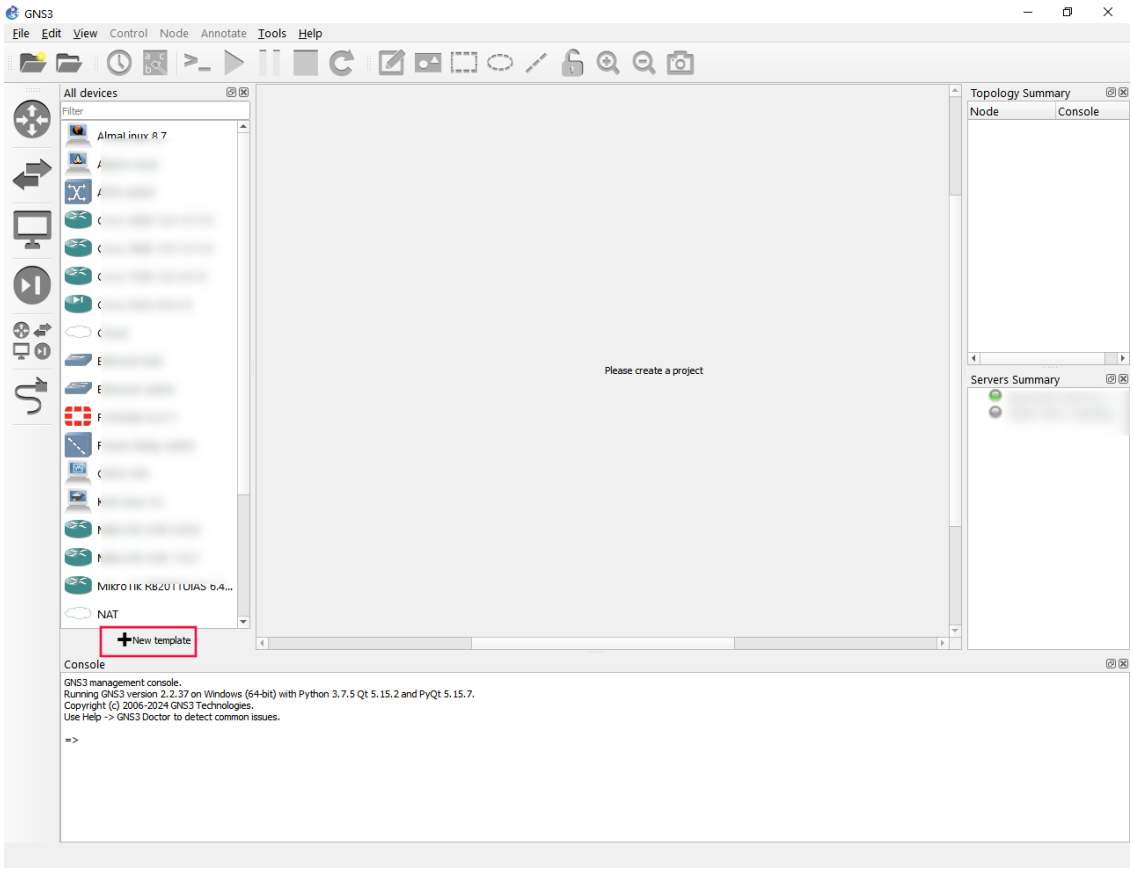
Використання

Для створення мережевих топологій в GNS3 в першу чергу необхідно додати шаблони пристроїв, на основі яких будуть створюватись моделі

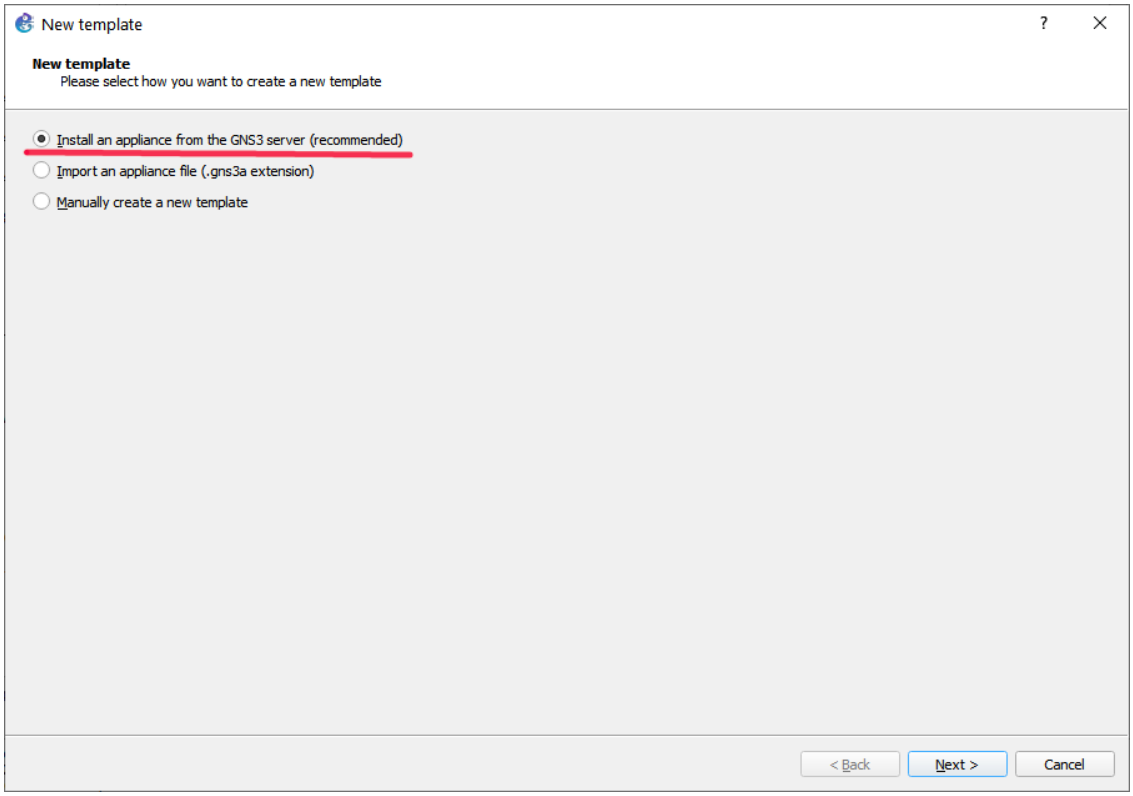
мереж. Шаблон пристроя можна або створити з нуля, або підвантажити образи операційних систем до існуючих шаблонів за серверів GNS3.

Розглянемо другий варіант. Для створення шаблона пристрою таким чином необхідно пройти наступні кроки:

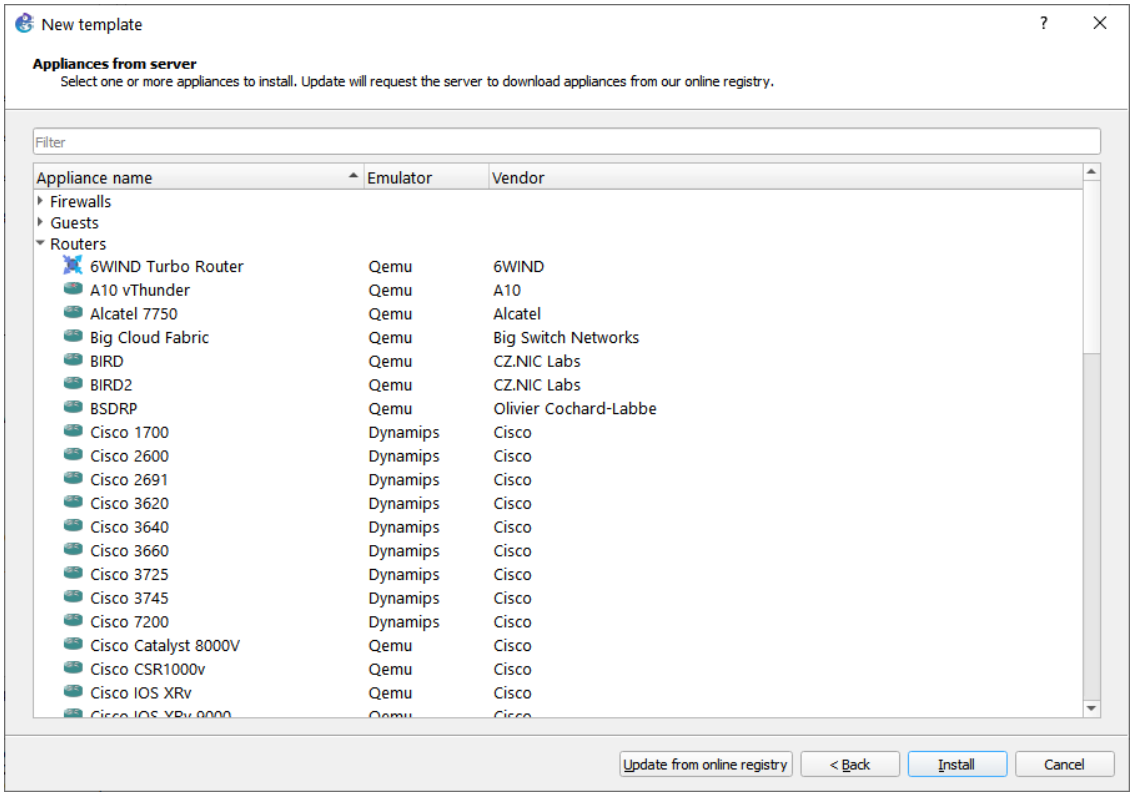
Крок 1: Натисніть на «переглянути всі пристрої”, а потім виберіть кнопку «Новий шаблон пристрою ».



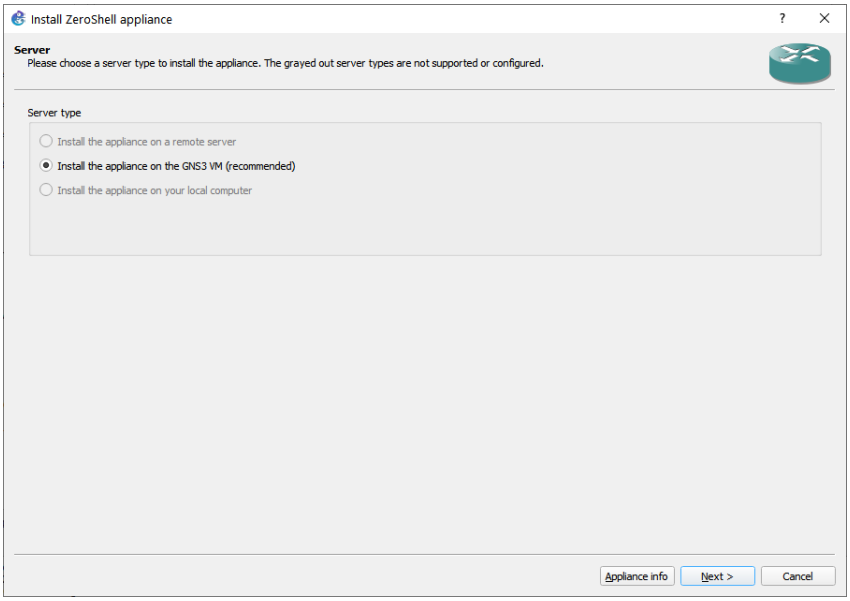
Крок 2: Виберіть «Встановити з сервера GNS3».



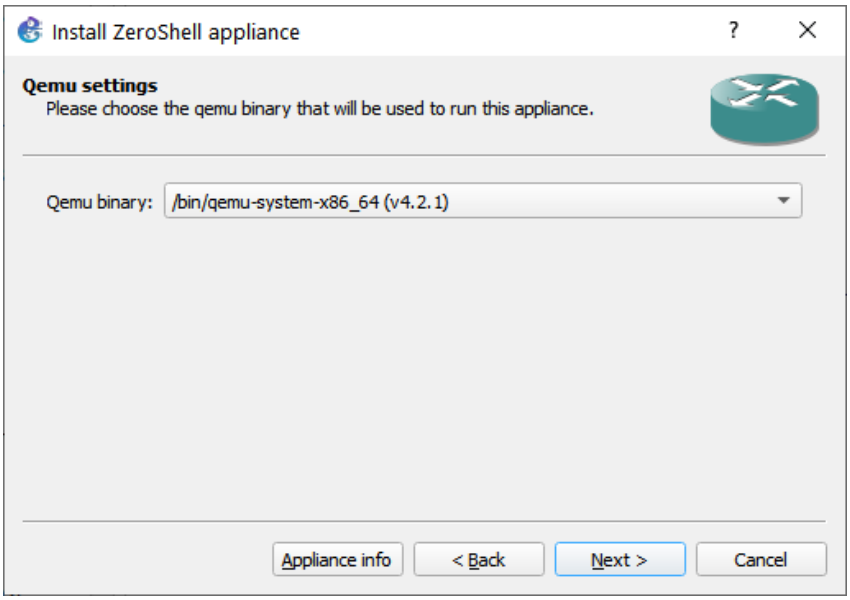
Крок 3: Обираємо категорію та пристрій із наданого списку, за необхідності список можна оновити з онлайн реєстру. Потім натисніть «Встановити».



Крок 4: Виберіть варіант встановлення із доступних — якщо у вас наявний локальний або віддалений сервер то відповідні опції будуть доступні інакше для більшості пристроїв залишається варіант віртуальна машина GNS3. Далі натискаємо «Далі».

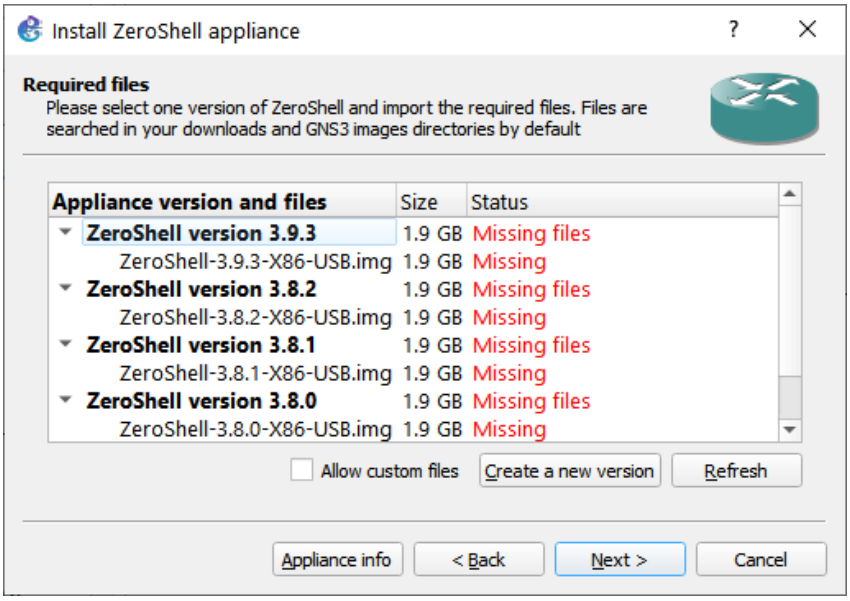


Крок 5: Зазвичай базову програму для запуску віртуальної машини міняти не треба, тому натискаємо «Далі».

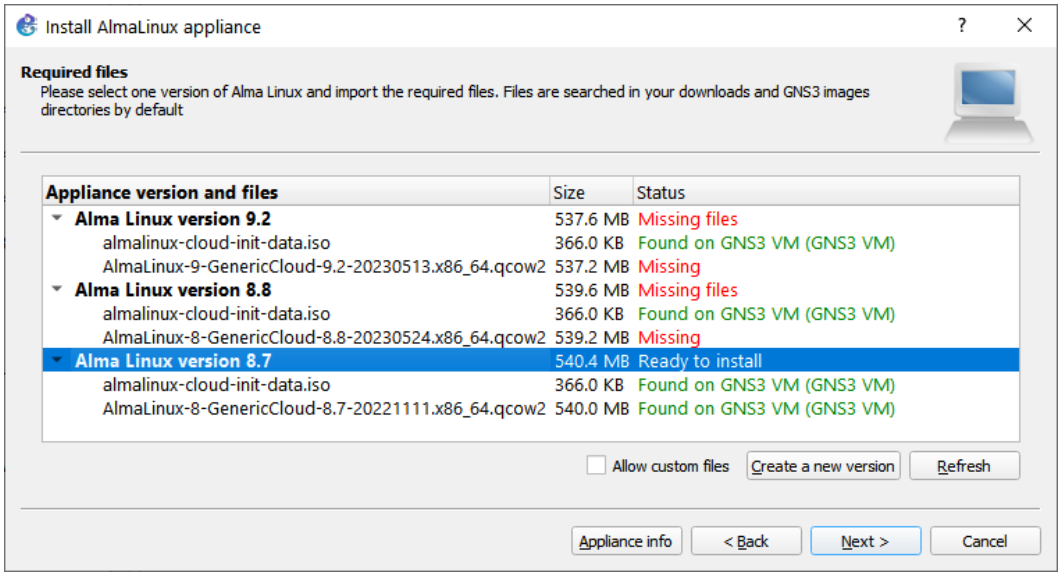


Крок 6: На наступному екрані бачимо перелік версій та наявні або відсутні файли необхідні для їх запуску. Для Open Source та безкоштовних продуктів можна натиснути кнопку «Завантажити», виділивши необхідний файл, в інших випадках файл треба шукати самостійно. завантажений файл треба імпортувати, а якщо версія не співпадає з вказаною

але ви вважаєте, що файл повинен підійти - треба поставити прапорець Дозволити всі файли, після імпорту статус файлу повинен змінитись на **наявний**.



Крок 7: Після успішного імпорту файлу виберіть «Далі».



Крок 8: Нарешті, ви можете побачити піктограму обраного пристроя у вкладці «Встановлені пристрої».

Детальні інструкції по створенню шаблонів опубліковані за наступними посиланнями «How to add Appliance and Image on GNS3», а також «Import GNS3 appliance»

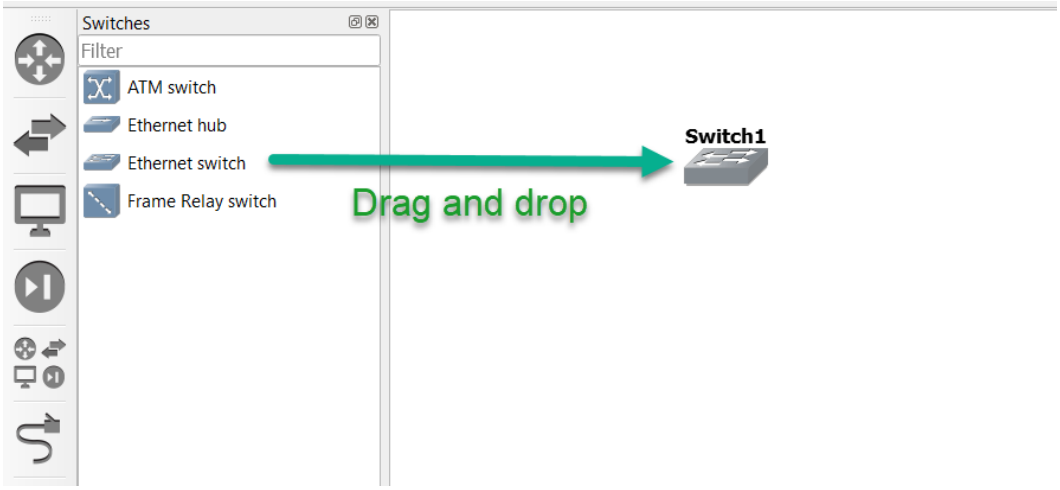
За необхідності можна створити потрібний шаблон самостійно (в тому числі і шаблони для використання готових віртуальних машин на

основі **VirtualBox**) за інструкцією розташованою за посиланням How to create a new GNS3 appliance template.

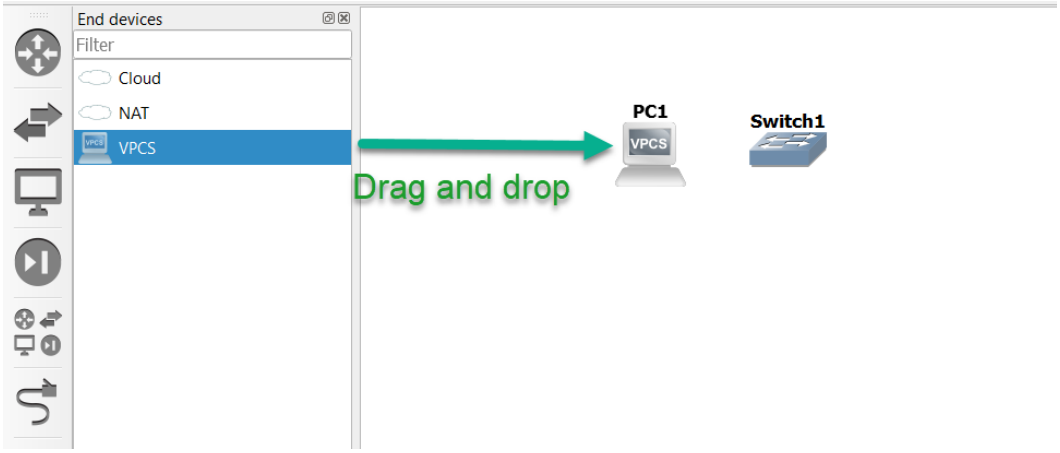
Детальну інструкцію по створенню мережевої топології в середовищі GNS3 можна знайти в офіційній документації за посиланням Your First GNS3 Topology

Щоб створити нову топологію, натисніть Browse End Devices на панелі інструментів пристроїв. Панель розшириться, показуючи доступні пристрої цього типу. У цьому прикладі VPCS є одним із доступних пристроїв

Для створення першої топології натисніть Switches на панелі пристроїв. Перетягніть вбудований Ethernet-комутатор до робочої області GNS3. Тепер пристрій з назвою Ethernetswitch-1 буде доступний у топології.

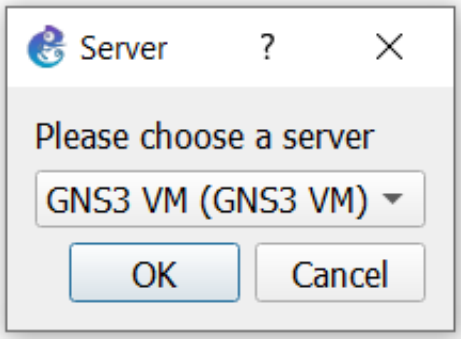


Потім натисніть End Devices, перетягніть VPCS (простий емулятор ПК) у робочу область, який називатиметься PC-1.



Якщо GNS-VM уже імпортовано та налаштовано, з'явиться запит, чи ви хочете запустити вузол VPCS на локальному сервері або в GNS3-VM.

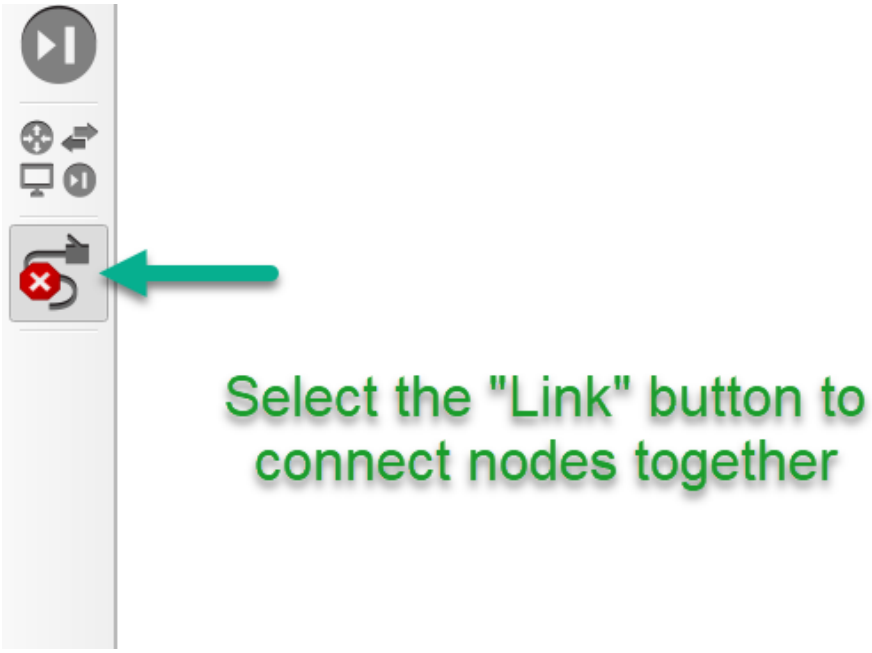
Виберіть будь-який варіант, оскільки VPCS може працювати в обох середовищах.



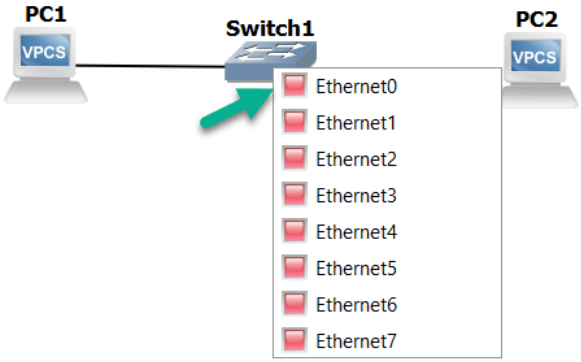
Додайте ще один вузол VPCS у робочу область, що створить другий вузол під назвою PC-2.

Щоб згорнути панель Browse End Devices, натисніть Browse End Devices знову або натисніть X.

Панель зведення топології відображає, що тепер у робочій області три вузли: Ethernetswitch-1, PC-1 і PC-2. Щоб додати зв'язки до топології, натисніть кнопку Add a Link. Курсор зміниться, що свідчить про можливість додавання зв'язків:



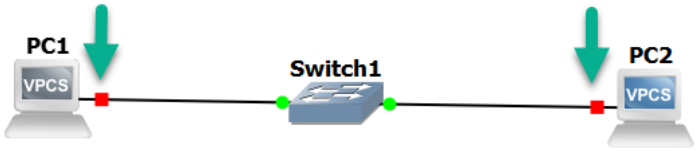
Натисніть на PC-1, щоб показати доступні інтерфейси. У цьому прикладі доступний лише Ethernet0 (залежить від пристрою). Натисніть Ethernet0 на PC-1, а потім клацніть на Ethernetswitch-1:



Виберіть Ethernet0 на Ethernetswitch-1, щоб завершити підключення. Створіть зв'язок між Ethernetswitch-1 і PC-2, натиснувши на будь-який із вузлів і обравши інтерфейс.

Клацніть Add a Link, щоб припинити додавання зв'язків, курсор зміниться на звичайний.

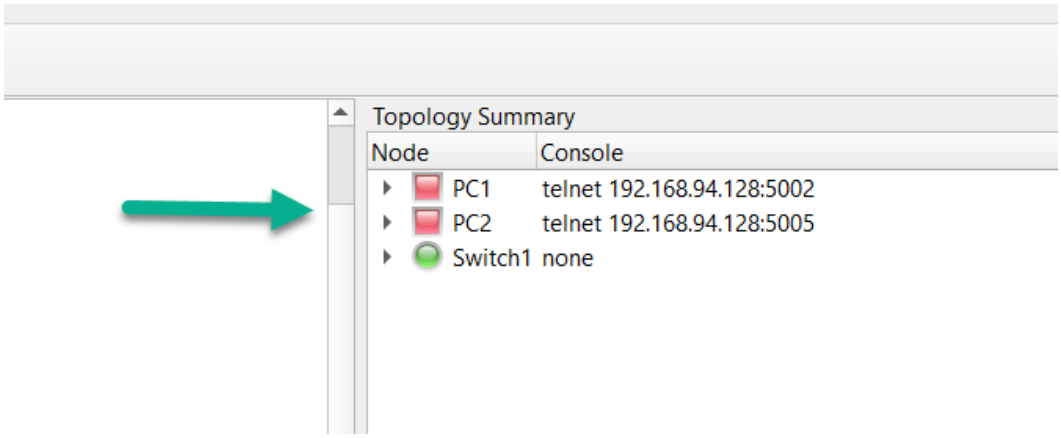
Поруч з пристроями ви побачите червоні індикатори, які вказують на те, що пристрої вимкнені:



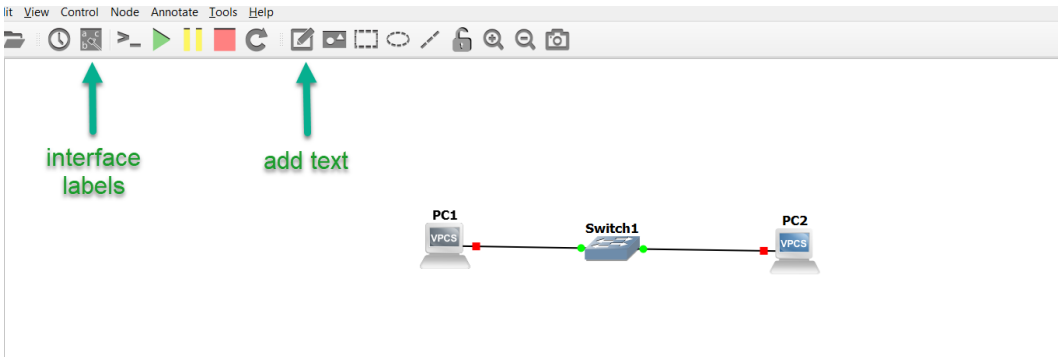
Notice how the link indicators on the PCs are red. Neither PC are currently running

Пристрої, які призупинено, мають жовті індикатори. Зелені індикатори показують, що пристрої увімкнені, навіть якщо їхні інтерфейси перебувають у стані down/down (наприклад, інтерфейси маршрутизатора/комутатора, які адміністративно вимкнено).

Стан увімкнення/вимкнення/призупинення також відображається у зведенні топології.

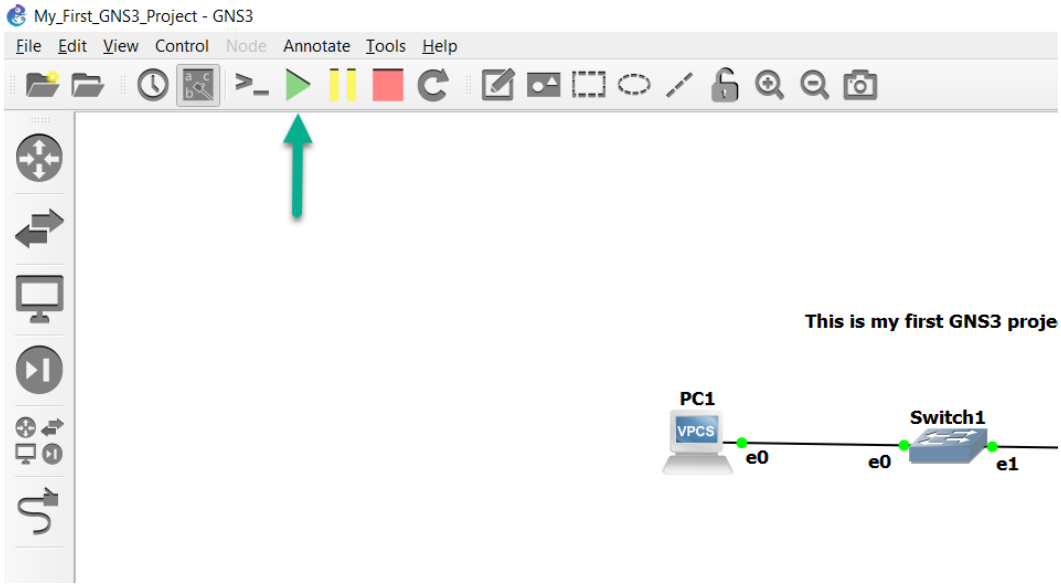


Корисно вмикати імена інтерфейсів і додавати примітки до топології, що полегшує перегляд підключень, а також додає позначення підмереж, IP-адреси, області OSPF, автономні системи BGP тощо. Ці дві кнопки на панелі інструментів дозволяють вмикати імена інтерфейсів та додавати примітки:

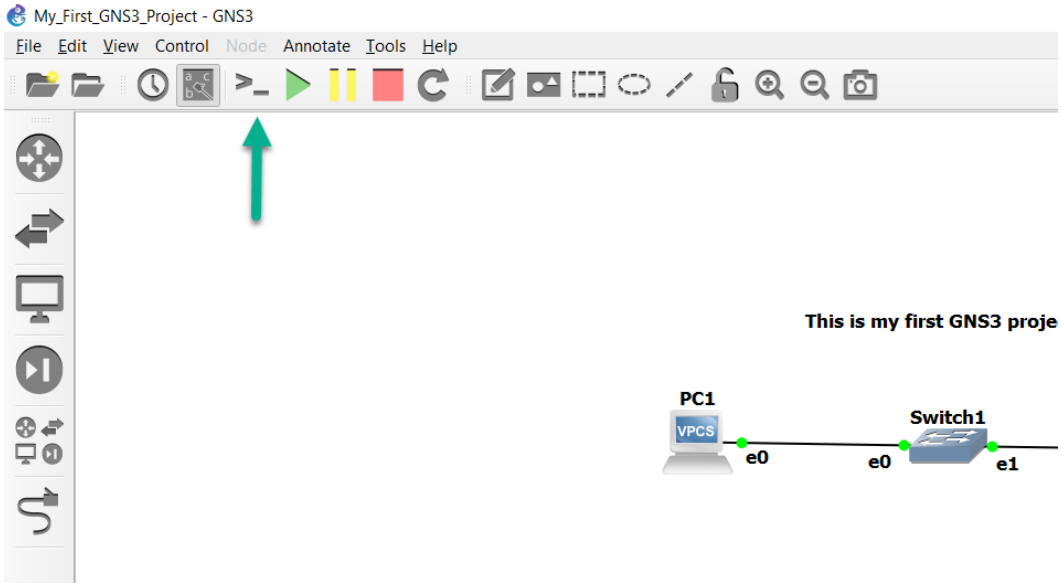


Нижче наведено приклад використання цієї функції. Інтерфейс e0 PC-1 підключений до e0 Ethernetswitch-1, а інтерфейс e0 PC-2 підключений до e1 Ethernetswitch-1 з простою приміткою.

Зелена кнопка «Відтворити» на панелі інструментів GNS3 вмикає всі пристрої у топології, жовта кнопка «Призупинити» призупиняє їх, а червона кнопка «Зупинити» вимикає всі пристрої:



Тепер можна налаштовувати пристрої. Натисніть кнопку Console connect to all devices на панелі інструментів GNS3, щоб відкрити з'єднання з усіма пристроями в топології:



За замовчуванням GNS3 використовуватиме Solar-PuTTY у Windows, оскільки він встановлюється як частина процесу інсталяції GNS3. Проте ви можете налаштувати GNS3 для використання інших емуляторів терміналу, таких як SecureCRT або Gnome-Term.

Піля успішного підключення до терміналу можна налаштовувати пристрої відповідно до вказаних задач.

Альтернативні варіанти

EVE-NG

Схожий на GNS3 програмний засіб, але при цьому має безкоштовну та комерційну версію. Функціонал безкоштовної обмежений. За деякими відгуками дане програмне забезпечення має кращий користувацький інтерфейс і є більш доопрацьованим. Але слід враховувати що принцип дії EVE-NG та GNS3 однаковий, тому кардинальної різниці в швидкодії бути не може.

Якщо ви приймете рішення використовувати даний варіант в якості альтернативи GNS3 то деталі та особливості використання ви можете знайти в документації розташованій за наступним посиланням www.eve-ng.net

Створення віртуальних машин вручну

Для уникнення подвійної віртуалцзації, при використанні GNS3 на операційній системі Windows, можна змоделювати аналогічну інфраструктуру створивши окремі віртуальні машини за допомогою будь-якого зручного для вас гіпервізора. Так наприклад VirtualBox дозволяє створювати будь-яку кількість внутрішніх мереж, а тому дає можливість емулювати мережі складної топології. Створити віртуальні машини можна як вручну так і скористатися можливостіми по інтеграції віртуальних машин VirtualBox та VMware в системі GNS3. Другий варіант спростить встановлення мережевих з'єднань.

Пошук вразливостей та сканування мереж

Nmap

Nmap ("Мережевий мапер") — це безкоштовний і відкритий інструмент для дослідження мережі та аудиту безпеки. Багато системних та мережевих адміністраторів також вважають його корисним для завдань, таких як інвентаризація мережі, керування розкладом оновлення послуг та моніторинг часу роботи хоста чи послуги. Nmap використовує необроблені пакети IP новими способами для визначення тих хостів, які доступні в мережі, які послуги (ім'я програми та версія) пропонують ці хости, які операційні систем (та версій операційних систем) вони використовують, які типи фільтрів пакетів/брандмауерів використовується та десятків інших характеристик. Його було розроблено для швидкого ска-

нування великих мереж, але добре працює проти окремих хостів. Nmap працює на всіх основних операційних системах комп'ютера, і доступні консольна та графічна версії [3].

Для реалізація такого широкого функціоналу а рамках консольної програми, її запуск передбачає велику кількість аргументів для гнучкого та детального керування пакетами, що надсилаються в мережу. Всі аргументи наводити в даному документі недоцільно, оскільки їх опис буде достатньо об'ємним, тим паче короткий опис кожного з аргументів можна отримати запустивши програму без аргументів або з аргументом **`--help`**. Також перелік основних аргументів доступний онлайн за посиланням Options Summary [4].

Увага

Виконувати тестування на наявність вразливостей, особливо такі, що передбачають активне втручання, можна тільки для ресурсів на тестування яких у вас є дозвіл, або спеціально призначених для тренування навичок з пошуку вразливостей ресурсів. Пам'ятайте, ***несанкціоноване втручання*** *в роботу комп'ютерів, комп'ютерних мереж чи мереж електрозв'язку є* ***кримінальним правопорушенням*** *(стаття 361 ККУ).*

Розглянемо деякі базові варіанти використання утиліти для сканування мереж. Почнемо з простого сканування:

```
nmap -sS <IP-address> # замість  <IP-address>
 ↪  підставляємо адресу мережі яку треба
 ↪  просканувати
#або
nmap -sT <IP-address>
```

Обидва варіанти сканування передбачають використання протокола TCP але з використанням різних флагів в пакетах. Дані варіанти сканування дозволяють виявити які вузли знаходяться онлайн в мережі. IP адресу можна вказувати як для одного вузла, наприклад **`10.0.0.2`**, так і для мережі, за допомогою маски **`10.0.0.2/24`** або діапазона **`10.0.0.2-42`**, використання доменних імен теж допустиме.

Час сканування залежить від того, наскільки великий діапазон ми намагаємось просканувати, тому бажано заздалегідь визначити приблизний діапазон адрес які треба сканувати. (Даний варіант сканування є достатньо швидким, у порівнянні із повним скануванням портів тому

допускає використання великих діапазонів адрес).

Для отримання інформації про операційну систему наявна наступна команда — її виконання сканує порти та намагається з деякими характерними ознаками визначити версії наявного мережевого програмного забезпечення, на основі чого приблизно визначає тип та версію операційної системи.

```
nmap -O <IP-address>
```

Детальну інформацію про операційну систему а також сканування портів виконує також і команда

```
nmap -A <IP-address>
```

Далі наведено перелік деяких варіантів команди що можна використовувати для сканування портів у визначеному діапазоні або виділених за іншим критерієм

```
nmap -p 1-100 <IP-адреса> #скану□мо порти в
 ↪  д□апазон□ в□д 1 до 100

nmap -sU <IP-адреса> #викону□мо сканування
 ↪  порт□в, як□ використовують UDP протокол

nmap -sV <IP-адреса> #викону□мо сканування
 ↪  порт□в та визначаэмо верс□□ програмного
 ↪  забезпечення, яке запущено

nmap -{}-packet-trace <IP-адреса> #виводимо
 ↪  докладний зв□т про кожен мережевий пакет,
 ↪  який був в□дправлений та отриманий п□д час
 ↪  сканування

nmap -oN <файл> -F -Pn <IP-адреса> #викону□мо
 ↪  швидке сканування порт□в (-F) та не
 ↪  перев□ря□мо доступн□сть хост□в (-Pn), а
 ↪  результати сканування збер□га□мо у файл□ з
 ↪  □менем <файл>
```

Більш детальні сценарії використання утиліти **Nmap** можна знайти в офіційній документації а також книзі «Nmap Network Scanning: Official Nmap Project Guide to Network Discovery and Security Scanning »[5], скорочена версія даної книги, яка не включає кілька розділів опублікована на офіційному сайті утиліти і доступна за посиланням nmap.org/book/toc.html.

Metasploit Framework

Фреймворк **Metasploit** — це платформа для проведення тестування на проникнення на основі мови програмування Ruby, яка складається з окремих модулів. Вона дозволяє користувачеві створювати, тестувати та виконувати код експлуатації наявних вразливостей. Фреймворк містить набір інструментів, за допомогою яких можна перевірити рівень безпеки системи, створити перелік мережі, виконати атаку та уникнути виявлення самої системи. У своєму основі фреймворк Metasploit складається з інструментів, які забезпечують повноцінне середовище для проведення тестування на проникнення та розробки експлуатаційних програм.

MSFconsole надає інтерфейс командного рядка для доступу та роботи з Metasploit Framework. MSFconsole є найчастіше використовуваним інтерфейсом для роботи з Metasploit Framework. Ця консоль дозволяє виконувати різні дії, такі як сканування цілей експлуатація вразливостей збір даних та багато іншого.

Відкрити Metasploit можна за допомогою команди **msfconsole**.

Щоб побачити допомогу під час роботи в консолі, використовуйте такі команди:

- Введіть **help**, щоб побачити список дійсних команд для поточного режиму. Коли ви перебуваєте в основному режимі, система відображає допомогу для глобальних команд, які доступні. Коли ви перебуваєте в режимі модуля, система відображає допомогу для команд і опцій, які доступні для модуля.

- Введіть **info <назва модуля>**, щоб побачити опції для модуля.

Metasploit будується навколо концепції модулів [6]. Найчастіше використовувані типи модулів включають:

- **Auxiliary** модулі — ці модулі не експлуатують цільовий об'єкт, але можуть виконувати збір даних або адміністративні завдання

- **Exploit** модулі — ці модулі використовують уразливості для виконання довільного коду на цільовому хості

- **Payloads** — довільний код, який може бути виконаний на віддаленій цільовій машині для виконання завдань, таких як створення користувачів, відкриття терміналів тощо
- **Post** модулі — ці модулі використовуються після компрометації машини. Вони виконують корисні завдання, такі як збір, збір або перерахування даних з сесії.

Для пошуку модулів можна використовувати команду **search**.

```
msf6 > search type:auxiliary http html title
 ↪  tag

Matching Modules
================

   #  Name                                Disclosure
    ↪  Date  Rank    Check  Description
   -  ----
    ↪  ---------------  ----    -----
    ↪  -----------
   0  auxiliary/scanner/http/title
    ↪  normal  No     HTTP HTML Title Tag
    ↪  Content Grabber


Interact with a module by name or index. For
 ↪  example info 0, use 0 or use
 ↪  auxiliary/scanner/http/title

msf6 >
```

Ви можете використовувати модуль Metasploit, вказавши повну назву модуля. Промпт буде оновлений для вказівки активного модуля:

```
msf6 > use auxiliary/scanner/http/title
msf6 auxiliary(scanner/http/title) >
```

Auxiliary модулі не експлуатують цільовий об'єкт, але можуть виконувати збір даних або адміністративні завдання. Наприклад, модуль, який

витягує заголовок HTTP з сервера, може бути використаний для збору інформації про сервер. Такі модулі можуть бути корисними для проведення розвідки або збору інформації про цільовий об'єкт.

Кожний модуль пропонує налаштовувані опції, які можна переглянути за допомогою команди **show options** або з скороченням **options**. Ця команда відображає список усіх доступних опцій для модуля та їхніх поточних значень.

Щоб встановити опцію модуля, використовується команда**set**. Нам потрібно встановити опцію **RHOST** — яка представляє цільовий хост(и), проти яких буде запущений модуль:

```
msf6 auxiliary(scanner/http/title) > set
 ↪  RHOSTS google.com
RHOSTS => google.com
```

Команда **run** запустить модуль проти цільового хоста, відображаючи заголовок HTTP цільового хоста.

У Metasploit 6 додана підтримка запуску модулів з опціями, встановленими як частина команди run. Наприклад, встановлення як RHOSTS, так і активація функціональності HttpTrace:

```
msf6 auxiliary(scanner/http/title) > run
 ↪  rhosts=google.com httptrace=true
```

Це дозволяє встановлювати опції модуля та запускати модуль проти цільового хоста в одному рядку команди.

Модулі експлуатації вимагають вразливої цільової машини. Модулі експлуатації зазвичай мінімально вимагають установки наступних опцій:

- **RHOST** — Адреса віддаленої цільової машини
- **LHOST** — Адреса для прослуховування. Важливо: Це може потрібно встановити як IP-адресу tun0 або подібну, якщо ви підключаєтесь до своєї цільової машини через VPN
- **PAYLOAD** — Код, який буде виконаний після успішної експлуатації. Наприклад, створення користувача або сеансу Metasploit. Зазвичай цю опцію можна залишити за замовчуванням, але іноді вона може потребувати конфігурації.

Відповідно до вимог експлуатаційного модуля, потрібно встановити ці опції перед запуском модуля.

Більш детальні інструкції по використанню Metasploit framework та окремих його модулів можна знайти у офіційній документації за посиланням docs.metasploit.com/docs/using-metasploit, а детальну документацію з переліком усіх модулів можна знайти за посиланням docs.metasploit.com/docs/modules.html

Завдання (деталізовно)

Частина 1. Моделювання захищеної мережі

Перед виконанням лабораторної роботи придумайте предметну область для вирішення задач в якій буде моделюватись мережа. Враховуйте можливості вашого обладнання для запуску системи моделювання і за необхідності використовуйте тільки мінімально необхідний набір компонентів. Повноцінну схему мережі можна побудувати окремо та додати в звіт. *В мінімальному варіанті моделі повинен бути хочаб один маршрутизатор, мервер та мережевий екран (у вигляді окремого пристроя мережевий екран можна розмістити замість маршрутизатора, або використовувати інтегрований в маршрутизатор (наприклад для маршрутизаторів Mikrotik)*

1) Створіть три маршрутизатори *(або іншу кількість яка відповідатиме обраній вами предметній області та мережіщо вирішуватиме завдання для вашої предметної області і при цьому дозволятиме запускати модель на вашому комп'ютері)*, підключіть їх до різних мережевих сегментів, а потім налаштуйте протокол маршрутизації, наприклад (**OSPF** або **RIP**), між ними.

2) Додайте кілька клієнтів до мережі, *(хочаб одного повноцінного клієнта, який матиме графічний інтерфейс)*, а також мережевий сервіс *(наприклад веб сервер, SMB сервер, або інший сервіс який можна просканувати на вразливості)*та один або кілька пристроїв безпеки (мережевих екранів, систем виявлення вторгнень чи комплексних рішень, при цьому їх можна додавати або окремими пристроями, або інтегрувати в сервер у вигляді програмних модулів).

3) Також не забудьте додати клієнт Kali Linux який далі буде використовуватися для тестування мережі.

4) Виконайте тестування, щоб переконатися, що маршрутизація працює коректно і всі вузли мають доступ один до одного в рамках задуманої топології.

Зверніть увагу

Остаточний варіант виконання даної частини може варіюватися, (наприклад маршрутизатори різних виробників, різні протоколи маршрутизації, різні варіанти клієнтів та мережевих сервісів), в залежності від обраної предметної області, тому кількість та типи пристроїв, що повинні бути присутні в моделі наведена орієнтовно, щоб відобразити складність моделі.

Увага

Виконувати тестування на наявність вразливостей, особливо такі, що передбачають активне втручання, можна тільки для ресурсів на тестування яких у вас є дозвіл, або спеціально призначених для тренування навичок з пошуку вразливостей ресурсів. Пам'ятайте, ***несанкціоноване втручання*** *в роботу комп'ютерів, комп'ютерних мереж чи мереж електрозв'язку є* ***кримінальним правопорушенням*** *(стаття 361 ККУ).*

Частина 2. Оцінка рівня захищеності мережі та та впливу технічних засобів на захищеність

1) Вимкніть системи безпеки для змодельованої мережі (наприклад дозвольте на мережевому екрані всі з'єднання, а систему виявлення вторгнен вимкніть).

2) Запустіть **`Kali Linux`** в режимі live середовища (не обов'язково, можливий будь-який варіант запуску в залежності від топології) для виконання тестування на проникнення, щоб виявити потенційні вразливості в мережі.

3) Для дослідження подальших етапів виконання скористатись програмним засобом **`Wireshark`**, що дозволяє відслідковувати проходження мережевих пакетів. В рамках системи GNS3 **`Wireshark`** може відслідковувати будь-яке з'єднання (при цьому немає необхідності встановлювати **`Wireshark`** на віртуальні комп'ютери моделі, **`Wireshark`** запускається на хост системі через інтеріейс GNS3).

4) За допомогою програми **Nmap** спробувати дослідити структуру мережі та виконати сканування портів для пошуку наявних вебсервісів.

5) Далі користуючись **Nmap** спробувати виконати DoS атаку на виявлений сервіс перевантаживши його великою кількістю запитів. Зафіксуйте виконання за допомогою ***Wireshark***.

6) За допомогою пакету **Metasploit Framework** спробуйте знайти потенційні вразливості для виявлених сервісів. *Скоріш за все це не вдасться, оскільки в актуальних версіях програмного забезпечення широко відомі вразливості будуть закритими.* Наведіть знайдені варіанти вразливостей (навіть якщо вони вже не актуальні) у звіті.

7) За допомогою **Metasploit Framework** спробуйте виконати DoS атаку на тестовий сервіс.

8) Оцініть вплив змодельованих атак на роботу мережевих сервісів.

9) За наявності в мережі операційних систем для яких в **Metasploit Framework** наявні **Payloads** можете спробувати в якості додаткового завдання (*таким чином можете підвищити оцінку за лабораторну роботу*) створити шкідливе програмне забезпечення для віддаленого доступу або віддаленого виконання команд та оцінити можливості антивірусного ПЗ якщо воно наявне для такої операційної системи .

10) Спробуйте налаштувати наявні в мережі засоби безпеки для мінімізації впливу проведених раніше атак та максимально обмежте можливості розкриття зловмисником інформації через сканування.

11) Повторіть спроби сканування та атак, оцініть ефективність роботи систем захисту.

Орієнтовні теми для теоретичних питань

- Для яких задач використовується моделювання комп'ютерних мереж
- Що дозволяє і що не дозволяє моделювати GNS3
- Функціонал основних засобів мережевої безпеки (Мережевого екрана, Системи виявлення вторгнень, Антивіруса)

- Поняття експлоїта
- Призначення та функціонал Nmap
- Metasploit Framework — призначення та особливості роботи.

Лабораторна робота № 3 Пошук вразливостей Веб-застосунків

Мета роботи

Мета лабораторної роботи полягає у вивченні та застосуванні методів перевірки безпеки веб-застосунків з метою виявлення потенційних вразливостей для підвищення рівня захисту інформаційних систем. Під час виконання роботи студенти повинні оволодіти основними техніками сканування і перевірки веб-застосунків та виявлення вразливостей, зокрема використовувати спеціалізовані інструменти для їх пошуку, аналізувати код веб-застосунків для виявлення потенційних вразливостей та оцінювати наслідки їх використання.

Завдання

Виявити якомога більше вразливостей в спеціальних тестових веб-застосунках.

Теоретичні відомості

Веб-додатки — це програми, написані скриптовою мовою (Perl, PHP, Javascript, Ruby, Golang та інші) або написані мовою високого рівня та відкомпільовані під відповідну ОС , які працюють на стороні веб-сервера та призначені для створення інтерфейсу між користувачем та веб-сайтом.

Результатом пошуку рішення, як ефективно запобігати новішим типам кібератак стало PEN-тестування. Якщо говорити узагальнено, то в

основі цього методу проста гіпотеза, щоб ефективно вдосконалювати системи та способи захисту продуктів від кібератак, необхідно розуміти, яким чином шукають ту чи іншу вразливість, і як вона влаштована зсередини. Іншими словами — щоб захиститися від хакерської атаки, потрібно усвідомлювати, за якими сценаріями вона може відбутися й де шукати головні слабкості системи. Саме завдяки активному аналізу системи на вміст дефектів коду чи в площині безпекових рішень можна виявити потенційні вразливості. Важливо, що PEN-тестування завжди проводиться з позиції потенційного нападника і може включати активне використання вразливостей. Тобто це контрольоване проникнення в систему [7]. Деякі з потенційних вразливостей наведено нижче.

- **XSS** (або ще міжсайтовий скриптинг) ін'єкції — досить поширена вразливість, яка зустрічається у багатьох застосунках. Головна ідея XSS в тому, що зловмиснику вдається додати на сторінку JavaScript-код, якого до цього не було. Цей код буде виконуватися щоразу, коли жертви (тобто користувачі) заходитимуть на сторінку застосунку, де цей код додав зловмисник.

- **SQL ін'єкція** (SQLi) — це тип ін'єкційної атаки, яка дозволяє модифікувати SQL команди для отримання даних або виведення з ладу програми. Зловмисники можуть модифікувати команди SQL, які впливають на ваш застосунок, через деякі вхідні дані на вашому сайті, наприклад, поле пошуку.Успішне виконання SQL ін'єкції може призвести до неавторизованого доступу до конфіденційних даних (це може бути дуже «чутлива» інформація — наприклад, паролі, адреси, дані кредитних карток і так далі). За останні кілька років багато випадків витоку інформації стали результатом саме SQL ін'єкцій.

- **Крос-сайтова підробка запиту** (CSRF) — це атака, коли зловмисник змушує користувача виконувати небажані дії без його згоди. Захист від CSRF включає використання токенів запитів (CSRF-токени) і перевірку Referer-заголовка [8].

- **Недоліки автентифікації та керування сеансами**. Слабка автентифікація та керування сеансами можуть призвести до компрометації облікових записів користувачів. Для захисту слід використовувати сильні паролі, двофакторну автентифікацію та надійне керування сеансами.

Увага

Виконувати тестування на наявність вразливостей, особливо такі, що передбачають активне втручання, можна тільки для ресурсів на тестування яких у вас є дозвіл, або спеціально призначених для тренування навичок з пошуку вразливостей ресурсів. Пам'ятайте, ***несанкціоноване втручання*** *в роботу комп'ютерів, комп'ютерних мереж чи мереж електрозв'язку є* ***кримінальним правопорушенням*** *(стаття 361 ККУ).*

Короткий відеокурс по пошуку вразливостей ви можете знайти за наступним посиланням на Youtube HackerOne. Для виконання лабораторної роботи бажано переглянути перші 11 відео, відео на тему криптографії, android та інші теми, що не стосуються веб-застосунків дивитись не обов'язково, оскільки вони не стосуються тематики даної роботи.

Для аналізу Веб-застосунків вам знадобиться програмне забезпечення для перехоплення запитів до веб сервера та їх модифікації. Одним із типів такого програмного забезпечення є перехоплюючі проксі сервери. Найбільш популярним варіантом такого програмного забезпечення є Burp Proxy. Повноцінна версія даної програми, яка містить всі засоби автоматизації та додаткові можливості є комерційною, але наявна і community версія яка є безкоштовною і містить тільки базовий функціонал, але його в більшості випадків достатньо для пошуку більшості вразливостей і цілком достатньо для виконання лабораторної роботи.

Іншим варіантом програмного забезпечення перехоплюючого проксі є ZAP Proxy, дана програма також э безкоштовною, і не зважаючи на обмежений у порівнянні з Burp функціонал цілком підходить для задач пошуку вразливостей.

Завдання (деталізовано)

В рамках даної лабораторної роботи вам пропонується виконати пошук вразливостей в веб-застосунках в форматі гри CTF.

CTF (Capture The Flag) — це тип змагання з безпеки інформаційних систем, яке полягає в перевірці безпеки вебзастосунків шляхом пошуків та експлуатації вразливостей.

Основна суть гри CTF полягає в наступному:

Мета гри: учасники гри мають завдання знайти та експлуатувати вразливості в вебзастосунках, щоб здобути доступ до захищеної інформації або виконати певні дії.

Учасники гри отримують інформацію про вебзастосунок, який потрібно перевірити на безпеку. Далі учасники грають роль зловмисників і намагаються знайти вразливості в вебзастосунку. Після цього учасники намагаються експлуатувати знайдені вразливості, щоб здобути доступ до захищеної інформації або виконати певні дії, при цьому учасники отримують очки за кожну знайдену вразливість та успішну експлуатацію.

1) Перейдіть на сайт Hacker 101 в розділ CTF за посиланням ctf.hacker101.com, та перейшовши в розділ для реєстрації зареєструватись на сайті.

2) Повернутись на сторінку CTF. Серед запропоноаних завдань обрати завдання із категорії **«Web»** відповідно до бажаної оцінки (див. Таблицю 1). При виборі завдання для вас буде створено персональну віртуальну машину з розгорнутим веб-застосунком який ви можете тестувати будь якими доступними способами. Завдання мають різні рівні складності, а також містять різну кількість флагів. В рамках лабораторної роботи не обов'язково шукати усі флаги з конкретного завдання, можете набрати необхідну кількість флагів з різних завдань.

3) Будь якими доступними засобами виявіть наявні у веб застосунках вразливості. При виявленні передбаченої завданням вразливості ви отримаєте флаг, у вигляді спеціального хеш значення вигляду *^FLAG^37ae56836 2f974017 fa575f08 cd215044cd 6bb395c3f 5e5e293 ee5324ba 6769c$FLAG$*, який треба внести на сайті в розділі **«Submit Flag»**. Після успішного занесення флага вам буде зараховано частину виконаного завдання та буде нараховано відповідну кількість балів.

4) На сторінці завдання також наявні підказки які можуть направити в пошуку флага, якщо зайшли в глухий кут і не знаєте де шукати вразливість — скористайтесь підказками. підказки відкриваються поступово і з затримкою в часі, щоб дати вам час обдумати підказку.

Оцінка	**Мінімальна кількість флагів***	**Додаткові умови**
Задовільно	5	Додаткових умов немає
Добре	10	Не менше 3 флагів на завданнях складності **Moderate**
Відмінно	14	Умови для оцінки «Добре»+ не менше одного флага складності **Hard**

Табл. 1. Завдання в залежності від очікуваної оцінки

Зверніть увагу

* — більше флагів знайдено - вище оцінка, але враховуйте, що на оцінку впливають також і відповіді на теоретичні питання.
Також лабораторна робота включає проходження короткого **тесту** на знання теорії в системі Moodle, **результат якого впливає на загальну оцінку** за лабораторну роботу

Також враховуйте, що не дивлячись на те, що виконувати завдання та вводити знайдені флаги можна в будь-якому порядку, порядок виведення підказок фіксований і зміні не підлягає, тому скориставшись підказками, виконувати завдання доведеться у описаному розробниками порядку.

Орієнтовні теми для теоретичних питань

- Загальне поняття «Вразливості». Причини виникнення вразливостей.
- Основні види вразливостей Web-застосунків, їх особливості.
- Шляхи виявлення вразливостей.
- Інструменти для пошуку вразливостей.

Розрахунково-графічна робота Проектування захищеної мережі підприємства

Мета роботи

Мета розрахунково-графічної роботи полягає у виконанні повного циклу проектування мережі підприємства, починаючи з визначення основних вимог і закінчуючи підбором реальних зразків мережевого обладнання та обладнання для захисту від кіберзагроз. Під час виконання роботи студенти повинні оволодіти навичками оцінки та розрахунку основних параметрів мережі з метою підбору правильної топології а також обладнання і програмного забезпечення необхідного для безпечного та безперебійного функціонування мережі.

Завдання

Зверніть увагу

Розрахунково-графічна робота може бути виконана як індивідуально так і ***невеликими*** *групами* ***до 3 людей*** *включно, але слід враховувати, що при груповому виконанні складність мережі повинна бути більшою аніж для індивідуального виконання. Також у випадку групового виконання потрібно чітко вказати внесок кожного з виконавців в загальний результат.*

Вибір предметної області та визначення основних вимог до мережі

Завдання даної розрахунково графічної роботи передбачає самостійний вибір предметної області для якої буде проектуватись комп'ютерна мережа, тому перший етап виконання РГР передбачає визначення підприємства або організації для яких буде проектуватись мережа. До даного етапу слід поставитись найбільш відповідально, оскільки початково обрані умови вплинуть на усі подальші етапи. Зокрема треба врахувати наступні моменти:

- Кількість користувачів (або співробітників) які користуються мережевими ресурсами
- Для вирішення яких задач буде використовуватись мережа
- Які ресурси повинна надавати мережа для вирішення задачі
- Які типи даних, які протоколи будуть використовуватись в мережі *(поки приблизно, далі, на етапі розрахунків це це питання буде деталізовано)*
- Мережа тільки для співробітників чи і для гостей
- Які засоби безпеки необхідні для захисту такої мережі *(На даному етапі за типами, виходячи із визначених мережевих сервісів, не забувайте, що до засобів безпеки, окрім мережевих екранів і антивірусів можна віднести і не пов'язані напряму з безпекою засоби, такі як засоби резервного копіювання та відновлення, що також становить основу безпечного та надійного функцінування інформаційної інфраструктури)*

По кожному з пунктів детально опишіть вашу мережу, але при цьому слідкуйте щоб понкти не протиречили один одному, описуючи цілісну мережу призначену для вирішення обраної задачі.

Планування топології мережі

Далі на основі визначених на попередньому етапі вимог до мережі вам необхідно спроектувати топологію. При цьому в першу чергу враховуйте вимог безпеки, що топологія давала можливість дотримуватись

основних принципів безпеки які актуальні для даних із обраної предметної області. Так мережа може ділитись на окремі зони з різними правами доступу, розподіліть робочі місця по мережі і визначте які типи обладнання будуть використовуватись та ін. На цьому етапі маємо попередній варіант топології який може бути скоригований на наступних етапах. Схему можете побудувати в **PacketTracer** або **GNS3**, при цьому налаштування мережевого обладнання можна не виконувати, достатньо зробити текстових описів основних параметрів.

Розрахунок параметрів обладнання

На даному етапі розрахунки варто почати з оцінки максимального потоку трафіку на робочих місцях та серверах, спираючись на контекст використання тих чи інших пристроїв а також часовий діапазон коли пристрої будуть використовуватись. Врахуйте наявність зовнішнього та внутрішнього трафіка, оскільки шляхи маршрутизації для низх часто відрізняються. Розрахувавши максимальний потік для конкретних кінцевих пристроїв розрахуйте навантаження на комутаційне обладнання та канали зв'язку що приходять до комутаторів, визначте мінімально необхідну ширину каналу що забезпечить стабільну роботу мережі.

*Наприклад, якщо у нас наявно 8 комп'ютерів і для роботи їм необхідно не більше 10Мбіт/с швидкості з'єднання, то для окремих пристроїв приєдднаних до комутатора достатньо 10Мбіт/с підключень, а підключення до маршрутизатора буде 8*10 = 80Мбіт/с, отже для такого потоку трафіка достатньо комутатора 100Мбіт/с. Зверніть увагу, що треба враховувати мінімально необхідну для комфортної роботи при максимальному навантаженні ширину каналу — це буде найгірший випадок коли всі одночасно активно використовують мережу, в реальних ситуаціях частіше за все буде доступна більша швидкість.*

Аналогічно треба розрахувати швидкість з'єднання для іншого обладнання. Якщо розрахована швидкість не підпадає під реально існуючі параметри можна трохи змінити топологію , наприклад використовувати два комутатора замість одного, щоб не використовувати обладнання із занадто великим запасом швидкості. Для серверів які доступні через інтернет врахуйте також зовнішній трафік, який приходить від клієнтів з глобальної мережі.

Також на даному етапі бажано розрахувати об'єми даних, що накопичуються задень, тиждень і за місяць, з метою розрахунку необхідного дискового простору для резервного копіювання. Також на цьому етапі треба визначити політики резервного копіюювання, якщо таке перед-

бачене для певних вузлів мережі, та розрахувати дисковий простір для серера резервних копій.

Всі розрахунки детально аргументуйте, та бажано розпишіть формули за якими велись розрахунки.

Вибір обладнання

На даному етапі на основі попередніх етапів запропонуйте варіанти обладнання, що відповідає отриманим вимогам. Намагайтесь використовувати актуальні на даний момент зразки обладнання та враховуйте що обладнання одного виробника краще сумісне муж собою аніж пристрої різних виробників. Особливу увагу зверніть на пристрої безпеки.

В якості пристроя безпеки може виступати і X86 платформа з OpenSource або комерційним програмним забезпеченням мережевого екрана та системи виявлення вторгнень. А можна використати і готові UTM рішення. робіть вибір виходячи з вимог до пропускної здатності.

Обрані варіанти мають бути обґрунтованими. Програмне забезпечення що стосується функціоналу безпеки та резервного копіювання, а також операційні системи для серверного обладнання теж варто розписати на даному етапі.

За необхідності ще раз внести корективи в топологію і оформити фінальний перелік обладнання у вигляді таблиці.

Бібліографія

[1] OMNeT. Дискретно-подійне моделювання — Вікіпедія — uk.wikipedia.org. https://uk.wikipedia.org/wiki/%D0%94%D0%B8%D1%81%D0%BA%D1%80%D0%B5%D1%82%D0%BD%D0%BE-%D0%BF%D0%BE%D0%B4%D1%96%D0%B9%D0%BD%D0%B5_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8E%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F. [Accessed 09-10-2024].

[2] Applications; INET 4.5.0 documentation — inet.omnetpp.org. https://inet.omnetpp.org/docs/users-guide/ch-apps.html. [Accessed 09-10-2024].

[3] Chapter 1.0;Getting Started with Nmap | Nmap Network Scanning — nmap.org. https://nmap.org/book/intro.html, 2009. [Accessed 04-11-2024].

[4] Options Summary | Nmap Network Scanning — nmap.org. https://nmap.org/book/man-briefoptions.html. [Accessed 04-11-2024].

[5] Gordon Lyon and Fyodor. *Nmap network scanning*. Nmap Project, January 2009.

[6] Running modules — docs.metasploit.com. https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html. [Accessed 05-11-2024].

[7] Найвідоміші вразливості веб застосунків. XSS та SQL ін'єкції, вразливості автентифікації — dou.ua. https://dou.ua/forums/topic/40613/. [Accessed 11-10-2024].

[8] Itproger. Безпека веб-додатків: найкращі практики та вразливості - стаття на itProger — itproger.com. https://itproger.com/ua/news/bezopasnost-veb-prilozheniy-luchshie-praktiki-i-uyazvimost [Accessed 11-10-2024].