Міністерство освіти і науки України
Центральноукраїнський національний технічний університет
Механіко-технологічний факультет
Кафедра кібербезпеки та програмного забезпечення

МЕТОДИЧНІ РЕКОМЕНДАЦІЇ

до виконання лабораторних робіт з навчальної дисципліни

«АДМІНІСТРУВАННЯ ІНФОРМАЦІЙНО-ТЕЛЕКОМУНІКАЦІЙНИХ СИСТЕМ»

для студентів денної форми навчання

галузі Інформаційні технології.

ЗАТВЕРДЖЕНО
на засіданні кафедри кібербезпеки та програмного забезпечення, протокол № 1 від 26.08.2025 року

Кропивницький
2025

Адміністрування інформаційно-телекомунікаційних систем: методичні рекомендації до виконання лабораторних робіт для студентів денної форми навчання галузі 12 «Інформаційні технології» / М-во освіти і науки України, Центральноукр. нац. техн. ун-т; [уклад. О.В. Коваленко, А.С. Коваленко] – Кропивницький: ЦНТУ, 2025. – 67 с.

Укладачі:

Коваленко О.В., докт. техн. наук, доц;

Коваленко А.С. канд. техн. наук, доц;

Рецензенти: Смірнов О. А., докт. техн. наук, професор, завідувач кафедри;

Якименко Н.М., к. ф.-м. наук, доцент.

ЗМІСТ

ВСТУП

Метою викладання дисципліни «Адміністрування інформаційно-телекомунікаційних систем» є забезпечення здобувачів вищої освіти комплексом знань, умінь та навичок, необхідних для застосування в професійній діяльності у сервісних та продуктових компаніях апаратного апаратно-програмного уклону галузі Інформаційні технології.

Навчальний курс «Адміністрування інформаційно-телекомунікаційних систем» призначений для набуття теоретичних знань та комплексу навичок, які відповідають за успішне штатне відновлення функціонування програмно-технічних систем та технологій з можливістю розробляти системне та прикладне програмне забезпечення на основі отриманих теоретичних засад, архітектурної будови та роботи операційної системи Windows у відповідності до існуючих обмежень.

Лекційні заняття проводяться в аудиторіях обладнаних мультимедійним проектором. Лабораторні роботи виконуються у аудиторіях кафедри кібербезпеки та програмного забезпечення, обладнаних відповідним апаратним та програмним забезпеченням (ауд 501, 507, 508, 517), з відкритою бездротовою мережею Wi-Fi, вільним доступом до Інтернету. Оскільки при вивченні дисципліни використовуються інформаційні технології навчання, система дистанційної освіти Moodle, студенту необхідно мати комп’ютерну техніку (з виходом у Internet) та оргтехніку для комунікації з викладачами, виконання тестових завдань в системі дистанційної освіти.

Лабораторне заняття – форма навчального заняття, спрямована на закріплення та вдосконалення студентом теоретичних знань, отриманих як на лекційних і практичних заняттях, так і в процесі самостійного вивчення матеріалу.

Провідна форма навчання – лекція. Лекція дозволяє дуже економно, з мінімальними затратами часу і викладача, і студентів, надати великий обсяг

інформації по темі, що розглядається. За характером логіки пізнання впроваджуються аналітичний, індуктивний та дедуктивний методи.

Супровідні методи – лабораторні роботи.

Основна дидактична мета практичного заняття – закріплення й деталізація знань, а головне – формування навичок і вмінь. Для проведення практичного заняття викладач готує відповідні методичні матеріали: тести для виявлення рівня оволодіння необхідними теоретичними положеннями ; набір практичних завдань різної складності для розв'язування їх на занятті та дидактичні засоби.

Під час лабораторного заняття студенти під керівництвом викладача набувають практичних навичок у роботі з обчислювальною технікою, оволодівають методикою створення програмних продуктів у програмному середовищі. При цьому у студентів формуються вміння й практичні навички використання різних програмних засобів ПК для розв’язання конкретних економічних задач відповідно до індивідуального завдання.

Проведення лабораторних занять ґрунтується на попередньо підготовлених методичних матеріалах: визначення підготовленості студентів до виконання завдань лабораторного заняття на основі тестового контролю знань основних положень теорії досліджуваної теми, усного контролю виконання домашнього завдання, пов'язаного з розробкою макетів документів, які необхідно розробити програмно під час заняття.

Індивідуальні завдання до кожної лабораторної роботи мають чітко виражену прикладну спрямованість, що враховує профіль підготовки студентів, тобто охоплюють питання автоматизації рішення різних завдань економіки і підприємництва.

На лабораторному занятті студенти під керівництвом викладача проводять розробку ПЗ в навчальних лабораторіях з використанням комп'ютерної техніки. Основною метою лабораторного заняття є практичне підтвердження окремих теоретичних положень та набуття практичних вмінь з виконання обчислювальних експериментів.

Головна особливість цих занять полягає у тому, що вони об'єднують теорію з практикою, забезпечують їх єдність. Сукупність лабораторних занять з дисципліни є лабораторним практикумом, що сплановане за єдиним задумом. Лабораторні заняття плануються після проведення лекцій. А при необхідності розробки програм, проектування баз даних або підготовки складних розрахунків і початкових даних перед лабораторними заняттями проводяться консультації.

Лабораторні роботи виконуються у такій послідовності:

– вивчення навчального матеріалу з теми лабораторної роботи з використанням конспекту лекцій, рекомендованих підручників і навчальних посібників;

– самостійна підготовка студентами макетів інтерфейсів програм, які мають бути практично створені на занятті;

– виконання завдання на ПК відповідно до виданого варіанта й подання результатів викладачеві.

По завершенню кожної роботи студенти готують і оформлюють звіт й захищають отримані результати.

Звіт повинен містити:

– тему й мету роботи;

– зміст завдання й короткий опис порядку його виконання;

– аналіз отриманих результатів та висновки роздруківку основних результатів виконання індивідуального завдання.

Напередодні проведення кожного лабораторного заняття (після відповідної лекції) студентам видається завдання, що містить: тему і мету заняття; скорочені теоретичні відомості щодо змісту лабораторного заняття; список питань для підготовки (це можуть бути контрольні питання по темі, що вивчається, заповнення роздатних матеріалів індивідуальними даними, розробка програм, таблиць і т.д.); послідовність підлягаючих виконанню на занятті дій (завдання на лабораторну роботу); вимоги до змісту звіту. Студент повинен вивчити навчальний матеріал, завдання, підготувати

необхідні для роботи на занятті матеріали і знать відповіді на контрольні питання.

У ході підготовки може бути створена заготовка звіту, що дозволить заощадити час на занятті. Лабораторні заняття проводяться в аудиторіях, академічна група ділиться на підгрупи.

Усі лабораторні заняття з дисципліни проводяться фронтально, кожний студент працює за окремим комп'ютером. На початку заняття, після оголошення теми, цільової установки і коротких указівок щодо особливостей роботи викладачем проводиться контроль підготовленості студентів, звичайно, шляхом перевірки відповідей на контрольні питання (тестів), рідше, у формі усної бесіди по темі заняття.

Для контролю може використовуватися і тестування. Обов'язково перевіряється наявність матеріалів для виконання роботи (програм, роздаткового матеріалу з відпрацьованими індивідуальними питаннями, початкових даних для вирішення задач, заготовок звіту і т. п.).

За відсутності матеріалів, необхідних для виконання роботи, і знань, які не дозволяють виконати роботу, студент до роботи не допускається, і йому пропонується виконати необхідну підготовку. Сама робота повинна виконуватися у додатковий час. У ході заняття студенти самостійно виконують передбачені завданням дії, заносячи результати в звіт. На це відводиться до 85–90% часу заняття.

Викладач здійснює контроль за роботою і надає допомогу при виникненні ускладнень, звертає увагу на складні ключові моменти. Причому основну увагу приділяється не вказівці на конкретну помилку, а методиці пошуку причин виникнення цих помилок.

Складання звіту – це відповідальний етап лабораторного заняття. При його складанні студенти розвивають навики аналізу, узагальнення і творчого осмислення результатів роботи, а також навики розробки документації до програмного продукту. Необхідно прагнути до того, щоб студенти

оформляли звіт про виконану роботу і представили його викладачу до кінця лабораторної роботи.

Цьому сприяє наявність наперед підготовленої заготовки, в яку послідовно заносяться всі необхідні дані і зроблені висновки.

Звіт повинен бути представлений у вигляді електронного документа. За наслідками контролю готовності студентів до роботи, об'єму і правильності її виконання, повноти і якості оформлення звіту і його захисту, терміну захисту викладач виставляє оцінку.

Звіти, які не представлені під час заняття, захищаються в додатковий час. В окремих випадках оцінка може виставлятися за групу взаємопов'язаних робіт.

При оцінці лабораторної роботи викладач ураховує правильність та розуміння роботи розроблених програмних продуктів, уміння працювати у програмному середовищі. Оцінки за кожну лабораторну роботу вносяться у відповідний журнал.

Студент, що пропустив лабораторне заняття або не допущений до нього, зобов'язаний виконати відповідну роботу під час самостійної підготовки і відзвітувати. Повторна здача робіт, які не були прийняті, проводиться під час консультацій або під час наступних лабораторних занять.

Оцінки, отримані студентом за окремі лабораторні заняття враховуються при виставленні поточної модульної оцінки з навчальної дисципліни.

У процесі лабораторного заняття викладач організує такі види методичної роботи зі студентами:вирішення поточних запропонованих індивідуальних завдань на лабораторну роботу; перевірку завдань щодо розробки програм та алгоритмів; захист лабораторних робіт окремих студентів.

Перелік тем лабораторних занять наведено у табл. 1.

Таблиця 1 – Перелік тем лабораторних занять

№ з/п	Назва теми	Кількість годин	
		ден. форм. навч.	заоч. форм. нав.
5 СЕМЕСТР			
1.	**Тема 1.** Моніторинг файлової підсистеми інформаційно-телекомунікаційної системи в реальному часі з формуванням звітності для відновлення штатного функціонування.	2	0,2
2.	**Тема 2.** Моніторинг підсистеми реєстру інформаційно-телекомунікаційної системи в реальному часі з формуванням звітності для відновлення штатного функціонування.	2	0,2
3.	**Тема 3**. Моніторинг протоколів прикладного рівня інформаційно-телекомунікаційної системи в реальному часі з формуванням звітності для відновлення штатного функціонування.	2	0,2
4.	**Тема 4**. Моніторинг протоколів мережного рівня інформаційно-телекомунікаційної системи в реальному часі з формуванням звітності для відновлення штатного функціонування.	2	0,2
5.	**Тема 5.** Моніторинг протоколів транспортного рівня інформаційно-телекомунікаційної системи в реальному часі з формуванням звітності для відновлення штатного функціонування.	4	0,3
6.	**Тема 6.** Моніторинг параметрів запуску інформаційно-телекомунікаційної системи в реальному часі з формуванням звітності для відновлення штатного функціонування.	2	0,2
7.	**Тема 7.** Моніторинг паралельних обчислень процесів інформаційно-телекомунікаційної системи в реальному часі з формуванням звітності для відновлення штатного функціонування.	2	0,7
	Усього годин	**16**	**2**
6 СЕМЕСТР			
8.	**Тема 1. Аналіз символьних даних двійкового коду ПЗ.** Отримати практичні навики використання методів та інструментів статичного аналізу даних.	2	0,2
9.	**Тема 2. Формат бінарних виконуваних файлів, об'єктного коду та динамічних бібліотек (dll).** Дослідження структури бінарних виконуваних файлів.	2	0,7
10.	**Тема 3. Інструменти аналізу поведінки підсистем ОС.** Отримати практичні навики використання методів та інструментів динамічного аналізу даних.	2	0,2
11.	**Тема 4. Аналіз дампів пам'яті операційних систем.** Отримати практичні навики використання інструментів аналізу пам’яті ОС. Використання командного підходу аналізу з документуванням спільного результату	2	0,2
12.	**Тема 5. Умови використання дизасемблерів.** Отримати практичні навики використання дизасемблерів для зворотного аналізу ПЗ.	4	0,3
13.	**Тема 6. Умови використання декомпіляторів.** Отримати практичні навики використання декомпіляторів для зворотного аналізу ПЗ.	2	0,2
14.	**Тема 7. Ідентифікація та класифікація зразків зловмисного ПЗ.** Отримати практичні навики реагування на інциденти та ліквідації наслідків, описувати роботу програмно-технічних засобів, працювати як індивідуально так і у складі команди.	2	0,2
	Усього годин	**16**	**2**

Лабораторна робота №1 (семестр 5)

ТЕМА: Моніторинг файлової підсистеми інформаційно-телекомунікаційної системи в реальному часі з формуванням звітності для відновлення штатного функціонування.

МЕТА: Отримати практичні навики використання систем моніторингу.

ЗНАТИ: Теоретичні основи архітектурної будови ОС Windows.

ВМІТИ: Встановлювати дистрибутиви технічних засобів та утиліт для управління, діагностики, усунення неполадок та моніторингу ОС Windows.

ТЕОРЕТИЧНІ ВІДОМОСТІ.

У зв'язку з великим обсягом інформації використовувати електронну документацію в залежності з обраним шляхом вирішення завдання (погоджувати з лектором).

Для вирішення пропонується використовувати наступну документацію та технічні засоби та утиліти для управління, діагностики, усунення неполадок та моніторингу:

- Sysinternals. https://docs.microsoft.com/en-us/sysinternals/
- Nirsoft https://www.nirsoft.net/utils/index.html
- Washington University Computer Science & Engineering https://www.cse.wustl.edu/~jain/cse567-06/ftp/os_monitors/index.html/
- Flexense Ltd SysGauge https://www.sysgauge.com/

ХІД ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

1. Визначити свій індивідуальний варіант завдання - відповідно до порядкового номеру студента у групі (погоджувати з лектором).

2. Запустити генератор подій “**AITS System EventsGenerator (1-7 lab).exe**” що зображено на рисунку 1.

3. Вибрати номер індивідуального варіанту (Select your variant) та номер лабораторної роботи (select task) у випадаючих списках як це показано на рисунку 2.

4. Запустити та налаштувати обраний засіб моніторингу та натиснути кнопку “Start”.

5. Сформувати звіт роботи генератора “**AITS System EventsGenerator (1-7 lab).exe**” та оформити звіт лабораторної роботи.

Звіт виконання лабораторної роботи повинен містити:

– Титульний лист.

– Тема та мета лабораторної роботи.

– Завдання до лабораторної роботи.

– Звіт моніторингу індивідуального варіанту завдання студента.

– Скріншоти та інші матеріали на думку автора які підтверджують правильність проведеного моніторингу.

– Відповіді на контрольні питання.

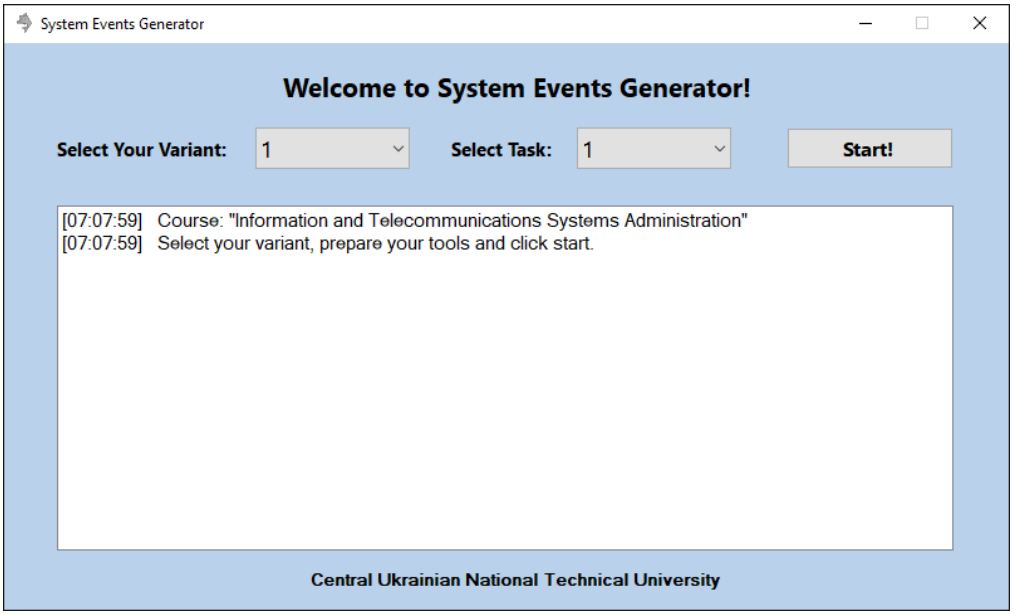

Рисунок 1 – Головне вікно генератора подій

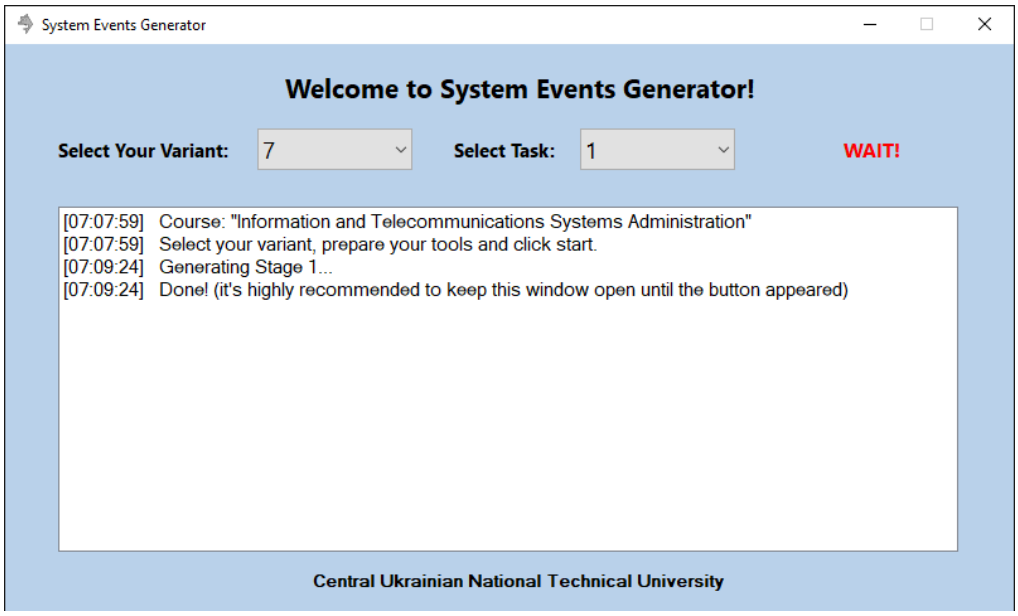

Рисунок 2 – Приклад встановлення та запуску генератора подій (Варіант 7, лабораторна робота 1)

КОНТРОЛЬНІ ЗАПИТАННЯ

1. Які бувають файлові системи та яка між ними різниця?
2. Які існують атрибути файлів та які основні шляхи їх зміни?
3. Наведіть приклади відомих файлових менеджерів та їх функціоналу.
4. Для чого використовуються Portable Executable файли?
5. Наведіть структуру та сигнатуру Portable Executable файлу.
6. Для чого використовується Common Language Runtime та як він розширює можливості PE формату.
7. Для чого використовується пакувальник виконуваних файлів?
8. Перерахуйте стандартні системні змінні середовища Windows 10 та Windows 11.

Лабораторна робота №2 (семестр 5)

ТЕМА: Моніторинг підсистеми реєстру інформаційно-телекомунікаційної системи в реальному часі з формуванням звітності для відновлення штатного функціонування.

МЕТА: Отримати практичні навики використання систем моніторингу.

ЗНАТИ: Теоретичні основи архітектурної будови ОС Windows.

ВМІТИ: Встановлювати дистрибутиви технічних засобів та утиліт для управління, діагностики, усунення неполадок та моніторингу ОС Windows.

ТЕОРЕТИЧНІ ВІДОМОСТІ.

У зв'язку з великим обсягом інформації використовувати електронну документацію в залежності з обраним шляхом вирішення завдання (погоджувати з лектором).

Для вирішення пропонується використовувати наступну документацію та технічні засоби та утиліти для управління, діагностики, усунення неполадок та моніторингу:

- Sysinternals. https://docs.microsoft.com/en-us/sysinternals/
- Nirsoft https://www.nirsoft.net/utils/index.html
- Washington University Computer Science & Engineering https://www.cse.wustl.edu/~jain/cse567-06/ftp/os_monitors/index.html/
- Flexense Ltd SysGauge https://www.sysgauge.com/

ХІД ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

1. Визначити свій індивідуальний варіант завдання - відповідно до порядкового номеру студента у групі (погоджувати з лектором).

2. Запустити генератор подій “**AITS System EventsGenerator (1-7 lab).exe**” що зображено на рисунку 1.

3. Вибрати номер індивідуального варіанту (Select your variant) та номер лабораторної роботи (select task) у випадаючих списках як це показано на рисунку 2.

4. Запустити та налаштувати обраний засіб моніторингу та натиснути кнопку “Start”.

5. Сформувати звіт роботи генератора “**AITS System EventsGenerator (1-7 lab).exe**” та оформити звіт лабораторної роботи.

Звіт виконання лабораторної роботи повинен містити:

– Титульний лист.

– Тема та мета лабораторної роботи.

– Завдання до лабораторної роботи.

– Звіт моніторингу індивідуального варіанту завдання студента.

– Скріншоти та інші матеріали на думку автора які підтверджують правильність проведеного моніторингу.

– Відповіді на контрольні питання.

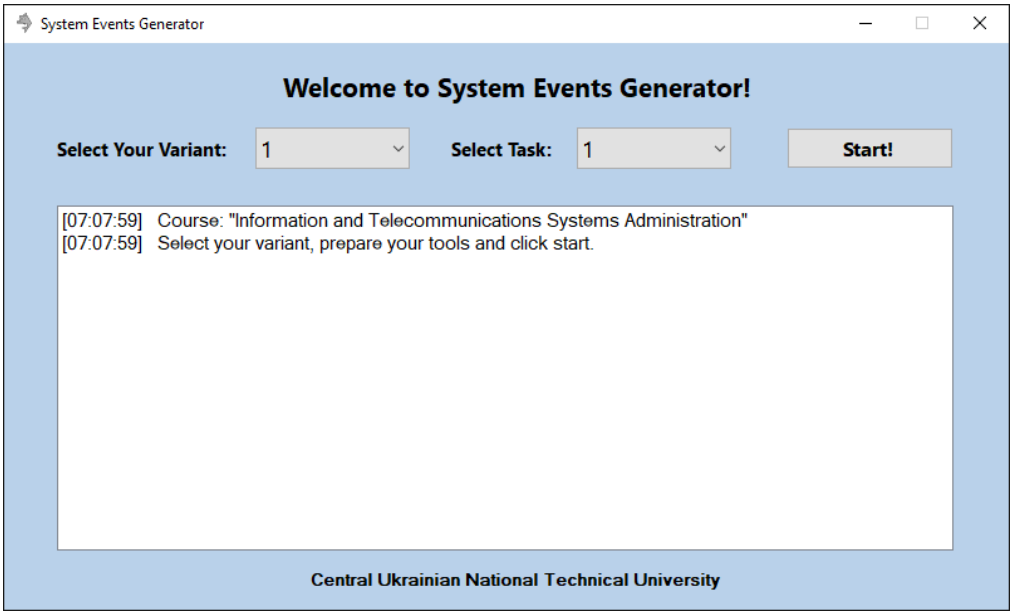

Рисунок 1 – Головне вікно генератора подій

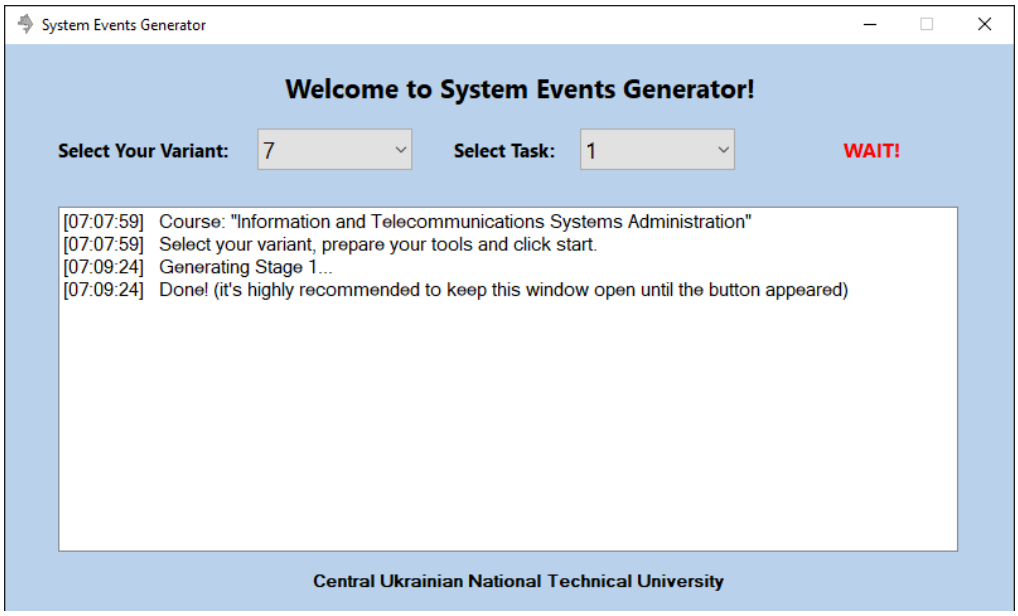

Рисунок 2 – Приклад встановлення та запуску генератора подій (Варіант 7, лабораторна робота 1)

КОНТРОЛЬНІ ЗАПИТАННЯ

1. Навіщо необхідно використовувати системний реєстр Windows?
2. Для чого використовуються ini файли?
3. Де фізично зберігається файли системного реєстру Windows?
4. Які існують розділи реєстру та для чого вони використовуються?
5. Перерахуйте існуючі типи даних та структури реєстру.
6. За необхідності як відновити ключі реєстру?
7. Наведіть приклади створення скрипта редагування даних реєстру.

Лабораторна робота №3 (семестр 5)

ТЕМА: Моніторинг протоколів прикладного рівня інформаційно-телекомунікаційної системи в реальному часі з формуванням звітності для відновлення штатного функціонування.

МЕТА: Отримати практичні навики використання систем мережного моніторингу протоколів прикладного рівня.

ЗНАТИ: Теоретичні основи базової еталонної моделі взаємодії відкритих систем – 7 рівень семирівневої моделі OSI.

ВМІТИ: Встановлювати дистрибутиви технічних засобів та утиліт для управління, діагностики, усунення неполадок та моніторингу ОС Windows.

ТЕОРЕТИЧНІ ВІДОМОСТІ.

У зв'язку з великим обсягом інформації використовувати електронну документацію в залежності з обраним шляхом вирішення завдання (погоджувати з лектором).

Для вирішення пропонується використовувати наступну документацію та технічні засоби та утиліти для управління, діагностики, усунення неполадок та моніторингу:

- Sysinternals. https://docs.microsoft.com/en-us/sysinternals/
- Nirsoft https://www.nirsoft.net/utils/index.html
- Washington University Computer Science & Engineering https://www.cse.wustl.edu/~jain/cse567-06/ftp/os_monitors/index.html/
- Flexense Ltd SysGauge https://www.sysgauge.com/

ХІД ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

1. Визначити свій індивідуальний варіант завдання - відповідно до порядкового номеру студента у групі (погоджувати з лектором).

2. Запустити генератор подій "**AITS System EventsGenerator (1-7 lab).exe**" що зображено на рисунку 1.

3. Вибрати номер індивідуального варіанту (Select your variant) та номер лабораторної роботи (select task) у випадаючих списках як це показано на рисунку 2.

4. Запустити та налаштувати обраний засіб моніторингу та натиснути кнопку "Start".

5. Сформувати звіт роботи генератора "**AITS System EventsGenerator (1-7 lab).exe**" та оформити звіт лабораторної роботи.

Звіт виконання лабораторної роботи повинен містити:

– Титульний лист.

– Тема та мета лабораторної роботи.

– Завдання до лабораторної роботи.

– Звіт моніторингу індивідуального варіанту завдання студента.

– Скріншоти та інші матеріали на думку автора які підтверджують правильність проведеного моніторингу.

– Відповіді на контрольні питання.

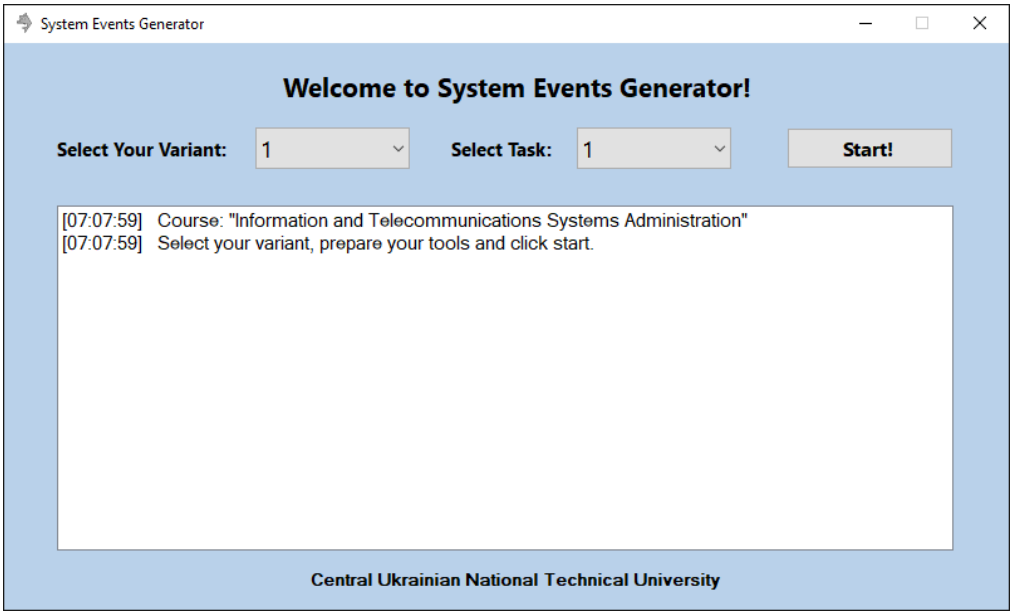

Рисунок 1 – Головне вікно генератора подій

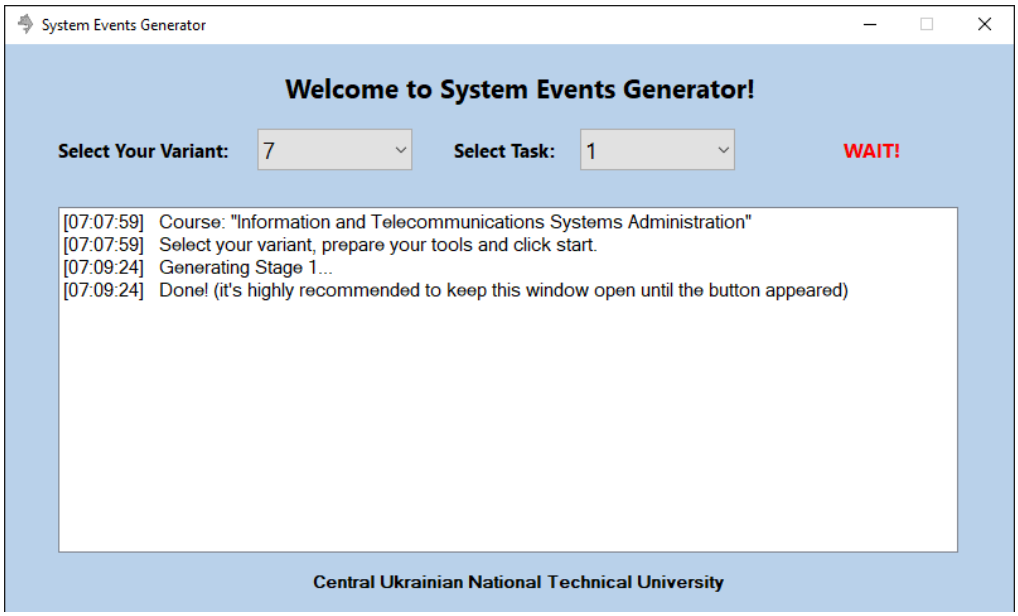

Рисунок 2 – Приклад встановлення та запуску генератора подій (Варіант 7, лабораторна робота 1)

КОНТРОЛЬНІ ЗАПИТАННЯ

1. Для чого використовують HyperText Transfer Protocol?
2. В чому відмінність HTTP від HTTPS?
3. Наведіть структуру протоколу HTTPS.
4. Наведіть існуючі методи запитів HTTP.
5. Я які групи формуються коди статусів HTTP?
6. Перерахуйте найбільш поширені коди статусів HTTP.
7. Наведіть приклад HTTP діалогу.
8. Для чого використовується метод GET HTTP?
9. Для чого використовується метод POST HTTP?

Лабораторна робота №4 (семестр 5)

ТЕМА: Моніторинг протоколів мережного рівня інформаційно-телекомунікаційної системи в реальному часі з формуванням звітності для відновлення штатного функціонування.

МЕТА: Отримати практичні навики використання систем мережного моніторингу протоколів мережного рівня.

ЗНАТИ: Теоретичні основи базової еталонної моделі взаємодії відкритих систем – 3 рівень семирівневої моделі OSI.

ВМІТИ: Встановлювати дистрибутиви технічних засобів та утиліт для управління, діагностики, усунення неполадок та моніторингу ОС Windows.

ТЕОРЕТИЧНІ ВІДОМОСТІ.

У зв'язку з великим обсягом інформації використовувати електронну документацію в залежності з обраним шляхом вирішення завдання (погоджувати з лектором).

Для вирішення пропонується використовувати наступну документацію та технічні засоби та утиліти для управління, діагностики, усунення неполадок та моніторингу:

- Sysinternals. https://docs.microsoft.com/en-us/sysinternals/
- Nirsoft https://www.nirsoft.net/utils/index.html
- Washington University Computer Science & Engineering https://www.cse.wustl.edu/~jain/cse567-06/ftp/os_monitors/index.html/
- Flexense Ltd SysGauge https://www.sysgauge.com/

ХІД ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

1. Визначити свій індивідуальний варіант завдання - відповідно до порядкового номеру студента у групі (погоджувати з лектором).

2. Запустити генератор подій “**AITS System EventsGenerator (1-7 lab).exe**” що зображено на рисунку 1.

3. Вибрати номер індивідуального варіанту (Select your variant) та номер лабораторної роботи (select task) у випадаючих списках як це показано на рисунку 2.

4. Запустити та налаштувати обраний засіб моніторингу та натиснути кнопку “Start”.

5. Сформувати звіт роботи генератора “**AITS System EventsGenerator (1-7 lab).exe**” та оформити звіт лабораторної роботи.

Звіт виконання лабораторної роботи повинен містити:

– Титульний лист.

– Тема та мета лабораторної роботи.

– Завдання до лабораторної роботи.

– Звіт моніторингу індивідуального варіанту завдання студента.

– Скріншоти та інші матеріали на думку автора які підтверджують правильність проведеного моніторингу.

– Відповіді на контрольні питання.

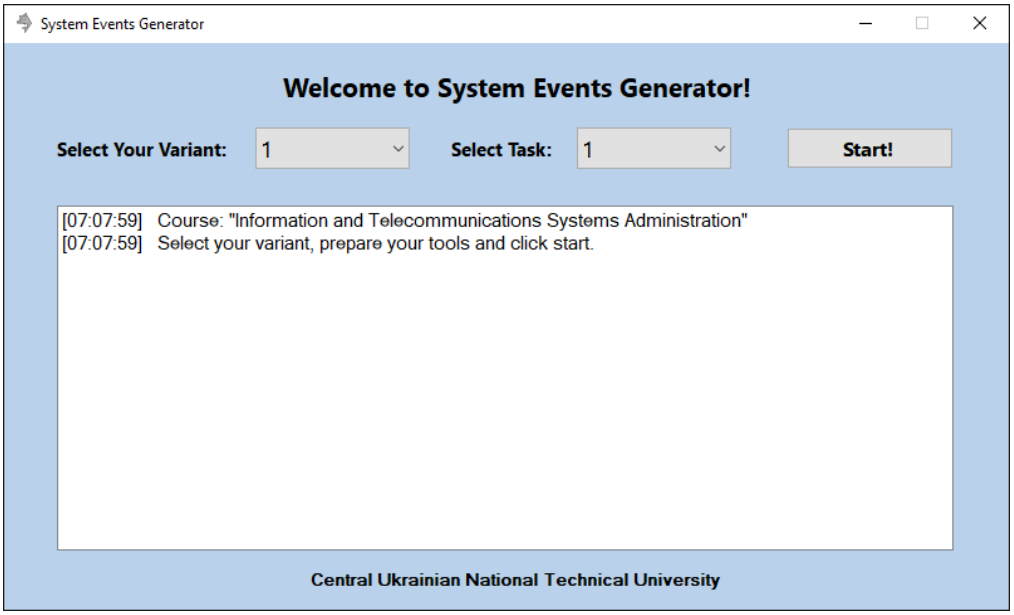

Рисунок 1 – Головне вікно генератора подій

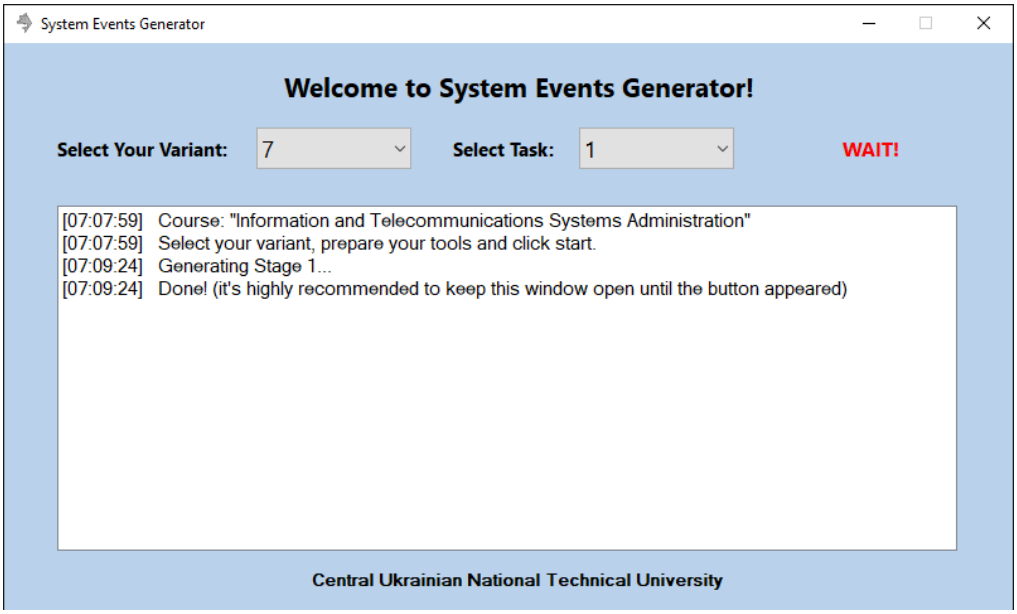

Рисунок 2 – Приклад встановлення та запуску генератора подій (Варіант 7, лабораторна робота 1)

КОНТРОЛЬНІ ЗАПИТАННЯ

1. Для чого використовують ECHO запити?
2. Яка відома мережна утиліта використовує ECHO запити?
3. Наведіть приклад роботи протоколу ICMP?
4. Яка використовується структура у ICMP пакеті?
5. Для чого використовується IPv4?
6. Наведіть приклад формування адреси на основі IPv4.
7. Для чого використовується IPv6?
8. Наведіть приклад формування адреси на основі IPv6.
9. Роз’ясніть термінологічне скорочення ICMP Header.
10. Роз’ясніть термінологічне скорочення ICMP Payload.

Лабораторна робота №5 (семестр 5)

ТЕМА: Моніторинг протоколів транспортного рівня інформаційно-телекомунікаційної системи в реальному часі з формуванням звітності для відновлення штатного функціонування.

МЕТА: Отримати практичні навики використання систем мережного моніторингу протоколів транспортного рівня.

ЗНАТИ: Теоретичні основи базової еталонної моделі взаємодії відкритих систем – 4 рівень семирівневої моделі OSI.

ВМІТИ: Встановлювати дистрибутиви технічних засобів та утиліт для управління, діагностики, усунення неполадок та моніторингу ОС Windows.

ТЕОРЕТИЧНІ ВІДОМОСТІ.

У зв'язку з великим обсягом інформації використовувати електронну документацію в залежності з обраним шляхом вирішення завдання (погоджувати з лектором).

Для вирішення пропонується використовувати наступну документацію та технічні засоби та утиліти для управління, діагностики, усунення неполадок та моніторингу:

- Sysinternals. https://docs.microsoft.com/en-us/sysinternals/
- Nirsoft https://www.nirsoft.net/utils/index.html
- Washington University Computer Science & Engineering https://www.cse.wustl.edu/~jain/cse567-06/ftp/os_monitors/index.html/
- Flexense Ltd SysGauge https://www.sysgauge.com/

ХІД ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

1. Визначити свій індивідуальний варіант завдання - відповідно до порядкового номеру студента у групі (погоджувати з лектором).

2. Запустити генератор подій "**AITS System EventsGenerator (1-7 lab).exe**" що зображено на рисунку 1.

3. Вибрати номер індивідуального варіанту (Select your variant) та номер лабораторної роботи (select task) у випадаючих списках як це показано на рисунку 2.

4. Запустити та налаштувати обраний засіб моніторингу та натиснути кнопку "Start".

5. Сформувати звіт роботи генератора "**AITS System EventsGenerator (1-7 lab).exe**" та оформити звіт лабораторної роботи.

Звіт виконання лабораторної роботи повинен містити:

– Титульний лист.

– Тема та мета лабораторної роботи.

– Завдання до лабораторної роботи.

– Звіт моніторингу індивідуального варіанту завдання студента.

– Скріншоти та інші матеріали на думку автора які підтверджують правильність проведеного моніторингу.

– Відповіді на контрольні питання.

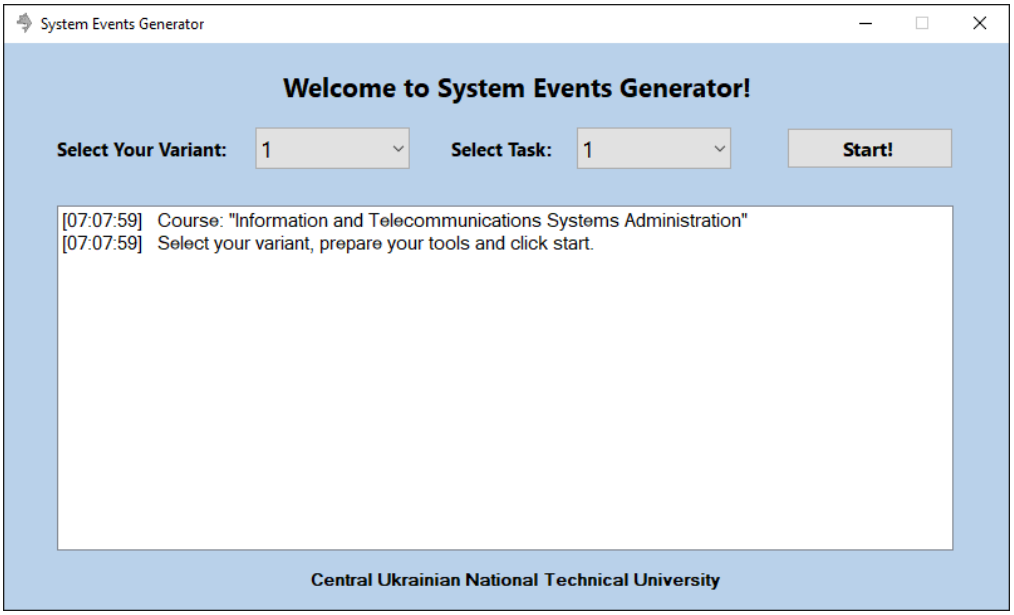

Рисунок 1 – Головне вікно генератора подій

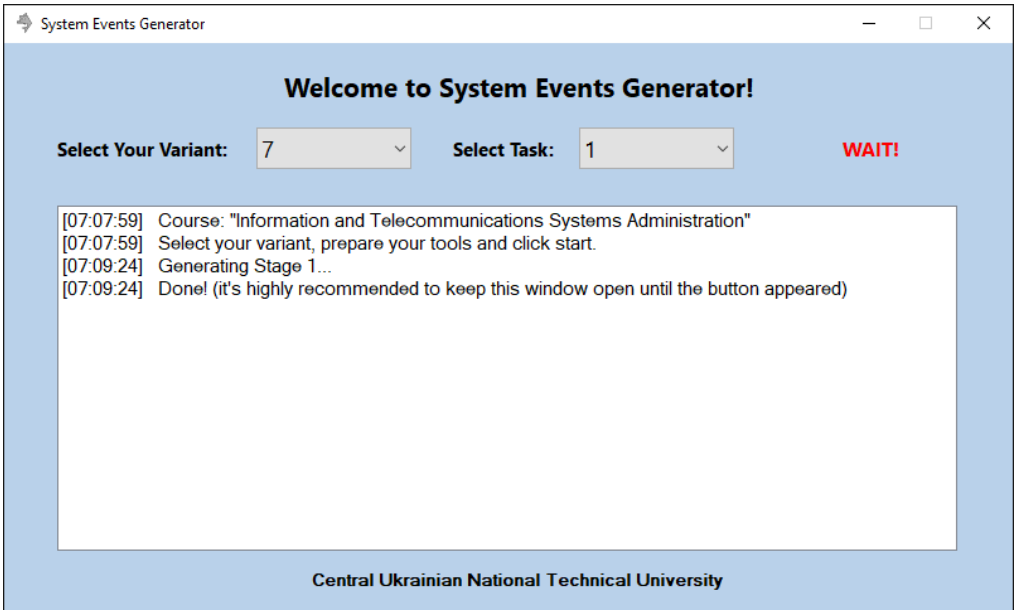

Рисунок 2 – Приклад встановлення та запуску генератора подій (Варіант 7, лабораторна робота 1)

КОНТРОЛЬНІ ЗАПИТАННЯ

1. Для чого використовують стек протоколів TCP/IP?
2. Наведіть формат TCP сегменту.
3. Перерахуйте поля заголовків TCP сегменту та керуючі біти.
4. Наведіть приклад використання протоколу TCP.
5. Для чого використовуються TCP-порти?
6. Наведіть приклади використання відомих TCP-портів.
7. У чому різниця TCP та UDP портів?
8. Порти TCP та UDP не залежать один від одного?
9. Яку кількість TCP портів можна відкрити одночасно?

Лабораторна робота №6 (семестр 5)

ТЕМА: Моніторинг параметрів запуску інформаційно-телекомунікаційної системи в реальному часі з формуванням звітності для відновлення штатного функціонування.

МЕТА: Отримати практичні навики використання систем моніторингу консольних параметрів.

ЗНАТИ: Теоретичні основи архітектурної будови ОС Windows.

ВМІТИ: Встановлювати дистрибутиви технічних засобів та утиліт для управління, діагностики, усунення неполадок та моніторингу ОС Windows.

ТЕОРЕТИЧНІ ВІДОМОСТІ.

У зв'язку з великим обсягом інформації використовувати електронну документацію в залежності з обраним шляхом вирішення завдання (погоджувати з лектором).

Для вирішення пропонується використовувати наступну документацію та технічні засоби та утиліти для управління, діагностики, усунення неполадок та моніторингу:

- Sysinternals. https://docs.microsoft.com/en-us/sysinternals/
- Nirsoft https://www.nirsoft.net/utils/index.html
- Washington University Computer Science & Engineering https://www.cse.wustl.edu/~jain/cse567-06/ftp/os_monitors/index.html/

– Flexense Ltd SysGauge https://www.sysgauge.com/

ХІД ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

1. Визначити свій індивідуальний варіант завдання - відповідно до порядкового номеру студента у групі (погоджувати з лектором).

2. Запустити генератор подій “**AITS System EventsGenerator (1-7 lab).exe**” що зображено на рисунку 1.

3. Вибрати номер індивідуального варіанту (Select your variant) та номер лабораторної роботи (select task) у випадаючих списках як це показано на рисунку 2.

4. Запустити та налаштувати обраний засіб моніторингу та натиснути кнопку “Start”.

5. Сформувати звіт роботи генератора “**AITS System EventsGenerator (1-7 lab).exe**” та оформити звіт лабораторної роботи.

Звіт виконання лабораторної роботи повинен містити:

– Титульний лист.

– Тема та мета лабораторної роботи.

– Завдання до лабораторної роботи.

– Звіт моніторингу індивідуального варіанту завдання студента.

– Скріншоти та інші матеріали на думку автора які підтверджують правильність проведеного моніторингу.

– Відповіді на контрольні питання.

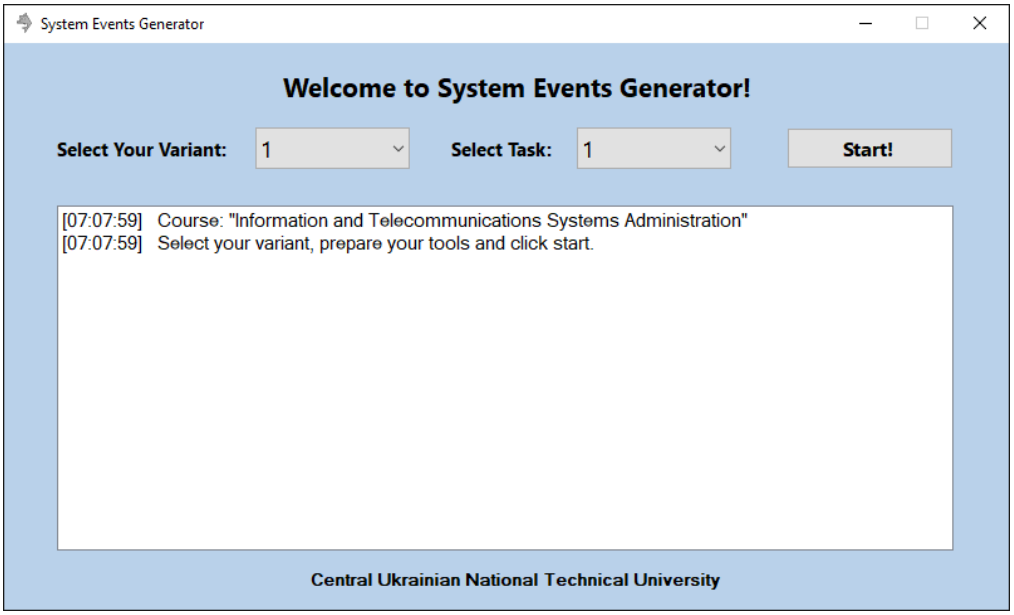

Рисунок 1 – Головне вікно генератора подій

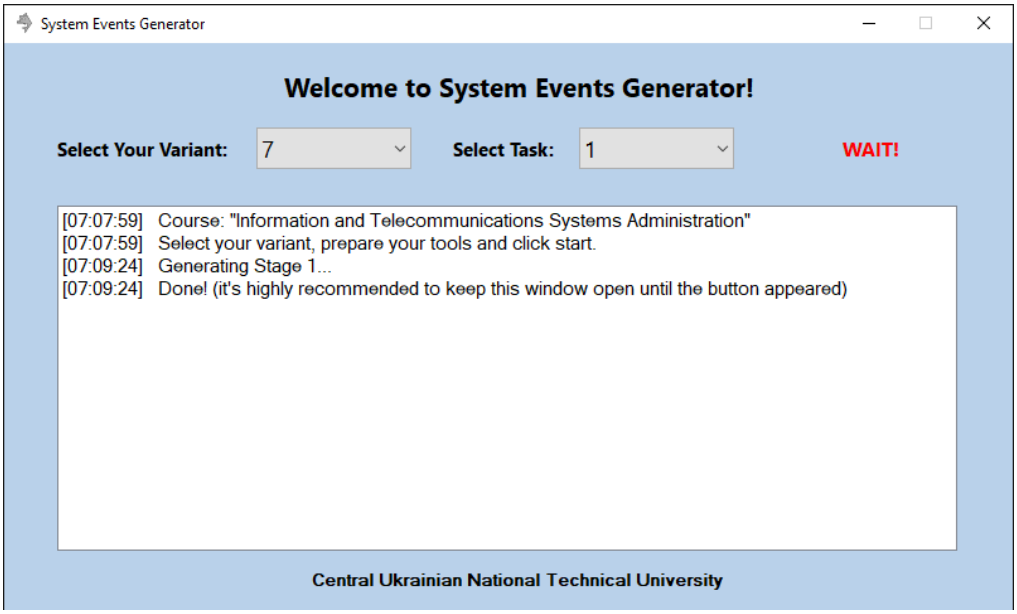

Рисунок 2 – Приклад встановлення та запуску генератора подій (Варіант 7, лабораторна робота 1)

КОНТРОЛЬНІ ЗАПИТАННЯ

1. В яких випадках використовується командний інтерпретатор?
2. В яких випадках використовується пакетний файл (batch file)?
3. Що таке PowerShell?
4. Для чого використовується PowerShell_ise?
5. Для чого використовують командлети?
6. Наведіть приклад формування скрипту з використанням командлету GET-CONTENT.
7. Наведіть приклад формування скрипту з використанням командлетів GET-SERVICE/ START-SERVICE/STOP-SERVICE.
8. Наведіть приклад формування скрипту з використанням командлетів GET-PROCESS/ STOP-PROCESS.

Лабораторна робота №7 (семестр 5)

ТЕМА: Моніторинг паралельних обчислень процесів інформаційно-телекомунікаційної системи в реальному часі з формуванням звітності для відновлення штатного функціонування.

МЕТА: Отримати практичні навики використання систем моніторингу консольних параметрів.

ЗНАТИ: Теоретичні основи архітектурної будови ОС Windows та основи паралельного програмування.

ВМІТИ: Встановлювати дистрибутиви технічних засобів та утиліт для управління, діагностики, усунення неполадок та моніторингу ОС Windows.

ТЕОРЕТИЧНІ ВІДОМОСТІ

У зв'язку з великим обсягом інформації використовувати електронну документацію в залежності з обраним шляхом вирішення завдання (погоджувати з лектором).

Для вирішення пропонується використовувати наступну документацію та технічні засоби та утиліти для управління, діагностики, усунення неполадок та моніторингу:

- Sysinternals. https://docs.microsoft.com/en-us/sysinternals/
- Nirsoft https://www.nirsoft.net/utils/index.html
- Washington University Computer Science & Engineering https://www.cse.wustl.edu/~jain/cse567-06/ftp/os_monitors/index.html/
- Flexense Ltd SysGauge https://www.sysgauge.com/

ХІД ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

1. Визначити свій індивідуальний варіант завдання - відповідно до порядкового номеру студента у групі (погоджувати з лектором).

2. Запустити генератор подій “**AITS System EventsGenerator (1-7 lab).exe**” що зображено на рисунку 1.

3. Вибрати номер індивідуального варіанту (Select your variant) та номер лабораторної роботи (select task) у випадаючих списках як це показано на рисунку 2.

4. Запустити та налаштувати обраний засіб моніторингу та натиснути кнопку “Start”. Для тестування роботи засобу пропонується також використати програму “NullTestProject (7 lab)”. Це пустий скомпільований проект у оболонці Microsoft Visual Studio на мові програмування C#. Пустий проект дозволяє побачити як працює багатопотоковість по замовчанню та розглянути властивості операційної системи.

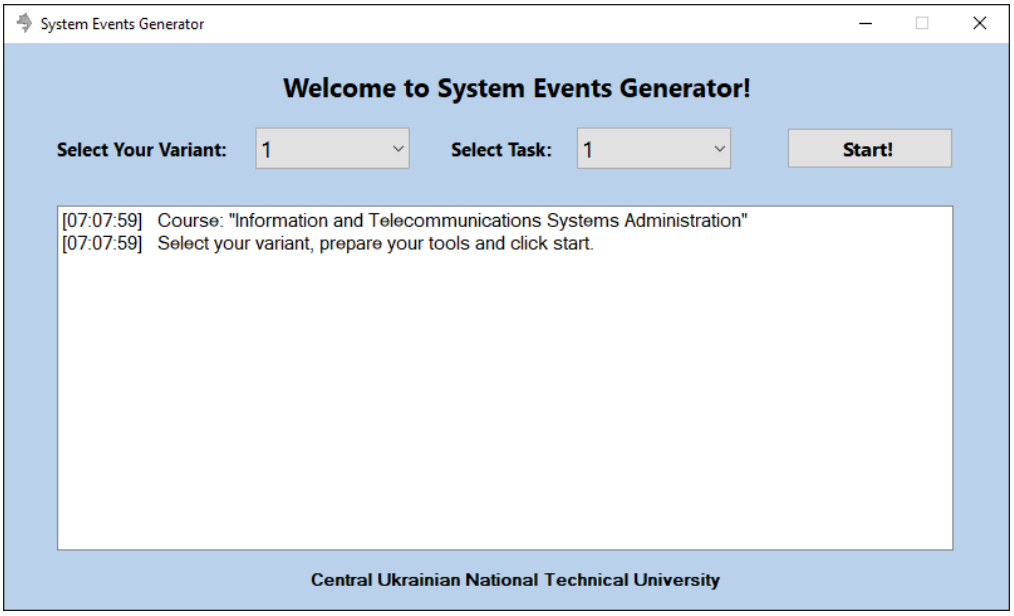

Рисунок 1 – Головне вікно генератора подій

5. Сформувати звіт роботи генератора “**AITS System EventsGenerator (1-7 lab).exe**” та оформити звіт лабораторної роботи.

Звіт виконання лабораторної роботи повинен містити:

– Титульний лист.

– Тема та мета лабораторної роботи.

– Завдання до лабораторної роботи.

– Звіт моніторингу індивідуального варіанту завдання студента.

– Скріншоти та інші матеріали на думку автора які підтверджують правильність проведеного моніторингу.

– Відповіді на контрольні питання.

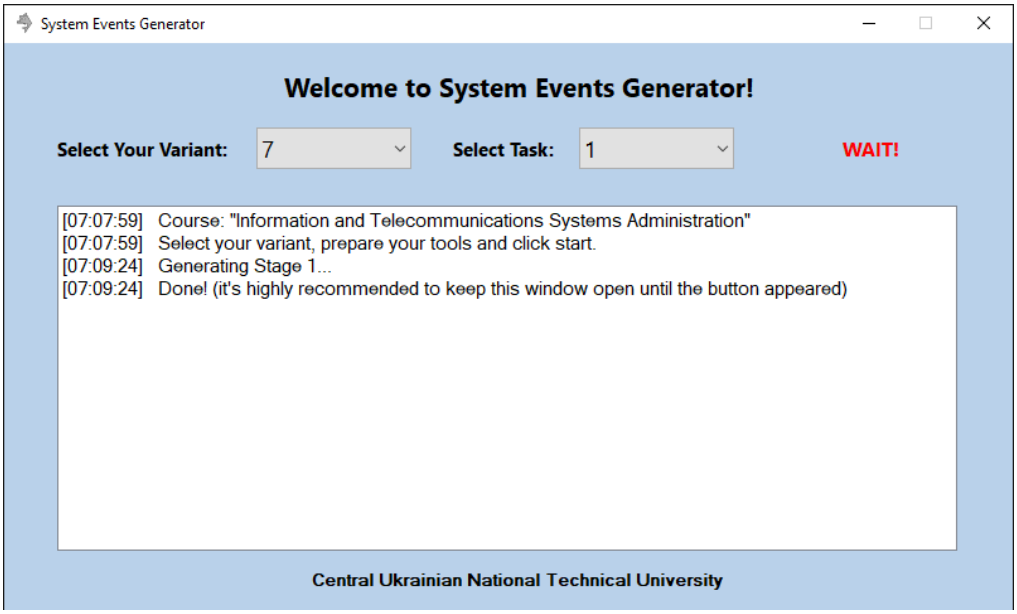

Рисунок 2 – Приклад встановлення та запуску генератора подій (Варіант 7, лабораторна робота 1)

КОНТРОЛЬНІ ЗАПИТАННЯ

1. Для яких цілей використовують потоки?
2. Дайте загальну характеристику потоків.
3. Скільки потоків (thread) використовується у програмному забезпеченні по замовчанню?
4. В чому різниця між потоком та процесом?

5. Які існують переваги та недоліки використання багатопотоковості?
6. Які існують рівні пріоритету потоків?
7. Які існують стани потоків?

Лабораторна робота №1 (семестр 6)

ТЕМА: АНАЛІЗ СИМВОЛЬНИХ ДАНИХ ДВІЙКОВОГО КОДУ ПЗ

МЕТА: Отримати практичні навики використання методів та інструментів статичного аналізу даних

ЗНАТИ: Основи програмування в ОС Windows.

ВМІТИ: Застосовувати інструментарій статичного аналізу даних.

ТЕОРЕТИЧНІ ВІДОМОСТІ

Аналіз символьних даних – це процес вилучення читабельних символів та строк формату Ascii та Unicode із двійкового коду програмного забезпечення, що аналізується. З урахуванням можливостей заплутування даних зловмисником (obfuscated string) за допомогою потокового шифрування символьних даних (xor, base64, rc4, та ін.) та відсікання генерованих хибних текстових даних що перешкоджають аналізу.

Для вирішення поставленого завдання лабораторної роботи пропонується використовувати системну утиліту командного рядка **Flare-Floss** – “floss.exe”.

Можна використовувати інші підходи чи написати власний аналог (вихідний код алгоритму Flare-Floss доступний).

Flare-Floss вирішує проблему з заплутаними та зашифрованими рядками – програма автоматично витягує заплутані (obfuscated) рядки із двійкового коду ПЗ, що аналізується. ПЗ об’єднує та автоматизує різні техніки, щоб виконати декодування рядків.

Замість того, щоб надійно захищати бекдори пакувальниками (наприклад UPX), багато авторів зловмисного ПЗ уникають евристичних виявлень, маскуючи лише ключові частини виконуваного файлу.

Часто ці частини є рядками та ресурсами, які використовуються для налаштування доменів (ftp, https, ip та ін), файлів та інших артефактів зараження. Ці ключові функції не відображатимуться як звичайний текст у виводі утиліти strings.exe, яку використовують під час базового статичного аналізу.

FLARE Obfuscated String Solver (FLOSS, раніше FireEye Labs Obfuscated String Solver) використовує розширені методи статичного аналізу для автоматичної деобфускації рядків із двійкових файлів шкідливого програмного забезпечення. Це поліпшена версія утиліти командного рядка strings.exe від Sysinternals, з можливістю аналізу заплутаних ресурсів.

У зв'язку з великим обсягом інформації використовувати електронну документацію відповідно обраного інструментарію, операційної системи та мови програмування (погоджувати з лектором).

Пропонується використовувати:

1. FLARE Obfuscated String Solver (Офіційна сторінка):
https://github.com/mandiant/flare-floss

2. FLARE (Source code and exe file Linux, Mac, Windows):
https://github.com/mandiant/flare-floss/releases

3. FLARE (Source testfiles):
https://github.com/mandiant/flare-floss-testfiles/tree/master/src

Пропонується переглянути додаткові мультимедіа матеріали:

4. FLOSS Every Day: Automatically Extracting Obfuscated Strings from Malware
https://www.youtube.com/watch?v=i2gDruusO3I&pp=ugMICgJydRABGAE%3D&ab_channel=SANSDigitalForensicsandIncidentResponse

ХІД ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

Визначити свій індивідуальний варіант завдання - відповідно до списку наданого у курсі “Адміністрування інформаційно-телекомунікаційних систем” (дистанційне навчання ЦНТУ http://moodle.kntu.kr.ua/course/view.php?id=1673), завантажити файл відповідно варіанту (1.exe-40.exe).

(Основне завдання)

Використовуючи обраний засіб обробки заплутаних та зашифрованих рядків проаналізувати виконуваний файл відповідно до індивідуального варіанту
(1.exe-40.exe). Отримати декодований унікальний ключ студента у форматі «Good job key: XX-XX-XX».

(Додаткове завдання, необов’язкове на підвищені бали)

Реалізувати кодування та декодування даних з перевіркою Flare-Floss на основі одного з наступних вихідних кодов декодування:

- (github.com) decode-base64;
- (github.com) decode-from-global;
- (github.com) decode-from-heap;
- (github.com) decode-from-stack;
- (github.com) decode-global-stack strings;
- (github.com) decode-in-place;
- (github.com) decode-local-stack strings;
- (github.com) decode-rc4;
- (github.com) decode-reencode-string;
- (github.com) decode-single-byte-xor;
- (github.com) decode-split-stackstrings;

– (github.com) decode-stackstrings-move-to-global;
– (github.com) decode-string-by-index;
– (github.com) decode-string-with-header;
– (github.com) decode-substitution-cipher;
– (github.com) decode-tightstring;
– (github.com) decode-to-global;
– (github.com) decode-to-heap;
– (github.com) decode-to-output-buf;
– (github.com) decode-to-stack-rep-mov;
– (github.com) decode-to-stack;
– (github.com) decode-wrapped-decoder.

Звіт виконання лабораторної роботи повинен містити:

– Титульний лист.

– Тема та мета лабораторної роботи.

– Завдання до лабораторної роботи.

– Звіт основного завдання.

– Звіт додаткового завдання з **блок схемою алгоритму роботи програми**.

– Скріншоти та інші матеріали на думку автора які підтверджують правильність виконаної лабораторної роботи.

– Відповіді на контрольні питання.

КОНТРОЛЬНІ ЗАПИТАННЯ

1. Таблиця кодування символів формату US-ASCII.
2. Таблиця кодування символів формату КОИ-8.
3. Таблиця кодування символів формату Windows-1251.
4. Таблиця кодування символів формату KOI8-U.
5. Таблиця кодування символів формату UTF-8.
6. Таблиця кодування символів формату UTF-16LE.
7. Дайте визначення та опис терміну rc4.
8. Дайте визначення та опис терміну base64.
9. Дайте визначення та опис терміну byte-xor.
10. З якими типами строк працює FLARE?

Лабораторна робота №2 (семестр 6)

ТЕМА: ФОРМАТ БІНАРНИХ ВИКОНУВАНИХ ФАЙЛІВ, ОБ'ЄКТНОГО КОДУ ТА ДИНАМІЧНИХ БІБЛІОТЕК (DLL)

МЕТА: Отримати практичні навики використання методів та інструментів статичного аналізу даних

ЗНАТИ: Основи програмування в ОС Windows.

ВМІТИ: Застосовувати інструментарій статичного аналізу даних.

ТЕОРЕТИЧНІ ВІДОМОСТІ

PE Format - це формат файлів, для всіх 32- і 64-розрядних Windows систем. На даний момент існує два формати PE-файлів: PE32 та PE32+. PE32 – формат для x86 систем, а PE32+ для x64.

Portable Executable (PE) – формат виконуваних бінарних файлів, об'єктного коду та динамічних бібліотек (DLL), що використовується в 32- та 64-розрядних версіях операційної системи Microsoft Windows.

PE є структурою даних, що містить всю інформацію, необхідну PE-завантажувачу для відображення файлу в пам'ять. Виконуваний код включає посилання для зв'язування динамічних бібліотек, таблиці експорту та імпорту API-функцій, дані для управління ресурсами і дані локальної пам'яті потоку (TLS).

У операційних системах сімейства Windows NT формат PE використовується для EXE, DLL, SYS (драйверів пристроїв) та інших типів файлів, що виконуються.

Платформа .NET корпорації Microsoft розширила формат PE за допомогою функцій, що підтримують загальномовне середовище виконання (Common Language Runtime - CLR). Серед додатків – заголовок CLR та секція даних CLR. Після завантаження двійкового файлу завантажувач ОС

призводить до виконання CLR через посилання у таблиці імпорту PE/COFF. Потім CLR завантажує заголовок CLR та секції даних.

У зв'язку з великим обсягом інформації використовувати електронну документацію та інструментарій (погоджувати з лектором).

Документація формату:

1. (microsoft.com) PE Format

https://learn.microsoft.com/en-us/windows/win32/debug/pe-format

2. Заголовний файл winnt.h (Win source):

https://source.winehq.org/source/include/winnt.h

3. Structure of a Portable Executable 32 bit

https://upload.wikimedia.org/wikipedia/commons/1/1b/Portable_Executable_32_bit_Structure_in_SVG_fixed.svg

Пропонується використовувати інструментарій:

1. PeStudio

https://www.winitor.com/download

2. CFF Explorer

https://ntcore.com/?page_id=388

3. PEiD Tool

https://www.softpedia.com/get/Programming/Packers-Crypters-Protectors/PEiD-updated.shtml

4. Resource Hacker

http://www.angusj.com/resourcehacker/

5. (off/on-line) HEX EDITOR

https://hexed.it/

Пропонується переглянути додаткові мультимедіа матеріали (більш детально на сторінці курсу у Moodle):

1. PE file format part1 - DOS Headers, Signature, File Header

https://www.youtube.com/watch?v=bZ-swh1S01A&list=PLCLxMnnAnGilosKSkzfg0LCdKZSmkdbDl&ab_channel=Tech69

ХІД ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

Визначити свій індивідуальний варіант завдання - відповідно до списку наданого у курсі “Адміністрування інформаційно-телекомунікаційних систем” (дистанційне навчання ЦНТУ http://moodle.kntu.kr.ua/course/view.php?id=1673), завантажити файл відповідно варіанту (1.exe-40.exe) та бібліотеку MalwareNetCard.dll.

(Основне завдання)

1. Виправте формат файлу.
2. Визначте компілятор.
3. Визначте оригінальне ім'я файлу.
4. Визначте та проаналізуйте бібліотеки що використовуються та експортуються.
5. Визначте та проаналізуйте строкові ресурси файлу та визначте незвичні HTTP та FTP посилання.
6. Визначте час та дату компіляції.
7. Визначте лишню секцію файлу. Відокремте та визначте її формат секції з послідуючим збереженням та проведенням відповідних дій.
8. Знайдіть у секції унікальний ключ студента у форматі «LBXXXXX».

(Додаткове завдання, необов’язкове на підвищені бали)

1. Створити власну схему та розказати повну структуру PE Format.

<u>Звіт виконання лабораторної роботи повинен містити</u>:

– Титульний лист.

– Тема та мета лабораторної роботи.

– Завдання до лабораторної роботи.

– Звіт основного завдання, з наступними відповідями: формат файлу; назва компілятору; оригінальне ім'я файлу; список бібліотеки що

використовуються та експортуються; строкові ресурси файлу (HTTP, FTP); час та дата компіляції; унікальний ключ студента.

– Звіт додаткового завдання – **створена власна схема** PE Format з описами розділів та секцій.

– Скріншоти та інші матеріали на думку автора які підтверджують правильність виконаної лабораторної роботи.

– Відповіді на контрольні питання.

КОНТРОЛЬНІ ЗАПИТАННЯ

1. Дайте визначення Portable Executable?
2. Для чого необхідно використовувати сигнатуру PE-файлу?
3. Яку структуру має PE32 файл?
4. Яку структуру має PE32+ файл?
5. Для чого необхідна таблиця імпорту PE-файлу?
6. Для чого необхідна таблиця експорту PE-файлу?
7. Для чого необхідна таблиця переміщень PE-файлу?
8. Які існують стандартні секції у PE-файлі?

Лабораторна робота №3 (семестр 6)

ТЕМА: ІНСТРУМЕНТИ АНАЛІЗУ ПОВЕДІНКИ ПІДСИСТЕМ ОС.

МЕТА: Отримати практичні навики використання методів та інструментів динамічного аналізу даних.

ЗНАТИ: Основи програмування в ОС Windows.

ВМІТИ: Застосовувати інструментарій динамічного аналізу даних.

ТЕОРЕТИЧНІ ВІДОМОСТІ

На відміну від статичного аналізу зловмисного програмного забезпечення, динамічний аналіз шкідливого програмного забезпечення виконується шляхом аналізу коду під час його роботи. Щоб вивчити поведінку виконуваного файлу, рекомендується працювати з віртуальними середовищами по типу Oracle VirtualBox.

Щоб зрозуміти функціональність зловмисного програмного забезпечення та запобігти його розповсюдженню, реверсивні інженери використовують налаштування під час розширеного динамічного аналізу зловмисного програмного забезпечення.

У зв'язку з великим обсягом інформації використовувати електронну документацію та інструментарій (погоджувати з лектором).

Пропонується використовувати інструментарій:

1. Process Monitor (Sysinternals)

https://learn.microsoft.com/en-us/sysinternals/downloads/procmon

2. ProcDOT

https://www.procdot.com/downloadprocdotbinaries.htm

3. Process Monitor plus ProcDOT install, CSV export

https://www.aldeid.com/wiki/ProcDOT

4. Process hacker

https://processhacker.sourceforge.io/downloads.php

Пропонується переглянути додаткові мультимедіа матеріали (більш детально на сторінці курсу у Moodle):

1. Visual Analysis with ProcDOT
https://www.youtube.com/watch?v=KRctlgDTJz4&ab_channel=13Cubed

ХІД ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

Визначити свій індивідуальний варіант завдання - відповідно до списку наданого у курсі “Адміністрування інформаційно-телекомунікаційних систем” (дистанційне навчання ЦНТУ http://moodle.kntu.kr.ua/course/view.php?id=1673), завантажити файл «mainlab.exe» та допоміжний файл «1.dat». Виконати файл у відповідності до варіанту студента та проаналізувати результат.

(Основне завдання)

9. Інсталювати ти налаштувати обраний інструментарій.
10. Запустити файл “mainlab.exe”, обрати у випадному списку варіант студента, натиснути кнопку «Go!».
11. Проаналізувати потокові зміни, з отриманих даних виділити унікальний ключ студента у форматі «XXX-XXX-XXX-XXX-XXX».

(Додаткове завдання, необов’язкове на підвищені бали)

Завдання отримати у лектора.

<u>Звіт виконання лабораторної роботи повинен містити</u>:

– Титульний лист.

– Тема та мета лабораторної роботи.

– Завдання до лабораторної роботи.

– Звіт основного завдання, з схемою сформованою ProcDOT та виділеним унікальним ключем студента у форматі «XXX-XXX-XXX-XXX-XXX».

– Звіт додаткового завдання. Відповідно до завдання лектора.

– Скріншоти та інші матеріали на думку автора які підтверджують правильність виконаної лабораторної роботи.

– Відповіді на контрольні питання.

КОНТРОЛЬНІ ЗАПИТАННЯ

1. Дайте визначення threads processes.
2. Дайте визначення Process ID (PID).
3. Дайте визначення threads ID.
4. Дайте визначення Parent Process (PPID).
5. Дайте визначення session id.
6. Дайте визначення authentication id.
7. Дайте визначення thread duration.
8. Дайте визначення processes duration.

Лабораторна робота №4 (семестр 6)

ТЕМА: АНАЛІЗ ДАМПІВ ПАМ'ЯТІ ОПЕРАЦІЙНИХ СИСТЕМ

МЕТА: ОТРИМАТИ ПРАКТИЧНІ НАВИКИ ВИКОРИСТАННЯ ІНСТРУМЕНТІВ АНАЛІЗУ ПАМ'ЯТІ ОПЕРАЦІЙНИХ СИСТЕМ.

ЗНАТИ: Основи програмування в ОС Windows/Linux.

ВМІТИ: Застосовувати інструментарій Memory Forensics Framework.

ТЕОРЕТИЧНІ ВІДОМОСТІ

Volatility – це фреймворк криміналістики пам'яті з відкритим кодом для реагування на інциденти та аналізу зловмисного програмного забезпечення. Він написаний на Python і підтримує Microsoft Windows, Mac OS X і Linux.

Volatility – це найпоширеніша у світі структура для вилучення цифрових артефактів із вибірок енергонезалежної пам'яті (RAM). Методи вилучення виконуються повністю незалежно від системи, що досліджується, але забезпечують видимість стану виконання системи.

Volatility підтримує різноманітні зразки форматів файлів і можливість конвертувати між цими форматами:

1. Raw/Padded Physical Memory.
2. Firewire (IEEE 1394).
3. Expert Witness (EWF).
4. 32- and 64-bit Windows Crash Dump.
5. 32- and 64-bit Windows Hibernation (from Windows 7 or earlier).
6. 32- and 64-bit Mach-O files.
7. Virtualbox Core Dumps.
8. VMware Saved State (.vmss) and Snapshot (.vmsn).
9. HPAK Format (FastDump).
10. QEMU memory dumps.

11. LiME format.

У зв'язку з великим обсягом інформації використовувати електронну документацію та інструментарій відповідно до обраної операційної системи (погоджувати з лектором).

Пропонується використовувати інструментарій:

1. Volatility Workbench виключно версії 3.0

https://github.com/volatilityfoundation/volatility3

Пропонується переглянути додаткові текстові та мультимедіа матеріали (більш детально на сторінці курсу у Moodle):

2. Introduction to Memory Forensics with Volatility 3

https://www.youtube.com/watch?v=Uk3DEgY5Ue8&ab_channel=DFIRScience

3. Digital Forensics and Incident Response

https://www.crowdstrike.com/cybersecurity-101/digital-forensics-and-incident-response-dfir/

4. Indicator of Compromise

https://encyclopedia.kaspersky.com/glossary/indicator-of-compromise-ioc/

ХІД ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

Визначити свій індивідуальний варіант завдання - відповідно до списку наданого у курсі “Адміністрування інформаційно-телекомунікаційних систем” (дистанційне навчання ЦНТУ http://moodle.kntu.kr.ua/course/view.php?id=1673), завантажити файл у відповідності до варіанту студента (Folder 1-40 “Windows7even.dmp”).

(Основне завдання)

12.Інсталювати ти налаштувати обраний інструментарій.

13.Провести аналіз дампу пам’яті у відповідності до варіанту студента.

14.Визначте наступні параметри процесу «tttttttt.exe»:

– Параметри командного рядка під час запуску процесу;

– Які динамічні бібліотеки використовує процес;

– Ключі реєстру які було створено процесом;

– Які порти TCP/IP використовує процес;

– Які файли було відкрито чи відчинено процесом.

(Додаткове завдання, необов'язкове на підвищені бали)

Завдання отримати у лектора.

Звіт виконання лабораторної роботи повинен містити:

– Титульний лист.

– Тема та мета лабораторної роботи.

– Завдання до лабораторної роботи.

– Звіт основного завдання, з визначеними відповідями.

– Звіт додаткового завдання. Відповідно до завдання лектора.

– Скріншоти та інші матеріали на думку автора які підтверджують правильність виконаної лабораторної роботи.

– Відповіді на контрольні питання.

КОНТРОЛЬНІ ЗАПИТАННЯ

1. Дайте визначення абревіатурі CERT-UA.
2. Дайте визначення абревіатурі CSIRT.
3 Дайте визначення абревіатурі DFIR.
4. Дайте визначення абревіатурі SOAR.
5. Дайте визначення абревіатурі IoC.
6. Дайте визначення терміну File System Forensics**.**
7. Дайте визначення терміну Memory Forensics**.**
8. Дайте визначення терміну Network Forensics**.**
9. Дайте визначення терміну Log Analysis.

Лабораторна робота №5 (семестр 6)

ТЕМА: УМОВИ ВИКОРИСТАННЯ ДИЗАСЕМБЛЕРІВ.

МЕТА: Отримати практичні навики використання дизасемблерів для зворотного аналізу програмного забезпечення

ЗНАТИ: Основи програмування в ОС Windows.

ВМІТИ: Застосовувати інструментарій декомпіляції.

ТЕОРЕТИЧНІ ВІДОМОСТІ

Дизасемблери є дуже корисним інструментом для дослідження скомпільованого виконуваного бінарного файлу та надання загальної картини розуміння того, що він робить.

Зворотна розробка чи аналіз (reverse engineering) – це дослідження деякої програми з метою розуміння принципів роботи досліджуваного об'єкта. Найчастіше використовується з метою створення об'єкта, за функціональністю аналогічного досліджуваному але без точного копіювання його функцій. Використовують в першу чергу для вивчення роботи шкідливого програмного забезпечення (malware reverse engineering) коли потрібно визначити алгоритм роботи.

Виконувані файли містять машинний код у вигляді двійкових даних. Дизасемблери переводять машинний код на більш зручний асемблерний код чи на декілька мов одночасно (наприклад C та Asm одночасно).

Дизасемблер (disassembler) – комп'ютерна програма що транслює мову машинних кодів.

Дизасемблер не є декомпілятором. Результатом роботи декомпілятора є представлення програмного коду мовою високого рівня. Дизасемблер представляє програмний код у вигляді асемблерного коду.

Результат роботи дизасемблера, дизасембльований код, форматується для простішого сприйняття людиною, перетворюючи дизасемблер у засіб для

зворотної розробки. Його використання дозволяє розібратися з деталями функціонування коду, провести певну оптимізацію окремих критичних ділянок коду, а також може використовуватися для усунення вбудованого захисту тобто злому.

Дизасемблери можуть просто подавати код у вигляді асемблерного коду, а можуть бути інтерактивними.

У зв'язку з великим обсягом інформації використовувати електронну документацію та інструментарій відповідно до обраної операційної системи (погоджувати з лектором).

Пропонується використовувати інструментарій:

Ghidra (National Security Agency)

https://github.com/NationalSecurityAgency/ghidra/releases

Пропонується переглянути додаткові мультимедіа матеріали (більш детально на сторінці курсу у Moodle):

Malware Analysis With Ghidra (HackerSploit)

https://www.youtube.com/watch?v=TJhfnItRVOA&list=PLBf0hzazHTGMSlOI2HZGc08ePwut6A2Io&index=17&ab_channel=HackerSploit

ХІД ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

Визначити свій індивідуальний варіант завдання - відповідно до списку наданого у курсі “Адміністрування інформаційно-телекомунікаційних систем” (дистанційне навчання ЦНТУ http://moodle.kntu.kr.ua/course/view.php?id=1673), завантажити файл у відповідності до варіанту студента (1.exe-40.exe) та отримати результат.

(Основне завдання)

1. Інсталювати ти налаштувати обраний інструментарій.
2. Провести дизасемблювання виконуючого бінарного файлу у відповідності до варіанту студента (1.exe-40.exe).

3. Проаналізувати дизасембльований код та отримати унікальний ключ студента у форматі «XXX» чи «XX».

(Додаткове завдання, необов'язкове на підвищені бали)

Завдання отримати у лектора.

Звіт виконання лабораторної роботи повинен містити:

– Титульний лист.

– Тема та мета лабораторної роботи.

– Завдання до лабораторної роботи.

– Звіт основного завдання, з отриманим унікальним ключем студента у форматі «XXX» чи «XX» та вставками дизасембльованого коду підтверджуючими проведену роботу.

– Звіт додаткового завдання. Відповідно до завдання лектора.

– Скріншоти та інші матеріали на думку автора які підтверджують правильність виконаної лабораторної роботи.

– Відповіді на контрольні питання.

КОНТРОЛЬНІ ЗАПИТАННЯ

1. Дайте визначення ASM команді безумовної передачі керування JMP.
2. Дайте визначення ASM команді умовної передачі керування JZ/JE.
3. Дайте визначення ASM команді умовної передачі керування JNZ/JNE.
4. Дайте визначення ASM команді умовної передачі керування JC/JNAE/JB.
5. Дайте визначення ASM команді умовної передачі керування JNC/JAE/JNB.
6. Які існують регістри загального призначення 8/32/64 розрядні?
7. Як зберігаються змінні у мові ASM?

Лабораторна робота №6 (семестр 6)

ТЕМА: УМОВИ ВИКОРИСТАННЯ ДЕКОМПІЛЯТОРІВ.

МЕТА: Отримати практичні навики використання декомпіляторів для зворотного аналізу програмного забезпечення

ЗНАТИ: Основи програмування в ОС Windows.

ВМІТИ: Застосовувати інструментарій декомпіляції.

ТЕОРЕТИЧНІ ВІДОМОСТІ

Дизасемблери є дуже корисним інструментом для дослідження скомпільованого виконуваного бінарного файлу та надання загальної картини розуміння того, що він робить, проте ASM код важко аналізувати.

Декомпіля́тор (також детранслятор чи обернений транслятор) – комп'ютерна програма, яка транслює модуль у вигляді двійкового машинного коду (об'єктний код отриманий на виході компілятора в процесі компіляції) у **функціонально подібний** вихідний код на мові програмування високого рівня – Python, Java, C, C#.

Процес трансляції коду декомпілятором називається декомпіляцією. Декомпілятори, як і дизасемблери використовуються для дослідження та зворотної розробки програмного забезпечення.

Процес компіляції є незворотним в тому значенні, що не існує однозначної функції, яка б дозволила отримати назад джерельний код в початковому вигляді, оскільки при цьому втрачається дуже багато інформації. Тому в декомпіляторі використовуються різні методи та способи зворотного аналізу. Отриманий таким чином вихідний код на одній з мов програмування високого рівня, як правило, тільки функціонально схожий до первинної версії вихідного коду, який використовувався для створення піддослідної програми. Однак деякі техніки допомагають відтворенню такого коду.

Декомпіляція машинного байт-коду з мов, які виконуються з допомогою віртуальної машини (Java, C#) є як правило набагато простішою, бо компілятори таких мов залишають набагато більше інформації, ніж компілятори в машинний двійковий код (C, C++).

У зв'язку з великим обсягом інформації використовувати електронну документацію та інструментарій відповідно до обраної операційної системи (погоджувати з лектором).

Пропонується використовувати інструментарій:

dnspy

https://github.com/dnSpy/dnSpy

Пропонується переглянути додаткові мультимедіа матеріали (більш детально на сторінці курсу у Moodle):

Malware Analysis With Amr Thabet (HackerSploit)

https://www.youtube.com/watch?v=ZKObRxxbOCQ&list=PLBf0hzazHTGMSlOI2HZGc08ePwut6A2Io&index=16&ab_channel=HackerSploit

ХІД ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

Визначити свій індивідуальний варіант завдання - відповідно до списку наданого у курсі “Адміністрування інформаційно-телекомунікаційних систем” (дистанційне навчання ЦНТУ http://moodle.kntu.kr.ua/course/view.php?id=1673), завантажити файл у відповідності до варіанту студента (1.exe-40.exe) та отримати результат.

(Основне завдання)

1. Інсталювати ти налаштувати обраний інструментарій (для роботи необхідно встановити dotnet sdk 5).
2. Провести декомпіляцію виконуючого файлу у відповідності до варіанту студента (1.exe-40.exe).
3. Проаналізувати декомпіляцій ний код та отримати унікальний ключ студента у форматі «XXXXXXXXXXXXXXXXX».

(Додаткове завдання, необов'язкове на підвищені бали)

Завдання отримати у лектора.

Звіт виконання лабораторної роботи повинен містити:

– Титульний лист.

– Тема та мета лабораторної роботи.

– Завдання до лабораторної роботи.

– Звіт основного завдання, з отриманим унікальним ключем студента у форматі «XXXXXXXXXXXXXXXXX» та вставками вихідного коду підтверджуючими проведену роботу.

– Звіт додаткового завдання. Відповідно до завдання лектора.

– Скріншоти та інші матеріали на думку автора які підтверджують правильність виконаної лабораторної роботи.

– Відповіді на контрольні питання.

КОНТРОЛЬНІ ЗАПИТАННЯ

1. Дайте визначення терміну .NET Framework.
2. Що таке керований код платформи .NET?
3. Для чого використовується JIT-компіляція?
4. Дайте визначення терміну ILSpy decompiler engine
5. Дайте визначення терміну Roslyn
6. Дайте визначення терміну dnlib
7. Дайте визначення терміну VS MEF
8. Дайте визначення терміну ClrMD

Лабораторна робота №7 (семестр 6)

ТЕМА: ІДЕНТИФІКАЦІЯ ТА КЛАСИФІКАЦІЯ ЗРАЗКІВ ЗЛОВМИСНОГО ПЗ

МЕТА: Отримати практичні навики реагування на інциденти та ліквідації наслідків.

ЗНАТИ: Основи програмування в ОС Windows/Linux.

ВМІТИ: Застосовувати інструментарій ідентифікації та класифікації.

ТЕОРЕТИЧНІ ВІДОМОСТІ

YARA – це інструмент, призначений (але не обмежуючись цим) допомогти дослідникам зловмисного ПЗ ідентифікувати та класифікувати зразки зловмисного ПЗ. За допомогою YARA можна створювати описи сімейств зловмисних програм (або будь-які інші, які ви хочете описати) на основі текстових або бінарних шаблонів. Кожен опис, або правило, складається з набору рядків і логічного виразу, які визначають його логіку.

Давайте подивимося на приклад:

```
rule silent_banker : banker
{
    meta:
        description = "This is just an example"
        threat_level = 3
        in_the_wild = true

    strings:
        $a = {6A 40 68 00 30 00 00 6A 14 8D 91}
        $b = {8D 4D B0 2B C1 83 C0 27 99 6A 4E 59 F7 F9}
        $c = "UVODFRYSIHLNWPEJXQZAKCBGMT"

    condition:
        $a or $b or $c
}
```

Наведене вище правило повідомляє YARA, що будь-який файл, який містить один із трьох рядків, має бути позначено як silent_banker.

Це лише простий приклад, складніші та потужніші правила можна створити за допомогою символів підстановки, рядків без урахування регістру, регулярних виразів, спеціальних операторів та багатьох інших функцій, які можна знайти в документації YARA.

У зв'язку з великим обсягом інформації використовувати електронну документацію та інструментарій відповідно до обраної операційної системи (погоджувати з лектором).

Пропонується використовувати інструментарій:

YARA

https://github.com/VirusTotal/yara

Samples yara-rules

https://github.com/godaddy/yara-rules/blob/master/example.yara

YARA Rule Examples

https://support.phishingtackle.com/hc/en-gb/articles/4410170814609-YARA-Rule-Examples

YARA's documentation

https://yara.readthedocs.io/en/stable/

Yara Rules Project

https://github.com/Yara-Rules

Writing YARA rules

https://yara.readthedocs.io/en/stable/writingrules.html?highlight=%24mz#accessing-data-at-a-given-position

Пропонується переглянути додаткові мультимедіа матеріали (більш детально на сторінці курсу у Moodle):

Finding Evil with YARA

https://www.youtube.com/watch?v=mQ-mqxOfopk&ab_channel=13Cubed

What are Yara Rules (and How Cybersecurity Analysts Use Them)

https://www.youtube.com/watch?v=BM23_H2GGMA&ab_channel=GeraldAuger%2CPhD-SimplyCyber

ХІД ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

Визначити свій індивідуальний варіант завдання - відповідно до списку наданого у курсі “Адміністрування інформаційно-телекомунікаційних систем” (дистанційне навчання ЦНТУ http://moodle.kntu.kr.ua/course/view.php?id=1673), на основі завдань виконаних лабораторних робіт 1-3, 5-6 завантажити бінарні файли у відповідності до варіанту студента (**Л/Р 1:**1.exe-40.exe; **Л/Р 2:**1.exe-40.exe+ MalwareNetCard.dll; **Л/Р 3**: mainlab.exe+1.dat; **Л/Р 5**:1.exe-40.exe; **Л/Р 6**:1.exe-40.exe;+add file) та провести наступні маніпуляції.

(Основне завдання)

1. **ЧІТКА HASH КЛАСИФІКАЦІЯ**.

Створити сигнатури hash виконуваних бінарних файлів за допомогою криптографічних алгоритмів обчислення **MD5, SHA-1, SHA-256**.

2. **НЕЧІТКЕ ХЕШУВАННЯ SSDEEP**

Створити сигнатури виконуваних бінарних файлів за допомогою нечіткого хешування **SSDEEP**.

3. **ХЕШ ІМПОРТОВАНИХ БІБЛІОТЕК IMPHASH**

Створити сигнатури виконуваних бінарних файлів за допомогою нечіткого хешування **IMPHASH**.

4. **YARA КЛАСИФІКАЦІЯ**.

Створити правила YARA виконуваних бінарних файлів з обов’язковим використанням наступних правил:

1. $ascii_string, $unicode_string, $hex_string (YARA string modifiers).
2. xor, filesize (YARA keywords).
3. dotnet.version, dotnet.assembly.name (YARA keywords dotnet).
4. pe.exports, pe.number_of_sections (YARA keywords PE module).

(Додаткове завдання, необов'язкове на підвищені бали)

Завдання отримати у лектора.

Звіт виконання лабораторної роботи повинен містити:

– Титульний лист.

– Тема та мета лабораторної роботи.

– Завдання до лабораторної роботи.

– Звіт основного завдання зі вставками HASH даних, YARA правил.

– Звіт додаткового завдання. Відповідно до завдання лектора.

– Скріншоти та інші матеріали на думку автора які підтверджують правильність виконаної лабораторної роботи.

– Відповіді на контрольні питання.

КОНТРОЛЬНІ ЗАПИТАННЯ

1. Дайте визначення терміну MISP-UA в розрізі ефективної бази даних IOC.

2. Функціонал та призначення VirusTotal.

3. Для яких цілей використовується параметри «-C,-c, -d, -f,-h,-i,-l,-x,-n,-N,-w» командного рядка програми yara64.exe?

4. Для яких цілей використовується параметри «-m,-D,-e,-S,-s,-L,-g,-r,-z,-k,-t,-p,-a,-v» командного рядка програми yara64.exe?

5. Які існують основні регулярні вирази програми yara64.exe?

6. Які існують додаткові модулі розширення основної функціональності YARA?

7. В яких випадках використовують зовнішні змінні YARA?

8. Що таке криптографічні хеш-функції?

9. Що таке нечіткі криптографічні хеш-функції?

10. У яких випадках використовується хеш імпортованих бібліотек?

Система оцінювання та вимоги

Критерії оцінки іспиту:

оцінку «відмінно» (90-100 балів, A) заслуговує студент, який:

– всебічно, систематично і глибоко володіє навчально-програмовим матеріалом;

– вміє самостійно виконувати завдання, передбачені програмою, використовує набуті знання і вміння у нестандартних ситуаціях;

– засвоїв основну і ознайомлений з додатковою літературою, яка рекомендована програмою;

– засвоїв взаємозв'язок основних понять дисципліни та усвідомлює їх значення для професії, яку він набуває;

– вільно висловлює власні думки, самостійно оцінює різноманітні життєві явища і факти, виявляючи особистісну позицію;

– самостійно визначає окремі цілі власної навчальної діяльності, виявив творчі здібності і використовує їх при вивченні навчально-програмового матеріалу, проявив нахил до наукової роботи.

оцінку « добре» (82-89 балів, B) – заслуговує студент, який:

– повністю опанував і вільно (самостійно) володіє навчально-програмовим матеріалом, в тому числі застосовує його на практиці, має системні знання достатньому обсязі відповідно до навчально-програмового матеріалу, аргументовано використовує їх у різних ситуаціях;

– має здатність до самостійного пошуку інформації, а також до аналізу, постановки і розв'язування проблем професійного спрямування;

– під час відповіді допустив деякі неточності, які самостійно виправляє, добирає переконливі аргументи на підтвердження вивченого матеріалу;

оцінку «добре» (74-81 бал, C) заслуговує студент, який:

– в загальному роботу виконав, але відповідає на заламені з певною кількістю помилок;

– вміє порівнювати, узагальнювати, систематизувати інформацію під керівництвом викладача, в цілому самостійно застосовувати на практиці, контролювати власну діяльність;

– опанував навчально-програмовий матеріал, успішно виконав завдання, передбачені програмою, засвоїв основну літературу, яка рекомендована програмою;

оцінку «задовільно» (64-73 бали, D) – заслуговує студент, який:

– знає основний навчально-програмовий матеріал в обсязі, необхідному для подальшого навчання і використання його у майбутній професії;

– виконує завдання, але при рішенні допускає значну кількість помилок;

– ознайомлений з основною літературою, яка рекомендована програмою;

– допускає на заняттях чи заламені помилки при виконанні завдань, але під керівництвом викладача знаходить шляхи їх усунення.

оцінку «задовільно» (60-63 бали, E) – заслуговує студент, який:

– володіє основним навчально-програмовим матеріалом в обсязі, необхідному для подальшого навчання і використання його у майбутній професії, а виконання завдань задовольняє мінімальні критерії. Знання мають репродуктивний характер.

оцінка «незадовільно» (35-59 балів, FX) – виставляється студенту, який:

– виявив суттєві прогалини в знаннях основного програмового матеріалу, допустив принципові помилки у виконанні передбачених програмою завдань.

оцінку «незадовільно» (35 балів, F) – виставляється студенту, який:

– володіє навчальним матеріалом тільки на рівні елементарного розпізнавання і відтворення окремих фактів або не володіє зовсім;

– допускає грубі помилки при виконанні завдань, передбачених програмою;

– не може продовжувати навчання і не готовий до професійної діяльності після закінчення університету без повторного вивчення даної дисципліни.

При виставленні оцінки враховуються результати навчальної роботи студента протягом семестру

Шкала оцінювання: національна та ЄКТС

Сума балів за всі види навчальної діяльності	Оцінка ЄКТС	Оцінка за національною шкалою	
		для екзамену, курсового проекту (роботи), практики	для заліку
90-100	A	відмінно	зараховано
82-89	B	добре	
74-81	C		
64-73	D	задовільно	
60-63	E		
35-59	FX	незадовільно з можливістю повторного складання	не зараховано з можливістю повторного складання
1-34	F	незадовільно з обов’язковим повторним вивченням дисципліни	не зараховано з обов’язковим повторним вивченням дисципліни

РЕКОМЕНДОВАНА ЛІТЕРАТУРА

Базова

1. «Читальний зал № 1 (ЦНТУ)» Глухов В. С., Костик А. Т. Дослідження і проектування комп'ютерних систем та мереж : навч. посіб. Львів : Магнолія 2006, 2025. 253 с. ISBN 978-617-574-265-5. Режим доступу: https://opac.kntu.kr.ua/cgi-bin/koha/opac-detail.pl?biblionumber=8698
2. «Читальний зал № 1 (ЦНТУ)» Комп'ютерні мережі : підручник. Львів : Магнолія 2006, 2025. 262 с. ISBN 966-8340-69-8. Режим доступу: https://opac.kntu.kr.ua/cgi-bin/koha/opac-detail.pl?biblionumber=8692
3. Смірнова Т.В., Моторін Ю.Ю., Буравченко К.О., Бочуля Т.В., Коваленко О.В. «Вибір оптимальної технології побудови хмарної інформаційно-комунікаційної системи автоматизації виробничих процесів» Вимірювальна та обчислювальна техніка в технологічних процесах, № 1 (2022). С. 15-26. 2022. Режим доступу: http://vottp.khmnu.edu.ua/index.php/vottp/article/view/30/36 (Фахове видання. Категорія «Б»)
4. Khudov H., Baranik O., Kovalenko O.,Yakovenko Y., Chahan Y. «The Information Technology for Determining Vehicle Route Based on Ant Colony Algorithms» International Journal of Emerging Technology and Advanced Engineering, 2022, 12(12), Pages 117–128. Режим доступу: https://www.scopus.com/record/display.uri?eid=2-s2.0-85130069202&origin=resultslist (Scopus).
5. Hennadii Khudov, Volodymyr Bashynskyi, Oleksandr Kovalenko, Kristina Tahyan, Oleksii Fakadii « The methods for improving the quality of detection of inconspicuous aerial objects through the use of external radiation sources» International Journal of Emerging Technology and Advanced Engineering, 2023, Volume 13, Issue 3, Pages 91-100. Режим доступу: https://doi.org/10.46338/ijetae0323_09 (Закордонне фахове видання).
6. Hennadii Khudov, Oleksandr Kostianets, Oleksandr Kovalenko, Oleh Maslenko, Yuriy Solomonenko «Using softwaredefined radio receivers for determining the coordinates of low-visible aerial objects» Eastern-European Journal of Enterprise Technologies Vol. 5 No. 9 (124), 2023, Pages 61-73. Режим доступу: https://www.scopus.com/record/display.uri?eid=2-s2.0-85172343893&origin=resultslist (Scopus).
7. Вінтенко Б.Ю., Смірнов О.А., Миронець І.В., Смірнова Т.В., Коваленко О.В., Мацуй А.М. «Модель шляхів отримання вхідних даних комп’ютерної інтелектуальної системи підтримки оперативного персоналу АЕС». Центральноукраїнський науковий вісник. Технічні науки. 2025. Вип. 11(42), ч. II. С.52-62. Режим доступу: https://mapiea.kntu.kr.ua/pdf/11(42)_II/11(42)_II_2025.pdf (Фахове видання. Категорія «Б»)

8. Коваленко О.В. Моделі та методи розробки програмного забезпечення комп’ютерних систем для підвищення безпеки даних: монография / О.В. Коваленко // К.: Вид. «КОД» – 2019. – 350 с.

Допоміжна

9. Pavel Yosifovich Windows 10 System Programming, Part 1. Independently published. 2020. 528 с.
10. Pavel Yosifovich Windows 10 System Programming, Part 2. Independently published. 2021. 555 с.
11. Jasper van Woudenberg, Colin O'Flynn The Hardware Hacking Handbook: Breaking Embedded Security with Hardware Attacks. No Starch Press. 2021. 512 с.
12. Mike Chapple, David Seidl CompTIA Security+ Certification Kit: Exam SY0-601. 2021. 1008 с.
13. Pavel Yosifovich, Mark Russinovich, David Solomon, Alex Ionescu Windows Internals: System architecture, processes, threads, memory management, and more, Part 1 (Developer Reference). Microsoft Press; 7th edition. 2017. 800 с.
14. Mark Russinovich, Andrea Allievi, Alex Ionescu, David Solomon Windows Internals, Part 2 (Developer Reference). Microsoft Press; 7th edition. 2021. 912 с.
15. Pavel Yosifovich Windows Kernel Programming. CreateSpace Independent Publishing Platform. 2019. 400 с.
16. Ayman Shaaban, Konstantin Sapronov Practical Windows Forensics: Leverage the power of digital forensics for Windows systems. Packt Publishing. 2016. 324 с.
17. Lee Holmes PowerShell Cookbook: Your Complete Guide to Scripting the Ubiquitous Object-Based Shell. O'Reilly Media. 2021. 1002 с.
18. Andrew Bettany, Mike Halsey Windows File System Troubleshooting. Apress. 2015. 170 с.
19. Mario Hewardt, Daniel Pravat Advanced Windows Debugging: Developing and Administering Reliable, Robust, and Secure Software. Addison-Wesley Professional. 2007. 842 с.
20. Yosifovich, P. Windows Kernel Programming (2nd Edition). 2023.
21. Vostokov, D. Accelerated Windows Malware Analysis with Memory Dumps (3rd Edition). OpenTask, 2022.
22. Vostokov, D. Practical Foundations of Windows Debugging, Disassembling, Reversing (3rd Edition). 2025.

Інформаційні ресурси

23. Курс «Адміністрування інформаційно-телекомунікаційних систем» на сервері дистанційної освіта ЦНТУ. – URL: https://moodle.kntu.kr.ua/course/view.php?id=736
24. Онлайн-курси UDEMY. – URL: https://www.udemy.com/ –платформа онлайн-курсів різних IT тематик.
25. Онлайн-курси Prometheus. – URL: https://prometheus.org.ua/ – українська платформа безкоштовних онлайн-курсів
26. Онлайн-курси Coursera. – URL: https://www.coursera.org –платформа онлайн-курсів різних IT тематик.
27. http://stackoverflow.com/ – система питань і відповідей для професійних програмістів та новачків у програмуванні.
28. https://dou.ua/ – український веб-сайт з елементами колективного блогу, створений для розповсюдження новин, аналітичних статей та свіжої інформації пов'язаної із інформаційними технологіями.
29. https://www.google.com/ – основна пошукова платформа.
30. https://www.youtube.com – Відеохостинг, що надає користувачам послуги зберігання, доставки та показу відео. На платформі розміщено багато курсів IT спрямованості.
31. https://biblprog.org.ua/ua/programming/ – каталог безкоштовних середовищ розроблення ПЗ.
32. Національна бібліотека України імені В. І. Вернадського: Електронні ресурси НБУВ [Електронний ресурс]. – Режим доступу: http://www.nbuv.gov.ua/.