ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ЗАТВЕРДЖУЮ
Декан ФІТ
_____ Савенко О.С.
«_31_» _____2022 р.

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Реверс-інжиніринг

Галузь знань	12 – Інформаційні технології
Спеціальність	125 – Кібербезпека
Рівень вищої освіти	Другий магістерський
Освітньо-професійна програма	Кібербезпека
Обсяг дисципліни	8 кредитів ЄКТС
Мова навчання	Українська
Шифр дисципліни	ВД.05
Статус дисципліни	вибіркова

Факультет інформаційних технологій
Кафедра кібербезпеки

Форма навчання	Курс	Семестр	Обсяг дисципліни		Кількість годин						Курсовий проект	Курсова робота	Форма семестрового контролю	
					Аудиторні заняття				Індивідуальна робота студента	Самостійна робота, у т.ч. ІРС				
			Кредити ЄКТС	Години	Разом	Лекції	Лабораторні роботи	Практичні заняття					Залік	Іспит
Д	-	-	8	240	90	36	54	-	-	150			+	

Програма складена _____ к.т.н., доцент _____ В. М. Чешун
Підпис Вчений ступінь, звання Ініціали, прізвище викладача(ів)

Схвалена на засіданні кафедри кібербезпеки

Протокол №_1_ від "_31_" _серпня_ 2022 р.

Зав. кафедри кібербезпеки _____ Ю.П. Кльоц
Підпис Ініціали, прізвище

Робоча програма розглянута та схвалена Вченою радою факультету інформаційних технологій

Голова Вченої ради факультету _____ О.С. Савенко
Підпис Ініціали, прізвище

Хмельницький 2022

РЕВЕРС-ІНЖИНІРИНГ

Тип дисципліни	Вибіркова
Рівень вищої освіти	Другий (магістерський)
Мова викладання	Українська
Семестр	-
Кількість встановлених кредитів ЄКТС	8,0
Форми навчання, для яких викладається дисципліна	Денна

Результати навчання. Студент, який успішно завершив вивчення дисципліни, повинен:
вміти застосовувати методи проведення зворотної розробки програмного забезпечення та апаратних пристроїв; досліджувати програмний код і дані; використовувати програмні засоби, які реалізують основні методи реверс інжинірингу; виявляти шкідливе програмне забезпечення, проводити аналіз шкідливих програм, розуміти технічні аспекти функціонування шкідливих програм; зменшувати негативні наслідки від впливу шкідливого програмного забезпечення; використовувати сучасні технології програмування; виконувати відтворення програмного коду програм, його динамічний і статичний аналіз, пошук і експлуатацію вразливостей, аналіз шкідливого програмного коду; проводити дослідження, обробляти та аналізувати отримані експериментальні дані.
володіти основними фундаментальними поняттями і методами реверс інжинірингу для їх використання в сучасних умовах; навичками збору і аналізу інформації; принципами побудови і алгоритмічними структурами сучасного програмного забезпечення; основним математичним апаратом та законами декомпіляції програмного забезпечення.

Зміст навчальної дисципліни. Основи реверс інжинірингу, низькорівневе програмування (асемблер), дизасеблерування, розпізнавання конструкцій мов високого програмування в асемблері, захищений режим процесора, внутрішній устрій операційної системи, WinApi функції, Native додатки, програмування служб; відтворення програмного коду програм, динамічний і статичний аналіз коду, shell коди, Metasploit, пошук і експлуатація вразливостей, аналіз шкідливого програмного коду.

Запланована навчальна діяльність: лекції – 36 год., лабораторні заняття – 54 год., самостійна робота – 150 год.; разом – 240 год.

Форми (методи) навчання: лекції (з використанням методів проблемного навчання і візуалізації); лабораторні роботи (з використанням методів комп'ютерного моделювання, тренінгів, майстер-класів), самостійна робота (індивідуальні завдання), наочні методи, робота в групі, використання сучасних інформаційних технологій та прикладного програмного забезпечення.

Форми оцінювання результатів навчання: усне опитування, захист лабораторних робіт; письмове опитування (тестування).

Вид семестрового контролю: залік.

Навчальні ресурси:
1. Andriesse D. Practical Binary Analysis. Build Your Own Linux Tools for Binary Instrumentation, Analysis, and Disassembly. San Francisco : No Starch Press, Inc., 2019. 460p.
2. Bratus S., Matrosov A., Rodionov E. Rootkits and Bootkits: Reversing Modern Malware and Next Generation Threats. San Francisco : No Starch Press, Inc, 2019. 450p.
3. Bulazel A. Reverse Engineering Windows Defender's JavaScript Engine. REcon Brussels, 2018. 147p.
4. Bulazel A. Reverse Engineering Windows Defender's Antivirus Emulator. REcon Brussels, 2018. 225p.
5. Franck De Goër de Herve. Reverse-engineering of binaries in a single execution. Universitй Grenoble Alpes, 2017. 252p.
6. Методи і алгоритми захисту інформаційних ресурсів комп'ютерних систем: навчальний посібник / В. М. Джулій, Ю. П. Кльоц, І. В. Муляр, В. М. Чешун. Хмельницький: ХмНУ, 2020. 196с.
7. Модульне середовище для навчання MOODLE. Доступ до ресурсу: https://msn.khnu.km.ua.
8. Електронна бібліотека університету. Доступ до ресурсу: http://lib.khnu.km.ua/asp/php_f/p1age_lib.php.

Викладач: канд. техн. наук, доцент Чешун В.М.

ВСТУП

Дисципліна «Реверс-інжиніринг» - вибіркова складова професійної підготовки магістрів в галузі інформаційних технологій зі спеціальності „Кібербезпека”, що охоплює сучасні підходи до декомпозиції і зворотного інжинірингу програмного забезпечення, інформаційних систем і ресурсів з метою усунення наявних вразливостей, виявлення та знешкодження шкідливого програмного забезпечення.

Мета дисципліни. Формування системи знань та розуміння основних понять та реверс-інжинірингу, необхідних для подальшої роботи та навчити застосуванню методів та засобів аналізу шкідливого програмного забезпечення в умовах широкого використання сучасних інформаційних технологій.

Предмет дисципліни. Основи, задачі, методи і засоби реверс-інжинірингу: низькорівневе програмування (асемблер), дизасеблерування, розпізнавання конструкцій мов високого програмування в асемблері, захищений режим процесора, внутрішній устрій операційної системи, WinApi функції, Native додатки, програмування служб; відтворення програмного коду програм, динамічний і статичний аналіз коду, shell коди, Metasploit, пошук і експлуатація вразливостей, аналіз шкідливого програмного коду.

Завдання дисципліни. Сформувати знання про принципи, що лежать в основі цифрової криміналістики, методи і засоби пошуку цифрових доказів, технології розслідування кіберзлочинів. Вивчення дисципліни має забезпечити набуття компетентностей та досягнення результатів навчання:

компетентності:

КЗ 1. Здатність застосовувати знання у практичних ситуаціях

КЗ 2. Знання та розуміння предметної області та розуміння професії

КЗ 4. Вміння виявляти, ставити та вирішувати проблеми за професійним спрямуванням.

Фахові компетентності

КФ 4. Здатність до проектування, впровадження, супроводження інформаційних мереж і ресурсів, безпеки інформаційних технологій (в т.ч. хмарних технологій та додатків), а також безпеки бізнес/операційних процесів з метою забезпечення функціонування інформаційно-комунікаційних систем згідно встановленої стратегії і політики інформаційної безпеки та/або кібербезпеки організації.

результати навчання:

РН 1. Використовувати результати самостійного пошуку, аналізу та синтезу інформації з різних джерел для ефективного рішення спеціалізованих задач професійної діяльності.

РН 2. Адаптуватися в умовах частої зміни технологій професійної діяльності, прогнозувати кінцевий результат.

РН 3. Вирішувати задачі захисту інформації, що обробляється в інформаційно-телекомунікаційних системах, з використанням сучасних методів та засобів криптографічного захисту інформації.

РН 4. Вирішувати задачі аналізу програмного коду на наявність можливих загроз.

РН 5. Використовувати сучасне програмно-апаратне забезпечення інформаційно-комунікаційних технологій.

Студент, який успішно завершив вивчення дисципліни, повинен:

вміти застосовувати методи проведення зворотної розробки програмного забезпечення та апаратних пристроїв; досліджувати програмний код і дані; використовувати програмні засоби, які реалізують основні методи реверс інжинірингу; виявляти шкідливе програмне забезпечення, проводити аналіз шкідливих програм, розуміти технічні аспекти функціонування шкідливих програм; зменшувати негативні наслідки від впливу шкідливого програмного забезпечення; використовувати сучасні технології програмування; виконувати відтворення програмного коду програм, його динамічний і статичний аналіз, пошук і експлуатацію вразливостей, аналіз шкідливого програмного коду; проводити дослідження, обробляти та аналізувати отримані експериментальні дані.

володіти основними фундаментальними поняттями і методами реверс інжинірингу для їх використання в сучасних умовах; навичками збору і аналізу інформації; принципами побудови і алгоритмічними структурами сучасного програмного забезпечення; основним математичним апаратом та законами декомпіляції програмного забезпечення.

СТРУКТУРА ЗАЛІКОВИХ КРЕДИТІВ ДИСЦИПЛІНИ

Назва теми	Кількість годин, відведених на:		
	лекції	**лабораторні роботи**	**самостійну роботу**
Тема 1. Основи реверс-інжинірингу	8	12	34
Тема 2. Реверс-інжиніринг і шкідливе програмне забезпечення	16	24	66
Тема 3. Методи протидії реверс-інжинірингу	12	18	50
Разом:	**36**	**54**	**150**

ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Зміст лекційного курсу

Номер лекції	Перелік тем лекцій, їх анотації	Кількість годин
	Тема 1. Основи реверс-інжинірингу	
1	**Введення в реверс-інжиніринг (Reverse Engineering).** Основні поняття та визначення. Основні задачі і сфери застосуваня реверс-інжинірингу. Літ.: [1] с.11-88; [2] с.3-34.	2
2	**Методи реверс-інжинірингу.** Реверс-інжиніринг програмного забезпечення: реверс коду і технічних підходів; реверс основної і мета-механіки ПЗ; реверс балансу; реверс на основі інтерфейсу; реверс художнього стилю та анімації; реверс рівнів. Реверс-інжиніринг «відчуттів»: реверс на основі досвіду користувача; звуковий реверс; реверс стилю; реверс за загальним враженням. Реверс-інжиніринг апаратного забезпечення. Реверс-інжиніринг за тетрадою елементів. Реверс-інжиніринг на основі AERM-таблиці. Літ.: [1] с.89-114; [4] с.56-70; ; [5] с.789-1001; [6] с.44-61.	2
3	**Статичний реверс-інжиніринг.** Введення в статичний реверс-інжиніринг. Базові статичні методики. Особливості статичного реверс-інжинірингу. Інструменти статичного реверс-інжинірингу. Літ.: [1] с.191-264, 413-417; [2] с.95-114; [7] с.55-62.	2
4	**Динамічний реверс-інжиніринг**. Поняття динамічного реверс-інжинірингу. Пісочниці як інструмент динамічного реверс-інжинірингу. Запуск шкідливих програм під динамічний реверс-інжиніринг. Моніторинг за допомогою Process Monitor. Літ.: [1] с.265-308; [2] с.115-133; [4] с.32-45; [7] с.63-72.	2
	Тема 2. Реверс-інжиніринг і шкідливе програмне забезпечення	
5	**Упаковане і обфуційоване шкідливе програмне забезпечення.** Методології проектування ПЗ. Характеристики методологій проектування. Стратегії конструювання ПЗ. Адаптивність процесу конструювання. Вибір методології проектування. Літ.: [9] с.30-64.	2
6	**Дослідження шкідливих програм в віртуальних машинах.** Структура віртуальної машини. Запуск віртуальної машини для аналізу шкідливого ПЗ. Налаштування спільної мережі Vmware. Використання декількох віртуальних машин для аналізу шкідливого ПЗ. Літ.: [11] с.133-150,	2
7	**Віртуальні машини і реверс-інжиніринг в задачах безпеки.** Використання віртуальної машини для аналізу безпеки. Підключення VMware до Інтернету і реверс-інжиніринг. Ризики при використанні VMware для аналізу безпеки. Літ.: [11] с.151-174.	2
8	**Асемблер для архітектури X86 і реверс-інжиніринг.** Комп'ютерна архітектура і організація програмного коду. Рівні програмного коду. Архітектура x86 і її вплив на задачі реверс-інжинірингу. Асемблер для архітектури X86. Літ.: [1] с.373-412; [4] с.46-55; [5] с.2-30; [6] с.31-43.	2
9	**Розпізнавання конструкцій мови С в асемблері.** Загальний аналіз задачі розпізнавання конструкцій мови С в асемблері. Дизасемблювання локальних і глобальних змінних мови С. Дизасемблювання арифметичних операцій мови С. Розпізнавання виразів if мови С в асемблері. Розпізнавання циклів в ході дизасемблювання мови С. Літ.: [3] с.49-67; [6] с.31-43	2
10	**Шкідливе програмне забезпечення Windows-систем – загальний аналіз.** Windows і її вразливості для шкідливого ПЗ. Дескриптори Windows.	2

	Спеціальні та загальні файли і їх використання шкідливим ПЗ. Файли, доступні через простор імен, та їх використання шкідливим ПЗ. Альтернативні потоки даних і шкідливе ПЗ. Реєстр Windows як джерело даних шкідливого ПЗ, поширені функції для роботи з реєстром. API для роботи з мережею - використання шкідливим ПЗ. Літ.: [2] с.35-84; [3] с.12-24; [8] с.7-48 .	
11	**Інструменти-аналізатори на рівні асемблера (OllyDbg).** OllyDbg як аналізатор на рівні асемблера. Завантаження шкідливого ПЗ. Підключення до запущеного процесу. Карта пам'яті – аналіз з OllyDbg. Перегляд потоків і стеків з OllyDbg. Літ.: [10] с.84-92.	2
12	**Налагодження ядра операційної системи.** Драйвери і код ядра. Підготовка до налагодження ядра. Налагоджувальник WinDbg від Microsoft. Налаштування символів Windows. Літ.: [3] с.5-11; [9] с.180-192.	2
Тема 3. Методи протидії реверс-інжинірингу		
13	**Антидизасемблювання.** Антидизасемблювання як процес. Дизасемблювання – загальний аналіз задачі. Лінійне дизасемблювання. Потокове дизасемблювання. методики антидизасемблювання. Літ.: [10] с.12-54.	2
14	**Антивідладка.** Антивідладка і шкідливе ПЗ. Виявлення відладчика в Windows. Антивідладка і виклики Windows API. Перевірка прапора ProcessHeap. Перевірка прапора NTGlobalFlag. Перевірка залишкових даних в системі. INT-сканування. Літ.: [10] с.55-74.	2
15	**Методи протидії віртуальним машинам в реалізації шкідливого ПЗ.** Ознаки присутності Vmware. Захист від пошуку слідів Vmware. Уразливі інструкції віртуальної машини. Використання методики Red Pill. Використання методики No Pill. Використання інструкції str. Інструкції анти-ВМ на платформі x86. Зміна налаштувань віртуальної машини. Втеча з віртуальної машини. Літ.: [11] с.201-210.	2
16	**Пакувальники і розпакування шкідливого програмного забезпечення.** Програми-пакувальники – призначення і анатомія пакувальника шкідливого ПЗ. Ознаки упакованої програми і розпізнавання упакованих програм. Способи розпакування програми: автоматизований статичний, автоматизований динамічний і ручний динамічний. [9] с.180-192	2
17	**Аналіз коду командної оболонки.** Визначення коду командної оболонки. Завантаження коду командної оболонки для аналізу. Визначення адреси виконання. Використання інструкцій call / pop. Використання інструкції fnstenv. Літ.: [9] с.93-102.	2
18	**Правові і етичні аспекти застосування реверс-інжинірингу.** Історія конфліктних прикладів реверс-інжинірингу. Реверс-інжиніринг і авторське право. Закони про авторське право України, Європи та США. Реверс-інжиніринг і етичні проблеми. Літ.: [6] с.29-30, 118-122; [12] с.12-54.	2
	Разом:	**36**

Зміст лабораторних робіт

№ з/п	Тема лабораторного заняття	Кількість годин
1	Виявлення та дослідження шкідливого програмного забезпечення. Літ.: [2] с.35-84; [3] с.12-48; [4] с.6-29;	6
2	Робота з системою реверс-інженерії програмного забезпечення IDAPro(Starter). Літ.: [2] с.95-114; [4] с.31; [Найкращі програми]	6
3	Динамічний аналіз шкідливого програмного забезпечення. Літ.: [2] с.115-133; [4] с.32-45; [Найкращі програми]	6
4	Дослідження шкідливих додатків Windows за допомогою Microsoft GINA. Літ.: [3] с.5-11; [Найкращі програми]	6
5	Відслідковування змін в реєстрі Windows. Літ.: [4] с.71-112; [Найкращі програми]	6
6	Аналіз хешів за допомогою Pwdump и Pass-the-Hash (PSH). Літ.: [Найкращі програми]	6
7	Аналіз шкідливого ПЗ на віртуальній машині. Літ.: [3] с.69-74;	6
8	Використання допоміжних утиліт реверс-інжинірингу Літ.: [Найкращі програми]	6
9	Підсумкове заняття. Тестування.	6
	Разом:	**54**

Зміст самостійної (у т.ч. індивідуальної) роботи

На самостійне опрацювання студентів виноситься опрацювання лекційного матеріалу, підготовка до виконання і захисту лабораторних робіт. Керівництво самостійною роботою та виконанням завдань здійснює викладач згідно з розкладом консультацій в позаурочний час, в тому числі із застосуванням технологій інтерактивного та дистанційного навчання.

Номер тижня	**Вид самостійної роботи**	**Кількість годин**
1	Опрацювання теоретичного матеріалу, підготовка до виконання ЛР1	9
2	Опрацювання теоретичного матеріалу, підготовка до захисту ЛР1	8
3	Опрацювання теоретичного матеріалу, підготовка до виконання ЛР2	9
4	Опрацювання теоретичного матеріалу, підготовка до захисту ЛР2	8
5	Опрацювання теоретичного матеріалу, підготовка до виконання ЛР3	9
6	Опрацювання теоретичного матеріалу, підготовка до захисту ЛР3	8
7	Опрацювання теоретичного матеріалу, підготовка до виконання ЛР4	9
8	Опрацювання теоретичного матеріалу, підготовка до захисту ЛР №4 Підготовка до тестування	8
9	Опрацювання теоретичного матеріалу, підготовка до виконання ЛР5	9
10	Опрацювання теоретичного матеріалу, підготовка до захисту ЛР5	8
11	Опрацювання теоретичного матеріалу, підготовка до виконання ЛР6	9
12	Опрацювання теоретичного матеріалу, підготовка до захисту ЛР6	8
13	Опрацювання теоретичного матеріалу, підготовка до виконання ЛР7	9
14	Опрацювання теоретичного матеріалу, підготовка до захисту ЛР7	8
15	Опрацювання теоретичного матеріалу, підготовка до виконання ЛР8	9
16	Опрацювання теоретичного матеріалу, підготовка до захисту ЛР8	8
17	Опрацювання теоретичного матеріалу. Підготовка до підсумкового заняття. Підготовка до тестування	9
18	Опрацювання теоретичного матеріалу.	5
	Разом:	**150**

Умовні позначення: ЛР – лабораторна робота

***** За чисельником / за знаменником (розрахунок здійснюється відповідно до розкладу лабораторнх занять)

ТЕХНОЛОГІЇ ТА МЕТОДИ НАВЧАННЯ

Процес навчання з дисципліни ґрунтується на використанні традиційних та сучасних методів, зокрема: лекції з використанням пояснювально-ілюстративних та проблемних методів і візуалізації; лабораторні роботи з використанням практичних, проблемних, продуктивних методів, тренінгових майстер-класів, з застосування інформаційно-комп'ютерних технологій.

Застосовувані при викладанні дисципліни методи навчання сприяють розвитку у студентів навичок soft skills: виконання частини лабораторних робіт передбачає роботу у малих групах з призначенням тім-лідера, що сприяє розвитку лідерських якостей у студентів, здатності до спілкування і організації командної роботи над спільними задачами, а змінюваність складу робочих груп між лабораторними роботами сприяє розвитку навичок адаптованості, гнучкості, комунікативності і оперативного налагоджування міжособистісних відносин в різних колективах; спілкування з проблемних питань під час лекцій, прилюдні захисти лабораторних робіт з обгрунтуванням прийнятих рішень щодо вибору методів рішення завдань в діалозі з викладачем і групою сприяють формуванню і удосконаленню вмінь публічних виступів, емпатичного слухання, відстоювання власної точки зору, самоаналізу і самокритики; адаптованість, вміння користуватися інтернет-ресурсами та іншими джерелами інформації, синтезувати та критично осмислювати інформацію з різних джерел передбачені специфікою дисципліни, що передбачає рішення проблемних завдань із застосуванням творчих підходів; обмежений час на виконання лабораторних робіт і тестових завдань, чітко визначені терміни проходження контрольних точок і відпрацювання заборгованостей сприяють розвитку пунктуальності, здатності до самоорганізації та управління часом (тайм-менеджменту).

МЕТОДИ КОНТРОЛЮ

Поточний контроль здійснюється під час лабораторних занять, а також у дні проведення контрольних заходів, встановлених робочим планом дисципліни.

При цьому використовуються такі методи поточного контролю:

- усне опитування;
- захист лабораторної роботи;
- тестування.

При виведенні підсумкової семестрової оцінки враховуються результати поточного контролю (залік за рейтингом формується автоматично за результатами поточного контролю).

ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ НАВЧАННЯ СТУДЕНТІВ

Оцінювання академічних досягнень студента здійснюється відповідно до «Положення про контроль і оцінювання результатів навчання здобувачів вищої освіти у ХНУ». Кожний вид роботи з дисципліни оцінюється за інституційною чотирибальною шкалою. Семестрова підсумкова оцінка визначається як середньозважена з усіх видів навчальної роботи, виконаних і зданих позитивно з урахуванням коефіцієнта вагомості. Вагові коефіцієнти змінюються залежно від структури дисципліни і важливості окремих видів її робіт.

Структурування дисципліни за видами робіт і оцінювання результатів навчання студентів у семестрі за ваговими коефіцієнтами

Аудиторна робота			**Контрольні заходи**	**Підсумковий контрольний захід**
Лабораторні роботи №:		Тестовий контроль 1	Тестовий контроль 2	Семестровий контроль
1 - 8		Т 1	Т 2-3	Залік
ВК:	0,6	0,2	0,2	

<u>Оцінювання лабораторних робіт</u>. Оцінка, яка виставляється за лабораторне заняття, складається з таких елементів: усне опитування студентів перед допуском до виконання лабораторної роботи; знання теоретичного матеріалу з теми; якість оформлення протоколу і звіту; вільне володіння студентом

спеціальною термінологією і уміння професійно обґрунтувати прийняті рішення; своєчасний захист лабораторної роботи.

Термін захисту звіту з лабораторної роботи вважається своєчасним, якщо студент захистив її в день виконання або на наступному після виконання роботи занятті. Пропущене лабораторне заняття студент зобов'язаний відпрацювати в лабораторіях кафедри у встановлений викладачем термін з реєстрацією у відповідному журналі кафедри, але не пізніше, ніж за два тижні до кінця теоретичних занять у семестрі.

Оцінку за лабораторне заняття викладач оголошує одразу після захисту звіту з лабораторної роботи і проставляє в електронний журнал дисципліни.

Оцінювання тестових завдань. Тематичний тест для кожного студента складається з п'ятнадцяти тестових завдань, кожне з яких оцінюється одним балом. Максимальна сума балів, яку може набрати студент, складає 15.

Відповідність набраних балів за тестове завдання оцінці, що виставляється студенту

Сума балів за тестове завдання	1–5	6–10	11–13	14–15
Оцінка за 4-ри бальною шкалою	2	3	4	5

На тестування відводиться 15 хвилин (для закритої форми тестів – по одній хвилині на кожне завдання). Правильні відповіді студент записує у талоні відповідей. При цьому усі графи для відповідей мають бути заповнені цифрами, що відповідають правильним, на погляд студента, відповідям. Через 15 хвилин студенти здають викладачу завдання з талонами відповідей.

Тестування студент може також пройти і в он-лайн режимі в модульному середовищі для навчання MOODLE.

Оцінку за тестування викладач проставляє в електронний журнал дисципліни.

Засвоєння студентом теоретичного матеріалу з дисципліни оцінюється за наведеними в таблиці критеріями оцінювання знань.

Критерії оцінювання знань студентів

Оцінка за інституційною шкалою	**Узагальнений критерій**
1	*2*
Відмінно	Студент глибоко і у повному обсязі опанував зміст навчального матеріалу, легко в ньому орієнтується і вміло використовує понятійний апарат; уміє пов'язувати теорію з практикою, вирішувати практичні завдання, впевнено висловлювати і обґрунтовувати свої судження. Відмінна оцінка передбачає, логічний виклад відповіді державною мовою (в усній або у письмовій формі), демонструє якісне оформлення роботи і володіння спеціальними інструментами. Студент не вагається при видозміні запитання, вміє робити детальні та узагальнюючі висновки. При відповіді допустив дві-три несуттєві похибки.
Добре	Студент виявив повне засвоєння навчального матеріалу, володіє понятійним апаратом і фаховою термінологією, орієнтується у вивченому матеріалі; свідомо використовує теоретичні знання для вирішення практичних задач; виклад відповіді грамотний, але у змісті і формі відповіді можуть мати місце окремі неточності, нечіткі формулювання закономірностей тощо. Відповідь студента будується на основі самостійного мислення. Студент у відповіді допустив дві–три несуттєві помилки.
Задовільно	Студент виявив знання основного програмного матеріалу в обсязі, необхідному для подальшого навчання та практичної діяльності за професією, справляється з виконанням практичних завдань, передбачених програмою. Як правило, відповідь студента будується на рівні репродуктивного мислення, студент має слабкі знання структури курсу, допускає неточності і суттєві помилки у відповіді, вагається при відповіді на видозмінене запитання. Разом з тим, набув навичок, необхідних для виконання нескладних практичних завдань, які відповідають мінімальним критеріям оцінювання і володіє знаннями, що дозволяють йому під керівництвом викладача усунути неточності у відповіді.
Незадовільно	Студент виявив розрізнені, безсистемні знання, не вміє виділяти головне і другорядне, допускається помилок у визначенні понять, перекручує їх зміст, хаотично і невпевнено викладає матеріал, не може використовувати знання при вирішенні практичних завдань. Як правило, оцінка "незадовільно" виставляється студенту, який не може продовжити навчання без додаткової роботи з вивчення дисципліни.

Якщо студент отримав негативну оцінку за певним видом робіт, то він має перездати її в установленому порядку, але обов'язково до терміну наступного контролю.

У випадку, коли студент не виконав індивідуальний план з дисципліни у заплановані терміни без поважних причин, то під час відпрацювання заборгованості при позитивній відповіді йому виставляється оцінка „задовільно".

Студент, який у встановлені терміни не виконав індивідуальний план поточної роботи з дисципліни повністю або частково, до здачі підсумкового контрольного заходу не допускається.

Залік вважається зданим при отриманні студентом за зведеними результатами поточного контролю підсумкової оцінки з дисципліни від 3,00 до 5,00 балів. При цьому за вітчизняною шкалою ставиться оцінка за двобальною шкалою, а за шкалою ECTS – оцінка, що відповідає набраній студентом кількості балів.

Підсумкова семестрова оцінка за інституційною шкалою і шкалою ЄКТС встановлюється в автоматизованому режимі після внесення викладачем усіх оцінок до електронного журналу.

Співвідношення вітчизняної шкали оцінювання і шкали оцінювання ЄКТС

Оцінка ЄКТС	**Інституційна інтервальна шкала балів**	**Інституційна оцінка, критерії оцінювання**		
A	4,75–5,00	5	Зараховано	***Відмінно*** – глибоке і повне опанування навчального матеріалу і виявлення відповідних умінь та навиків
B	4,25–4,74	4		***Добре*** – повне знання навчального матеріалу з кількома незначними помилками
C	3,75–4,24	4		***Добре*** – в загальному правильна відповідь з двома-трьома суттєвими помилками
D	3,25–3,74	3		***Задовільно*** – неповне опанування програмного матеріалу, але достатнє для практичної діяльності за професією
E	3,00–3,24	3		***Задовільно*** – неповне опанування програмного матеріалу, що задовольняє мінімальні критерії оцінювання
FX	2,00–2,99	2	Незарахо-вано	***Незадовільно*** – безсистемність одержаних знань і неможливість продовжити навчання без додаткових знань з дисципліни
F	0,00–1,99	2		***Незадовільно*** – необхідна серйозна подальша робота і повторне вивчення дисципліни

ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ ЗДОБУТИХ СТУДЕНТАМИ РЕЗУЛЬТАТІВ НАВЧАННЯ

1. Введення в реверс-інжиніринг (Reverse Engineering) - основні поняття та визначення.
2. Основні задачі і сфери застосуваня реверс-інжинірингу.
3. Реверс-інжиніринг коду і технічних підходів.
4. Реверс основної і мета-механіки ПЗ.
5. Реверс балансу.
6. Реверс на основі інтерфейсу.
7. Реверс художнього стилю та анімації.
8. Реверс рівнів.
9. Реверс-інжиніринг «відчуттів».
10. Реверс-інжиніринг апаратного забезпечення.
11. Реверс-інжиніринг за тетрадою елементів.
12. Реверс-інжиніринг на основі AERM-таблиці.
13. Введення в статичний реверс-інжиніринг.
14. Базові статичні методики.
15. Особливості статичного реверс-інжинірингу.
16. Інструменти статичного реверс-інжинірингу.
17. Поняття динамічного реверс-інжинірингу.
18. Пісочниці як інструмент динамічного реверс-інжинірингу.
19. Запуск шкідливих програм під динамічний реверс-інжиніринг.
20. Моніторинг за допомогою Process Monitor.
21. Упаковане і обфуційоване шкідливе програмне забезпечення.
22. Структура віртуальної машини.
23. Запуск віртуальної машини для аналізу шкідливого ПЗ.
24. Налаштування спільної мережі Vmware.
25. Використання декількох віртуальних машин для аналізу шкідливого ПЗ.
26. Використання віртуальної машини для аналізу безпеки.
27. Підключення VMware до Інтернету і реверс-інжиніринг.
28. Ризики при використанні VMware для аналізу безпеки.
29. Комп'ютерна архітектура і організація програмного коду.
30. Рівні програмного коду.
31. Архітектура x86 і її вплив на задачі реверс-інжинірингу.
32. Асемблер для архітектури X86.
33. Загальний аналіз задачі розпізнавання конструкцій мови С в асемблері.
34. Дизасемблювання локальних і глобальних змінних мови С.
35. Дизасемблювання арифметичних операцій мови С.
36. Розпізнавання виразів if мови С в асемблері.
37. Розпізнавання циклів в ході дизасемблювання мови С.
38. Windows і її вразливості для шкідливого ПЗ.
39. Дескриптори Windows.
40. Спеціальні та загальні файли і їх використання шкідливим ПЗ.
41. Файли, доступні через простор імен, та їх використання шкідливим ПЗ.
42. Альтернативні потоки даних і шкідливе ПЗ.
43. Реєстр Windows як джерело даних шкідливого ПЗ, поширені функції для роботи з реєстром.
44. API для роботи з мережею - використання шкідливим ПЗ.
45. OllyDbg як аналізатор на рівні асемблера.
46. Підключення OllyDbg до запущеного процесу.
47. Карта пам'яті – аналіз з OllyDbg.
48. Перегляд потоків і стеків з OllyDbg.
49. Драйвери і код ядра операційної системи.
50. Підготовка до налагодження ядра.
51. Налагоджувальник WinDbg від Microsoft.
52. Налаштування символів Windows.
53. Антидизасемблювання як процес.
54. Дизасемблювання – загальний аналіз задачі.
55. Лінійне дизасемблювання.
56. Потокове дизасемблювання.
57. Методики антидизасемблювання.

58. Антивідладка і шкідливе ПЗ.
59. Виявлення відладчика в Windows.
60. Антивідладка і виклики Windows API.
61. Перевірка прапора ProcessHeap.
62. Перевірка прапора NTGlobalFlag.
63. Перевірка залишкових даних в системі.
64. INT-сканування.
65. Ознаки присутності Vmware.
66. Захист від пошуку слідів Vmware.
67. Уразливі інструкції віртуальної машини.
68. Використання методики Red Pill.
69. Використання методики No Pill.
70. Використання інструкції str.
71. Інструкції анти-ВМ на платформі x86.
72. Зміна налаштувань віртуальної машини.
73. Втеча з віртуальної машини.
74. Програми-пакувальники – призначення і анатомія пакувальника шкідливого ПЗ.
75. Ознаки упакованої програми і розпізнавання упакованих програм.
76. Способи розпакування програми: автоматизований статичний, автоматизований динамічний і ручний динамічний.
77. Визначення коду командної оболонки.
78. Завантаження коду командної оболонки для аналізу.
79. Визначення адреси виконання.
80. Використання інструкцій call / pop.
81. Використання інструкції fnstenv.
82. Реверс-інжиніринг і авторське право.
83. Закони про авторське право України, Європи та США.
84. Реверс-інжиніринг і етичні проблеми.

МЕТОДИЧНЕ ЗАБЕЗПЕЧЕННЯ

Навчальний процес з дисципліни повністю і в достатній кількості забезпечений необхідною навчально-методичною літературою, розміщеною в електронному варіанті в модульному середовищі.

РЕКОМЕНДОВАНА ЛІТЕРАТУРА

Основна

1. Andriesse D. Practical Binary Analysis. Build Your Own Linux Tools for Binary Instrumentation, Analysis, and Disassembly. San Francisco : No Starch Press, Inc., 2019. 460p.
2. Bratus S., Matrosov A., Rodionov E. Rootkits and Bootkits: Reversing Modern Malware and Next Generation Threats. San Francisco : No Starch Press, Inc, 2019. 450p.
3. Bulazel A. Reverse Engineering Windows Defender's JavaScript Engine. REcon Brussels, 2018. 147p.
4. Bulazel A. Reverse Engineering Windows Defender's Antivirus Emulator. REcon Brussels, 2018. 225p.
5. Yurichev D. Reverse Engineering for Beginners. Creative Commons Attribution, 2017. 1069p.
6. Kowalski R. Penetration Testing and Reverse Engineering. ESD Cloud Media, 2017. 376p.
7. Franck De Goër de Herve. Reverse-engineering of binaries in a single execution. Universitй Grenoble Alpes, 2017. 252p.
8. Методи і алгоритми захисту інформаційних ресурсів комп'ютерних систем: навчальний посібник / В. М. Джулій, Ю. П. Кльоц, І. В. Муляр, В. М. Чешун. Хмельницький: ХмНУ, 2020. 196с.
9. Alhusain S. Intelligent Data-Driven Reverse Engineering of Software Design Patterns. A thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy. Montfort University, 2017. 220p.
10. Landman D. Reverse Engineering Source Code. UvA-DARE, 2017. 168p.
11. Selander D. Advanced Apple Debugging and Reverse Engineering. Razeware LLC, 2017. 475p.
12. Самойленко О. А. Виявлення та розслідування кіберзлочинів: навчально-методичний посібник. Одеса : Національний університет «Одеська юридична академія», 2020. 112с.

Додаткова

13. Інформаційна безпека : навчальний посібник / Ю. Я. Бобало та інші ; за заг. ред. д-ра техн. наук, проф. Ю. Я. Бобала та д-ра техн. наук, доц. І. В. Горбатого. Львів : Видавництво Львівської політехніки, 2019. 580с.
14. Тарнавський Ю. А. Технології захисту інформації: підручник. Київ : КПІ ім. Ігоря Сікорського, 2018. 162 с.
15. Найкращі програми для реверс-інжинірингу. URL: https://spy-soft.net/reverse-engineering-software/ (дата звернення: 11.01.2022).
16. Dang B., Gazet A., Bachaalany E. Practical Reverse Engineering: x86, x64, ARM, Windows. Indianapolis : John Wiley & Sons, Inc. , 2014. 383p.
17. Ra'Fat Ahmad AL. Reverse Engineering Feature Models From Software Variants to Build Software Product Lines. A thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy. Académie De Montpellier, 2014. 225p.
18. Найдьон Я. Поняття та класифікація віртуальних слідів кіберзлочинів. *Криміналістика*. 2019. №5. С. 304-307.
19. Микитишин А. Г., Митник М. М., Стухляк П. Д. Комплексна безпека інформаційних мережевих систем: навчальний посібник. Тернопіль: ТНТУ, 2016. 255 с.
20. Practical Information Security: A Competency-Based Education Course / [Izzat Alsmadi, Robert Burdwell, Ahmed Aleroud, Abdallah Wahbeh, Mahmoud Ali Al-Qudah, Ahmad Al-Omari]. Cham, Switzerland : Springer International Publishing AG, 2018. 328 p.

21. Про національну безпеку України: Закон України [Електронний ресурс] / Затверджено Указом Президента України від 21 червня 2018 року № 2469^Ш. URL: https://zakon.rada.gov.ua/laws/show/2469-19 (дата звернення: 11.01.2022).

22. Стратегія кібербезпеки України [Електронний ресурс] / Указ Президента України від 15.01.2016 р. № 96/2016 – URL: https://zakon5.rada.gov.Ua/laws/show/96/2016#n11 (дата звернення: 11.01.2022).

23. ДСТУ ISO/IEC 27032:2016 (ISO/IEC 27032:2012, IDT). Інформаційні технології. МЕТОДИ ЗАХИСТУ. Настанови щодо кібербезпеки. Чинний від 2016-27-12. Київ : ДП «УкрНДНЦ», 2018. 50 с.

24. Kim D. Fundamentals of information systems security / David Kim, Michael G. Solomon. – Third edition. – Burlington :Jones & Bartlett Learning, 2018. – 571 p.

25. Імовірнісні алгоритми та методи забезпечення безпеки IP-телефонії / А. В. Джулій, В. М. Джулій, В. М. Чешун, В. І. Чорненький // Міжнародний науково-технічний журнал «Вимірювальна та обчислювальна техніка в технологічних процесах» 2021. № 1. С. 133-138.

26. Sreeram Reddy, M Manzoor Hussain, K Srinivasa Rao. Latest Research on Reverse Engineering Technology. Proceedings of the International conference on Paradigms in Engineering & Technology (ICPET2016). Methodist College of Engineering & Technology, Hyderabad. P. 945-948.

27. Reverse engineering approach for improving the quality of mobile applications. / Eman K. Elsayed, Kamal A., Enas E. Naglaa E. Research article Software Engineering. August 19, 2019. P. 1-23.

28. Оціночні функції і метрики для виявлення помилок при тестуванні програмного забезпечення / І. В. Гурман, А. В. Джулій, В. М. Чешун, В. І. Чорненький // Міжнародний науково-технічний журнал «Вимірювальна та обчислювальна техніка в технологічних процесах», 2021. № 2. С. 97-102.

ІНФОРМАЦІЙНІ РЕСУРСИ

1. Модульне середовище для навчання MOODLE (розміщені усі необхідні матеріали з дисципліни, в тому числі завдання для поточного та семестрового контролю знань) Доступ до ресурсу: https://msn.khnu.km.ua.

2. Електронна бібліотека університету. Доступ до ресурсу: http://lib.khnu.km.ua/asp/php_f/p1age_lib.php.