МЕТОДИЧНІ ВКАЗІВКИ

ДО САМОСТІЙНОЇ РОБОТИ СТУДЕНТІВ

при вивченні дисципліни

«ЗАХИСТ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ»

спеціальності 125 «Кібербезпека»

(«Безпека інформаційних і комунікаційних систем»)

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра захисту інформації

МЕТОДИЧНІ ВКАЗІВКИ ДО САМОСТІЙНОЇ РОБОТИ СТУДЕНТІВ

при вивченні дисципліни

«ЗАХИСТ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ»

спеціальності 125 «Кібербезпека»
(«Безпека інформаційних і комунікаційних систем»)

Вінниця
ВНТУ
2018

Рекомендовано до друку Методичною радою Вінницького національного технічного університету Міністерства освіти і науки України (Протокол №7 від 28.03. 2018 р.)

Рецензенти:

О. П. Войтович, кандидат технічних наук, доцент
В. П. Майданюк, кандидат технічних наук, доцент

Методичні вказівки до самостійної роботи студентів при вивченні дисципліни «Захист програмного забезпечення» / уклад. В. А. Каплун, Ю. В. Баришев. – Вінниця: ВНТУ, 2018. – 36 с.

Методичні вказівки призначені для надання допомоги при самрстіному вивченні дисципліни "Захист програмного забезпечення". Самостійна робота студентів розбита на змістовні модулі, які містять перелік відповідних тем та основних понять по кожній темі, а також перелік практичних завдань для самостіного опрацювання та список контрольних запитань. Особливо дані методичні вказівки можуть стати у нагоді для студентів заочної форми навчання.

Навчальне самостійне електронне мережне видання

Методичні вказівки
до самостійної роботи студентів при вивченні дисципліни
«Захист програмного забезпечення»
для студентів галузі знань 12 «Інформаційні технології»
спеціальності 125 «Кібербезпека»

Укладачі:
Каплун Валентина Аполінаріївна
Баришев Юрій Володимирович

Макет підготовлено В. Каплун

Електронний ресурс PDF.
Підписано до видання 25.07.2018 р. Зам. № Р2018-012
Видавець та виготовлювач - Вінницький національний технічний університет,
Інформаційний редакційно-видавничий центр.
ВНТУ, ГНК, к.114, Хмельницьке шосе, 95, м. Вінниця, 21021,
тел. (0432) 65-18-06.
press.vntu.edu.ua;
Email: irvc.vntu@gmail.com.
Свідоцтво суб'єкта видавничої справи
серія ДК № 3516 від 01.07.2009 р.

ЗМІСТ

ВСТУП

З метоб більш повного і всебічного засвоєння навчального матеріалу студентами у робосій програмі дисципліни «Захист прграмного забезпечення» передбачено вид занять – самостійна робота. Даний вид засвоєння матеріалу передбачає опрацювання літературних джерел, вивчення публікацій фахових видань, присвячених різноманітним способам захисту програмного забезпечення, а також застосування практичнх навичок, здобутих на лабораторних заняттях, для вирішення конкретних задач у галузі захисту інформації.

Основний зміст самостійної роботи студентів полягає у вивченні та використанні системи знань у галузі теорії та практики застосування прграмного, програмно-апаратного забезпечення інформаційної безпеки у сфері професіональної та управлінської діяльності, у вивченні документів програмних комплексів, які застосовуються при виконанні лабораторних робіт, а також у вивченні та освоєнні методичних вказівок до виконання лабораторних робіт і аналізі відповідної додаткової літератури.

Основними завданнями вивчення дисципліни «Захист програмного забезпечення» є набуття систематизованої інформації про сучасний стан систем захисту програмного забезпечення (СЗПЗ), детальний аналіз методів зламу програмного забезпечення (ПЗ) та протидії їм; сучасні методи створення СЗПЗ, класифікацію засобів і методів захисту програмного забезпечення, основні алгоритми захисту, методи захисту від несанкціонованого доступу до програм, способи протидії несанкціонованому копіюванню, використанню, статичному вивченню та динамічному дослідженню захищених програм.

Самостійна робота студентів розділена на змістові модулі, що включають перелік відповідних тем. По кожній з тем пропонується відповісти на ряд контрольних запитань, а також виконати конкретні практичні завдання.

1 СИСТЕМИ ЗАХИСТУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА СПОСОБИ ЇХ ЗЛАМУ

Теоретичні питання

Тема	Основні питання	Джерела
Загальні положення СЗПЗ	Проблеми, що призвели до необхідності захисту ПЗ: промислове шпигунство; крадіжка і копіювання; несанкціонована модифікація ПЗ; піратство. Способи розповсюдження ПЗ: FreeWare, CommercialWare, ShareWare, TrialWare, DemoWare, CriptWare, CrippleWare, Nagware, AdWare, CardWare, MailWare та ін.	[1, с. 7-8] [2, с. 99–106]
Загальна класифікація СЗПЗ	Класифікація СЗПЗ за методом установлення: навісні; вбудовані; комбіновані. Класифікація СЗПЗ за використовуваними механізмами захисту: складні логічні механізми; використання шифрування ПЗ; комбіновані. Класифікація СЗПЗ за принципом функціонування: шифрувальники; пакувальники, СЗПЗ від несанкціонованого копіювання, СЗПЗ від несанкціонованого використання, СЗПЗ від несанкціонованого дослідження, СЗПЗ від дампінгу.	[1, с. 20-24]
Загальні методи та алгоритми захисту	Алгоритми заплутування. Алгоритми мутації. Алгоритми компресії даних. Алгоритми шифрування даних. Обчислення складних математичних виразів. Методи утруднення дизасемблювання. Методи утруднення налагодження. Емуляція процесорів та ОС. Нестандартні методи роботи з програмним та апаратним забезпеченням. Використання спеціалізованого ПЗ.	[1, с. 15–17]
Показники застосовності та критерії оцінки СЗПЗ	Показники застосовності СЗПЗ: технічні; економічні; організаційні. Критерії оцінювання СЗПЗ: захист як такий; стійкість до дослідження/зламу; відмовостійкість (надійність); незалежність від конкретних реалізацій ОС; сумісність; незручності для користувача ПЗ; побічні ефекти; вартість; доброякісність.	[1, с. 17–19; 2, с. 39–44]

Тема	Основні питання	Джерела
Засоби та інструменти для подолання систем захисту	Програмні оболонки ОС. Сервісні програми ОС. Програми статичного та динамічного аналізу. Програми статичної і динамічної модифікації. Програми розпакування, дешифрування та криптоаналізу. Апаратні засоби, що полегшують злам СЗПЗ.	[1, с. 31-45]

Завдання для самостійного виконання

1. Моніторинг роботи СЗПЗ [1, с. 5–12].
   - Знайти в мережі Інтернет програми для здійснення моніторингу за системами захисту програмного забезпечення (моніторинг звернень до системного реєстру – RegMon, до файлової системи – FileMon, мережевої активності – NetworkMon, використання клавіатури – KeyLogger, виклику API-функцій – APIMon).
   - Дослідити можливості цих програм і дати власну оцінку.

2. Пакування як метод захисту програм [1, с. 13-21; 2, с. 143–158].
   - Завантажити декілька програм-пакувальників: AsPack, WinPack, UPX, PeCompact або інші, програми ідентифікації пакування та програми-розпакувальники.
   - Підготувати 2-3 виконуваних файли різного обсягу і виконати пакування цих файлів різними пакувальниками. Порявняти якість пакування.
   - Використовуючи програми ідентифікації пакування, дослідити запаковані програми. Впевнитись, що спосіб пакування визначено правильно.
   - Використовуючи програми-розпакувальники, розпакувати отримані запаковані файли. Перевірити робото здатність розпакованих програм.

Контрольні запитання

1. Вказати мету і довести доцільність використання СЗПЗ.
2. Дати класифікацію систем захисту ПЗ за методом їх установлення та механізмом захисту, який вони використовують.
3. Дати загальну характеристику основним методам та алгоритмам захисту програмного забезпечення.
4. Охарактеризувати програми-пакувальники та програми-шифратори, навести їх сильні та слабкі сторони.
5. Дати загальну характеристику позитивних та негативних факторів систем захисту від несанкціонованого копіювання.

6. Навести перелік показників, які використовуються для розробки систем захисту програмного забезпечення.
7. Навести та дати пояснення основним критеріям, за якими оцінюються системи захисту програмного забезпечення.
8. Навести загальний порядок здійснення зламу захисту.
9. Охарактеризувати програми-монітори звернень до файлів, їх призначення.
10. Дати характеристику програмам стеження за системним реєстром, їх використанню для аналізу систем захисту.
11. Охарактеризувати програми-монітори API-викликів.
12. Як можуть бути використані програми моніторингу при дослідженні роботи систем захисту програмного забезпечення?
13. В чому різниця між архіваторами та пакувальниками?
14. Який принцип дії програм-пакувальників?
15. Які позитивні і негативні риси пакувальників?
16. Назвіть декілька програм для пакування виконуваних файлів.
17. Наведіть приклади програм для ідентифікації пакування файлів. Яке їх призначення?
18. Яке програмне забезпечення виконує злам програмних продуктів, захищених пакувальниками?
19. Назвіть основні методи, що їх використовують розпакувальники.

Література

1. Дудатьєв А. В. Захист програмного забезпечення. Ч. 1 : навчальний посібник / Дудатьєв А. В., Каплун В. А, Семеренко В. П. – Вінниця : ВНТУ, 2005. – 140 с.
2. Скляров Д. Искусство защиты и взлома информации. / Д. Скляров – СПб.: БХВ-Петербург, 2004. – 288 с.
3. Каплун В. А. Захист програмного забезпечення : лабораторний практикум / Каплун В. А., Дмитришин О. В., Баришев Ю. В. – Вінниця : ВНТУ, 2016. – 75 с.

2 ОСНОВНІ ПОНЯТТЯ ОПЕРАЦІЙНИХ СИСТЕМ, НЕОБХІДНІ ДЛЯ РЕАЛІЗАЦІЇ ЗАХИСТУ ПРОГРАМ

Теоретичні питання

Тема	Основні питання	Джерела
Основні поняття ОС	Функції ОС (підсистеми керування даними, пристроями, перефирійними пристроями, пам'яттю, процесам і задачами). BIOS (призначення BIOS, таблиця параметрів BIOS, версії BIOS). CMOS (призначення, функції для доступу до CMOS). Переривання (програмні і апаратні).	[1, с. 49–50]
Робота з дисками на фізич-ному рівні	Дискові пристрої, їх будова. Програмні засоби для визначення в програмах архітектури ЕОМ та характеристик основних пристроїв.	[2, с. 42–58] [1, с. 53–64]
Логічна структура диску	Таблиця розділів жорсткого диску. Доступ з програм до логічної структури дисків. Основні API-функції для отримання характеристик диску.	[1, с. 78–89]
Файлові системи	Файлова система FAT і її особливості. Файлова система NTFS. API-функції для роботи з файлами, дисками, каталогами тощо.	[2, с. 303–319] [1, с. 53–64]

Завдання для самостійного виконання

1. Знайти в мережі Інтернет і завантажити програму для зчитування вмісту головного завантажувального запису (наприклад, програму ReadMBR) і проаналізувати результати роботи цієї програми на прикладі власного комп'ютера:
   - програма початкового завантаження;
   - таблиця розділів диску;
   - логічна структура жорсткого диску.

2. Знайти в мережі Інтернет і завантажити програму для зчитування інформації на жорсткому диску (наприклад, програму DiskExplorer, - для файлових систем FAT і NTFS) і за допомогою цієї програми проаналізувати характеристики файлів на власному комп'ютері, таблицю атрибутів файлів (у fat-системах), таблицю метафайлів та файлові потоки (у NTFS).

3. За допомогою опції «Панелі керування» отримати характеристики параметрів власного комп'ютера і вказати серед них ті, які можна використати для прив'язки програми до параметрів комп'ютерної системи. Для цього можна використати програму SysInfo.

Контрольні запитання

1. Дати поняття логічних дисків та причини розбиття диску на розділи.
2. Охарактеризувати головний завантажувальний запис, його функції та складові.
3. Що таке таблиця розділів диску, що таке активні та неактивні розділи, первинні та розширені розділи?
4. Яку інформацію несуть елементи розділів диску і як їх можна використати для захисту інформації?
5. Охарактеризувати процес завантаження ОС і роль завантажувального запису.
6. Формат розширеного блоку параметрів BIOS.
7. Формат завантажувального запису для файлової системи FAT32.
8. Логічні номери секторів та переривання для доступу до логічних секторів.
9. Що таке кластери? Що являє собою ланцюжок кластерів, розподілених файлу? Як використати його для захисту?
10. Охарактеризувати кореневий каталог, розташування і розмір, його особливості для різних файлових систем.
11. Складові кореневого каталогу для різних файлових систем.
12. З чого складаються дескриптори файлів? Якими можуть бути атрибути файлів? Як використати їх для захисту?
13. Наведіть засоби для отримання довідкової інформації про дискову систему.
14. Охарактеризуйте засоби для роботи з каталогами файлової системи.
15. Дайте характеристику засобам для пошуку інформації у каталогах. Як це можна використати для побудови системи захисту?
16. Програмні засоби для роботи з файлами (записування, зчитування, позиціювання тощо).
17. Засоби для доступу до дескрипторів файлів і можливість їх зміни.

Література

1. Дудатьєв А. В. Захист програмного забезпечення. Ч. 1 : навчальний посібник / Дудатьєв А. В., Каплун В. А, Семеренко В. П. – Вінниця : ВНТУ, 2005. – 140 с.
2. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. / Э. Таненбаум, Х. Бос — СПб.: Питер, 2015. — 1120 с.

3 ЗАХИСТ ПРОГРАМ ВІД НЕСАНКЦІОНОВАНОГО КОПІЮВАННЯ І ВИКОРИСТАННЯ

Теоретичні питання

Тема	Основні питання	Джерела
Види та технології захисту від НСК	Нестандартне форматування дистрибутивного носія. Захист програм методом прив'язки до дистрибутивних носіїв інформації (інформація в проміжках; сектори з Bad CRC; фізичні дефекти на носії, Bad-сектори; неоднозначні біти; "пропалювання" під різними кутами. StarForce: захист від копіювання.	[1, с. 53–74] [1, с. 56–77] [2]
Захист інформації шляхом прив'язки до комп'ютера	Прив'язка до параметрів жорсткого диску. Прив'язка до BIOS. Прив'язка до архітектури, периферії. Прив'язка до швидкодії компонентів комп'ютера (периферійних пристроїв, швидкості роботи програми в цілому і окремих фрагментів зокрема). Прив'язка до логічної структури дистибутивного носія ПЗ. Прив'язка до системного програмного забезпечення комп'ютера. Основні функціїдля роботи з файловою системою.	[1, с. 108–117; 3] [4, с. 25-82]
Обмеження на використання ПЗ	Ідентифікація і автентифікація. Біометричні способи захисту. Захист шляхом ідентифікації за клавіатурним почерком.	[5]
Програмно-апаратні методи захисту	Електронні ключі (ЕК). Типи ЕК, принципи функціонування та можливості. Захист ПЗ за допомогою електронних ключів. Сучасні електронні ключів ідентифікції. Інші апартні способи захисту інформації.	[1, с. 117–138; 6, с. 133–140]
Обмеження на використання програмного забезпечення	Основні функції для роботи з реєстром. Використання реєстру для захисту програмного забезпечення. Захист за допомого демо-версій програм. Захист з використанням довідників.	[1, с. 136–139]

Завдання для самостійного виконання

1. Знайти в мережі Інтернет і завантажити програми для захисту шляхом різноманітних прив'язок (наприклад, програму Orien). Дослідити можливості захисту за допомогою цих програм, взявши в якості об'єктів захисту власні прогоами. Переконатись в ефективності використаного способу захисту – перевірити роботоздатність захищених програм на інших комп'ютерах.
2. Виконати розробку програми, що здійснює прив'язку до параметрів вінчестера (моделі, підмоделі і серійного номеру), параметрів «флешки» (назви, серійного номеру, файлової системи) та до параметрів процесора. Інформацію про параметри прив'язки записати у прихований системний файл [7, с. 22–32].
3. Виконати розробку програми, що здійснює захист шляхом перевірки дати і кількості запусків. Інформаці. Про дату і кількість запусків зберігати у системному реєстрі [7, с. 22–32].
4. Знайти в мережі Інтернет інформацію про сучасні апаратні способи захисту інформації вітчизняного виробництва: електронні ключі, ключі ідентифікації, криптографічні модулі, IP-шифратори, шлюзи захисту, смарткарти. Класифікувати знайдену інформацію за можливостями захисту і областю застосування.

Контрольні запитання

1. Навести характеристику сучасних технологій захисту програмного за безпечення від копіювання.
2. Охарактеризувати рівні роботи з дисковою системою комп'ютерів.
3. Навести перелік методів захисту від НСК шляхом прив'язки до дистрибутивного носія.
4. Що означає поняття нестандартного форматування і яким чином воно використовується для захисту від НСК? Навести приклади.
5. Що означає метод, що базується на опитуванні довідника? Що може бути довідником?
6. Вкажіть методи доступу до файлової системи комп'ютера.
7. Наведіть засоби для отримання довідкової інформації про дискову систему.
8. Охарактеризуйте засоби для роботи з каталогами файлової системи.
9. Дайте характеристику засобам для пошуку інформації у каталогах. Як це можна використати для побудови системи захисту?
10. Програмні засоби для роботи з файлами (записування, зчитування, позиціювання тощо).
11. Охарактеризувати методи прив'язки до вінчестера як захист від несанкціонованого копіювання.
12. Прив'язка до BIOS як метод захисту від копіювання.

13. Способи визначення параметрів системи. Дати коротку характеристику кожному з них.
14. Вимірювання продуктивності апаратури комп'ютера як метод захисту.
15. Що таке електронні ключі, в чому доцільність їх використання? Яка будова ЕК?
16. Охарактеризувати види електронних ключів.
17. Дати характеристику способів захисту ПЗ за допомогою електронного ключа.
18. Які методи зламу захистів за допомогою ЕК і способи протидії йому?
19. Як можна підвищити стійкість ЕК до зламу?
20. Охарактеризувати спектр можливостей електронних ключів та перспективи у їх розвитку.
21. Дати коротку характеристику сучасних відомих ключів захисту.
22. Наведіть правила використання електронних ключів та поясніть їх.
23. В чому полягає суть захисту за допомогою опитування довідників і які шліхи його здійснення?
24. Пояснити можливості захисту ПЗ за допомогою введення обмежень на використання програмного продукту.

Література

1. Дудатьєв А. В. Захист програмного забезпечення. Ч. 1 : навчальний посібник / Дудатьєв А. В., Каплун В. А, Семеренко В. П. – Вінниця : ВНТУ, 2005. – 140 с.
2. STAR FORCE. Разработчики программного обеспечения [Електронний ресурс]. – Режим доступу : URL : http://www.star-force.ru/solutions/ software-developers/– Назва з екрану.
3. Методи захисту програмного забезпечення від несанкціонованого копіювання [Електронний ресурс]. – Режим доступу : URL : http://www.studfiles.ru/preview/3905114 – Назва з екрану.
4. Румянцев П. В. Работа с файлами в Win 32 API / Павел Румянцев – [2-е изд., доп.] – М. : Горячая линия-Телеком, 2002. – 216 с.
5. Чередниченко В. Б. Біометричні методи у системах захисту інформації / В. Б. Чередниченко, К. Е. Чередниченко // Системи обробки інформації. – 2012. – Вип. 4(1). – С. 145-148. – Режим доступу: http://nbuv.gov.ua/ UJRN/soi_2012_1_4_34.
6. Скляров Д. В. Искусство защиты и взлома информации. — СПб. : БХВ – Петербург, 2004. - 288 с.
7. Каплун В. А. Захист програмного забезпечення : лабораторний практикум / Каплун В. А., Дмитришин О. В., Баришев Ю. В. – Вінниця : ВНТУ, 2016. – 75 с.

4 ЗАХИСТ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ВІД ДИНАМІЧНОГО ДОСЛІДЖЕННЯ

Теоретичні питання

Тема	Основні питання	Джерела
Базові поняття	Несанкціонований доступ (НСД). Система захисту від НСД. Злам програми. Основні принципи при реалізації систем захисту програм. Основні функції систем захисту від НСД. Основні методи дослідження: статистичні, статичні, динамічні, синтаксичні.	[2, с. 6–11]
Інструментарій зламника	Класифікація засобів дослідження. Декомпілятори і вузькоспеціалізовані налагоджувачі. Утиліти для розпакування та дампінгу процесів і задач. Їх призначення. Утиліти аналізу файлів. Шістнадцяткові редактори і редактори ресурсів. API-шпигуни та інші утиліти моніторингу. Інші утиліти і програми.	[1, с.32–43; 2, с. 6–11]
Вимоги до систем захисту від НСД	Доцільність та необхідність захисту від несанкціонованого налагодження. Вимоги до процесу і результату вбудовування захисних механізмів. Структура програм, захищених від дослідження: ініціалізатор, конфіденційна частина, деініціалізатор.	[1, с.7–10; 2, с. 11-12]
Програми для динамічного НСД	Налагоджувачі реального та захищеного режимів. Огляд і класифікація існуючих налагоджувальників: SoftIce, WinIce, OllyDbg тощо.	[2, с. 65–66]
Методи захисту від НСД	Способи виявлення налагоджувача реального режиму. Методи боротьби з налагоджувачами захищеного режиму. Рекомендації та методика “вбивання” конкретних налагоджувачів. Написання “триків” (пасток). Перехоплення викликів API-функцій з метою захисту від налагодження. Додаткові прийоми антиналагоджувального програмування.	[2, с. 66–73]
Використання хуків	Поняття хуків та фільтрувальних функцій. Типи хуків. Встановлення та зняття хуків. Опис функції-фільтра. Можливості хуків. Приклади застосування хуків.	[2, с. 75–83]

Завдання для самостійного виконання

1. Знайти в інтернеті і завантажити на свій комп'ютер налагоджувач OllyDbg, ознайомитись з його можливостями [3]
2. Використовуючи інтернет-джерела [4], здійснити динамічний злам запропонованих програм.
3. Зробити висновки щодо можливостей зламу програм і надати пропозиції щодо можливого посилення захисту програм від дослідження.
4. Використовуючи інтернет-джерела [5], розробити програму, яка здійснює перехоплення будь-яких повідомлень (за бажанням).

Контрольні запитання

1. В чому різниця між налагоджувачами реального і захищеного режимів?
2. Наведіть існуючі програми-налагоджувачі реального та захищеного режимів?
3. Яким чином можна виявити присутність в операційній системі налагоджувачів?
4. Які основні переривання використовують налагоджувачі?
5. Наведіть додаткові методи захисту від налагоджувачів реального режиму.
6. Які є методи захисту від дослідження програм під налагоджувачами захищеного режиму?
7. Яким чином реалізація еквівалентних гілок допоможе протистояти налагоджувачам?
8. В чому сутність захисту за допомогою засмічування коду?
9. Охарактеризуйте налагоджувач SoftIce.
10. В чому, на вашу думку, проявляються позитивні і негативні риси налагоджувачів?

Література

1. Дудатьєв А. В. Захист програмного забезпечення. Ч. 1 : навчальний посібник / Дудатьєв А. В., Каплун В. А, Семеренко В. П. – Вінниця : ВНТУ, 2005. – 140 с.
2. Каплун В.А., Баришев Ю. В., Дмитришин О. В.. Захист програмного забезпечення. Навчальний посібник. Частина 2. – Вінниця: ВНТУ, 2013. – 151 с.
3. OllyDbg. [Електронний ресурс]. – Режим доступу : URL : http://www.ollydbg.de/ – Назва з екрану.
4. Урок крэкинга с помощью OllyDbg. [Електронний ресурс]. – Режим доступу : URL : http://daxa.com.ua/article/num1/– Назва з екрану.
5. Методы перехвата API-вызовов в Win32. [Електронний ресурс]. – Режим доступу : URL : ttps://rsdn.org/article/baseserv/ – Назва з екрану.

5 ЗАХИСТ ПРОГРАМ ВІД СТАТИЧНОГО ДОСЛІДЖЕННЯ

Теоретичні питання

Тема	Основні питання	Джерела
Структура виконуваних файлів для ОС Windows	Структура exe-заголовків. Можливості вбудовування в exe-заголовки критичної інформації, необхідної для захисту програм. Механізми впровадження захисного коду в PE-файли. Поняття Х-коду. Види Х-коду і способи вбудовування. Деякі алгоритми впровадження. Запобіжні заходи при впровадженні Х-коду	[1, с. 16–28; 2, с. 14–112; 4, с. 204-248]
Статичне дослідження програм	Інструментарій хакера для статичного дослідження. Декомпілятори, дизасемблери, шістнадцяткові редактори та редактори ресурсів. Автоматичні та інтерактивні дизасемблери. Прикдади інтерактивних дизасемблерів.	[1, с. 13–16]
Основні методи для захисту від дизасемблювання	Доцільність та необхідність захисту від статичного дослідження. та їх класифікація. Основні пособи захисту від статичного дослідження: маніпулювання заголовками виконуваних файлів, пакування, шифрування, емуляція програмих середовищ. Етапи побудови систем захисту від дослідження програм.	[1, с. 16–28]
Обфускація як захист від статичного дослідження	Поняття обфускації. Види обфускації. Лексична обфускація, ЇЇ види і можливості реалізації. Обфускація даних, її види та можливості реалізації. Вимоги до обфускації програм. Сучасний програмний ринок обфускаторів. Обфускація графа потоку керування: маніпулювання функціями, використання непрозорих предикатів, внесення недосяжного, мертвого або надлишкового коду, зчеплення дуг, клонування базових блоків, використання різноманітних перетворень циклів. Додаткові методи утруднення логіки захищуваних програм та способи їх реалізації. Використання засобів мови асемблера для маскування фрагментів захисту. Емуляція процесора та мультизадачності як способи протистояння статичному вивченню програм.	[1, с. 30–64]

Завдання для самостійного виконання

1. Дослідження структури виконуваних файлів [5, с.33-37; 7, с. 11-71].
   1.1. Підготувати для дослідження декілька виконуваних файлів різного розміру, упакувати ці файли різними пакувальниками і зберегти їх.
   1.2. За допомогою програм *Hiew* та *PE Explorer* проаналізувати структуру підготовлених виконуваних файлів. Зробити висновки щодо того, яким чином у виконувані файли вбудовують фрагменти захисту програми-пакувальники.
2. Дослідження роботи декомпіляторів [5, с. 38-46].
   2.1. Знайти в Інтернеті працездатний декомпілятор з мови C/C++ (наприклад, recStud), декомпілювати програми, написані мовою C++ (знайти в інтернеті або використати роботи, виконані в ході лабораторних робіт з дисципліни «Технологія програмування». Оцінити якість декомпілювання, порівнюючи згенерований код із початковим (для цього вибрати фрагмент початкового коду і знайти відповідний фрагмент у згенерованому декомпілятором).
   2.2. Знайти в Інтернеті працездатний декомпілятор з мови Visual Basic (наприклад, VbDecompiler і декілька програм, створених у Visual Basic. Оцінити якість декомпілювання, порівняти якість декомпіляції програм, відкомпільованих у native code і у p-code.
   2.3. Знайти в Інтернеті працездатний декомпілятор з мови Java (наприклад, DecafePro) і взяти 2-3 програми, створені інтерпретатором Java (у форматі .jar або .class). Оцінити якість декомпілювання, порівнюючи згенерований код із початковим (для цього вибрати фрагмент початкового коду і знайти відповідний фрагмент у генерованому декомпілятором).
   2.4. Знайти в Інтернеті працездатний декомпілятор з мови C#, знайти програми, реалізовані мовою C#, декомпілювати їх. Оцінити якість декомпілювання, порівнюючи згенерований код із початковим (для цього вибрати фрагмент початкового коду і знайти відповідний фрагмент у генерованому декомпілятором).
   2.5. Зробити загальний висновок про якість декомпіляторів різних мов програмування. Надати рекомендації щодо методів захисту програм, реалізованих різними мовами.
3. Дослідження роботи редакторів ресурсів [5, с. 38-46].
   3.1. Знайти в Інтернеті працездатний редактор ресурсів (PeExplorer, PeTuner) і ознайомитись з ними, дослідити їх можливості. За допомогою редактора ресурсів дослідити програми-об’єкти, внести деякі зміни і створити нові виконувані файл.
   3.2. Зробити висновок щодо захисту програмних засобів від модифікації засобами для редагування ресурсів.

4. Дизасемблювання виконуваних файлів [5, с. 48–62; 6, с. 187–262].
   4.1. Дослідити інтерфейс і можливості програм для дослідження і модифікації виконуваних кодів: IDAPro та HIEW.
   4.2. Знайти декілька програм, захищених паролем, обмеженням кількості запусків тощо. Користуючись вищенаведеними програмами, навчитись дизасемблювати програми.
   4.3. Дизасемблювати код програми і відредагувати його таким чином, щоб зняти захист.
   4.4. Зробити висновки щодо ефективності даного виду захисту: причини недостатньої стійкості та рекомендації щодо його покращення.

Контрольні запитання

1. В чому доцільність і необхідність захисту від статичного дослідження?
2. Класифікуйте методи захисту від статичного дослідження.
3. В чому сутність методу емуляції процесора з точки зору захисту програмного забезпечення?
4. Як може допомогти емуляція мультизадачності у боротьбі зі статичним дослідженням?
5. Охарактеризуйте структуру виконуваних файлів.
6. Що таке заголовки виконуваних файлів, які є їх види? Для чого їх використовує операційна система?
7. Що таке таблиця об'єктів (розділів виконуваного файлу), які об'єкти існують у програмах від різних виробників?
8. Які є способи впровадження захисних механізмів у виконувані файли?
9. Дайте поняття Х-коду та наведіть вимоги до нього.
10. Наведіть приклади впровадження Х-коду у заголовки виконуваних файлів. За якими алгоритмами вони здійснюються?
11. Що розуміють під терміном "обфускація"? Які її види ви знаєте?
12. Наведіть основні кроки для здійснення лексичної обфускації.
13. Які існують види обфускації даних?
14. Що таке обфускація графа потоку керування?
15. Що означає поняття маскування програми? Які його етапи?
16. Які маніпулювання функціями допомагають захиститись від несанкціонованого дослідження?
17. Що таке непрозорі предикати? Наведіть приклади.
18. Як допомагає внесення недосяжного, мертвого або надлишкового коду захиститись від дослідження?
19. В чому сутність зчеплення дуг?
20. Що таке клонування базових блоків?
21. Які перетворення циклів допомагають запобіганню несанкціонованого дослідження програм?
22. Яка різниця між автоматичними та інтерактивними дизасемблерами?
23. Які методи боротьби з автоматичними дизасемблерами ви знаєте?

24. Охарактеризуйте особливі методи боротьби з інтерактивними дизасемблерами.
25. Поясність, що таке "динамічний фуфель" як прийом для захисту від дослідження.
26. В чому сутність методології емуляції процесора?
27. Які є шляхи реалізації емуляції багатозадачності для захисту від дослідження?
28. В чому полягає паралельне виконання гілок задачі з точки зору емуляції мультизадачності?
29. Які позитивні і негативні риси методів протистояння дизасемблерам ви можете назвати?

Література

1. Каплун В. А., Баришев Ю. В., Дмитришин О. В.. Захист програмного забезпечення. Навчальний посібник. Частина 2. – Вінниця: ВНТУ, 2013. – 151 с.
2. Румянцев П. В. Исследование программ Win32: до дизассемблера и отладчика – М.: Горячая линия-Телеком, 2009. – 367 с.
3. Касперски К. Компьютерные вирусы изнутри и снаружи / Крис Касперски – СПб. : Питер, 2006. – 527 с.
4. Обфускация и защита программных продуктов. [Електронний ресурс]. – Режим доступу : URL : http://citforum.ck.ua/security/articles/obfus/ – Назва з екрану.
5. Каплун В. А. Захист програмного забезпечення : лабораторний практикум / Каплун В. А., Дмитришин О. В., Баришев Ю. В. – Вінниця : ВНТУ, 2016. – 75 с.
6. Румянцев П. В. Исследование программ Win32: до дизассемблера и отладчика / Румянцев П. В. – М. : Горячая линия-Телеком, 2004. – 367 с.
7. Касперски К. Техника и философия хакерских атак / Крис Касперски. – М. : Солон-Пресс, 2004. – 272 с.

6 ЗАХИСТ ПРОГРАМ ВІД ЗНЯТТЯ З ПАМ'ЯТІ

Теоретичні питання

Тема	Основні питання	Джерела
Поняття дампінгу	Адресний простір програм в оперативній пам'яті. Порядок завантаження програми і виділення пам'яті процесу. Доступ до пам'яті та списку процесів. Отримання дампу пам'яті обраного процесу. Зняття програм з пам'яті.	[1, с.84-88]
Методи захисту від дампінгу програм.	Програми для зняття дампу і захист від них. Антидампінг у нульовому кільці. Динамічне розпаковування.	[1, с.89-96]

Завдання для самостійного виконання

1. Ознайомитись з програмами для зняття дампу процесів і задач, завантаживши їх з інтернету (PETOOLS, OllyDump, LordPE Delux, Process Dumper або інші).
2. Підібрати декілька програм і спобувати зняти повний або частковий дамп пам'яті цих програм.
3. Зробити висновки щодо вибору методів захисту від дампінгу.

Контрольні запитання

1. Що таке дамп пам'яті?
2. Який порядок завантаження програми і виділення пам'яті процесу?
3. Як отримати дамп отриманого процесу?
4. Які методи захисту від дампінгу ви можете назвати? Охарактеризуйте їх.
5. Які програми для зняття програм з пам'яті ви знаєте і як від них захиститись?

Література

1. Дудатьєв А. В. Захист програмного забезпечення. Ч. 1 : навчальний посібник / Дудатьєв А. В., Каплун В. А, Семеренко В. П. – Вінниця : ВНТУ, 2005. – 140 с.

ГЛОСАРІЙ

Несанкціонований доступ (НСД) – ***unauthorized access*** – нелегальні дії щодо використання, зміни та знищення виконуваних модулів.

Система захисту від несанкціонованого доступу (system of protection against unauthorized access) – комплекс програмних засобів, що забезпечують ускладнення або заборону нелегального розповсюдження, використання і/або зміну програмних продуктів.

Надійність системи захисту (reliability of protection) – здатність протистояти спробам проникнення в алгоритм її роботи і обходу механізмів захисту.

Зломом програми (breaking program) – порушення функціональності об'єктів захисту програмного забезпечення.

Обфускація (obfuscation) – це один з методів захисту програмного коду, який дозволяє ускладнити процес реверсивної інженерії коду програмного продукту, що захищається.

Хук (Hook) – механізм перехоплення особливою функцією подій (таких, наприклад, як повідомлення від маніпулятора миші або клавіатури) до того, як вони дійдуть до програмного додатка.

Динамічне дешифрування – дешифруваня у міру виконання конфіденційної частини програми.

Relocation Table – таблиці для настроювання адрес. Складається зі значень у форматі <сегмент : зсув>. До зсувів у завантажувальному модулі, на яких указують значення в таблиці, після завантаження програми в пам'ять повинна бути додана сегментна адреса, з якої завантажена програма.

Таблиця об'єктів або таблиця розділів (object) – сукупність даних певного призначення: про експортовані та імпортовані функції, про ресурси, про переміщення (relocations) і т. д., які компактно розміщені у виконуваному файлі.

Х-код – частина коду програми, яку ми збираємося впроваджувати у програму з метою її захисту.

Лексична обфускація полягає в форматуванні коду програми, зміні його структури таким чином, щоб він став нечитабельним, менш інформативним і важчим для дослідження дизасемблерами і декомпіляторами.

Символьна обфускація – заміна імен ідентифікаторів (імен змінних, масивів, структур, функцій, процедур і т. д.) на самовільні довгі набори символів, які важко сприймати людині.

Обфускація даних – ускладнення розуміння даних програми, пов'язане із трансформацією структур даних.

Маскування програми – це таке перетворення її тексту, яке повністю зберігає функціональність, але робить розуміння, зворотну інженерію і модифікацію тексту програми завданням неприйнятно високої вартості.

Змінна є непрозорою, якщо існує деяка властивість щодо цієї змінної, яка відома в момент заплутування програми, але важко відновлюється після того, як заплутування завершене.

Непрозорі предикати (opaque predicates) – вираз, значення якого відоме в момент заплутування програми, але важко відновлюване після його завершення.

Недосяжний *код (unreachable code)* – фрагмент програми, що ніколи не виконується.

Мертвий *код (dead code)* – фрагмент коду, що у програмі виконується, але його виконання ніяк не впливає на результат роботи.

Надлишковий код (redundant code) – фрагмент коду, що виконується, і результат його виконання використовується надалі в програмі, але такий код можна спростити або зовсім видалити, оскільки обчислюється або константне значення, або значення, уже обчислене раніше.

Автоматичні дизасемблери – програмні засоби, що аналізують код виконуваного файлу й формують відповідний йому вихідний текст або лістинг.

Інтерактивні дизасемблери – програмні засоби, що формують вихідний текст/лістинг виконуваного коду програми так само, як це роблять автоматичні дизасемблери, однак відрізняються наявністю потужного користувацького інтерфейсу, що значно полегшує аналіз дизасемблерної програми.

Дамп пам'яті (memory dump) – це копія вмісту оперативної пам'яті, що знаходиться на жорсткому диску або іншому енергонезалежному пристрої пам'яті.

Дампери (dampers) – програми для знання дампу.

Навчальне видання

Валентина Аполінаріївна Каплун
Юрій Володимирович Баришев

МЕТОДИЧНІ ВКАЗІВКИ ДО САМОСТІЙНОЇ РОБОТИ СТУДЕНТІВ при вивченні дисципліни "ЗАХИСТ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ" для студентів галузі знань 12 «Інформаційні технології» спеціальності 125 «Кібербезпека»

Редактор В. Дружиніна

Укладачі: Каплун Валентина Аполінаріївна
Баришев Юрій Володимирович

Оригінал-макет підготовлено В. Каплун