Міністерство освіти і науки України

Національний технічний університет «ХПІ»

Навчально-науковий інститут комп’ютерних наук та інформаційних технологій

Кафедра комп’ютерної інженерії та програмування

**ЗВІТ**

з лабораторної роботи

з дисципліни «СУЧАСНІ ТЕХНОЛОГІЇ БЕЗПЕЧНОГО ПРОГРАМУВАННЯ»

**«Захист від зміни бінарного файлу»**

Виконав: студент групи КН-М922б

\_Кліщов Б. Р.\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(прізвище, ініціали)

Перевірив: доцент кафедри

\_Бульба С. С \_\_\_\_\_\_\_\_\_\_

(прізвище, ініціали)

Харків - 2022

**Мета:** Навчитися підписувати виконувані файли.

**Завдання:**

* створити сертифікат
* проінсталювати його в систему, щоб він був "довіреним"
* використовуючи проект будь-якої попередньої роботи, виконати підпис виконуваного файлу за допомогою утиліти SignTool (або JarSigner)
* виконати верифікацію підпису (бажано на рівні самого кода при завантаженні додатка):
  + чи є підписаний сертифікат валідним
  + чи не було (бінарної) зміни файлу та його код цілісний

P.S. Лабораторна робота розрахована для windows-виконуючих файлів або java файлів. Але якщо робота будет зроблена для Linux/Macos виконуючих балів - це вам плюс до карми та можливість отримати додаткові бали.

**Хід роботи**

**Ключові елементи програми:**

func signFile(priKey string, filePath string) error {

file, err := ioutil.ReadFile(filePath)

if err != nil {

return err

}

hash := sha256.Sum256(file)

signature, err := rsa.PriKeyEncrypt(string(hash[:]), priKey)

if err != nil {

return err

}

created, err := os.Create("./sha.signature")

if err != nil {

return err

}

\_, err = created.Write([]byte(signature))

if err != nil {

return err

}

return nil

}

func verifySignature(pubKey string, signature string, filePath string) (bool, error) {

file, err := ioutil.ReadFile(filePath)

if err != nil {

return false, err

}

hash := sha256.Sum256(file)

decrypt, err := rsa.PublicDecrypt(signature, pubKey)

if err != nil {

return false, err

}

if decrypt != string(hash[:]) {

return false, nil

}

return true, nil

}

**Фрагмент основної програми:**

func main() {

priKey, err := ioutil.ReadFile("./keys/key.pem")

if err != nil {

log.Fatal(err)

}

pubKey, err := ioutil.ReadFile("./keys/key.pub")

if err != nil {

log.Fatal(err)

}

err = signFile(string(priKey), "./sha")

if err != nil {

log.Fatal(err)

}

signature, err := ioutil.ReadFile("./sha.signature")

if err != nil {

log.Fatal(err)

}

isValid, err := verifySignature(string(pubKey), string(signature), "./sha")

if err != nil {

return

}

if isValid {

fmt.Println("the program is valid")

} else {

fmt.Println("the program invalid")

}

}

**Результати роботи програми**

****

Рисунок 1 – Результат роботи програми

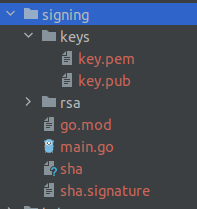


Рисунок 2 – Структура програми

**Висновки:** Навчився підписувати виконувані файли