МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Спеціальність: 123 «Комп'ютерна інженерія»

Освітня програма: «Комп'ютерна інженерія»

Група: 2БКС-26

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

здобувача освіти денної форми навчання

БКС.26.14.000.КРБ

ПРОТАСЕНКО
ОЛЕКСАНДР ІГОРОВИЧ

м. Одеса
2022 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Спеціальність: **123 «Комп'ютерна інженерія»**
Освітня програма: **«Комп'ютерна інженерія»**
Група: **2БКС-26**

ПОЯСНЮВАЛЬНА ЗАПИСКА

До кваліфікаційної роботи бакалавра на тему: **«Аналіз та практична реалізація методів захисту каналів зв'язку пристроїв клавіатурного введення»**

Проектний матеріал складається з пояснювальної записки на 61 сторінках та графічного (презентаційного) матеріалу на 15 аркушах (слайдах)

Виконавець _____ (Протасенко О.І.)
Керівник проекту_____(Кривченко Ю.В.)

Консультанти:
з охорони праці_____ (Чорновол Н.І.)
з дотримання вимог ЄСКД _____ (Петрашова В.І.)
старший консультант _____ (Скорнякова О.В.)

До захисту допущений
Завідувачка кафедри _____ (Іванова Л.В.)
Завідувач відділення _____ (Суліма Ю.Ю.)

Захист «____»_____ 202____ р. Протокол ДКК №_____
Оцінка ДКК _____

Секретар ДКК _____

АНОТАЦІЯ

Метою даної роботи є дослідження методів протидії стеження за клавіатурним вводом та принципів функціонування клавіатурних шпигунів, а також практична реалізація деяких з них.

Робота містить аналіз методів протидії програмам-шпигунам, практичну реалізацію деяких методів протидії клавіатурним шпигунам, огляд методів стеження за клавіатурним введенням.

Досліджено методи та технології атаки на існуючі бездротові пристрої введення інформації та способи захисту від них. Визначено можливості сучасних пристроїв для аналізу захисту бездротових пристроїв введення інформації.

Описано модель апаратного вводу системи Windows. Виконано огляд пристроїв для фіксації символів, набраних з клавіатури. Проаналізовані можливі зони апаратного перехоплення інформації. Описано функціональну схему апаратного кейлогера. Визначено загальну функціональність для клавіатур різних типів. Описано процес перехоплення функцій GetMessage та PeekMessage.

У якості прикладу використання описаних методів захисту виконано розробку утиліти для захисту від клавіатурних шпигунів, реалізовано кейлогер та антикейлогер. Кінцевим програмним продуктом може бути оболонка, яка представляє у значній мірі автоматизований інтерфейс для реалізації методів протидії клавіатурним шпигунам.

Реалізовано електричну принципову схему активного захисту бездротових пристроїв введення інформації на базі мікроконтролеру та апаратного генератору випадкових чисел. Розроблено алгоритм та програмне забезпечення на мові програмування С для пристрою захисту. Проведено аналіз роботи пристрою захисту каналу зв’язку.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Відділення *Комп'ютерних систем* Кафедра *Комп'ютерної інженерії*

Освітньо-професійна програма *«Комп'ютерна інженерія»*

Спеціальність *123 «Комп'ютерна інженерія»*

ЗАТВЕРДЖУЮ:

Заст. дир. з НВР *Беркань І.В.*

"_____" _____202____р.

ЗАВДАННЯ

на кваліфікаційну роботу бакалавра

Здобувачеві (здобувачці) освіти *Протасенку Олександру Ігоровичу*

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи *Аналіз та практична реалізація методів захисту каналів зв'язку пристроїв клавіатурного введення*

затверджена наказом по коледжу від "____"_____202____р. №_____

2. Термін здачі кваліфікаційної роботи_____ _____

3. Вихідні дані до роботи *1. Вимоги до інформаційної безпеки; 2. Характеристики і перелік відомих клавіатурних шпигунів; 3. Модель апаратного вводу ОС Windows; 4. Розробити модулі кейлогера та антикейлогера; 5. Проаналізувати роботу клавіатурних шпигунів на базі руткіт-технології у UserMode та KernelMode; 6. Реалізувати схему пристрою захисту бездротового каналу зв'язку з клавіатурою та проаналізувати його ефективність*

4. Зміст розрахунково-пояснювальної записки (перелік питань, які необхідно розробити)

Огляд методів стеження за клавіатурним введенням

Аналіз методів захисту пристроїв клавіатурного введення

Практична реалізація методів протидії клавіатурним шпигунам

Практична реалізація пристрою захисту каналу бездротового зв'язку

Охорона праці

5. Перелік графічного (презентаційного) матеріалу (з точним зазначенням обов'язкових креслень, кількості слайдів)

Рейтинг загроз інформаційної безпеки, Модель апаратного вводу системи Windows, Кейлогери для фіксації символів, набраних з клавіатури, Можливі зони апаратного перехоплення інформації, Функціональна схема апаратного кейлогера, Підключення драйвера-фільтра до стека клавіатурного драйверу, Перехоплення функцій GetMessage і PeekMessage, Перехоплювач на базі руткіт-технології в KernelMode, Ілюстрація затримок при наборі на клавіатурі, Результати аналізу натискання клавіш за допомогою застосунку gqrx, Принципова електрична схема пристрою захисту каналу бездротового зв'язку з клавіатурою

6. Консультанти по кваліфікаційній роботі, із зазначенням розділів роботи, що стосується їх

Розділ	Консультант	ПІДПИС	
		Завдання видав	Завдання прийняв
Технологічний	*Кривченко Ю.В.*		
Охорона праці	*Чорновол Н.І.*		
Нормоконтроль	*Петрашова В.І.*		
Старший консультант	*Скорнякова О.В.*		

7. Дата видачі завдання _____

Керівник роботи *Кривченко Ю.В.* _____
(підпис)

Завдання прийняв до виконання _____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/р	Назва етапів кваліфікаційної роботи	Термін виконання етапів кваліфікаційної роботи	Примітка
1.	*Вступ. Постановка задач розробки*	*5.05.2022*	
2.	*Огляд методів стеження за клавіатурним*	*7.05.2022*	
	вводом		
3.	*Аналітичний огляд небезпек для*	*11.05.2022*	
	комп'ютерних систем		
4.	*Аналіз принципів стеження за клавіатурним*	*13.05.2022*	
	вводом		
5.	*Огляд апаратних клавіатурних шпигунів*	*16.05.2022*	
6.	*Дослідження клавіатурного шпигуна на базі*	*18.05.2022*	
	драйвера		
7.	*Аналіз програмних заходів захисту*	*20.05.2022*	
8.	*Розробка пристрою захисту та його ПЗ*	*23.05.2022*	
9.	*Практична реалізація методів протидії*	*27.05.2022*	
	шпигунам		
10.	*Аналіз роботи пристрою захисту введення*	*30.05.2022*	
11.	*Реалізація та опис кейлогера та*	*3.06.2022*	
	антикейлогера		
12.	*Аналіз результатів, підготовка слайдів*	*6.06.2022*	
	презентації		
13.	*Розробка питань з охорони праці*	*8.06.2022*	
14.	*Підготовка до захисту*	*10.06.2022*	

Виконавець _____
(підпис)

Керівник роботи _____
(підпис)

Формат	Зона	Поз.	Позначення	Назва	Кіл.	Примітка
				Документація		
			БКС 26. 14 000. 00 КРБ	Випускна робота		
А4			БКС 26. 14 000. 00 КРБ ПЗ	Пояснювальна записка	1	

Зм.	Арк.	№ докум.	Підпис	Дата	БКС 26. 14 000. 00 КРБ		
Розробив		Протасенко О.І.			Аналіз та практична реалізація методів захисту каналів зв'язку пристроїв клавіатурного введення	Літ.	Аркуш
Перевірив		Кривченко Ю.В.				Н Д П	4 / 61
Н. Контр.		Петрашова В.І.				ВСП "ОТФК ОНТУ" гр.2БКС-26	
Затверд.		Скорнякова О.В.					

ЗМІСТ

ВСТУП

Шпигунське програмне забезпечення використовує особисті дані користувача, отримані без його відома і згоди. З проблемою шпигунського ПЗ (spyware) стикаються багато компаній. У звіті Enterprise Security Survey 2021 (IDC, липень 2021 р.) вказано, що ця підмножина стоїть на 2-му місці в списку погроз мережевої безпеки – після вірусів, черв'яків і троянських програм.

Згідно опиту, проведеному дослідницьким центром TechTarget у червні 2021 р. серед 325 співробітників американських компаній, 43% респондентів вважають шпигунське ПЗ однією з трьох найбільш актуальних проблем IT-безпеки, а 62% вважають, що в майбутньому цей вид погроз стане найнебезпечнішим [1]. Тому настільки актуальним стає захист від даного виду погроз, що надається незалежними постачальниками анти-шпигунського ПЗ, і захист, інтегрований у рішення виробників антивірусних продуктів.

Як видно з результатів дослідження іншої компанії, WatchGuard Technologies, 67% з 596 опитаних IT-менеджерів і адміністраторів визнають, що їх компанії менше захищені від загрози з боку шпигунського ПЗ, ніж від вірусів, троянів і фішингових атак. На думку аналітиків, дане дослідження демонструє, що, хоча керівники IT-служб серйозно стурбовані загрозою шпигунського ПЗ, більшість їх підлеглих не мають чіткого уявлення про небезпеку.

У квітні 2021 р. компанія Trend Micro досліджувала в трьох країнах (США, Німеччині і Японії) вплив, який програми-шпигуни надають на діяльність компаній. Його учасниками стали біля 1000 співробітників організацій різного масштабу – від транснаціональних корпорацій до підприємств малого бізнесу. «Призером» виявилися США: 47% респондентів заявили, що вони стикалися з програмами-шпигунами на практиці. У Германії таких знайшлося всього 28%, а в Японії – 16%. Але у всіх цих країнах найбільший збиток від атак такого роду терпіть представники малого і середнього бізнесу. Що стосується порівняння «шпигунів» з іншими погрозами, більшість вважає їх за небезпечніші, ніж спам. За даними вже згаданого звіту IDC, сьогодні анти-шпигунським ПЗ користуються близько 90% компаній. Показник виглядає досить значним, проте

за результатами опиту Computerworld, проведеного серед співробітників IT-департаментів (Computerworld Spyware Survey, 2021), лише 54% компаній реально використовує антишпигунське ПЗ, яке задовольняє вимогам крупних мережевих інфраструктур до централізованого управління. Високо оцінили здатність використовуваних рішень виявляти і видаляти шпигунське ПЗ лише 38% опитаних, а функціями запобігання його завантаженню залишилися задоволеними лише 22% респондентів.

Гранди IT-індустрії зібрали достатню статистику по інцидентах з шпигунським ПЗ. Так, згідно Microsoft, більше половини всіх звітів про помилки, створювані ОС Windows, мають відношення саме до них. За даними Dell і крупних сервіс-провайдерів, подібні типи погроз викликають близько 15% звернень в їх служби підтримки. Але найчастіше з «шпигунами» стикаються антивірусні компанії. Дослідження Gartner («Antivirus Vendors Strike Back with Anti-spyware Product») показують, що до 50% всіх дзвінків в служби підтримки антивірусних компаній відбувається через шпигунське ПЗ. Аналітики IDC стверджують, що більше двох третин всіх офісних комп'ютерів інфіковано програмами-шпигунами.

Більш того, навіть захищений бездротовий зв'язок пристроїв введення надійними та перевіреними алгоритмами шифрування даних є вразливим до атак по стороннім каналам: завдяки останнім досягненням в математиці є можливим проаналізувати зв'язок та частково або повністю відновити зміст передаваної інформації від пристроїв введення.

Випускна бакалаврська робота присвячена аналізу та практичній реалізації методів захисту каналів зв'язку пристроїв клавіатурного введення. Метою даної роботи є дослідження існуючих векторів атак за сторонніми каналами, дослідження та реалізація методів протидії атакам, а також створення відповідних моделей.

1 ТЕХНОЛОГІЧНИЙ РОЗДІЛ

1.1 Огляд методів стеження за клавіатурним введенням

Вживання моніторингових програм дає досить широкі можливості фахівцеві, відповідальному за інформаційну безпеку підприємства. Він може наступне:

- визначати (локалізувати) всі випадки спроб несанкціонованого доступу до конфіденційної інформації з точною вказівкою часу і мережевого робочого місця, з якого така спроба здійснювалася;
- визначати факти несанкціонованої установки програмного забезпечення;
- контролювати можливість використання персональних комп'ютерів в неробочий час і визначити мету такого використання;
- визначати всі випадки несанкціонованого використання модемів в локальній мережі шляхом аналізу фактів запуску несанкціоновано встановлених спеціалізованих додатків;
- визначати всі випадки набору на клавіатурі критичних слів і словосполук, підготовки яких-небудь критичних документів, передача яких третім особам приведе до матеріального збитку;
- визначати факти нецільового використання персональних комп'ютерів;
- отримувати достовірну інформацію, на підставі якої розроблятиметься політика інформаційної безпеки підприємства;
- контролювати доступ до серверів і персональних комп'ютерів;
- контролювати контакти власних дітей при серфінгу в мережі Інтернет;
- проводити інформаційний аудит;
- досліджувати і розслідувати комп'ютерні інциденти;
- проводити наукові дослідження, пов'язані з визначенням точності, оперативності і адекватності реагування персоналу на зовнішні дії;
- визначати завантаження комп'ютерних робочих місць підприємства;
- відновлювати критичну інформацію після збоїв комп'ютерних систем і так далі.

Одна з найбільш небезпечних функцій всіх програм-шпигунів і апаратних пристроїв-кейлогерів – реєстрація натиснень клавіш, зроблених користувачем, з метою контролю комп'ютерної активності. Коли користувач набирає на клавіатурі пароль і дані своєї кредитної картки, можливо, у цей момент записується кожне натиснення клавіші. Окрім цього, сучасні програми-шпигуни дозволяють захоплювати текст з вікон додатків і робити знімки (скріншоти) екрану і окремих вікон. Іншими словами, програма-шпигун може перехопити текст документа, навіть якщо користувач його не набирав з клавіатури, а просто відкрив і проглянув файл.

1.1.1 Шпигунські програми та клавіатурні шпигуни

Spyware – це термін, що визначає додатки, які записують інформацію про поведінку користувача в мережі Інтернет і повідомляють про це своїх творців. Результатом їх дії може стати як спливаюча реклама, так і серйозніші порушення в безпеці системи, включаючи крадіжку інформації, запис натиснутих клавіш, зміну параметрів з'єднання з глобальною мережею, а також установку «чорного ходу».

Spyware-додатки зазвичай потрапляють в систему за допомогою умовно безкоштовного ПЗ, заснованого на показі банерів і реклами. Інші джерела включають програми для обміну повідомленнями, різні додатки Peer-to-Peer, популярні download-менеджери, online-ігри, множинні порно-сайти і багато іншого. Слід зазначити, що в основному spyware-додатки направлені проти браузера Microsoft Internet Explorer. Користувачі сучасних альтернативних web-браузерів, типу Mozilla Firefox або Apple Safari, менш схильні до дії spyware.

Останні методи впровадження, використовувані spyware-додатками, не вимагають жодної взаємодії з користувачем. Відомі як "Drive-by downloads" (викачувані нальоту), spyware-додатки доставляються на комп'ютер користувача без його відома, або при відвідуванні певної web-сторінки, або при відкритті зараженого файлів, або при натисненні на спливаюче віконце, що містить активний елемент типу ActiveX, Java Applet і тому подібне. Spyware-модулі

можуть також міститися в графічних файлах, а іноді навіть в драйверах для нового устаткування.

Програми-шпигуни – це ПЗ, завантажуване в комп'ютер разом з яким-небудь корисним (і не дуже) програмним продуктом. Користувач викачує його з Інтернету або встановлює з CD або інших носіїв. Багато що з подібних програм потрапляє на комп'ютери фактично легально, коли користувач сам дає добро на їх установку, «підписуючи» не дивлячись умови ліцензійної угоди встановлюваного продукту.

Подібне ПЗ може надавати різний вплив на роботу комп'ютера: задіювати обчислювальні ресурси мережі, перехоплювати оброблювану інформацію, відправляти повідомлення. У «полегшеному» варіанті заняття шпигунством предметом інтересу можуть стати адреси електронної пошти, часто відвідуваних користувачем веб-сайтів, а також персональні дані, витік яких робить його мішенню для нав'язливої реклами і спаму.

Якщо на ПК «оселився» серйозніший «шпигун», це загрожує перехопленням вводу з клавіатури і образів екрану, отриманням несанкціонованого доступу до паролів, PIN-кодів і іншої конфіденційної інформації. Такий «шпигун» може здійснювати моніторинг будь-якої діяльності користувача на зараженій машині – і із заданою періодичністю посилати зібрану інформацію своїм творцям.

Рекламне ПЗ (adware) виконує іншу функцію – воно нав'язує користувачеві проглядання реклами протягом всього часу запуску програми. Такий «додаток» часто включається до складу різного безкоштовного ПЗ або непомітно завантажується в комп'ютер (аналогічно cookies) при відвідуванні абсолютно нешкідливих на перший погляд сторінок сайтів.

Програми додзвону (dialers) служать для підключення зараженої системи до платних ресурсів якої-небудь служби. Результатом їх роботи стає рахунок за користування сервісом, про який користувач навіть не підозрює.

Клавіатурні шпигуни входять в категорію шкідливих програм, що представляє чималу загрозу для безпеки користувача. Як і програми, що

відносяться до категорії RootKit, клавіатурні шпигуни не є вірусами, оскільки не володіють здібністю до розмноження.

Клавіатурні шпигуни – це програми для прихованого запису інформації про натискувані користувачем клавіші. В терміну «клавіатурний шпигун» існує ряд синонімів: Keyboard Logger, KeyLogger, кейпотер; рідше зустрічаються терміни «снупер», «snoop», «snooper» (від англ. snoop – буквально «людина, що суне ніс у чужі справи»).

Як правило, сучасні клавіатурні шпигуни не просто записують коди натиснутих клавіш, вони як би прив'язують клавіатурний ввід до поточного вікна і елементу вводу. Крім того, багато клавіатурних шпигунів відстежують список запущених додатків, умівють робити знімки екрану за заданим розкладом або подією, шпигувати за вмістом буфера обміну і вирішувати ряд задач, націлених на приховане стеження за користувачем.

Записувана інформація зберігається на диску, і більшість сучасних клавіатурних шпигунів можуть формувати різні звіти, передавати їх по електронній пошті або по протоколах FTP і HTTP. Крім того, ряд сучасних клавіатурних шпигунів користується технологіями RootKit для маскування слідів своєї присутності в операційній системі.

Для системи клавіатурний шпигун, як правило, нестрашний, а ось для користувача він надзвичайно небезпечний, оскільки з його допомогою можна перехопити паролі і конфіденційну інформацію, що вводиться користувачем.

На жаль, останнім часом відомі сотні різноманітних кейпотерів, причому багато з них не розпізнаються антивірусами. Перед описом основних принципів роботи клавіатурного шпигуна необхідно розглянути модель апаратного вводу системи Windows. Достатньо докладний опис цієї моделі можна знайти в книзі «Windows для професіоналів» Д.Ріхтера.

При виникненні якихось подій вводу (наприклад, при натисненні клавіші або переміщенні миші) події обробляються відповідним драйвером і поміщаються в системну чергу апаратного вводу (рис. 1.1).

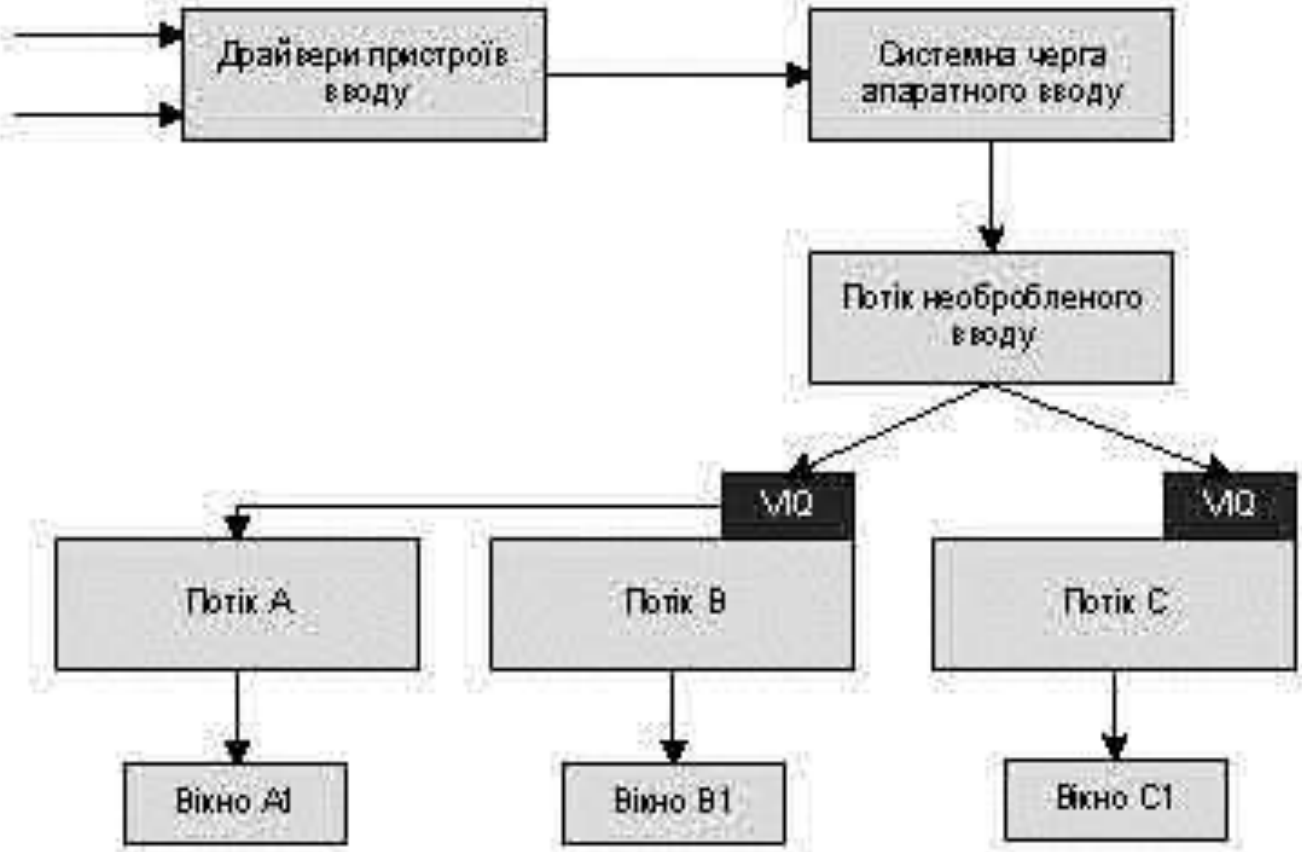


Рисунок 1.1. Модель апаратного вводу системи Windows

У системі є особливий потік необробленого вводу, званий RIT (Raw Input Thread), який витягує події з системної черги і перетворює їх в повідомлення. Отримані повідомлення поміщаються в кінець черги віртуального вводу одного з потоків (Virtualized Input Queue, VIQ віртуальна черга потоку). При цьому RIT сам з'ясовує, в чергу якого конкретного потоку необхідно помістити подію. Для подій миші потік визначається шляхом пошуку вікна, над яким розташований курсор миші.

Клавіатурні події зазвичай відповідають лише одному, так званому активному потоку (тобто потоку, якому належить вікно, з яким працює користувач). Але насправді це не завжди так – зокрема, на малюнку показаний потік A, що не має черги віртуального вводу. В даному випадку виходить, що потоки A і B спільно використовують одну чергу віртуального вводу. Це досягається шляхом виклику функції Windows API AttachThreadInput, яка дозволяє одному потоку підключитися до черги віртуального вводу іншого потоку. Слід зазначити, що потік необробленого вводу відповідає за обробку

спеціальних поєднань клавіш, зокрема Alt+Tab і Ctrl+Alt+Del.

1.1.2 Стеження за клавіатурним введенням за допомогою пасток

Дана методика є класичною для клавіатурних шпигунів, а суть її полягає у використанні механізму пасток (hook) операційної системи. Пастки дозволяють спостерігати за повідомленнями, які обробляються вікнами інших програм. Установка і видалення пасток проводяться за допомогою добре документованих функцій бібліотеки user32.dll (функція SetWindowsHookEx дозволяє встановити пастку, UnhookWindowsHookEx – зняти її). При установці пастки вказується тип повідомлень, для яких повинен викликатися обробник пастки. Зокрема, існує два спеціальні типи пасток: WH_KEYBOARD і WH_MOUSE – для реєстрації подій клавіатури і миші відповідно. Пастка може бути встановлена для заданого потоку і для всіх потоків системи, причому останнє дуже зручно для побудови клавіатурного шпигуна.

Код обробника подій пастки має бути розташований в DLL. Ця вимога пов'язана з тим, що DLL з обробником пастки проектується системою в адресний простір всіх GUI – процесів. Цікавою особливістю є те, що проектування DLL відбувається не у момент установки пастки, а при отриманні GUI – процесом першого повідомлення, що задовольняє параметрам пастки.

На компакт-диску, що додається до журналу, є демонстраційний клавіатурний шпигун, побудований на основі пастки. Він реєструє клавіатурний ввід у всіх GUI – додатках і дублює введений текст на своєму вікні. Даний приклад можна використовувати для тестування програм, протидіючих клавіатурним шпигунам.

Методика пасток дуже проста і ефективна, але у неї є ряд недоліків. Одним з них можна вважати те, що DLL з пасткою проектується в адресний простір всіх GUI-процесів, що може застосовуватися для виявлення клавіатурного шпигуна. Крім того, реєстрація подій клавіатури можлива лише для GUI-додатків – це легко перевірити за допомогою демонстраційної програми [2].

1.1.3 Стеження за клавіатурним введенням за допомогою опитування та перехоплення функцій WinAPI

Дана методика заснована на періодичному опитуванні стану клавіатури. Для опиту стану клавіш в системі передбачена спеціальна функція GetKeyboardState, що повертає масив з 255 байт, в якому кожен байт містить перебування певної клавіші на клавіатурі. Даний метод вже не вимагає впровадження DLL в GUI-процеси, і в результаті шпигун менш помітний.

Проте зміна статусу клавіш відбувається у момент зчитування потоком клавіатурних повідомлень з його черги, тому подібна методика працює лише для стеження за GUI-додатками. Цього недоліку позбавлена функція GetAsyncKeyState, що повертає стан клавіші на момент виклику функції.

Недоліком клавіатурних шпигунів такого типу є необхідність періодичного опиту стану клавіатури з досить високою швидкістю – не менше 10-20 опитів в секунду.

Методика стеження за клавіатурним введенням за допомогою перехоплення функцій WinAPI не набула широкого поширення, але проте може з успіхом застосовуватися для побудови клавіатурних шпигунів. Методики перехоплення функцій API детально розглядувалися в статті, присвяченій RootKit [3].

Різниця між RootKit і клавіатурним шпигуном в даному випадку невелика – шпигун перехоплює функції з метою моніторингу, а RootKit – з метою модифікації принципів роботи і результатів виклику.

Простим способом може бути перехоплення функцій GetMessage, PeekMessage і TranslateMessage бібліотеки User32, що дозволить вести моніторинг всіх повідомлень, що отримуються GUI-додатками.

1.1.4 Апаратні клавіатурні шпигуни

Це мініатюрні вбудовані пристрої, розташовані між клавіатурою і комп'ютером. Через маленькі розміри вони часто залишаються непоміченими тривалий час, проте вони вимагають фізичного доступу до устаткування. Ці пристрої можуть записувати сотні символів, введених з клавіатури, включаючи

поштові і банківські реквізити.

Апаратні кейлогери (keystroke recording device, hardware keylogger і ін.) є мініатюрними пристосуваннями, які можуть бути прикріплені між клавіатурою і комп'ютером або вбудовані в саму клавіатуру (рис.1.2). Вони реєструють всі натиснення клавіш. Процес цей абсолютно непомітний для користувача. Апаратні кейлогери не вимагають установки якої-небудь програми на комп'ютері об'єкту щоб успішно перехоплювати всі натиснення клавіш. Такий пристрій може бути таємно прикріплений до ПК ким завгодно – колегою, прибиральницею, відвідувачем і так далі. Коли апаратний кейлогер прикріплюється, абсолютно не має значення, в якому стані знаходиться комп'ютер – включеному або вимкненому.

Рисунок 1.2. Пристрої для фіксації символів, набраних з клавіатури

Атакуючий може зняти пристрій в будь-який зручний момент, а його вміст (записані натиснення клавіш) викачати, коли йому це буде зручно. Об'єми внутрішньої незалежної пам'яті даних пристроїв дозволяють записувати до 10 мільйонів натиснень клавіш. Прикріпити даний пристрій до комп'ютера користувача дуже легко. Зовнішній вигляд цих пристроїв настільки багатообразний, що навіть фахівець не в змозі іноді визначити їх наявність при проведенні інформаційного аудиту.

Особливо відомі на ринку апаратні кейлогеры KeyKatcher, KeyGhost, MicroGuard, Hardware KeyLogger, виробниками яких є компанії Alien Concepts, Inc., Amecisco, KeyGhost, Ltd., MicroSpy, Ltd.

Апаратні кейлогеры підрозділяються на зовнішні і внутрішні, їх особливості описані нижче.

Зовнішні апаратні кейлогери підключаються між звичайною клавіатурою ПК і комп'ютером і реєструють кожне натиснення клавіш. Їм не потрібні ні батареї, ні програми, і вони можуть працювати на будь-якому ПК. Можна підключити їх до одного комп'ютера, щоб записати інформацію, а потім до іншого, щоб відтворити її. Сучасні апаратні кейлогери є пристроями, які виглядають як устаткування для ПК і розміщуються в різних зонах (рис.1.3).

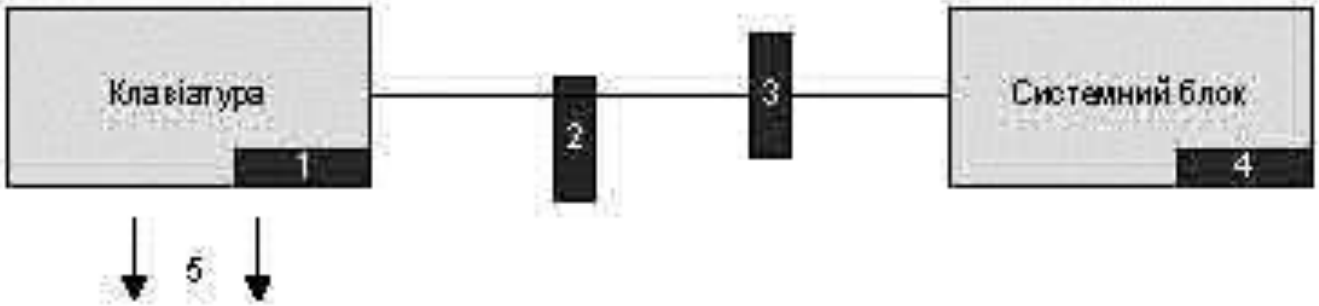


Рисунок 1.3. Можливі зони апаратного перехоплення інформації

Найскладніше виявити (і знешкодити) внутрішній апаратний кейлогер, в якого апаратний модуль перехоплення натиснень клавіш вбудований у корпус клавіатури. Він є вбудованим невеликим пристроєм, встановленим в розрив шнура клавіатури і покритим ізоляційним матеріалом.

Для пошуку клавіатурних шпигунів на домашньому комп'ютері цілком достатньо упевнитися у відсутності програм-кейлогерів. Але в корпоративному середовищі, зокрема на комп'ютерах, вживаних для виконання банківських операцій, для проведення електронних торгів або для вирішення задач, пов'язаних з обробкою секретних документів, є небезпека вживання апаратних засобів, призначених для реєстрації інформації, що вводиться з клавіатури. Розглянемо основні канали просочування інформації з точки зору вживання апаратних засобів [1].

У будь-якій клавіатурі зазвичай буває багато порожнин, розмір яких достатній для розміщення невеликої плати. Живлення пристрою і зняття інформації може проводитися шляхом безпосереднього підключення до друкованої плати контролера клавіатури. Апаратна закладка може бути встановлена уручну або промисловим способом (рис.1.4). Наприклад, на сайті

http://www.keyghost.com можна побачити рекламу клавіатур з вбудованим апаратним кейлогером.

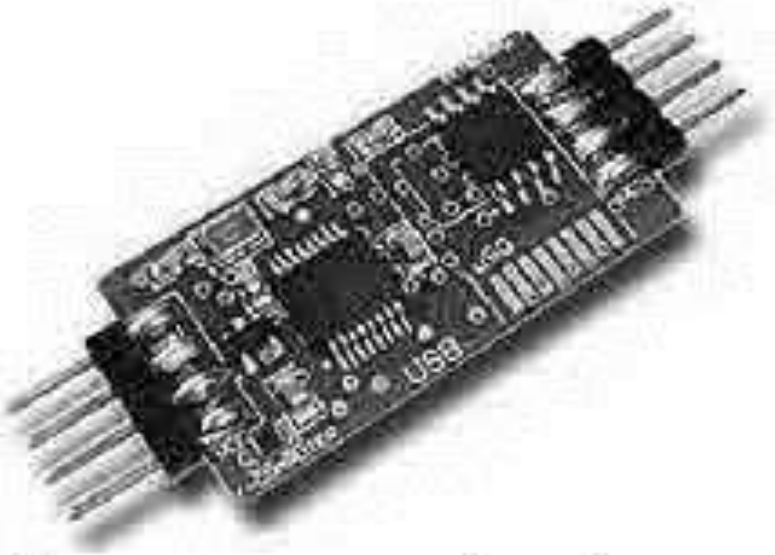


Рисунок 1.4. Апаратна закладка усередині клавіатури у вигляді плати

Методика протидії: розтин клавіатури і її перевірка на предмет наявності сторонніх електронних вузлів з подальшим пломбуванням корпусу клавіатури за допомогою пломбіру або стікера.

Методика зчитування даних з кабелю клавіатури безконтактним методом передбачає зчитування інформації за допомогою безконтактного датчика. Для установки такого пристрою не вимагається ні розтину клавіатури, ні включення яких-небудь пристроїв в розрив кабелю. Безконтактний апаратний кейлогер повинен мати автономне живлення, а його пристрій значно складніший, ніж в разі безпосереднього підключення. На вигляд подібний пристрій може виглядати як знімний фільтр перешкод, що надягається на кабель. Методика протидії: безконтактне зчитування найефективніше в разі розміщення датчика в безпосередній близькості від кабелю клавіатури (а ще краще – довкола цього кабелю), а тому при перевірці робочих місць необхідно переконатися у відсутності сторонніх предметів невідомого призначення поблизу кабелю клавіатури або безпосередньо на ньому.

Клавіатурні шпигуни з включенням пристрою в розрив кабелю є найпоширенішими і їх легко як встановити, так і виявити. Апаратний кейлогер виконується у вигляді невеликого пристрою, який включається в PS/2- або USB-

роз'їм комп'ютера, а клавіатура включається в роз'їм на корпусі кейлогера. Для виконання подібної операції не потрібно жодної кваліфікації, причому підключення кейлогера до USB-клавіатури може проводитися без виключення комп'ютера. Відомий ряд серійно випускаємих пристроїв, наприклад KEYKatcher Hardware Keyloggers (http://www.keykatcher.com), який випускається в двох видах – для PS/2- і USB-клавіатур. Інший приклад – KeyGhost (рис.1.5).

Рисунок 1.5. Підключення апаратного кейлогера KeyGhost

Апаратний кейлогер може виглядати як фільтр перешкод або перехідник. Пристрій складається з вхідних ланцюгів, призначених для фільтрації перешкод і захисту пристрою від перенапруження, мікроконтролера з малим споживанням електроенергії і Flash-пам'яті, призначеної для зберігання збираної інформації (рис.1.6). Об'єм Flash-пам'яті варіюється від 32 Кбайт до десятків мегабайт; типовий об'єм – від 128 Кбайт до 2 Мбайт.

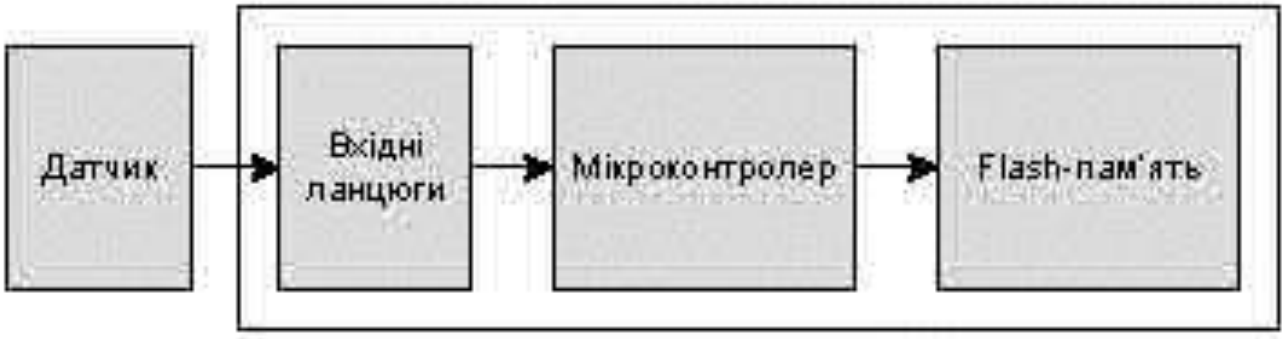


Рисунок 1.6. Функціональна схема апаратного кейлогера

Методика протидії: періодичний огляд робочого місця на предмет наявності сторонніх пристроїв, включених в розрив кабелю клавіатури. Штекер клавіатури досить просто захистити стікером, що порушується при витяганні штекера з роз'єму.

Рисунок 1.7. Зовнішній вигляд апаратного кейлогера KeyDemon DVI 2Gb

На сьогоднішній день існують також кейлогери у вигляді DVI-HDMI перехідника. Їх особливість є те, що вони передають не набір набраних на клавіатурі символів, а зображення (рис.1.7).

Подібних апаратів в даний момент на ринку одиниці. Використовувати пристрій украй просто. Необхідно підключити його до відповідного (DVI, VGA, HDMI) порту монітора або комп'ютера і включити прилад. Після чого кейлогер почне циклічно, з періодом в декілька секунд, фіксувати знімки з екрану і записувати їх на flash-пам'ять, якої в даному пристрої передбачено 2Гб, це вистачить для величезної кількості знімків. Щоб зчитати отриману інформацію, потрібно просто перемкнути кейлогер в режим «Flash drive» і працювати з ним як із звичайною флешкою. Зберігаємі зображення кодуються по алгоритму JPEG. Якості стискування даного формату цілком достатньо для роботи із скріншотами. Часто такий пристрій беруть для того, щоб простежити за умовами роботи співробітників і підвищити ефективність використання трудових ресурсів; визнати, чим займається і захоплюється дитина в інтернет; захистити дитину від можливих неприємностей в інтернет.

Апаратна закладка усередині системного блоку не відрізняється за

принципом дії від попередніх пристроїв, але розміщується усередині системного блоку. Встановити його може лише фахівець, причому для цього буде потрібен розтин корпусу.

Методика протидії: пломбування системного корпусу за допомогою стікерів. Перед пломбуванням необхідно досліджувати вміст системного блоку і переконатися у відсутності сторонніх пристроїв (типове місце підключення – материнська плата; підключення проводиться паралельно роз'єму клавіатури).

Уловити електромагнітне випромінювання клавіатури на відстані вельми складно (хоча теоретично і можливо), але уловити акустичні шуми – на значно простіше. Навіть при розмові по телефону іноді можна виразно почути, як співбесідник вводить інформацію на клавіатурі. Дослідження фахівців в області безпеки показують, що кожна клавіша при натисненні відтворює специфічний звук, що дозволяє ідентифікувати натиснуті клавіші. Найбільш відома робота в цьому напрямку, проведена вченими Каліфорнійського університету в Берклі (докладніше – http://zdnet.ru/?ID=498415), які прийшли до висновку, що по звичайному звукозапису можна розпізнавати від 60 до 96% введених символів. Без вживання спеціалізованих програм для аналізу можна достатньо просто встановити кількість символів, набрану в паролі і наявність символів, що повторюються.

Методика протидії: основний спосіб захисту від витоку інформації шляхом аналізу акустичних сигналів – постійний і планомірний інструктаж персоналу.

1.1.5 Клавіатурний шпигун на базі драйвера

Даний метод ще ефективніший, ніж описані вище. Можливі як мінімум два варіанти реалізації цього методу – написання і установка в систему свого драйвера клавіатури замість штатного або установка драйвера-фільтру. Вживання драйвера-фільтру є найбільш коректною методикою.

Робота кейлогерів на основі драйвер-фільтру заснована на установці драйвера, що підключається до драйвера клавіатури як фільтр. Приклад реалізації фільтру клавіатури наведений в DDK, і даний шпигун є одним з

найпростіших в плані як реалізації, так і виявлення. Клавіатурний шпигун цього типа може бути побудований по одній з двох схем:

- драйвер-фільтр у поєднанні із управляючим додатком, який виконує установку, завантаження і налаштування драйвера. Драйвер передає інформацію про натиснуті клавіші управляючому додатку, який проводить обробку і протоколювання отриманих даних;
- повністю автономний драйвер самостійно веде запис подій. В даному випадку потрібно лише провести первинну установку драйвера в систему, після чого встановлений драйвер додаток може самознищуватися. Можливе рішення і без додатка – інсталятора – в цьому випадку драйвер встановлюється за допомогою INF-файлу.

У момент завантаження драйвер повинен підключитися до стека клавіатурного драйвера за допомогою функцій IoCreateDevice і IoAttachDevice. У більшості відомих реалізацій фільтр підключається до стека пристрою «\\Device\\KeyboardClass0», що є драйвером класу і що реалізує загальну функціональність для клавіатур різних типів (рис. 1.8).

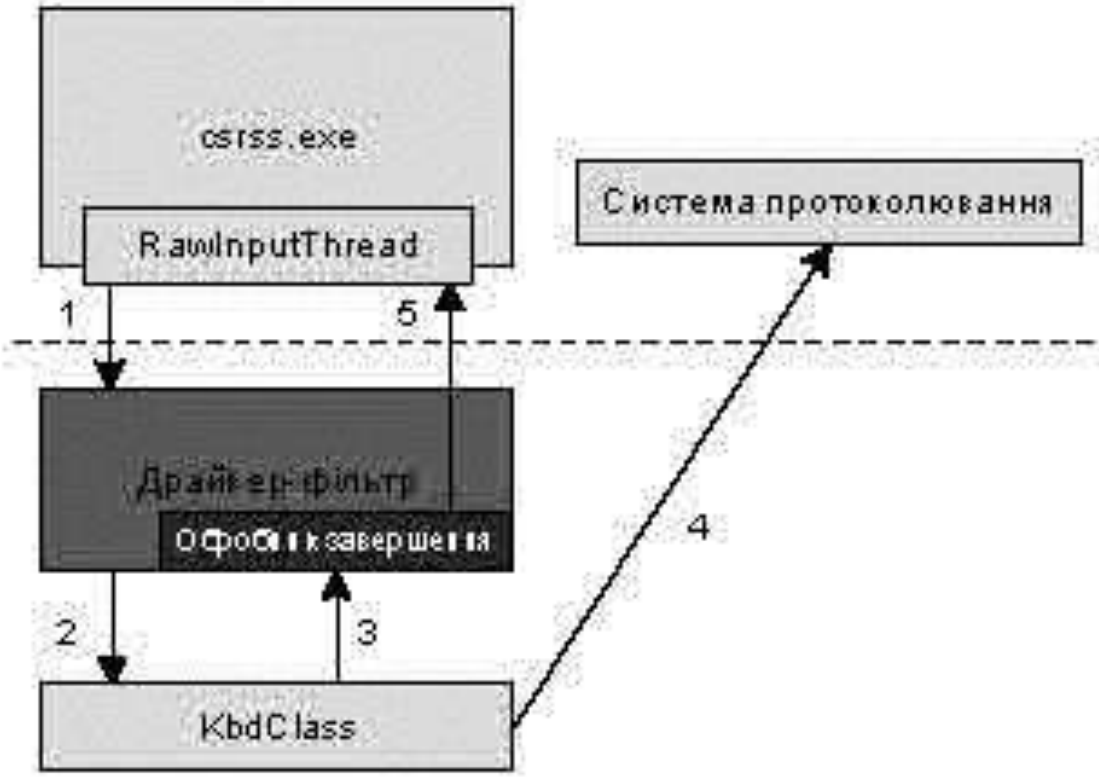


Рисунок 1.8. Підключення драйвера-фільтра до стека клавіатурного драйверу

Для клавіатурного шпигуна представляють інтерес лише переривання типу IRP_MJ_READ, оскільки на основі їх аналізу можна отримати коди клавіш. Джерелом цих IRP-запитів є процес csrss.exe, а точніше – належний цьому процесу потік необробленого вводу RawInputThread. Відправлене цим потоком переривання спочатку потрапляє до драйвера-фільтру шпигуна (крок 1), який встановлює свій обробник завершення за допомогою функції IoSetCompletionRoutine і передає IRP драйверу Kbdclass (крок 2). Той, у свою чергу, позначає IRP як чекаючий завершення і ставить в чергу. При виникненні клавіатурних подій Kbdclass витягує з черги чекаючий завершення IRP, вносить до його буфера інформацію про натиснуті клавіші і завершує IRP. А оскільки в IRP встановлена адреса обробника завершення, то буде проведений виклик цього обробника (крок 3). Обробник може проаналізувати вміст буфера і передати інформацію, що міститься там, системі протоколювання (крок 4). Далі IRP повертається в потік RawInputThread, а потім весь процес повторюється. Природно, що наявність коректно написаного драйвера-фільтру ніяк не позначається на роботі додатків і забезпечує глобальне перехоплення клавіатурного вводу.

Виявлення описаного шпигуна не представляє особливої складності – для його пошуку досить вивчити стек клавіатурного драйвера на предмет наявності невідомих драйверів-фільтрів.

Метод на основі підміни драйвера клавіатури заснований на підміні драйвера Kbdclass або одного з низькорівневих драйверів клавіатури драйвером власної розробки [2]. Складність реалізації вказаного методу полягає в тому, що заздалегідь невідомий тип вживаної користувачем клавіатури, а тому підміну драйвера порівняно просто виявити. Внаслідок цього подібний метод практично не зустрічається.

Принцип дії клавіатурного шпигуну на базі руткіт-технології в UserMode заснований на перехопленні ряду функцій USER32.DLL для моніторингу їх викликів. Дані шкідливі програми поки не набули особливого поширення, але це лише питання часу. Небезпека вживання руткіт-технології в клавіатурному

шпигунові пояснюється тим, що, по-перше, багато антихейпогерів не розраховані на пошук шпигунів такого типа і не здатні їм протидіяти, а по-друге, антируткіти часто не перевіряють перехоплення функцій бібліотеки user32.dll.

Принцип роботи шпигуна досить простий: за допомогою будь-якої з відомих рутків-технологій проводиться перехоплення однієї або декількох функцій, що дозволяють отримати контроль над інформацією, що вводиться з клавіатури. Найпростішим є перехоплення функцій GetMessage і PeekMessage (рис. 1.9).

Рисунок 1.9. Перехоплення функцій GetMessage і PeekMessage.

Робота шпигуна організована таким чином. Додаток викликає функцію PeekMessage для того, щоб взнати, чи є в черзі повідомлення вказаного типа. Цей виклик перехоплюється по рутків-принципу (методика в даному випадку не має значення). Потім перехоплювач викликає реальну функцію PeekMessage з user32.dll і аналізує повертаємі результати. Якщо функція повертає true, це означає, що повідомлення було в черзі і що воно поміщене в той буфер, покажчик на який передається як перший параметр функції. В цьому випадку перехоплювач перевіряє повідомлення в буфері на предмет виявлення повідомлень типу WM_KEYDOWN (натиснення клавіші), WM_KEYUP (відпускання клавіші), WM_CHAR (посилається вікну після обробки WM_KEYDOWN за допомогою TranslateMessage). При виявленні подібного повідомлення можна взнати код натиснутої клавіші і передати його системі протоколювання і аналізу (крок 4). Далі управління повертається додатку (крок 5), який не знає про наявність перехоплювача.

Подібний клавіатурний шпигун дуже небезпечний, оскільки:

- він не виявляється стандартними методиками пошуку клавіатурних шпигунів;
- можливе впровадження перехоплювача за певними умовами, внаслідок чого він упроваджується не у всі GUI-процеси, а в строго визначені (наприклад, в процеси браузера або в додаток типа WebMoney);
- проти нього марні екранні клавіатури і інші засоби боротьби з клавіатурними шпигунами;
- окрім перехоплення функцій PeekMessage і GetMessage, можуть бути перехоплені функції копіювання інформації при роботі з буфером обміну (OpenClipboard, CloseClipboard, GetClipboardData, SetClipboardData), опиту стану клавіатури (GetKeyState, GetAsyncKeyState, GetKeyboardState) і інші функції user32.dll, що підсилює небезпеку шпигуна, а перехоплення функцій типа CreateWindow дозволяє відстежувати створення вікон.

Принцип дії клавіатурного шпигуну на базі руткіт-технології в KernelMode аналогічний UserMode, але в даному випадку проводиться перехоплення однієї або декількох функцій win32k.sys. Як і у випадку з UserMode, найбільший інтерес для перехоплювача представляє функція PeekMessage і її аналоги, оскільки це дозволяє проводити моніторинг і модифікацію абсолютно всіх отримуваних програмою повідомлень без установки пастки або фільтру.

На рис. 1.10 перехоплювач показаний умовно, оскільки для його встановлення існують різні методики, зокрема:

- перехоплення SYSCALL і INT2E;
- перехоплення функції з підміною адреси у відповідному елементі таблиці KeServiceDescriptorTableShadow. Єдиною складністю для творця подібного шпигуна є пошук цієї таблиці, яка не експортується ядром і не документована, проте відомі шляхи подолання цієї проблеми;
- модифікація машинного коду win32k.sys. Для цього теж необхідний пошук таблиці KeServiceDescriptorTableShadow. При цьому можлива

цікава ситуація: функція може бути вже перехоплена (наприклад, антикейлогером), і тоді буде проведена модифікація машинного коду перехоплювача, що ще більше ускладнить виявлення шпигуна.

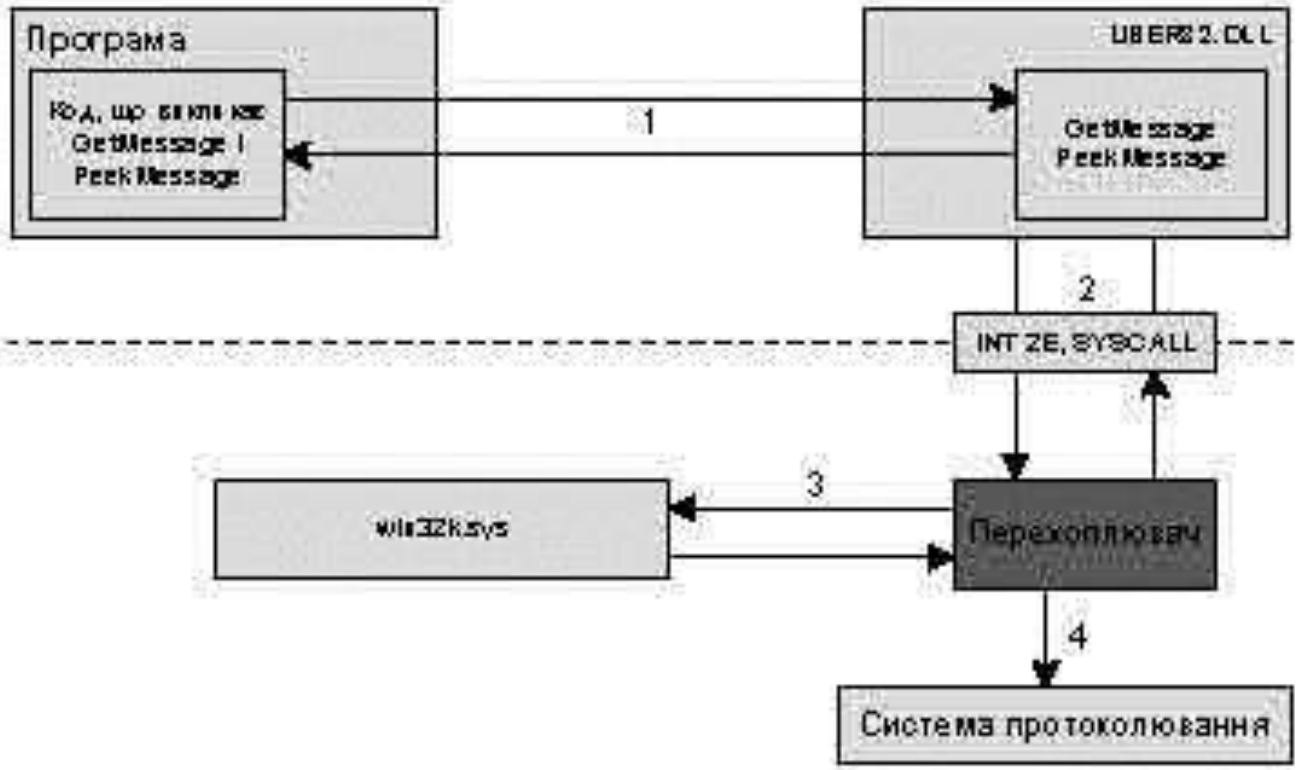


Рисунок 1.10. Перехоплювач на базі руткіт-технології в KernelMode

Алгоритм роботи шпигуна вельми простий. Додаток викликає функцію бібліотеки user32.dll (крок 1; як приклад розглянемо виклик PeekMessage). Функція PeekMessage в user32.dll за своєю суттю є перехідником, а кінець кінцем за допомогою SYSCALL в Windows XP або INT 2E в Windows NT і Windows 2000 буде проведений виклик функції ядра (крок 2). Цей виклик буде перехоплений шпигуном (місце розташування перехоплювача залежить від методики перехоплення). Перехоплювач, у свою чергу, викличе реальну функцію (крок 3) і проаналізує повернені нею результати. В разі успішного витягання з черги повідомлення потрібного шпигунові типа він проведе його аналіз і запротоколює результати (крок 4). Робота шпигуна абсолютно непомітна для всіх додатків, і вилікувати його можна лише спеціальними програмами, що проводять пошук перехоплень і модифікацій машинного коду модулів ядра.

1.1.6 Аналіз отриманих пакетів та вилучення інформації

Затримки між натисканнями зазвичай залежать від положення клавіші на клавіатурі, тому можна з певною ймовірністю відтворювати інформацію, що вводиться. Наприклад, більшість людей зазвичай набирають "s" після "a" значно швидше, ніж "g" після "s". Маючи точні моменти передачі пакетів можна дати оцінку – наскільки ці моменти відповідають тому чи іншому слову.

Наприклад, при введені на клавіатурі "привіт дерево" маємо різні затримки між натисканнями. Затримки проілюстровано на рис. 1.11, де сині лінії – затримка у 150-200мс, рожеві лінії – затримка у 180-250мс, помаранчеві лінії – затримка у 200-280мс, червоні лінії – затримка у 230-320мс, зелені лінії – затримка у 250-360мс, чорна цятка – затримка у 400-700мс.

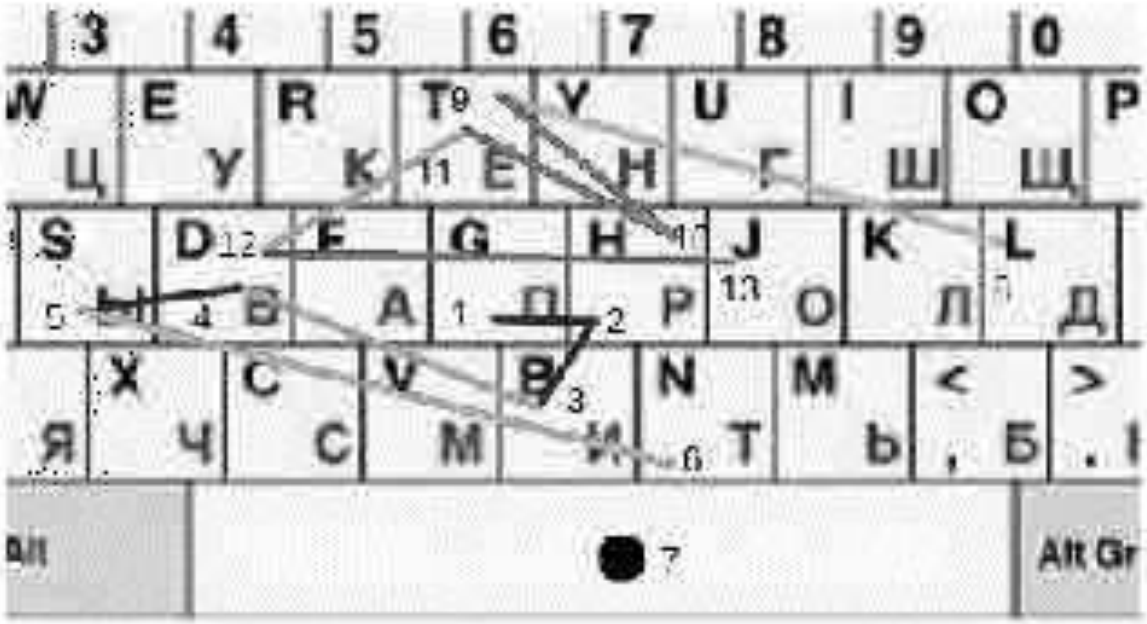


Рисунок 1.11. Ілюстрація затримок при наборі на клавіатурі

Для наступного аналізу необхідно мати перевірочний набір моментів натискання символів та отриманий набір моментів. Перевірочний набір моментів – набір моментів натискання та відпускання клавіш, про який відомо, які саме клавіші були натиснутими. Даний набір можна отримати при звичайному наборі тексту або при переписці із жертвою. Отриманий набір моментів – набір моментів натискання та відпускання клавіш, який був перехоплений при прослуховуванні трафіку жертви.

1.2 Аналіз методів захисту пристроїв клавіатурного введення

1.2.1 Методи та засоби захисту бездротового зв'язку пристроїв клавіатурного введення

Деякі виробники пристроїв клавіатурного введення передбачають програмовані донгли (приймачі Wi-Fi або Bluetooth для бездротових клавіатур та мишей) – це такі, у яких прошивку можна замінити. На жаль, велика кількість донглів мають пам'ять тільки для читання, а отже їх вразливість виправити неможливо, що, в свою чергу, залишить вразливими до зловмисників декілька мільйонів пристроїв, що регулярно використовуються людьми у всьому світі.

Компанія Logitech випустила модифіковану прошивку для бездротових пристроїв. Компанія Lenovo надає консультативну підтримку та розробила оновлення прошивки, але прошивку можна застосовувати тільки на момент виробництва, тобто, не буде завантажених виправлень. Компанія Dell заявила, що клієнти із пристроями, такими, як клавіатури та миші KM714, можуть використати оновлення Logitech, використовуючи технічну підтримку Dell. Компанія Microsoft випускає серію бездротових клавіатур, що допомагають захистити спілкування за допомогою покращеного стандарту шифрування (AES). Технологія шифрування AES зашифровує інформацію під час натискання клавіш до початку передачі її на комп'ютер або інший пристрій. Покращений стандарт шифрування (AES) розроблений Національним інститутом стандартів і технологій (National Institute of Standards and Technology – NIST) та прийнятий національним урядом США та інших країн з метою надійного захисту інформації та конфіденційних даних [5]. Клавіатури Microsoft, що використовують AES шифрування, застосовують генерування випадкових даних і унікальні ідентифікатори для кожного переданого пакету даних, щоб запобігти складнішим атакам.

1.2.2 Аналіз методів протидії програмам-шпигунам

Для виявлення і видалення моніторингових програмних продуктів, які можуть бути встановлені без відома користувача ПК, в даний час використовуються програми, які за допомогою сигнатурного аналізу

забезпечують більш менш ефективний захист лише проти відомих програм-шпигунів. Для ефективної роботи програм цього типа необхідно отримати зразок програми-шпигуна, виділити з неї сигнатуру і включити її в свою базу. При оновленні сигнатурної бази користувачі персонального комп'ютера дістають можливість боротися з даним варіантом програми-шпигуна. За таким принципом працюють багато відомих фірм – виробників антивірусного програмного забезпечення.

Але є і інша група програм-шпигунів, яка найбільш небезпечна для будь-яких автоматизованих систем, – це невідомі програми-шпигуни. Вони підрозділяються на п'ять типів.

1. Програми-шпигуни, що розробляються під егідою урядових організацій (приклад – продукт Magic Lantern, проект під назвою Cyber Knight, США).

2. Програми-шпигуни, які можуть створюватися розробниками різних операційних систем і включатися ними до складу ядра операційної системи.

3. Програми-шпигуни, які створені в обмеженій кількості (часто лише в одній або декількох копіях) для вирішення конкретного завдання, пов'язаного з викраданням критичної інформації з комп'ютера користувача (наприклад програми, вживані хакерами-професіоналами). Такі програми можуть бути трохи видозміненими відкритими початковими кодами програм-шпигунів, узятими з Інтернет і скомпільованими самим хакером, що дозволяє змінити сигнатуру програми-шпигуна.

4. Комерційні, особливо, корпоративні програмні продукти, які дуже рідко вносяться до сигнатурних баз, а якщо і вносяться, то лише по політичних мотивах (приклад – програмні продукти таких відомих фірм, як WinWhat-Where Corporation, SpectorSoft Corporation, ExploreAnywhere Software LLC, Omniquad, Ltd).

5. Програми-шпигуни, що є keylogging-модулями, що входять до складу програм-вірусів. До внесення сигнатурних даних до вірусної бази дані модулі є невідомими. Приклад – всесвітньо відомі віруси, що натворили багато бід останніми роками, мають в своєму складі модуль перехоплення натиснень

клавіатури і відправки отриманої інформації в мережу Інтернет, наприклад:

- W32.Dumaru.Y@mm – http://securityresponse.symantec.com/avcenter/venc/data/w32.dumaru.y@mm.html;
- W32.Yaha.AB@mm – http://securityresponse.symantec.com/avcenter/venc/data/w32yaha.ab@mm.html;
- W32.Bugbear.B@mm – http://securityresponse.symantec.com/avcenter/venc/data/w32.bugbear.b@mm.html;
- W32.HLLW.Fizzer@mm – http://securityresponse.symantec.com/avcenter/venc/data/w32hllw.fizzer@mm.html;
- W32.Badtrans.B@mm – http://securityresponse.symantec.com/avcenter/venc/data/W32.Badtrans.B@mm.html

Інформація про програми-шпигуни першого і третього типа, як правило (якщо не відбувається просочування інформації), ніде не публікується, і, відповідно, їх код не може бути внесений до сигнатурних баз, тому їх не можуть виявити ніякі програмні продукти, що використовують сигнатурний аналіз.

Інформація про програми-шпигуни другого типу ніде не публікується, даний код працює на рівні ядра операційної системи і, відповідно, вони не можуть виявлятися жодними додатками.

Інформація про програми-шпигуни четвертого типу вноситься до сигнатурних баз дуже рідко, оскільки це протидіє законодавству багатьох країн світу. Але навіть якщо це і відбувається, то деактивувати, а тим більше видалити їх часто неможливо без руйнування операційної системи. Ці програми не мають своїх процесів, а ховаються у вигляді потоків в системні процеси. Вони мають режими контролю цілісності і самовідновлення після збоїв, можуть працювати лише з пам'яттю комп'ютера і не працювати з жорстким диском.

Інформація про програми-шпигуни п'ятого типа вноситься до сигнатурних баз через декілька годин або днів після початку відповідної вірусної атаки. А за цей час конфіденційна інформація користувача персонального комп'ютера вже може бути викрадена і відіслана в Інтернет на заздалегідь підготовлену вірусо-розробником адресу.

Розглянемо, що може протиставити користувач персонального комп'ютера програмам-шпигунам. Вирішення даної проблеми можливе лише у використанні комплексу програмних продуктів.

Програмний продукт №1 – той, який використовує евристичні механізми захисту, створені фахівцями, що мають великий досвід боротьби з програмами-шпигунами. Його захист безперервний, при цьому він не використовує жодні сигнатурні бази.

Програмний продукт №2 – антивірусний програмний продукт, що використовує постійно обновлювані сигнатурні бази.

Програмний продукт №3 – персональний firewall, контролюючий вихід в Інтернет з персонального комп'ютера на підставі установок самого користувача.

Така послідовність вибрана неспроста. Антивірусний програмний продукт встигає відреагувати на проникнення вірусу з keylogging-модулем, коли вже здійснено перехоплення інформації, оскільки вірусна база ще не встигла поповнитися новою інформацією, а відповідно, і оновитися на комп'ютері користувача. Персональний firewall ставить багато питань, на які навіть дуже добре підготовлений користувач може відповісти некоректно, тим самим неправильно його сконфігурувавши. Наприклад, деякі комерційні моніторингові програми використовують процеси програмних продуктів, яким свідомо дозволений вихід в Інтернет (браузери, поштові клієнти). Це приводить до того, що та інформація, яка вже була вкрадена при повній бездіяльності антивірусної програми, буде передана в мережу на заздалегідь підготовлену хакером (або кимось іншим) інтернет-адресу. Лише програмний продукт першого типа працює мовчки, не ставлячи непотрібних питань користувачеві, і здійснює свою роботу безперервно у фоновому режимі [3].

Антивірусних програмних продуктів, що використовують постійно оновлювані сигнатурні бази, в світі створена велика кількість (AVP, Dr.Web, Panda Antivirus, Norton Antivirus і багато інших). Персональних міжмережевих екранів створено ще більше (Norton Internet Security, BLACKICE Defender, GuardianPro Firewall, Tiny Personal Firewall та інші). Захисні програми першого

типа представлені на сьогоднішній день всього лише одним продуктом, що не має аналогів в світі – PrivacyKeyboard™.

PrivacyKeyboard™ блокує роботу програм-шпигунів без використання сигнатурних баз. Це стало можливим завдяки тому, що були знайдені рішення і розроблені алгоритми, які дозволили відрізнити роботу програми-шпигуна від будь-якого іншого додатку, який встановлений в системі.

PrivacyKeyboard™ має в своєму складімодулі, що забезпечують:

- захист відперехоплення натиснень клавіш клавіатури;
- захист відперехоплення тексту з вікон;
- захист від зняття зображення робочого столу;
- захист від зняття зображення активних вікон.

Для власного захисту від зовнішньої руйнівної дії програм-шпигунів програма PrivacyKeyboard™ має систему контролю цілісності і інші захисні функції.

Жодні програмні продукти не в змозі визначити наявність встановлених апаратних пристроїв, які забезпечують перехоплення натиснень клавіатури користувачем персонального комп'ютера.

Сьогодні існує лише два методи протидії апаратним кейлогерам при роботі на стандартному персональному комп'ютері:

- фізичний пошук і усунення апаратного кейлогера;
- використання віртуальних клавіатур для вводу особливо важливої інформації (логіни, паролі, коди доступу, PIN-коди кредитних карт).

1.2.3 Заходи захисту клавіатурного введення

На відміну від спаму і іншого небажаного ПЗ шпигунські програми володіють функціоналом, за допомогою якого можуть бути скомпрометовані конфіденційні дані. Потенційні результати «діяльності» цих програм – істотне уповільнення роботи комп'ютера або складності при відвідуванні будь-якого веб-сайту, зниження продуктивності роботи співробітників і інформаційної безпеки компанії.

Вирішити проблему шпигунського ПЗ можна лише комплексними

заходами в масштабах всієї організації.

Зараз достатньо гостро відчувається відставання засобів виявлення і видалення шпигунського ПЗ від вимог, що пред'являються до централізованого управління даними продуктами, а швидкість розробки захисних заходів набагато нижче за швидкість створення шпигунського ПЗ. Цей пропуск зараз скорочується зусиллями як незалежних компаній, орієнтованих головним чином на малий і середній бізнес, так і великих антивірусних вендорів, доповнюючих лінійки продуктів рішеннями для боротьби з шпигунським ПЗ.

Жодне з антишпигунських рішень, що існують на ринку, не забезпечує 100%-го захисту. Навіть у абсолютно замкнутому програмному середовищі можливе зараження, якщо в дозволеному встановленому ПЗ шпигунський модуль присутній спочатку.

Головна міра захисту – «виховання» користувачів. Це необхідна складова процесу забезпечення безпеки корпоративної мережі. До тих пір, поки співробітники не усвідомлять, що не можна «не дивлячись» завантажувати на свої комп'ютери якесь ПЗ, на перший погляд корисне або забавне, максимального ефекту від засобів захисту від погроз, подібних шпигунському ПЗ, чекати не варто.

Шпигунське ПЗ стоїть на 2-му місці в списку погроз мережевої безпеки – після вірусів, черв'яків і троянських програм.

Користувачі повинні добре уявляти собі ризики і фінансовий збиток, які несе з собою шпигунське ПЗ, а також дотримувати превентивні заходи.

Багато шпигунських програм потрапляє на комп'ютери легально: користувач сам дає добро на їх установку, не дивлячись підписуючи умови ліцензійної угоди встановлюваного продукту.

П'ять основних наслідків атак шпигунського ПЗ (версія Trend Micro):

- задіювання обчислювальних ресурсів мережі;
- зниження продуктивності роботи користувачів;
- зниження пропускної спроможності мережі;
- завантаження шкідливого ПЗ;

- вторгнення в приватне життя.

Спершу визначимо конкретні шляхи проникнення шпигунського ПЗ. Деякі з них – CD і інші носії, оновлення ПЗ. Але найчастіше це використання безкоштовного ПЗ. Іншим джерелом можуть бути функції інтернет-браузера, що забезпечують закачування оновлень без участі користувача: вони виступають як інструмент для завантаження небажаного ПЗ. Для контролю за подібним завантаженням, впровадженим, наприклад, за допомогою ActiveX, можна застосовувати функцію ActiveX Control, що реалізує запит на підтвердження завантаження. Робота в Інтернет за наявності незакритих уразливостей ОС або інтернет-браузера теж надзвичайно небезпечна. Слід остерігатися і програм, що маскуються під автоматичні оновлення ОС.

Особливу увагу слід звертати на такі точки проникнення в мережу шкідливого ПЗ, як засоби миттєвої передачі повідомлень (інтернет-пейджинг) і файлообмінні мережі (Kazaa, eDonkey і ін.). За даними Osterman Research, більше 90% компаній, в яких дані засоби дозволені до використання, ніяк не контролюють їх вживання і не мають відповідних засобів захисту.

Розглянемо захисні заходи, які необхідно зробити в масштабах організації. Перш за все, і це головне, в компанії повинна існувати політика безпеки, регулююча правила доступу користувачів в Інтернет [4]. Слід регулярно проводити навчання персоналу згаданим вище «правилам гігієни» при роботі в Інтернет, щоб кожен знав «оперативні дані» про існуючі інтернет-погрози. Подібне навчання рекомендується проводити на конкретних, зрозумілих користувачеві прикладах, а не у вигляді загальної теорії. Якщо користувач не усвідомлюватиме, чим може загрожувати викачування «цікавої» програми з мережі, то регламентуючі заходи не принесуть очікуваних результатів.

Другий принцип – централізоване впровадження керованого антишпигунського рішення, в якому політики встановлюються адміністратором безпеки, а розклад сканування, обслуговування клієнтських ПК, збір інформації про інциденти і створення звітів автоматизовано. Одночасно слід використовувати технології превентивного захисту від шпигунського ПЗ, здатні

видаляти підозрілі програми до того, як вони будуть встановлені на комп'ютер.

В даний час існує досить широкий спектр засобів виявлення і видалення шпигунського ПЗ. Вибір визначається лише масштабом мережі компанії. Так, для невеликої компанії має сенс використовувати рішення незалежного постачальника антишпигунського ПЗ (вони, як правило, дешевше при добрих функціональних можливостях, але не завжди мають гнучку систему управління). При пошуку постачальника допомагають результати порівняльного тестування, що публікуються в Інтернет (наприклад, www.spywareremoversreview.com або www.adwarereport.com). Там же можна викачати пробні версії для тестування. Серед випробників, чиї розробки отримали досить високі оцінки, можна назвати ParetoLogic (XoftSpy), NoAdware (однойменний продукт), AluriaSoftware (Spyware Eliminator), PCTools (Spyware Doctor), Webroot Software (SpySweeper) і ін. В даній області незалежні порівняння не виявлені, тому при виборі постачальника краще покладатися на результати власного попереднього тестування.

Що стосується організацій з складною мережевою інфраструктурою, то для них ситуація не настільки проста. При використанні продуктів незалежних постачальників antispyware часто виникає проблема інтеграції централізованої консолі управління такого продукту з існуючою системою управління. Додатковий засіб управління знижує ефективність реагування на інциденти безпеки (через час, що витрачається адміністратором продукту на обслуговування ще однієї консолі), а також збільшує вартість обслуговування системи в цілому.

Складна мережа вимагає рішення, яке інтегрується в існуючу систему безпеки і забезпечує використання єдиних інструментів управління, у тому числі і для антишпигунських і антивірусних продуктів. Відмітимо, що найбільші розробники антивірусів пропонують свої рішення для боротьби з шпигунським ПЗ у вигляді як окремого продукту, так і вбудованого в антивірусний пакет функціонала. Перелік достатньо широкий: Symantec (Symantec Antivirus Corporate Edition і Symantec Client Security), MacAfee (McAfee Antispyware

Enterprise), Trend Micro (Trend Micro Anti-Spyware Enterprise Edition), Computer Associates (eTrust PestPatrol Anti-Spyware), ESET (NOD32 for Enterprise), Microsoft (Microsoft Defender) і ін. Платний порівняльний аналіз цих засобів «AntiSpyware Product Comparison Guide» доступний на сайті TIPS-IT.com (www.tipsit.com/product.php?pid=3).

Однак, за функціональними можливостями інтегровані продукти часто програють рішенням незалежних постачальників. Але антивірусні компанії зараз активно працюють в цьому напрямку і можна чекати, що в найближчому майбутньому швидкість і якість оновлення антишпигунських баз буде значно підвищено, а спектр виявлюваних погроз, які прийнято відносити до даної категорії, розширений.

1.2.4 Методики пошуку клавіатурних шпигунів

1. Пошук за сигнатурами.

Даний метод не відрізняється від типових методик пошуку вірусів. Сигнатурний пошук дозволяє однозначно ідентифікувати клавіатурні шпигуни, при правильному виборі сигнатур вірогідність помилки практично дорівнює нулю. Проте сигнатурний сканер зможе виявляти лише заздалегідь відомі і описані в його базі даних об'єкти.

2. Евристичні алгоритми.

З назви зрозуміло, що це методики пошуку клавіатурного шпигуна по його характерних особливостях. Евристичний пошук носить імовірнісний характер. Як показала практика, цей метод найбільш ефективний для пошуку клавіатурних шпигунів найпоширенішого типу – тих, що засновані на пастках. Проте подібні методики дають багато помилкових спрацьовувань. Дослідження показали, що існують сотні безпечних програм, що не є клавіатурними шпигунами, але що встановлюють пастки для стеження за клавіатурним вводом і мишею. Найбільш поширені приклади – програми Punto Switcher, словник Lingvo, програмне забезпечення для мультимедійних клавіатур і мишей.

3. Моніторинг функцій Windows API, використовуваних клавіатурними шпигунами.

Дана методика заснована на перехопленні ряду функцій, вживаних клавіатурним шпигуном, – зокрема функцій SetWindowsHookEx, UnhookWindowsHookEx, GetAsyncKeyState, GetKeyboardState. Виклик даних функцій яким-небудь додатком дозволяє вчасно підняти тривогу, проте проблеми багаточисельних помилкових спрацьовувань будуть тими ж, що і при вживанні методу 2.

4. Відстежування використовуваних системою драйверів, процесів і сервісів.

Це універсальна методика, вживана не лише проти клавіатурних шпигунів. У простому випадку можна застосовувати програми типу Kaspersky Inspector або Adinf, які відстежують появу в системних файлів [5].

Всі антивіруси в тій чи іншій мірі можуть знаходити клавіатурних шпигунів, проте клавіатурний шпигун не є вірусом, тому користі від антивірусу мало.

Прикладом утиліт, що реалізовують механізм сигнатурного пошуку і евристичних механізмів пошуку може служити утиліта AVZ, що поєднує сигнатурний сканер і систему виявлення клавіатурних шпигунів на базі пасток.

Спеціалізовані утиліти і програми, призначені для виявлення клавіатурних шпигунів і блокування їх роботи найбільш ефективні для виявлення і блокування клавіатурних шпигунів, оскільки, як правило, можуть блокувати практично всі різновиди клавіатурних шпигунів

1.2.5 Підтримка рівня захисту на належному рівні

Необхідно своєчасно встановлювати патчі, закриваючі виявлені уразливості в ОС і інтернет-браузері.

При установці якого-небудь ПЗ треба уважно читати «ліцензійну угоду», а також супровідну документацію (наприклад, файл ReadMe), де можуть міститися відомості про додаткове ПЗ або про додаткові функції основного ПЗ, пов'язані із

збором і відправкою інформації.

Не можна погоджуватися на установку ПЗ, якщо це пропонується зробити натисненням кнопки «Yes» у вікні повідомлення, що з'явилося на екрані.

Ніколи не можна відповідати на спам-повідомлення і не натискувати кнопки в спливаючих вікнах.

Завжди треба встановлювати найвищий рівень безпеки в інтернет-браузері. Якщо це перешкоджить проглядання потрібних веб-сторінок, можна додати їх в список виключень.

Використання менш популярного, ніж Internet Explorer, інтернет-браузера теж підвищує захищеність від інтернет-погроз, оскільки найчастіше зловмисники використовують уразливості найбільш поширеного ПЗ.

Шпигунське програмне забезпечення може потрапити на комп'ютер користувача двома основними шляхами:

– в ході відвідування сайтів Інтернет. Найчастіше проникнення шпигунського ПЗ відбувається про відвідуванні користувачем хакерських і warez-сайтів, сайтів з безкоштовною музикою і порносайтів. Як правило, для установки шпигунського ПЗ застосовуються ActiveX компоненти або троянські програми категорії TrojanDownloader по класифікації лабораторії Касперського. Багато сайтів хакерів можуть видати "крек", що містить шпигунську програму або TrojanDownloader для її завантаження;

– в результаті установки безкоштовних або умовно-безкоштовних програм. Подібних програм існує велика кількість, вони поширюються через інтернет або на піратських компакт-дисках. Класичний приклад – кодек DivX, що містить утиліту для прихованого завантаження і установки SpyWare.Gator. Більшість програм, що містять SpyWare-компоненти, не повідомляють про це користувача.

Точних критеріїв для занесення програми в категорію "SpyWare" не існує, і дуже часто творці антивірусних пакетів відносять програми категорій "Adware", "Hijacker" і "BHO" до категорії "SpyWare" і навпаки.

Для визначеності пропонується ряд правил і умов, при дотриманні яких програму можна класифікувати як SpyWare. У основу класифікації покладені

проведені автором дослідження найбільш поширених програм SpyWare:

– програма приховано встановлюється на комп'ютер користувача. Сенс даного пункту полягає в тому, що інсталятор звичайної програми повинен повідомити користувача про факт установки програми (з можливістю відмови від установки), запропонувати вибрати каталог для установки і конфігурацію. Крім того, після установки інсталятор повинен створити пункт в списку "Установка і видалення програм", виклик якого виконає процес деінсталяції. Шпигунське програмне забезпечення зазвичай встановлюється екзотичним способом (часто з використанням троянських модулів категорії) приховано від користувача, при цьому його деінсталяція в більшості випадків неможлива. Другий шлях інсталяції SpyWare – прихована установка в комплекті з якою-небудь популярною програмою;

– програма приховано завантажується в пам'ять в процесі завантаження комп'ютера. Варто відзначити, що розробники сучасних SpyWare почали застосовувати Rootkit технології для маскування процесу в пам'яті і файлів на диску. Крім того, стає популярним створення «невбиваних» процесів – тобто запуск двох процесів, які перезапускають один одного в разі зупинки. Така технологія зокрема застосовується в SpyWare.WinAd;

– програма виконує деякі операції без вказівки користувача – наприклад, приймає або передає яку-небудь інформацію з Інтернет;

– програма завантажує і встановлює свої оновлення, доповнення, модулі розширення або інше ПЗ без відома і згоди користувача [7]. Дана властивість притаманна багатьом шпигунським програмам і надзвичайно небезпечна, оскільки завантаження і установка оновлень і додаткових модулів відбувається приховано і часто веде до нестабільної роботи системи. Більш того, механізми автоматичного оновлення можуть бути використані зловмисниками для впровадження на ПК користувача троянських модулів;

– програма модифікує системні налаштування або втручається у функціонування інших програм без відома користувача. Наприклад, шпигунський модуль може змінити рівень безпеки в налаштуваннях браузера

або внести зміни в налаштування мережі;

– програма модифікує інформацію або інформаційні потоки. Типовим прикладом є різні розширення для програми Outlook Express, які при відправці листа приписують до нього свою інформацію. Другий поширений приклад – модифікація завантажуваних з Інтернет сторінок (у сторінки включається рекламна інформація, деякі слова або фрази перетворюються на гіперпосилання).

1.2.6 Аналіз захисту бездротового зв'язку типового пристрою введення інформації

Більшість сучасних бездротових клавіатур використовує для з'єднання із комп'ютером донгл із радіоприймачем. Один з прикладів таких донглів – NRF24LU1.

Проаналізувати трафік від клавіатури можна за допомогою спеціалізованого пристрою – USB-донглу RTL-SDR (рис. 1.12). USB-донгл RTL-SDR можна використовувати у якості комп'ютерного радіо-сканеру для прийому радіосигналів. Залежно від конкретної моделі, він може приймати частоти від 500 кГц до 2.4 ГГц. Більшість програмних засобів для RTL-SDR розробляється мережевою спільнотою та надається безкоштовно.

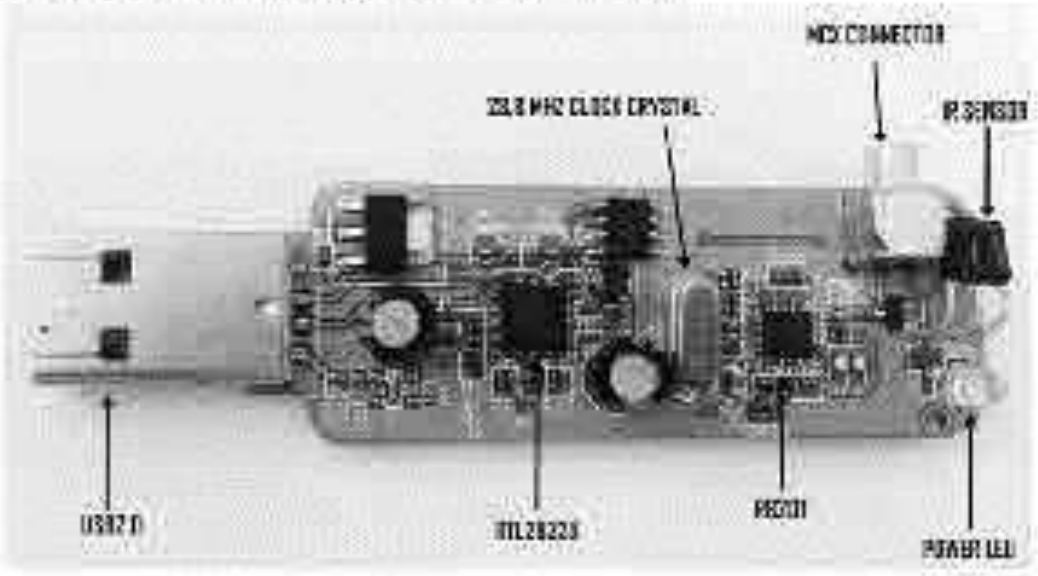


Рисунок 1.12. USB-донгл RTL-SDR

USB-донгл RTL-SDR заснований на чіпсеті RTL2832U [6]. Необроблені дані введення / виведення на чіпсеті RTL2832U можна отримати безпосередньо. Завдяки RTL-SDR будь-хто може отримати доступ до радіоспектру.

Радіо-компоненти, такі як модулятори, демодулятори та тюнери, традиційно реалізуються в аналогових апаратних компонентах. Поява сучасних обчислювальних засобів та аналогово-цифрових перетворювачів дозволяє більшість цих традиційно апаратних компонентів реалізовувати за допомогою програмного забезпечення. Це дозволяє легко обробляти сигнал і, таким чином, створювати дешеві широкосмугові радіо-сканери [6].

У якості програмного застосунку для аналізу сигналу можна використовувати gqrx – програмно-визначений радіоприймач, що працює на GNU Radio і інструментарії Qt GUI. Він може обробляти дані I/Q від різних типів пристроїв введення, включаючи Funcube Dongle Pro / Pro +, rtl-sdr, HackRF і універсальні програмні радіо-периферійні пристрої (USRP).

Виявлено, що для передачі даних натискання клавіші відправляються два пакети – під час натискання клавіші та під час відпускання клавіші. Результат роботи аналізатору зв'язку зображено на рис. 1.13.

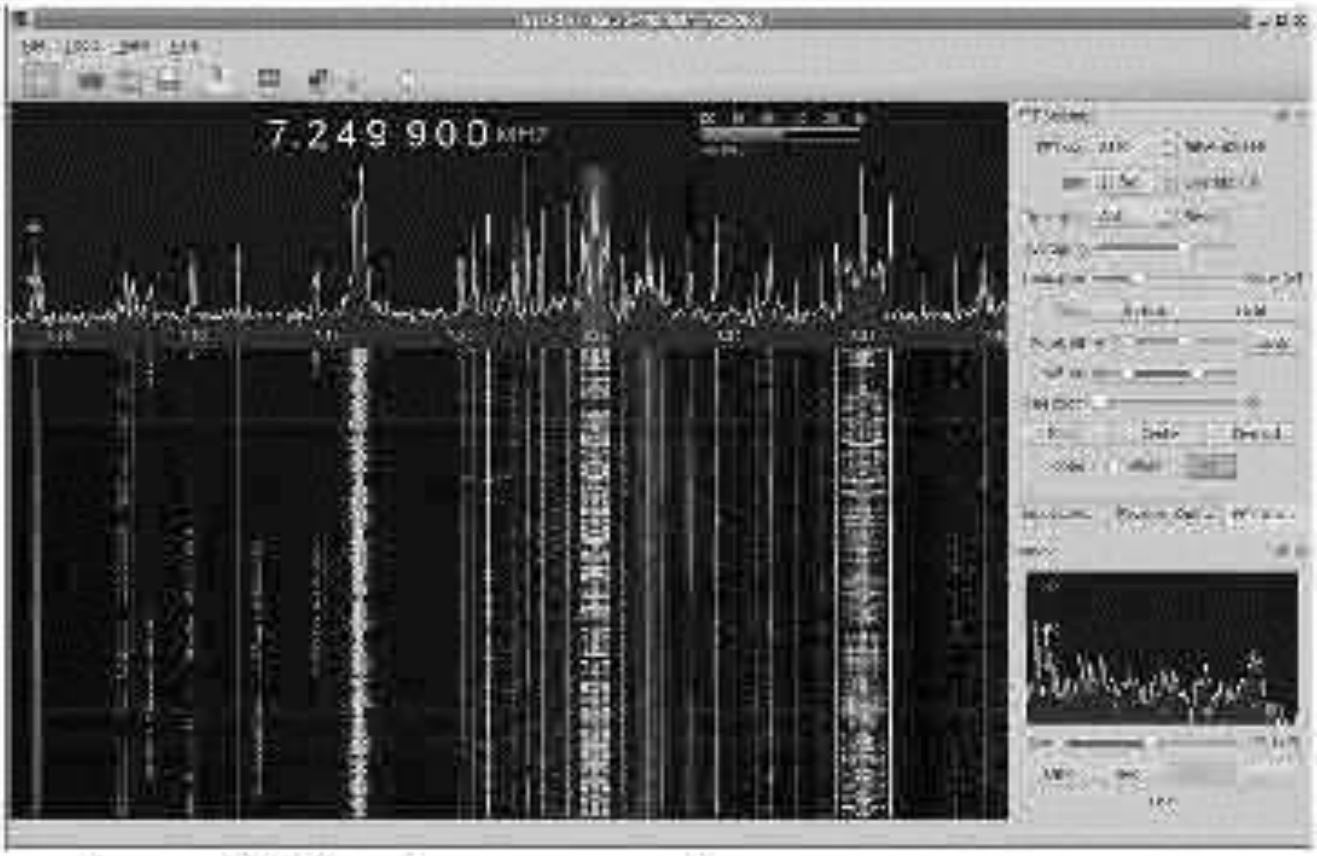


Рисунок 1.13. Аналіз натискання клавіші за допомогою застосунку gqrx

1.3 Практична реалізація методів протидії клавіатурним шпигунам

1.3.1 Утиліта для захисту від клавіатурних шпигунів

Переважна більшість клавіатурних шпигунів використовують для моніторингу натиснень клавіш hook-процедуру WH_KEYBOARD. Щоб клавіатурне повідомлення не потрапило у встановлену пастку, достатньо перехопити виклик цієї hook-процедури і відмінити її. Це можна зробити, встановивши свій hook – WH_DEBUG. Процедура цього hook'а отримуватиме управління при виклику інших hook-процедур (у тому числі і WH_DEBUG, якщо такі вже є). Таким чином, ми встановлюємо hook для інших hook'ів, отримуючи, в результаті, досить потужний засіб [8].

Приступимо до реалізації. Створимо новий DLL-проект; VCL можна відключити.

Код процедури в DLL:

```
extern "C" __export LRESULT CALLBACK
    DebugProc(int nCode, WPARAM wParam, LPARAM lParam)
{
if(nCode == HC_ACTION)
 {
 if(wParam == WH_KEYBOARD)
  {
  if(MessageBox(NULL, "Do you want to pass keyboard message to
WH_KEYBOARD
hook procedure?",
        "Confirmation"
        MB_YESNO | MB_ICONQUESTION | MB_DEFBUTTON2 |
        MB_TOPMOST | MB_SYSTEMMODAL) == IDNO)
   return 1;
  }
 }
```

```
return CallNextHookEx(NULL, nCode, wParam, lParam);
}
```

Робота програми дозволі проста – перевіряється тип викликаємої пастки на відповідність WH_KEYBOARD, і, якщо рівність вірна, запрошується підтвердження на виклик цієї процедури. Якщо в пропуску відмовлено, повертається ненульове значення.

Тепер залишилося написати додаток, відповідальний за установку і зняття цієїпастки.

```
// Глобальні змінні:
HMODULE hDLL = NULL;
HHOOK hHook = NULL;
//---------------------------------------------------------------------
void __fastcall TForm1::Button1Click(TObject *Sender)
{
if(hHook != NULL)
 {
 ShowMessage("The Hook has already been set.");
 return;
 }
hDLL = LoadLibrary("hook.dll");
if(hDLL == NULL)
 {
 ShowMessage("Can't load hook.dll. Reason:\n" +
SysErrorMessage(GetLastError()));
 return;
 }
typedef LRESULT __import (CALLBACK *fnType)(int, WPARAM, LPARAM);
fnType DebugProc = (fnType)GetProcAddress(hDLL, "DebugProc");
if(DebugProc == NULL)
 {
```

```
    FreeLibrary(hDLL);
    hDLL = NULL;
    ShowMessage("Can't find \"DebugProc\" in hook.dll.");
    return;
    }
  hHook = SetWindowsHookEx(WH_DEBUG, (HOOKPROC) DebugProc, hDLL,
0);
  if(hHook == NULL)
    {
    FreeLibrary(hDLL);
    hDLL = NULL;
    ShowMessage("Can't set WH_DEBUG hook. Reason:\n" +
SysErrorMessage(GetLastError()));
    return;
    }
  }
//---------------------------------------------------------------------
void __fastcall TForm1::FormDestroy(TObject *Sender)
{
if(hDLL)
  FreeLibrary(hDLL);
if(hHook)
  {
  UnhookWindowsHookEx(hHook);
  hHook = NULL;
  }
}
```

Недоліки прикладу:

- до моменту запуску цього додатка hook'и WH_KEYBOARD можуть бути встановлені іншими програмами, які не є шпигунами. Тому,

можливо, запит з DLL з'являтиметься при будь-якому натисненні клавіш, навіть якщо не запущені жодні кейлогери.

- при короткочасному натисненні на одну клавішу доведеться відповідати на два повідомлення (оскільки були дві події – натиснення і віджимання). Але повідомлення можна прибрати взагалі, вони потрібні лише для демонстрації.

Сам клавіатурний шпигун може встановити WH_DEBUG-hook і блокувати викликати решти DEBUG-процедур.

1.3.2 Реалізація кейлогера і антикейлогера

У прикладі показано, як за допомогою DLL припинити роботу звичайного KeyLogger'а на основі SetWindowsHookEx. Спочатку представлений програмний код простого KL:

```
library kl;
Uses
  Windows, Messages;
Var
 hook: HHook = 0;
Procedure WriteLog(const log: PChar);
Var
  hFile : THandle;
  dwError: DWord;
  buf: array[0..1] of Char;
  dwWritten: DWord;
Begin
  hFile := CreateFile(PChar('kl'), GENERIC_WRITE, 0, nil, OPEN_ALWAYS,
FILE_ATTRIBUTE_NORMAL, 0);
  Try
   If (hFile <> INVALID_HANDLE_VALUE) Then
   Begin
```

```
      dwError := SetFilePointer(hFile, 0, nil, FILE_END);
      If (dwError <> $FFFFFFFF) Then
      Begin
        WriteFile(hFile, log^, length(log), dwWritten, nil);
        buf[0] := #13;
        buf[1] := #10;
        WriteFile(hFile, buf, 2, dwWritten, nil);
      End;
    End;
    Finally
     CloseHandle(hFile);
    End;
  End;
  Function HookProc(nCode : LongInt; wParam, lParam : LongInt) : LongInt
stdcall;
  Var
    lpszName: Array[0..255] Of Char;
  Begin
    If (nCode = HC_ACTION) And ((lParam shr 31) = 1) Then
    Begin
      GetKeyNameText(lParam, @lpszName, $FF);
      WriteLog(PChar(@lpszName));
    End;
    Result := CallNextHookEx(Hook, nCode, wParam, lParam);
  End;
 procedure sethook(flag:bool);export; stdcall;
 begin
 if flag then
    hook := SetWindowsHookEx(WH_KEYBOARD, @HookProc, hInstance, 0)
 else
```

```
  begin
   unhookwindowshookex(hook);
   hook:=0;
  end;
end;
exports sethook;
begin
end
```

KL записує імена всіх натиснутих клавіш у вказаний файл. Цього достатньо для перевірки і написання антикейлогера.

Антикейлогер це також ж EXE-додаток, встановлюючий Hook за допомогою SetWindowsHookEx і використовуючий DLL. Єдина відмінність в тому, що хук встановлюватиметься не на WH_KEYBOARD, а на WH_DEBUG.

```
library akl;
Uses
  Windows;
Var
 hook:HHook = 0;
Function DebugProc(nCode : LongInt; wParam, lParam : LongInt) : LongInt
stdcall;
Begin
  If (nCode = HC_ACTION) Then
  Begin
    If (wParam = WH_KEYBOARD) Then
    Begin
      Result := 1;
      Exit;
    End;
  End;
  Result := CallNextHookEx(Hook, nCode, wParam, lParam);
```

```
End;
procedure sethook(flag:bool);export; stdcall;
begin
 if flag then
   hook := SetWindowsHookEx(WH_DEBUG @DebugProc, hInstance, 0)
 else
  begin
   unhookwindowshookex(hook);
   hook:=0;
  end;
end;
exports sethook;
begin
end
```

Особливість роботи антикейлогера в тому, що виявивши wParam = WH_KEYBOARD, встановлюється Result := 1 і здійснюється вихід без передачі управління на іншу пастку (CallNextHookEx).

1.4 Практична реалізація пристрою захисту каналу бездротового зв'язку

Для реалізації захисту каналу бездротового зв'язку між клавіатурою та комп'ютером необхідно створити пристрій, що буди підключатися у розрив каналу зв'язку, перехоплювати потік символів від клавіатури та передавати їх до бездротового приймача у шифрованому вигляді. При цьому доцільно використовувати апаратний генератор випадкових чисел, адже саме завдяки цьому передаваний шифрований код можна буде оновлювати через деякий інтервал часу.

1.4.1 Вибір апаратного генератору випадкових чисел

Для генерації випадкових чисел доцільно використовувати фізичні явища для їх утворення. Завдяки аналогово-цифровому перетворювачу можна створити

апаратний генератор випадкових чисел.

Рисунок 1.14. Апаратний генератор випадкових чисел Infinite Noise TRNG

Один із пристроїв, що використовує тепловий шум – Infinite Noise TRNG, апаратний генератор випадкових чисел з USB-ключем (рис. 1.14). Він використовує "модульний множник ентропії" (Infinite Noise Multiplier або FireBug) та виконує розподіл Пуассона. Пристрій природним чином захищає від впливу зовнішніх сигналів, таких як радіоперешкоди та перешкоди в джерелі живлення, що спрощує створення надійної конструкції без залучення фахівця з аналогової схемотехніки [11]. Модульні множники ентропії виробляють доказовий і легко вимірюваний рівень ентропії, заснований на тепловому шумі, приблизно рівний $\log_2(K)$ на вихідний біт, де K – коефіцієнт посилення між 1 і 2, встановлений двома резисторами навколо операційного підсилювача. "Монітор працездатності" може відстежувати це і перевіряти, чи знаходиться ентропія на виході в очікуваному діапазоні, який для описаного нижче нескінченного шуму TRNG знаходиться в межах 2% від $\log_2(1.82)$.

Модульні ентропійні множники підходять як для реалізації на рівні плати, так і для реалізації ASIC. Швидкість обмежена швидкістю каскаду посилення та компаратора і може працювати зі швидкістю понад 100 Мбіт/с за секунду з високопродуктивними компонентами. Більш простіші схеми з чотириканальними операційними підсилювачами CMOS можуть працювати зі швидкістю 8 Мбіт/с.

Суміжні біти з модульного ентропійного множника корельовані, тому перед використанням необхідно повторно обчислювати криптографічно безпечну хеш-функцію, таку як SHA-512, Blake2b, Keccak-1600 (SHA3), або потоковий шифр, такий як ChaCha. У даній реалізації використовується Keccak-

1600 з криптографічно безпечним повторним заповненням більше 400 біт ентропії за раз, що дозволяє подолати проблему низької швидкості, яка існує у системі GNU/Linux/dev/random.

1.4.2 Схема апаратної частини пристрою захисту каналу зв'язку

Схему апаратної частини пристрою захисту каналу зв'язку на базі мікроконтролеру ATMega328P-PU розроблено у середовищі NI Multisim – це САПР для проектування електронних схем та обладнання у галузі електроніки. Створену схему пристрою зображено на рис. 1.15.

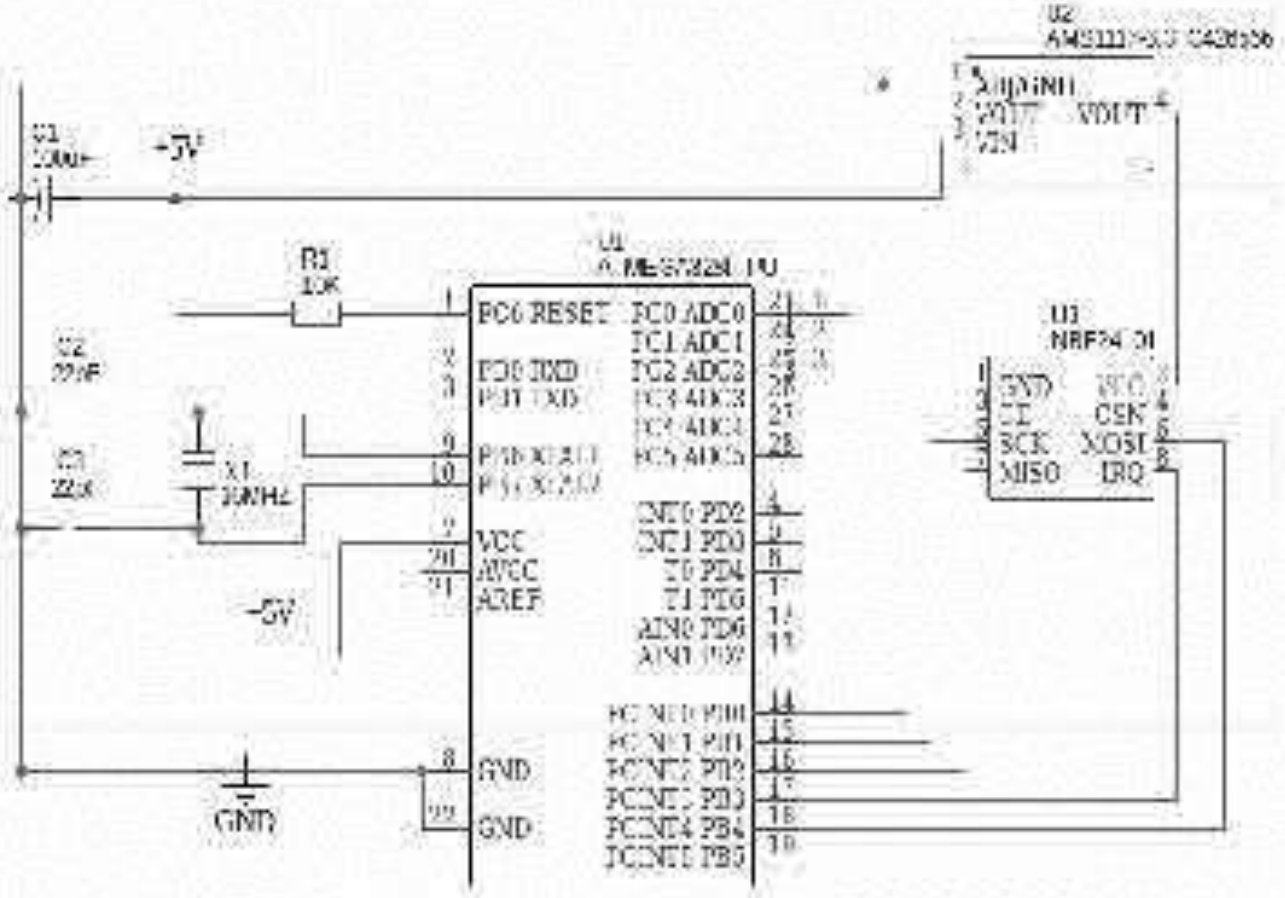


Рисунок 1.15. Принципова електрична схема пристрою захисту каналу зв'язку

1.4.3 Програмне забезпечення для пристрою захисту каналу зв'язку

Створений на базі мікроконтролеру ATMega328P-PU пристрій захисту каналу зв'язку має забезпечити роботу апаратного генератора випадкових чисел та перетворювача бездротового інтерфейсу радіо-каналу зв'язку між клавіатурою та комп'ютером. Лістинг програми на мові програмування С

прошивки мікроконтролеру ATMega328P-PU у вигляді файлів Makefile та main.c наведено нижче:

Лістинг файлу Makefile:

```
MCU=atmega328p
PROGRAMMER=usbasp
F_CPU=16000000
CC=avr-gcc
OBJCOPY=avr-objcopy
CFLAGS=-std=c99 -Wall -g -Os -mmcu=$(MCU) -DF_CPU=$(F_CPU) -I
TARGET=target\main
SRCS=src\main.c src\rtf24101.c
all:
mkdir -p target
$(CC) $(CFLAGS) -o $(TARGET).bin $(SRCS)
$(OBJCOPY) -j .text -j .data -O ihex $(TARGET).bin $(TARGET).hex
flash:
avrdude -p $(MCU) -c $(PROGRAMMER) -U flash:w:$(TARGET).hex:i -F -P
usb
clean:
rm -fr target
```

Лістинг файлу main.c:

```
#include <avr\io.h>
#include <avr\interrupt.h>
#include <math.h>
#include <stdbool.h>
#include <string.h>
#include <util\delay.h>
#include "rtf24101.h"
#include "rtf24101-mnemonics.h"
```

```
nRF24L01 *setup_rf(void);
void process_message(char *message);
void prepare_led_pin(void);
void set_led_high(void);
void set_led_low(void);
double poisson_keypress(void);
static char *rand_payload(char *str, size_t size);
void delay_ms(uint16_t n);
void delay_us(uint16_t n);
uint16_t hw_random();
double poisson_dist(uint16_t lambda, uint16_t input, uint16_t max_input);
volatile bool rf_interrupt = false;
int main(void) {
char * tmp_payload;
uint8_t address[5] = { 0x6B, 0xB7, 0x82, 0x9E, 0x31 }; //predefined keyboard
prepare_led_pin();
sei();
nRF24L01 *rf = setup_rf();
uint8_t addr[5];
nRF24L01_read_register(rf, CONFIG, addr, 1);
tmp_payload = (char *) malloc(16);
set_led_high();
while(1){
return 0;
}
nRF24L01Message msg;
rand_payload(tmp_payload, 16);
memcpy(msg.data, tmp_payload, 16);
msg.length = 17;
nRF24L01_transmit(rf, address, &msg);
```

```
set_led_high();
delay_us((int)(poisson_dist(1, hw_random() %(4096, 4096)*4096));
set_led_low();
}
nRF24L01 *setup_rf(void) { nRF24L01 *rf = nRF24L01_init();
rf->ss.port = &PORTB;
rf->ss.pin = PB2;
rf->ce.port = &PORTB;
rf->ce.pin = PB1;
rf->sck.port = &PORTB;
rf->sck.pin = PB5;
rf->mosi.port = &PORTB;
rf->mosi.pin = PB3;
rf->miso.port = &PORTB;
rf->miso.pin = PB4;
// interrupt on falling edge of INT0 (PD2) EICRA |= _BV(ISC01);
EIMSK |= _BV(INT0);
nRF24L01_begin(rf);
return rf;
}
void prepare_led_pin(void) { DDRC |= _BV(PC5);
PORTC &= ~_BV(PC5);
}
void set_led_high(void) { PORTC |= _BV(PC5);
}
void set_led_low(void) { PORTC &= ~_BV(PC5);
}
// nRF24L01 interrupt ISR(INT0_vect) {
rf_interrupt = true;
}
```

```
uint16_t hw_random () {sei();
ADMUX = 0b11111111;
ADCSRA = 0b10001100;
ADCSRA = ADCSRA | (1<< ADSC);
ADCSRB = 0b10001100;
ADCSRB = ADCSRB | (1<< ADSC);
return ADCSRA|ADCSRB;
}
double poisson_dist(uint16_t lambda, uint16_t input, uint16_t max_input){
return -(1./lambda)*(double)log((double)input/(double)max_input);
}
static char *rand_payload(char *str, size_t size)
{
if (size) {
--size;
for (size_t n = 0; n < size; n++) { char key = hw_random() %255;
str[n] = key;
}
}
return str;
}
void delay_ms(uint16_t n) {while(n--) {
_delay_ms(1);
}
}
void delay_us(uint16_t n) {while(n--) {
_delay_us(1);
}
}
```

1.4.5 Аналіз роботи пристрою захисту каналу зв'язку

Проаналізуємо спочатку роботу клавіатури без підключення розробленого пристрою протидії атакам. Введемо, наприклад, слово "Привіт" та проаналізуємо діаграму потоку введення, отриману за допомогою USB-донглу RTL-SDR (рис. 1.16).

Рисунок 1.16. Діаграма потоку введення без підключення пристрою захисту

Після підключення і увімкнення розробленого пристрою протидії атакам та повторного введення слова "Привіт", отримано діаграму потоку клавіатури, зображену на (рис. 1.17).

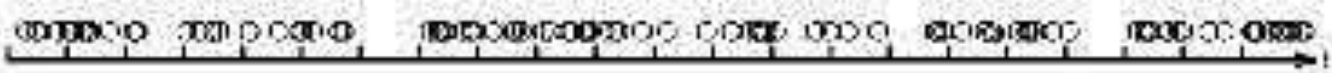


Рисунок 1.17. Діаграма потоку введення без підключення пристрою захисту

За отриманими та наведеними на рис. 1.16 та 1.17 можна зазначити, розроблений пристрій захисту бездротового зв'язку надійно приховує передавані повідомлення від перехоплення зловмисниками. Відповідно до отриманих результатів можна зазначити, що виконання зловмисниками атаки за часом є неможливим, адже пристрій захисту створив непередбачувану серію "фантомних" кодів натиснень за допомогою апаратного генератору випадкових чисел. Проблем у роботі клавіатури під час використання протягом години помічено не було. Розроблений пристрій є працездатним, виконує поставлену мету та не заважає штатній роботі клавіатури.

2 ОХОРОНА ПРАЦІ

2.1 Вступ

Закон України "Про охорону праці" визначає основні положення по реалізації конституційного права громадян на охорону їх життя і здоров'я в процесі трудової діяльності, регулює взаємини між адміністрацією і працівником в незалежності від форм власності, встановлює єдиний порядок організації охорони праці в Україні [12].

Згідно закону України «Про підприємства в Україні» усі роботодавці повинні турбуватись про дотримання у своєї діяльності вимог законів України стосовно охорони праці та навколишнього природного середовища.

2.2 Структура управління охороною праці на підприємстві

Система управління охороною праці (СУОП) є комплексом дій з підготовки, прийняття та реалізації рішень з метою виконання організаційних, технічних, санітарно-гігієнічних і лікувально-профілактичних заходів.

Головна мета введення СУОП на підприємстві – забезпечення безпеки, збереження життя, здоров'я та працездатності працівників під час трудового процесу.

Управління охороною праці здійснюється на підприємстві у цілому – директором підприємства безпосередньо та через заступника. У підрозділах та відділах – керівниками підрозділів. Контроль за дотриманням вимог із питань охорони праці та навколишнього середовища, підготовка звітності, рішень та пропозицій щодо покращення умов праці, виконує фахівець із охорони праці.

2.3 Аналіз та безпека умов праці працівників на робочому місці

Науково-технічний прогрес призвів до серйозних змін в умови виробничої діяльності робітників розумової праці. Їх праця стала більш інтенсивним, напруженим, які вимагають значних витрат розумової, емоційної і фізичної енергії. Це зажадало комплексного рішення проблем ергономіки, гігієни і організації праці, регламентації режимів праці та відпочинку.

Робота з комп'ютером характеризується значною розумовою напругою і нервово-емоційним навантаженням операторів, високою напруженістю зорової роботи і достатньо великим навантаженням на м'язи рук при роботі з клавіатурою ЕОМ. Велике значення має раціональна конструкція і розташування елементів робочого місця, що важливо для підтримки оптимальної робочої пози розробника.

2.3.1 Загальна характеристика приміщення та робочого місця

Приміщення офісу, в якому проводиться розробка за завданням. Згідно з НПАОП 0.00-1.28-2010 [14] в приміщені може перебувати 6 працівників.

За умовами завдання це виконується повністю. В приміщенні відсутні умови, які можуть створювати підвищену або особливо підвищену небезпеку, тому воно відноситься до класу звичайних приміщень (згідно ПУЕ). Джерелом живлення є трифазна мережа напруги 380/220 В з глухо заземленою нейтраллю, з частотою 50 Гц (згідно НПАОП 0.00-1.28-2010)

2.3.2 Мікроклімат

Приміщення для роботи з комп'ютерами мають бути обладнані системами опалення, кондиціонування повітря, або припливно-витяжною вентиляцією. У приміщеннях на робочих місцях мають забезпечуватись оптимальні значення параметрів мікроклімату: температури, відносної вологості й рухливості повітря у відповідності до ГОСТ 12.1.005-88, СН 4088-86. Для підтримки допустимих значень мікроклімату та концентрації позитивних та негативних іонів необхідно передбачати установки або прилади зволоження та/або штучної іонізації, кондиціювання повітря.

2.3.3 Освітлення

Приміщення, в яких встановлені комп'ютери, повинні мати природне та штучне освітлення відповідно до СНиП II-4-79. Природне освітлення має здійснюватися через світлові прорізи. Штучне освітлення в приміщеннях з робочими місцями має здійснюватися системою освітленості (КПО) не нижче ніж 1,5%. загального рівномірного освітлення. Застосування світильників без

розсіювачів та екрануючих грат заборонено. Слід передбачити обмеження прямого відблиску від джерел природного та штучного освітлення.

2.3.4 Шум та вібрація

Шум та вібрація у робочому приміщенні У приміщенні причинної шуму і вібрації являються апарати, прилади і устаткування друкуючі пристрої, комп'ютери, вентилятори, кондиціонер та ін. При їхній роботі рівень вібрації не вище 33 дБ, рівень шуму не повинен перевищувати 50 дБА, що є нормою для даного виду діяльності відповідно до НПАОП 0.00-1.28-2010 [14]. Заходи по забезпеченню встановлених норм: використання спеціальних шум-поглинаючих перегородок, застосування меблів, які сприяють зменшенню шуму і вібрації, установка апаратів і приладів на спеціальні амортизуючи підкладки. Під матричні принтери потрібно підкладати вібраційні килимки для гасіння вібрації та шуму.

2.3.5 Електробезпека

Персональні комп'ютери, периферійні пристрої, інше устаткування (апарати управління, контрольно-вимірювальні прилади, світильники), електропроводи та кабелі за виконанням і ступенем захисту мають відповідати класу зони, мати апаратуру захисту від струму короткого замикання та інших аварійних режимів.

Під час монтажу та експлуатації ліній електромережі необхідно повністю унеможливити виникнення електричного джерела загорання внаслідок короткого замикання та перевантаження проводів, обмежувати застосування проводів з легкозайнистою ізоляцією і, за можливості, застосовувати негорючу ізоляцію. Лінія електромережі для живлення персональних комп'ютерів і периферійних пристроїв виконується як окрема групова трипровідна мережа шляхом прокладання фазового, нульового робочого та нульового захисного провідників. Не допускається підключати комп'ютери та периферійні пристрої до звичайної двопровідної електромережі, в тому числі з використанням перехідних пристроїв.

2.3.6 Організація обладнання робочого та місця.

При розміщенні робочих столів з персональними комп'ютерами слід дотримувати:

- відстань між бічними поверхнями персональних комп'ютерів 1,2 м.;
- відстань від тильної поверхні одного персонального комп'ютера до екрана – 2.5 м.

Конструкція робочого місця користувача персонального комп'ютера має забезпечити підтримання оптимальної робочої пози офісного працівника. Робочий стіл має відповідати сучасним вимогам ергономіки і забезпечувати оптимальне розміщення на робочій поверхні використовуваного обладнання (дисплея, клавіатури, принтера) і документів. Крім того, він повинен мати простір для ніг заввишки не менше ніж 600мм, завширшки не менше ніж 500мм, завглибшки (на рівні колін) не менше ніж 450мм, на рівні простягнутої ноги не менше кутом і нажилу сидіння та спинки і за відстанню від спинки до переднього краю сидіння поверхня сидіння має бути плоскою, передній край – заокругленим.

2.4 Режими праці та відпочинку

При організації праці, що пов'язана з використанням персональних комп'ютерів, для збереження здоров'я працюючих, запобігання професійним захворюванням і підтримки працездатності слід передбачити внутрішньозмінні регламентовані перерви для відпочинку. За основну роботу з персональним комп'ютером слід вважати таку, що займає не менше 50% часу впродовж робочої зміни. Тривалість обідньої трудової перерви визначається чинним законодавством про працю і Правилами внутрішнього трудового розпорядку. Встановлюються такі внутрішньозмінні режими праці та відпочинку при роботі з ЕОМ при 8-годинній денній робочій зміні в залежності від характеру праці. Для розробників програм слід призначати регламентовану перерву для відпочинку тривалістю 15 хвилин через кожну годину роботи за персональним комп'ютером.

У всіх випадках, коли виробничі обставини не дозволяють застосувати регламентовані перерви, тривалість безперервної роботи з персональним комп'ютером не повинна перевищувати 4 години. При 12-годинній робочій зміні перерви повинні встановлюватися в перші 8 годин роботи аналогічно перервам при 8-годинній робочій зміні, а протягом останніх 4-х годин роботи, незалежно від характеру трудової діяльності, через кожну годину тривалістю 15 хвилин.

2.5 Пожежна безпека

Будівля та приміщення, де розміщені робочі місця, відповідають вимогам нормативно-технічної та експлуатаційної документації виробника персональних комп'ютерів ДСанПіН 3.3.2-007-98 та Правил.

При виконанні завдання випускної роботи було звернено увагу на предмети обстановки та наявність: пошкоджених пристроїв, обладнання, дротів тощо; розкиданих предметів побутового вжитку (пакетів, коробок, сумок, валіз тощо); металевих, скляних банок з під напоїв, продуктів, фарби; предметів, з характерним звуком роботи годинникового механізму, індикаторними лампочками, вимикачами. З вище перерахованого нічого не було виявлено.

Найбільш зручними для використання в умовах офісу є вогнегасники. Попри обладнання будівлі установкою пожежогасіння, пожежною сигналізацією та внутрішніми пожежними кранами, офісні приміщення також забезпечені первинними засобами пожежогасіння. В кожній організації наказом або розпорядженням керівника повинна бути призначена особа, відповідальна за експлуатацію вогнегасників.

2.6 Висновки за розділом

Під час виконання випускної роботи були дотримані всі основні норми охорони праці на об'єкті. Були дотримані норми щодо роботи з персональним комп'ютером та периферійними пристроями і норми відпочинку під час роботи для уникнення небезпечних факторів для здоров'я. Також були дотримані всі правила та норми пожежної безпеки, електробезпеки, тощо. Під час робіт небезпечних факторів не виявлено.

ВИСНОВКИ

Проведений у випускній роботі аналіз методів захисту каналів зв'язку пристроїв клавіатурного введення дозволив відібрати оптимальніші з них та на їх основі розробити дієві механізми протидії атакам.

Визначено, що клавіатурний шпигун не є вірусом, але проте представляє велику загрозу для користувачів, оскільки дозволяє зловмисникові стежити за роботою користувача і може застосовуватися для викрадання конфіденційної інформації, у тому числі паролів користувача.

Найчастіше застосовуються методики перехоплення функцій клавіатурного введення в режимі користувача, але останнім часом з'явилися вельми ефективні реалізації із застосуванням драйверів. Відомо, що сьогодні існує універсальна і надійна методика, що дозволяє обійти апаратний клавіатурний шпигун, – це використання екранної клавіатури і інших способів вводу інформації без застосування клавіатури. Пошук апаратних кейлогерів неодмінно слід включити в посадові обов'язки співробітників служби інформаційної безпеки.

Визначено, що переважна більшість клавіатурних шпигунів використовує для моніторингу натискань клавіш hook-процедуру для перехоплювача. Кейлогер записує імена всіх натиснутих клавіш у зазначений файл. Проаналізовані у роботі методи захисту каналів зв'язку пристроїв клавіатурного введення дозволили реалізувати практично кейлогер та антикейлогер, а також пристрій захисту каналу бездротового зв'язку між клавіатурою та комп'ютером. Пристрій активного захисту формує фіктивні пакети, розраховує затримку та відправляє у ефір. Такий метод захищає наявний зв'язок, а завдяки випадковій затримці емулює активну роботу пристрою введення інформації та не заважає його нормальній роботі.

Розроблений метод протидії атакам може використовуватися в складі автономного пристрою активного захисту, а також може бути інтегрований у системи введення інформації.

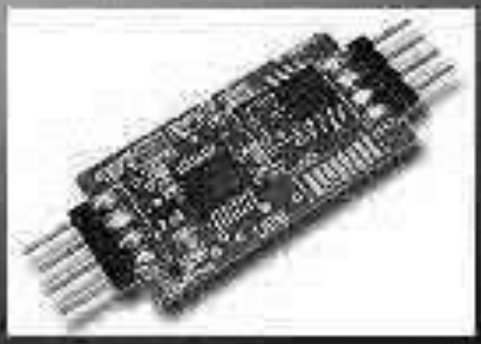
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Щеглов А.Ю. Защита компьютерной информации от несанкционированного доступа. – Санкт-Петербург: Наука и техника, 2004. – 384 с.
2. С Г. Баричев, В.В. Гончаров, Р.Е. Серов, Основы современной криптографии, 2-е издание, Москва, "Горячая линия - Телеком", 2002
3. Нильс Фергюсон, Брюс Шнайер Практическая криптография – Москва: «Диалектика», 2004. – С. 420
4. Дьюхарст Программирование на C++ / Дьюхарст, Старк Стефан; Кэти. – М.: ДиаСофт, 2015. – 272 с.
5. Фридман, А. C/C++. Архив программ / А. Фридман, Л. Кландер, М. Михэлис, и др. – М.: ЗАО Издательство БИНОМ, 2016. – 640 с.
6. Статья «Охота за шпионом, или АнтиКейлоггер» [Електронний ресурс] – режим доступу: http://www.sources.ru/magazine/0805/artikeylogger.html
7. Об опасностях беспроводных клавиатур и мышей [Електронний ресурс] – Режим доступу: https://habr.com/company/ptblog/325932/.
8. Обзор видов атак по побочным каналам на криптографические устройства. Подлесное А. В. Молодой учёный. 2015. № 1. с. 187-189
9. Реализация DDIO в чипах Intel допускает сетевую атаку по определению нажатий клавиш в сеансе SSH [Електронний ресурс] – Режим доступу: https://www.opennet.ru/opennews/art.shtml?num=51467
10. Статья «C++: Защита от клавиатурных шпионов» [Електронний ресурс] – режим доступу: http://www.buildercpp.net.ru/articles/raznoe/raz035.htm
11. ATMega8 – 8-bit AVR [Електронний ресурс] – Режим доступу: https://www.microchip.com/wwwproducts/en/ATMEGA128.
12. Закон України Про охорону праці, №235-IV, 22.11.2002.
13. ДНАОП 0.03-8.03-97 Гігієнічна класифікація праці за показниками шкідливості та небезпечності факторів виробничого середовища, важкості та напруженості трудового процесу.
14. ГОСТ 12.003–74 ССБТ. Опасные и вредные производственные факторы.

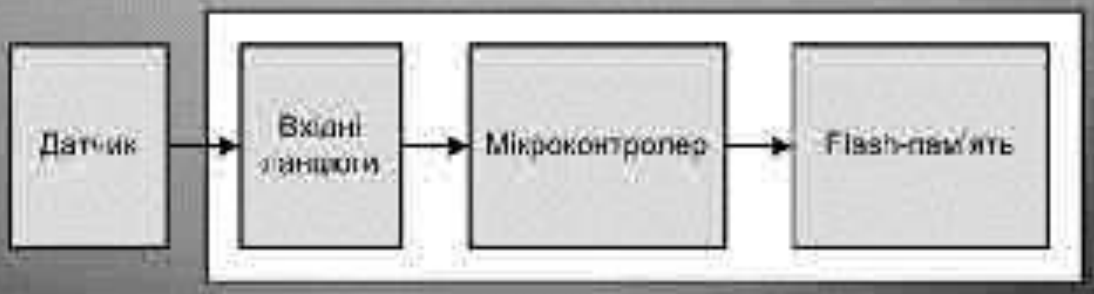
ДОДАТОК А

Слайди мультимедійної презентації

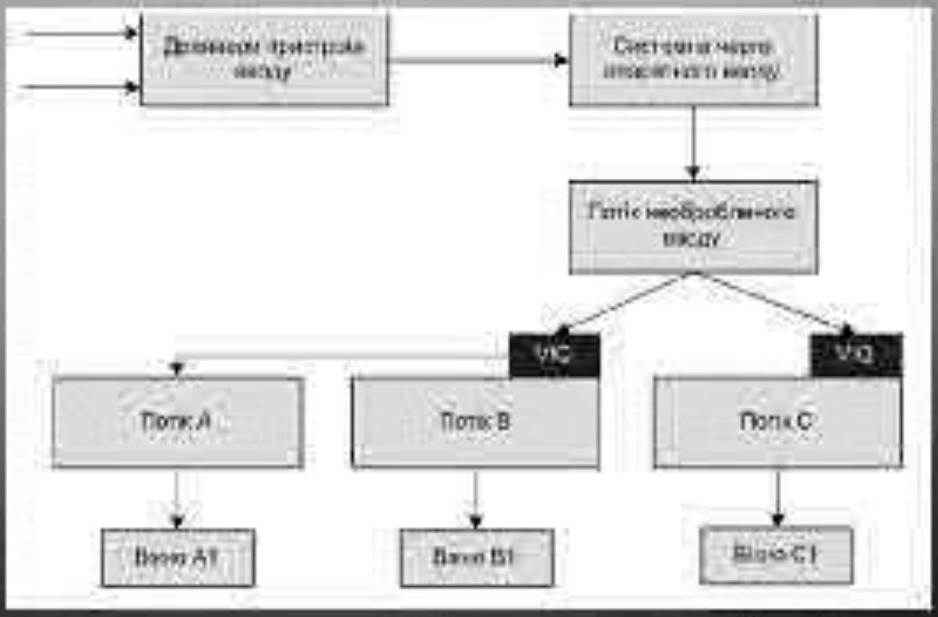
КЕЙЛОГЕРИ ДЛЯ ФІКСАЦІЇ СИМВОЛІВ, НАБРАНИХ З КЛАВІАТУРИ



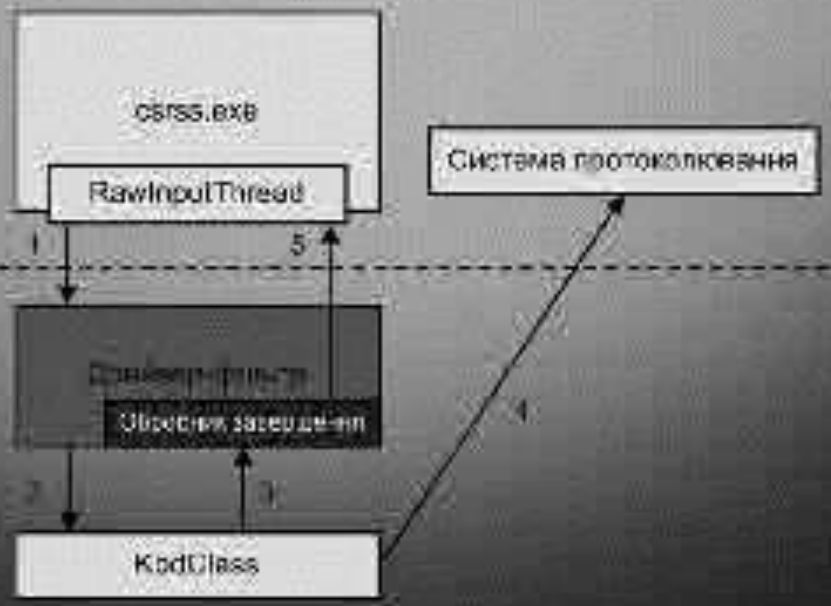
ФУНКЦІОНАЛЬНА СХЕМА АПАРАТНОГО КЕЙЛОГЕРА



МОДЕЛЬ АПАРАТНОГО ВВОДУ СИСТЕМИ WINDOWS

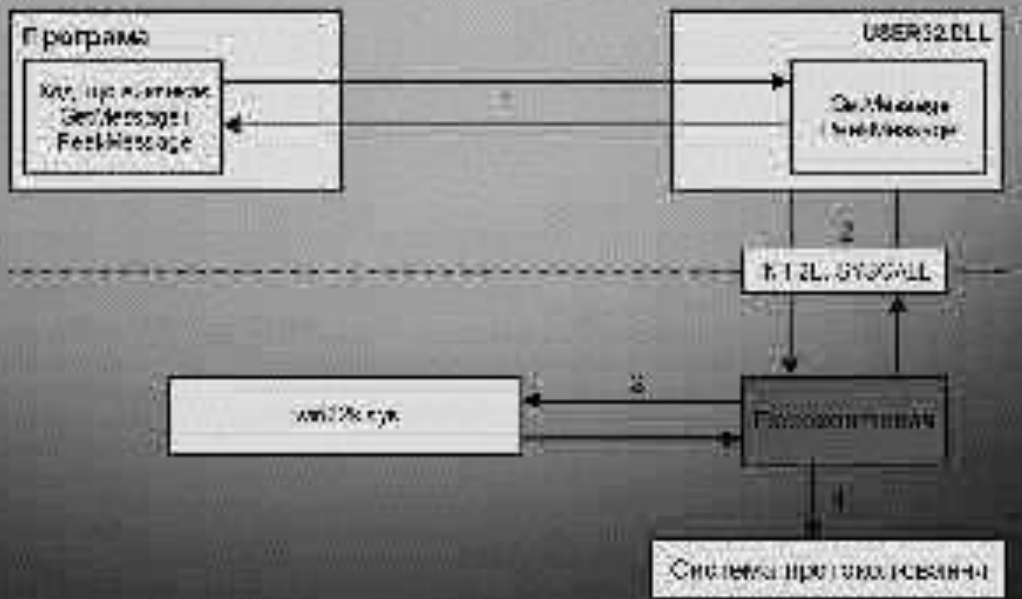


ПІДКЛЮЧЕННЯ ДРАЙВЕРА-ФІЛЬТРА ДО СТЕКА КЛАВІАТУРНОГО ДРАЙВЕРУ



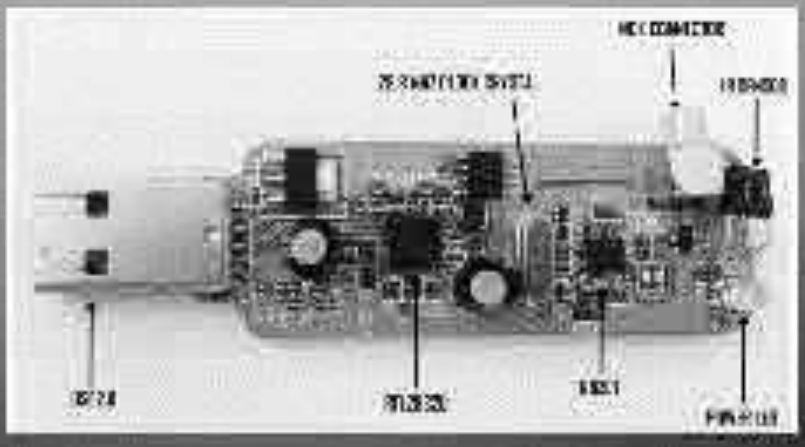
ПЕРЕХОПЛЕННЯ ФУНКЦІЙ GETMESSAGE І PEEKMESSAGE

ПЕРЕХОПЛЮВАЧ НА БАЗІ РУТКІТ-ТЕХНОЛОГІЇ В KERNELMODE

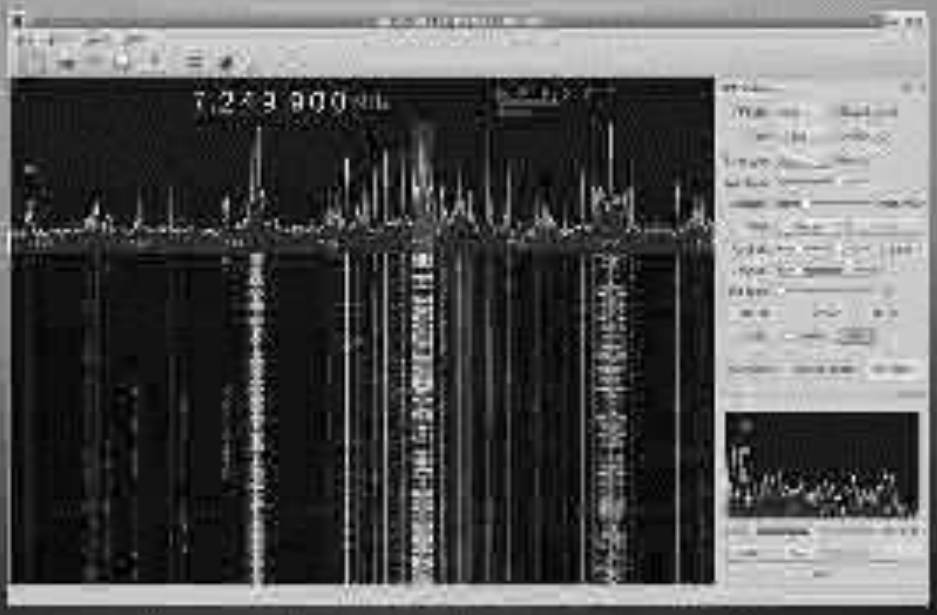


ІЛЮСТРАЦІЯ ЗАТРИМОК ПРИ НАБОРІ НА КЛАВІАТУРІ

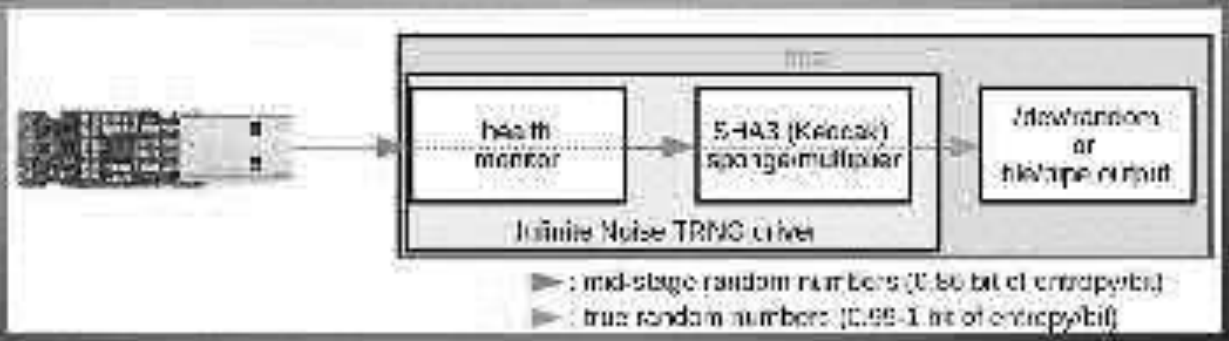
USB-ДОНГЛІ RTL-SDR ДЛЯ АНАЛІЗУ ТРАФІКУ ВІД КЛАВІАТУРИ



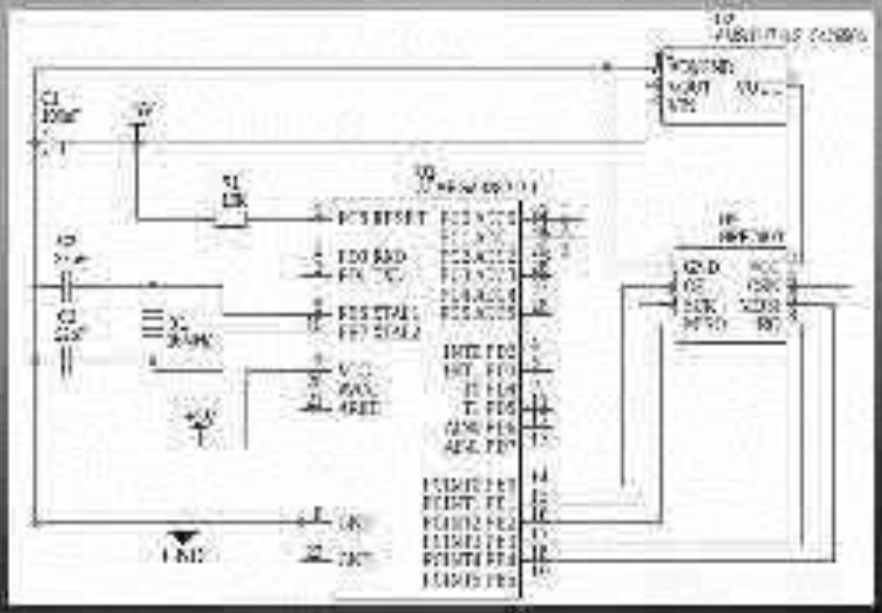
АНАЛІЗ НАТИСКАННЯ КЛАВІШ ЗА ДОПОМОГОЮ ЗАСТОСУНКУ GQRX



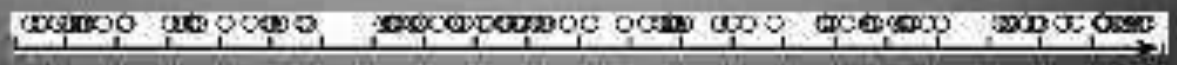
АПАРАТНИЙ ГЕНЕРАТОР ВИПАДКОВИХ ЧИСЕЛ INFINITE NOISE TRNG



ПРИНЦИПОВА ЕЛЕКТРИЧНА СХЕМА ПРИСТРОЮ ЗАХИСТУ КАНАЛУ БЕЗДРОТОВОГО ЗВ'ЯЗКУ З КЛАВІАТУРОЮ



Аналіз роботи пристрою захисту каналу зв'язку



Діаграма потоку сообщений без підключення і з пристрою захисту