



(19) **UA** (11) **57 855** (13) **C2**
(51)МПК ⁷ **G 06F 9/42, 12/14**

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
УКРАИНЫ

ГОСУДАРСТВЕННЫЙ ДЕПАРТАМЕНТ
ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ УКРАИНЫ

(21), (22) Заявка: 2001042325, 01.10.1999

(24) Дата начала действия патента: 15.07.2003

(30) Приоритет: 09.10.1998 DE 198 46 676.5

(46) Дата публикации: 15.07.2003

(86) Заявка PCT:
PCT/DE99/03169, 19991001

(72) Изобретатель:

Мей Кристиан, DE,
Фрайвальд Юрген, DE,
Бриксель Олаф, DE

(73) Патентовладелец:

ИНФИНЕОН ТЕКНОЛОДЖИЗ АГ, DE

(54) СПОСОБ ЗАЩИТЫ АДРЕСОВ ТОЧЕК ВХОДА

(57) Реферат:

В соответствии с изобретением предлагается более эффективный способ защиты от неразрешенного доступа к точкам входа в компьютерные программы. Предлагаемый способ заключается в том, что активизация разрешенных адресов точек входа осуществляется непосредственно, а для распознавания разрешенных адресов точек входа используется корреляция между данными, которые не могут одновременно присутствовать в одной и той же команде. Программа имеет структуру, при которой

компилятор или компоновщик обеспечивает то, что указанной корреляции между данными соответствуют только разрешенные адреса входа в программу.

Официальный бюллетень "Промышленная собственность". Книга 1 "Изобретения, полезные модели, топографии интегральных микросхем", 2003, N 7, 15.07.2003. Государственный департамент интеллектуальной собственности Министерства образования и науки Украины.



(19) **UA** (11) **57 855** (13) **C2**
(51) Int. Cl.⁷ **G 06F 9/42, 12/14**

MINISTRY OF EDUCATION AND SCIENCE OF
UKRAINE

STATE DEPARTMENT OF INTELLECTUAL
PROPERTY

(12) **DESCRIPTION OF PATENT OF UKRAINE FOR INVENTION**

(21), (22) Application: 2001042325, 01.10.1999

(24) Effective date for property rights: 15.07.2003

(30) Priority: 09.10.1998 DE 198 46 676.5

(46) Publication date: 15.07.2003

(86) PCT application:
PCT/DE99/03169, 19991001

(72) Inventor:

May Christian, DE,
Friewalde Jurgen, DE,
Brixel Olaf, DE

(73) Proprietor:

INFINEON TECHNOLOGIES AG, DE

(54) **METHOD FOR PROTECTING ENTRY ADDRESSES**

(57) Abstract:

A substantially more efficient method for protecting entry addresses in computer programs, where by authorized entry addresses can be activated in a direct manner and can be recognized by a correlation of data that cannot be found inside the same individual instructions. Organization of the program code enables

compilers or linkers to guarantee that only legal entry addresses correspond to said correlation.

Official bulletin "Industrial property". Book 1 "Inventions, utility models, topographies of integrated circuits", 2003, N 7, 15.07.2003. State Department of Intellectual Property of the Ministry of Education and Science of Ukraine.



(19) **UA** (11) **57 855** (13) **C2**
(51)МПК ⁷ **G 06F 9/42, 12/14**

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ДЕРЖАВНИЙ ДЕПАРТАМЕНТ
ІНТЕЛЕКТУАЛЬНОЇ ВЛАСНОСТІ

(12) ОПИС ВІНАХОДУ ДО ПАТЕНТУ УКРАЇНИ

(21), (22) Дані стосовно заявки:
2001042325, 01.10.1999

(24) Дата набуття чинності: 15.07.2003

(30) Дані стосовно пріоритету відповідно до Паризької
конвенції : 09.10.1998 DE 198 46 676.5

(46) Публікація відомостей про видачу патенту
(деклараційного патенту): 15.07.2003

(86) Номер та дата подання міжнародної заявки
відповідно до договору РСТ:
PCT/DE99/03169, 19991001

(72) Винахідник(и):

Мей Крістіан , DE,
Фрайвальд Юрген , DE,
Бріксель Олаф , DE

(73) Власник(и):

ІНФІНЕОН ТЕКНОЛОДЖІС АГ, DE

(54) СПОСІБ ЗАХИСТУ АДРЕС ПЕРЕХОДІВ

(57) Реферат:

Запропоновано більш ефективний спосіб захисту адрес переходів, причому допустимі адреси переходів можуть бути використані безпосередньо і можуть бути розпізнані шляхом встановлення кореляції між даними, які не можуть

перебувати в одній і тій же окремій інструкції. При цьому за допомогою компілятора або компоновщика може бути забезпечено, що лише легальні адреси переходів виконують умову цієї кореляції.

Опис винаходу

Для роботи на майбутніх чіп-картках мають бути допущені прикладні програми третіх фірм, які можуть викликати функції операційної системи. При цьому може існувати небезпека, що ці прикладні програми третіх фірм містять спроби маніпулювання шкідництва стосовно інших частин програм. Таке втручання могло б бути здійснене не шляхом використання дійсного посилання функції програми операційної системи із прикладної програми, а шляхом використання іншої адреси переходу. Внаслідок цього код операційної системи міг би бути виконаним неправильно і, наприклад, могла б бути спровокована втрата інформації шляхом помилкового переписування вмісту області пам'яті.

Згідно з рівнем техніки планують перехід на попередньо задані адреси з постійними інтервалами. При цьому спочатку слід перевірити, чи лежить попередньо задана адреса у діапазоні адрес, дозволених для модуля. В разі позитивного результату здійснюється перехід на попередньо задану адресу, а звідти виконується подальший перехід на власне функцію. Недоліком такого способу є те, що при цьому має бути здійснений вирахований перехід, який веде до пропуску у потоці інструкцій процесора. Альтернативно перехід на функцію може бути здійснений із затримкою після кількох програмних інструкцій. Це створює проблеми оптимізації для компілятора, якщо на початку функції введено цикл і постійно має здійснюватися перехід туди і назад між кодом у попередньо заданій адресі і завантаженим кодом.

Тому задачею винаходу є розробка способу захисту адрес переходу вказаного вище типу, згідно з яким відпадає необхідність у здійсненні вирахованого переходу і, таким чином, не виникають пропуски у потоці інструкцій процесора.

Згідно з винаходом ця задача вирішена завдяки тому, що допустимі адреси переходів можуть бути використані безпосередньо і можуть бути розпізнані шляхом встановлення кореляції між даними, які не можуть перебувати всередині однієї і тієї ж інструкції.

При цьому компілятор або компоновщик шляхом організації програмного коду може забезпечити, що лише легальні адреси переходу виконують умови цієї кореляції. Наприклад, кореляція може бути встановлена завдяки тому, що копірка пам'яті безпосередньо до або після адреси переходу містить адресу корельованих даних.

Переважаюча можливість полягає в тому, що копірка пам'яті безпосередньо до або після адреси переходу містить посилання на відповідний запис у захищеному списку легальних адрес переходу.

Особливо доцільною є автоматична перевірка наявності кореляції між даними при виконанні виклику функції.

Наступною доцільною ознакою є додаткова автоматична перевірка розміщення корельованих даних у попередньо заданій, зарезервованій області пам'яті, при виконанні виклику функції.

Поки програмні інструкції не перевищують певного максимального числа n байт, може бути застосоване ще одне рішення згідно з винаходом, а саме використання спеціального коду пустої команди, який служить для уникнення випадкової кореляції і може бути додатково введений компілятором або компоновщиком.

При цьому особливо доцільним є встановлення кореляції між кодовими даними, віддаленими щонайменше на n байт.

Крім того, згідно з винаходом адреса переходу може бути захищена шляхом введення спеціальної послідовності байтів, яка не може виникнути всередині регулярного коду, наприклад, пустого коду.

Таким чином, згідно з винаходом можна уникнути пропуску у потоці інструкцій процесора шляхом безпосереднього переходу на адресу покажчика функції. Правда, в такому разі слід потурбуватися, щоб легальні адреси переходів відрізнялися нелокальною кореляцією кодів. Компілятор чи компоновщик мусить шляхом організації програмного коду забезпечити, щоб таку кореляцію мали лише легальні адреси переходів. При цьому "нелокальна кореляція" означає кореляцію між даними, розміщеними не в одній і тій же окремій інструкції.

Таким чином, можливі такі переважні форми виконання винаходу:

1. Кореляція з даними у зарезервованих для цього областях пам'яті: В разі простої реалізації в комірку пам'яті може міститися, наприклад, безпосередньо перед адресою переходу адреса корельованих даних, які, наприклад, відповідають легальній адресі переходу функції. При виконанні виклику функції може автоматично здійснюватися перевірка виконання умови цієї кореляції і/або перевірка розміщення корельованих даних у спеціально передбаченій, зарезервованій області пам'яті. Механізм на перший погляд дуже схожий на використовуваний досі механізм попередньо заданої адреси переходу, але його перевагою є те, що не використовується вирахований перехід і інструкції функції можуть утримуватися безпосередньо в модулі вибірки з упередженням (префетчері) потоку даних. Пропуск виникає лише у помилковому випадку нелегального переходу.

2. Кореляція з програмними даними у незарезервованих областях пам'яті. Передумовою цього рішення є те, що програмні інструкції не перевищують певну максимальну кількість n байт. Тоді області даних більшої довжини у кодовому сегменті мають бути вилучені. Крім того, передумовою цього методу є використання спеціального пустого коду, який не використовується у нормальному коді і додатково вводиться компілятором/компоновщиком лише для уникнення випадкової кореляції. В межах цієї форми виконання винаходу можна вирізнити ще два різні типи рішень:

а) Встановлюють кореляцію між даними, віддаленими щонайменше на n байт. При цьому компілятор чи компоновщик має забезпечити уникнення можливої випадкової кореляції у коді шляхом введення проміжного пустого коду.

Можлива реалізація виглядає таким чином:

Безпосередньо перед адресою переходу функції стоїть значення, що є функцією наступних $n+m$ ($m \geq 0$) байт.

Якщо ця кореляція випадково виникне де-небудь у коді, то компілятор чи компоновщик має усунути цю випадкову кореляцію: Оскільки в ході $n+m$ байт згідно з передумовою закінчується щонайменше одна істинна інструкція, після їх закінчення може бути введена серія пустих кодів, доки значення функції не зміниться. При цьому функція може бути довільно вибрана в певних межах.

б) Адреса переходу захищається шляхом введення спеціальної послідовності байт, яка не може виникнути всередині регулярного коду. Прикладом може бути серія пустих кодів.

Таким чином, згідно з винаходом адреси переходів захищаються шляхом нелокальної кореляції кодів, яка може виникнути лише в адресах переходів.

Завдяки цьому можна відмовитися від механізму із попередньо заданими адресами переходів, при якому використовують вирахований перехід, що спричинює пропуск у потоці інструкцій.

Згідно з винаходом за адресою переходу здійснюється безпосередній перехід у функцію. Наступні інструкції можуть бути завантажені в потік незалежно від позитивної чи негативної верифікації адреси переходу. Завдяки цьому, підвищується ефективність тестованих викликів функцій.

Формула винаходу

1. Спосіб захисту адрес переходів, який відрізняється тим, що допустима адреса переходу може бути розпізнана шляхом встановлення кореляції між даними, які не можуть перебувати в одній і тій же окремій інструкції, причому в комірці пам'яті безпосередньо перед або після адреси переходу розміщують адресу корельованих даних.

2. Спосіб за п. 1, який відрізняється тим, що в комірці пам'яті безпосередньо перед або після адреси переходу розміщують посилання на відповідний запис у захищеному списку легальних адрес переходу.

3. Спосіб за пп. 1 або 2, який відрізняється тим, що адресу переходу використовують безпосередньо.

4. Спосіб за одним із пунктів 1-3, який відрізняється тим, що при виконанні виклику функції здійснюють автоматичну перевірку виконання відповідною адресою переходу умови кореляції даних.

5. Спосіб захисту адрес переходів, який відрізняється тим, що допустима адреса переходу може бути розпізнана шляхом встановлення кореляції між даними, які не можуть перебувати в одній і тій же окремій інструкції, причому встановлюють кореляцію між даними в незарезервованих областях пам'яті.

6. Спосіб за п. 5, який відрізняється тим, що програмні інструкції не перевищують певну максимальну кількість n байтів, причому для уникнення випадкової кореляції використовують спеціальний код пустої команди, який може бути додатково введений компілятором чи компоновщиком.

7. Спосіб за пп. 5 або 6, який відрізняється тим, що встановлюють кореляцію між кодовими даними, віддаленими щонайменше на n байтів.

8. Спосіб за одним із пп. 5-7, який відрізняється тим, що адресу переходу захищають введенням спеціальної послідовності байтів, яка не може виникати у регулярному коді.

9. Спосіб за п. 8, який відрізняється тим, що як спеціальну послідовність байтів використовують спеціальний код пустої команди.

10. Спосіб за одним із пп. 5-9, який відрізняється тим, що адресу переходу використовують безпосередньо.

Офіційний бюлетень "Промислова власність". Книга 1 "Винаходи, корисні моделі, топографії інтегральних мікросхем", 2003, N 7, 15.07.2003. Державний департамент інтелектуальної власності Міністерства освіти і науки України.