

Тематичний напрям № 8
Кібербезпека, хмарні технології, Інтернет промов (IoT) та кіберфізичні системи

УДК 004.42

ОСНОВНІ МЕТОДИ ЗАСТОСУВАННЯ РЕГІСТРІВ НАЛАГОДЖЕННЯ ДЛЯ ЗАХИСТУ ВІД ДОСЛІДЖЕННЯ

Олександр Рисований, Олена Литовська
НТУ «Харківський політехнічний інститут», м. Харків, Україна

Захист програмного забезпечення від реверс-інжинірингу та аналізу є критично важливим аспектом інформаційної безпеки. Для протидії налагоджувачам розробники використовують як стандартні механізми операційних систем (наприклад, перевірку прапорів у блоці оточення процесу – PEB [1]), так і низькорівневі апаратні можливості процесорів архітектури x86/x64.

Особливе місце серед апаратних методів займають регістри налагодження (Debug Registers: DR0-DR7 [2]). Вони підтримуються лише на рівні процесора і активно застосовуються для виявлення і запобігання спроб дослідження коду.

Процесори архітектури x86/x64 містять набір спеціалізованих регістрів налагодження, які керуються привілейованими інструкціями або через контекст потоку (CONTEXT):

– DR0 – DR3 (адресні регістри точок зупинення). Вони містять лінійні адреси пам'яті, куди встановлені апаратні точки зупинення (hardware breakpoints).

– DR4 та DR5. Ці регістри зарезервовані архітектурою (у сучасних процесорах їх використання викликає виняток, а то й відключено спеціальним бітом у регістрі управління).

– DR6 (реєстр стану). Вміст регістру вказує, який саме регістр DR0–DR3 викликав спрацювання точки зупинення, а також містить службові прапорці (наприклад, слідування по кроках – Trap Flag).

– DR7 (реєстр управління). Вміст регістру визначає режим роботи кожної точки зупинення (тип доступу: читання, запис, виконання), а також активує локальні та глобальні дозволи.

В роботі розглянуто основні методи захисту з використанням регістрів налагодження та створено програми в середовищі masm64.

Метод прямого читання контексту потоку (GetThreadContext).

Дослідники часто використовують апаратні точки зупинення (DR0–DR3), щоб зупиняти виконання програми на потрібних інструкціях без модифікації самого коду (на відміну програмних точок зупинки INT 3 / 0xCC).

Захисний механізм може періодично або під час запуску перевіряти стан регістрів налагодження поточного потоку за допомогою системного виклику функції GetThreadContext (або аналогічних нативних API): якщо хоча б один із регістрів DR0–DR3 містить ненульове значення, це свідчить про те, що зовнішній налагоджувач встановив апаратний брейкпоінт.

Основний блок програми визначення наявності налагодження (точок зупинки) може бути таким:

```
invoke GetCurrentThread ;
   mov hThread, rax
mov ctx.ContextFlags, CONTEXT_DEBUG_REGISTERS or CONTEXT_FULL
invoke GetThreadContext, hThread, addr ctx;
   test rax, rax
   jz _error
```

```
; hardware breakpoints
   mov rax, ctx._Dr0
   test rax, rax
   jnz _debugger
   mov rax, ctx._Dr1
   test rax, rax
   jnz _debugger
   mov rax, ctx._Dr2
   test rax, rax
   jnz _debugger
   mov rax, ctx._Dr3
   test rax, rax
   jnz _debugger
```

Метод обробки винятків апаратних точок зупинки (Exception Handling).

Зловмисники або дослідники можуть намагатися використовувати апаратні брейкпоїнти для аналізу логіки роботи захищеної програми. Захист може перехоплювати винятки (EXCEPTION_SINGLE_STEP або STATUS_BREAKPOINT), використовуючи структуру структурної обробки винятків (SEH) або векторні обробники (AddVectoredExceptionHandler).

Якщо програма несподівано фіксує спрацювання налагоджувального переривання в критичній ділянці коду, де розробник не планував використовувати точки зупинки, це сприймається як спроба аналізу, після чого виконання аварійно завершується або змінює свою поведінку (шлях хибної логіки).

Метод використання регістрів DR6 та DR7 для верифікації цілісності.

Регістр керування DR7 та регістр стану DR6 містять прапори стану, які можуть аналізуватись захисним кодом «на льоту». Крім того, деякі алгоритми захисту навмисно встановлюють пастки в налагоджувальні регістри власного коду, щоб відстежувати спроби несанкціонованого доступу до захищених ділянок пам'яті.

Як і більшість методів протидії налагодженню, перевірка регістрів налагодження може бути нейтралізована дослідниками:

- патчінг перевірок (Patching), що позначає заміну умовних переходів (jnz, jz) після виклику GetThreadContext на протилежні або заміна інструкцій на NOP;

- перехоплення API (API Hooking). Перехоплення функції GetThreadContext за допомогою бібліотек таких як ScyllaHide для того, щоб підробляти значення, і завжди повертати нульові значення регістрів DR0–DR3;

- використання програмних точок зупинки, що позначає перехід на програмні брейкпоїнти (0xCC), якщо апаратні відстежуються захистом (хоча це у свою чергу відкриває вектор для виявлення INT 3 через сканування пам'яті).

Висновки. Застосування регістрів налагодження (DR0-DR7) є ефективним рівнем захисту від динамічного аналізу, оскільки воно спирається безпосередньо на апаратні можливості процесора. Найбільш надійні захисні комплекси використовують комбінацію перевірки вмісту регістрів через функцію GetThreadContext, аналіз структур оточення потоку та моніторинг винятків одиночного кроку, що значно ускладнює завдання дослідника щодо зворотного проектування програми.

Список використаних джерел

1. Рисований О.М. Реверсне програмування. Формат файлу PE. Середовище програмування masm64: навчально-методичний посібник для студентів спеціальності 123 "Комп'ютерна інженерія" всіх форм навчання [електронне видання]. Харків: НТУ «ХПІ», 2025. 124 с. https://repository.kpi.kharkov.ua/handle/KhPI-Press/95746

2. Рисований О.М. Методичні вказівки щодо виконання практичних та лабораторних робіт з курсу «Реверсне програмування. Антиналагоджувальні прийоми захисту від реверсу. Середовище програмування masm64: для студентів спеціальності 123 - "Комп'ютерна інженерія" всіх форм

навчання [електронне видання]. Х.: НТУ "ХПІ", 2024. Харків. 132 с. https://repository.kpi.kharkov.ua/handle/KhPI-Press/75508 .

УДК 004.056:339.543

ОБ'ЄКТИ КОМПЛЕКСНОЇ СИСТЕМИ ЗАХИСТУ ІНФОРМАЦІЇ МИТНОГО ПОСТУ ТА СТУПЕНІ ЇЇ УШКОДЖЕННЯ

Віра Сокольцова

Дніпровський національний університет ім. Олеся Гончара, м. Дніпро, Україна

Науковий керівник: В. Ю. Клим,
канд.тех.наук, доцент
Дніпровський національний університет ім. Олеся Гончара

Після руйнівного впливу митний пост може повернутися до роботи лише тоді, коли відновлено його комплексну систему захисту інформації (КСЗІ). Будівлю й обладнання можна замінити, проте обробляти митні декларації, платежі та персональні дані без працездатної і перевіреної КСЗІ не можна. Тому відновлення митного посту доцільно розглядати насамперед як відновлення його КСЗІ.

Мета дослідження полягає в визначенні об’єктів КСЗІ та систематизації ступенів ушкодження для подальшого обґрунтування алгоритму відновлення КСЗІ митного посту після фізичного втручання (руйнівного впливу). Об’єктом дослідження є комплексна система захисту інформації митного посту. Предметом дослідження є засоби відновлення КСЗІ після фізичного руйнівного впливу.

Для досягнення поставленої мети насамперед було проаналізовано нормативно-правову базу України. Окремого акту, який регламентував би відновлення КСЗІ після фізичного руйнування, в Україні немає, тому дослідження спирається на кілька груп джерел. Базовим є Закон України «Про захист інформації в інформаційно-комунікаційних системах», який відносить до об’єктів захисту як інформацію, так і програмне забезпечення для її обробки, покладає відповідальність за захист на власника системи та вимагає обробляти державні інформаційні ресурси в системах із КСЗІ [1]. Митні пости працюють у Єдиній автоматизованій інформаційній системі Держмитслужби (ЄАІС) [2], а отже, підпадають під ці вимоги. Закон України «Про критичну інфраструктуру» доповнює їх вимогою забезпечувати стійкість об’єктів, порушення роботи яких впливає на безпеку держави [3]. З цих норм випливає ключова вимога: повернення посту до обробки інформації допустиме лише після відновлення і перевірки КСЗІ, а не одразу після ремонту обладнання.

Національні норми доповнюють міжнародні стандарти та настанови. Найбільше до теми дослідження стосуються контролі ISO/IEC 27001:2022 A.5.29 (безпека під час критичної події) [4] і A.7.1 (фізичні межі безпеки) [5]. Стандарти ISO 22301 і ISO 22313 вводять показники відновлення MTPD, MBCO, RTO та RPO [6], керівництва NIST SP 800-34 [7] і SP 800-61 Rev. 3 [8] описують планування відновлення інформаційних систем і реагування на інциденти, а Настанови Всесвітньої митної організації (ВМО) пропонують митним адміністраціям цикл «готовність – реагування – відновлення» [9]. Крім того, оскільки фізичний рівень КСЗІ залежить від стану споруд, враховано Порядок № 473 щодо невідкладних робіт з ліквідації наслідків пошкодження будівель [10] та порядок їх обстеження [11]: доки за результатами обстеження не прийнято рішення, повертати обладнання в пошкоджену серверну не можна.

Визначивши нормативні вимоги, необхідно встановити, що саме підлягає відновленню, тобто саме об’єкти КСЗІ. Оскільки фізичний вплив зачіпає різні об’єкти КСЗІ по-різному, тому основні інформаційні ресурси митного посту згруповано за структурними підрозділами, які з ними працюють безпосередньо. Для кожної групи визначено критичні властивості інформації (конфіденційність, цілісність, доступність, автентичність, які позначено відповідними буквами К, Ц, Д, А) та загрози, що виникають після фізичного втручання.