УДК 004.056.55

В. А. Каплун

КЛАСИФІКАЦІЯ ВАД ЗАХИСТУ ЧЕРЕЗ ПОМИЛКИ У ПРОГРАМНОМУ ЗАБЕЗПЕЧЕННІ

Вінницький національний технічний університет

Анотація: *У статті йдеться про вразливості взагалі системи захисту програмного забезпечення, а саме про вразливості систем захисту, що можуть бути спричинені наявним у ній програмним кодом і дають змогу обійти впроваджені програмні засоби захисту. Розглянуто вади захисту, які є ненавмисними і такими, що вносяться у системи захисту під час його розроблення.*

Ключові слова: *вразливості систем захисту, помилки у програмному коді, переповнення буфера.*

Abstract*: In this article presented the vulnerability of software security systems in general, namely the vulnerability of security systems, which may be caused by the software code available to it and allow you to bypass the implemented software tools. Defects that are unintentional and those that are introduced into the security system during its development are considered.*

Keywords*: security vulnerabilities, program code errors, buffer overflow.*

Вступ

Необхідною умовою створення захищених систем захисту програмного забезпечення є проведення аналізу здійснених порушень безпеки з метою їх узагальнення і класифікації, з метою виявлення причин і закономірностей появи та існування уразливостей. Це дасть змогу в подальшому, під час розроблення систем захисту, спрямовувати зусилля на усунення першопричин появи уразливостей, що допоможе ефективніше протидіяти загрозам.

Вади захисту можуть бути внесені в систему навмисно чи випадково [1]. До вад захисту, що вносять у систему навмисно, належать програмні закладки або спеціальне програмне забезпечення, здатне послабити засоби захисту. Вадами захисту, що вносяться у систему ненавмисно, можуть бути помилки, яких припускаються розробники програмного забезпечення під час його проектування, реалізації, впровадження або супроводження.

Постановка задачі

Системи захисту інформації, що відповідають за цілісність і конфіденційність інформації, яка зберігаються на персональних комп'ютерах та інших пристроях або передається через мережі, функціонують під керуванням операційних систем, встановлених на цих пристроях. Отже, першим джерелом помилок, що призводять до вад у системах захисту, може бути сама операційна система.

Операційна система для своєї роботи використовує системне програмне забезпечення (драйвери, утиліти копіювання, пошуку файлів і послідовностей, програмні оболонки тощо). А тому наступним джерелом помилок може бути саме системне програмне забезпечення.

Прикладні програми, які зазвичай слугують інтерфейсом між користувачем та інформацією, що зберігається на пристроях і обробляється певним чином, а у частинному випадку реалізують певну систему захисту, також можуть містити у собі різного роду помилки, що призводять до появи уразливостей.

Отже, місцем виникнення вад систем захисту можуть бути [2]:

- операційні системи;
- системне програмне забезпечення;
- прикладне програмне забезпечення.

Операційна система, системне та прикладне програмне забезпечення – все це програми різного рівня і спрямування, а тому помилки можуть виникнути на будь-якому етапі розробки цих програм, починаючи з етапу формування технічного задання і закінчуючи супроводженням та експлуатацією.

Причини уразливостей систем захисту на різних етапах розроблення програм

Етап розроблення технічних вимог. Першим етапом створення програмного засобу є розроблення технічних вимог і специфікацій. На цьому етапі є дві проблеми, які можуть спричинити появу вад захисту у програмному продукті:

- протиріччя між вимогами безпеки і загальними вимогами до функціональності системи. Тут можуть бути прийняті певні компромісні рішення, що сприяють послабленню безпеки (наприклад, спрощення процедур ідентифікації й автентифікації, розширення прав користувачів, збільшення квот на використання пам'яті, процесорного часу, дискового простору тощо);
- невідповідність реальної системи технічним вимогам і середовища, в якому система функціонуватиме. Наприклад, програма експлуатується на пристрої з непередбаченою операційною системою, або здійснюється підключення системи до глобальної мережі в обхід запроектованих правил. Тоді всі обрані рішення із захисту втратять свою ефективність.

Етап розроблення алгоритмів. Алгоритми, що реалізують функції безпеки, також є потенційним джерелом вад захисту. Оскільки алгоритми підлягають перевірці, внесення навмисних помилок на цьому етапі малоймовірне, але ненавмисні помилки, наприклад, в алгоритмах роботи системи розмежування доступу, цілком імовірні [1].

Етап кодування. На цьому етапі припускаються найбільше помилок. Майже неможливо виявити ці помилки тестуванням скомпільованої програми, навіть якщо застосовувати для цього технологію «зворотної інженерії», тобто декомпіляцію. Причинами таких помилок є:

- складність програмного коду;
- участь у роботі над проектом великої кількості виконавців;
- використання фрагментів уже готового коду (бібліотек). Наприклад, найтиповіші помилки переповнення буфера, які надають зловмисникам можливість виконувати довільні команди на комп'ютері, як правило, виникають або внаслідок використання бібліотечних функцій з такою вразливістю, або через те, що ці функції викликаються іншими бібліотечними функціями;
- ненавмисне внесення недокументованих можливостей (з метою спрощення процедур тестування і налагодження програми, а інколи і задля того, щоб у подальшому можна було скористатися ресурсами системи. Іноді вносяться нешкідливі програмні закладки, які за виконання певних умов демонструють, наприклад, інформацію про розробників;
- навмисні програмні закладки, які здійснюють шпигунську місію, приховано надсилаючи з системи конфіденційну інформацію, або виконують руйнівні дії. Це цілком імовірне у програмах, отриманих із сумнівних джерел.

Етап компіляції. За допомогою сучасних компіляторів можна робити численні налаштування, які мають відповідати розробленим специфікаціям і вихідним текстам програм: обирати модель використання пам'яті, змінювати розмір сегментів, формат змінних за замовчуванням і встановлювати додаткові перевірки параметрів під час виклику процедур, компіляції та у разі відключення додаткових модулів. Крім того, компілятори можуть містити недокументовані функції.

Етап впровадження. На цьому етапі виконується налагодження системи захисту в конкретному програмному та апаратному середовищі. Тут вади захисту можуть бути впроваджені через помилки в адмініструванні системи: відсутність повної документації на систему з вичерпною інформацією про параметри, що можуть змінюватися під час інсталяції системи, недостатній досвід персоналу в адмініструванні конкретної системи.

Супроводження. Тут можуть бути внесені випадкові помилки, які спричиняють вади захисту: через недостатню обізнаність програмістів, що вносять зміни в систему, оскільки внесення будь-яких змін потенційно загрожує безпеці системи. Способом захисту від цієї загрози є всебічне тестування системи після здійснення будь-яких її модифікацій (щоразу як нової системи).

Етап експлуатації. Тут можливі декілька джерел виникнення вад захисту:

- помилки в адмініструванні системи, хоча добре спроектована система захисту зазвичай відстежує такі помилки, як відключення окремих захисних функцій;
- звуження кола контрольованих об'єктів;
- надання підвищених привілеїв користувачам чи процесам. Адміністратори часто не зважають на попереджання системи і діють на свій розсуд, ігноруючи вимоги безпеки задля спрощення процедур адміністрування.

Помилки, що виникають у процесі програмної реалізації системи захисту

Існує ряд типових помилок, які призводять до появи вад, розглянутих вище. Переважно ці помилки з'являються у кінцевому продукті через ненавмисні (випадкові) дії, хоча на певному етапі розроблення системи програмісти можуть додати (або, навпаки, вилучити) деякі функції навмисно задля спрощення процедур налагодження і тестування [3].

1) Помилки контролю припустимих значень параметрів – з'являються, коли певний механізм приймає хибне рішення щодо відповідності деякого параметра припустимим значенням. Це може також стосуватися кількості параметрів» їх типу, розміру тощо.

2) Помилки визначення областей – виникають за наявності неконтрольованого доступу в захищені домени (області пам'яті, файли на диску тощо). Наприклад, коли після видалення об'єкта з пам'яті буде скасовано контроль доступу до області, яку він займав, але не видалено інформацію, що містилася в об'єкті.

3) Помилки послідовності дій – виникають внаслідок асинхронного характеру функціонування комп'ютерних систем. Не завжди можна організувати перевірку і виконання дій, що відповідають результату цієї перевірки, як неподільну операцію. Частіше перевірку виконує одна функція, а результат передається іншій. Помилка виникає, коли підміняють результат перевірки або ідентифікатор об'єкта, для якого було виконано перевірку. Наприклад, перевіряють права доступу до одного файлу, а здійснюють доступ до іншого.

4) Помилки ідентифікації й автентифікації – виникають, коли підміняють автентифікаційну інформацію або виконують дії (створюючи для цього певні умови без необхідної автентифікації суб'єкта/об'єкта.

5) Помилки перевірки границь об'єктів – у виникають через неконтрольованість виходу об'єкта за межі області пам'яті, виділеної для його зберігання. Це можуть бути текстові рядки, масиви, файли тощо. Саме до помилок цього типу належать найпоширеніші помилки переповнення буфера.

Таким чином, помилки, що є причинами уразливостей систем захисту, можуть виникати на будь-якому етапі розробки програмного забезпечення. Єдиним способом, який може запобігти таким помилкам, є всебічне тестування, - структурне тестування і тестування функціональне, тестування як самими розробниками (альфа-тестування), так і зовнішнє (бета-тестування).

Список використаних джерел

1. Статистика українського інтернету [Електронний ресурс] // ukralio. – 2017. – Режим доступу до ресурсу: https://www.ukralio.com/statistikaukrainskogo-internetu
2. How security flaws work: The buffer overflow [Електронний ресурс] // arstechnica. – 2015. – Режим доступу до ресурсу: https://arstechnica.com/security/2015/08/how-security-flaws-work-the-bufferoverflow/
3. Говорущенко Т. О. Класифікація відмов та вразливостей системного програмного забезпечення / Т. О. Говорущенко, А. В. Мевша, В. А. Криськов // Інтелектуальні технології в системному програмуванні : III Всеукр. наук.-практ. конф. молодих учених та студ. : зб. наук. пр. – Хмельницький : Гонта А.С., 2018. – С. 318-328.

Каплун Валентина Аполінаріївна, ст. викл. кафедри захисту інформації, Вінницький національний технічний університет, Вінниця, valuka8379@gmail.com.

Valentyna A. Kaplun – Lecturer of the Chair of Safety of Infomation and Communication Systems, Vinnytsia National Technical University, Vinnytsia, valuka8379@gmail.com.