УДК 004.056:323.1

к.т.н. Хусаінов П. В. (ВІТІ)
Руднєв В. М.
Баканов В. С. (ВІТІ)

АНАЛІЗ УМОВ ОРГАНІЗАЦІЇ ЕКСПЕРИМЕНТАЛЬНОГО ОЦІНЮВАННЯ ЗАХИЩЕНОСТІ ПРИКЛАДНОГО ОБЧИСЛЮВАЛЬНОГО ПРОЦЕСУ

Необхідність забезпечення ефективної діяльності суб'єктів Національної системи кібербезпеки обумовлює актуальність розроблення науково-методичного апарату оперативного реагування на кіберінциденти (кібератаки). Принципова неможливість досягнення алгоритмічної та інформаційної повноти технічних засобів кіберзахисту передбачає впровадження процесу підтримки прийняття відповідних рішень оперативним персоналом органів кібербезпеки. Інший фактор невизначеності рішень полягає у відсутності апріорних даних для ідентифікації величини шкоди від наслідків кіберінциденту. Останнє обумовлене тим, що опис кіберінциденту складається з сукупності ознак виявлення можливої (потенційної) кібератаки, але величина шкоди в цей момент не може бути достовірно відомою. Визначення величини шкоди на етапі виявлення кібератаки може бути здійснена з використанням моделей доведення безпеки теорії захисту інформації, які засновані на суб'єктно-об'єктному представленні об'єкта кіберзахисту. Використання згаданих моделей вимагає знання ймовірності захищеності від нав'язування непередбаченого виконання для прикладних обчислювальних процесів всіх типів у складі об'єкта кіберзахисту.

Оцінювання захищеності прикладного обчислювального процесу пропонується здійснювати у формі натурного випробування. Експеримент дозволяє одержати найбільш повне уявлення про можливості та особливості використання типових вразливостей програмної реалізації цільового обчислювального процесу, а його організація полягає у вставлені факту виконання активного контенту зі складу спеціальних кодових комбінацій вхідних даних. Результат експерименту: опис кодової комбінації вхідних даних (підмножини можливих комбінацій), оброблення якої, призвело до виконання активного контенту; середній сумарний час активного експериментування з цільовим обчислювальним процесом до моменту передачі управління активному контенту.

Стаття присвячена викладенню результатів аналізу умов, які необхідно врахувати при організації експериментального оцінювання захищеності прикладного обчислювального процесу.

Ключові слова: *кібератака, вразливість, випробування.*

Хусаинов П.В., Руднев В. Н., Баканов В.С. Анализ условий организации экспериментального оценивания защищенности прикладного вычислительного процесса.

Необходимость обеспечения эффективной деятельности субъектов Национальной системы кибербезопасности обуславливает актуальность разработки научно-методического аппарата оперативного реагирования на киберинциденты (кибератаки). Принципиальная невозможность достижения алгоритмической и информационной полноты технических средств киберзащиты предусматривает внедрения процессов поддержки принятия соответствующих решений оперативным персоналом органов кибербезопасности. Другой фактор неопределенности заключается в отсутствии априорных данных для идентификации величины ущерба вследствие киберинцидента. Последнее обусловлено тем, что описание киберинцидента состоит из совокупности признаков для выявления возможной (потенциальной) кибератаки, но величина ущерба в этот момент не может быть достоверно известна. Определение величины ущерба на этапе выявления кибератаки может быть осуществлена с использованием моделей доказательства безопасности в теории защиты информации, которые базируются на субъектно-объектном представлении объекта киберзащиты. Использование упомянутых моделей требует знания вероятности защищенности от непредусмотренного выполнения алгоритма для прикладных вычислительных процессов всех типов в составе объекта киберзащиты.

Оценивание защищенности прикладного вычислительного процесса предлагается осуществлять в форме натурного эксперимента. Он позволяет получить наиболее полное представление о возможностях и особенностях использования типичных уязвимостей программной реализации целевого вычислительного процесса, а его организация заключается в определении факта выполнения активного контента из состава специальных кодовых комбинаций входных данных. Результат эксперимента: описание кодовой комбинации входных данных (подмножества возможных комбинаций), обработка которой приводит к выполнению активного контента; среднее суммарное время активного экспериментирования с целевым вычислительным процессом до момента выполнения активного контента.

Статья посвящена изложению результатов анализа условий, которые необходимо учитывать при организации экспериментального оценивания защищенности прикладного вычислительного процесса.

Ключевые слова: *кибератака, уязвимость, натурный эксперимент*

Khusainov Pavlo, Rudnev Bolodumir, Bakanov Valentin Analysis of the conditions of organization of experimental evaluation security of applied computer process.

The need to ensure the effective operation of entities of the National Cyber Security Systemstipulatethe urgency of developing a scientific and methodological apparatus for rapid response to cyber incidents (cyberattacks).The

fundamental impossibility of achieving algorithmic and information completeness of cyber defense equipment anticipates the implementation of a process to support the decision-making of the operational staff of cybersecurity.Another factor of decision uncertainty is the lack of a priori data to identify the magnitude of the damage from the effects of the cyber incident.The latter is due to the fact that the description of a cyber incident consists of a set of signs of detection of a possible (potential) cyberattack, but the amount of damage cannot be reliably known instantly.Determining the amount of damage at the moment of detecting a cyberattack can be done using security proof models of information security theory, based on the subject-object representation of the object of cybersecurity.The use of these models requires knowledge of the probability of protection against the imposition of unforeseen execution for applied computing processes of all types in the object of cybersecurity.

The proposition is to evaluate the reliability from influence on the applied computational process in the form of an experiment.The experiment allows obtaining the most complete image of the possibilities and features of the use of typical vulnerabilities of the software implementation of the target computational process. The organization of the experiment is to establish the fact of the execution of an active codefrom the composition of special code combinations in input data.The result of the experiment: a description of the code combination of input data (subsets of possible combinations)the processing of which led to the execution of the active payload;the average time of the experiment with the target computing process before the transfer of control to the active payload. The article is devoted to the presentation of the results of the analysis of conditions that must be taken into account when organizing an experimental assessment of the possibility of influence on the applied computational process.

Keywords: *cyberattack, vulnerability, experiment*

Постановка завдання у загальному вигляді

Реагуючи на світові тенденції та виклики сьогодення з 2017 року у інформаційно-правовому просторі нашої держави з'явилася низка законодавчих положень, які визначають правові та організаційні основи забезпечення захисту життєво важливих інтересів людини і громадянина, суспільства та держави, національних інтересів України у кіберпросторі, основні цілі, напрями та принципи державної політики у сфері кібербезпеки, повноваження державних органів, підприємств, установ, організацій, осіб та громадян у цій сфері, основні засади координації їхньої діяльності із забезпечення кібербезпеки [1].

Так, згідно Закону України "Про основні засади забезпечення кібербезпеки України" кібербезпека досягається та забезпечується якісним виконанням сукупності заходів кіберзахисту (організаційних, правових, інженерно-технічних, криптографічного та технічного захисту інформації), які спрямовані на запобігання кіберінцидентам, виявлення та захист від кібератак, ліквідацію їх наслідків, відновлення сталості і надійності функціонування об'єктів кіберзахисту. Кіберінцидент – подія або ряд несприятливих подій ненавмисного характеру (природного, технічного, технологічного, помилкового, у тому числі внаслідок дії людського фактора) та/або таких, що мають ознаки можливої (потенційної) кібератаки, які становлять загрозу безпеці систем електронних комунікацій, систем управління технологічними процесами, створюють імовірність порушення штатного режиму функціонування таких систем (у тому числі зриву та/або блокування роботи системи, та/або несанкціонованого управління її ресурсами), ставлять під загрозу безпеку (захищеність) електронних інформаційних ресурсів.

На сучасному етапі основний підхід до забезпечення безпеки інформації в автоматизованих системах полягає в організації розмежування доступу, який базується на відповідній концепції диспетчера доступу Повноваження на доступ обчислювального процесу до об'єктів інформаційного домену визначаються засобами операційної системи за результатами успішної авторизації користувача автоматизованої системи [2 – 4]. Найбільш суттєвим недоліком концепції диспетчера доступу є принципове припущення про сталість правильного функціонування прикладних обчислювальних процесів на всьому часовому інтервалі експлуатації автоматизованої системи тобто неможливість виконання ними непередбачених розробником дій у будь-який момент часу.

Аналіз останніх публікацій за темою дослідження

Непривілейований авторизований користувач автоматизованої системи об'єкта кіберзахисту немає можливості розширити (змінити) свої повноваження, але у будь-який момент часу він може ініціювати цілеспрямовану зловмисну діяльність інсайдера (від англ. *insider* – внутрішній порушник). При цьому, на теперішній час, відомо чимало прикладів

організації такої інформаційної взаємодії між прикладними обчислювальними процесами при посередництві системних об'єктів операційної системи (у межах відповідних їм інформаційних доменів (доступу), що перетинаються), яка може призвести до нав'язування непередбаченого виконання алгоритму виконання [5 – 9].

Отже можна констатувати, що розроблення науково-методичного оперативного реагування на кіберінциденти (кібератаки) тісно взаємопов'язаний з обґрунтуванням визначення ймовірності захищеності прикладних обчислювальних процесів у складі об'єкта кіберзахисту від нав'язування непередбаченого виконання на основі даних відповідних випробувань. На підставі викладеного, *метою статті є аналіз умов для організації експериментального оцінювання захищеності прикладного обчислювального процесу від нав'язування непередбаченого розробником виконання на основі модифікації його алгоритму.*

Виклад основного матеріалу

Модифікація алгоритму роботи цільового обчислювального процесу шляхом руйнівного впливу на зміст її оперативної пам'яті досягається за рахунок нав'язування непередбачених значень тим чи іншим критичним даним. Критичними даними цільового обчислювального процесу будемо називати будь-які розташовані в його оперативній пам'яті дані, нав'язування непередбачених значень яким дозволяє інсайдеру вигідним для себе чином модифікувати (спотворити) алгоритм роботи. В цьому смислі можна говорити, що критичні дані визначають алгоритм. Серед критичних даних цільової програми будемо розрізняти управляючі та інші критичні дані.

Управляючими даними цільового обчислювального процесу будемо називати дані, що прямо або опосередковано визначають потік передачі управління, точніше, значення, яке на певному етапі виконання цільової програми потрапить у лічильних команд процесора. До таких даних можна віднести, наприклад, адреси повернення з функцій та покажчики на них (збережені значення покажчика стекового кадру), покажчики на функції та покажчики на покажчики. Нав'язування непередбачених значень управляючим даним призводить до непередбаченої модифікації (спотворення) потоку передачі управління у цільовому обчислювальному процесі. В результаті управління може (непередбачено) передаватись введеному в адресний простір цільового процесу сторонньому коду або, наприклад, деякому привілейованому фрагменту коду або функції, що дозволяє запускати довільні зовнішні програми. Завдяки цьому може (непередбачено) запускатись як одна із штатних програм, так і попередньо введена в цільову систему стороння програма (в залежності від можливостей та інтересів інсайдера). Нав'язування непередбачених значень управляючим даним дозволяє, в принципі, нав'язати їй довільний напрямок (адресу) передачі управління.

До *інших критичних даних цільового обчислювального процесу* будемо відносити будь-які її критичні дані, що не впливають на потік передачі управління в ній (або, якщо впливають, то не настільки, щоб шляхом їх непередбаченої модифікації можна було б нав'язати довільний напрямок передачі управління, як у випадку критичних даних, що виступають операндами умовних конструкцій). Серед таких критичних даних можна виділити наступні важливі групи:

команди, що передаються цільовою програмою її оточенню;

дані авторизації суб'єктів доступу до ресурсів;

перемикачі режиму роботи захисних механізмів;

ідентифікатори ресурсів, що використовуються;

ідентифікатори ресурсів до яких забезпечується доступ;

дані, що виводяться;

покажчики на перелічені дані та індекси в їх переліках.

Для того, щоб модифікувати алгоритм роботи цільового обчислювального процесу, треба непередбачено модифікувати ті чи інші з її критичних даних, причому, протягом проміжків часу, які обмежуються, з одного боку, моментом ініціалізації або модифікації цих критичних даних, з другого боку – моментом їх використання.

Критичні дані також варто поділити на такі, що прямо (безпосередньо) і непрямо (опосередковано) визначають алгоритм роботи цільового обчислювального процесу. Будемо вважати, що критичні дані прямо (безпосередньо) визначають алгоритм роботи, коли вони не є покажчиками на інші критичні дані; такі критичні дані будемо називати також кінцевими критичними даними. Відповідно, критичні дані, що є покажчиками на інші критичні дані, будемо вважати такими, що непрямо (опосередковано) визначають алгоритм роботи цільового обчислювального процесу.

Модифікація алгоритму роботи цільового обчислювального процесу (шляхом руйнівного впливу на зміст її оперативної пам'яті) завжди здійснюється на основі фальсифікації тих чи інших критичних даних, які прямо (безпосередньо) визначають алгоритм її роботи, тобто на основі фальсифікації тих чи інших кінцевих критичних даних цієї програми. Фальсифікація критичних даних може виконуватись прямо (безпосередньо) і непрямо (опосередковано). Пряма (безпосередня) фальсифікація критичних даних полягає в нав'язуванні цим даним непередбачених значень, наприклад, шляхом їх ініціалізації такими значеннями або їх непередбаченої модифікації.

Для кібератак, які здійснюються методом нав'язування коду (injection-based attacks) – передбачають нав'язування цільовому обчислювальному процесу певного активного контенту з подальшою його активізацією у відповідному інформаційному домені доступу – передбачено застосування наступних понять (важливі для розуміння механізму реалізації):

вектор атаки (injection vector);

активний контент ("корисне навантаження"; payload);

зона активізації контенту (activation zone);

типовий результат активізації контенту (payload activation impact).

Вектор атаки – це механізм введення активного контенту в середовище виконання цільового обчислювального процесу (включно з форматом повідомлень, якими доставляється активний контент, та протоколом взаємодії, у рамках якого здійснюється доставка йому активного контенту).

Активний контент – це ті придатні до активізації дані, уведення яких в середовище виконання цільового обчислювального процесу і подальша активізація в цьому середовищі забезпечує успіх кібератаки.

Зона активізації контенту – це той елемент або етап виконання цільового програмного забезпечення, який (на якому) активізує(ться) введений контент.

Умови реалізації кібератак даного класу:

програмна реалізація завантажувального модуля цільового обчислювального процесу має вразливість до руйнівного впливу на зміст її оперативної пам'яті;

інсайдер (суб'єкт атаки) має можливість спровокувати активізацію цієї вразливості, тобто передати цільовому обчислювальному процесу дані для активізації цієї вразливості.

Типові вразливості програмної реалізації завантажувального модуля цільового обчислювального процесу, що створюють можливість атак даного класу:

переповнення буфера (в стеку, в статичному або динамічному сегменті даних);

переповнення розрядної сітки цілочисельних змінних;

нав'язування форматних рядків.

Переповнення буфера має місце, коли існує можливість спровокувати перезапис комірок пам'яті, розташованих за межами буфера, в який повинен виконуватись запис даних на тому чи іншому етапі її виконання. У випадку програм з відкритою пам'яттю, послідовне переповнення буфера в простій формі дозволяє виконати непередбачену модифікацію тільки даних, розташованих безпосередньо над уразливим буфером (у тих чи інших межах), в одному з ним сегменті, в складній формі, за рахунок непередбаченої модифікації розташованих над уразливим буфером покажчиків, що визначають адреси, за якими в подальшому виконується запис або читання даних – також непередбачену модифікацію або непередбачене читання даних, розташованих у інших місцях адресного простору вразливого

процесу, в загальному випадку – непередбачену модифікацію або непередбачене читання довільної області адресного простору.

Переповнення розрядної сітки цілочисельних змінних має місце, коли існує можливість спровокувати присвоєння тим чи іншим її цілочисельним змінним значень, що виходять за межі допустимого для них згідно їх типу діапазону. Коли відбувається переповнення розрядної сітки цілочисельної змінної, як правило, просто відкидаються ті старші розряди двійкового представлення значення, що присвоюється цій змінній, які не поміщаються в область пам'яті, виділену для цієї змінної. У випадку беззнакових цілочисельних змінних при цьому старший розряд результуючого значення інтерпретується як значущий, у випадку знакових цілочисельних змінних – як знаковий. Внаслідок таких перетворень результуюче значення, як правило, буде значно відрізнятись від того, що присвоюється.

Форми (способи виконання) переповнення розрядної сітки цілочисельних змінних (вони ж – види руйнівного впливу на значення цілочисельних змінних):

переповнення розрядної сітки цілочисельної змінної зверху;

переповнення розрядної сітки цілочисельної змінної знизу.

Переповнення розрядної сітки цілочисельної змінної зверху має місце, коли цій змінній присвоюється значення, яке перевищує максимально допустиме для неї згідно її типу, а знизу – коли цій змінній присвоюється значення, яке є меншим від мінімально допустимого для неї згідно її типу. Переповнення розрядної сітки може мати міце не тільки у випадку явного присвоєння значень тим чи іншим цілочисельним змінним, але й при передачі (цілочисельних) аргументів функціям, а також під час (неявного) розміщення в пам'яті результатів функцій або арифметичних операцій. Переповнення розрядної сітки цілочисельних змінних саме по собі є руйнівним впливом на зміст оперативної пам'яті програми і може безпосередньо (без виконання додаткових руйнівних впливів) призводити до компрометації її безпеки.

Безпосередня компрометація безпеки програми внаслідок переповнення розрядної сітки цілочисельної змінної може полягати у зациклюванні та/або спотворенні логіки алгоритму. Окрім того, воно може створювати умови для додаткових руйнівних впливів на зміст оперативної пам'яті програми.

Нав'язування форматних рядків має місце, коли існує можливість спровокувати використання в якості форматних наданих ззовні текстових рядків. У випадку програм із відкритою пам'яттю, нав'язування форматних рядків (функціям форматованого виведення) дозволяє непередбачено скористатись можливостями функцій форматованого виведення стосовно читання та модифікації розташованих в оперативній пам'яті вразливої програми даних для непередбаченого читання або непередбаченої модифікації комірок адресного простору цільового процесу.

Руйнівний вплив на зміст оперативної пам'яті цільової програми досягається нав'язуванням вразливій функції форматованого виведення непередбачених специфікаторів формату, тобто форматного рядка, який містить непередбачені специфікатори формату. Функції форматованого виведення мають наступні спільні риси:

приймають змінну кількість аргументів;

приймають один чи більше фіксованих (обов'язкових) аргументів, останнім серед яких є так званий форматний рядок, який і повідомляє їм повну кількість аргументів;

обробляють всі аргументи, що йдуть після форматного рядка, згідно зі специфікаторами формату, що містяться в форматному рядку (як правило, виводять ці аргументи або зміст буферів, на які вони вказують).

Форматний рядок представляє собою суміш зі звичайних символів і так званих специфікаторів формату, які виступають замінниками для наступних аргументів. Відповідно, кожному специфікатору формату відповідає один із наступних аргументів, і при правильному застосуванні форматних рядків кількість специфікаторів формату в них повинна дорівнювати кількості переданих разом із ними функції форматованого виведення додаткових аргументів.

Загальні причини вразливості цільового обчислювального процесу до руйнівного впливу на зміст оперативної пам'яті:

відсутність в них або недостатність процедур забезпечення коректності вхідних даних;,

некоректна реалізація процедур оброблення вхідних даних.

Часто вразливості даного класу є наслідком покладання їх розробників на (невірні) припущення щодо того, що певні вхідні дані не можуть бути некоректними. Типові вектори атак даного класу тобто способи введення в оперативну пам'ять цільового обчислювального процесу руйнівних даних (узагальнено):

(через) файли з даними, призначеними для оброблення;

(через) аргументи командного рядка;

(через) змінні оточення;

(через) конфігураційні файли;

(через) файли з додатковими даними;

(через) відповіді на запити на введення додаткових даних;

(через) повідомлення, що надходять через канали комункації.

Контрольований інсайдером обчислювальний процес може бути розташований як локально (в одному обчислювальному середовищі з цільовим обчислювальним процесом), так і віддалено по відношенню до нього (в одному обчислювальному середовищі іншого елементу об'єкта кіберзахисту).

Висновки

Основним елементом організації експериментального оцінювання захищеності цільового обчислювального процесу від нав'язування непередбаченого виконання алгоритму є вибір вектора атаки. В якості останнього пропонується використання мережевого інтерфейсу, а конкретніше, певне поле повідомлення, що на певному етапі, згідно комунікаційного протоколу, за яким здійснюється віддалена взаємодія, надається цільовому обчислювальному процесу. Активним контентом буде включений у це повідомлення і призначений для нав'язування байт-код відкриття сеансу роботи через мережу з інтерфейсом командного рядка, а зоною активізації контенту – момент виходу потоку виконання цільового обчислювального процесу з уразливої функції (команда повернення з цієї функції).

Напрямком подальших досліджень є розробка методики експериментального оцінювання середнього часу нав'язування цільовому обчислювальному процесу виконання байт-коду відкриття сеансу роботи через мережу з інтерфейсом командного рядка.

ЛІТЕРАТУРА

1. Про основні засади забезпечення кібербезпеки України: Закон України від 5 жовтня 2010 року № 2163-VIII // Відомості Верховної Ради України. – 2017 . – № 45. – Ст. 403.

2. Автоматизовані системи. Терміни та визначення: ДСТУ 2226-93. – [Чинний від 1994-07-01]. – 94 с.

3. Системи оброблення інформації. Взаємозв'язок відкритих систем. Базова еталонна модель: ДСТУ 2230-93. – [Чинний від 1994-07-01]. – 59 с.

4. Термінологія в галузі захисту інформації в комп'ютерних системах від несанкціонованого доступу: НД ТЗІ 1.1–003–99. – Київ: ДСТСЗІ СБ України, 1999. – 26 с.

5. Антонюк А.О. Теоретичні основи моделювання та аналізу систем захисту інформації: [монографія]. – Ірпінь: Національний університет ДПС України, 2010. – 310 с.

6. Хусаінов П. В., Субач І. Ю., Сілко О. В., Любарський С. В. Основи побудови операційних систем, комплексів та засобів автоматизації управління військами: Навчальний посібник. – К.: ВІТІ, 2016. – 220 с.

7. CommonVulnerability Enumeration // – Режим доступу: http://cve.mitre.org.

8. CommonWeakness Enumeration // – Режим доступу: http:// cwe.mitre.org

9. Common Attack Pattern Enumeration and Classification // – Режим доступу: http://capec.mitre.org.