МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
«ОДЕСЬКА ЮРИДИЧНА АКАДЕМІЯ»
ФАКУЛЬТЕТ КІБЕРБЕЗПЕКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КІБЕРБЕЗПЕКА В СУЧАСНОМУ СВІТІ: АКТУАЛЬНІ ВИКЛИКИ

МАТЕРІАЛИ
III Всеукраїнської науково-практичної конференції

19 листопада 2021 року

НУ «ОЮА»
2021 р.

УДК 004.056(063)
К38

Відповідальний редактор – Дикий Олег Вікторович, декан факультету кібербезпеки та інформаційних технологій, к. ю. н., доцент.

Укладачі:
Горбаченко Станіслав Анатолійович – завідувач кафедри кібербезпеки, д. е. н., доцент.
Логінова Наталія Іванівна – завідуюча кафедрою інформаційних технологій, к. пед. н, доцент;

Кібербезпека в сучасному світі : матеріали III Всеукраїнської науково-практичної конференції (м. Одеса, 19 листопада 2021 р.) / за ред. О. В. Дикого ; уклад.: С. А. Горбаченко, Н. І. Логінова. – Одеса, 2020. – 148 с. DOI 10.32837/11300.15973

У збірнику містяться матеріали доповідей, присвячених актуальним проблемам кібербезпеки в сучасному світі, що змінюється.
Подані матеріали відповідають сучасним тенденціям розвитку кібербезпеки в Україні та можуть бути корисними як у навчальному процесі, так і при проведенні наукових досліджень студентами та молодими вченими, а також для практичних працівників у зазначеній сфері.

УДК 004.056(063)

ПЕРЕДМОВА

Шановні учасники ІІІ Всеукраїнської науково-практичної конференції «Кібербезпека в сучасному світі: актуальні виклики»!

Радий вітати Вас в роботі чергового науково-технічної заходу, що проводиться Національним університетом «Одеська юридична академія». Головною метою проведення конференцій чи круглих столів в університеті завжди було об'єднання талановитих науковців, ідей і досвіду задля вирішення актуальних теоретичних та практичних проблем. До того ж проведення наукових конференцій надає можливість студентам, аспірантам та науковцям вільно висловлювати свої думки та ідеї щодо питань забезпечення кібербезпеки, запропонувати практичні напрацювання та ділитися результатами своїх досліджень, що є одим із фундаментальний принципів діяльності Університету.

Учасники конференції проявили значну зацікавленість у дослідженні технічних питань функціонування інформаційних систем, інституційно-правових механізмів забезпечення інформаційної безпеки та соціально-політичних умов формування кіберпростору. Особливу увагу було приділено тематиці формування культури інформаційної безпеки у сучасному суспільстві, яка зараз є пріоритетною у формуванні та реалізації державної політики України. Глибокий інтерес у майбутніх фахівців із кібербезпеки виник також до питань протидії кібербулінгу, фішингу та іншим негативним явищам в кіберпросторі, адже це є одним з пріоритетних напрямів роботи державних інституцій та однією з найбільших проблем використання кіберсередовища.

Загалом сфера інформаційної безпеки та інформаційних технологій сьогодні є найбільш перспективною. У зв'язку з цим з перших днів існування факультету кібербезпеки та інформаційних технологій керівництво прагнуло, насамперед, створити сучасний та студентоцентрований центр підготовки фахівців в сфері ІТ технологій, який був би конкурентоспроможним в Україні та за її межами. Для цього застосовуються сучасні технології в організації навчального процесу, використовується позитивний інноваційний досвід провідних зарубіжних закладів вищої освіти, модернізується матеріально-технічна база. Зараз вчені

факультету кібербезпеки та інформаційних технологій продовжують традиції одеської школи, успішно розвивають і примножують наукові досягнення своїх попередників. На факультеті кібербезпеки та інформаційних технологій створені найкращі умови для підготовки кваліфікованих фахівців з кібербезпеки та інформаційних технологій. Саме тому проведення таких наукових заходів вже стало традиційним. Багато робіт учасників конференції підготовлено іноземними мовами, що дозволяє інтегрувати надбання української науки до світового середовища.

Хочу щиро подякувати кожному учаснику всеукраїнської науково-практичної конференції «Кібербезпека в сучасному світі: актуальні виклики» за Ваш внесок до сучасної науки та побажати успіхів у подальших дослідженнях на науковій діяльності!

СЕРГІЙ КІВАЛОВ,
президент Національного університету
«Одеська юридична академія»,
доктор юридичних наук, професор,
академік НАПрН України,
заслужений юрист України

РОЗДІЛ 1.
ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ БЕЗПЕКИ ФУНКЦІОНУВАННЯ КОМП'ЮТЕРНИХ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Виходець Юрій Олександрович

Управління протидії кіберзлочинам в Одеській області Департаменту кіберполіції Національної поліції України, начальник, кандидат юридичних наук

ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ БЕЗПЕКИ В ІНФОРМАЦІЙНИХ СИСТЕМАХ ТА ЗАГРОЗИ СТАБІЛЬНОСТІ РОБОТИ КОМП'ЮТЕРНИХ МЕРЕЖ

Будь-кому відомо, що сучасне суспільство є основним генератором інформаційного середовища і повноцінним її споживачем. Базуючись на даному факті, можна з упевненістю говорити про те, що в міру розвитку такого явища, виникають внутрішні безперервні зв'язки і залежності між користувачами глобального, або окремо взятого, інформаційного простору. Починаючи з часів, коли невеликі групи людей стали утворювати племена, поселення, міста, імперії, і закінчуючи сьогоднішнім днем, з плином часу, всередині таких людських, або, якщо говорити про новітній період розвитку суспільства, всередині людино-машинних інфраструктур, почали виникати скупчення інформації, які з тих чи інших причин не розголошувались або не могли бути розголошеними - простіше кажучи, всередині системи обміну інформацією з'явилися секрети. Секрети, або секретну інформацію, людство намагається замаскувати і краще приховати від ворогів ще з часів єгипетських фараонів. Даний період закладає основу процесу побудови комплексу технічних засобів та прийомів, покликаних досягнути необхідного та достатнього рівня забезпечення безпеки в інформаційних мережах. Чим швидше розвивалося суспільство, тим більш витонченими ставали методи контрольованого спотворення інформації - її шифрування. До 19 століття, все шифрування було "ручним", тобто, повністю виконувалося рукою людини. З появою складних обчислювальних механізмів і роторних машин,

частина цього завдання лягла на механізми. Але, коли в життя людства вторглися потужні ЕОМ - комп'ютери, роль людини в цьому процесі звелася до мінімуму і полягала в розробці таких алгоритмів шифрування нового покоління, які давали б можливість так перетворити інформацію, використовуючи потужності обчислювальної техніки, що б ворог жодним чином не зміг її розсекретити. Складність методів і алгоритмів шифрування збільшується прямо пропорційно стрімкому розвитку самих ЕОМ. На сьогоднішній день розроблено безліч алгоритмів шифрування, які в достатній мірі здатні зберегти цілісність і конфіденційність інформації, з огляду на сучасні можливості обчислювальної комп'ютерної техніки. Найбільш популярними алгоритмами в сучасному інформаційному середовищі є симетричні алгоритми - DES, 3DES, AES, - і асиметричні алгоритми, такі як, наприклад, група шифрів RSA зі змінною довжиною ключа.

Концепція захисту даних за допомогою шифрування дуже важлива та має своє місце навіть сьогодні. Так, наприклад, у більшості месенджерів для передачі сигналів та знаків використовується шифрування типу end-to-end, а розробники одного с найбільш захищених месенджерів – Telegram, винайшли свій власний протокол шифрування – MTProto 2.0, який являється яскравим прикладом поєднання декількох алгоритмів шифрування в одне ціле для досягнення найвищого рівня захисту даних.

Шифрування розроблялось для забезпечення потреб в аналізі загроз інформаційної безпеки та дозволяє виділити складові сучасних комп'ютерних загроз - їх джерела і рушійні сили, способи і наслідки реалізації. Аналіз виключно важливий для отримання всієї необхідної інформації про інформаційні загрози, визначення потенційної величини збитку, як матеріальної, так і нематеріальної, і вироблення адекватних заходів протидії. Однією з найбільш розповсюджених загроз кібербезпеки є витік даних (близько 80% кібер-інцидентів).

Однією з найбільших загроз кібербезпеки є можливість однозначного збереження цілісності інформації при зовнішніх або внутрішніх перебоях різноманітного втручання. Для забезпечення захисту від такого типу кіберзагроз

використовують періодичне побітове копіювання інформації з закріпленням контрольних сум масивів даних. Якщо аналізувати ризики та загрози кібербезпеки у 21-му сторіччі, то можна сміливо говорити, що однією з найбільших проблем є проблема «мережевої неграмотності», яка, часто, полягає у не вмінні проектувальників-техніків закривати за собою порти, які, в подальшому, можуть слугувати точками входу для різноманітних програмних засобів або стати майданчиком для заливу «шеллів». «Шелли», в свою чергу, відвантажують до інформаційної системи вихідні коди вірусів-вимагачів - один з видів шкідливого ПЗ. Вони вимагають гроші, блокуючи доступ до файлів або комп'ютерних систем до сплати викупу. При цьому сплата викупу не гарантує відновлення доступу до файлів або системам. Тому, аби вберегтись від можливого зараження системи ШПЗ треба вміти вчасно та зі 100%ю вірогідністю розпізнавати шкідливе програмне забезпечення або сценарій.

Експерти стверджують, що останнім часом спостерігається помітне зростання числа фішингових атак. Ця широко поширена загроза кібербезпеки створює величезний ризик для окремих осіб і організацій. Фішинг атака - це методи, використовувані зловмисниками для крадіжки конфіденційної інформації, коли користувачів намагаються змусити її передати. Більшість користувачів про цю загрозу навіть не підозрюють. Фішинг, як спосіб шахрайства в тій чи іншій формі використовується вже протягом багатьох років. Починався він ще з телефонних дзвінків. Але сьогодні кіберзлочинці розглядають фішинг-атаки як успішний і простий спосіб підготовки більш складних кібер-атак.

З кожним днем користувачі мережі Інтернет створюють все більше інформаційних складових для побудови цілісного інформаційного простору. Однією з таких складових є створення та подальше адміністрування електронних поштових скриньок. Але, масиви таких скриньок зберігаються у величезних контейнерах-серверах, які, в свою чергу вразливі до зламу. В ході отримання хакерами доступу до баз даних велика кількість унікальних пар логін+пароль розміщуються на тіньових майданчиках з метою продажу. Дана загроза кібербезпеки відноситься до типу загроз «несанкціонованого розподілення та

розповсюдження інформації з обмеженим доступом» та виникає у випадку передачі сервісами авторизації або аутентифікації таких даних компаніям, які будують маркетинг на розсилці масових інформаційних повідомлень.

Людина часто стає першою слабкою ланкою при проведенні кібератаки. Виконавець такої атаки після проведення розвідки і вивчення своєї цілі використовує отриману інформацію для входу до облікового запису або інших важливих відомостей, які проведуть його до захищених систем і ресурсів. Найчастіше зловмисникові щастить, і при мінімальних зусиллях і ще меншому ризику з його боку він може дізнатися облікові дані користувача просто на основі припущень, що може бути реалізовано навіть в автоматичному режимі. Тему соціальної інженерії складно обговорювати, тому що багато користувачів надмірно впевнені в своїх здібностях захистити облікові дані і не дотримуються належної «кібергігієни». Це вже призвело до кількох гучних порушень, про які можна докладніше прочитати в Інтернеті.

Говорячи про технічне забезпечення інформаційної безпеки в комп'ютерних мережах можна з впевненістю вказувати на те, що без розробки новітніх шифрів та систем захисту інформації, такий процес є неможливим. В той же час, системи безпеки розробляються безпосередньо людьми, які є найвразливішими до кібер-атак. Враховуючи викладене, можна говорити про те, що технічне забезпечення інформаційної безпеки повинно мати комплексну структуру, яка включає в себе як побудова стресостійких систем, так і проведення відповідної підготовки серед співробітників, які обслуговують такі системи.

Ключові слова: шифрування, інформаційна система, кібер-атака, загроза, зловмисники, захист.

Ключевые слова: шифрование, информационная система, кибер-атака, угроза, злоумышленник, защита.

Keywords: encryption, information system, cyber-attack, threat, attackers, protection.

Кузавков Василь Вікторович

Військовий інститут телекомунікацій та інформатизації імені Героїв Крут, начальник кафедри Побудови телекомунікаційних систем, доктор технічних наук, доцент

Болотюк Юлія Володимирівна

Військовий інститут телекомунікацій та інформатизації імені Героїв Крут, ад'юнкт науково-організаційного відділу

КОНТРОЛЬ ТЕХНІЧНОГО СТАНУ ВБУДОВАНИХ ОБЧИСЛЮВАЛЬНИХ СИСТЕМ

В процесі функціонування вбудованих обчислювальних систем (ВОС) контроль технічного стану (розв'язання задач технічного діагностування) набуває особливо важливого значення. Охоплюючи широкий спектр завдань і територій, відмова ВОС (втрата працездатності зазначеної системи) може призвести до критичних наслідків.

Відмінністю ВОС від інших технічних систем є тісний зв'язок між функціонуванням апаратної та програмної складовими системи, а також стандартизована організація обчислювальних процесів (обміну інформацією) відповідно до моделі OSI. Визначення відхилень в процесі функціонуванні ВОС (навіть на відстані) можливий завдяки фізичному підходу в виборі діагностичних ознак та діагностичного параметру. Виникнення нових безконтактних методів збору діагностичної інформації (реєстрації значень діагностичного параметру) дозволяють не лише вирішувати дві основні задачі технічної діагностики – визначення технічного стану і локалізація несправності, а й впритул підійти до вирішення третьої задачі – прогнозування технічного стану ВОС.

До діагностичних ознак в цьому випадку відносяться не лише енергетичні значення вимірюваного параметру, а й час (часові затримки виконання перевірних тестових повідомлень). Діагностування ВОС складається з

взаємопов'язаних частин (рис.1): апаратної (конструктивні параметрі h) та програмної, яка визначає алгоритм функціонування апаратної частини (конструктивні параметрі *h).* Тому діагностичний параметр повинен складатися з двох складових: перша – енергетична на основі стохастичних процесів зміни струму, друга – часова для відображення часу (зміни часу) виконання елементарних функцій при опрацюванні вхідної тестової послідовності.

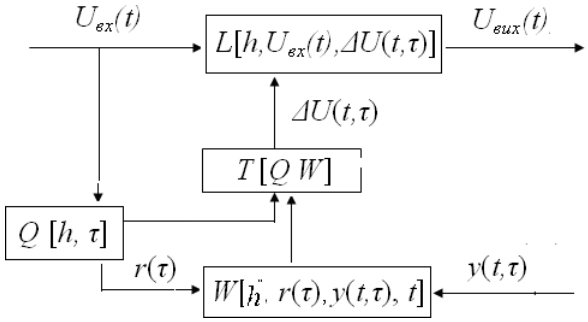

Рис.1 Узагальнена схема діагностичної моделі вбудованих обчислювальних систем

Процес діагностування (проходження тестових повідомлень) ВОС має наступні складові: $y(t, \tau)$, $y(t, \tau)_{ет}$ – діагностичний параметр та еталонне значення діагностичного параметру (ДП), які залежать від конструктивних параметрів:

h – апаратної складової об'єкту контролю, амплітудної та фазочастотної характеристики тракту формування (АЦП, ЦАП перетворень) та передачі тестових повідомлень;

t – час напрацювання об'єкту контролю;

τ – тривалості перевірочних тестових послідовностей;

$\Delta\tau_{\sum} = \Delta\tau_1 + \Delta\tau_2 + \Delta\tau_1$ – сумарне прирощення часу;

$\Delta\tau_1\Delta\tau_1$ – часове прирощення, пов'язане з функціонуванням апаратної частини об'єкту контролю;

$\Delta\tau_2$ – прирощення часу, яке характеризує затримки, обумовлені дисципліною обслуговування (обробки) тестових повідомлень у термінах теорії масового обслуговування, вплив параметрів налаштування операційного середовища h.

Оператор Q пов'язує між собою конструктивні параметри h (склад

програмних модулів або елементарних функцій) та вхідні перевірочні тести (з тривалістю τ). Параметр $r(\tau)$ відображає вплив програмної складової на апаратну складову об'єкту контролю.

Оператор W пов'язує між собою конструктивні параметри $\dot{h}$ (склад апаратної частини), вплив керуючої програмної складової $r(\tau)$, та енергетичну складову діагностичного параметру $y(t, \tau)$.

Оператор T введено для мультиплексування (поєднання) та відображення внеску обох конструктивних параметрів об'єкту контролю $\dot{h}$, h (програмного та апаратного) у фактор збудження $\Delta U(t, \tau)$ (поєднання операторів операторами Q та W).

Наведена модель враховує два види процесів, які відбуваються в тракті проходження тестових повідомлень: швидкі τ – флуктуація експлуатаційних (функціональних) показників і повільні t – розвиток дефектів в напівпровідникових структурах. Швидкі процеси визначають технічний стан вбудованих обчислювальних систем на момент діагностування (в момент часу контролю), а повільні – параметричну надійність об'єкту контролю.

Отримані дані застосовуються при експлуатації з метою попередження аварійних станів або при виході з ладу для скорочення часу відновлення. Враховуючи особливості ВОС, прийняття рішення на відновлення працездатного стану в реальному часі неможливо без застосування автоматизованих систем та заздалегідь підготовленого плану відновлювальних робіт.

Список використаних джерел:

1. Кузавков В. В., Хусаінов П. В. Прогнозування технічного стану однотипних програмно-апаратних засобів. *Інформатика та математичні методи в моделюванні*. 2018. № 1, С. 57-68.

Ключові слова: контроль технічного стану, вбудовані обчислювальні системи, технічна діагностика.

Ключевые слова: контроль технического состояния, встроенные вычислительные системы, техническая диагностика.

Keywords: technical control, embedded system, technical diagnostics.

Логінова Наталія Іванівна

Національний університет «Одеська юридична академія»

завідуюча кафедрою інформаційних технологій,

кандидат педагогічних наук, доцент

БЕЗПЕКА БАЗ ДАНИХ

Цифрова трансформація всіх сфер життєдіяльності призвела до того, що за допомогою інформаційно-комунікаційних технологій стало можливим зберігати, передавати та обробляти величезні масиви даних. Без використання таких масивів в електронному вигляді сьогодні не обходиться жодна організація чи підприємство, незалежно від форми власності. При цьому в сучасних інформаційних системах циркулюють різні типи даних, зокрема конфіденційні. Це створює передумови для здійснення різних неправомірних діянь та кіберзлочинів, зокрема, несанкціонованого доступу до інформації, крадіжки персональних даних, модифікації або знищення їх. Цінність такої інформації й сприяє підвищенню безпеки інформаційної системи, у тому числі й баз даних, які є основними елементами таких систем.

Головне завдання будь-якої бази даних – збереження величезних масивів даних, які повинні зберігатися з гарантуванням безпеки та конфіденційності.

Захист баз даних є одним із складних завдань у сфері інформаційної безпеки, оскільки не існує однакових баз даних. Вони відрізняються структурою побудови, способами обробки та доступу до даних, прикладними системами та багатьом іншим. Тому для кожної бази даних необхідно використовувати індивідуальний підхід до організації захисту та забезпечення безпеки.

Захист баз даних включає велику кількість технологій та інструментів, які дозволяють забезпечити безпеку збережених даних і захистити системи керування базами даних [1]. Найбільш ефективними заходами в цій сфері є:

- розподіл ресурсів баз даних та серверів;
- шифрування файлів та резервних копій баз даних;
- оновлення програмного забезпечення;
- здійснення контролю безпеки доступу до бази даних;
- аудит та моніторинг активності бази даних.

Численні інциденти хакерських атак показують, що найбільш уразливими є сервери, розміщені в «хмарах» і на різних зовнішніх хостингах. Розподіл ресурсів бази даних та серверів є надійним способом захисту даних. Наприклад, згідно з п. 2.2.1 Стандарту безпеки даних платіжних карток PCI-DSS (Payment Card Industry Data Security Standard) будь-яка база даних, що містить конфіденційну інформацію, повинна зберігатися на сервері, який безпосередньо не взаємодіє з Інтернетом [2].

Шифрування даних є стандартною процедурою багатьох баз даних. Але для захисту необхідно регулярно створювати резервні копії бази даних, які повинні бути зашифровані та зберігатися окремо від ключів дешифрування. Резервне копіювання бази даних захищає не тільки від атак хакерів, але і від збоїв, які можуть статися з обладнанням та програмним забезпеченням.

Оновлення програмного забезпечення бази даних має відбуватися регулярно. Необхідно встановлювати всі виправлення безпеки та захисту від вразливостей інформаційної системи, що постійно виявляються. Потрібно використовувати елементи керування безпеки, що надаються базою даних. Це особливо важливо для баз даних, підключених до великої кількості сторонніх застосунків, кожний з яких потребує виправлення.

Безпечний доступ до бази даних є одним із способів захисту. Необхідно прагнути мінімальної кількості користувачів, які мають доступ до ресурсів. Користувачам треба надавати мінімальні привілеї, що необхідні для виконання роботи з використанням бази даних, і лише у певний період. Потрібно керувати дозволами за допомогою груп та розмежувати ролі, а не надавати прямий доступ.

Рекомендується застосовувати автоматизоване керування доступом за допомогою спеціального програмного забезпечення. Це дозволить користувачам надавати тимчасовий пароль, який буде генеруватиметься при кожному вході до бази даних. При цьому щоразу надаватимуться тільки ті права, які їм необхідні в даний момент при роботі з базою даних. Це також дозволить відстежувати всі дії, виконані цим користувачем. Для захисту бази даних необхідно налаштувати стандартні процедури безпеки облікових записів. Користувачі повинні використовувати надійні паролі, хеш яких повинні зберігатися в зашифрованому вигляді. Необхідно налаштувати блокування записів користувачів після мінімальної кількості спроб входу до бази даних.

Моніторинг спроб входу до бази даних дозволяє визначити аномальні активності та зламування облікових записів. Моніторинг відбувається аналізом трафіку протоколу, який здійснюється від сервера управління мережею. За допомогою програмних агентів, які розміщуються на сервері керування можна налаштувати спостереження за активністю локальної бази даних на окремому сервері бази даних. Аудит працездатності бази даних дозволяє виявити системні проблеми та порушення політики конфіденційності та витоків даних, зіставляти дії у базі даних із подіями безпеки, створювати журнали аудиту всіх дій у базі даних.

Отже, для запобігання витоку інформації необхідно реалізувати комплексний захист бази даних. Необхідно використовувати розподіл доступу користувачів, розподілені ресурси баз даних та серверів, проводити моніторинг дій користувачів та аудит працездатності бази даних. Використовувати засоби шифрування файлів та резервних копій. Всі ці заходи необхідні для мінімізації ризиків, пов’язаних з втратою конфіденційної інформації баз даних і захисту інформаційної системи.

Список використаних джерел:

1. Database Security: 7 Best Practices & Tips. URL: https://www.esecurityplanet.com/networks/database-security-best-practices/ (дата звернення: 01.11.2021)

2. Is Splitting off Resources for Your Database Right for You? URL: https://www.liquidweb.com/blog/is-splitting-off-resources-for-your-database-right-for-you/ (дата звернення: 01.11.2021)

Ключові слова: база даних, захист даних, безпека баз даних, ресурси баз даних, інформаційна система

Ключевые слова: база данных, защита данных, безопасность баз данных, ресурсы баз данных, информационная система

Keywords: database, data protection, database security, database resources, information system

Василенко Микола Дмитрович

Національний університет «Одеська юридична академія», професор кафедри кібербезпеки, доктор фізико-математичних наук, доктор юридичних наук, професор

Шевченко Тетяна Вікторівна

Державний університет «Одеська політехніка», доцент кафедри міжнародних відносин та права, кандидат юридичних наук

ШТУЧНИЙ ІНТЕЛЕКТ ТА ЙОГО БЕЗПЕКА В ПУБЛІЧНОМУ АДМІНІСТРУВАННІ

Темпи розвитку штучного інтелекту (ШІ) дозволяють говорити про його суттєве використання в публічному адмініструванні. В Україні на виконавчому рівні задекларовано впровадження ШІ (високого порядку) в публічне управління і не тільки [1]. Однак ШІ такого порядку потребує відповідних інтелектуальних систем захисту інформації. Вони передбачають інтелектуалізації цих систем. При цьому все ж таки залишається відсутнім належне обґрунтування наслідків самих заходів щодо застосування ШІ в адміністративному управлінні щодо

мислення інтелекту машинного походження та його захисту, хоча вже відбулося затвердження плану заходів з реалізації Концепції розвитку ШІ в Україні на 2021-2024 роки [2].

Для розуміння можливостей ШІ вважають за можливе уявляти інтелект в цілому та використовувати його за подобою інтелекту та розробляти відповідні моделі та проводити їх навчання. В науці зі ШІ під інтелектом назагал вважається спроможність мозку до розумової діяльності, тобто до оперування знаннями для прийняття певних рішень стосовно конкретної задачі. Ця здатність мозку враховується при вирішуванні інтелектуальних задач шляхом придбання, запам’ятовування та цілеспрямованого перетворення знань у процесі навчання, отримання життєвого досвіду та адаптації до різноманітних зовнішніх та внутрішніх обставин. У цьому визначенні «інтелекту» під терміном «знання» мається на увазі не тільки та інформація, що поступає в мозок через органи чуття: вона інформація важлива, але недостатня для інтелектуальної діяльності. Річ у тім, що об’єкти навколишнього середовища володіють властивістю не тільки впливати на органи чуття, але і знаходитися один з одним в певних відносинах. Для того, щоб здійснювати в навколишньому середовищі інтелектуальну діяльність (або хоч би просто існувати), необхідно мати знання щодо моделі цього світу. У цій інформаційній моделі навколишнього середовища реальні об’єкти, їх властивості та відносини між ними не тільки відображаються і запам’ятовуються, але і можуть подумки цілеспрямовано перетворюватися. При цьому істотним є те, що формування моделі зовнішнього середовища відбувається у процесі навчання, отримання життєвого досвіду та адаптації до різноманітних обставин в системі середовища: природа – техніка – наука – суспільство.

Машинне навчання, яким користуються при створенні згаданих систем, становить підмножину технологій ШІ, яку можна описати як набір алгоритмів, що дозволяють комп'ютерам навчатися тому, на що вони з самого початку не запрограмовані. Іншими словами йдеться про алгоритмічну програму, яка сама може створювати програми для вирішення різних завдань. Алгоритм машинного навчання може навчатися на вибірці вже вирішених прикладів завдання (це

називається навчання з учителем), або він повинен самостійно виявити загальні риси і зв'язки в заданій множині об'єктів (навчання без вчителя). У разі навчання з учителем алгоритм може працювати в режимі навчання або в бойовому режимі. У режимі навчання алгоритм отримує навчальну вибірку, в якій об'єкти представлені своїми ознаками та мітками класів. З точки зору машинного навчання, нейронна мережа являє собою окремий випадок методів розпізнавання образів, дискримінантного аналізу, методів кластеризації тощо. Питання безпеки зазначеного ШІ розв’язується на програмно-технічному рівні інформаційної безпеки, який представляє собою рівень, у якому реалізуються різні програмно-технічні заходи, створені задля контроль працездатності устаткування й програмного забезпечення. Оскільки із загальної кількості загроз в останні роки все більшу питому вагу займають внутрішні загрози з боку персоналу, стосовно яких процедурні заходи не дають великого ефекту, саме на програмно-технічні заходи покладаються великі надії в силу, перш за все, великої різноманітності виконуваних ними функцій та способів їх реалізації [3 с. 13].

Одним із центральних понять для програмно-технічного рівня стало поняття сервісу (функції безпеки). Найбільш поширеними сервісами безпеки вважають [3, с. 13]: ідентифікація та аутентифікація; керування доступом; протоколювання та аудит; шифрування; контроль цілісності; екранування; аналіз захищеності; забезпечення відмовостійкості; забезпечення безпечного відновлення; тунелювання; керування. Деякі з цих сервісів виступають як «майданчик» для реалізації відповідних функцій захисту за допомогою методів штучного інтелекту. Зміст кожної функції достатньо добре визначено (див. [4]). Передбачається, що основні функції, які виконує інформаційна система, відносяться до її основних сервісів, а сервіси безпеки належать до групи допоміжних сервісів (функцій) системи.

Для підвищення ефективності публічного адміністрування важливим стає те, яким чином закладений алгоритм в ШІ, впливатиме на рішення, що приймається адміністративними, судовими або будь-якими іншими органами державної влади. Нині лідером у використанні ШІ у судочинстві залишаються США, де залучаючи ШІ до розгляду цивільних та кримінальних справ, судді

отримують «асистента» під час обрання запобіжного заходу щодо підсудного, оскільки через суб'єктивність та людський фактор трапляються розбіжності у прийнятті рішень різними суддями у подібних чи навіть тотожних ситуаціях. Натомість ШІ, аналізуючи ризики неупереджено на основі попередньої практики, визначає необхідність у запобіжному заході. Хоча, звісно, актуальним у цій ситуації залишається питання об'єктивності самого масиву даних для аналізу, адже попередні рішення приймалися звичайними суддями. Технології ШІ в ЄС, як і в Україні, відстають від США. Однак системи ШІ вже застосовуються для цифрової ідентифікації та верифікації особистості, а також впроваджуються: менше – у галузі охорони здоров'я, більше – для аналізу, прогнозування та моделювання показників ефективності публічного управління, для виявлення недобросовісної діяльності чиновників. Наявність відкритих державних даних стає першою передумовою до використання можливостей ШІ для покращення якості державних послуг. Так, Government AI Readiness Index оцінює готовність до впровадження штучного інтелекту в публічних сервісах. Індекс включає три основні компоненти: урядову політику, спроможність технологічного сектору, потенціал даних та інфраструктури. Розвивати ШІ в Україні планують і в різних напрямках публічного адміністрування. Так, ведуться роботи з використання ШІ в наявних технологіях – Електронний суд, Єдиний реєстр досудових розслідувань і так далі. WINCOURT – модуль автоматичного аналізу на платформі Суд на долоні. Він оцінює подібність судових документів, які завантажує користувач, до тих, на основі яких були вже вирішені подібні справи, та надає прогноз стосовно успішності їх розгляду. Verdictum PRO – сервіс з аналогічним функціоналом та спеціалізацією в господарському судочинстві. Для вирішення завдань щодо публічного адміністрування необхідно ще прийняти відповідні законодавчі акти, що урегулюють суспільні відносини у визначеній сфері. Однак, законодавче забезпечення регулювання системи ШІ та їх безпеки, наприклад зміни у судовій системі, взагалі бачаться достатньо складними, у той час коли технології ШІ натепер розвиваються досить швидко.

Список використаних джерел:

1. Про схвалення Концепції розвитку штучного інтелекту в Україні: Розпорядження Кабінету Міністрів України від 2 грудня 2020 р. No 1556-р // Урядовий портал. URL: https://www.kmu.gov.ua/npas/pro-shvalennyakoncepciyi-rozvitku-shtuchnogo-intelektu-v-ukrayini-s21220

2. Про затвердження плану заходів з реалізації Концепції розвитку штучного інтелекту в Україні на 2021-2024 роки. Розпорядження Кабінет Міністрів України від 12 травня 2021 р. № 438-р URL: https://zakon.rada.gov.ua/laws/show/438-2021-%D1%80#

3. Васильев В. И. Интеллектуальные системы защиты информации: учеб. пособие. Машиностроение, 2013. 172 с.

4. Галатенко В. А. Основы информационной безопасности. Курс лекций: учеб. пособие. М.: ИНТУИТ.РУ. 2004. 264 с.

Ключові слова: штучний інтелект , машинне навчання, публічне адміністрування, програмно-технічні заходи, безпека

Ключевые слова: искусственный интеллект, машинное обучение, публичное администрирование, программно-технические мероприятия, безопасность

Keywords: risk, artificial intelligence, machine learning, public administration, software and hardware, security

Бойко Віктор Дмитрович

Національний університет «Одеська юридична академія»,

доцент кафедри кібербезпеки, кандидат технічних наук, доцент

ПРАКТИЧЕСКИЕ АСПЕКТЫ ПЕРЕХОДА НА СВОБОДНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ В СОВРЕМЕННОМ АКАДЕМИЧЕСКОМ ОБРАЗОВАНИИ

В предыдущем докладе была обоснована актуальность и необходимость перехода на открытое программное обеспечение [1]. Были рассмотрены

причины, по которым учебные процессы в высших заведениях оказались насыщены пиратским (взломанным, либо не лицензированным явно) программным обеспечением, необходимость и неотвратимость все более строго контроля за лицензионной чистотой используемого программного обеспечения, устойчивый тренд среди разработчиков и профессионалов в IT-отрасли к увеличению и расширению использования средств открытого ПО - а значит и расширяющийся рыночный спрос на специалистов в этой области или специалистов из смежных областей, использующих открытое ПО, как инструментарий.

В качестве практического примера можно привести процесс переход на open source учебных курсов и компьютерных аудиторий кафедры кибербезопасности. Первоначально компьютерные аудитории использовали операционную систему Microsoft Windows 8. В качестве офисного пакета использовался Microsoft Office, представителями учебного программного обеспечения были Visual Studio (и язык C#), СУБД Microsoft SQL Server и пакет прикладных программ для решения задач технических вычислений MATLAB.

В соответствии с рекомендациями был выполнен анализ того, какими open source средствами и открытыми форматами данных можно заменить используемые в учебном процессе приложения.

В качестве базового языка программирования большинство курсов было переведено на язык Python с различными дополнительными инструментальными библиотеками, инструментами и фреймворками (numpy, scipy, matplotlib, pandas, sympy, simpy, http.server, unicorn, flask, django).

Преимуществом языка python является простота, компактность, развитость встроенного инструментария (Python поставляется по системе batteries included - то есть с большим выбором средств программирования сразу после установки дистрибутива языка), богатая база дополнительных приложений и библиотек.

При этом язык пользуется спросом на рынке труда. На момент написания, Python занимает первое место в рейтинге популярности языков программирования TIOBE (на втором находится язык Си, на третьем - Java, на четвертом и пятом - C++ и C# соответственно) [2]. По версии TIOBE Python

объявлялся языком года в 2007, 2010, 2018 и 2020 годах [3]. Все перечисленное обусловило стремление возможно более широко задействовать его в учебном процессе.

Дополнительное преимущество состояло в кросс-платформенности Python, что позволило в будущем совершить смену операционной системы с минимальным переучиванием.

Аналогичным образом был выполнен анализ возможных кандидатов на смену СУБД. Наиболее перспективными системами оказались следующие базы данных. Sqlite, работа с которой поддерживается языком python без сторонних библиотек, напрямую в режиме "из коробки". Mysql (и mariadb, которая практически не отличается по реализации, однако быстро и динамично развивается, в отличие от mysql). Postgresql, как полностью свободного представителя SQL баз данных, а также MongoDB, как наиболее широко используемого представителя NoSQL баз данных.

Перечисленные СУБД пользуются спросом на рынке труда. Например, на момент написания тезисов из 9851 вакансии на сайте jobs.dou.ua [4] знание Microsoft SQL Server требуется в 222 вакансиях (2.25%), PostrgreSQL - 901 вакансий (9.14%), MySQL - 1016 вакансий (10.31%). Похожее распределение наблюдается в опросе StackOverflow Survey 2021 [5]. В мае 2021 более 80000 разработчиков приняли участие в опросе платформы StackOverflow и по результатам опроса наиболее популярные СУБД распределились следующим образом - на первом месте MySQL (50.18%), на втором PostrgreSQL (40.42%), на третьем - SQLite (32.18%), на четвертом - MongoDB (27.7%). Microsoft SQL Server занимает пятое место (26.87%).

Следует отметить, что используемые в учебном процессе базы данных закрывают различные целевые направления разработки - MySQL и PostrgreSQL являются лидерами среди СУБД общего назначения, SQLite чаще всего используется, как встроенная БД, а MongoDB является представителем NoSQL баз данных.

В качестве офисного пакета поощряется максимально широкое использование пакета LibreOffice, кроме того, на основе "политики постепенного

перехода" осуществлялось поощрение работы с открытыми форматами (в частности - сдача протоколов лабораторных работ и другой документации в открытых форматах (OASIS Open Document Format for Office Applications (OpenDocument) / odt).

Для выдачи заданий использовалась документация в формате Portable Document Format (PDF), который является кросс-платформенным открытым форматом электронных документов. Благодаря кросс-платформенности и открытости для документов этого формата существует множество различного программного обеспечения практически на всех программных платформах.

Здесь перечисляется только основные представители открытого ПО, поскольку на самом деле список получился бы гораздо более объемным (GNU Octave, CAS Maxima, Scilab, GIMP, Inkscape, Scribus, Audacity - и так далее). Например, в качестве замены MATLAB предполагалось сочетание пакетов Scilab и GNU Octave, однако на практике оказалось удобным использовать Jupyter - в сочетаниями с математическими библиотеками Python (Numpy, Scipy, Matplotlib, Sympy, pandas и библиотеки, базирующиеся на них - scikit-learn, scikit-image, statsmodels и так далее) и интерфейсами к CAS Maxima.

Часть курсов в целях понимания работы процессорной архитектуры использует средства для работы с 16-битной архитектурой (NASM и некоторые элементы abandonware - например отладчик debug), поэтому в качестве дополнительного средства было решено использовать эмулятор DOSBox, позволяющий запускать DOS-приложения в наиболее распространенных операционных системах.

Таким образом учебные курсы были постепенно и последовательно переориентированы на использование свободного, кросс-платформенного ПО, чтобы подготовить смену операционной системы и избежать привязки к поставщику (англ. vendor lock-in либо proprietary lock-in).

Процесс перехода на использования открытого ПО, открытых форматов данных и выстраивания рабочих процессов в своей рабочей фазе происходил постепенно и последовательно - в соответствии с третьим пунктом рекомендаций предыдущего доклада.

После завершения процесса перехода, было начато осуществление следующего этапа - переход на свободную операционную систему. В качестве базовой системы использовался, основанный на Debian GNU/Linux, дистрибутив операционной системы Ubuntu. Не углубляясь в технические подробности, можно отметить удобство использования пакета clonezilla для больших парков однородных по своей комплектации рабочих компьютеров.

Переход на открытое кросс-платформенное ПО, с одной стороны облегчило переход на открытую операционную систему, с другой - оставило студентам свободу выбора средств самостоятельной работы с материалами курсов. Например, все работы могут быть выполнены в среде операционной системы macOS, если студент использует ноутбук фирмы Apple, либо в среде Windows, если такая установлена на его домашнем компьютере. В рамках некоторых курсов работы в порядке эксперимента даже выполнялись на платформе Android с использованием эмулятора терминала Termux.

Таким образом, учебные курсы и учебные материалы в соответствии с рекомендациями предыдущей публикации были сначала приведены к максимальному использованию кросс-платформенного и лицензионно-чистого (в основном - открытого) ПО, после чего был выполнен переход на платформу GNU Linux. Это позволило решить вопрос с лицензионной чистотой используемого ПО, уменьшило расходы по эксплуатации парка компьютерных классов, повысило устойчивость классов к вирусным атакам и распространению malware, улучшило многие вопросы, связанные с организацией учебных и рабочих процессов.

Использованная литература:

1. Бойко В.Д. Вопросы перехода на свободное программное обеспечение в современном академическом образовании: Наука та суспільне життя України в епоху глобальних викликів людства у цифрову еру (з нагоди 30-річчя проголошення незалежності України та 25-річчя прийняття Конституції України) (м. Одеса, 2021 р.). 2020. С. 607-610.

2. Home | TIOBE - The Software Quality Company [Електронний ресурс]. – Режим доступу : https://tiobe.com/tiobe-index/ (дата звернення: 2021-11-01). – Назва з екрана.

3. Python | TIOBE - The Software Quality Company [Електронний ресурс]. – Режим доступу : https://www.tiobe.com/tiobe-index/python/ (дата звернення: 2021-11-01). – Назва з екрана.

4. Вакансії DOU.UA [Електронний ресурс]. – Режим доступу : https://jobs.dou.ua/vacancies (дата звернення: 2021-11-01). – Назва з екрана.

5. Stack Overflow Developer Survey 2021 [Електронний ресурс]. – Режим доступу : https://insights.stackoverflow.com/survey/2021#technology-most-popular-technologies (дата звернення: 2021-11-01). – Назва з екрана.

Ключові слова: відкрите програмне забезпечення, відкриті формати даних, відкриті протоколи взаємодії, загрози і ризики, освіта.

Keywords: open software, open data formats, open communication protocols, threats and risks, education.

Ключевые слова: открытое программное обеспечение, открытые форматы данных, открытые протоколы взаимодействия, угрозы и риски, образование.

Задерейко Александр Владиславович

Национальный университет «Одесская юридическая академия»,

доцент кафедры информационных технологий, кандидат технических наук, доцент

Кухаренко Сергей Викторович

Национальный университет «Одесская юридическая академия»,

доцент кафедры кибербезопасности, кандидат технических наук

ЗАЩИТА ПЕРСОНАЛЬНЫХ ДАННЫХ ОТ УТЕЧЕК В ОПЕРАЦИОННЫХ СИСТЕМАХ СЕМЕЙСТВА WINDOWS

Большинство пользователей операционной системы (ОС) Windows, встречали в списке диспетчера задач, множество процессов с именем *svchost.exe*.

В зависимости от версии ОС Windows и от ее конфигурации, количество таких процессов может составлять от нескольких единиц до нескольких десятков.

Это объясняется тем, что *svchost.exe* является главным процессом для системных служб ОС Windows, загружаемых из динамических библиотек (*.dll). Исполняемый файл *svchost.exe* размещен в системном каталоге \Windows\system32\. Запуск системных служб организован так, что позволяет существенно уменьшить затраты оперативной памяти и ресурсов процессора. Все копии процессов *svchost.exe* запускаются системным процессом *services.exe* в ходе начальной загрузки ОС Windows, а также в ходе выполнения системных процессов, а также процессов связанных с запуском прикладного программного обеспечения (ППО) пользователя [1].

Системные процессы ОС Windows, особенно те, за запуск которых отвечает процесс *svchost.exe*, являются необходимыми компонентами для ее полноценной работы [2]. В процессе загрузки ОС Windows осуществляется запуск служб:

- диспетчера сетевых подключений;
- планировщика заданий;
- Plug and Play и HID;
- центра обновления;
- защитника ОС Windows;
- фоновой интеллектуальной службы передачи BITS и других.

При запуске ОС Windows служба *svchost.exe* проверяет разделы реестра со службами и создает список служб, которые загружаются в оперативную память. Каждый сеанс службы *svchost.exe* может выполняться в виде группирования служб. Такая организация необходима для уменьшения количества различных запущенных служб и улучшает их возможную отладку, если будет в этом необходимость. К примеру, один процесс *svchost.exe* запускает несколько служб связанным с встроенным сетевым экраном ОС Windows [3], который по умолчанию беспрепятственно пропускает все его исходящие соединения при наличии подключения к цифровому пространству Интернет (рис. 1).

Рис. 1. Схема организации соединений службы *svchost.exe* с цифровым пространством Интернет.

Такие настройки встроенного сетевого экрана являются предустановленными при установке ОС Windows на компьютер пользователя. Это обстоятельство в свою очередь приводит к полному или частичному дублированию всех исходящих/входящих соединений, осуществляемых прикладным программным обеспечением (ППО) пользователя, что само по себе представляет потенциальную угрозу для утечек персонализированных данных пользователя (рис. 2).

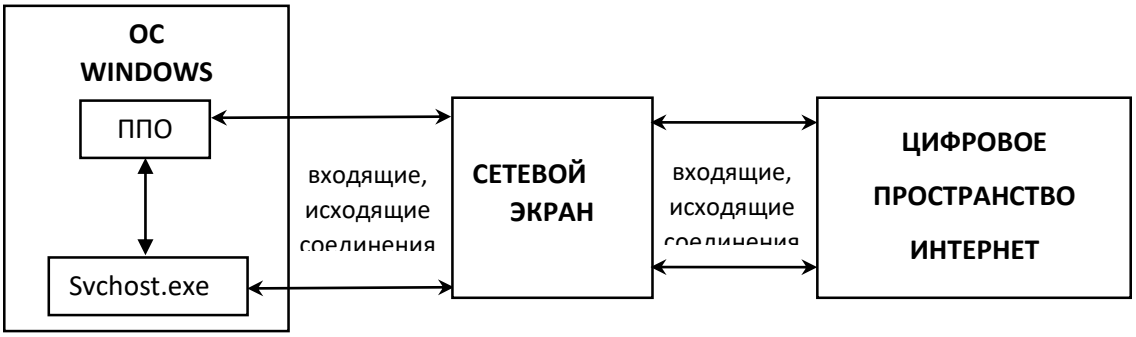

Рис. 2. Организация обмена данными между ОС Windows, ППО и цифровым пространством Интернет.

Для осуществления регламентированного доступа к цифровому пространству Интернет в семействе ОС Windows предусмотрено наличие встроенного брандмауэра (сетевого экрана) – системного программного обеспечения, которое выполняет контроль за обменом данными между ОС Windows и цифровым пространством Интернет [4].

Однако, как показывает долговременная практика углубленного исследования сетевого трафика ОС Windows, ППО и сетевого процесса *svchost.exe* с цифровым пространством Интернет с применением снифферов [5] – встроенный сетевой экран беспрепятственно пропускает большинство исходящих/входящих соединений системного ПО ОС Windows и ППО, которые неизбежно приводят к утечкам персональных данных пользователя [6, 7].

Анализ трафика исходящих/входящих соединений позволил разделить их на рабочие и сторонние. Также в процессе углубленного анализа было установлено наличие исходящих соединений сетевого процесса *svchost.exe*, которые можно классифицировать, как сторонние, так как они инициируются ППО и в большинстве случаев не соответствуют исходящим соединениям самого ППО (рис. 3).

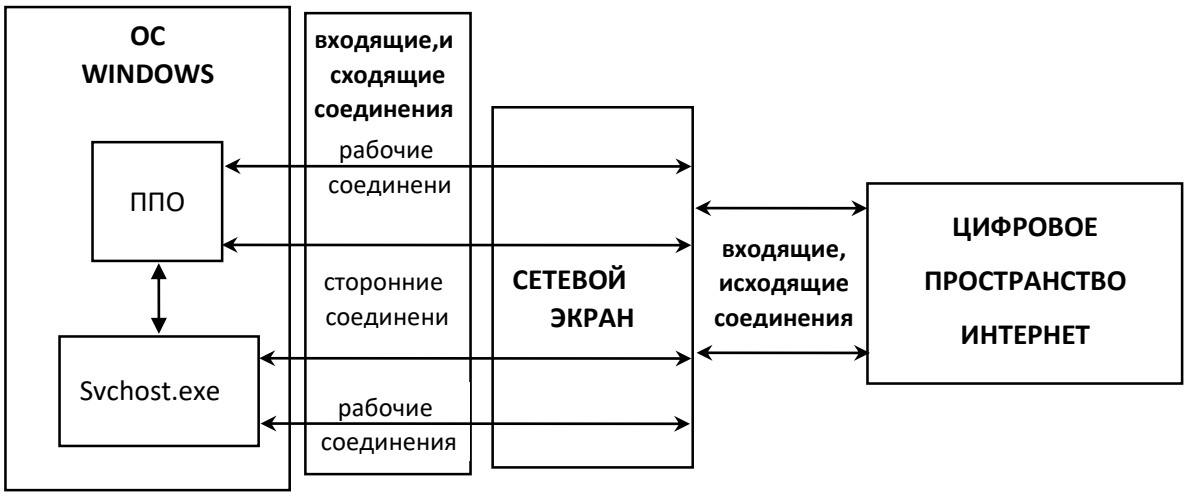

Рис. 3. Взаимосвязь сторонних и рабочих соединений ОС Windows, ППО и цифровым пространством Интернет.

Такая организация исходящих/входящих соединений в ОС Windows может с высокой долей вероятности привести к утечкам персональных данных пользователей. Для устранения этой проблемы, необходимо выполнить принудительную блокировку сторонних соединений ППО и сетевого процесса *svchost.exe* (рис. 4).

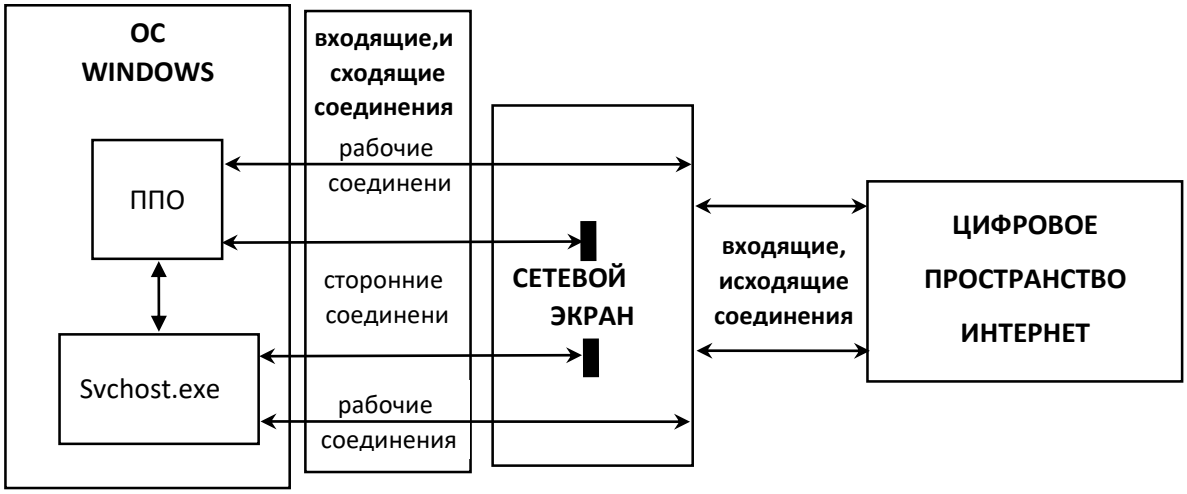

Рис. 4. Организация блокировки сторонних соединений ОС Windows и ППО с цифровым пространством Интернет.

В процессе длительного мониторинга и последующего анализа исходящих/входящих соединений, при организации блокировки сторонних соединений ОС Windows и ППО (см. рис. 4), была экспериментально подтверждена эффективность предлагаемого подхода для организации защиты персональных данных от потенциально возможных утечек.

Список использованных источников:

1. Процесс svchost.exe и системные службы, связанные с ним. : Веб сайт. URL: https://ab57.ru/svchost.html (Дата звернення: 1.11.2021).

2. Назначение процесса svchost.exe в ОС Windows. : Веб сайт. URL: https://www.starusrecovery.ru/articles/assigning-the-svchost-exe-process-windows.html (Дата звернення: 1.11.2021).

3. svchost.exe: Что это за хост-процесс? Почему грузит систему? [Электронный документ] - URL: https://mywebpc.ru/windows/sluzhba-uzla-svchost-exe/ (Дата звернення: 1.11.2021).

4. Брандмауэр и защита сети в системе безопасности Windows. : Веб сайт. URL: https://support.microsoft.com/ru-ru/topic-брандмауэр-и-защита-сети-в-системе-безопасности-windows-2385b621-9c5c-4834-9914-c7393f75fa84 (Дата звернення: 1.11.2021).

5. Снифферы (Sniffers). : Веб сайт. URL: https://www.anti-malware.ru/threats/sniffers (Дата звернення: 1.11.2021).

6. Algorithm of user's personal data protection against data leaks in Windows 10 OS / Olexander V. Zadereyko, Olena G. Trofymenko, Nataliia I. Loginova // Informatyka, Automatyka, Pomiary w Gospodarce i Ochronie Środowiska, vol.9, - p. 41 – 44. URL: https://yadda.icm.edu.pl/yadda/element/bwmeta1.element.baztech-b9e2407a-7fc7-41bf-a947-c285ad6b6d70.

7. Zadereyko A. Development of algorithm to protect user communication devices against data leaks / A. Zadereyko, Y. Prokop, O. Trofymenko, N. Loginova, O. Plachinda // Eastern-European Journal of Enterprise Technologies. – V. 1/2 (109) – Technology Center, 2021. – P. 24-34. URL: http://dspace.onua.edu.ua/handle/11300/14489.

Ключевые слова: утечки персональных данных, утечки данных в ОС Windows, блокировка утечек персональных данных.

Keywords: personal data leaks, data leaks in Windows OS, blocking of personal data leaks.

Ключові слова: витоки персональних даних, витоки даних в ОС Windows, блокування витоків персональних даних.

Лобода Юлія Геннадіївна

Національний університет «Одеська юридична академія»,

доцент кафедри інформаційних технологій,

кандидат педагогічних наук, доцент

ПРОГРАМНІ АНАЛІТИЧНІ ІНСТРУМЕНТИ BIG DATA ANALYTICS

Програмне забезпечення Big Data Analytics широко застосовується для ефективної обробки даних та досягнення конкурентної переваги на ринку. Програмні аналітичні інструменти Big Data допомагають відстежувати поточні ринкові зміни, потреби клієнтів, а також отримувати різну цінну інформацію. Розглянемо найпопулярніші інструменти аналітики.

Apache Hadoop. Лідер в обробці великих даних. Hadoop - це відкрита та безплатна система зберігання великих даних, набору супутніх утиліт, бібліотек, фреймворків, дистрибутивів для розробки. Ця основна технологія зберігання та обробки великих даних є проєкт верхнього рівня Apache Software Foundation [1]. Hadoop складається з чотирьох частин:

- HDFS – це розподілена файлова система, призначена для роботи на стандартному устаткуванні.
- MapReduce – модель розподілених обчислень, представлена Google, що використовується для паралельних обчислень.
- YARN – технологія, призначена для управління кластерами.
- Бібліотеки – для роботи інших модулів з HDFS.

Ambari. Вебінструмент для ініціалізації, управління та моніторингу кластерів Apache Hadoop, який включає підтримку Hadoop HDFS, Hadoop MapReduce, Hive, HCatalog, HBase, ZooKeeper, Oozie, Pig та Sqoop. Ambari також надає панель моніторингу для перегляду стану кластера, наприклад теплових карт, та можливість візуального перегляду додатків MapReduce, Pig та Hive, а також функції для діагностики їх характеристик продуктивності у зручній для користувача формі.

X-plenty. Ця хмарна платформа, що масштабується, пропонує рішення ETL і інструменти конвеєра даних. X-plenty обробляє як структуровані, так і неструктуровані дані та поєднується з різними джерелами, включаючи Amazon Redshift, сховища даних SQL, бази даних NoSQL та хмарні служби зберігання [2]. Основні переваги:

- легке перетворення даних;
- гнучкість у використанні;
- чудова безпека;
- різні джерела даних;
- клієнтоорієнтований підхід.

Spark. Сьогодні цей потужний інструмент аналітики з відкритим вихідним кодом є одним з основних в арсеналі компаній, включаючи Amazon, eBay та Yahoo. Apache Spark це технологія для роботи з великими даними за допомогою

розподілених обчислень в оперативній пам'яті, що збільшує швидкість обробки. Він заснований на Hadoop і є, по суті, еволюцією концепції MapReduce, використовуючи інші типи обчислень, включаючи інтерактивні запити та потокову обробку. Spark створений для широкого спектра робочих завдань, таких як пакетні програми, ітераційні алгоритми, інтерактивні запити та потокова передача. Це робить його ідеальним варіантом як аматорського використання, так професійної обробки великих обсягів даних [3].

Cassandra. Безплатна база даних NoSQL з відкритим вихідним кодом зберігає значення у вигляді пар ключ-значення. Цей інструмент – ідеальний вибір, коли потрібна масштабованість та висока доступність без шкоди для продуктивності [4]. Завдяки своїм архітектурним особливостям Apache Cassandra має такі переваги:

- масштабованість та надійність внаслідок відсутності центрального сервера;
- гнучка схема даних;
- висока пропускна здатність, особливо операцій записи;
- власна SQL-подібна мова запитів;
- налаштована узгодженість та підтримка реплікації;
- автоматичне вирішення конфліктів.

HBase. Масштабована розподілена база даних, що підтримує структуроване сховище для великих таблиць.

Hive. Інфраструктура сховища даних, що забезпечує підсумовування даних та спеціальні запити.

Mahout. Масштабована бібліотека машинного навчання та інтелектуального аналізу даних.

Talend. Аналітична програма, а точніше безплатний інструмент ETL з відкритим вихідним кодом, що спрощує та оптимізує інтеграцію великих даних. ETL спрощує перетворення необроблених даних на інформацію, яку можна використовувати для практичної бізнес-аналітики [5]. Програмне забезпечення Talend може похвалитися такими функціями, як хмара, великі дані, інтеграція корпоративних додатків, якість даних та керування основними даними. Він

також містить єдиний репозиторій для зберігання та повторного використання метаданих та перевірки якості даних. Особливості:

- більш швидка розробка та розгортання;
- менше витрат та безплатне завантаження;
- сучасне рішення;
- єдина платформа;
- величезно віддана спільнота.

Аналізуючи можливості та проблеми використання програмного забезпечення Big Data Analytics можна зробити висновки, що широкі спектри інструментів для роботи з великими даними допомагають зберігати, аналізувати, візуалізувати, складати звіти. Програмне забезпечення Big Data стимулює глобальні бізнес-процеси та сприяє прийняттю рішень, що базуються на знаннях.

Список використаних джерел:

1. The Apache™ Hadoop® project develops open-source software, available at: http://hadoop.apache.org/ (Accessed 10 Nov 2021).
2. Xplenty Simplified ETL & Reverse ETL, available at: https://www.xplenty.com (Accessed 10 Nov 2021).
3. Apache Spark™ - Unified Engine for large-scale data analytics, available at: https://spark.apache.org/ (Accessed 10 Nov 2021).
4. Apache Cassandra - The Apache Software Foundation!, available at: https://cassandra.apache.org/ (Accessed 10 Nov 2021).
5. Talend: Healthy Data, Healthy Business - Modern Cloud ETL, available at: https://www.talend.com (Accessed 10 Nov 2021).
6. Samoylenko, Lesya (2018), "Opportunities and problems of using big data technologies by domestic companies", Efektyvna ekonomika, [Online], vol. 1, available at: http://www.economy.nayka.com.ua/?op=1&z=6066 (Accessed 10 Nov 2021).

Ключові слова: програмне забезпечення, великі дані, технологія, аналітичні інструменти, база даних.

Ключевые слова: программное обеспечение, большие данные, технология, аналитические инструменты, база данных.

Keywords: software, big data, technology, analytical tools, database.

Манаков Сергій Юрійович

Національний університет «Одеська юридична академія»,

доцент кафедри інформаційних технологій, кандидат технічних наук

ПОРІВНЯННЯ МІКРОКОНТРОЛЕРІВ ТА ОДНОПЛАТНИХ КОМП'ЮТЕРІВ ДЛЯ ПОБУДОВИ СИСТЕМ ВІДЕОСПОСТЕРЕЖЕННЯ

Системи відеоспостереження є невід’ємною частиною сучасних систем контролю та безпеки. Актуальною є задача вибору мікроконтролера [1] або мікропроцесора у складі одноплатного комп’ютера, для побудови таких систем. На сьогодні існує низка популярних та доступних рішень, серед яких можна виділити: 8-бітні мікроконтролери, 32-бітні мікроконтролери та 64-бітні одноплатні комп’ютери.

Серед характеристик які впливають на якість таких систем можна виділити тактову частоту процесора, його розрядність та смугу пропускання використаного каналу зв’язку.

Вибір конкретного типу та моделі процесора залежить від завдань, які покладаються на проектовану систему відеоспостереження. Розглянемо їх детальніше.

8-бітні мікроконтролери. Серед представників цієї ланки популярними є моделі фірм-виробників Atmel, PIC, Texas Instruments тощо. Тактова частота таких мікроконтролерів зазвичай лежить у межах 1..20 МГц. Переважна кількість моделей має у своєму складі відомі та стандартизовані інтерфейси міжмікросхемної взаємодії, такі як USART, SPI та I^2C, та існує можливість використання зовнішніх комунікаційних модулей (Ethetnet, Wi-Fi) для підключення відеокамери та передавання відеосигналу. Зручними у

використанні є відомі плати Arduino [2, 3] з великою кількістю готових до підключення модулів та програмних бібліотек написаних мовою C++. Але, недоліком таких рішень є недостатня тактова частота контролерів, що призводить до неприйнятних значень частоти розгортки кадрів на пристроях відображення відеоінформації.

32-бітні мікроконтролери. Це мікроконтролери підвищеної розрядності, серед яких найвідомішими є представники сімейства STM32 [4] компанії STMicroelectronics. Всі вони базуються на 32-бітовому ядрі ARM, наприклад, Cortex-M7F, Cortex-M4F, Cortex-M3, Cortex-M0. Також за останні роки на ринку з'явилися доступні моделі сімейства ESP32 компанії Espessif, які постачаються на власній платі. Також, існує модель ESP32-CAM із вбудованою 2-мегапіксельною відеокамерою OV2640 та відеобуфером. Виробник цієї моделі передбачив web-інтерфейс для відображення та налагодження камери, а прошивка підтримує розпізнавання обличчя. Але, як показує практика, такі складні алгоритми не можуть повноцінно працювати на таких пристроях. Для того щоб забезпечити задовільну швидкодію потрібно зменшувати роздільну здатність до 320 / 360p, що не відповідає сучасним вимогам.

64-бітні одноплатні комп'ютери. Серед представників на сьогодні найвідомішими є різноманітні моделі Raspberry Pi [5]. Ці моделі являють собою повноважний одноплатний комп'ютер, з мікропроцесорами архітектури ARM. На платах представлені усі необхідні інтерфейси для побудови комп'ютерних систем різного призначення, такі як USB, Wi-Fi, Ethernet, аудіо- та відеовходи тощо. Тактова частота процесора до 2 МГц та об'єм оперативної пам'яті до 8 Гб разом із архітектурою дозволяють встановлювати операційні системи на ядрі Linux, та, навіть, Android. Підтримка ОС Linux та висока швидкодія дозволяють використовувати бібліотеки машинного навчання, такі як OpenCV чи TensorFlow, для функцій розпізнавання облич чи об'єктів. Також, можливе використання програмних кодеків для стиснення сигналу. До того ж, використання зовнішнього накопичувача HDD/SSD через інтерфейс USB дозволяє надійно зберігати великі обсяги відеоінформації, на відміну від на карт

пам’яті SD з обмеженням на кість циклів запису, які можливо приєднати до звичайного мікроконтролера по інтерфейсу SPI.

Як висновок, можна зазначити сфери застосування перелічених типів обладнання у контексті систем відеоспостереження.

Рішення на основі 8-бітних мікроконтролерів мають найнижчу вартість, та найнижчу швидкодію. Привабливими рисами є, також, невелике енергоспоживання та наявність режимів енергозбереження. Вони можуть знайти використання для відеофіксації зображень у складі простих систем сигналізації. Наприклад, автоматичне відсилання фото користувачеві за спрацьовуванням певного сенсора (руху, диму, затоплення тощо).

32-бітні мікроконтролери мають на порядок вищу швидкодію, яка дозволяє здійснювати відображення відеоінформації у реальному часі на рідкокристалічних дисплеях невеликого та середнього розміру. Прикладами застосування є проста веб-камера чи домофон з функцією розпізнавання облич. Такі рішення зазвичай використовують для зберігання даних карти пам’яті SD, що накладає обмеження на кількість циклів запису.

Одноплатні комп’ютери типу Raspberry Pi надають повний спектр можливостей, але характеризуються порівняльно високою вартістю та вимогами до енергоживлення і систем охолодження.

Список використаних джерел:

1. Tom Igoe. Making Things Talk, 2nd Edition. O`Reilly. 2011. ISBN: 9781449392437.

2. Marco Schwartz. Building a Wireless Security Camera With Arduino. CreateSpace Independent Publishing Platform. 2016. 38 p. ISBN: 978-1530780457.

3. Giuseppe Pervine. Arduino Ov7670 Camera: Step Guide To Use The Ov7670 With The Arduino: Ov7670 Camera Module Connection With Arduino. Independently published. 2021. 228 p. ISBN: 979-8536433720.

4. STM32F7 – DCMI. Digital Camera Inteface. Revision 1.0. Електронний документ. URL: https://www.st.com/content/ccc/resource/training/technical/product_training/group0/

90/5a/91/24/1a/14/4b/40/STM32F7_Peripheral_DCMI/files/STM32F7_Peripheral_DCMI.pdf/jcr:content/translations/en.STM32F7_Peripheral_DCMI.pdf

5. Dogan Ibrahim. Camera Projects Book: 39 Experiments with Raspberry Pi and Arduino. Elektor International Media; Main edition. 2019. 254 p. ISBN: 978-1907920776.

Ключові слова: Мікроконтролери, відеоспостереження, Arduino, Raspberry Pi, машинне навчання.

Ключевые слова: Микроконтроллеры, видеонаблюдение, Arduino, Raspberry Pi, машинное обучение.

Keywords: Microcontrollers, CCTV, Arduino, Raspberry Pi, machine learning.

Чепурна Олена Євгенівна

Національний університет «Одеська юридична академія»,

доцент кафедри кібербезпеки, кандидат фіз. - мат. наук

Кулешова Євгенія Романівна

викладач математики і інформатики ЗОШ №15 м.Одеси

МЕТОДИ МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ В СИСТЕМІ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ

Розвиток інформаційного суспільства диктує нові правила взаємодії між людьми, підприємствами, державними інституціями і світовими державами взагалі. Інтернет, який «видає», «поглинає» та обробляє великі об’єми інформації стає не тільки джерелом комфортного життя, але і найнебезпечніших загроз. Росте залежність від комп’ютерних технологій, додаток «Дія», наприклад, містить всю основну інформацію про людину: ID картка, ІПН, свідоцтво народження дитини, водійські права, документи підприємця та податкові декларації. Всі головні документи знаходяться в смартфоні, відомості

карткових рахунків і багато іншої особистої інформації. Великі маркетплейси і звичайні інтернет магазини при здійснені операцій використовують особисті дані. Захист інформації, в такому випадку, стає нагальною необхідністю. З точки зору кібербезпеки, ще більші загрози виникають при забезпеченні національної безпеки країни [1].

Тривалий час розуміння інформаційної безпеки в наукових та нормативно-правових джерелах ототожнювалося тільки з безпекою інформації, що значно звужувало її сутність. Саме тому, з низки питань, присвячених розгляду проблеми забезпечення інформаційної безпеки, найбільш вивченими та дослідженими її аспектами є безпека інформації (інформаційно-технічна безпека). Різні аспекти забезпечення інформаційної безпеки та вдосконалення управління нею досліджено у працях багатьох вітчизняних і зарубіжних учених, зокрема визначення змісту інформаційної безпеки, аналіз загроз та індикаторів інформаційної безпеки висвітлено в працях О.Ф. Бєлова, О.І. Барановського, М.А. Бендикова, М.М. Єрмошенка, Я.А. Жаліла, Т.Т. Ковальчука, Г.В. Козаченка, Н.О. Лоханова, О.М. Ляшенко, В.І. Мунтіяна, Є.А. Олейнікова, С.К. Реверчука та ін.[2].

Закон України «Про основні засади забезпечення кібербезпеки України» дає таке визначення: «Кібербезпека — захищеність життєво важливих інтересів людини і громадянина, суспільства та держави під час використання кіберпростору, за якої забезпечуються сталий розвиток інформаційного суспільства та цифрового комунікативного середовища, своєчасне виявлення, запобігання і нейтралізація реальних і потенційних загроз національній безпеці України у кіберпросторі»[3].

Застосування методів математичного моделювання, наприклад, моделі часових рядів AR(P), MA(Q), ARIMA(P,D,Q), лінійний дискримінантний аналіз (LDA), алгоритми дискримінантного аналізу, квадратичний дискрімінатний аналіз, на наш погляд, корисно на етапі прогнозування та попередження реальних і потенційних загроз.

Крім описаних моделей, досвід показує, що в роботі щодо забезпечення інформаційної безпеки, можуть використовувати специфічні прийоми

моделювання та особливі різновиди моделей. Найбільш прийнятними тут є логіко-математичні та розумові моделі, так як їх створення і вивчення найбільш доцільно у випадках з незаконним доступом до службової інформації.

Кіберзагрози не існують окремо від особи, яка спричиняє такі протиправні дії. В частині моделювання дій порушника інформаційної безпеки потрібно проводити аналіз поведінки такої особи(або групи осіб) з точки зору виділення ним ресурсів на напад, визначити ймовірності виділення кількості ресурсів, додати аналіз існуючих моделей порушників, вивчати статистику нападів на інформаційні мережі та системи

Додатково, доцільно, також залучити методи імітаційного моделювання, і реалізувати їх, наприклад, за допомогою GPSS (з англ. General Purpose Simulation System – системи моделювання загального призначення), які застосовують для дослідження систем масового обслуговування, так звані випадкові Марковські процеси. З точки зору теорії Марковських процесів, відносно до загрози виникнення вразливості атаки (при усуненні вразливості зникає і загроза атаки), інформаційну систему можна розглядати як систему з відмовами та відновленням характеристик інформаційної безпеки.

На сьогоднішній день існує багато задач оптимізації складних систем, що не можуть бути вирішені за допомогою аналітичних моделей. У такому випадку, на наш погляд, потрібно здійснювати нейромережеве моделювання для прогнозування рівня інформаційної безпеки.

Ефективному розвитку систем інформаційної і кібербезпеки, сприяє багато факторів, серед яких можна виділити застосування методів математичного моделювання на різних етапах її реалізації.

Список використаних джерел

1. Про рішення Ради національної безпеки і оборони України від 14 травня 2021 р. «Про Стратегію кібербезпеки України»: Указ Президента України. №447/2021, URL: https://www.president.gov.ua/documents/4472021-40013

2. Убийвовк І. І. Інформаційна безпека діяльності підприємств

Причорноморські економічні студії. 2016. №9. С. 2.

3. Закон України «Про основні засади забезпечення кібербезпеки України» від 5 жовтня 2017 року № 2163-VIII. URL: https://zakon.rada.gov.ua/laws/show/2163-19#Text

4. Левченко Є. Г. , Рабчун А. О. Оптимізаційні задачі менеджменту інформаційної безпеки. *Сучасний захист інформації*. 2010. №1 (1), С. 16-24.

Ключові слова: математична модель, інформаційна безпека, часові ряди

Ключевые слова: математическая модель, информационная безопасность, временные ряды

Keywords: mathematical model, information security, time series

Янковський Олег Георгійович

Національний університет «Одеська юридична академія»,

доцент кафедри інформаційних технологій,

кандидат технічних наук, доцент

ПРОБЛЕМНІ ПИТАННЯ ВИКОРИСТАННЯ ТЕХНОЛОГІЇ АУТЕНТИФІКАЦІЇ FACE ID

Сучасний світ неможливо уявити без використання інформаційних технологій у найрізноманітніших сферах людської діяльності. Обсяг електронної інформації, що використовується, стрімко зростає з кожним днем. Значну частину цієї інформації складає особиста інформація і саме тому питання захисту конфіденційних даних не втрачає своєї актуальності.

Сучасні системи контролю доступу до власних інформаційних ресурсів досить різноманітні та використовують різні підходи. Процедура встановлення належності користувачеві інформації в системі пред'явленого ним ідентифікатора називається автентифікацією. З позицій інформаційної безпеки автентифікація є частиною процедури надання доступу для роботи в

інформаційній системі, наступною після ідентифікації і передує авторизації [1].

Одним із перспективних способів аутентифікації є біометрична.

Біометрична автентифікація основана на унікальності певних антропометричних характеристик людини. У галузі інформаційних технологій термін біометрія застосовується в значенні технології ідентифікації особистості. Біометричний захист ефективніший ніж такі методи як, використання смарт-карток, паролів, PIN-кодів [1]. Найчастіше використовуються:

- параметри голосу;
- візерунок райдужної оболонки ока і карта сітківки ока — аутентифікація за райдужною оболонкою ока;
- риси обличчя;
- форма долоні;
- відбитки пальців;
- форма і спосіб особистого підпису — електронний цифровий підпис, верифікація підпису.

Система розпізнавання облич — це технологія, здатна ідентифікувати або перевірити особу на цифровому зображенні або відеокадрі. Існує багато методів, які використовуються в системах розпізнавання осіб, але в цілому вони ґрунтуються на порівнянні рис обличчя заданого зображенням з обличчями, які зберігаються в базі даних. Він також описується як біометричний додаток на основі штучного інтелекту, який може однозначно ідентифікувати людину шляхом аналізу моделей на основі текстур обличчя та форми людини [2].

На початку свого розвитку системи розпізнавання облич використовувались як окремі додатки. Але останнім часом вони все частіше реалізуються на мобільних платформах та в інших технологіях. Зазвичай цей метод використовується для контролю доступу в системах безпеки нарівні з іншими біометричними системами, такими як розпізнавання райдужної оболонки, або відбитків пальців. Незважаючи на те, що точність роботи системи розпізнавання облич як біометричної технології є нижчою, ніж розпізнавання райдужної оболонки ока та розпізнавання відбитків пальців, вона широко застосовується завдяки безконтактному та неінвазивному процесу. Останнім часом вона також

стала популярною як комерційний інструмент ідентифікації та маркетинговий інструмент [2].

Біометричні системи розпізнавання і програми порівняння облич поступово стали частиною нашого повсякденного життя, завдяки їх високій точності при виявленні та перевірці особистості людини - у цілях забезпечення безпеки, у комерції та інших сферах.

Одним з прикладів біометричної системи аутентифікації є Face ID - інноваційний спосіб автентифікації шляхом розпізнавання обличчя, в основі якого лежить створення структурної карти обличчя.

У сучасних умовах, пов'язаних з карантинними обмеженнями, особливу актуальність набуває технологія електронного підпису, яка використовується і у банківських операціях, і при отриманні вкрай необхідного COVID19-сертифікату про вакцинацію у системі «Дія».

Саме неупинно зростаюча популярність та все більш широке залучення (не завжди добровільне!) користувачів до використання електронних елементів бізнесу і державного сектору вивели на поверхню проблемні питання практичного використання технології Face ID.

Ще один з творців технології автоматичного розпізнавання облич Вуді Бледсоу (1966) описував наступні труднощі її використання: «Проблема розпізнавання ускладнюється великою мінливістю повороту і нахилу голови, інтенсивністю і кутом освітлення, виразом обличчя, віком, тощо. Однак метод кореляції (або зіставлення зі зразком) необроблених оптичних даних, які часто використовуються деякими дослідниками, безсумнівно, зазнає невдачі у випадках, коли мінливість велика. Зокрема, кореляція дуже низька між двома зображеннями однієї людини з двома різними поворотами голови» [1].

Було проведено аналітичне дослідження шляхом аналізу власного досвіду та опитування користувачів порталу «Дія», який використовує активний тип розпізнавання реальності особи (Дія.Підпис): просить користувача повернути голову певним чином або виконати якусь дію чи жест, дотримуючись випадкової інструкції — кліпнути 5 разів, закрити ліве око правою рукою, змінити вираз обличчя і т. д. (так звана методика «виклик-відповідь»).

У результаті були визначені основні фактори, які ускладнюють практичне використання технології Face ID:

- недостатня комп’ютерна грамотність конкретного користувача;
- залежність якості авторизації від зовнішніх умов;
- залежність якості авторизації від технічних параметрів пристрою доступу (низька роздільна здатність фіксуючого пристрою);
- залежність якості авторизації від фізичного стану пристрою доступу (забруднення, волога, низька температура тощо);
- недоліки мобільного інтерфейсу (малий шрифт підказок);
- важливість підтримання певної дистанції до камери (важко попасти у рамку);
- обмеженість часу на виконання певних дій.

Помилки авторизації виникали найбільш частіше у наступних випадках:

- зміна освітлення (недостатнє або зайве освітлення);
- зміна фону;
- наявність окулярів та інших аксесуарів;
- зміна зачіски (форма та колір волосся);
- зміна форми вусів або бороди;
- будь-яка міміка (посмішка, похмурий погляд);
- макіяж;
- одяг (шарф, головний убір);
- невпевнене та неточне відпрацювання команд.

Цілком зрозуміло, що використанні подібної технології викликало найбільше проблем у літніх людей та людей з обмеженими можливостями.

Список використаних джерел:

1. Автентифікація : веб-сайт. URL: https://uk.wikipedia.org/wiki/ (дата звернення 03.11.2021).

2. Система_розпізнавання_облич : веб-сайт. URL: https://uk.wikipedia.org/wiki/ (дата звернення 03.11.2021).

Ключові слова: автентифікація, біометричні методи, Face ID, електронний підпис.

Ключевые слова: аутентификация, биометрические методы, Face ID, электронная подпись.

Keywords: authentication, biometric methods, Face ID, electronic signature.

Баландіна Наталя Миколаївна

Національний університет «Одеська юридична академія»,

старший викладач кафедри кібербезпеки

ДОСЛІДЖЕННЯ НЕЛІНІЙНОСТІ S-БЛОКА НА ОСНОВІ ТЕОРІЇ ДИНАМІЧНОГО ХАОСУ ПРИ ЙОГО УЯВЛЕННІ ФУНКЦІЯМИ БАГАТОЗНАЧНОЇ ЛОГІКИ

Стійкість сучасних криптографічних алгоритмів у великій мірі залежить від криптографічних властивостей S-блоків, що застосовуються у них. Класичний підхід [1] до оцінки криптографічної якості S-блоків передбачає їх уявлення за допомогою математичного апарату булевих функцій, до яких застосовується стандартний набір критеріїв криптографічної якості: нелінійність, лавинні характеристики, кореляційні характеристики.

Однак дослідники все більш наполягають [2] на необхідності вивчення не тільки криптографічних показників компонентних булевих функцій сучасних шифрів, а також їх компонентних функцій багатозначної логіки (ФБЛ). Сьогодні існує набір критеріїв криптографічної якості ФБЛ, подібний до набору [1], створеного для булевих функцій. Відомі також публікації, присвячені дослідженню криптографічної якості відомих шифрів при їх виявленні за допомогою ФБЛ [3, 4].

Тим не менш, сучасні дослідження показують, що одними з найперспективніших методів синтезу S-блоків, є методи, засновані на теорії динамічного хаосу, яка дозволяє отримати підстановлювальні конструкції, що є

високоякісними з точки зору їх уявлення булевими функціями, тоді як їх властивості при уявленні за допомогою ФБЛ, залишаються невідомими.

Метою цієї роботи є дослідження та порівняльний аналіз нелінійних властивостей S-блока на основі теорії динамічного хаосу при його уявленні за допомогою ФБЛ.

Задля ясності подальшого викладу матеріалу коротко розглянемо сучасний підхід до вимірювання нелінійності булевих функцій та ФБЛ. Традиційним методом є оцінка відстані нелінійності булевих функцій у часовій області [5], що передбачає знаходження мінімуму серед відстаней Гемінга від досліджуваної булевої функції до кожного з кодових слів афінного коду. Іншим методом знаходження відстані нелінійності є метод, заснований на дослідженні максимального значення серед модулів коефіцієнтів перетворення Уолша-Адамара.

У [2] метод оцінки відстані нелінійності булевих функцій в області коефіцієнтів перетворення Уолша-Адамара був узагальнений на випадок ФБЛ. У цьому випадку відстань нелінійності визначається за такою основною формулою

$$N_f = \begin{cases} q^k - \max\left\{\left|\Omega_f\right|\right\}, & q > 2; \\ 2^{k-1} - \dfrac{1}{2}\max\left\{\left|W_f\right|\right\}, & q = 2, \end{cases} \qquad (1)$$

де Ω_f — коефіцієнти перетворення Віленкіна-Крестенсона ФБЛ, які знаходять шляхом множення її експоненціальної таблиці істинності, представленої над алфавітом

$$z_k = e^{j\frac{2\pi}{q}k}, \quad k \in \{0,1,\ldots,q-1\}, \qquad (2)$$

на матрицю Віленкіна-Крестенсона, рядки якої можна представити у вигляді

$$v_t(x) = e^{\frac{2\pi}{q}\sum_{i=1}^{k} t_i x_i}, \qquad (3)$$

де q — основа системи числення,

t_i — i-й елемент числа, записаного в позиційній q-ічній системі числення,

k — кількість елементів у q-ічному представленні значення N, що визначає довжину вибірки сигналу, $N = q^k$.

У випадку булевих функцій у формулі (1) замість перетворення Віленкіна-Крестенсона використовується перетворення Уолша-Адамара. Матриця перетворення Уолша-Адамара будується відповідно до такої рекурентної конструкції

$$A_{2^k} = \begin{bmatrix} A_{2^{k-1}} & A_{2^{k-1}} \\ A_{2^{k-1}} & -A_{2^{k-1}} \end{bmatrix}, \qquad (4)$$

де $A_1 = 1$.

Теорія динамічного хаосу є альтернативним рішенням для синтезу сучасних S-блоків. В даний час є багато публікацій, присвячених синтезу S-блоків на основі хаотичних відображень. Ми представляємо дослідження одного з якісних S-блоків на основі хаотичних відображень, який був запропонований в [5]

$S =$

	0	1	2	3	4	5	6	7	8	9	*A*	*B*	*C*	*D*	*E*	*F*
0	4B	DC	CF	DB	EA	5B	5F	65	88	01	4F	1B	FE	E7	3B	CE
1	72	22	71	DD	FA	D1	47	E8	3D	E4	2A	3F	B4	2C	CA	60
2	AF	FD	A3	0E	8E	9F	EF	74	8C	2D	6D	E9	C9	F0	89	85
3	04	67	0B	63	90	A6	05	B2	07	82	20	6A	7B	0F	AA	CD
4	11	23	1C	92	93	5E	D2	28	C2	9B	E6	AB	19	6B	1F	24
5	C3	E1	00	7D	08	FC	A9	D3	5A	73	BE	99	52	54	D0	77
6	E0	7F	3A	FB	43	55	8B	25	45	F9	02	DF	C0	9C	B7	38
7	C1	2B	40	37	A2	B5	98	ED	BC	AE	F2	09	80	FF	EB	E2
8	75	14	83	7C	D7	A7	44	87	64	2F	DE	9D	4C	D5	91	CB
9	48	21	57	F4	0A	1A	A0	30	46	7E	B8	84	F6	D6	13	F3
A	68	8A	9A	12	51	06	0C	D4	62	E3	3C	4A	A1	A5	6C	D9
B	6F	70	8F	5D	35	E5	56	39	5C	C6	EC	7A	4D	DA	3E	10
C	BA	D8	0D	1E	94	2E	AC	66	78	F5	CC	97	53	BD	33	32
D	C8	9E	AD	4E	50	03	EE	16	49	8D	C7	B9	26	61	95	B6
E	69	F8	B0	81	59	F1	36	A4	B3	96	27	79	76	C5	18	F7
F	34	29	31	15	BF	6E	42	58	41	B1	1D	17	A8	C4	BB	86

. (5)

Результати дослідження нелінійності S-блока (5), при його уявленні за допомогою компонентних булевих функцій, а також компонентних ФБЛ, наведені в табл. 1.

Таблиця 1 — Значення нелінійності компонентних функцій S-блока, синтезованого на основі теорії динамічного хаосу

Булеві функції							
f_{20}	f_{21}	f_{22}	f_{23}	f_{24}	f_{25}	f_{26}	f_{27}
112	112	112	112	112	112	112	112
4-функції				16-функції			
f_{40}	f_{41}	f_{42}	f_{43}	f_{161}		f_{162}	
219.1218	216.6046	219.9445	217.5292	219.3823		216.5184	

Аналіз даних табл. 1 показує, що нелінійність компонентних функцій S-блока (5) на основі хаотичного відображення є стабільною в булевому випадку і нестійкою у випадку представлення компонентними 4-функціями і 16-функціями. У випадку компонентних булевих функцій нелінійність S-блока дорівнює $N_{S,2}=112$, що становить 93.3% від максимального значення. При представленні S-блока за допомогою компонентних 4-функцій нелінійність дорівнює $N_{S,4}=216.6046$, що становить 90.25% від максимального значення. При представленні компонентними 16-функціями нелінійність дорівнює $N_{S,16}=216.5184$, що становить 90.22% від максимального значення.

Задля порівняння властивостей S-блока (5) з існуючими аналогами скористаємося даними дослідження [6], та побудуємо відповідну порівняльну табл. 2.

Таблиця 2 — Порівняння нелінійних властивостей S-блока (5), побудованого на основі теорії динамічного хаосу та відомих аналогів

S-блок	AES (без афінного перетворення)	Калина	BelT	Кузнєчік	Хаотичне відображення (5)

$N_{f_{2i}}$, мінімальне значення	112	104	104	102	112
$N_{f_{4i}}$, мінімальне значення	216.5538	211.7281	215.5031	213.9524	216.6046
$N_{f_{16i}}$, мінімальне значення	212.4385	201.1621	223.4209	208.5308	216.5184

Аналіз даних табл. 2 приводить до висновку, що S-блок (5), побудований на основі теорії динамічного хаосу перевищує за показниками нелінійності S-блоки відомих криптоалгоритмів: AES (США), Калина (Україна), BelT (Білорусь), Кузнєчік (Росія).

Відзначимо основні результати проведених досліджень:

1. Проведено дослідження та порівняльний аналіз нелінійних властивостей S-блока, побудованого на основі теорії динамічного хаосу при його уявленні за допомогою булевих функцій та ФБЛ з відомими аналогами.

2. Встановлено, що S-блок на основі хаотичних відображень показує результати, які перевищують результати S-блоків відомих криптоалгоритмів. Причому, найбільша перевага відзначається при представленні S-блоків за допомогою 16-функцій.

3. Таким чином, S-блоки на основі теорії динамічного хаосу є перспективними криптографічними конструкціями, які мають суттєві значення показників криптографічної якості не тільки при їх виявленні за допомогою булевих функцій, але і при уявленні за допомогою ФБЛ.

Список використаної літератури:

1. Жданов О.Н. Методика выбора ключевой информации для алгоритма блочного шифрования. М.: ИНФРА-М, 2013. 90 с.

2. Соколов А. В., Жданов О. Н. Криптографические конструкции на основе функций многозначной логики. Монография. М: Научная мысль, 2020. 192 с.

3. Sokolov A. V. Djiofack Temgoua Vanissa Noel. Nonlinear Properties of Rijndael S-boxes Represented by the Many-Valued Logic Functions. Proceedings of the International Workshop on Cyber Hygiene, Kyiv, Ukraine, November 30, 2019. P. 96—106.

4. Sokolov A. V., Radush V. V. Avalanche characteristics of Nyberg construction S-boxes represented by the many-valued logic functions. Informatics & Mathematical Methods in Simulation, 2019. No. 9 (3). P. 111-119.

5. Hussain I., Anees A., Al-Maadeed T. A. et al., Construction of S-Box Based on Chaotic Map and Algebraic Structures. Symmetry. 11. 351. 2019.

6. Kazakova N. F., Karpinski M., Sokolov A. V., Gancarczyk T. Nonlinearity of Many-Valued Logic Component Functions of Modern Cryptographic Algorithms S-boxes. Procedia Computer Science. Vol. 192. P. 2731-2741.

Ключові слова: криптографія, S-блок, нелінійність, булева функція, функція багатозначної логіки.

Ключевые слова: криптография, S-блок, нелинейность, булева функция, функция многозначной логики.

Keywords: cryptography, S-box, nonlinearity, Boolean function, many-valued logic function.

Толокнов Анатолій Арнольдович

Національний університет «Одеська юридична академія»,

асистент кафедри інформаційних технологій

ОГЛЯД БАЗ ДАНИХ ВРАЗЛИВОСТЕЙ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Під вразливістю інформаційної системи розуміють таку властивість системи, з якої можлива реалізація загрози її безпеки [1]. Оцінивши ризики та ймовірність певних загроз, можна вибрати, яка вразливість є більш прийнятною з точки зору ймовірності реалізації дочірніх загроз та величини отриманих від них збитків. Не існує інформаційних систем, що не містять вразливостей, існують лише настільки захищені, стосовно яких можна сказати, що ймовірність виникнення актуальних загроз безпеці інформації є прийнятною.

У 90-ті роки XX століття, у зв'язку з розвитком Інтернету, стала очевидною необхідність систематичного збору даних про вразливості програмного забезпечення (ПЗ) та їх каталогізації. Актуальність цього завдання була зумовлена і появою нових видів ПЗ: операційних систем та серверних платформ, прикладних програм, веб-застосунків та ПЗ для вбудованих систем, збільшенням числа версій у лінійці одного і того ж програмного продукту, а також зростанням частоти виявлення в цих програмах вразливостей, що дозволяє зловмиснику отримати повний контроль над комп'ютерною системою або доступом до критичної інформації.

Створення реєстрів та баз даних з інформацією про виявлені вразливості потребувало уніфікованого способу їх ідентифікації та класифікації. Дані про індивідуальні вразливості в сукупності з організованим зберіганням, функціями пошуку по базі та автоматичного отримання повідомлень про нові виявленні вразливості можуть бути корисні широкому колу фахівців у галузі інформаційної безпеки. Кінець 90-х і початок 2000-х років характеризувався появою цілого ряду каталогів і баз даних вразливостей, підтримкою яких займалися різні, незалежні один від одного, організації. Поява та розвиток незалежних однієї від одної баз

даних та реєстрів вразливостей породило проблему синхронізації інформації між ними.

Стандарт Common Vulnerabilities and Exposures (CVE), розроблений американською некомерційною дослідницькою корпорацією MITRE Corporation у 1999 році [2], де-факто є на сьогоднішній день основним стандартом у галузі уніфікованого найменування та реєстрації виявлених вразливостей програмного забезпечення.

Основними задачами організацій, що входять в проект MITRE CVE, є:

- пошук та збір інформації про вразливість програмного забезпечення;
- класифікація знайдених вразливостей;
- резервування CVE-ідентифікаторів для знайдених вразливостей;
- оновлення відповідної інформації в двох офіційних каталогах – реєстр вразливостей CVE List (https://cve.mitre.org/cve/) самої корпорації MITRE і бази даних вразливостей NVD (National Vulnerability Database, https://nvd.nist.gov/) , підтримується національним Інститутом Технології та Стандартів США.

На сьогоднішній день бази вразливостей MITRE CVE List і NVD містять близько 98 тисяч записів про окремі вразливості, виявлені за період з 1999 року до теперішнього часу. Ідентифікатори CVE мають формат CVE-YYYY-NNNN, відображають в перших чотирьох цифрах рік реєстрації вразливості та в наступних чотирьох-шістьох цифрах – унікальний номер вразливості у рамках цього року.

База VND (Vulnerability Notes Database) підтримується центром реагування CERT/CC при Інституті програмної інженерії в університеті Карнегі-Меллона, США [3].

Кожен запис у базі VND агрегує інформацію про безліч подібних вразливостей для будь-якого конкретного ПЗ, посилаючись на множину відповідних CVE ідентифікаторів. Дана агрегація є характерною відмінністю бази VND від баз CVE List і NVD, дозволяючи перевірити безліч вразливостей подібної природи в конкретному вразливому ПЗ або його компоненті.

Крім цього, записи в базі VND часто містять докладний і детальний посібник з усунення вразливостей та/або запобігання їх експлуатації

зловмисником, а також огляд поточної ситуації з наявністю або успішним закриттям виявленої вразливості різними вендорами.

Дані характеристики бази VND належать до її сильних сторін і доповнюються дозволом на безкоштовне використання всіх матеріалів бази для будь-яких цілей і можливістю повного завантаження всіх записів бази у форматі JSON за допомогою спеціального програмного забезпечення, що безкоштовно надається.

Альтернативним підходом до каталогізації інформації про виявлені вразливості ПЗ є реєстрація не самих вразливостей, а сценаріїв їх експлуатації (експлойтів, exploits) або прикладів експлуатації вразливості (Proof of Concept). Прикладом реалізації такого підходу є база Exploit Database [4], сформована організацією Offensive Security Team, що спеціалізується на проведенні тренінгів з інформаційної безпеки та тестування комп'ютерних систем на проникнення.

База Exploit Database на даний момент містить близько 44,5 тисяч записів, розбитих на різні категорії (експлойти для веб-застосунків, віддаленої та локальної експлуатації вразливостей, приклади атак Denial of Service та можливі фрагменти коду shellcode для різних вразливостей переповнення стека або доступу до пам'яті). Дані записи покривають безліч вразливостей, виявлених з 2000 року до теперішнього часу.

Типовий запис у базі Exploit Database містить короткий опис вразливості, вказівку вразливих версій додатків або їх компонентів, вразливу програмну платформу (операційну систему або фреймворк веб-застосунку), CVE-ідентифікатор, присвоєний даній вразливості (за його наявності). Однак найважливіша і змістовна частина запису – це детальний опис причин виникнення вразливості, місця локалізації вразливості в коді (з безпосередньою демонстрацією вразливого фрагмента коду, якщо код програми публічно доступний) і опис працездатних сценаріїв експлуатації вразливостей або сценаріїв Proof of Concept.

Різноманітність різних реєстрів та баз даних вразливостей викликає у фахівців у галузі інформаційної безпеки бажання використовувати різного роду агрегатори інформації, які б забезпечували автоматизований збір доступної

інформації про вразливості та додаткові функції пошуку та фільтрації інформації.

Прикладом такого сервісу є CVEDetails (https://www.cvedetails.com/) – простий спеціалізований агрегатор інформації про вразливості, який збирає всю доступну з публічних реєстрів та баз даних вразливостей інформацію щодо конкретного CVE-ідентифікатора та об'єднує її в єдиний запис.

Фактичним функціоналом даного сервісу є автоматизація пошуку всієї доступної інформації по CVE-ідентифікатору з додатковими функціями пошуку за вендорами, типами вразливостей, оцінкою критичності за метриками CVSS тощо. Також реалізовано збір та зберігання різноманітної статистики за вразливістю, наприклад, розподіл вразливостей за ступенем критичності (згідно з метрикою CVSS), розподіл вразливостей за вендорами ПЗ та ін. – зі зручним переходом до списку вразливостей, що задовольняють обраному критерію.

Таким чином, бази даних та реєстри вразливостей корисні широкому колу фахівців у галузі інформаційної безпеки. Мережеві адміністратори або співробітники, відповідальні за безпеку комп'ютерних систем організації, можуть своєчасно дізнатися з баз і реєстрів про появу нових загроз для систем, що захищаються, визначити пріоритетні заходи реагування на ці загрози.

При цьому для фахівців важливі оперативність оновлення баз даних вразливостей, зручність отримання цих оновлень і ступінь покриття обраним реєстром вразливостей як основних видів програмного забезпечення, що захищається, так і всієї безлічі вразливостей, виявлених для цього ПЗ. Також значущими характеристиками будуть наявність рекомендацій щодо усунення вразливостей і можливість визначення потенційних векторів атак на комп'ютерні системи, що захищаються.

Список використаних джерел:

1. НД ТЗІ 1.1-003-99: Термінологія в галузі захисту інформації в комп’ютерних системах від несанкціонованого доступу. Київ: Департамент спеціальних телекомунікаційних систем та захисту інформації Служби безпеки України. – 1999. – С. 5.

2. About CVE [Електронний ресурс] // The MITRE Corporation. – 2021. – Режим доступу до ресурсу: https://cve.mitre.org/about/index.html.

3. Vulnerability Notes Database [Електронний ресурс] // Carnegie Mellon University. – 2021. – Режим доступу до ресурсу: https://www.kb.cert.org/vuls/.

4. Exploit Database [Електронний ресурс] // OffSec Services Limited. – 2021. – Режим доступу до ресурсу: https://www.exploit-db.com/.

Ключові слова: вразливості, інформаційна безпека, бази даних, комп'ютерні системи.

Ключевые слова: уязвимости, информационная безопасность, базы данных, компьютерные системы.

Keywords: vulnerabilities, information security, databases, computer systems.

Плаксін Олександр Андрійович

Національний університет «Одеська юридична академія»,

студент 3-го курсу факультету кібербезпеки та інформаційних технологій

СУЧАСНІ КРИПТОГРАФІЧНІ МЕТОДИ ЗАБЕЗПЕЧЕННЯ БЕЗПЕКИ ДАНИХ

Криптографія – це наука про приховування інформації від несанкціонованих осіб.

Криптографія починає свою історію ще з далеких часів Римської імперії, коли був придуманий один із перших шифрів - Шифр Цезаря. З того часу криптографія просунулась далеко вперед і тепер використовується по всьому світу навіть у звичайнісіньких для нас речах: месенджерах, соціальних мережах, при перегляді ТБ, при серфінгу Інтернету тощо.

Зараз у світі існує безліч способів приховування інформації. Розберемо принцип їхньої дії, їх переваги та недоліки.

1) RSA (абревіатура від прізвищ Rivest, Shamir та Adleman) — один з найвідоміших методів шифрування, більш відомий як метод "ключа та замка". Працює він за таким принципом: Співрозмовник 1 створює свій ключ і замок, співрозмовник 2 робить так само. Далі вони обмінюються "замками". Після цього співрозмовник 1 закриває своє повідомлення замком, який йому дав співрозмовник 2. Так, тепер навіть сам співрозмовник 1 не може відкрити своє повідомлення, але це легко може зробити співрозмовник 2, тому що він має ключ від свого ж замку, яким закрито повідомлення від співрозмовника. 1.Метод працює шляхом створення величезних чисел за рахунок 2х невеликих натуральних чисел. Величезне число, що вийшло внаслідок множення двох натуральних чисел - це і є той самий "замок", тобто публічний ключ. А пара чисел, що дає велике число при перемноженні - це і є приватний ключ. Компрометація приватного ключа може призвести до компрометування всіх зашифрованих даних.

Переваги методу RSA:

1. Діапазон застосування. Даний метод використовується практично в кожній системі шифрування, починаючи від сайтів, закінчуючи різними сервісами обміну повідомлень.

2. Стабільність. Імовірність "неправильного шифрування" при використанні цього методу досить низька за рахунок невеликої кількості факторів, що забезпечують шифрування.

3. Простота використання. Достатньо згенерувати пару ключів, зашифрувати публічним ключем свої дані, а приватний ключ можна перемістити на флешку та носити його із собою. Так, компрометація приватного ключа зводиться нанівець.

Недоліки методу RSA:

1. Не найкраща криптостійкість. Вже є випадки, коли ці два натуральні числа вдалося підібрати через їх слабку генерацію. Це призвело до повної компрометації даних, незважаючи на гарний захист приватного ключа.

2. Кастомізація — не під силу кожному.

Для створення надійної пари ключів зі своїми числами та певною довжиною, необхідно володіти деякими знаннями в галузі математики.

3. Відсутність додаткових захисних механізмів. Незважаючи на те, що приватний ключ можна зашифрувати паролем під час використання, це не виключає ризик підбору цього пароля методом грубої сили.

Незважаючи на це, RSA залишається досить надійним та популярним способом зашифрувати свої дані від третіх осіб.

У руках людини, яка розуміється на математиці та криптографії, цей метод перетворюється вже на щось більше, ніж просто пара ключів і "замків".

2) AES (Advanced Encryption Standard) - сучасний "золотий стандарт" шифрування, ухвалений основним стандартом шифрування урядом США.

Використовує симетричну (на відміну RSA, яка асиметрична) блокову систему шифрування.

Переваги методу AES**:**

1. Широке використання у всьому світі. AES - золотий стандарт, на який всі намагаються дорівнювати. Потужніше шифрування, ніж AES, поки що, не придумали, хіба що багаторазове шифрування цим способом.

2. Висока криптостійкість. Завдяки алгоритму генерації раундових ключів, зміщення рядків та зворотної лінійної трансформації, метод став по-справжньому стійким до різних технік зламів, які ефективні проти інших методів, але не ефективні проти AES.

3. Висока надійність. AES використовує симетричний метод шифрування, тобто ключ, який шифрує - він і розшифровує. Створивши надійний пароль, припустимо, на 32+ символи, на його перебір підуть роки, якщо не століття. Це єдиний спосіб скомпрометувати дані.

Недоліки методу AES:

1. Симетричність. Якщо пароль буде скомпрометовано – всі дані будуть доступні третім особам. Пароль, на сьогоднішній день, не є надійним фактором захисту через нестійкість до перебору.

2. Потрібні деякі навички в галузі криптографії. Існує ймовірність "неправильного шифрування" даних. У такому разі є великий ризик втратити

свої дані для недосвідченого користувача. Перед використанням даного методу краще ознайомитися з інструкцією.

3. Квантова нестійкість. Нещодавні дослідження швейцарських криптографів показали, що алгоритм AES може бути "вскритий" за допомогою квантового суперкомп'ютера протягом кількох років [1].

Тим не менш, на сьогоднішній день, AES залишається одним із найкращих світових стандартів шифрування.

3) GreyNigma(Gigma) - нова розробка в області криптографії, що дозволяє конвертувати текст, що вводиться, в цифри, закриті кривими дужками [2].

Всі цифри, на які замінюються символи - повністю довільні і виставляються користувачем, а також можуть бути згенеровані за допомогою спеціального вбудованого генератора.

Gigma використовує симетричний спосіб шифрування. Усі цифри, на які замінюються символи, зберігаються в спеціальній конфігурації, яку потрібно передати співрозмовнику. Після цього співрозмовник зможе розшифровувати та писати зашифровані повідомлення. Якщо конфігурація не співпадатиме з тією, якою шифрувалося повідомлення - на виході вийде випадковий набір із символів.

Gigma має такі захисні механізми:

- Зміщення літер**.** Всі букви, що вводяться, можуть бути "зсунуті" вліво або вправо, подібно до шифру Цезаря. Зрушення виробляється на певне число, визначене у конфігурації, воно може бути від -26 до 26, оскільки поки що підтримуються лише літери англійського алфавіту, а він має 26 літер.
- Поділ рядків**.** Усього є цілих 3 варіації рядка. Основна їхня відмінність - це те, що у кожного рядка свої значення для кожного символу. У перший рядок можна записати справді важливу інформацію, а в два інші - спотворену чи зовсім непотрібну.
- Випадковий шаблон**.** Кінцеве зашифроване повідомлення може мати абсолютно випадковий порядок рядків, що лише посилює криптостійкість шифру, але ніяк не впливає на розшифровку та кінцевий результат.
- Присолювання**.** Для збільшення потужності шифру, а також штучного збільшення розміру зашифрованого повідомлення, існують спеціальні порожні

рядки, що називаються "солі". Вони виступають як баласт, але при розшифровці правильною конфігурацією вони просто зникають.

- Розворот рядка. Крім того, все повідомлення може бути представлене у зворотному порядку. Ця опція так само визначається конфігурації і повністю регульована. Наприклад, слово "Hello" може перетворитися на "olleH", тим самим лише забезпечуючи ще більший ступор третіх осіб, які можуть намагатися розшифрувати повідомлення.

Конфігурація може бути зашифрована методом RSA за допомогою спеціально розробленого інструменту та безпечно передана співрозмовнику, не переймаючись тим, що може бути перехоплена.

З недоліків можна відзначити тільки те, що все ж таки, цей шифр може бути зламаний шляхом перебору, але враховуючи те, що в сукупності всього існує мінімум 225! (Факториал) варіацій розмістити символи для одного рядка (а їх 3) в конфігурації для одного повідомлення, це може зайняти десятки років, або навіть сторіччя.

Таким чином, у кожного розглянутого метода шифрування є свої недоліки і переваги, що дозволяє використовувати їх безпосередньо за їх призначенням.

Список використаних джерел:

1. Дослідження швейцарських криптографів щодо вразливості AES. URL: https://roskomsvoboda.org/69451 (Дата звернення: 1.11.2021).

2. Grey_Hat_Cybersecurity / GreyNigma. URL: https://notabug.org/Grey_Hat_Cybersecurity/GreyNigma (Дата звернення: 1.11.2021).

Ключові слова: Криптографія, шифрування, RSA, AES, GreyNigma, безпека

Ключевые слова: Криптография, шифрование, RSA, AES, GreyNigma, безопасность

Keywords: Cryptography, encryption, RSA, AES, GreyNigma, security

Науковий керівник: *к.т.н., доц. Задерейко О.В.*

Астахов Даниїл Юрійович

Національний університет «Одеська юридична академія»,

студент 2-го курсу факультету кібербезпеки та інформаційних технологій

ЗАСОБИ ЗАХИСТУ ПЕРСОНАЛЬНИХ ДАНИХ В МЕРЕЖІ ІНТЕРНЕТ

Сьогодні Інтернет є найбільш важливим ресурсом для спілкування та навчання. Це дуже великий крок для людства в цілому. В мережі інтернет є позитивні сторони, такі як багатий обсяг будь-якої інформації і про що завгодно, але звідси і виходять негативні сторони, тобто, переважна незахищеність персональних даних користувачів.

Базові способи захисту даних користувачів мережі інтернет?

Проблема з втратою персональних даних на користь зловмисника завжди була, є і буде. Персональними даними не обов'язково можуть бути ім'я чи прізвище, чи номер телефону, але і IP-адреса та будь-яка інша інформація, яка може ідентифікувати особистість користувача у цифровому просторі.

Є декілька порад для того, щоб максимально захистити користувача під час з'єднання з мережею інтернет:

1. Не підключатись до громадських мереж WI-FI, які не захищені паролем, тому, що вони можуть бути створеними мережами-двійниками для перехоплення даних користувачів.

2. Створювати складні паролі та змінювати їх як мінімум раз на місяць. Слід придумати багато різних та складних паролів, записати їх у блокнот для того, щоб максимально захистити особисті дані. Також є застосунки, які самі створюють складні паролі, прикладом такого застосунку є LastPass.

3. Використовувати двофакторну аутентифікацію – дуже важливий критерій для захисту даних. Кожен раз, коли користувач вводить свій пароль, він

має підтвердити, що це саме він через повідомлення. Це найголовніший засіб зупинення злочинця.

4. Слідкувати за адресним рядком. Краще використовувати захищені підключення, такі як HTTPS, а не HTTP.

5. Шифрувати повідомлення. В багатьох месенджерах вже давно є спеціальні системи, які допомагають захистити дані за допомогою шифрування повідомлень. Наприклад, в Фейсбуці є таємне листування. В системних налаштуваннях IPhone можна увімкнути функцію шифрування [1].

Методи захисту інформації, якими забезпечують розробники

У 2018 році в Європі було прийнято правила захисту персональних даних, а саме GDPR. За їх порушення прийдеться платити штраф.

GDPR – загальний регламент захисту персональних даних. Правила збору, зберігання, обробки, та розповсюдження даних дуже сильно змінено з цим регламентом, наприклад:

1. Право на доступ. Кожна людина має можливість отримати свої данні, і в даному випадку мова йде не тільки про ті дані, які представив сам користувач, а й дані, які збирає той чи інший сервіс про нього. Також користувач має право знати: для якої мети його дані було зібрано, скільки ці дані будуть зберігатися, чи має він право на видалення свої даних тощо.

2. Право на уточнення. Користувач має право на змінення застарілих або недостовірних даних.

3. Право на компенсацію. Кожен користувач мережі інтернет має право на відшкодування збитку від втрати даних. Також сама організація має сплатити штраф за порушення GDPR [2].

Також є американський аналог, який було запущено в 2020 році. Називається він California Consumer Privacy Act, або CCPA. Це закон про захист персональних даних користувачів штату Каліфорнія, який на сьогодні є одним з найбільш регламентованих нормативних актів у сфері захисту персональних даних на території США [3].

Він розповсюджується саме на юридичні та фізичні особи саме в штаті Каліфорнія. За порушення правил, згідно CCPA, треба буде заплатити від 2500 до 7500 доларів, усе залежить від того, чи навмисно було порушення.

Розвиток йде дуже-дуже швидко, тому деякі користувачі просто не встигають зрозуміти, що треба зробити, щоб не віддати свої данні у руки зловмисників. Саме тому питання захисту даних в мережі інтернет є, мабуть, одним з найважливіших проблем сучасного світу. Кожного дня люди втрачають свої дані, чи за своєю провиною, чи за провиною розробників. Але треба розуміти, що наведені вище правила не зможуть захистити дані користувачів на 100%, але користувач зможе зробити усе можливе, що зловмиснику було дуже складно якось отримати про нього інформацію.

Тому, треба розповсюджувати інформацію, як захистити себе та користувачів додатків або сайтів, які використовують будь-які персональні дані.

***Список використаних джерел*:**

1. 5 шляхів захисту особистих даних в інтернеті. : Веб сайт. URL: https://beetroot.academy/blog/general/5-shlyahiv-zahistu-osobistih-danih-v-interneti (Дата звернення: 1.11.2021).

2. Что такое GDPR и какие требования предъявляет. : Веб сайт. URL: https://data-privacy-office.com/what-is-gdpr/ (Дата звернення: 1.11.2021).

3. California Consumer Privacy Act: новий тренд США із захисту персональних даних у 2020 році. : Веб сайт. URL: https://uz.ligazakon.ua/ua/magazine_article/EA013372 (Дата звернення: 1.11.2021).

Ключові слова: захист, персональні дані, користувач, розробник

Ключевые слова: защита, персональный данные, пользователь, разработчик

Keywords: security, personal data, user, developer

Науковий керівник: *к.т.н., доц. Задерейко О.В.*

Вогністий Олег Валентинович

Національний університет «Одеська юридична академія»,

студент 2-го курсу факультету кібербезпеки та інформаційних технологій

БЕЗПЕКА ПОПУЛЯРНИХ JAVASCRIPT ФРЕЙМВОРКІВ

Фреймворки JavaScript є своєрідним набором різноманітних інструментів для створення веб-застосунків, веб-сайтів як для десктопних, так і для мобільних пристроїв.

Станом на 2021 рік у світі серед веб-розробників найбільш популярними є такі фреймворки [1]:

1. React.js – 40.14%;
2. jQuery – 34.43%;
3. Express – 23.82%;
4. Angular – 22.96%;
5. Vue.js – 18.97%.

Кожен з них має унікальний функціонал, структуру, програмні модулі і компоненти. Також у фреймворків є свої стандарти для забезпечення безпеки майбутніх проектів. Але не зважаючи на те, що ці стандарти відрізняються один від одного, види порушень безпеки є доволі одноманітними.

Найпоширенішим видом загрози є міжсайтовий скриптинг або XSS-атака. Такого роду атаки дозволяють хакерам внести на веб сторінку JavaScript-код, який буде виконуватися щоразу, коли користувачі заходитимуть на цю сторінку [2]. Отримавши файли cookie, хакер матиме змогу отримати особисті дані користувача, наприклад: контактні дані, дані кредитної картки, дані авторизації (паролі, електронна пошта тощо).

Іншим видом загрози є SQL-ін’єкції. Вдалі SQL-ін’єкції надають доступ до бази даних, в яких хакери можуть змінювати або видаляти дані без відома користувача [3]. Завдяки цьому можна, наприклад, підробити якісь документи, змінивши в них дані, або зробити їх непридатними для використання.

Останнім з поширених видів порушень безпеки є підробка міжсайтових запитів (CSRF – Cross-site request forgery). Ця загроза дає змогу хакеру спонукати користувача до виконання певних дій, які не є обов'язковими, але на які може «клюнути» користувач-жертва. Наприклад, це може бути повідомлення про те, що треба змінити електронну пошту або оновити пароль, і в разі, якщо користувач перейде на сторінку із відповідним запропонованим запитом, хакер отримає контроль над його обліковим записом.

Як свідчить статистика [1], нині найбільш популярним фреймворком є React.js. Проте цей фреймворк не має стандартних налаштувань безпеки. Хоча деякі бібліотеки та компоненти, можуть допомогти ліквідувати загрозу, але цього замало. Тому веб-розробникам треба власноруч створювати механізми для захисту даних у своїх проектах, адже React.js уразливий до всіх видів загроз розглянутих вище. За часів існування даного фреймворка створено численні інструкції та рекомендації щодо захисту веб-сайтів чи веб-застосунків. Так, для захисту від XSS-атак є рекомендовано[4]:

1. використовувати брандмауер веб-застосунків у коді програми;
2. використовувати спеціальні інструменти для сканування вбудованих програм на наявність XSS-вразливостей;
3. реалізувати бібліотеки сніпетів, як-от: ES7 React, Redux тощо;
4. уникати написання коду, в який хакери потенційно можуть вставити інструкції для запуску шкідливих сценаріїв, наприклад, виключити HTML-теги: <script>, <object>, <embed> і <link>;

Такі заходи протидії загрозам себе виправдовують, але це не означає, що програми, написані у React.js, будуть захищені на 100%. Адже для цього треба доволі часто протягом всього життєвого циклу проекту тестувати програмний код на визначення рівня захищеності.

На відміну від React.js, фреймворк Angular вважається більш захищеним, завдяки наявності системних методів для забезпечення безпеки проектів. Angular підтримує серверну автентифікацію та авторизацію, що значно мінімізує виникнення можливих атак [5]. Оскільки стандартні API-інтерфейси DOM браузера не мають змоги захистити від атак, при розробці доцільно

використовувати шаблони Angular та уникати прямої взаємодії з DOM[6]. Завдяки потужності і захищеності цей фреймворк часто використовують для створення веб-сайтів або веб-застосунків для великих бізнес-компаній або для таких, які тісно взаємодіють із користувачем. У документації Angular є інструкції того, як уникати тих чи інших загроз для запобігання хакерських атак.

Фреймворк Vue.js переважно використовується для створення невеликих проектів, де потрібна швидкість. Через те, що Vue.js має порівняно слабку захищеність і від атак типу XSS, SQL-injection навряд чи захистить. Тому для забезпечення максимального захисту проекту, веб-розробникам у Vue.js, так само як і розробникам, що використовують React.js, треба самостійно встановлювати необхідні параметри для контролю вхідних і вихідних даних. Розробники у Vue можуть дезінфікувати HTML-код перед реалізацією або використовувати зовнішні бібліотеки для захисту від атак. Якщо HTML безпечний, можна відображати вміст HTML і захищати програму як до, так і після рендерингу [7].

Отже, безпека веб-сайтів та веб-застосунків при використанні фреймворків лише на 50% залежить від самого фреймворка. Якщо при розробці не приділити достатньо уваги саме забезпеченню безпеки проекта, то він автоматично себе сам захистити не зможе, і всі зусилля по його розробленню зійдуть нанівець. Для тих, хто має намір стати веб-розробником, треба ретельно вивчати і залучати інструменти захисту проекту від можливих загроз. А користувачам потрібно не поспішати клікати на всі підряд кнопки задля того, щоб не запустити вірус до своїх даних.

Список використаних джерел

1. Most used web frameworks among developers worldwide, as of 2021. URL: https://www.statista.com/statistics/1124699/worldwide-developer-survey-most-used-frameworks-web/

2. Что такое XSS-уязвимость и как тестировщику не пропустить ее. URL: https://habr.com/ru/post/511318/

3. Веб-безопасность. URL: https://developer.mozilla.org/ru/docs/Learn/Server-side/First_steps/Website_security.

4. Руководство по безопасности React.js: угрозы, уязвимости и способы их устранения. URL: https://bestprogrammer.ru/izuchenie/rukovodstvo-po-bezopasnosti-react-js-ugrozy-uyazvimosti-i-sposoby-ih-ustraneniya.

5. Angular. FAQ. URL: http://angular-doc.herokuapp.com/misc/faq.

6. Лучший JavaScript-фреймворк 2021: React или Vue? URL: https://evrone.ru/react-vs-vue.

7. Наилучшие практики угловой безопасности. URL: https://www.cisin.com/coffee-break/ru/technology/angular-security-best-practices.html

Ключові слова: безпека вебсайтів, SQL-ін’єкції, міжсайтовий скриптинг.

Ключевые слова: безопасность веб-сайтов, SQL-инъекции, межсайтовый скриптинг.

Keywords: website security, SQL injection, cross-site scripting.

Науковий керівник: *к.т.н., доц. Трофименко О. Г.*

Дрига Артем Вадимович

Національний університет «Одеська юридична академія», студент 2-го курсу факультету кібербезпеки та інформаційних технологій

РОЗГОРТАННЯ ОПЕРАЦІЙНИХ СИСТЕМ ЗА ДОПОМОГОЮ CLONEZILLA

Комп’ютерні класи та аудиторії потребують постійного технічного обслуговування, зокрема періодичної перестановки операційних систем. Це може зайняти багато часу та ресурсів через те, що процес встановлення операційної системи "з нуля" це тривала процедура, що вимагає присутності

людини-оператора, що відповідає на питання установника. Крім того, справа зазвичай не обмежується встановленням і операційна система вимагає налаштування, до встановлення програмного забезпечення користувача тощо. В такому випадку потрібен інструмент, який зможе розгорнути операційну систему та програмне забезпечення на всі пристрої за мінімальний період.

При розгортанні та заміні операційних систем у комп'ютерних класах Факультету кібербезпеки та інформаційних технологій було обрано програму Clonezilla за її універсальність та простоту реалізації.

Clonezilla - безкоштовне програмне забезпечення з відкритим початковим кодом, що призначене для клонування розділів жорсткого диску, а також резервного копіювання системи в разі її екстреного відновлення. Вся інформація, що була клонована, може бути збережена як у форматі ISO образа, так і у форматі дубльованої копії даних. Також інформація може бути збережена двома способами:

- На локальній машині;
- Віддалено, наприклад, на SSH сервері. [1]

Для розгортання систем у класах було обрано перший варіант. Спочатку було встановлено операційну систему на один з комп’ютерів класу. Далі, використовуючи ту обставину, що комп'ютери в аудиторіях були однотипними за комплектацією та складом апаратного забезпечення, було зроблено копію Ubuntu для подальшого розгортання на весь комп’ютерний парк навчального корпусу.

Таким чином, було отримано USB-накопичувач з налаштованою операційною системою та вже встановленим необхідним програмним забезпечення.

Для розгортання образу використовувалася така послідовність дій. Було підготовлено два носія інформації (дві порожні флешки, причому одна з них на 32 гігабайт), та комп’ютер, з якого буде копіюватися операційна система та пристрої, на які вона буде встановлена.

1. Образ Clonezilla було загружено з офіційного сайту (https://clonezilla.org/downloads.php).

2. Додатково було встановлено Tuxboot – саме за допомогою неї буде зроблено завантажувальну флешку з цим образом;

3. Було встановлено все необхідне програмне забезпечення на диск, що буде скопійований для подальшого завантаження на пристрої;

4. Після підготовки носія був налаштований біос використовуваного комп'ютера, в якому був виставлений пріоритет завантаження потрібного носія;

5. При перезапуску операційна система завантажилася з підготовленого носія та увійшла до оболонки системної утиліти Clonezilla.

Робота в середовищі оболонки системної утиліти Clonezilla здійснювалася за таким алгоритмом:

1. В меню завантаження треба обрати пункт Clonezilla live, потім обираємо мову, на якій нам буде зручно продовжувати завантаження, натискаємо кнопку "Don’t touch keymap" і після неї - "Start Clonezilla";

6. Тепер треба вибрати з двох опцій – "device-image" та "device-device". Різниця полягає в тому, що при першому варіанті копія буде збережена як образ, а при другому – файли будуть скопійовані відразу на інший диск. Так як нам знадобиться образ в майбутньому, обираємо першу опцію;

7. В наступному меню потрібно обрати, куди буде збережений образ. Обираємо опцію "local_dev", тобто збереження на USB-накопичувач. Після цього вставимо флешку на 32 гігабайти;

8. На наступній вкладці необхідно буде вибрати диск, на який ми зберігатимемо образ. Обираємо нашу флешку;

9. Далі обираємо каталог USB-накопичувача, до якого буде скопійовано образ. Копіюємо в кореневий каталог;

10. В меню вибору режиму роботи Clonezilla обираємо пункт "Beginner";

11. Так як ми хочемо скопіювати весь диск, в наступному меню натискаємо кнопку "savedisk". Після цього даємо назву образу та обираємо той диск, що потрібно скопіювати;

12. Обираємо пункти "Skip checking/repairing source file system", "No, skip checking the saved image" та "Not to encrypt the image";

13. Обираємо пункт "Reboot", щоб відразу перезавантажити систему після закінчення копіювання. Залишилось почекати, коли образ буде готовий.

Тепер можна перенести отриманий образ операційної системи на інші машини. Для цього знадобляться ще декілька флешок, після чого виконується наступна послідовність:

1. Розкладаємо образ операційної системи на флешки;
2. Окремо запускаємо "Clonezilla lite-server". Для цього виконуємо всі шаги, що й при запуску Clonezilla (зміна пріоритетів в BIOS, вибір мови тощо), але в меню вибору режиму роботи Clonezilla вибираємо меню "lite-server";
3. Далі обираємо пункт "start", далі "netboot", "auto-detect" та "local-dev";
4. Тепер серед пристроїв обираємо будь-який диск та будь-яку директорію в ньому;
5. Обираємо пункт "Beginner" та після нього "interactive-client";
6. Натискаємо на опції "from-image", "restoredisk", вибираємо будь-який диск та обираємо "reboot", "multicast", та "clients+time-to-wait", закінчуємо налаштування сервера;
7. Після того, як lite-server стартував, на клієнтській машині включаємо завантаження по мережі BIOS. Вставляємо флешку. Завантажуємось по мережі;
8. По мережі до нас прилітає образ Clonezilla - вибираємо його із флешки та завантажуємо його на машину.

Використання Clonezilla дозволило значно заощадити час та розподілити людський ресурс на інші важливі задачі. Також цей образ в майбутньому можна використовувати для відновлення або переустановлення систем, що вийшли з ладу.

Список використаної літератури:

1. Офіційний сайт Clonezilla. URL: https://clonezilla.org/

Ключові слова: Clonezilla, операційна система, образ системи.

Ключевые слова: Clonezilla, операционная система, образ системы.

Keywords: Clonezilla, operating system, system image.

Науковий керівник: *к.т.н., доц. Бойко В.Д.*

Ігнатенко Анастасія Ігорівна

Національний університет «Одеська юридична академія», студентка 2-го курсу факультету кібербезпеки та інформаційних технологій

ТИПИ DDoS-АТАК НА ІНТЕРНЕТ-РЕСУРСИ

DDoS-атаки проводять, коли хочуть, щоб інтернет-ресурс тимчасово перестав працювати або суттєво зменшив швидкість своєї роботи. Це популярний спосіб паралізувати роботу онлайн-бізнесу: поки інтернет-ресурс недоступний, відвідувачі будуть співпрацювати з потенційними конкурентами або чекати відновлення його нормальної роботи. В результаті бізнес втрачає прибуток, а іноді і власну репутацію.

DDoS-атаку можна порівняти із пробкою на дорогах. Коли рух перевантажений, ніхто не може рухатися швидко, але всі рухаються повільно. Теж саме відбувається і в мережі, коли інтернет-ресурс не може обробляти всі запити користувачів швидко чи взагалі зупиняє їх обробку.

Здійснення DDoS-атаки використовує велику кількість комп'ютерів, які можна орендувати або заразити шкідливим програмним забезпеченням. У останньому випадку комп'ютер користувача може ненавмисно брати участь в DDoS-атаці [1].

DDoS-атака може бути орієнтована на певні компоненти мережі. Більшість атак типу Denial of Service "відмова в обслуговуванні" націлені на наступні три рівні:

- Мережевий рівень - атаки на цьому рівні включають атаки Smurf, повені ICMP та атаки фрагментації IP/ICMP;
- Рівень передачі - атаки включають, серед іншого, SYN-флуд, UDP-флуд та атаки на TCP-порт.

- Рівень програми - атаки із шифруванням HTTP.

Різниця DDoS-атак до вимог в обслуговуванні

1. *Атаки на TCP-з'єднання*. Атака на TCP-з'єднання встановлює з'єднання із сервером, що обслуговує інтернет-ресурс, але зловмисники навмисно залишають з'єднання відкритим, а сервер може очікувати. Передбачається, що сервер зможе отримувати інші запити. Зловмисник продовжує робити аналогічні запити до сервера, перевантажуючи його і зрештою інтернет-ресурс перестає відповідати на запити будь-яких користувачів.

2. *Об'ємна атака*. Найбільш поширена атака відмови в обслуговуванні називається об'ємною атакою. Зловмисники мають намір зробити так багато одночасних запитів до серверу, використовуючи, наприклад, велику кількість ботів на заражених комп'ютерах, що сервер не може обробляти запити і зрештою відмовляє в обслуговуванні користувачів.

Цей тип атак часто реалізується, наприклад, шляхом виконання численних запитів до відкритого DNS-сервера. Передбачається, що DNS-сервер зрештою надішле надто багато даних і не зможе їх обробити.

3. *Атака фрагментацією*. Атака фрагментації використовує переваги фрагментації IP пакетів, відправляючи на сервер підроблені пакети даних, які перенавантажують сервер. Ця атака використовується в мережах передачі даних, в яких використовується технологія міжмережевого екрану на основі пакетів фільтрації.

4. *Атаки на рівні додатків.* Атаки на рівні додатків - найпростіший із типів атак. Вони виконуються рівні імітації дій користувачів, тому інтернет-ресурсу «здається», що користувачі постійно перезавантажують одну й ту ж сторінку або їх велику кількість. Ці атаки зазвичай більш ефективні та їх важче виявити [2].

Типи посилених DDoS-атак

У рамках посиленої DDoS-атаки зловмисники націлені на вразливість у DNS-серверах. Назва «посилена DDoS-атака» походить від того факту, що невеликі запити перетворюються на великі, що обмежує смугу пропускання сервера і зрештою зупиняє процеси на цільовому сервері. Існує два різних типи посилених DDoS-атак: відображення DNS та відображення CharGEN.

Відображення DNS. Атака з відображенням DNS копіює IP-адресу сервера та надсилає запити на DNS-сервер, а також запитує великі відповіді. У разі такої атаки, може йтися про відповіді, які в 70 разів перевищують нормальний розмір, що швидко завдає шкоди інтернет-ресурсу або обслуговуючому його серверу.

Відображення CharGEN. CharGEN протокол має ряд вразливостей, що надає можливість виконувати атака CharGEN Reflection, коли зловмисник відправляє на сервер безліч невеликих пакетів даних. Зрештою, зловмисник перевантажує сервер відповідями UDP, що призводить до тимчасового виходу його з робо спроможного стану [2].

Які збитки завдають атаки?

Наслідки відмови серверів в обслуговуванні інтернет-ресурсів можуть мати серйозні наслідки будь-якого суб'єкта цифрового простору. Важко підрахувати прямий ефект, коли ціна обумовлена, наприклад, втраченим бізнесом, усуненням пошкоджень та можливими судовими витратами. Згідно з звітом Corero NetWork Security за 2018 рік, компанії, що спеціалізується на запобіганні DDoS-атак, типова DDoS-атака може коштувати компанії, загалом до 50 000 доларів. Для більших компаній сума може бути ще вищою.

Проведення DDoS-атаки не є безкоштовним. Багато кіберзлочинців побачили тут можливість для бізнесу та пропонують бажаючим заказати DDoS-атаку. Наприклад, ціна атак типу "відмова в обслуговуванні", що продаються в темній мережі, варіюється в залежності від того, як довго триває атака і наскільки велика вона [4].

Законодавчі аспекти DDoS-атак

Атаки типу «відмова в обслуговуванні» вважаються незаконними за законами багатьох країн, і в майбутньому вони будуть визначені як незаконні за законами багатьох інших країн. На відміну від спуфінгу, наприклад, окремі компоненти атаки типу «відмова в обслуговуванні» можуть бути незаконними.

Атака відмови у обслуговуванні може порушувати закони кількох країн одночасно. У США, наприклад, атака типу «відмова в обслуговуванні» вважається федеральним злочином, а у Великій Британії за DDoS-атаку можна отримати до 10 років в'язниці. Звісно, треба відшкодувати нанесені збитки [5].

Можливості відстеження DDoS-атаки

Відстеження атак типу «відмова в обслуговуванні» може бути важким. Проблема в тому, що вони використовують велику кількість комп'ютерних ботів, і буває складно знайти справжнього організатора.

Атаки відмови в обслуговуванні можуть бути ідентифіковані, коли вони відбуваються за допомогою різних інструментів безпеки. Однак можливо вже занадто пізно зупиняти їх на цьому етапі. Справді, багато компаній активно витрачають багато грошей на боротьбу з DDoS-атаками [5].

Запобігання DDoS-атаці "відмова в обслуговуванні"

Атаки типу Denial of Service «відмова в обслуговуванні» найчастіше використовуються для виведення з ладу інтернет-ресурсів та корпоративних служб. Однак у деяких ситуаціях, таких як онлайн-ігри, атака відмови в обслуговуванні також може бути використана проти користувача.

VPN приховує IP-адресу користувача, тому проти нього не може бути проведена пряма DDoS-атака.

Список використаних джерел:

1. Як це працює: DDoS. : Веб сайт. URL: https://ssl.com.ua/blog/what-is-ddos/ (Дата звернення: 1.11.2021).

2. Що таке DDOS-атаки? : Веб сайт. URL: https://aws.amazon.com/ru/shield/ddos-attack-protection/ (Дата звернення: 1.11.2021).

3. Які бувають DDoS-атаки і чому захищатися складніше з року в рік. : Веб сайт. URL: https://www.orange-business.com/ru/blogs/kakie-bivayut-ddos-ataki-i-pochemu-zaschischatsya-slozhnee-iz-goda-v-god (Дата звернення: 1.11.2021).

4. Corero – перша лінія захисту від DDoS атак. : Веб сайт. URL: https://ko.com.ua/corero_-_pervaya_liniya_zashhity_ot_ddos_atak_107497 (Дата звернення: 1.11.2021).

5. Атака відмови в обслуговуванні. : Веб сайт. URL: https://uk.wikitechpro.com/858843-ddos-GBFMYW (Дата звернення: 1.11.2021).

Ключові слова: DDoS-атаки, DNS-сервера, IP-адресу, запобігання, відстеження**,** законодавчі аспекти.

Ключевые слова: DDoS-атаки, DNS-сервера, IP-адрес, предотвращение, отслеживание, законодательные аспекты.

Keywords: DDoS-attacks, DNS-servers, IP-address, prevention, tracking, legislative aspects.

Науковий керівник: *к.т.н., доц. Задерейко О.В.*

Іванова Ірина Олександрівна

Національний університет «Одеська юридична академія»,

студентка 3-го курсу факультету кібербезпеки та інформаційних технологій

ЗАБЕЗПЕЧЕННЯ ЗАХИСТУ БАЗ ДАНИХ

Актуальність теми обумовлена зростанням використання інформаційних баз у державних установах, корпораціях та підприємствах. Вторгнення, викрадення або пошкодження яких призводить до руйнівних наслідків у фінансовій, репутаційній, а інколи й в юридичній сферах.

Можна виокремити декілька основних причин які привертають увагу до безпеки даних та змушують використовувати більше ресурсів задля забезпечення безпеки БД.

Насамперед це кіберзлочинність, засоби якої постійно удосконалюються, з’являються нові способи проникнення до систем, створюється нове шкідливе програмне забезпечення та віруси.

По-друге, це проблема юридичної відповідальності. Права людини в міжнародному законодавстві про захист персональних даних являються одним з основоположних в інформаційному просторі, а отже постійно змінюються і стають все більш суворими. Відповідальність за недотримання нормативних

вимог, щодо збереження конфіденційних даних покладається на організації, котрі отримують її в ході діяльності.

Процес захисту систем баз даних охоплює сукупність програмних засобів, процесів та технологій, використання яких запобігає несанкціонований доступ, модифікацію, порушення цілісності, знищення або копіювання [1].

Захист будь-якої інформаційної системи, в такому випадку БД має бути багаторівневим, а отже діяти комплексно. Система безпеки не повинна обмежуватись лише базою даних, адже при доступі через мережу, будь-яка вразливість компоненту мережної інфраструктури, робочої станції або пристрою користувача буде являтись загрозою для інформаційної системи. Високорівнева система захисту повинна складатись з:

- Безпеки на фізичному рівні. Незалежно від розміщення бази даних(на локальному сервері або в хмарному сховищі) це має бути захищене місце, з контролем фізичного доступу до обладнання.
- Адміністративного контроль надання прав доступу. Користувачам повинні надаватись мінімальні права доступу, згідно з задачами які вони виконують.
- Безпеки пристроїв та облікових запитів. Забезпечення моніторингу доступу до БД та дій з даними. Контроль даних дозволяє вчасно отримувати інформацію про нетипові або підозрілі маніпуляції з даними. Кожний пристрій який під’єднано до мережі інформаційної системи, має бути захищений фізично.
- Шифрування та токенізації (технологія, що дозволяє забезпечити електронні платежі за допомогою системи шифрування даних). Дані в БД та ідентифікаційні дані користувачів, повинні бути захищені в процесі передачі та під час зберігання шляхом шифрування. Для захисту особливо конфіденційних даних бажано використовувати токенізацію.
- Безпеки програмного забезпечення. Необхідно вчасно встановлювати останні версії ПЗ, оскільки при кожному оновленні знешкоджується цілий ряд помилок та потенційних загроз.

- Безпеки веб-серверу. Необхідно забезпечити безперервне тестування безпеки серверу, який взаємодіє з БД, адже він може стати каналом проведення атаки.
- Безпеки резервних копій. Кожна копія або образ БД мають бути захищені на такому ж рівні, як і сама інформаційна система.
- Аудиту. Реєстрація кожної виконаної дії або зміни на сервері, в операційній системі та базі даних. Проведення регулярного аудиту щодо забезпечення кібербезпеки [2].

Окрім, формування багаторівневої системи захисту в мережі, необхідно організувати контроль та безпечний доступ до бази даних, шляхом впровадження: адміністративного контролю (керування змінами та конфігураціями БД), профілактичний контроль (управління доступом та шифруванням), виявлення (запобігання несанкціонованого доступу до інформаційної системи).

Політики інформаційної безпеки, які регулюють інформаційну діяльність в організації задля цілісного забезпечення та підтримки інформаційної безпеки БД, повинні бути успадковані хмарним середовищем. Крім того, мають бути визначенні відповідальні особи за обслуговування та аудит заходів безпеки. Формальні політики безпеки повинні бути підкріплені програмним навчанням, підвищенням обізнаності користувачів у сфері кібербезпеки, визначенням стратегій тестування на можливість проникнення та оцінка вразливостей системи [3].

Список використаних джерел:

1. Machine Learning and Security: Protecting Systems with Data and Algorithms / C. Chio, F. David., 2018. 386 с.

2. Database Security Best Practices: The Essential Guide 2021. URL: https://securityintelligence.com/articles/database-security-best-practices-essential-guide/

3. Consumer Privacy and Data Protection / Daniel J. Solove, Paul M. Schwartz., 2020. 422 с.

Ключові слова: база даних, інформаційна система, безпека.

Ключевые слова: база данных, информационная система, безопасность.

Keywords: data base, information system, security.

Науковий керівник: *к.т.н., доц. Кухаренко С.В.*

Кравченко Іван Сергійович

Національний університет «Одеська юридична академія»,

студент 3 курсу факультету кібербезпеки та інформаційних технологій

МЕТОДИ ЗАХИСТУ ВІД DDOS-АТАК

Мета атаки DDoS (Distributed Denial of Service) – домогтися відмови в обслуговуванні пристроїв які спроможні з’єднуватись з Інтернетом: обладнання мережевого контролю, різних Інтернет-магазинів, веб-сайтів та веб-додатків, інфраструктури Інтернету речей. Найбільша кількість атак у мережі розгортаються наступними кроками:

1) збір даних про цільовий об’єкт та їх аналіз з метою виявлення найбільш видимих та уразливих недоліків системи, вибір способу атаки;

2) націлюватися до атак шляхом підготовки зловмисного коду на комп'ютерах і всіх підключених до мережі Інтернет пристроях, які були перехоплені та встановлені під контроль зловмисника;

3) створення багаточисельних запитів з більшості підконтрольних пристроїв;

4) підведення підсумків атаки.

Якщо бажаний результат не був виконаний, в багатьох випадках буде проведений повторний та більш якісний аналіз, і знову повторення першого кроку.

При умовах успішної атаки на WEB-додаток, або базу даних, цільовий

ресурс буде демонструвати суттєве зниження продуктивності, або зовсім втратить можливість обробляти різні типи запитів від користувачів та серверу. Наслідками цього може бути довгі зависання ресурсу, звичайні користувачі не можуть скористатися цим ресурсом, додатком, або базою даних у потрібний для них момент. Це стає через неможливість доступу до Інтернету, та некоректно працювати. Наміри зловмисників бувають дуже різноманітними. Частіше це є недобросовісна конкуренція, або замовлення колишнього бізнес партнеру, також це може шантаж, конфлікт інтересів, поглядів, особисті переконання. Один з варіантів цих дій може бути переконання у навичках злому, так зване тренування. Також доволі часто недоброчесні програмісти використовують це з метою додаткового заробітку [1].

Для захисту крім технічних рішень, потрібно зробити дії з налаштуваннями проти DDoS-атак. Першим способом є зведення до мінімуму розміру зони яка може бути атакована. Потрібно переконатися що доступ до портів, додатків, протоколів, який не передбачений, є закритим. Основними елементами нейтралізації DDoS-атак є такі терміни як пропускна можливість і продуктивність серверу, яка достатня для поглинання атаки, правильного налаштування маршрутизатора та брандмауера. Також постійний аналіз аномалій в мережевому трафіку, і кожен раз коли є велике збільшення трафіку, при звертанні до хосту, можна використовувати максимально можливий обсяг трафіку, який хост може обробити без втрати доступності. Більш новітні методи захисту з новими можливостями дозволяють встановлювати обмеження на вхідний трафік. Для цього і подібних до цього засобів потрібно швидкі можливості Інтернету.

Проти атак, які дозволяють зловмиснику використовувати вразливість у ресурсі, наприклад проти впровадження SQL-коду чи злоякісного втручання до запиту, рекомендується використовувати Web Application Firewall. Через несхожість атак, треба самостійно нейтралізувати заборонені запити до баз даних, WEB-сервісів або інших додатків, які можуть надходити від підозрілих IP-адресів, з інших географічних регіонів [2].

Список використаних джерел:

1. Distributed Denial of Service. URL: https://stormwall.pro/knowledge-base/termin/ddos-protection

2. DDoS-атаки/ URL: https://aws.amazon. com/ru/shield/ddos-attack-protection/

Ключові слова: DDoS-атаки, ресурси, мережевий контроль, трафік, аналіз, злом, несанкціонований доступ.

Ключевые слова: DDoS-атаки, ресурсы, сетевой контроль, трафик, анализ, взлом, несанкционированный доступ.

Keywords: DDoS-attack, resource, network control, analysis, traffic, hack, unauthorized access.

Табала Ксенія Андріївна

Національний університет «Одеська юридична академія»,

студентка 2-го курсу факультету кібербезпеки та інформаційних технологій

ЦИФРОВІ ВІДБИТКИ ВЕБ-БРАУЗЕРІВ

Фінгерпрінтинг - це унікальний ідентифікатор, який створюється з застосуванням технології створення цифрового «відбитка» пристрою комунікації користувача. Ця технологія базується на аналізі інформації, яку веб-браузер відправляє на Інтернет-ресурси, якій користувач відвідує.

На основі декількох типів даних, які можна отримати завдяки запитуючому Інтернет-ресурсу, а саме: мовних налаштувань, встановлених шрифтів, часовому поясу, версії веб-браузера та його плагинів тощо - створюється унікальний «цифровий відбиток» веб-браузера. Завдяки використанню фінгерпрінтингу існує можливість розпізнати окремого користувача за створеним цифровим відбитком. Аналізуючи поведінку користувача веб-браузера в Інтернеті,

Інтернет-ресурси можуть використовувати отриману інформацію у комерційних, або інших цілях [1].

Чому створюються відбитки браузерів?

Відбитки веб-браузерів використовуються з метою запобігання шахрайства, а також «крадіжки особистості». Користувач Інтернету може навіть не підозрювати яку кількість інформації про себе він «віддає» Інтернет-ресурсам, без будь-якої згоди на те. З підвищенням кількості зібраних даних формується цифровий профіль користувача, який містить: стать, вік, сімейний та фінансовий стани, інтереси, звички тощо.

Небезпека для користувача зі створенням цифрових відбитків полягає у:

- загрозі конфіденційності;
- присвоєнні глобального унікального ідентифікатору;
- регенерації шкідливих *cookies* та фіксації IP-адреси.

Загроза конфіденційності є головною причиною чому користувачу потрібно враховувати можливості технології фінгерпрінтинг. Ця технологія набагато підступніше, ніж файли *cookies*. Від відбитків веб-браузерів складніше захиститися тому, що неможливо дізнатися: їдеться збір даних користувача чи ні.

Фінгерпрінти як глобальний ідентифікатор полягають у тому, що відбитки веб-браузера роблять його власника відомим не лише на часто відвідуваних Інтернет-ресурсах, але і в різних цифрових платформах. Фіксуючи цілісну картину, яку Інтернет-ресурс отримує від веб-браузера, фінгерпрінтинг ідентифікує користувача навіть при змінах в налаштуваннях веб-браузерів. Цифрові відбитки можуть звести нанівець конфіденційність як ділової, так і особистої переписки користувача.

Фінгерпрінти як регенератор шкідливих *cookies* та фіксатор IP-адреси користувача полягає у тому, що цифровий відбиток веб-браузера може не лише відновити всю бібліотеку *cookies*, але й знайти користувача за його основними мережевими даними [1].

Способи захисту від фінгерпрінтингу в Інтернеті

Існує багато способів змінити відбитки веб-браузерів та зробити їх менш унікальними - оновлення веб-браузерів, використання типових плагинів, видалення нестандартних шрифтів з операційної системи тощо.

Система може «забувати» про веб-браузери окремої конфігурації. Це показує тест, в якому користувач відвідує експериментальний Інтернет-ресурс декілька разів, а потім повертається на нього через деякий час. Результати цього тесту не є надто точними. Унікальність веб-браузерів, з яких заходили на Інтернет-ресурс повторно через 2 – 2,5 тижні, знижувалась на 30% та більше. Вважається, що зміни наступають вже після 5 днів відсутності активності.

Ще кращий результат можна отримати при зміні часового пояса. Користувачі, які мешкають у різних часових поясах, виставляли неправильний час, а після відвідування Інтернет-ресурсу змінювали його на правильне. У користувача з найбільш віддаленим (від початкового показника) часовим відрізком унікальність знижувалась максимально [2].

Гарантовано дієвих засобів захисту від фінгерпрінтингу допоки не було знайдено, однак є загально прийняті заходи для радикального зниження унікальності веб-браузера. Досить ефективно відключення виконання Flash, Javascript та WebGL. Це набагато ускладнює визначення унікальності веб-браузера.

Засоби, ефективні проти *cookies*, можуть не працювати проти технології фінгерпрінтингу. Наприклад, режим «Інкогніто» у веб-браузері блокує виконання деяких небажаних сценаріїв, однак ніяк не змінює відбитки веб-браузера і навіть навпаки – його легше впізнати серед інших [3].

Робота з UMatrix

Набагато кращих результатів дозволяє отримати використання спеціалізованого плагіну uMatrix – інтерактивного блокувальника будь-яких запитів, що відправляє Інтернет-ресурс при зверненні веб-браузера до нього. Він дозволяє вибірково заборонити відправлення даних на домени, що викликаються при загрузці скриптів, фреймів, систем статистики при зверненні веб-браузера до Інтернет-ресурсу [4].

Гнучкість налаштувань цього плагіну дозволяє отримати користувачу повний контроль над підключенням його веб-браузера до всіх доменів, пов‘язаних з використанням поточного Інтернет-ресурсу. Користувач сам вирішує, з якими доменами здійснювати обмін приватною інформацією – тим самим він повністю контролює особисту конфіденційність.

Таким чином, можна констатувати, що технології створення відбитків веб-браузерів базуються на зборі великої кількості даних про пристрій комунікації користувача. Повністю захиститися від фінгерпрінтингу неможливо, але існує можливість мінімізації ймовірності ідентифікації користувача шляхом встановлення спеціалізованих плагінів, що дозволяють керувати виконанням сценаріїв (особливо Java та Flesh), які відвідуються на сторінках Інтернет-ресурсів.

Список використаних джерел:

4. Отпечатки браузеров fingerprints. Часть 1 : Веб сайт. URL: https://whoer.net/blog/ru/unikalnye-otpechatki-brauzera-fingerprints-chast-1/ (Дата звернення: 1.11.2021).

5. Отпечатки браузеров fingerprints. Часть 2. : Веб сайт. URL: https://whoer.net/blog/ru/unikalnye-otpechatki-brauzera-fingerprints-chast-2/ (Дата звернення: 1.11.2021).

6. Фингерпринт: что это такое и как это скрыть? : Веб сайт. URL: https://vc.ru/u/738105-adspower-browser/212732-fingerprint-otpechatok-brauzera-chto-eto-takoe-i-kak-ego-skryt (Дата звернення: 1.11.2021).

7. uMatrix. : Веб сайт. URL: https://addons.mozilla.org/ru/firefox/addon/umatrix/ (Дата звернення: 1.11.2021).

Ключові слова: відбитки браузерів, Інтернет-ресурси, система.

Ключевые слова: отпечатки браузеров, Интернет-ресурсы, система.

Keywords: browser fingerprints, Internet resourses, system.

Науковий керівник*: к.т.н., доц. Задерейко О.В.*

РОЗДІЛ 2
ІНСТИТУЦІЙНІ НА ПРАВОВІ МЕХАНІЗМИ ЗАБЕЗПЕЧЕННЯ КІБЕРБЕЗПЕКИ

Rychlik Andrzej
Lodz University of Technology
PhD, associate professor at the Institute of Information Technology,
Faculty of Technical Physics, Information Technology and Applied Mathematics
University of Technology and Humanities in Radom
PhD, associate professor at the Department of Computer Science and Teleinformatics
Faculty of Transport, Electrical Engineering and Computer Science

THE NEED TO AMEND THE ACT ON THE NATIONAL CYBERSECURITY SYSTEM IN POLAND

From August 28, 2018, the Act on the national cybersecurity system [2] is in force in Poland, implementing the Directive of the European Parliament and of the Council (EU) on measures for a high common level of security of network and information systems in the territory of the European Union (Directive 2016/1148) , the so-called the NIS Directive. [5] This act was adopted by the Polish parliament with an overwhelming majority of votes, although the work on it and public consultations were full of contradictions and doubts.

Figure 1 shows the architecture of the national cybersecurity system built on the basis of the applicable act. The national cybersecurity system aims to ensure cybersecurity at the national level, in particular:

- uninterrupted delivery of essential digital services,
- achieving a sufficiently high level of security of ICT systems used to provide these services.

Although two years have passed since the adoption of the law, a complete national cybersecurity system based on the architecture shown in the figure has not yet been

built. The act defines an incident as an event that has or may have an adverse impact on cybersecurity. Based on it we are obliged to report each incident to the appropriate Computer Security Incident Response Team. The Cybersecurity College is a consultative and advisory body on planning, supervising and coordinating the activities of CSIRTs, sectoral cybersecurity teams and competent authorities.

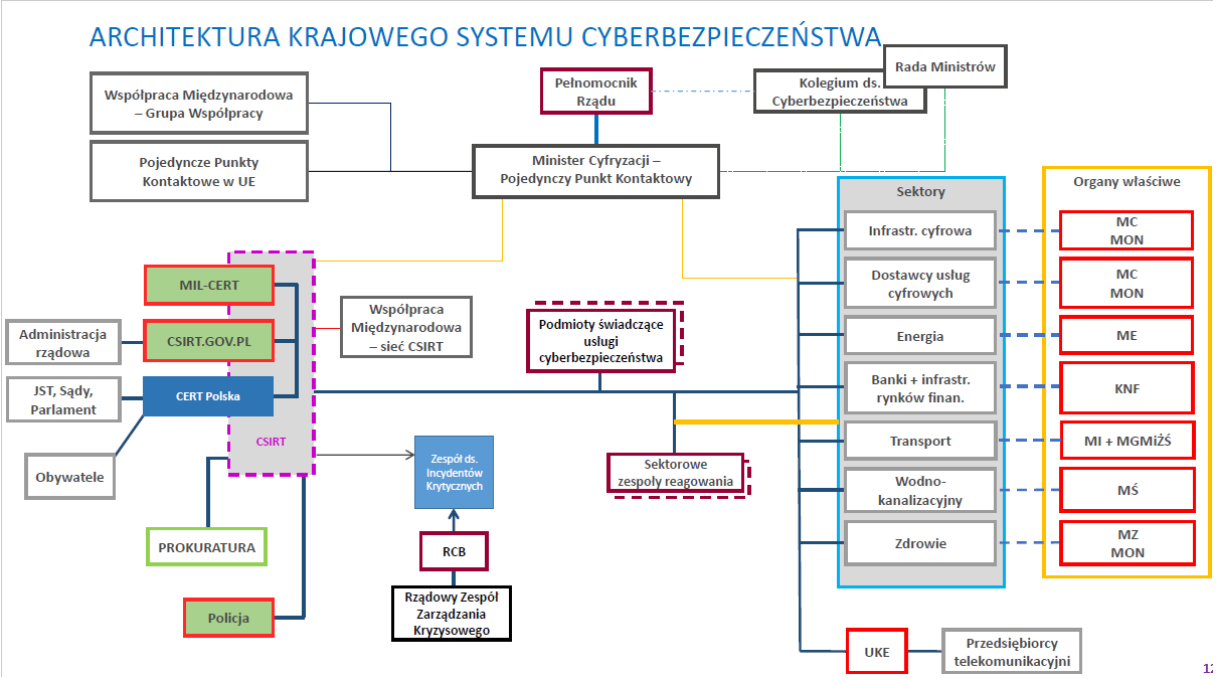

Fig. 1. Architecture of the national cybersecurity system [6]

Despite the statutory possibility, all sectoral cybersecurity teams have not been established so far. So far, only one such team has been established - the Polish Financial Supervision Authority's CSIRT for the financial sector at the Polish Financial Supervision Authority. The team was created based on internal resources and human resources of the Office of the Polish Financial Supervision Authority. In order to increase the effectiveness of responding to incidents, it is necessary to establish sectoral CSIRTs for each of the key sectors of the Polish economy. As a result, operators of essential services will be able to deal with incidents faster and more effectively, as they will receive direct support in responding to incidents. The College also issues opinions on cybersecurity requirements regarding the decisions of the President of the Office of Electronic Communications on frequency reservations. Due to the development of information and communication technologies, the transforming telecommunications infrastructure, the changing international political situation, user preferences, the

national cybersecurity system of 2018 did not stand the test of time and therefore needs to be reconfigured.

On July 24, 2020, the European Commission, with the support of ENISA and EU Member States, published the 5G Toolbox Implementation Report describing progress in implementing the EU toolbox and strengthening 5G network security measures. Toolbox was published in January 2020. It sets out a number of measures and actions to effectively reduce risk and ensure the implementation of secure 5G networks throughout Europe. 5G Toolbox provides standards-compliant features and reference examples for modelling, simulating and verifying 5G New Radio (NR) communication systems. Currently, Member States are gradually introducing these provisions.

On September 2, 2019, the US-Poland Joint Declaration on 5G was signed, in which we read: Taking into account that secure fifth generation wireless communications networks (5G) will be vital to both prosperity and national security in the near future, Poland and the United States declare their desire to strengthen our cooperation on 5G. Therefore, Poland and the United States endorse the Prague Proposals, the Chair's statement from the Prague 5G Security Conference, [3] as an important step toward developing a common approach to 5G network security and ensuring a secure and vibrant 5G ecosystem. Poland and the United States note that as part of a robust and comprehensive approach to network security, a careful and complete evaluation of 5G component and software providers is essential. [4]

On October 12, 2021, the government of the Republic of Poland noticed the delay of the legal system in the field of cybersecurity to the current ecosystem and sent a draft amendment to the act of July 5, 2019 to public consulations. The amendment made it possible to start work on the further development of the national cybersecurity system, and the experience gained over the two years of the system's operation in Poland indicated the need for changes at the statutory level. One of the most common problems is the lack of appropriate structures at operators of key services responsible for cybersecurity or the scope of their qualifications (including human resources) and the availability of information on cyber threats, which makes it difficult to effectively respond to incidents. The issue of access to expert knowledge on cyber threats is a key issue. The amendment also introduces certification in the field of cybersecurity.

The solutions adopted also serve to implement the Cybersecurity Strategy of the Republic of Poland for 2019-2024. A Cybersecurity Fund will be created from which activities aimed at ensuring cybersecurity in Poland will be financed. Resilience to cyber threats is highly dependent on the security of your hardware, software, and services. This applies to both ICT systems, telecommunications networks and industrial control systems. Therefore, the amendment provides for the introduction of a procedure for recognizing a supplier of hardware or software for entities of the national cybersecurity system as a high-risk supplier. Proceedings in this matter will be conducted by the minister responsible for computerization. The procedure will be based on transparent procedures specified in the Code of Administrative Procedure. It should be emphasized that the assessment of supplier risk profiles is one of the strategic tools (Strategic Measure - SM03) agreed by the Member States of the European Union, the European Commission and ENISA in the 5G Toolbox. [1]

The author proposes supplementing the governmental draft amendment with articles obliging operators of critical infrastructure to ensure energy independence for this structure. This is especially important in times of natural disasters, wars and revolutions. The second proposal is to ensure a high level of security of digital data transmission for the general public by educating it to use encryption and electronic signing of transmitted documents.

References

1. Commission Recommendation (EU) 2019/534 on the Cybersecurity of 5G networks, 26 March 2019

2. Ustawa z dnia 5 lipca 2018 r. o krajowym systemie cyberbezpieczeństwa Dz. U. z 2020 r. poz. 1369

3. The Prague Proposals, The Chairman Statement on cyber security of communication networks in a globally digitalized world, Prague 5G Security Conference

4. U.S.-Poland Joint Declaration on 5G, Signed in Warsaw, Poland, on September 2, 2019

5. Directive (EU) 2016/1148 of the European Parliament and of the Council of 6 July 2016 on measures for a high common level of security of network and information systems within the territory of the Union (Journal of Laws of the EU 2016 No. 194)

6. Szyszko A., Department of Safety and Crisis Management, Ministry of Energy, Meeting of the LTE working group

Ключові слова: кібербезпека, 5G Toolbox, інцидент, критична інфраструктура, сертифікація

Ключевые слова: кибербезопасность, 5G Toolbox, инцидент, критическая инфраструктура, сертификация

Keywords: cybersecurity, 5G Toolbox, incident, critical infrastructure, certification

Dykyi Oleh Viktorovych

National University «Odesa Law Academy»,

Dean of the Faculty of Cybersecurity and Information Technologies, Candidate of Law, Associate Professor

MODERN METHODOLOGICAL APPROACHES TO THE STUDY OF CYBER CRIMES

The XIX century was an era of total informatization and cybernation of all spheres of life in most countries. On the one hand, it has improved and simplified people's lives, and on the other, it has created new challenges, such as cybercrime.

In recent years, cybercrime has reached alarming proportions not only in Ukraine but in the world. This is largely due to the lack of proper international cooperation and the rapid processes that are taking place in the cyber environment. Every year, technologies change, new data systems are created, new software is used by criminals,

and so on. Of course, this requires adequate measures to respond to and prevent threats, which should be based on scientific and practical recommendations.

Given current trends, the United Nations General Assembly, in its resolution of 17 December 2018, expressed concern over the increase in cybercrime and the use of information and telecommunications technologies in the commission of many types of crime. In this regard, it recommended that Member States intensify their efforts to combat cybercrime and the criminal use of information and communication technologies in all its forms and to develop international cooperation in this field, including the exchange of electronic evidence [1].

No less important is the scientific support of combating cybercrime, which in the domestic doctrine is not sufficiently developed, despite the applied research of scientists. There can be many reasons for this, for example, the lack of clear boundaries of the subject of research, outdated methods of collecting and analyzing empirical material, ignoring the use of research results of scientists, especially from developed countries in Europe and the United States and more. As a result, the nature of cybercrime is distorted, simplified to cybercrime or in the field of information technology. Moreover, the lack of thorough criminological research on the cyber environment creates real problems in the practice of combating this type of crime in the context of international cooperation. Thus, the report on the twenty-fifth session of the Commission on Crime Prevention and Criminal Justice of the UN Economic and Social Council for 2016 stressed the importance of having adequate national legislation and intensifying international cooperation in the fight against cybercrime. Also, different opinions were expressed about the best approach to combating cybercrime at the international level [4]. For example, participants drew attention to the need to develop a new comprehensive international legal instrument on cybercrime, which will focus on, in particular, procedural issues. However, this issue is still unresolved, and the main international document is the Council of Europe Convention on Cybercrime, adopted in 2001.

Cyber criminology was first proposed as a scientific field in 2007 by Indian scientist K. Jaishankar. In his work, the scientist defines the definition of cyber criminology: «as the doctrine of causation in the commission of crimes occurring in

cyberspace and their impact on physical space» [2]. Of course, this definition does not reflect the essence of criminology in general and cyber criminology as its component, and therefore requires additional research through the prism of modern research.

In domestic scientific opinion, the term «cyber criminology» is almost non-existent. There are only descriptive mentions in Russian works, such as: «At the junction of computer science and criminology, a new scientific field of cyber criminology has appeared. Today, cyber criminology is a private criminological theory that studies crime in cyberspace (cybercrime), its causes, personality traits of cybercriminals and measures to combat this phenomenon» [3].

Obviously, cyber criminology is by nature a multidisciplinary field of study. It covers criminology, sociology, psychology, victimology, computer and Internet sciences. In addition, cyber criminology is based on the study of criminal behavior and victimization in cyberspace from a behavioral theoretical point of view, which on the one hand seems promising, on the other - extremely complex. This is primarily due to the empirical material. Thus, it is quite easy for scientists and practitioners to identify victims of cybercrime, but not criminals, and the available studies on the basis of convictions in criminal cases form an idea not of a professional cybercrime, but of an amateur or beginner. This is one of the important problems that cyber criminology needs to address in order to develop effective measures to combat this type of crime. It is cyber criminology that should become the field of knowledge that will allow a more detailed study of cyberspace in terms of violation of criminal law.

References:

1. Укрепление программы Организации Объединенных Наций в области предупреждения преступности и уголовного правосудия, в особенности ее потенциала в сфере технического сотрудничества. Принята резолюцией 73/186 Генеральной Ассамблеи ООН от 17 декабря 2018 года. *Организация Объединенных Наций : офіційний веб-сайт.* URL: https://undocs.org/pdf?symbol=ru/A/RES/73/186

2. Jaishankar K. Cyber criminology: Evolving a novel discipline with a new journal. *International Journal of Cyber Criminology*. 1(1). P. 1–6.

3. Шестаков Д.А., Дикаев С.У., Данилов А.П. Летопись Санкт-Петербургского международного криминологического клуба. Год 2014 // Криминология: вчера, сегодня, завтра. – 2015. – №1 (36). – С.73

4. Доклад о работе двадцать пятой сессии (11 декабря 2015 года и 23-27 мая 2016 года) Комиссии по предупреждению преступности и уголовному правосудию Экономического и Социального Совета ООН. *Организация Объединенных Наций: офіційний веб-сайт.* URL: https://undocs.org/pdf?symbol=ru/E/2016/30

Ключові слова: кіберкримінологія, кіберпростір, кіберзлочини.

Ключевые слова: киберкримология, киберпростир, киберзлочини.

Keywords: cyber criminology, cyber space, cybercrime.

Чанышев Рашид Ибрагимович

Национальный университет «Одесская юридическая академия», доцент кафедры информационных технологий, кандидат юридических наук, доцент

О НЕОБХОДИМОСТИ КОМПЛЕКСНОЙ ПОДГОТОВКИ СПЕЦИАЛИСТОВ ПО НОВЫМ ПРОФЕССИЯМ, СВЯЗАННЫМ С ЗАЩИТОЙ ИНФОРМАЦИИ

В последнее время во всем мире отмечается резкий рост числа инцидентов, связанных с информационной безопасностью. С учетом растущей цифровизации практически всех сфер жизнедеятельности человечества, эти угрозы стали представлять непосредственную опасность для жизни и здоровья людей [1].

Так, 4 октября 2021 года у пользователей ряда социальных сетей начались проблемы с доступом к Facebook, Instagram и WhatsApp. Неожиданным следствием этой проблемы стало то, что сотрудники компании Facebook не смогли попасть в собственный офис, а множество обычных пользователей не

смогли пройти авторизацию на сайтах, использующих в качестве средства авторизации учетные записи Facebook, Instagram и WhatsApp. Кроме того, персонал Facebook был лишен возможности оперативно устранить проблему из-за того, что указанные проблемы привели к тому, что удаленный доступ к серверам оказался невозможным.

Как выяснилось, глобальная проблема была вызвана всего лишь одной неудачной попыткой обновления программного обеспечения маршрутизатора [2].

В условиях, когда проблемы глобального характера могут быть вызваны даже не сознательными действиями злоумышленников, а простой технической ошибкой, вопросы подготовки специалистов по информационной безопасности становятся особенно актуальными. Ввиду острой необходимости принятия срочных мер по обеспечению кибербезопасности Украины, Государственная служба связи и защиты информации Украины (Госспецсвязь) предложила расширить список профессий, связанных со сферой обеспечения безопасности информации, защиты компьютерных систем и расследования компьютерных инцидентов и преступлений.

Министерство экономики Украины своим приказом от 25 октября 2021 года внесла изменения в Национальный классификатор профессий ДК 003: 2010 [3]. Среди прочих нововведений в Классификатор был включен ряд новых профессий, связанных со сферой защиты информации.

Ранее в Классификаторе были предусмотрены всего две профессии, связанные с защитой информации: специалист в сфере защиты информации и профессионал по организации информационной безопасности. С учетом современных тенденций этого было явно недостаточно.

В число новых профессий входят специалисты по анализу угроз информационным системам, по разработке, тестированию и поддержке новых систем защиты информации, специалисты по криптографической и технической защите информации, специалисты по реагированию на инциденты кибербезопасности и специалисты по разработке и контролю за соблюдением

принятых методик информационной безопасности и кибербезопасности. Все эти профессии можно отнести к категории инженерных специальностей.

Кроме того, в Классификаторе введены новые профессии, больше связанные с юридической, чем с инженерной сферой деятельности, такие, как профессии эксперта-криминалиста, дознавателя и следователя в сфере компьютерной безопасности и защиты информации.

Выступая на Международном форуме «Кибербезопасность. Защитим бизнес – защитим государство» заместитель главы Госспецсвязи Александр Потий отметил, что важнейшими задачами на данный момент является разработка квалификационных требований к каждой новой профессии, разработка систем квалификации и сертификации этих профессий, а также создание особых квалификационных центров, в которых выпускники высших учебных заведений смогут повысить свою квалификацию по указанным специальностям и сдать соответствующие экзамены.

В связи с этим Александр Потий предложил объединить усилия преподавателей высших учебных заведений и специалистов в области компьютерной безопасности в деле разработка новых профессиональных стандартов и методик обучения специалистов. [4]

Прямым следствием появления новых специальностей является необходимость организации подготовки специалистов соответствующего профиля. И если вопрос о подготовке специалистов инженерного профиля более-менее изучен и соответствующие методики обучения успешно внедрены в процесс обучения, то вопросы подготовки специалистов, работающих на стыке столь разнородных сфер деятельности как право и инженерия, требуют тщательной разработки.

Проблема заключается в том, что специалисты, обладающие знаниями в области программирования, системного администрирования и разработки электронных устройств, не обладают необходимыми знаниями в области права и наоборот.

Для обеспечения компьютерной безопасности необходимо учитывать три основных направления решения проблем, вызываемых умышленными

действиями злоумышленников, проблем, вызываемых неумышленными действиями персонала или пользователей, и проблем, связанных с работой аппаратно-программных систем.

Решение первой из указанных проблем неразрывно связано с расследованием компьютерных преступлений и с компьютерной криминалистикой (computer forensics). В свою очередь, в компьютерной криминалистике выделяется отдельное направление сетевой криминалистики (network forensics), задачей которой является осуществление анализа компьютерного сетевого трафика в целях обнаружения вторжений, сбора информации и юридических доказательств. [5]

Специалисты по компьютерной криминалистике, с помощью специального оборудования, осуществляют экспертное исследование компьютерного оборудования, например, мобильных телефонов, носителей информации, восстанавливая удаленную злоумышленниками информацию, проводят исследование программного обеспечения, использованного преступниками.

Расследование компьютерных преступлений связано с изучением криминальной практики – способов совершения преступлений, используемых при совершении преступлений технических средств, изучением и анализом оперативной, следственной и судебной практики расследования компьютерных преступлений, исследованием методов предотвращения компьютерных преступлений и минимизации их последствий, изучением специфических наклонностей лиц, занимающихся компьютерными преступлениями, особенностей их психологии.

Таким образом, специалисты по компьютерной безопасности работают на стыке права, компьютерных технологий, психологии и социологии.

В Национальном университете «Одесская юридическая академия» уже создана база и есть все возможности, позволяющие осуществлять комплексную подготовку специалистов по всем указанным направлениям и практически по всем новым специальностям в области компьютерной безопасности, включенным в Классификатор. Возможно, для такой подготовки потребуется включить в учебные планы новые дисциплины, например, компьютерную

криминалистику, объединить усилия преподавателей разных кафедр или даже создать новые кафедры.

Список использованной литературы:

1. Чанышев Р. И. Вирусы-шифровальщики (Ransomware), угрозы и меры противодействия / Р. И. Чанышев // Наука та суспільне життя України в епоху глобальних викликів людства у цифрову еру (з нагоди 30-річчя проголошення незалежності України та 25-річчя прийняття Конституції України) : у 2 т. : матеріали Міжнар. наук.-практ.конф. (м. Одеса, 21 трав. 2021 р.) / за загальною редакцією С. В. Ківалова. – Одеса : Видавничий дім «Гельветика», 2021. – Т. 1. – С. 583-587.

2. Understanding How Facebook Disappeared from the Internet [Electronic resource]. – URL: http://surl.li/apnzd. – Last access: 04.11.2021.

3. Про затвердження Зміни №10 до національного класифікатора ДК 003:2010: наказ міністерства економіки України від 25 жовтня 2021 року №810.

4. Розширено перелік професій у сфері кібербезпеки [Електронний ресурс]. – URL: http://surl.li/apnzc. – Дата звернення: 5.11.2021 р.

5. John Mounteer Network Forensics 101 [Electronic resource]. – URL: http://surl.li/apnzd. – Last access: 04.11.2021.

Ключові слова: інформаційна та комп'ютерна безпека, підготовка фахівців, класифікатор професій, комп'ютерна криміналістика, мережева криміналістика..

Ключевые слова: информационная и компьютерная безопасность, подготовка специалистов, классификатор профессий, компьютерная криминалистика, сетевая криминалистика.

Keywords: Information and computer security, training of specialists, classifier of professions, computer forensics, network forensics.

Щербина Юрій Володимирович

Національний університет «Одеська юридична академія»,

доцент кафедри інформаційних технологій, кандидат технічних наук, доцент

Єфремов Владислав Анатолійович

Національний університет «Одеська юридична академія»,

студент факультету кібербезпеки та інформаційних технологій

АКТУАЛЬНІ ПИТАННЯ ТА ПРОБЛЕМИ ЗАХИСТУ СУЧАСНОГО КІБЕРПРОСТОРУ

Втілення кіберфізичних систем (Cyber-Physical System, CPS) у сучасне виробництво та процеси, пов’язані з управлінням складними технологічними процесами, приводять до кардинальних змін у суспільстві. Вони будуються на базі природних об’єктів та інтелектуальних мереж (Smart Grid), що використовуються для підвищення ефективності автоматизації управління інфраструктурою енергопостачання, телекомунікаційних та оборонних систем та їм подібних [1]. CPS відносять до складу великих складних системах, у яких відбувається збір та оброблення великих потоків інформації, а також моніторинг стану технологічного обладнання. На жаль, крім переваг, що дає використання інтелектуальних технологій, цифровізація виробництва, суспільного життя, і бізнесу виникає віртуальний простір для скоєння кіберзлочинів. Із цього факту логічно витікає необхідність постійного проведення досліджень пов’язаних із реагуванням на події у середовищі експлуатації, що викликають необхідність відновлення ресурсів, та нормального стану інформаційних процесів.

Кіберзлочини, на відміну від інформаційних злочинів, мають на меті порушення нормального функціонування процесів управління інфраструктурними об’єктами, завдання економічних збитків конкурентам, внесення порушень у виробничі процеси або підготовку і скоєння терористичних атак.

Проблема попередження подібних дій ускладнюється надзвичайною розподіленністю, інтелектуальних інформаційно-телекомунікаційних мереж. Складність їх архітектури та апаратного забезпечення збільшує ймовірність потенційно можливих мережевих атак на критично важливі процедури управління та обчислювальні ресурси. Через це, очевидною є важливість постійного моніторингу ризиків у середовищі експлуатації кіберфізичних систем та своєчасне запобігання зовнішнім втручанням [2].

Оскільки наслідками кібератак можуть бути події, що утворюють загрози життю людей, виникненню техногенних катастроф а також великих матеріальних втрат, системи кібербезпеки повинні зменшувати ризики реалізації загроз, виявляти та ідентифікувати аномальну поведінку системи, а також оперативно відновлювати нормальний стан системи.

Актуальні атаки на кіберфізичні системи

Комп'ютерна злочинність виникла майже одночасно із появою відкритої колективної обчислювальної мережі та розподілених обчислювальних процесів. Зростання кола послуг, що Internet надає своїм користувачам лише ускладнює цю проблему. Наприкінці 1999 року у світ вийшли «Загальні критерії», прийняті в Україні як ДСТУ ISO/IEC 15408:2017 [3]. Разом з іншими суміжними документами, вони визначали термінологію та засади захисту інформації у відкритому мережевому просторі. Питання управління та організації захисту вперше були викладені у стандарті BS 7799, виданому Британським інститутом стандартів (British Standards Institution – BSI) під назвою «Практичні правила управління інформаційною безпекою» у 1995 році. Після 2000 року на його основі ISO/IEC JTC 1/SC 27 випускав міжнародні стандарти серії ISO/IEC 27000, де визначається модель та функції управління безпекою. Вони є результатом консенсусу, згоди і накопиченого досвіду експертів різних країн.

З урахуванням зміни, що відбулися загальному інформаційному просторі, ISO/IEC оприлюднило новий міжнародний стандарт ISO/IEC 27032:2012 «Інформаційні технології. Методи забезпечення безпеки. Настанови із кібербезпеки» [4]. У цьому стандарті було визначено такі поняття як кіберпростір та кібербезпека і надано їх тлумачення. Кіберпростір визначається

як комплексне віртуальне середовище, яке виникає під час взаємодії людей, програмного забезпечення та Internet-послуг, що підтримуються міжнародними розподіленими фізичними інформаційно-телекомунікаційними технологіями та мережами зв’язку. Не зважаючи на його “віртуальність” сплановані у ньому атаки на ресурси кіберфізичних систем можуть викликати економічний колапс, паралізувати оборонні структури, або значно знизити ефективність управління державними органами. Ще два десятиріччя тому головними задачами захисту були збереження конфіденційності та цілісності даних, то сьогодні мова іде про кібернетичну війну і кібернетичну зброю.

Одним із серйозних видів, актуальних на даний час кібернетичних атак, розглядається атака повторного відтворення (Replay attacks). Її ціллю є система автентифікації. Шляхом запису і подальшого відтворення коректного повідомлення або його частини [5] в систему вводяться хибні дані. Для імітації автентичності вона використовує будь-яку незмінну інформацію, наприклад, пароль або біометричні дані. Паралельно із цією атакою зловмисники використовують атаку червоточини (Wormhole attack). Вони перехоплюють інформацію між двома кінцевими точками і передають її іншим суб’єктам, створюючи таким чином “тунель” контролю.

Іншою, поширеною і небезпечною у кібернетичному просторі атакою, є атака, обману (Deception attacks). Її ціллю є втручання у фізичні і кібернетичні процеси через системи телекомунікації для отримання контролю над окремими її частинами [6]. Обман може бути визначений як взаємодія між двома суб'єктами – суб’єктом, що є джерелом атаки і суб’єктом-ціллю. Джерело атаки намагається змусити ціль прийняти неправдиву версію дійсності, бажану обманщикові.

На відміну від природного середовища, канали зв'язку у кіберпросторі несуть менше інформації, ніж канали при звичайній взаємодії суб'єктів, а інформація у кіберпросторі схильна до постійних змін. Обидва ці фактори сприяють реалізації шахрайських дій. Тому атаки обману не мають окремої типової моделі. Їх сценарії визначаються в залежності від поставлених цілей, вразливостей і наявних ресурсів порушників безпеки. У разі нападу на технологічні системи, основними цілями атак обману є маніпуляції із

показаннями датчиків, підробка керуючої інформації та отримання доступу до важливих ресурсів систем.

По мірі розвитку систем кібернетичного захисту будуть вдосконалюватися і сценарії нападу на критично важливі об’єкти кіберфізичних систем.

Сьогодні існує велика кількість методів виявлення і зупинення атак такого типу. Через це, для організації надійного захисту необхідним є постійний моніторинг мережевих подій, вивчення вразливостей системи та сценаріїв атак, що мали місце раніше, їх оцінка і пошук способів ефективної протидії а також створення відповідних експертних систем [7].

Протидія кіберзлочинності ґрунтується на створенні їх математичних моделей, які на формальному рівні описують сценарії мережевих атак, головною з яких є модель DoS-атаки, побудованої на основі схеми Бернуллі. Вхідними даними для неї є результати моніторингу вторгнень у кіберфізичний простір та накопичений досвід в організації протидії зовнішнім втручанням.

Складність проблеми пояснюється динамічністю поведінки складових частин кіберфізичних систем, а також тим, що передбачити можливі сценарії атак у кіберпросторі майже неможливо. Ця задача ще чекає свого ефективного вирішення. Вона не є цілком технічною, оскільки порушникам не завжди володіють всією необхідною для організації атак інформацією, вони вимушені користуватись ще й агентурними методами.

Виходячи із цього, захисники мереж зі свого боку повинні вирішувати проблему багатоцільової оптимізації. Це складна задача, але її вирішення у даний момент виглядає дуже важливим.

Список використаних джерел:

1. Janssen M.C. The Smart Grid Drivers, PAC World, 2010, 77 p.

2. National Institute of Standards and Technologies (NIST): ‘Guidelines for smart grid cybersecurity’ (NIST Special Publication, Gaithersburg, MD, 2014). Available at url: http://www.dx.doi.org/10.6028/NIST.IR.7628r1

3. ДСТУ ISO/IEC 15408-1:2017 Інформаційні технології. Методи захисту. Критерії оцінки. Частина 1. Вступ та загальна модель (ISO/IEC 15408-1:2009, IDT)

4. ISO/IEC 27032:2012 Information technology – Security techniques – Guidelines for Cybersecurity. – 2012-07

5. Dutt, V., Ahn, Y. S., & Gonzalez, C.: Cyber situation awareness modeling detection of cyber-attacks with instance-based learning theory. Human Factors: The Journal of the Human Factors and Ergonomics Society, 55(3), 605-618 (2013).

6. D. Ding, Z. Wang, Q.-L. Han, G. Wei, Security control for a class of discretetime stochastic nonlinear systems subject to deception attacks, IEEE Trans. Syst. Man Cybern.Syst. doi:10.1109/TSMC.2016.2616544.

7. Sridhar, S., Govindarasu, M.: 'Model-based attack detection and mitigation for automatic generation control', IEEE Trans. Smart Grid, 2014, 5, (2), pp. 580–591.

Ключові слова: кіберпростір, кіберфізичні система, кібератака, кібербезпека.

Ключевые слова: киберпространство, киберфизическая система, кібератака, кибербезопасность.

Keywords: cyberspace, cyberphysical system, cyberattack, cybersecurity.

Колосенко Андрій Андрійович

Національний університет «Одеська юридична академія»,

студентка 4-го курсу факультету кібербезпеки та інформаційних технологій

ІНСТИТУТ ДЕРЖАВНО-ПРИВАТНОГО ПАРТНЕРСТВА ЯК ОДИН З МЕХАНІЗМІВ ЗАБЕЗПЕЧЕННЯ КІБЕРБЕЗПЕКИ УКРАЇНИ

У часи невпинного технічного прогресу, діяльність держави, підприємств та організацій нерозривно пов'язана з інформаційно-комунікаційними технологіями.

Так, забезпечення безпеки та конфіденційності інформації, захищеності об’єктів критичної інфраструктури почало займати пріоритетне значення для будь-яких державних, так й приватних установ, в зв’язку з чим значних змін зазнали відносини між державною владою та приватним сектором економіки, а взаємодія між такими суб’єктами почала сприйматись як один з найбільш перспективних механізмів забезпечення кібербезпеки та національної безпеки держави в цілому.

Вказана взаємодія пояснюється тим, що в умовах сучасних реалій кібертероризм, кіберзлочинність та інформаційні війни змушують державу шукати та залучати приватні організації для того, щоб протидії вказаним явищам

З метою вирішення вищезазначених проблем в Україні набув розвитку інститут Державно-приватного партнерства (далі – ДПП), який фактично необхідно розуміти як сукупність спільних заходів державного та приватного сектору, з метою досягнення попередньо обумовлених сторонами цілей.

Сьогодні ДПП визнається майже усіма світовими країнами, а також приватними підприємствами, як ключовий елемент функціонування ефективної та прогресивної системи протидії кіберзлочинності та забезпечення кібербезпеки держави. Майже кожна стратегія кібербезпеки (державного або наддержавного рівня) або нормативно-правові акти державних структур містять положення про необхідність розбудови інституту державно-приватного партнерства.

Проте, вказаний механізм взаємодії держави та бізнесу в Україні потребує значних змін та рушійних дій, про що добре відомо й державній владі. Оновлена редакції Стратегії кібербезпеки, яка була прийнята 14 травня 2021 року, наголошує, що на даний момент в Україні відсутня дієва модель державно-приватного партнерства у сфері забезпечення кібербезпеки [1].

Однак, варто зазначити, що станом на сьогоднішній день, на українському ринку кібербезпеки вже функціонує достатня кількість підприємств, які мають великий досвід та технічні рішення в зазначеній сфері. Їх діяльність пов’язана з виявленням мережевих атак, розслідуванням кіберзлочинів, отриманням доказів під час обстеження комп'ютерних та інформаційних систем, проведенні комп’ютерних експертиз.

Такі компанії разом з державними органами безпеки та оборони безумовно можуть своєю взаємодією позитивно вплинути на забезпечення кібербезпеки України.

При цьому, можна виокремити наступні форми реалізації такої взаємодії: укладення договорів щодо надання послуг і виконання вищезазначених робіт, отримання консультацій по поточній діяльності, розробка спільних рішень та пропозицій щодо реалізації державної політики в сфері кібербезпеки, проведення спільних заходів з метою інформаційного забезпечення державних, приватних підприємств та громадян з питання дотримання заходів із забезпечення кібербезпеки тощо [2].

Крім того, ДПП може позитивно вплинути і на вітчизняний сектор економіки, оскільки такий механізм надає змогу залучити додаткові інвестиції у фінансування високобюджетних проектів.

Варто зазначити, що державою разом з приватним сектором проводяться ряд тематичних заходів в сфері інформаційної безпеки.

Так, з 2018 року Державна служба спеціального зв'язку та захисту інформації України та Комітетом електронних комунікацій ТПП України розпочато проведення щорічного заходу під назвою «Місяць кібербезпеки».

У жовтні 2019 та 2020 року він був проведений вдруге та відповідно втретє, участь в таких заходах взяли участь близько 100 компаній як з України, так й зі всього світу. Ключовими подіями заходу стали: Міжнародний форум «Кібербезпека. Захисти свій бізнес!», «LEGAL CYBERSECURITY FORUM», PKI-форум з проблематики електронного підпису та інфраструктури відкритих ключів.

Подібні заходи присвячені підвищенню обізнаності приватного сектору і пересічних громадян про кібербезпеку і розвиток державно-приватного партнерства.

В свою чергу, зі сторони українських IT-компаній за останні роки були розроблені та запропоновані органам державної влади та звичайним користувачам власні антивірусні продукти.

Прикладом такого продукту, який отримав сертифікацію Держспецзв'язком України, є програмне забезпечення «Zillya! Антивірус для Бізнесу» однойменної компанії Zillya. Антивірус пропонує користувачам комплексний захист комп'ютерної системи та мереж від всіх типів кібератак та кіберзагроз з можливістю централізованого управління та оновлення антивірусних баз.

Іншим своєрідним проявом державно-приватного партнерства може слугувати взаємодія держструктур з окремими групами «білих» хакерів. Так, протягом 2017-2019 років спільнотою «Український КіберАльянс» з метою підтримки інформаційної безпеки в Україні проводились заходи, спрямовані на виявлення вразливостей в об'єктах критичної інформаційної інфраструктури українських державних органів і організацій стратегічного значення [3].

Під час таких заходів було виявлено більше тисячі різних вразливостей інформаційних систем державних установ, в тому числі НАЕК «Енергоатом», команди реагування на кіберінциденти CERT-UA та Національної поліції.

Тобто можна спостерігати, що окремі компанії, організації та спілки громадян вже досить ефективно кооперуються з суб'єктами державної влади.

Водночас, така кооперація має набути характер системності, що пояснюється щоденним зростанням загроз в сфері інформаційних технологій.

Прискорити такий процес можливо шляхом об'єднання основних компаній ринку кібербезпеки у відповідні асоціації, до яких, крім приватного сектора, долучаться також представники державних структур.

Взагалі, в Україні на ринку кібербезпеки діє досить багато приватних асоціацій, серед них можна виділити такі: Всеукраїнська асоціація "Інформаційна безпека та інформаційні технології", громадська організація ISACA, Центр реагування на комп'ютерні надзвичайні події "CERT", тощо [2].

Однак, створення організаційної структури на рівні держави дозволить всім зацікавленим сторонам протидії кіберзлочинності формувати та здійснювати оптимальну та ефективну політику забезпечення інформаційної безпеки на різних рівнях, що дозволить в межах ДПП досягнути стратегічних цілей держави у вказаній сфері.

Отже, як висновок можна зазначити, що для прискорення розвитку інституту ДПП у сфері забезпечення кібербезпеки в Україні необхідним є:

- розробка та ефективна реалізації стратегії розвитку недержавного сектора, який надасть змогу забезпечувати належний рівень кібербезпеки як державних структур, так й приватно-комерційного бізнесу, враховуючи інтереси кожної зі сторін;

- реформування та вдосконалення системи кооперації державних та приватних структур на міжнародному, національному, міжрегіональному, місцевому рівнях шляхом створення відповідних організацій та асоціації;

- укладання та забезпечення виконання партнерських угод за участю держави та провідних вітчизняних IT-компаній;

- створення та забезпечення реалізації проектів для забезпечення обміну досвіду, технічних рішень та об'єднання ресурсів між державою та приватним сектором.

Список використаних джерел:

1. Про рішення Ради національної безпеки і оборони України від 14 травня 2021 року «Про Стратегію кібербезпеки України»: Указ Президента України №447/2021. URL: https://www.president.gov.ua/documents/4472021-40013 (дата звернення 04.11.2021р.)

2. Заскока Ю.В. Державно-приватне партнерство в сфері кібербезпеки україни: стан та проблеми забезпечення. Наукові перспективи № 9 2021. URL: https://goo.su/97V0 (дата звернення 04.11.2021р.)

3. Український кіберальянс (Ukrainian Cyber Alliance). URL: https://goo.su/97v3 (дата звернення 04.11.2021р.)

Ключові слова: державно-правове партнерство, кібербезпека, інформаційна безпека.

Ключевые слова: государственно-правовое партнерство, кибербезопасность, информационная безопасность.

Keywords: state-legal partnership, cybersecurity, information security.

Науковий керівник: *к.т.н., доц. Кухаренко С.В.*

РОЗДІЛ 3.
СОЦІАЛЬНО-ПОЛІТИЧНІ УМОВИ ФОРМУВАННЯ КІБЕРПРОСТОРУ

Горбаченко Станіслав Анатолійович

Національний університет «Одеська юридична академія»

завідувач кафедри кібербезпеки, доктор економічних наук, доцент

НЕДОБРОСОВІСНА КОНКУРЕНЦІЯ ЯК ПРОБЛЕМА КІБЕРБЕЗПЕКИ

В останні роки у світовому кіберпросторі поширюються небезпечні тенденції щодо зростання кількості кіберзлочинів, кібератак та випадків кібершпигунства. Вони, в свою чергу, обумовлюють постійний пошук та впровадження ефективних дій у відповідь, як на міжнародному та державному рівнях, так і у межах окремих бізнес-одиниць. Паралельно із класичними кіберзлочинами, що вже достатній час є об'єктами правового регулювання, кіберпростір починають використовувати також і як середовище для недобросовісної конкуренції. Означена проблема поки що далека від вирішення і потребує не стільки залучення інноваційних технологій скільки ефективних управлінських рішень.

Згідно із ст. 1 Закону України «Про захист від недобросовісної конкуренції» недобросовісною конкуренцією вважаються будь-які дії в конкуренції, що суперечать торговим та іншим чесним звичаям у господарській діяльності [1]. Головною метою подібних дій є досягнення неправомірних переваг над конкурентами. Під неправомірними перевагами слід розуміти такі, що супроводжується порушенням чинного законодавства.

Одним з частих порушень законодавства про захист від недобросовісної конкуренції з боку суб'єктів господарювання, який є до того ж найбільш використовуваним у кіберпросторі, відповідно до ст. 15-1 Закону України «Про захист від недобросовісної конкуренції», є поширення інформації, що вводить в

оману. Зокрема, мова йде про відомості які містять неповні, неточні або неправдиві дані про походження товару, виробника, продавця, спосіб виготовлення, джерела та спосіб придбання, реалізації, кількість, споживчі властивості, якість, комплектність, придатність до застосування, стандарти, характеристики, особливості реалізації товарів, робіт, послуг, ціну та знижку на них, а також про суттєві умови договору.

Основною метою поширення в мережі Інтернет неправдивих, неточних, неповних відомостей, які можуть вводити в оману споживачів, контрагентів, органи державної влади є вплив на клієнтську базу: зменшення лояльності, перехід до іншого продавця чи постачальника, зменшення середньої суми чеку тощо. Окремим напрямом для кібератак з боку конкурентів виступає особистий бренд засновника або керівника суб'єкта господарювання, особливо в сегментах, де продажі ґрунтуються саме на довірі до особистості.

Серед форм поширення неправдивої інформації, яка вводить в оману споживачів, є не тільки ті, що безпосередньо пов'язані із кіберпростором, а й зовнішні рекламні носії, поліграфічна продукція, реклама в роликах на телебаченні та радіо, реєстрація торгової марки із завідомо оманливою інформацією тощо. Проте саме Інтернет середовище виступає найбільш сприятливою платформою для розповсюдження подібної інформації.

Як результативні інструменти недобросовісної конкуренції в кіберпросторі можуть використовуватися фейкові статті в Інтернет – виданнях, поширення неправдивої інформації із залученням лідерів думок у соціальних мережах негативні відгуки на сайті або коментарі на бізнес – сторінках, штучно створенні замовлення в Інтернет – формах або за допомогою чат – ботів (наприклад, з метою порушення нормальної роботи в сфері послуг).

В останні роки в конкурентній боротьбі активно почали використовуватися й фішингові атаки. Зокрема на початку 2021 року спеціалісти виявили масову розсилку фішингових електронних листів по державних установах нібито від Адміністрації Держспецзв'язку. Листи було надіслано з електронної поштової скриньки, домен якої зовні був схожим зі справжнім доменним ім'ям Адміністрації Держспецзв'язку. Файл, який містився у цих електронних листах,

надавав доступ зловмисникам до віддаленого управління ураженим комп'ютером. Тобто особа, яка отримала доступ, мала можливість знищувати, копіювати, змінювати дані, які містяться на уражених комп'ютерах [2].

Метою фішингових атак з боку конкурентів є, насамперед, доступ до персональних дані клієнтів. Персональними або особовими даними вважають відомості чи сукупність відомостей про фізичну особу, яка ідентифікована або може бути конкретно ідентифікована. Адже переважна більшість бізнес структур зберігає та використовує щонайменше такі особові дані клієнтів як ідентифікаційні дані (прізвище, ім'я, по батькові, адреса, телефон тощо); паспортні дані; особисті відомості (вік, стать, сімейний стан тощо); склад сім'ї; освіта; професія; житлові умови; фінансова інформація тощо. Деякі суб'єкти господарювання, які вважають себе клієнтоорієнтованими можуть також формувати, структурувати та зберігати дані щодо способу життя, інтересів та захоплень, споживчих звичок тощо.

Виходячи з вищесказаного побудова ефективної системи кіберзахисту від недобросовісної конкуренції має ґрунтуватися на наступних засадах.

1. Система аудиту згадувань підприємства чи бренду за ключовими словами.

2. Наявність власного юридичного відділу для протистояння недобросовісній конкуренції у правовому полі.

3. Безперервний моніторинг конкурентів з огляду на конкурентні дії які вони використовували раніше у ретроспективі.

4. Постійна робота з персоналом на предмет поведінки в Інтернет середовищі та засвоєння співробітниками основ інформаційної та кібербезпеки.

5. Формування системи контрзаходів у випадку спроб застосувати інструменти недобросовісної конкуренції.

6. Надійний захист основних нематеріальних активів підприємства.

7. Застосування кількісних методів оцінки ризиків від недобросовісної конкуренції (зокрема, аналізу сценаріїв або моделювання методом Монте-Карло).

Вказані засади мають стати основою для побудови суб'єктами господарювання комплексу конкретних заходів щодо прогнозування, ідентифікації та протидії загрозам недобросовісних дій з боку конкурентів на основі наявних технічних можливостей, людських та інформаційних ресурсів, а також нормативно-правового базису.

Список використаних джерел:

1. Про захист від недобросовісної конкуренції: Закон України № 236/96-ВР в редакції від 11.12.2019, підстава - 286-IX. URL: https://zakon.rada.gov.ua/laws/show/236/96-%D0%B2%D1%80#Text

2. Харченко О., Яремич В. Протидія кібератакам, базованим на прийомах соціальної інженерії URL: http://surl.li/arnjt

Ключові слова: конкуренція, кібербезпека, особистий бренд, персональні дані.

Ключевые слова: конкуренция, кибербезопасность, личный бренд, персональные данные.

Key words: competition, cybersecurity, personal brand, personal data.

Трофименко Олена Григорівна

Національний університет «Одеська юридична академія»,

доцент кафедри інформаційних технологій, кандидат технічних наук, доцент

Чепендюк Олександр Олександрович

Національний університет «Одеська юридична академія»,

студент 2-го курсу факультету кібербезпеки та інформаційних технологій

ЧОМУ C++?

Хоча мова програмування C++ з'явилася в далеких 80-х [1], ще й досі вона актуальна, завдяки своїм унікальним перевагам. Не зважаючи на поширені і

тривалі запевнення про те, що C++ ось-ось помре, ця мова і нині входить в топ-10 [2], а то і в топ-5 [3] рейтингів найзатребуваніших мов програмування на ринку IT-праці.

Ця мова дає можливість розробнику максимально контролювати всі аспекти структур, підтримує взаємодію з пам'яттю на апаратному рівні і саме тому підходить для створення проектів, в яких важлива швидкість і продуктивність. Поширено C++ застосовується для розробки системних драйверів, сучасних операційних систем, середовищ розробки, веб-браузерів, графічних редакторів тощо. Наприклад, операційні системи Windows та macOS мають елементи, написані засобами C++, а YouTube використовує C++ для обробки відео [4]. Крім того, C++ використовують при створенні реальної приватної та публічної інфраструктури блокчейну, для розробки шейдерів для графічного програмного забезпечення (ПЗ) та ігор, а також для вивчення нейромереж. C++ потрібна для розробок вбудованих (embedded) систем, щоб пов’язати «залізо» з програмним забезпеченням. Спектр застосування C++ вражає, адже програмовані вбудовані системи використовуються повсюдно: від пристроїв на нашій кухні до промисловості, сільського господарства, медицини, фінансової сфери, автомобілебудування, космічної галузі та фундаментальних досліджень у науці [4, 5]. Розумні будинки, системи моніторингу (від відеонянь до операційних), розробка серверів для аудіо/відеоконференцій, медіаплеєри потребують програмних рішень, для яких мова C++ є основною. Тому C++-розробники будуть ще довго затребувані і створювати справді круті проекти, що змінюватимуть наш світ на краще та якісно покращуватимуть життя людей.

Мова C++ швидка й універсальна. Її творець Бйорн Страуструп розробив цю мову як статично типізовану, мультипарадигмальну універсальну мову програмування середнього рівня складності на основі мови C [5]. Оскільки це розширена версія мови програмування C, C та C++ часто позначають разом як C/C++.

Для кращого розуміння живучості C++ розберемося з її перевагами:

– *швидкість* – одна з головних переваг C++. Швидшою за C++ із поширених зараз мов програмування є лише мова C. Як швидкодієві засоби

програмної розробки для реалізації локальних потреб можуть підійти мови C, Go або Rust, але за допомогою C++ можна писати програми будь-якої складності з гарною швидкодією. Важливо, що йдеться не про час написання коду, який, наприклад, у мов Python, C# і Java може бути суттєво меншим, а код компактнішим. Проте кінцевого користувача мало цікавить час розробки програми, на відміну від зацікавленості у швидкісній роботі відповідного програмного застосунку;

– *кросплатформність* – можливість розробляти програми для різних платформ та операційних систем. C++ є достатньо універсальною, адже її компілятори є в кожній операційній системі, і більшу частину програм можна перенести з однієї платформи на іншу без проблем із бібліотеками та середовищем розробки. При цьому нині важко знайти програмний застосунок, повністю написаний лише засобами C++, але частково його використовує практично кожен більш менш серйозний програмний проєкт. Тому знання цієї мови, чи хоча б певних навичок роботи з нею, ніколи не будуть зайвими для розробника ПЗ;

– підтримка численних парадигм програмування: процедурне програмування, модульне програмування, абстракція даних, об'єктно-орієнтоване програмування та узагальнене програмування. C++ допомагає зрозуміти різні підходи до програмування та методології розробки, які можна застосовувати і в інших мовах. Саме через це більшість навчальних закладів вищої освіти, які готують ІТ-фахівців, мають курс по вивченню C++;

– можливість роботи на апаратному рівні з портами, доступ до API операційної системи, безпосередній доступ до пам'яті комп'ютера, з одного боку, є потужними засобами розробки, а з іншого – ці могутні механізми вимагають певної культури програмування, адже їхнє безконтрольне використання небезпечне в невмілих руках і може спричинити помилки. Проте не варто називати молоток небезпечним інструментом, коли хтось не вміє користуватися ним. Довільна арифметика вказівників, механізми виділення і вивільнення пам'яті, конструктори і деструктори, стандартні шаблони, передача параметрів за посиланням дозволяють продуктивно працювати з пам'яттю на апаратному

рівні. А відсутність в C++ засобів автоматичної перевірки вказівників на вихід за межі діапазону вимагає значної відповідальності і кваліфікації розробника.

C++ доречний для проектів, для яких важливою є продуктивність коду. Ця мова не ідеальна, адже не існує ідеальних мов програмування. Наявність безлічі можливостей, що порушують принципи типобезпеки, може спричинити важковловимі помилки в C++-програмах. Більшість проблем типобезпеки C++ успадкована від C. Відсутність у C++ вбудованої системи прибирання сміття, з одного боку, не рідко через неуважність, особливо у новачків, спричиняє помилки типу «переповнення буфера», а з іншого – відмова автора мови від ідеї використовувати автоматичне керування пам'яттю дозволяє користувачеві самому вибрати стратегію керування ресурсами і завдячує гарній швидкості виконання коду, що є певною візитівкою C++. Також слід зважати на те, що долучення через директиву препроцесора #include до інтерфейсу великої кількості зовнішніх модулів відчутно уповільнює компіляцію. Тому для усунення цього специфічного недоліку поширено реалізується механізм перекомпіляції заголовних файлів (англ. Precompiled Headers).

Багато хто жаліється на складність вивчення мови C++. При цьому в основі популярних Java, JavaScript, C# закладено синтаксис мови C++. Тому специфіку роботи цих мов достатньо непросто опанувати, якщо розробник не знайомий з плюсами. Саме через це C++ тривалий час є однією з перших мов у навчанні програмуванню. Так, за даними дослідження GlobalLogic у Львові, 30% студентів визначили основним напрямком саме C++ [4]. Переважно причиною критики C++ є невміння роботи з нею, тобто небажання враховувати наслідки безконтрольного використання наявних свобод у конструкціях мови, технічну відсутність обмежень на використання її засобів. Адже ця мова вибаглива до знання та врахування її специфіки та тонкощів, але при цьому дозволяє створювати ті об'єкти, які не можливо створювати засобами інших мов програмування. До того ж C/C++ дозволяє проводити агресивну оптимізацію такого рівня, який є недосяжним для інших мов високого рівня.

Мова C++ жива і постійно розвивається, щоб відповідати сучасним вимогам. Офіційна стандартизація C++ розпочалася 1998 року, коли був вперше

опублікований стандарт мови ISO/IEC 14882:1998 (більш відомий як C++98). Наступний стандарт C++ ISO/IEC 14882:2003 був опублікований 2003 року, а переглянутий 2009 року. У 2011 з'явився стандарт C++11, у 2014 – C++14, у 2017 – C++17, а в 2020 – C++20, який, до речі, є діючою версією стандарту [6]. З кожною новою версією додаються нові можливості, конструкції мови ускладнюються, але сам код при цьому стає більш зручним для читання.

З усього вище зазначеного можна зробити висновок, що C++ як мова програмування і досі перспективна, не зважаючи на те, що існують більш нові високорівневі мови. Зараз професійним розробникам ПЗ варто знати кілька мов, адже не існує ідеальних засобів розробки, та й не повинна мова загального призначення бути ідеальною для вирішення всіх завдань. Те, що є найкращим засобом розробки для одного завдання, часто виявляється недоречним тягарем для іншого.

У C++ є свої переваги і недоліки. Ця мова створювалася як зручний інструмент для вирішення широкого кола завдань, але зараз C++ виручає там, де потрібна продуктивність і надійність. Наприклад, засобами C++ написані такі успішні продукти [7] на ринку, як: кросплатформний і перший за швидкістю та продуктивністю браузер Google Chrome; дуже вимогливі до продуктивності системи Adobe Creative Suite, що працюють з відео, аудіо та графікою; широко поширена й насичена фічами система Microsoft Office; десктопний клієнт World of Warcraft, який переробляє величезну кількість подій, слідує купі правил, малює графіку і робить ще купу всього пов'язаного з грою. Важко назвати хоча б одну іншу мову програмування, яка дозволяє написати програми подібні до цих і надати такий самий рівень функціональності, стабільності, розширюваності, портабельності. Тому можна з впевненістю стверджувати, що C++ залишатиметься актуальною ще довго.

Список використаних джерел:

1. История возникновения языка программирования C++. URL: http://cppstudio.com/post/1984/

2. Рейтинг мов програмування 2021. URL: https://dou.ua/lenta/articles/language-rating-jan-2021/

3. TIOBE Index for November 2021. URL: https://www.tiobe.com/tiobe-index/

4. Що нового у C++20: можливості та перспективи. URL: https://dou.ua/lenta/columns/c-plus-plus-perspectives/

5. Talking C++: An Interview with Bjarne Stroustrup. URL: https://www.codecademy.com/resources/blog/bjarne-stroustrup-interview/

6. ISO/IEC 14882:2020. Programming languages – C++. URL: https://www.iso.org/ru/standard/79358.html.

7. Про ненависть до C++. URL: https://habr.com/ru/post/111199/

Ключові слова: C++, мова програмування, парадигми програмування.

Ключевые слова: C++, язык программирования, парадигмы программирования.

Keywords: C++, programming language, programming paradigms.

Трофименко Олена Григорівна

Національний університет «Одеська юридична академія»,

доцент кафедри інформаційних технологій, кандидат технічних наук, доцент

Цьоць Артем Олександрович

Національний університет «Одеська юридична академія»,

студент 2-го курсу факультету кібербезпеки та інформаційних

ЩОДО ЗАСТОСУВАННЯ ТЕХНОЛОГІЙ BIG DATA

Ажіотаж навколо поняття «великі дані» (Big Data), не в останню чергу, зумовлений прагненням компаній та корпорацій отримати якомога більше інформації про потенційних покупців та/або користувачів певної продукції чи

сервісів. Для цього вони в різний спосіб збирають дані про інтереси клієнтів з метою формування їх цифрового психологічного портрета (цифрового відбитка), а згодом надають їм рекламні пропозиції товарів та послуг відповідно до цих даних, щоби максимально зацікавити в їх використанні.

Світовий ринок великих даних та аналітики у 2020 році сягнув 208 млрд доларів США. За прогнозами Expert Market Search до 2026 року він подвоїться і складе 450 млрд доларів [1]. Прогнозовано, що вже 2025 року у світі існуватиме 150 трильйонів гігабайт даних. Сьогодні дані, окрім того, що вони надзвичайно великі, є переважно неструктурованими (приблизно 80% з них), і, вочевидь, традиційні технології не можуть обробляти такі дані [2]. Варто враховувати, що великі дані швидко змінюються і мають багато форм. Крім того, дані постійно архівуються під час їх аналізу. Отже, СКБД або інші традиційні технології не в змозі керувати ними та обробляти їх. Тобто, великі обсяги інформації самі по собі не мають сенсу для людини і потребують інноваційних рішень для аналізу та прийняття рішень.

Щоб їх застосувати для досягнення певної мети, дані необхідно проаналізувати. Аналітика великих даних дозволяє розблокувати приховані закономірності і більш якісно розв’язувати невирішені проблеми, пов’язані з керуванням та обробкою даних. Відповідно, завдяки своїм інструментам, методам і технологіям, рішення Big Data дозволяють швидко отримувати, зберігати, шукати та аналізувати дані. Це, в свою чергу, дає змогу виявити зв’язки та ідеї для інновацій та підвищення конкурентоспроможності.

Для обробки відомостей використовують різні інструменти, перелік яких постійно оновлюється. Серед них виокремлюють такі техніки та методики [3]:

- classification – для передбачення поведінки споживачів у певному сегменті ринку;
- cluster analysis – для класифікації об’єктів за групами завдяки виявленню їхніх спільних ознак;
- crowdsourcing – для збору інформації з великої кількості джерел;
- data mining – для виявлення раніше невідомих і корисних відомостей, які стануть у пригоді для прийняття рішень у різних сферах;

– machine learning – створення нейронних мереж, які самонавчаються, а також якісно і швидко обробляють інформацію;

– signal processing – для розпізнавання сигналів на тлі шуму і їхнього подальшого аналізу;

– змішування й інтеграція – для переведення даних у єдиний формат (наприклад, перетворення аудіо- та відеофайлів на текст);

– unsupervised learning – для виявлення прихованих функціональних взаємозв'язків у даних;

– візуалізація – для презентації результатів аналізу у вигляді діаграм і анімації.

Аналітику великих даних наразі використовують у різних сферах. Великі ритейли збирають інформацію про харчові звички своїх покупців. Авіакомпанії відстежують популярність напрямків та залежно від попиту змінюють маршрути та цінові пропозиції. Влада міст обробляє інформацію про рух автівок та громадського транспорту в режимі реального часу, щоб обмежити швидкість, розвантажити дороги, а також запланувати бюджети на ремонт чи розбудову нових розв'язок [4]. Користь від Big Data очевидна для різних індустрій, зокрема, і для маркетингу, освіти, охорони здоров'я та багатьох аспектів нашого повсякденного життя. Так, оператор мобільного зв'язку Vodafone використовує їх для оцінки ризиків або інтересу до покупки продукту клієнтами [5] для:

1) оцінки ризику неповернення кредиту;

2) визначення шахрайських дій;

3) оцінки ризику страхування;

4) інших можливостей скорингу.

Big Data, як і Data Mining, є частиною науки Data Science, яка в свою чергу охоплює збір та аналіз статистики, машинне навчання, нейронні мережі, використання математичних моделей для використання їх в наявних проектах. Робота у цій сфері вимагає великої кількості знань та умінь і використовує, зокрема, такі методи і алгоритми:

– метод кластерного аналізу *k*-середніх, мета якого є поділ *m* спостережень (з простору) на *k* кластерів;

– регресійний аналіз – набір статистичних методів дослідження впливу однієї чи кількох незалежних змінних;

– метод опорних векторів, який використовується для задач класифікації та регресійного аналізу;

– метод “аналіз соціальних мереж” застосовується для аналізу відносин у соціальних мережах, звідки й пішла назва метод, задля добору можливих друзів, зареєстрованих у соціальних мережах, з яким користувач зустрічався або має спільних друзів, тощо.

Наприклад, інтернет-сервіс потокового аудіо Spotify, що дозволяє легально прослуховувати музичні композиції, завдяки збору та аналізу даних на основі методу соціальних мереж, визначає, які саме пісні та подкасти варто пропонувати потенційному клієнту. Так, якщо пісню переключили, не прослухавши її повністю, алгоритм вважатиме, що вона не сподобалася, і перестане рекомендувати подібні твори, а якщо користувач неодноразово прослухав трек і додав його у свій плейлист, то сервіс продовжить рекомендувати цей жанр [6]. Такий підхід допомагає і виконавцям-початківцям, треки яких нещодавно слухали користувачі.

Нині інструменти великих даних, такі як аналіз даних і машинне навчання, відіграють важливу роль при розробці, поширенні та оцінці вакцин [7]. Останнім часом розробка вакцини проти COVID-19 привернула увагу світу як основного інструмента для подолання поточної пандемії. Засоби, які використовуються для розробки цих вакцин, стрімко еволюціонують, а використання технологій великих даних у цьому нині вже стало стандартом.

Іншою окремою областю, де застосовується Big data, є спорт. Технології збору та аналізу даних використовують на тренуваннях та матчах у вигляді датчиків, які кріпляться на спортсменів та вимірюють дистанцію, що пробіг гравець, і ритм його серцебиття, та допомагають налаштувати навантаження індивідуально для кожного гравця, щоб отримати максимальну користь для гри. Крім того, на основі минулих матчів можна спрогнозувати найімовірніший результат матчу.

Метеорологи використовують аналіз даних про погоду за останні 100 років, що дозволяє виявити закономірності того, за який період року/місяця настає потепління, похолодання чи починається сезон опадів. На основі цих відомостей можливо прогнозувати погоду на найближчий період [8].

Завдяки аналітиці великих даних (Big Data Analytics), можна швидко і якісно інтерпретувати різну інформацію, знаходити закономірності і складати прогнози. Наприклад, за допомогою Big Data визначають, в якій частині міста існує потреба в певних товарах чи послугах, яка продукція зацікавить потенційних покупців, передбачають сплески захворювань і навіть місця, де найімовірніше відбудуться злочини [5]. Чим більше відомостей вдасться дослідити, тим точнішим буде остаточний результат. Саме тому бізнес-аналітика великих даних нині є одним з головних інноваційних трендів в інформаційно-технологічній інфраструктурі.

Перелік використаної літератури:

1. Global Big Data Market Outlook. URL: https://www.expertmarketresearch.com/reports/big-data-market
2. Al-Khasawneh M. Big Data Applications and Tools. AkiNik Publications, 2020. 15 p. URL: https://www.researchgate.net/publication/352413480
3. Технології Big Data: ключові характеристики, особливості та переваги. URL: https://aiconference.com.ua/uk/news/tehnologii-big-data-klyuchevie-harakteristiki-osobennosti-i-preimushchestva-97883.
4. Big Data для маркетингу: як аналітика допомагає продавати ефективніше. URL: https://hub.kyivstar.ua/news/big-data-dlya-marketyngu-yak-analityka-dopomagaye-prodavaty-efektyvnishe/
5. Spotify: Big Data Shows Big Results. URL: https://business.vodafone.ua/produkty/bigdata-scoring
6. DiFranza A. Spotify: Big Data Shows Big Results. URL: https://www.northeastern.edu/graduate/blog/spotify-big-data/.

7. Kasten J.E. Big Data Applications in Vaccinology. International Journal of Big Data and Analytics in Healthcare. 2021. Vol.6(2). P.59-80. DOI: 10.4018/IJBDAH.20210701.oa5/

8. Прохорова Т.В., Колесніков В.О. Застосування технологій Big Data та штучного інтелекту в нових технологічних процесах. Сучасна наука: стан, проблеми, перспективи. 2020. Старобільск. С. 43-46.

Ключові слова: великі дані, застосування великих даних, інструменти великих даних.

Ключевые слова: большие данные, применение больших данных, инструменты больших данных.

Keywords: big data, application of big data, big data tools.

Антіпіна Надія Сергіївна

Національний університет «Одеська юридична академія»,

студентка 2-го курсу факультету кібербезпеки та інформаційних технологій

VR ТА AR ТЕХНОЛОГІЇ

Зараз технології віртуальної і доповненої реальності (Virtual Reality, VR та Augmented Reality, AR) активно впроваджуються в різні сфери діяльності, особливо за часів пандемії Covid-19. Наразі інноваційні підходи з їх застосуванням можна зустріти в секторах навчання, медицини, оборони, архітектури та проектування, логістики, туризму, промисловості тощо. Проте найбільш популярною сферою застосування VR та AR є індустрія розваг, зокрема ігрова, ринок VR-продуктів якої за станом на 2020 рік складав $6.26 млрд [1].

Іноді технологій VR та AR поєднують з терміном Extended reality/XR (з англ. «розширена реальність»), проте між ними є відмінності [2]:

– VR – штучне середовище, створене засобами комп'ютерних технологій. На відміну від традиційних інтерфейсів, де зображення транслюється на екрані, VR дає користувачеві змогу повністю зануритися у цю «паралельну реальність» та взаємодіяти з 3D-моделями. Шляхом стимулювання якомога більшої кількості органів чуття таке віртуальне середовище повністю замінює реальне й позбавляє реакції на зовнішні зміни;

– AR імітує штучні об'єкти в реальному середовищі, тобто зв'язок з реальним світом не зникає. Користувач продовжує отримувати інформацію зі свого фізичного оточення, на яке через пристрої / застосунки AR накладається певне згенероване зображення. Засобами відповідних алгоритмів та за допомогою датчиків технологія відтворює 3D-об'єкти відповідно до розташування та орієнтації камери.

Прикладом такої «доповненої реальності» є популярна не так давно мобільна гра Pokemon GO, де користувач дивиться на реальній світ через камеру та екран свого девайсу й бачить накладені застосунком об'єкти – покемонів, котрих потрібно спіймати. Вони розташовувалися у випадкових точках з радіусі користувача; переміщення забезпечувалося на основі визначення поточного місцезнаходження.

На відміну від AR, для користування VR-застосунками одного гаджету недостатньо. Ключовою складовою стандартного комплекту устаткування є VR-шолом, що, в свою чергу, має [3]:

– корпус, завдання якого – ізолювати зір та слух користувача від реального оточення та сторонніх звуків;

– лінзи, від якості налаштування яких залежить «натуральність» отриманого зображення. Додаткові параметри (діаметр лінз та міжзінична відстань) допомагають налаштувати пристрій під анатомічні особливості користувача);

– датчики (гіроскоп, магнітометр, акселерометр тощо), що відстежують положення та рух голови в просторі;

– мікросхема для обчислювальних процесів.

Стосовно руху і відстеження у VR існує термін Degrees of Freedom (DoF, з англ. «ступені свободи»); найпоширенішими є 3DoF і 6DoF (3 і 6 ступенів свободи відповідно). Пристрої, що забезпечують лише 3DoF, фіксуватимуть лише рухи голови користувача, але не її положення в просторі (по координатах x, y, z), в той час як 6DoF відстежує обидва параметри. Більшість нових пристроїв віртуальної реальності здатні забезпечити повне відстеження позиції через 6DoF [4].

VR-шолом потрібно використовувати разом з якісною парою навушників. Також існують й додаткові аксесуари від ручних контролерів до бігових доріжок, призначені для максимально повного відчуття перебування в «іншому світі».

Популярність XR-продуктів продовжує рости, а індустрія – розвиватися. Через це з'являється й потреба в більшій кількості розробників таких засобів. VR-розробники, перш за все, мають вибрати, з якою платформою вони будуть мати справу. Існують такі напрямки [5]:

– *розробка для віртуальних шоломів*, наприклад, ігор та інших програм для шоломів HTC Vive чи Oculus. Основними мовами програмування для цього є C# та C++. Зазвичай використовують спеціальні ігрові рушії (англ. Game engine), найпопулярніші з яких Unity та Unreal Engine 4;

– *розробка для мобільних застосунків*. Тут, окрім рушіїв, потрібно розбиратися й у розробленні застосунків. Використовують мови програмування: Swift для iOS і Kotlin для Android;

– *розробка для web* засобами JavaScript-фреймворків Three.js та Babylon.js, які мають готові інструменти для налаштування ефектів та видів камер. Також вартий уваги конструктор VR-просторів Vizor.

Двома найбільш популярними мовами програмування XR-продуктів вважають C# та C++ та відповідно й два ігрові рушії, що їх використовують: Unity (C#) та UE4 (C++). Обидва рушії оптимізовані для роботи з усіма останніми технологіями XR, а отже підходять для складних XR проектів, здатні створювати приголомшливі візуальні ефекти та кросплатформні застосунки. Вони можуть застосовуватися і в промисловості, не зважаючи на те, що є ігровими: ВМС США використовують застосунки VR/AR на базі Unity3D для підвищення

ефективності навчання персоналу [6]. Компанія з розробки ліків C4X Discovery використовує Unreal Engine 4 для візуалізації молекулярних даних і, в такий спосіб, прискорює розробку нових ліків. Візуалізація складних даних у віртуальній реальності дозволила вченим спостерігати за сполуками з нової точки зору, легше виявляти недоліки, а також ефективно співпрацювати по всьому світу без потреби перебувати в одному офісі [7]. Інший приклад – Air Canada, яка використала Unreal Engine для створення VR-презентації для свого найвищого сервісу бізнес-класу [8]. Ігрові рушії мають попит і поза межами ігрової індустрії, саме через затребуваність на високу графіку й інтерактивність. Розробка 3D-моделей речей без Unreal займе в двадцять разів більше часу, оскільки потрібно буде розробляти всю інфраструктуру самостійно [9].

Який же засіб розробки вибрати для створення продуктів віртуальної чи доповненої реальності? За роки їх існування Unity став вважатися кращим вибором для розробки інді (від англ. Indie game – independent video game) та DIY ігор, тоді як Unreal Engine більш доречний для масштабних проектів, зважаючи на його успішне використання в іграх AAA-класу. Unity рушій був використаний для розробки 60% від загальної кількості VR/AR продуктів [10]. Мова C# приваблює нових розробників з декількох причин:

– простий синтаксис та простота;

– наявність прибиральника сміття, що допомагає уникати витоків пам'яті;

– наявність численної навчальної літератури по C# для Unity;

– широка приватна спільнота розробників на C# для Unity, готова допомогти розробкам-початківцям.

Середовище C++ Unreal Engine вважається більш придатним для масштабних або технічно вимогливих проектів, завдяки таким перевагам [11]:

– код мовою C++ швидкий й ефективний. Враховуючи, що C++ дозволяє керувати пам’яттю вручну та компілювати безпосередньо в машинний код, масштабні програми можна оптимізувати до максимальної продуктивності, тоді як із C# від Unity така ефективність програми недоступна;

– багатофункціональність дозволяє реалізувати функціонал засобами об'єктно-орієнтованої та функціональної парадигм програмування.

Проте, щоб максимально використовувати можливості C++ Unreal Engine, об'єктно-орієнтованого та процедурного парадигм програмування, розробка в цьому середовищі вимагає досвідчених фахівців, а тому дорожча.

Вибір засобу розробки VR та AR залежить від попереднього досвіду розробника та сфери застосування. Так, для проекту, зосередженому на презентації та графіці, наприклад, візуальній презентації для бізнесу, кращим вибором буде Unreal Engine 4 із системою Blueprints та чудовими візуальними ефектами. Однак, для тих, хто тільки починаєте працювати як розробник, C# від Unity буде більш практичним, оскільки має більшу спільноту та кодову базу VR. Для роботи над складним бізнес-рішенням, яке вимагає багато налаштувань, ітеративної розробки, створення прототипів та оптимізації, доречно вибір зосередити на середовищі Unity3D.

Список використаних джерел:

1. Virtual Reality (VR) in Gaming Market Size, Share & COVID-19 Impact Analysis. URL: https://www.fortunebusinessinsights.com/industry-reports/virtual-reality-gaming-market-100271
2. Joe Bardi. What is Virtual Reality? [Definition and Examples] URL: https://www.marxentlabs.com/what-is-virtual-reality/
3. VR шлеми. URL: https://dronestore.com.ua/product-category/vr-headsets/
4. Baker H. Beginner’s Guide to VR: FAQ And Everything You Need to Know (Updated 2021). URL: https://uploadvr.com/beginners-guide-vr-faq-everything-you-need-to-know/
5. Кушнир Е. Что учить разработчику, чтобы работать с VR: советы экспертов. URL: https://vc.ru/dev/136142-chto-uchit-razrabotchiku-chtoby-rabotat-s-vr-sovety-ekspertov.
6. Circuit Stream clients include Lockheed Martin, the US Navy, VMWare, GE, Raytheon, INVISTA, Hershey's, and Boeing. URL: https://www.thevrara.com/blog2/2019/9/19/congrats-to-circuit-stream-for-launching-a-new-product-an-enterprise-xr-platform

7. Unreal Engine: from gaming to ground-breaking cures. URL: https://www.pharmaceutical-technology.com/features/unreal-engine-gaming-ground-breaking-cures/

8. Air Canada uses VR to showcase its top-flight business class service. URL: https://www.unrealengine.com/en-US/spotlights/air-canada-uses-vr-to-showcase-its-top-flight-business-class-service .

9. Игровые движки теперь не только для игр. URL: https://ru.hexlet.io/blog/posts/igrovye-dvizhki-teper-ne-tolko-dlya-igr

10. Get Unity Certified. URL: https://unity.com/products/unity-certifications/vr-developer

11. C# vs C++. Complete comparison between Unity and Unreal programming languages. URL: https://circuitstream.com/blog/c-vs-c-complete-comparison-between-unity-and-unreal-programming-languages/

Ключові слова: віртуальна реальність, доповнена реальність, XR-розробник, ігрові рушії.

Ключевые слова: виртуальная реальность, дополненная реальность, XR-разработчик, игровые движки.

Keywords: virtual reality, augmented reality, XR-developer, gaming engines.

Науковий керівник: *к.т.н., доц. Трофименко О. Г.*

Баланюк Никита Сергеевич

Национальний университет «Одесская юридическая академия»,

студент 3-го курса факультета кибербезопасности и информационных технологий

КИБЕРБЕЗОПАСНОСТЬ НА УДАЛЕННОЙ РАБОТЕ

В последнее время переход на удаленную работу сопутствуется с нововведениями и проблемами. В частности, организация информационной

безопасности должна была претерпеть изменения в связи с переносом рабочей среды в домашние офисы. А значит существует необходимость в более высоком уровне информационной безопасности и ее более технологичной технической составляющей. Но основные задачи остались неизменными: доступность конфиденциальность и целостность информации, а сейчас с учетом удаленности сотрудников.

1. Угрозы безопасности для дистанционных работников.

Работники находятся в зоне рисков по разным причинам. Устанавливаемое ПО может быть зараженным, незащищенные соединения, вредоносные веб-ресурсы. Никто не отменял простейший фишинг во время перехода на сайт компании для входа в среду разработки или кейлоггер скачанный вместе с ПО из интернета. Необходимость в современных и гибких решениях, как никогда необходима в связи последними событиями в мире. Пандемия стала катализатор для активизации всевозможных хакерских групп, улучшенная инфраструктура и технические возможности позволят им сильнее влиять своими действиями на современный мир. Мобильные устройства остаются самыми желанными целями т.к. каждый из нас держит огромный объем информации о себе в телефоне. В 2021 году программы-вымогатели атаковали каждую 61 организацию в мире каждую неделю. Злоумышленники нацелены на компании, которые могут позволить себе уплату выкупа. Планируется что в 2022 году атаки с вымогателями станут только изощреннее. Хакеры будут все чаще использовать инструменты проникновения для настройки атак в реальном времени, а также для работы в сетях жертв [1].

2. Необходимость в защищенном виртуальном рабочем месте для удаленного сотрудника (VDI).

Для защиты работника первичной необходимостью является защищенное рабочее место. Самым простым способом является перенос приложения, не изменяя процедур и изоляция людей. Или менять подход, если к примеру человек застрял в другой стране без доступа к своему рабочему устройству. Единая идентификация применяется для гибкости работы и мобильности сотрудника. Дает возможность подключения и идентификацию с любого устройства и доступ

ко всем необходимым материалам. Это все реализуется с помощью VDI, они запускаются в центре обработки данных (ЦОД). В этом случае ЦОД не подвергается опасности, т.к. мы запускаем свой VDI и просто передаем картинку. И в этом случае вы получаете возможность обезопасить себя от различного рода зловредов, которые могут находиться на ваших конечных устройствах. Существуют также виртуальные офисы, такое предоставляют отдельные компании вместе с набором необходимых инструментов. На практике легче и безопаснее заказать такую услугу.

3. Безопасная среда для разработки.

Для защиты ЦОД в большинстве компаний используется виртуализирование, которое придает ей гибкость и доступность. Такая технология дает доступ к общему облаку, а также дает возможность развертки сразу на нескольких серверах. Что делает ее уязвимой для атак. Для обеспечения безопасности используют сетевые экраны или фаерволы. Также используется ПО для мониторинга и настройки политик безопасности. Также существуют права доступа и при крайней план восстановления системы при потере доступа к облачным сервисам. Также можно использовать микросегментацию. Этот подход помогает разделить защиту на сегменты на уровне рабочих нагрузок. А значит позволяет создать защиту для каждого приложения и не дать больше нужного уровня доступа чем необходимо и распределить нагрузку. Механизм адаптивный и динамичный благодаря его политикам, которые также могут генерироваться автоматически. Может создаваться на основе межсетевых экранов или на основе защиты конечного хоста что подходит лучше. Защитить устройства пользователя. В большинстве случаев компания не может предоставить своим сотрудникам оборудование поэтому они используют свое. Необходимо идентифицировать как само устройство, так и его пользователя. MDM-системы дают возможность управления любыми устройствами сотрудников, это решение позволяет мониторить действия сотрудника и предотвращать возможные атаки, но обеспечивают безопасность только на самом устройстве.

4. Защита каналов связи.

Использование интернет в качестве канала связи на данный момент зачастую является единственно возможным решением. При этом необходимо понимать, что сеть интернет не защищает своих пользователей от перехвата и компрометации сеансов связи.

1) Применяемые меры.
2) Построение виртуальных сетей связи с помощью средств VPN.
3) Защищенные протоколы передачи данных.
4) Криптографическая защита информации, передаваемой посредством сети Интернет.
5) Настройка алгоритмов шифрования.
6) Контроль за целостностью документов.
7) Динамическая аутентификация пользователей.
8) Примером этой технологии является SMS-рассылка одноразовых паролей.
9) Применение электронного сертификата.
10) Электронный сертификат подтверждает принадлежность ключа владельцу, используется при создании ЭЦП [2].

Список использованной литературы:

1. Информационный портал TADVISER Статья:Главные тенденции в защите информации веб-сайт. URL: https://www.tadviser.ru/index.php/Статья:Главные_тенденции_в_защите_информации (дата звернення: 11.05.2021)

2. Информационный портал TADVISER Интервью TAdviser: Александр Василенко, VMware – о формуле эффективности для «удаленки»веб-сайт. URL: https://www.tadviser.ru/index.php/ Статья: Безопасный доступ с любого устройства к любому приложению в любом облаке – формула эффективности VMware для удаленки (дата звернення: 11.05.2021)

Ключові слова: пристрій, безпека, доступ.

Ключевые слова: устройство, безопасность, доступ.

Keywords: device, security, access.

Науковий керівник: *к.т.н., доц. Кухаренко С.В.*

Дацко Іван Андрійович

Національний університет «Одеська юридична академія», студент 2-го курсу факультету кібербезпеки та інформаційних технологій

ЗАСОБИ .NET ДЛЯ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

.NET – це платформа від Microsoft, що дозволяє розробляти програмне забезпечення. Є доволі популярна думка, що .NET використовується тільки для роботи з мовою програмування C#, але це не так [1]. Звісно, вони тісно пов’язані, але нині цей фреймворк також підтримує розробку мовами C++, F# та VB.NET.

Код компілюється в збірку загальною мовою CIL (Common Intermediate Language), яка далі зберігається у файлі формату PE (Portable Executable). Ця збірка компілюється у байт-код та за допомогою технології JIT (Just-in-time compilation), транслюється у машинний код під час роботи програми [2]. Саме через CIL стає можливою розробка різних модулів програмного забезпечення різними мовами.

2016 року Microsoft випустила у реліз .NET Core, який зараз має назву .NET 6. Головною відміною від свого попередника .NET Framework, що зараз офіційно не підтримується (останнє оновлення було 2019 року [3]), є наявність кросплатформності. Тобто програми на .NET 6 можна розробляти як на Windows, так і на MacOS та Linux. Крім того, з’явилася можливість створювати застосунки для мобільних платформ.

Окрім кросплатформності та підтримки декількох мов програмування, фреймворк від Microsoft має доволі велику та різноманітну бібліотеку класів, яка дозволяє застосовувати їх для розробки різних застосунків. Завдяки цій

бібліотеці та загальномовному середовищу виконання CLR [4], що по суті є віртуальною машиною для виконання програмного коду, написаного відповідними мовами, .NET підтримує широкий спектр технологій, які в свою чергу розширюють можливості для створення програм. Так, для розробки баз даних за допомогою цієї платформи використовують ADO.NET, що є набором класів для отримання доступу до даних у реляційних і XML-базах даних та даних з клієнтських застосунків [5]. Також для роботи з БД користуються мовою запитів Transact-SQL та ORM-фреймворком Entity Framework (ORM – об'єктно реляційна проекція, тобто технологія програмування, що зв'язує між собою бази даних та основні концепції ООП, і створює так звану «віртуальну базу даних» [6]), що дозволяє розробникам працювати з даними через об'єкти класів, не працюючи з базовими таблицями баз даних.

Для розробки вебзастосунків в .NET є ASP.NET – кросплатформне середовище з відкритим вихідним кодом. Воно підходить як для створення сучасних хмарних застосунків, так і для написання серверної частини клієнтського забезпечення, та навіть для програмування застосунків для роботи Інтернету речей (IoT) [7]. Одночасно ASP.NET може бути рішенням і для створення вебінтерфейсу, і для API (Application Programming Interface). У контексті цього середовища також використовують вищезгаданий ADO.NET для легкого доступу до опрацьовуваних даних.

За допомогою .NET можна розробляти комп'ютерні ігри. Популярним середовищем для розробки ігор, особливо для розробників інді-ігор, є Unity. Воно підтримує розробку скриптів на .NET та розширює його функціонал численними класами, які дозволяють взаємодіяти з ігровими об'єктами. Рушій Unity надає широкий вибір інструментів для взаємодії з елементами у грі та їх редагуванням. Unity дозволяє навіть новачкам створювати як 2D, так і 3D ігри під будь-які платформи – будь-то VR, мобільні девайси, ПК або інші засоби відтворення ігор. Також це середовище приваблює вже готовими рішеннями для фізики об'єктів, оскільки має великий вибір параметрів для налаштування [8].

Іншим кросплатформним фреймворком для розробки клієнтських застосунків для мобільних телефонів на .NET, є платформа Xamarin [9]. Цей

фреймворк з відкритим вихідним кодом надає можливість створювати застосунки як для Android, так і для iOS, при чому при портуванні програми з однієї платформи на іншу 90% її коду не буде змінено. Програмне забезпечення на Xamarin можна розробляти як на Windows, так і на MacOS, і компілювати його у пакети, наприклад, .apk для Android чи .ipa для iOS.

Потужним засобом розробки серверної частини вебсайтів є ASP.NET MVC. Він забезпечує виведення необхідного контенту з бази даних у потрібні ділянки веб-сайту, автоматизує процес збору інформації про користувачів, визначає роботу бізнес логіки на сервері, захищає сайт від злому, DoS та DDoS атак.

Отже, основними перевагами у використанні платформи .NET є: кросплатформність, технології CLR, CIL і JIT, стандартна бібліотека класів, широкий вибір допоміжних технологій та платформ (ASP.NET, Unity, Xamarin та ін.), що розширюють можливості .NET.

Зараз .NET програмісти доволі затребувані на ринку праці, оскільки .NET дозволяє розробляти програмне забезпечення самого різного призначення: веб-застосунки, десктопні програми, хмарні послуги, ігри, мобільні застосунки тощо. Наприклад, Unity/Game Developer – це фахівець, який розробляє ігри за допомогою рушія Unity та мови C#. Крім того, цей розробник може створювати бізнес-програми, які можуть застосовуватися в рекламних кампаніях..NET Desktop Developer розробляє десктопне програмного забезпечення для цілого спектру продуктів Microsoft з ОС Windows 10: десктопи, планшети, пристрої інтернету речей, мобільні пристрої, приставки Xbox тощо. WPF і UWP технології допомагають цьому фахівцю в роботі, а ITVDN стане надійним помічником у вивченні основ, що дозволяють створювати якісні настільні програми для користувачів Windows. ASP.NET MVC Developer відповідає за розробку серверної частини вебсайту. Цей перелік фахівців в .NET далеко не вичерпний, а технології .NET розвиваються і вдосконалюються.

Список використаних джерел:

1. Что такое .NET и чем занимаются .NET-разработчики. URL: https://training.epam.ua/#!/News/301?lang=ru

2. CIL – Common Intermediate Language. URL: https://www.scriptol.com/programming/cil.php

3. Язык C# и платформа .NET Core. URL: https://metanit.com/sharp/tutorial/1.1.php

4. Common Language Runtime. (CLR) overview. URL: https://docs.microsoft.com/en-us/dotnet/standard/clr

5. ADO.NET. URL: https://docs.microsoft.com/ru-ru/dotnet/framework/data/adonet/

6. What is Entity Framework? URL: https://www.entityframeworktutorial.net/what-is-entityframework.aspx

7. Документация ASP.NET. URL: https://docs.microsoft.com/ru-ru/aspnet/core/introduction-to-aspnet-core?view=aspnetcore-5.0

8. Overview of .NET in Unity. URL: https://docs.unity3d.com/Manual/overview-of-dot-net-in-unity.html

9. Что такое Xamarin? URL: https://docs.microsoft.com/ru-ru/xamarin/get-started/what-is-xamarin

Ключові слова: .NET, кросплатформність, C#, веб, CLR.

Ключевые слова: .NET, кроссплатформность, C#, веб, CLR.

Key words: .NET, cross platform, C#, web, CLR.

Науковий керівник: *к.т.н., доц. Трофименко О. Г.*

Люлік Анастасія Юріївна

Національний університет «Одеська юридична академія»,

студентка 1-го курсу факультету кібербезпеки та інформаційних технологій

ЗАСОБИ РОЗРОБКИ МОБІЛЬНИХ ЗАСТОСУНКІВ

У наші технологічні часи більшість людей вже не може уявити своє життя без смартфонів. Нині під час карантину через пандемію COVID-19 люди

щоденно використовують мобільні застосунки для праці, замовлень на дім та відпочинку, а організації прагнуть поширити свій бізнес і в цифрові мережі. За даними статистики 2018 року світовий дохід від мобільних застосунків склав понад 365 млрд доларів США, а за прогнозами у 2023 році мобільні застосунки принесуть понад 935 млрд доларів доходу від платних завантажень і реклами в застосунках [1].

Через це розробка мобільних застосунків є вельми актуальною. Для її реалізації, перш за все, потрібно визначитися з типом створюваного мобільного застосунку, а вже потім з мовою програмування. Розрізняють три різновиди програмних мобільних застосунків: нативні, веб та гібридні. Розглянемо їх.

Нативний тип застосунків (Native App) розробляється під конкретну апаратно-програмну платформу, наприклад, iOS або Android, і пишуться мовами, створеними для цієї платформи. Відповідні платформи мають свої набори засобів розробки (SDK – Software Development Kit) і свій стек технологій, зав'язаний на певну мову програмування. Так, рідними мовами для Android є Java та Kotlin, а для iOS, відповідно – Swift та Objective-C [2]. Специфікою нативних застосунків є те, що вони тонко налаштовані під відповідні операційні системи і можуть отримувати вільний доступ до всіх служб, сервісів та ресурсів телефону: камери, мікрофона, геолокатора, акселерометра, календаря, медіафайлів, повідомлень тощо. Через це перевагами «нативок» є: гарна продуктивність, що забезпечує плавну анімацію і високу швидкість завантаження відповідного застосунку; можливість роботи в автономному режимі (в офлайні); економне витрачання ресурсів телефону (батареї); забезпечення надійного захисту даних, завдяки різним рівням захисту операційної системи і максимально ефективному використанню апаратних ресурсів; покращений користувальницький досвід, завдяки налаштованим на платформу компонентам UI/UX. Попри всі переваги варто враховувати те, що для кожної платформи потрібно розробляти свою версію нативного веб-застосунку. При цьому: розробка «нативок» реалізується швидше (оскільки вона для однієї конкретної платформи), легше запобігти помилкам і технічним проблемам, немає залежності від сторонніх фреймворків і бібліотек, а простіша база коду та екосистема суттєво покращують

продуктивність. З іншого боку, розробка Native App для більш ніж однієї операційної системи, наприклад, для iOS та Android, через те, що рідні мови програмування для них вельми специфічні, потребує найняти розробників з відповідними наборами навичок. У підсумку, для завершення нативного проєкту потрібно більше часу і відповідно більше коштів. Тому розробка нативних застосунків є хорошим вибором, якщо потрібно створити застосунок для однієї цільової платформи, наприклад, ігровий застосунок або програму, яка має «важку» анімацію. Тоді нативна розробка програми допоможе впоратися з цим, завдяки швидшій продуктивності.

Веб-застосунки (Web App) запускаються через браузери, при цьому ззовні та функціонально вони схожі на нативні, хоча і не потребують окремо пам'яті чи сховища на пристрої користувача. При цьому не варто плутати веб-застосунки і веб-сайти. Веб-застосунки, на відміну від інформаційного веб-сайту, надають додаткову функціональність та інтерактивність. Наприклад, мегаінформативний сайт Вікіпедія по суті лише надає інформацію, а веб-застосунок Facebook має інтерактивний характер. Прогресивні веб-застосунки (Progressive Web Applications, PWA) мають можливість надсилати push-повідомлення, працювати в автономному режимі та завантажувати на головному екрані. На відміну від нативних мобільних застосунків, розробка Web App може бути простою та швидкою, але це залежить від вимог. Переважна більшість розробок Web App виконується за допомогою JavaScript, CSS і HTML5 [2]. Веб-застосунки не треба завантажувати через AppStore, Google Play чи то інші магазини і при цьому вони будуть працездатні в усіх операційних систем. Вони легкі у знаходженні через пошукові системи і доступні для використання на різних платформах. З іншого боку Web App не мають доступу до вбудованих можливостей мобільних пристроїв, чим зумовлена їхня доволі обмежена продуктивність. Здебільшого веб-застосунок розробляють як гарний мінімалістичний спосіб перевірити ідею і швидко отримати веб-програму, схожу на мобільний застосунок, перш ніж інвестувати в його нативну реалізацію. PWA не є заміною мобільних застосунків, а скоріше оновленням поточного веб-UX, оскільки вони не охоплюють всіх основ для мобільного виявлення.

Гібридний тип (Hybrid Mobile App) поєднує в собі характеристики обох попередніх різновидів мобільних застосунків. Він працює як вебзастосунок, а завантажується як нативний. Написаний за допомогою веб-технологій Hybrid Mobile App запускає код у власному контейнері (webview – спрощений браузер безпосередньо у програмі), використовує рушій браузера, але не сам браузер для рендерінга HTML та локальної обробки JavaScript чи іншої обраної мови [3]. Усі переваги гібридних мобільних застосунків випливають з того, що замість того, щоб створювати дві нативні програми для iOS та Android, можна створити одну і налаштовувати її так, щоб вона працювала на обох платформах. Тому для розробки гібридного застосунку потрібна лише одна кодова база для керування, а отже і менше розробників, менше часу, менше коштів. При цьому Hybrid Mobile App має доступ до функцій пристрою, як у нативних застосунках, завдяки PhoneGap, що є певним мостом між SDK і webview, в якому працює програма. З іншого боку, мабуть найбільшим недоліком гібридних програм є продуктивність, що пояснюється завантаженням у webview, продуктивність якого поки що значно поступається нативним застосункам [4]. Не можна не сказати і про складність розробки кросплатформних веб-застосунків. Прилаштувати гібридний застосунок для прийнятної продуктивності на кожній платформі зазвичай вимагає значної роботи і кваліфікації розробника. Інколи загальна вартість може стати порівняною з повністю нативними програмами. Інколи розробники можуть надавати увагу лише одній операційній системі, через що на іншій можуть з’являтися помилки та погіршуватися якість. Тому слід виконувати тестування на кількох пристроях, щоб переконатися у справності програми.

Отже, кожен з типів мобільних застосунків має свої переваги та недоліки. Обираючи, який саме розробляти, варто зважати на них. Звичайно, все залежить від ідеї, котру треба реалізувати. Після визначення з типом мобільного веб-застосунку наступний крок – вибір мови програмування.

Для розробки нативних застосунків для iOS на пристроях від Apple частіше перевагу віддають мовам Objective-C та Swift. Мова Swift, на відміну від Objective-C, має спрощений синтаксис, адже вона новіша і створена з

урахуванням позитивного досвіду попередників. Ще Swift виділяється своєю швидкістю, численні тести показали продуктивність, близьку до C++ [5]. Якщо орієнтація створюваного веб-застосунку спрямована на операційну систему Android, популярними мовами є Java та Kotlin. Ці дві мови програмування мають свої особливості. Порівнюючи Java і Kotlin, думки розробників сильно різняться [6], адже Java, хоча і вважається доволі складною, вже перевірена часом і має потужні бібліотеки та інструменти, а «молода» мова Kotlin є зручною і лаконічною, що дозволяє спростити написання великих проектів, і при цьому уникнути численних помилок. Крім того, існує кілька платформ, які дозволяють розробляти кросплатформні нативні мобільні застосунки, наприклад: Xamarin, React Native, PhoneGap, Titanium і Flutter від Google. Так, за допомогою Xamarin були розроблені Slack, Pinterest і Honeywell. Засоби React Native використовувались при розробленні Facebook, Walmart, Tesla та Airbnb. Деякі з найбільш відомих програм, створених за допомогою Titanium, – це eBay, ZipCar, PayPal і Khan Academy [4].

Для розробки веб-застосунків поширено використовують мови програмування JavaScript (JS), PHP та Python. Так, швидка і продуктивна скриптова мова JavaScript підтримує усі сучасні браузери, оскільки вона інтегрується з HTML. JS використовується і для фронтенду, і для бекенду, проте вона не відрізняється стабільністю. Потужна мова для створення інтерактивних і динамічних веб-сторінок PHP, на відміну від JS, інтерпретується на стороні сервера в HTML-код, який передається на сторону клієнта. Це з одного боку погіршує інтерактивність і продуктивність сторінок, але краще з точки зору безпеки. Високорівнева універсальна мова Python стійкіша до навантажень сервера, ніж PHP, має простий синтаксис і потужні бібліотеки. Проте ця мова має низьку швидкість, займає багато пам’яті, що може стати суттєвим недоліком у створенні веб-застосунків. Не існує ідеальних засобів розробки, а обґрунтований вибір мови залежить від специфіки створюваного веб-застосунку [7]. До того ж не слід забувати про шаблони та фреймворки для розробки веб-застосунків, як-от: Angular, React, Vue.js тощо.

При розробленні гібридних застосунків можуть знадобитися усі розглянуті вище мови програмування, позаяк, за правилом, інтерфейсу частину розробляють як нативний тип, а ядро створюють як веб-застосунок. Крім того, існують гібридні платформи розробки PhoneGap, Cordova і Canvas. Найвідоміша серед гібридних платформ PhoneGap є, ймовірно, найпростішою для веб-розробника. Наприклад, за допомогою PhoneGap створені гібридні веб-застосунки Sworkit і Untappd. Cordova підтримується Adobe і дозволяє створювати міжбраузерні мобільні застосунки з JavaScript, HTML і CSS. Canvas дозволяє з наявного мобільної вебпрограми або успішного адаптивного сайту створити мобільний застосунок, оптимізувавши його для мобільних пристроїв, як нативну програму для iOS та Android.

Отже, існують різні напрямки розробки мобільних застосунків і різни засоби їх реалізації. Всі вони мають свої плюси і мінуси. При виборі слід враховувати вимоги і специфіку створюваного застосунку, а також можливі обмеження, наприклад, в часі чи грошах, що підштовхне до прийняття певного рішення.

Список використаних джерел:

1. Worldwide mobile app revenues in 2014 to 2023. URL: https://www.statista.com/statistics/269025/worldwide-mobile-app-revenue-forecast/
2. Best Language For Mobile App Development. URL: https://www.webiotic.com/best-language-for-mobile-app-development-what-you-need-to-know/
3. What is a Hybrid Mobile App? URL: https://www.telerik.com/blogs/what-is-a-hybrid-mobile-app-
4. Saccomani P. Native Apps, Web Apps or Hybrid Apps? What’s the Difference? URL: https://www.mobiloud.com/blog/native-web-or-hybrid-apps#6
5. Swift-Speed-Test. URL: https://github.com/futursolo/Swift-Speed-Test
6. Google объявила Kotlin приоритетным языком программирования для разработки Android-приложений. URL: https://vc.ru/dev/66728-google-obyavila-kotlin-prioritetnym-yazykom-programmirovaniya-dlya-razrabotki-android-prilozheniy.

7. Трофименко О.Г. Сучасний інструментарій веб-розробника. *Інформаційне суспільство: проблеми та перспективи* : матер. V всеукр. наук.-практ. конф. (22 травня 2020 р.). URL: http://conf.inf.od.ua/doklady-konferentsii/2020/223-trofimenko.

Ключові слова: нативний застосунок, веб-застосунок, гібридний застосунок, мови програмування.

Ключевые слова: нативное приложение, веб-приложение, гибридное приложение, языки программирования.

Key words: native application, web application, hybrid application, programming languages.

Науковий керівник: *к.т.н., доц. Трофименко О. Г.*

Струк Ніkіта Олександрович

Національний університет «Одеська юридична академія»,

студент 2-го курсу факультету кібербезпеки та інформаційних технологій

МОЖЛИВОСТІ БІБЛІОТЕКИ OPENGL У КОМП'ЮТЕРНІЙ ГРАФІЦІ

OpenGL - одна з програмних технологій обробки двовимірної та тривимірної графіки, яка є конкурентною відносно Direct3D від Microsoft. Ця бібліотека має дуже широкий спектр можливостей та інструментів, до того ж вона безкоштовна, реалізована на усіх сучасних платформах, у тому числі мобільних [1].

Специфіка OpenGl організована досить чітко, тобто за допомогою імені функції тут же можна зрозуміти, яка з функцій є універсальною або "місцевою" для якого-небудь набору відеокарт, чи скільки в ній є параметрів [1].

Технологія OpenGL ефективно поєднується з усіма високорівневими мовами програмування і призначена для розробки ігор, хоча на її основі можна будувати і інші форми мультимедійних-додатків. Раніше вважалося, що ця технологія конкурує лише з бібліотекою Direct3D в частині відображення графіки, але тепер є бібліотеки, які можуть забезпечити набагато більший потенціал програмі.

Основними можливостями бібліотеки OpenGL є:

1. **Робота з кольором** (OpenGL надає програмістові можливість роботи з кольором в режимі RGBA (Red-Green-Blue-Alpha) або використовуючи індексний режим, де колір вибирається з палітри (#...).

2. **Використання в роботі растрових та геометричних примітивів** (на основі примітивів будуються усі об'єкти. З геометричних примітивів бібліотека надає: точки, лінії, полігони. За допомогою них формуються векторні зображення. З растрових: бітовий масив (bitmap) і образ (image)). З них формуються растрові зображення [2].

3. **Згладжування** (завдяки згладжуванню можна приховати ступінчастість, властиву растровим зображенням. Згладжування міняє інтенсивність пікселів біля лінії, а лінія виглядає на екрані без будь-яких зиґзаґів).

4. **Надання прозорості об’єктам**.

5. **Накладання текстури** надає об'єктам більш реалістичний вигляд. Візьмемо певний об'єкт, наприклад куб, на неї накладаємо текстуру (будь-яке зображення), внаслідок чого наш об'єкт тепер виглядає не просто як куб, а він буде різнокольоровим.

6. **Додавання освітлення** задає джерела світла, їх розташування, інтенсивність, і багато інших параметрів.

7. **Накладання візуальних ефектів** дозволяє надати об'єктам або сцені реалістичність, а також "відчути" глибину сцени (наприклад туман, дим).

8. **Подвійна буферизація** надає як одинарну так і подвійну буферизацію. У випадку мультиплікації, використовується подвійна буферизація, яка усуває мерехтіння, тобто зображення кожного кадру спочатку малюється в першому

(невидимому) буфері і тільки потім, коли кадр повністю намальований - увесь буфер відображається на екрані [2].

9. **Перетворення моделей** дозволяє розташовувати об’єкти в просторі, обертати їх, змінювати форму, а також змінювати положення камери з якої ведеться спостереження.).

10. **Використання B-сплайнів** застосовується для малювання кривих по опорних точках [2].

Функціонал і спектр використання бібліотеки OpenGL постійно розширюються. Загалом, можна виділити такі наступні переваги:

1. **Продуктивність** була закладена «вкрай бажана» функція відмальовки динамічних сцен. Для отримання потрібних результатів у систему введено багато параметрів, або, як кажуть, режимів малювання. Якщо певний режим або їх комбінація на цьому обладнанні не в змозі забезпечити інтерактивної взаємодії та необхідної частоти оновлення сцени, то користувач вручну або програма автоматично відключає додаткові функції, оскільки це потрібно для отримання «живої» картинки без небажаних затримок та зависань [3].

2. **Інтероперабельність** дозволяє передавати у мережевому оточенні дані між різними платформами. Тому OpenGL заздалегідь орієнтована на роботу в режимі клієнт-сервер, навіть якщо і клієнт і сервер розташовані на одному комп'ютері.

3. **Повнота функціоналу** відповідає набору функцій, що надаються сучасними апаратними засобами графічної акселерації. OpenGL намагається уникати всього, що має бути реалізовано програмно. Принаймні, гарантується отримання робочої картинки, навіть якщо продуктивність комп’ютера не дозволяє отримати зображення з усіма подробицями і деталями [3]. Інакше кажучи, якщо щось працює на одній платформі, то цей же код буде працювати і на іншій платформі хоча, можливо, і з іншим результатом.

4. **Ортогональність** усіх функції робить їх незалежними одну від одної. Крім того їх можна використовувати їх у довільній комбінації. Для прикладу: використання кольору на певному об’єкті не обмежує можливості його зробити більш темним або більш світлішим.

5. **Розширюваність функціоналу** розрахована на максимальну відповідність можливостям апаратного забезпечення яке має тенденцію розвиватися - в OpenGL також вбудовані механізми включення нових функцій [3].

Однак OpenGL також має свої недоліки, загалом вони пов'язані із роботою з новими можливостями компонентів комп'ютера, через те що ці можливості ще не ввійшли у стандарт. Попри це також виділяють такі недоліки:

1. Методика візуалізації, що використовується в OpenGL, заснована на значних спрощеннях оптичних процесорів, що відбуваються в дійсності, через що деякі шейдери можуть не підтримуватися.

2. Погана підтримка зі сторони ATI (постачальника графічних процесорів AMD Radeon) [4].

Дехто вважає що технологія OpenGL скоро застаріє і стане непотрібною, проте це не так. Як мінімум, вона не втратить свою актуальність у найближчому майбутньому через кросплатформеність. Хоч Microsoft і має свою технологію обробки графіки DirectX, однак в деяких проектах вони використовують OpenGL. Прикладом цього є всім відома гра Minecraft, яка якраз має свою мобільну, консольну та комп'ютерну версії. Так як DirectX не підтримується на мобільних додатках і на консолі, кращим вибором буде використання саме OpenGL. Однак не тільки Microsoft володіє такою технологією. Для своїх програмних продуктів також використовують такі компанії:

- Apple (в розробці мобільних додатків та комп'ютерних програм [5].
- NVIDIA (в тестуванні відеокарт) [5].
- Harris Corporation (в космічній та оборонній сфері) [5].

І це лише невелика частка списку компаній.

Отже технологія має душе широкий спектр використання та багатий функціонал. OpenGL має великі можливості розвитку у майбутньому і багато яскравих прикладів розробок. Вона до цих пір залишається основою в обробці двовимірної та тривимірної графіки.

Список використаних джерел:

1. Зачем нужен OpenGL. [Електронний документ] - URL:https://pozitive.org/windows/stati/opengl.html#novosib.

2. Что такое OpenGL? [Електронний документ] - URL:https://3dnews.ru/169184.

3. Преимущества OpenGL. [Електронний документ] - URL: https://studbooks.net/2080851/informatika/preimuschestva_opengl.

4. В чем недостатки технологии OpenGL? [Електронний документ] - URL: https://otvet.mail.ru/question/21388617

5. Companies Currently Using Open Graphics Library. [Електронний документ] - URL:https://discovery.hgdata.com/product/open-graphics-library-opengl.

Ключові слова: бібліотека OpenGL, технологія обробки графіки, кросплатформеність, розробка ігор.

Ключевые слова: библиотека OpenGl, технология обработки графики, кроссплатформеность, разработка игр.

Keywords: OpenGl library, graphics technology, cross-platform, game development.

Науковий керівник: *к.т.н., доц. Задерейко О.В.*

Сорокін Владислав Сергійович

Національний університет «Одеська юридична академія», студент 2-го курсу факультету кібербезпеки та інформаційних технологій

МОВИ ПРОГРАМУВАННЯ ДЛЯ РОБОТОТЕХНІКИ

Сучасний етап розвитку науки й техніки характеризується зростанням популярності робототехніки та розширенням сфери використання роботів.

Загальні тенденції впровадження робототехніки в освіту змінились протягом останніх років після появи доступних компонентів для навчання робототехніки, що уможливило широке розповсюдження власноруч сконструйованих роботів.

За даними ResearchAndMarkets, очікується, що ринок промислової робототехніки зросте на 48 166,9 мільйона доларів США до 2025 року з 16 847,6 мільйонів доларів США у 2017 році [1].

Програмування роботів – це мистецтво навчання робота в режимі онлайн або офлайн для виконання певних завдань з необхідною точністю, швидкістю та повторюваністю. Програмування код має бути зрозумілим для робота, який по суті є програмованим багатофункціональним маніпулятором. Наприклад, промисловий робот призначений для переміщення матеріалів, деталей, інструментів або інших спеціалізованих пристроїв для виконання різноманітних завдань в межах заводського цеху.

Програмування промислового робота – це не лише кодування інструкцій за допомогою низькорівневої мови програмування. Оскільки технології, які лежать в основі робототехніки, продовжують розвиватися, з'явилися нові методи програмування, призначені для полегшення роботи користувачів.

З моменту виникнення програмованих обчислювальних систем засоби розробки керувального програмного забезпечення постійно вдосконалюються. Вони еволюціонували від машинних кодів та асемблера до мов програмування високого рівня та об'єктно орієнтованого програмування.

Комунікація між роботом та людиною відбувається за такою схемою [2]:

1. людина пише програму мовою програмування;
2. написаний код програми проходить через компілятори, інтерпретатори або транслятори і перетворюється на машинний код або байт-код для виконання віртуальною машиною, зрозумілий роботу;
3. машинний код потрапляє до «мозку» робота, роль якого може відіграти схема, чіп або мікроконтролер, де і виконується.

Для виконання певної дії роботи програмуються або за допомогою програмування в автономному режимі, або за допомогою наведення. Роботи отримують інструкції у вигляді комп'ютерних команд, і це називається

автономним програмуванням на рівні маніпулятора. Більшість промислових роботів запрограмовані шляхом наведення робота від точки до точки через фази операції, при цьому кожна точка зберігається в системі керування робота [1].

Використання офлайнового програмування передбачає мови високого рівня, в яких роботизовані дії визначаються завданнями або цілями.

Програмісти-роботи повинні володіти знаннями про різні типи мов програмування, оскільки перехід від комп'ютерів до роботів не є плавним переходом, як можуть думати багато розробників/програмістів.

Охарактеризуємо різні типи мов програмування роботів.

C/C++ – найпопулярніші мови програмування в робототехніці, які дозволяють високопродуктивно взаємодіяти з апаратним забезпеченням у реальному часі. Наприклад, мікроконтролер Arduino використовує С-подібну мову програмування. Завдяки численним інструментам, бібліотекам і функціям, C/C++ є лідером серед засобів програмування серед «інженерів-робототехніків». Саме C/C++ використовують, коли треба забезпечити максимальну продуктивність робота, тобто високу якість роботи на швидкій швидкості.

Python – це мова програмування високого рівня, яка відіграє ключову роль у створенні та тестуванні роботів. Ця мова програмування дозволяє написати сценарій, який обчислюватиме, записуватиме та моделюватиме всю програму робота замість того, щоб вручну навчати робота кожному оператору. Це допомагає швидко протестувати та візуалізувати рішення під час моделювання, а також уточнити програму та її логіку. Саме Python використовується з Raspberry Pi. Python є популярною мовою, завдяки її використанню в машинному навчанні, а також тому, що її можна використовувати для розробки пакетів роботизованих систем ROS (Robot Operating System). Python рекомендується для новачків, які починають працювати в робототехніці та програмуванні.

JAVA пропонує набір API, які створені відповідно до потреб сфери робототехніки. Розпізнавачі команд і керування, системи диктування та синтезатори мовлення можуть бути створені за допомогою Java Speech API, а Java Media Framework може використовуватися для отримання й обробки візуальних зображень. Java містить усі функції високого рівня, необхідні в

індустрії робототехніки, особливо коли йдеться про штучний інтелект, машинне навчання, пошук та нейронні алгоритми. Мова Java «приховує» основні функції пам'яті від програміста, що полегшує програмування. Теорія використання цієї мови полягає в тому, що можна використовувати один і той самий код на різних машинах, завдяки віртуальній машині Java, хоча це дещо уповільнює виконання коду.

LISP і Prolog – мови логічного програмування, які використовують для програмування штучного інтелекту (ШІ), що останнім часом стрімко набирає популярності. Багато важливих розділів ROS написані засобами LISP, а Prolog використовувався для програмування Watson AI від IBM. Хоча нині програмувати ШІ можна і засобами інших мов програмування, однак LISP і Prolog залишаються в основі деяких реалізацій ШІ і, безумовно, заслуговують на своє місце в цьому списку.

C#/.NET – мова програмування від Microsoft, яка використовується для розробки у Visual Studio. Для робототехніки C#/.NET зазвичай використовується в програмуванні на рівні портів і сокетів. C# може використовуватися для впровадження нейронних мереж. Приклади включають Netduino, FEZ Rhino тощо [3].

Отже, програмування є одним з основних навиків, який необхідно розвивати для роботи в сфері робототехніки. Мови програмування мають глибокий вплив на інтеграцію роботів у промислові машини, а роботизоване програмне забезпечення відіграє ключову роль у виконанні складних операцій і точної функціональності.

Список використаних джерел

1. Different Types of Robot Programming Languages. URL: https://www.plantautomation-technology.com/articles/different-types-of-robot-programming-languages.

2. Owen-Hill A. What is the Best Programming Language for Robotics? URL: https://blog.robotiq.com/what-is-the-best-programming-language-for-robotics

3. Davidson A. Must learn top programming languages for robotics in 2022. URL: https://codersera.com/blog/top-programming-languages-for-robotics/

Ключові слова: робототехніка, мови програмування, програмування робота.

Ключевые слова: робототехника, языки программирования, программирование робота.

Keywords: robotics, programming languages, robot programming.

Науковий керівник: *к.т.н., доц. Трофименко О. Г.*

Титиевский Виталий Олегович

Национальный университет «Одесская юридическая академия», студент 2-го курса факультета кибербезопасности и информационных технологий

БЫСТРОДЕЙСТВИЕ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ

Ни для кого не секрет, что существует огромное количество языков программирования. С каждым годом их становится всё больше и больше. Логичным будет вопрос: «Какой язык программирования лучше?» Естественно, есть много критериев, по которым можно выбрать тот или иной язык, при этом стоит учитывать, что каждый язык хорош по-своему и у каждого программиста есть свои предпочтения. Одним из объективных критериев является критерий быстродействия языка.

Скорость языка программирования зависит от нескольких факторов: кроссплатформенность, компилируемость, статическая и динамическая типизация [1]. Проанализируем их.

Первый фактор – наличие компилятора, то есть возможности преобразовать программный код в набор инструкций для данного типа процессора (машинный код) и далее в исполняемый файл, который может быть запущен на выполнение

как отдельная программа. Важно то, что компилируемые языки всегда будут работать быстрее, чем интерпретируемые, поскольку интерпретатор просматривает программу по строкам и преобразует её в машинный код, а компилятор компилирует весь код сразу [2]. Поэтому по критерию быстродействия такие известные языки, как JavaScript, Python, Ruby, PHP не являются лидерами [3].

Второй фактор – кроссплатформенность как способность программного обеспечения (ПО) производить процессы с несколькими операционными системами и аппаратными платформами [4]. Некоторые языки, например, Java и C#, находятся между компилируемыми и интерпретируемыми. А именно, программа компилируется не в машинный язык, а в машинно-независимый код низкого уровня, байт-код. Для выполнения байт-кода обычно используется интерпретация, хотя отдельные его части для ускорения работы программы могут быть транслированы в машинный код непосредственно во время выполнения программы по технологии компиляции «на лету». Для Java байт-код исполняется виртуальной машиной Java (Java Virtual Machine, JVM), для C# — Common Language Runtime. Таким образом, кроссплатформенность замедляет эти языки программирования.

Ещё одним фактором, влияющим на быстродействие языка программирования, является его типизация. Существует динамически и статически типизированные языки. Различие между этими типами заключается в том, что в языках со статистической типизацией тип переменной известен во время компиляции, а не выполнения, а в языках с динамической типизацией – во время выполнения программы.

Исходя из данных факторов был сформирован ориентировочный рейтинг быстродействия языков [1]. При этом в этом рейтинге не учитывались другие параметры, характеризующие те или иные сильные стороны этих языков программирования. Охарактеризуем лидеров этого рейтинга.

Возглавляет рейтинг быстродействия языков программирования платформенно-независимый быстрый и энергоэффективный язык C. Его особенность заключается в простоте и максимальной близости к аппаратному

слою, что позволяет работать с кодом низкого уровня. Язык C изначально создавался для системного программирования, поэтому неудивительно, что его активно используют для создания операционных систем и программного обеспечения. Ведь любая операционная система, в первую очередь, должна быстро запускаться и эффективно управлять системными ресурсами. Разработчик на C может самостоятельно реализовывать структуры данных, производя тонкую настройку операционной системы. Такая высокая гибкость – весомый аргумент в пользу языка. Кроме того, этот язык программирования широко используется для проектирования встроенных систем [5]. Итак, C – самый быстрый язык, но в отличие от своего собрата C++, ещё уязвимей и в целом считается самым уязвимым среди популярных языков.

Второе место в рассматриваемом рейтинге занимает C++ как один из самых скоростных и эффективных языков программирования. В общем и целом, его используют преимущественно из-за высокой скорости выполнения задач. В связанных системах этот язык позволяет работать непосредственно с ресурсами памяти. Поэтому беспилотные автомобили, умные часы, сенсоры и устройства, использующие технологию Интернета вещей (например, кофемашины), имеют встроенное ПО, целиком написанное на C или C++. Кроме того, C++ является одним из лучших для разработки игр. Например, игры Counter-Strike, StarCraft: Brood War, Diablo I, World of Warcraft написаны на C++ [5]. Средства разработки C++ могут совладать даже с самой сложной игровой графикой. Они позволяют оптимизировать и регулировать то, как именно будут использоваться ресурсы памяти и структуры данных в игре. Если есть сильная сторона – должна быть и слабая. C++ страдает от такой неприятной вещи как ошибки буфера.

Почётное третье место занимает мультипарадигмальный высокоуровневый язык программирования Rust. Этот язык, что называется, ловит золотую середину – балансирует между скоростью и надежностью. Rust не имеет сборщика мусора – это одна из причин его скорости. Используется он преимущественно для обеспечения безопасности и производительности.

Четвертое место занимает не очень популярный язык Fortran. Этот язык является языком общего назначения и используется преимущественно для

научных расчетов. Благодаря своей скорости и производительности, Fortran используется в рейтинге самых быстрых суперкомпьютеров.

Пятое место в рейтинге занимает высокопроизводительный язык программирования Julia. Этот язык эффективен как для реализации математических вычислениях, так и для написания программ общего назначения. Julia – достаточно быстрый язык, но есть причина, по которой он не может возглавлять этот рейтинг: динамическая типизация.

Далее в рейтинге идут: объектно-ориентированный высокоуровневый язык Ada, весьма популярный строго типизированный объектно-ориентированный язык Java, объектно-ориентированный язык программирования с несколькими парадигмами C#, мультипарадигмальный язык программирования F# и т.д.

Исходя из данных исследования [1], самым быстрым языком является язык программирования C. Но при этом стоит понимать, что, если язык считается одним из самых быстрых – это не значит, что он является лучшим языком программирования. Для многих разрабатываемых программных проектов быстродействие – это не самое главное.

Языки кардинально отличаются друг от друга. Один язык более читабельный и прост в изучении, а другой напротив более громоздкий, но при этом позволяет создавать более сложные, проработанные приложения. Язык может не блистать характеристикой быстродействия, но его простота в использовании позволит значительно сократить затраты на рабочую силу и разработку. Каждый выделяет для себя свой лучший язык, сколько существует людей – столько и мнений.

Список использованных источников:

1. Top 10 Fastest Programming Languages. URL: https://www.geeksforgeeks.org/top-10-fastest-programming-languages/

2. Джок А. Компиляторы, интерпретаторы и байт-код. URL: https://www.osp.ru/cw/2001/06/9339

3. Компилируемые и интерпретируемые языки программирования. URL: http://itmentor.by/articles/kompiliruemye-i-interpretiruemye-yazyki-programmirovaniya

4. Кроссплатформенность. URL: https://wiki2.org/ru/Кроссплатформенность.

5. Why and where should you still use C/C++ languages? URL: https://hackernoon.com/why-and-where-should-you-still-use-cc-languages-6l1r838gh

Ключевые слова: скорость языка программирования, компилятор, кроссплатформенность.

Ключові слова: швидкість мови програмування, компілятор, кросплатформність.

Keywords: programming language speed, compiler, cross-platform.

Науковий керівник*: к.т.н., доц. Трофименко О. Г.*

ЗМІСТ