УДК 004.056.53:005
DOI: 10.31673/2412-9070.2022.030311

Ю. І. КАТКОВ, доктор техн. наук, доцент;
О. В. ЗІНЧЕНКО, доктор техн. наук, доцент;
С. С. ЦИБУЛЬНИК, студент;
Ю. О. ВІТЕНКО, аспірант,
Державний університет телекомунікацій, Київ

ЗАСОБИ ПРОТИДІЇ ЗАГРОЗАМ ДЛЯ ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ ВІД БЕЗФАЙЛОВОГО ЗЛОВМИСНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Розглянуто актуальне питання пошуку засобів протидії загрозам для інтелектуальних систем від безфайлового зловмисного програмного забезпечення. Поставлено задачу: для своєчасної локалізації та мінімізації можливих збитків від впливу загроз (уразливостей, атак) від безфайлового зловмисного програмного забезпечення на критичні об'єкти IT-інфраструктури інтелектуальних систем підприємства потрібно визначити найкращий напрям створення методів їх своєчасного виявлення. У статті показано, що інтелектуальна система підприємства має вразливість та критичність від слабкого впливу загроз (уразливостей, атак) зловмисника на критичні об'єкти IT-інфраструктури підприємства. Наслідком цього є загострення протиріч між складністю методів захисту об'єктів критичної IT-інфраструктури підприємства від шкідливого програмного забезпечення, зокрема безфайлового зловмисного програмного забезпечення, та їх результативністю з погляду своєчасної локалізації та мінімізації можливих збитків від впливу загроз. Але розв'язання цього протиріччя можливе завдяки створенню гнучких організаційних структур спостереження за синергетичними уявленнями процесу адаптації інтелектуальних систем підприємства до загроз від безфайлового зловмисного програмного забезпечення. Показано, що передусім є потреба в постійному вдосконаленні методів виявлення впливу загроз в напрямі їх передбачення. Для цього, ґрунтуючись на виконаний аналіз алгоритму дії безфайлового зловмисного програмного забезпечення через набори експлойтів, шкідливі макроси Microsoft Word та скомпрометоване мережне обладнання, визначено механізм впливу на PowerShell операційних систем Windows, Unix. На основі цього механізму запропоновано дії щодо захисту від безфайлових шкідливих програм. Для реалізації цих дій показано, в який спосіб можливо здійснювати пошук макросів, виявляти та підтверджувати наявність безфайлових загроз. Запропоновано застосування перевірки інформаційної безпеки підприємства за допомогою методів аудиту. Як основний метод аудиту рекомендовано використовувати методологію моделювання атак хакерів Red Teaming та методи тестування penetration testing. Розглянуто способи їх вживання.

Ключові слова: модель загроз; відкритий вихідний код; безфайлове програмне забезпечення; PowerShell.

Вступ

Сьогодні в умовах бурхливого розвитку «цифровізації» суспільства особливої актуальності набуває розвинення підприємств (організацій, фірм, компаній), пов'язане з упровадженням інтелектуальних та інформаційних технологій (*Intellectual information technology*, ІІТ). Перевага ІІТ у тому, що вони генерують корисну інформацію, яка зменшує затримки, витрати і ризики ухвалення рішень, тобто підвищують якість функціонування підприємств. На основі ІІТ створюються інтелектуальні системи підприємств, призначені для нарощування ефективності керування та ухвалення рішень в умовах, пов'язаних із виникненням проблемних ситуацій [1].

Інтелектуальна система (*intelligent system*, ІС) підприємства — це комп'ютерна система підприємства, яка здатна збирати, аналізувати та реагувати на дані, котрі надходять із навколишнього середовища, вона може навчатися на досвіді та адаптуватися відповідно до поточних даних, здатна вирішувати завдання, які традиційно вважаються творчими. Фактично ІС підприємства готові виконувати дедалі більше ролей у суспільстві, зокрема: автоматизація процесів керування підприємством, сервісна робототехніка, медична допомога, освіта, розважальна галузь, візуальний огляд, розпізнавання символів, ідентифікація людини з використанням різних біометричних модальностей (наприклад, обличчя, відбиток пальця, райдужної оболонки ока, руки), візуальне спостереження, інтелектуальний транспорт. Цей перелік свідчить, наскільки тепер важливе значення мають ІС підприємства і чому вони стали об'єктами для атак зловмисників.

З технічного погляду ІС підприємства мають критичні об'єкти у своїй IT-інфраструктурі. Дійсно, ІС підприємства має відповідну для розв'язання завдань IT-інфраструктуру, яка містить апаратну та програмну частини, а також персонал, що їх обслуговує (оператори, системні адміністратори та ін.). Апаратна частина — це устаткування робочих місць та серверне обладнання, що розміщується в мережі належним чином. Програмна частина — це три складові, а саме: база знань, програмний розв'язувач та інтелектуальний інтерфейс.

Відповідно до [2] IT-інфраструктуру ІС підприємства можна розглядати як складну організаційно-

технічну систему. Там показано, що будь-яка складна організаційно-технічна система має вразливість та критичність від слабкого впливу загроз (уразливостей, атак) зловмисника на критичні об'єкти ІТ-інфраструктури ІС підприємства. Основною проблемою таких систем є те, що під дією слабкого впливу загрози критичний об'єкт раптово втрачає задані властивості і набуває інших, які не можуть бути передбачені під час її проєктування через потенційну вразливість окремих її елементів (критичних об'єктів).

Яскраво це демонструють приклади, що характеризують уразливість та критичність від слабкого впливу загрози на критичний об'єкт [2]: зараження шкідливою програмою головного сайту компанії, яке з легкістю виявляється захисним засобом і повністю усувається. Але за час наявності шкідливої програми на сайті інтелектуальні пошукові системи інтернету послаблюють рейтинг сайту до такого рівня, що буде потрібно місяці, аби відновити колишню позицію, що призведе до жахливих наслідків для бізнесу компанії, навіть до межі банкрутства; фейкові новини (чутки) щодо неплатоспроможності банку, на які зважають інтелектуальні засоби аналітики фінансового ринку, можуть призвести до паніки серед вкладників, які своїми діями здатні спричинити реальне банкрутство банку.

Отже, наявність критичних об'єктів в ІТ-інфраструктурі ІС підприємства зумовлює потребу в постійній адаптації заходів захисту (кіберзахисту) критичних об'єктів стосовно впливів шкідливого програмного забезпечення, щоб своєчасно їх нейтралізувати або зменшити наслідки негативного впливу.

Далі розглянемо, для чого потрібно постійно адаптувати заходи захисту. У [1; 2] показано, що цілями дії зловмисника в кіберпросторі складної організаційно-технічної системи (в нашому разі в ІС підприємства) є доступ до системних ресурсів (бажання атакувати інші хости та отримати пропускну здатність або пам'ять, які зловмисники можуть використати незаконно), до конфіденційної інформації (можуть здобути інформацію для компрометації або для незаконного прибутку), а також здійснення вандалізму (для порушення безпеки та завдання шкоди компанії) та саботажу (для підриву довіри до компанії, виведення її з ладу або припинення її діяльності). Дії зловмисника та види його нових загроз (уразливостей, атак) постійно вдосконалюються, тому постає потреба в постійній адаптації ІТ-інфраструктури до нових загроз (атак).

Звідси стає зрозумілим, що нейтралізація наслідків кіберзагроз зумовлює появу низки нових завдань для служб ІТ-захисту підприємства стосовно захисту критичних об'єктів, а саме: визначення переліку загроз, виявлення наслідків їх прояви та оцінювання ступеня їх небезпеки, розроблення та оцінювання ефективності заходів (алгоритмів) нейтралізації впливу загроз (уразливостей, атак). На основі розв'язання цих завдань фахівці служб ІТ-захисту підприємства розробляють політики захисту та безпеки, які передбачають такі кроки: підготовку стратегій та планів керування ризиками; забезпечення захисту ІТ-інфраструктури компанії; захист мережного периметра; використання засобів захисту від загроз (атак); розроблення плану реагування на інциденти кібербезпеки.

Відповідно до [1-3] це можливо забезпечити завдяки створенню гнучких структур за умови зміни постулату жорсткості на постулат гнучкості, що є парадигмою для спостереження за синергетичними уявленнями процесу адаптації ІС підприємства. Синергетичні уявлення процесу адаптації ІС підприємства мають безліч напрямів, більшість з яких не мають наукового опису і мало досліджені. Інструментом удосконалення ступеня небезпеки ІС підприємства (захисту кіберпростору ІТ-інфраструктури ІС підприємства) є застосування організаційно-технічних методів захисту об'єктів критичної ІТ-інфраструктури на основі результатів моделювання ймовірних загроз (уразливостей, атак) зловмисника на ІТ-інфраструктуру ІС підприємства.

Відомо, що організаційно-технічні методи (правові, технічні, програмні і криптографічні) захисту критичних об'єктів ІТ-інфраструктури є базисом комплексної системи захисту інформації. На їх основі формуються політики безпеки, в яких визначаються умови обмеження та розмежування доступу до ресурсів системи, умови атестації та роботи сертифікованого обладнання та персоналу, вимоги та права користувачів, алгоритми контролю, аналізу та оцінювання ефективності функціонування системи захисту інформації. Серед методів, які використовують для виявлення загроз (уразливостей, атак), варто виокремити моделювання загроз, адже цей метод дає змогу вже на ранніх етапах вживати заходів, що знизять витрати на ліквідацію загроз (уразливостей, атак) та спростять їх усунення.

Одним з об'єктів моделювання дії ймовірних загроз (уразливостей, атак) зловмисника на критичні об'єкти ІС підприємства є вплив від шкідливого програмного забезпечення. Дійсно, зловмисники, щоб вислизнути від виявлення їх шкідливих програм засобами ІТ-захисту підприємства, завжди шукають способи їх поширення в різноманітних формах, наприклад як віруси, програми-шпигуни, програми-вимагачі, безфайлове програмне забезпечення тощо. Зазвичай їх поширюють завдяки фішинговим схемам, коли жертвам відправляють інфіковані посилання з підроблених вебсайтів.

Через цікавість, необережність, необізнаність та інколи жадібність жертви відвідують небезпечні вебсайти та необачливо відкривають інфіковані вкладення електронної пошти або небезпечних посилань, у такий спосіб під'єднуючи до обладнання інфіковані змінні носії пам'яті.

У шкідливому програмному забезпеченні порівняно новим небезпечним видом є безфайлове зловмисне програмне забезпечення. Безфайлові шкідливі програми швидко стають одним із найпопулярніших способів створення загроз. Небезпека таких програм полягає в тому, що вони не залежать від файлів і не залишають слідів, ускладнюючи виявлення та видалення критичними об'єктами ІТ-інфраструктури ІС підприємства.

Отже, сьогодні загострилося протиріччя між складністю організаційно-технічних методів захисту об'єктів критичної ІТ-інфраструктури ІС підприємства від шкідливого програмного забезпечення, зокрема безфайлового зловмисного програмного забезпечення, та їх результативністю з погляду своєчасної локалізації та мінімізації можливих збитків від впливу загроз. Тому передусім постає потреба в постійному вдосконаленні методів їх виявлення.

Постановка завдання. Для вчасної локалізації та мінімізації можливих збитків від впливу загроз (уразливостей, атак) від безфайлового зловмисного програмного забезпечення на критичні об'єкти ІТ-інфраструктури ІС підприємства потрібно визначити найкращий напрям створення методів їх своєчасного виявлення.

Аналіз останніх досліджень. Безумовно, розв'язання цього завдання в сучасних умовах розвитку цифровізації суспільства є надзвичайно нагальним та актуальним. Сьогодні посилено ведеться пошук організаційно-технічних методів захисту об'єктів критичної ІТ-інфраструктури від шкідливого програмного забезпечення, зокрема безфайлового зловмисного програмного забезпечення. Вирішенню зазначеного протиріччя присвячено багато наукових праць. У [1; 2] розглянуто теоретичні аспекти, у [3] визначено причини критичних ситуацій в інформаційно-інтелектуальних системах, у [4] — стратегію атак та оборони, у [5-10] описано перелік загроз та знаходження наслідків їх прояви, у [11-13] запропоновано методи блокування PowerShell для захисту від зловмисників, у [14-16] — методи тестування на проникнення загроз. Ці технології дають змогу ІТ-службам безпеки підприємства швидко ідентифікувати вже відому загрозу завдяки наявності бази даних загроз (бібліотеки образів виявлених загроз), яка надає можливість отримувати інформацію про алгоритм дії зловмисника та швидко реагувати на загрозу, що безумовно є шляхом підвищення ефективності процесу захисту. Але загроза безфайлового шкідливого програмного забезпечення продовжує зберігатися в усьому світі, вона дуже відрізняється від загроз, що створюються іншими шкідливими програмами, тому постає потреба в розробленні цільових стратегій керування ризиками. Звідси пошук організаційно-технічних методів захисту об'єктів критичної ІТ-інфраструктури від безфайлового зловмисного програмного забезпечення набуває сьогодні особливої актуальності.

Основна частина

Традиційним методом своєчасного виявлення збитків від впливу загроз (уразливостей, атак) від безфайлового зловмисного програмного забезпечення на критичні об'єкти ІТ-інфраструктури ІС підприємства є моделювання ймовірних загроз (атак) зловмисника для критичних об'єктів ІС підприємства від шкідливого програмного забезпечення.

Відомо [5-9], що шкідлива програма — це комп'ютерна програма або переносний код, призначений для реалізації загроз даним, що зберігаються в ІС підприємства, або для прихованого нецільового використання її ресурсів, або іншої дії, що перешкоджає нормальному функціонуванню об'єктів критичної ІТ-інфраструктури ІС підприємства. Серед типів шкідливих програм найпоширенішими є рекламне програмне забезпечення, віруси, черв'яки, хробаки, трояни, боти, програми-вимагачі, шпигунське програмне забезпечення, руткіти, мобільні шкідливі програми, безфайлові шкідливі програми.

Ці шкідливі програми крадуть, шифрують та видаляють конфіденційні дані; можуть змінювати або захоплювати основні обчислювальні функції та відстежувати активність комп'ютерів кінцевих користувачів; створюють умови швидкого поширення завдяки здатності приєднуватися до резидентних програм, надавати можливість зловмиснику доступу до носіїв даних тощо. Шкідливі програми вносять дезорганізацію процесів та деструктивні дії, наприклад, можуть згенерувати команду на видалення файлів або форматування дисків, на внесення змін у процеси оброблення даних через блокування запуску певних програм, заповнення оперативної пам'яті спамом, ураження нових об'єктів, збирання і пересилання копії кодів доступу до даних, спотворення зображення на екрані монітора, використання уражених комп'ютерів для колективних атак на інші комп'ютери в мережах тощо.

Серед цих шкідливих програм особливої уваги заслуговує безфайлове зловмисне програмне забезпечення, яке з'явилося досить недавно. Це загрози класу Advanced Volatile Threat (AVT) більш відомі як безфайлові зловмисні шкідливі програми.

Безфайлове зловмисне програмне забезпечення [10-12] — це тип шкідливого програмного забезпечення, яке використовує законні програми для зараження комп'ютера. Це шкідливий код, який не потребує застосування файлу, що виконується у файловій системі кінцевої точки, крім уже наявних. Таке програмне забезпечення часто послуговується системними процесами операційних систем Windows, Linux, MacOS, які доступні та довірені. Зазвичай шкідливий код упроваджується в якийсь запущений процес операційної системи пристрою і виконується тільки в оперативній пам'яті. Шкідливий код, який завдав втрат, може автоматично зникати, не залишаючи навіть і сліду. Це значно ускладнює виявлення або запобігання традиційному антивірусному програмному забезпеченню та іншим продуктам для захисту кінцевих точок через невеликий розмір та відсутність файлів для сканування. Цей тип загроз не передбачає запису якихось своїх файлів на жорсткий диск. Натомість він працює з пам'яті оперативної пам'яті. Є багато способів запуску коду на пристрої без застосування файлів, що виконуються. Ось кілька прикладів: VBScript, JScript, інструментарій керування Windows (WMI), Mshta та rundll32 (або інші файли з підписом Windows, здатні запускати шкідливий код), пакетні файли PowerShell. Розглянемо дії безфайлового зловмисного програмного забезпечення на прикладі використання ним PowerShell.

Відомо [12], що PowerShell — це кросплатформне рішення для автоматизації завдань, яке має у своєму складі інтерпретатор командного рядка, скриптову мову та платформу керування конфігурацією. PowerShell підтримується у Windows, Linux та macOS. PowerShell — це сучасна командна оболонка, в якій реалізовано найкращі можливості інших популярних оболонок. На відміну від більшості оболонок, які лише приймають та повертають текст, PowerShell приймає та повертає об'єкти .NET. Це рішення пропонує такі можливості: надійний журнал командного рядка; заповнення натисканням клавіші TAB та підставлення команд; підтримання псевдонімів команд та параметрів; створення конвеєра для об'єднання команд; систему довідки в консолі, схожу на сторінки man у Unix. Скриптова мова PowerShell забезпечує розширюваність із використанням функцій, класів, скриптів та модулів; систему форматування, що розширюється, для зручного виведення; систему типів, що розширюється, для створення динамічних типів; вбудоване підтримання поширених форматів даних, зокрема CSV, JSON та XML. До того ж, PowerShell виходить за межі Windows: вона також дає можливість користувачам контролювати певні програми, такі як Microsoft Exchange, SQL Server та IIS. Тобто PowerShell дає змогу системним адміністраторам повністю автоматизувати завдання на серверах та комп'ютерах.

Отже, якщо кібератаці вдається отримати зворотний зв'язок, використовуючи інтерпретатор PowerShell, зловмисник може здобути широкі права стосовно корпоративної системи, що уможливить безперешкодне впровадження інших шкідливих програм. Тому PowerShell як системна консоль Windows (CLI) є ідеальним вектором атаки для шкідливих безфайлових програм.

Звідси постає проблема для IT-служб безпеки, як завадити зловмисникам використовувати законний інструмент операційних систем PowerShell для отримання прав адміністратора керувати комп'ютерною системою або контролювати її. Проблема захисту операційної системи Windows полягає в тому, що критичним елементом в IT-інфраструктурі системи захисту Windows є антивірусні системи. Вони не можуть ідентифікувати шкідливе безфайлове програмне забезпечення, оскільки воно сканує сигнатури вірусів на жорсткому диску. Але на такому диску немає слідів, сканери не можуть їх виявити, якщо тільки евристичне сканування пам'яті не виконується часто. Іншим критичним елементом в IT-інфраструктурі системи захисту Windows є процес перевірки безпеки під час завантаження файлів в оперативну пам'ять операційної системі. Проблема зводиться до того, що в процесі перезавантаження операційної системи безфайловий шкідливий код скидається з пам'яті, а отже, його не можна перехопити під час сканування на етапі завантаження.

Алгоритм дії безфайловного шкідливого програмного забезпечення такий. Безфайлове шкідливе програмне забезпечення потрапляє в систему контролю та керування комп'ютерною системою, після чого йому стає легше виконувати різні завдання. Наприклад, він може змінити законний процес та запровадити новий процес, ексфільтрувати дані, а також змінити адміністративні привілеї. Це робиться через набори експлойтів, які можуть бути націлені на вразливість браузера, щоб змусити браузер запускати шкідливий код, або застосовувати макроси Microsoft Word, або використовувати утиліту Microsoft PowerShell.

Експлойтом (exploit) називається будь-яка несанкціонована та протиправна атака, що здійснюється з використанням уразливості в програмному забезпеченні, мережах чи встаткуванні. Наприклад, Angler Exploit kit; Sweet Orange Exploit kit; Nuclear Exploit kit; Fiesta Exploit kit; Magnitude Exploit kit; Neutrino Exploit kit; Astrum Exploit kit; RIG Exploit kit.

Розглянемо приклад виникнення загрози дії безфайлового шкідливого програмного забезпечення на критичні об'єкти IT-інфраструктури ІС підприємства. Слід зауважити, що жертвами безфайлового шкідливого програмного забезпечення зазвичай є користувачі, зацікавлені в приховуванні якихось даних, або коли дії користувача можуть його скомпрометувати. Тому спочатку жертвами стали клієнти медичних закладів, готелів, розважальних закладів, банків США, Канади та Європи.

Процес проникнення простий. Спочатку клієнту-жертві надсилається електронний лист, що містить шкідливий документ Word. Для привернення уваги використовуються відкриті дані або дані, здобуті зі скопійованих зловмисником баз даних. Така інформація зазвичай поширена в спам-кампаніях рекламних агентств. Адже люди часто залишають номери своїх телефонів, електронні адреси, повне ім'я для отримання бонусних карток у маркетах. Тому ці електронні листи містять конкретну інформацію про жертву, наприклад номер телефону, фізичну адресу, повне ім'я людини. Часто це може викликати почуття довіри, коли її бачить жертва, а відтак здатне призвести до відкриття шкідливих документів Word або посилань відкритих вкладень.

У цих листах здебільшого пропонуються такі пропозиції для виконання жертвою дій: [Ім'я], підтвердить [Something] Подарункову картку від [Місце][ім'я]; Шановний [Ім'я] подарунковий ваучер на [Ім'я]; Будь ласка, закрийте це неоплачене зобов'язання #[Числа] [Ім'я]; Нове бронювання в [Місце] [Ім'я]; Будь ласка, сплатіть цей невиплачений залишок [ID|Ref].[Numbers] [Name]; Будь ласка, сплатіть цей прострочений платіж [ID|Ref].[Numbers].

Якщо жертва відкриє шкідливий документ Microsoft Word, прикріплений до електронного листа, то вона надає можливість шкідливому макросу, що міститься в цьому файлі, виконати шкідливій код. Тобто макрос пропонує жертві здійснити під час першого відкриття документа Microsoft Word дії, які дозволять запуск макроса залежно від налаштувань безпеки Microsoft Word. Цей макрос викликає службу WMI для створення прихованого екземпляра powershell.exe з аргументами URL-адреси. Водночас у зловмисників з'являється швидкий спосіб визначити, в якій операційній системі працює макрос: 32-бітовій або 64-бітовій. Далі макрос завантажує шкідливий файл, який починає виконуватися. Залежно від цілей вторгнення шкідливого програмного забезпечення зазвичай починається пошук потрібної інформації в комп'ютері або в мережі. Проблема в тому, що завантажений файл шкідливого програмного забезпечення стає одним із сценаріїв роботи PowerShell, що містить Shell-код. Саме цей Shell-код згодом декодується та реалізується, розшифровуючи та виконуючи вбудоване корисне для зловмисника навантаження. Фактично він здійснює низку дій на користь зловмисника, зокрема: відбувається розвідка хоста-жертви, визначається, де і які конфіденційні дані можна копіювати, проводиться кешування URL-адрес на комп'ютері-жертві. Якщо ці дані ідентифікуються як цікаві для зловмисників, вони позначаються спеціальним маркером-кодом і використовуються в наступних HTTP-запитах. Як тільки нову мету та маршрут до неї визначено, шкідливе програмне забезпечення починає переміщуватися всередині мережі до інших критичних об'єктів. Це стає можливим, адже шкідлива програма перебирає на себе керування базами даних жертви для пошуку потрібної інформації в комп'ютері-жертві, а через нього і в мережі. Після того, як розвідку хоста-жертви буде виконано, шкідливе програмне забезпечення надсилає HTTP-запит на один із серверів. Оскільки сервер ідентифікує запит як від правомірного клієнта, то він відповідає, тобто він надає зловмиснику зашифровану бібліотеку даних DLL. Ця DLL може бути розшифрована та тимчасово перезаписана на вказаний зловмисником інший диск, аби зловмисник мав змогу викликати дані в будь-який момент.

Отже, PowerShell є ідеальним інструментом переміщення в мережі для зловмисників після того, як вони скомпрометували мережу. Відсутність файлів на жорсткому диску унеможливлює виявлення безфайлового шкідливого програмного забезпечення для традиційних систем захисту, а тому така загроза є серйозною проблемою для інформаційної безпеки, наприклад для Windows Defender та інших традиційних рішень безпеки, які не можуть виявити атаку, за винятком деяких систем евристичного моніторингу.

Для захисту від безфайлових шкідливих програм потрібно виконувати такі дії [13]:

1. Відімкнути PowerShell, якщо вона не потрібна для адміністрування систем. Це може бути лише в разі, якщо адміністратор використовує інший інструмент для автоматизації своїх завдань. У цих випадках відімкнення PowerShell може стати найкращою та найдешевшою формою захисту.

2. Переконатися в тому, що комп'ютери та мережа підприємства працюють із останньою версією PowerShell, тобто PowerShell 5 та вищих, які мають додаткові засоби безпеки для Windows.

3. Вмикати лише певні функції PowerShell, що робить обмеженим режим Constrained Language. Він може зупиняти потенційно небезпечні дії, зокрема довільні виклики Windows API або деакти-

вацію певних макросів, проте він не має змоги зупинити всі типи атак.

4. Використовувати більшу кількість розширених опцій логування PowerShell, тобто автоматичну транскрипцію команд, особливо для дій, які виявились симптомом кібератаки. Додавання таких функцій транскрипції допоможе виконувати свої завдання експертного аналізу, якщо вони підозрюють наявність атаки за допомогою безфайлових шкідливих програм.

5. Застосовувати передові рішення інформаційної безпеки з опціями розширеного захисту (наприклад, превентивні технології) зокрема:

• перманентні антишкідливі служби, які доступні завдяки використанню великих даних та машинного навчання (штучного інтелекту) для виявлення атак, заснованих на аномальній поведінці;

• технологію поведінкового аналізу та виявлення індикаторів атак (Indicators of Attack, IoA) та індикаторів компрометації (Indicators of Compromise, IoC), що дасть змогу організаціям іти попереду та зупиняти запуск невідомих процесів, наприклад тих, що генерують безфайлові шкідливі програми.

6. Здійснювати пошук макросів, виявляти та підтверджувати наявність безфайлових загроз, що містять шкідливе програмне забезпечення, позначати шкідливі файли, вміти ідентифікувати це шкідливе програмне забезпечення. Оскільки це шкідливе програмне забезпечення засновано на шкідливих макросах у документах Microsoft Word, користувачам слід переконатися, що макроси не включені за замовчуванням, і обережно відкривати будь-які макроси у файлах, отриманих із ненадійних джерел.

Як можна побачити, п. 1-5 пов'язані з програмними засобами захисту. Що стосується п. 6, то це потребує окремого розгляду організаційних методів його реалізації.

Єдиний спосіб здійснити пошук макросів, виявити та підтвердити наявність безфайлових загроз — це проаналізувати код, що працює в пам'яті. Аналіз пам'яті дає змогу фахівцям із безпеки виявляти безфайлові шкідливі програми, отримувати інформацію про те, як їх було розгорнуто, та визначати збитки. Також безфайлові загрози можна шукати за точкою входу, котра вказує, як безфайлове шкідливе програмне забезпечення може потрапити в критичний об'єкт (персональний комп'ютер або серевер). Тобто треба розуміти, що вони можуть надходити через експлойт, через скомпрометоване обладнання або через регулярне виконання застосунків та скриптів. Усе це можна здійснити завдяки заходам щодо перевірки безпеки ІС підприємства.

Існує велика кількість способів перевірки безпеки ІС підприємства, серед яких застосування аудитів ступеня їх небезпеки. Аудит — це незалежні перевірка та оцінювання ступеня небезпеки ІС підприємства сторонніми фахівцями чи компаніями.

Згідно з ISO 19011:2011 аудит інформаційної безпеки підприємства — найбільш загальна форма систематичного, незалежного і документованого процесу оцінювання стану інформаційної безпеки об'єкта аудиту і об'єктивного їх аналізу з метою встановлення ступеня відповідності критеріям аудиту. Аудит здійснюється на відповідність будь-яким вимогам, сформульованим як зацікавленими особами, так і нормативними документами. Аудит може охоплювати реалізацію різних способів тестування підсистем і процесів об'єкта аудиту, аналіз документації та інших інформаційних джерел, інтерв'ювання фахівців тощо. Одним із пунктів перевірки безпеки ІТС організації, що може входити до аудиту безпеки, є тестування на проникнення [17].

Звідси створення платформ кібербезпеки для проведення перевірки захищеності інтелектуальних систем, що забезпечує всебічну видимість загроз, виявляє поверхню атаки, що постійно змінюється, дозволяючи фахівцям із безпеки розуміти і визначати пріоритети вразливостей, відстежувати загрози і швидко реагувати на атаки, а також застосовувати правильні заходи безпеки в потрібний час для зниження ризиків — є нагальним та актуальним завданням. У цьому контексті застосування методології Red Teaming та методів тестування на проникнення залишається одним із перспективних напрямів. Але потрібно розуміти, як їх ефективніше застосовувати.

Сучасна методології Red Teaming має набір методів, що забезпечують підвищення безпеки цільової системи. Перевагою методології Red Teaming є можливість описати процеси захисту чи сценарії атак, які перевіряють поточну безпеку системи організації, намагаючись зламати її як справжній хакер. Завдяки цим сценаріям атак можна візуалізувати стратегію безпеки системи та її реакцію на атакувальника, забезпечуючи ширше уявлення про стан безпеки організації. Методологія Red Teaming охоплює процеси тестування на проникнення, соціальну інженерію, фізичне вторгнення, експлуатацію прикладного рівня та експлуатацію мережних служб. Вона допомагає класифікувати всі пов'язані активи відповідно до рівня їх ризику, виявляючи всі проблеми безпеки та лазівки, присутні в системі. Це також сприяє максимізації віддачі від інвестицій, зроблених для забезпечення безпеки організації.

Процедури тестування на основі методів Red Teaming мають багато варіантів, кожен з яких підходить для різних умов або галузей. Це дає змогу оцінювати систему захисту організації, піддаючись кільком кібератакам, а також допомагає організації дізнатися, наскільки безпечні її політики. Для реалізації такого підходу компанії або звертаються до власної IT-команди, щоб взяти на себе роль хакерів, або звертаються до зовнішньої групи експертів для отримання об'єктивної та детальної інформації. Отже, методи Red Teaming допомагають оцінити, наскільки добре працює система безпеки організації під час атаки.

Водночас є методи тестування на проникнення (*penetration testing*, *pentest*). Пентест — це комплекс заходів тестування на проникнення, які імітують реальну атаку на мережу або застосунок. Мета пентесту — зрозуміти, чи може гіпотетичний зловмисник зламати систему. Для цього тестувальники самі намагаються її зламати або здобути контроль над даними. Фахівця, який проводить випробування на проникнення, називають пентестером. Процес тестування на проникнення охоплює активний аналіз системи на наявність потенційних уразливостей, які можуть спровокувати некоректну роботу цільової системи або повну відмову в обслуговуванні. Аналіз ведеться з позиції потенційного атакувальника і може містити активне використання вразливостей системи. Мета випробувань на проникнення – оцінити можливість його здійснення та спрогнозувати економічні втрати внаслідок успішної реалізації атаки. Випробування проникнення є частиною аудиту безпеки. Отже, результатом проведення випробування на проникнення зазвичай є звіт, що містить виявлені в процесі аналізу вразливості та рекомендації щодо їх усунення.

В основі випробувань на проникнення можуть бути використані різні методики. Головними їх відмінностями є наявність інформації про систему, що досліджується. Вона може бути закритою (напівприкритою), відкритою або цільовою. На етапі перевірки закритих систем (систем типу «чорний ящик») атакувальник не має початкових відомостей про пристрій цілі, який атакують. Початкове завдання такого виду перевірки — збирання потрібної інформації щодо розташування цільової системи, її інфраструктури. Під час перевірки відкритих систем (доступна повна інформація про цільову систему або є лише часткова інформація) атакувальник може мати деякі початкові відомості про пристрій цілі, що атакується. До цільових систем належать комп'ютерні системи з доступом до інтернету. Випробування проникнення має здійснюватися до запуску цільової системи масового використання. Це дає певний рівень гарантії, що будь-який атакувальник не зможе завдати шкоди досліджуваній системі.

Звідси випливає, що методи Red Teaming та тестування на проникнення, маючи свої переваги та недоліки, найкраще підходять для досягнення конкретних цілей. Наприклад, методологія Red Teaming прагне якнайшвидше проникнути всередину й отримати доступ до конфіденційної інформації. Для цього застосовуються методи, що імітують дії хакера та намагаються уникнути виявлення. Проте тестування на проникнення має тенденцію виявляти якомога більше можливих ризиків або вразливостей та прогалин у конфігурації безпеки в певний час для системи. Тобто використовує виявлені проблеми та оцінює ризик, пов'язаний з уразливістю.

Відповідно до досліджень [14-16] процес тестування на проникнення зазвичай відбувається протягом одного-двох тижнів, тоді як здобуття результатів за допомогою методології Red Teaming може тривати до трьох-чотирьох тижнів. При цьому Red Teaming не шукає численних уразливостей у вашій системі. Натомість кожна атака набирає способу мислення хакера, який має обмежений час, щоб знайти і відразу скористатися доступними вразливостями, які допоможуть їм досягти своїх цілей.

Отже, методологія Red Teaming — це практика енергійного тестування політик, планів, систем та припущень безпеки за допомогою змагального підходу. Моделювання атак хакерів робить методологію Red Teaming більш надійною, оскільки вона виявляє вразливості системи і реалізує її можливу експлуатацію як хакера. Комбінуючи такі процеси там, де це потрібно, методологія Red Teaming зламує цифрову безпеку компанії, щоб з'ясувати її найгірші сторони. Водночас тестування на проникнення є вибором для організації, безпека якої перебуває на початковому етапі. Однак, якщо компанія шукає більш кращих політик безпеки та заходи щодо посилення безпеки, то тут треба застосовувати інші методи Red Teaming.

Висновки

Таким чином, за допомогою спам-кампаній рекламних агентств зловмисники мають змогу доставляти на електронну адресу потенційних жертв електронні листи бонусних карток маркетів, до яких прикріплено шкідливі документи типу Word. Електронні листи містять конкретну інформацію про жертву, спричинюючи почуття довіри, коли її бачить жертва, та прохання відкрити документи Word або посилання відкритих вкладень, одразу запускаючи макроси. Далі макрос завантажує шкідливий файл, який заповнює оперативну пам'ять операційної системи файлом шкідливого

програмного забезпечення, що стає одним із сценаріїв роботи PowerShell, яка містить Shell-код. Тому безфайлове шкідливе програмне забезпечення слід розглядати як серйозну загрозу. Шкідливі скрипти PowerShell іноді важко виявити та ще важче розслідувати, оскільки вони можуть залишати мало цифрових криміналістичних слідів. Щоб розгорнути зловмисне програмне забезпечення, зловмисникам, як і раніше, потрібно спочатку скомпрометувати машину або через крадіжку облікових даних, або завдяки використанню експлойтів.

Отже, постає потреба у створенні платформ кібербезпеки для проведення перевірки захищеності ІС підприємства, яка забезпечує всебічну видимість, виявляє поверхню атаки, що постійно змінюється, даючи змогу фахівцям із безпеки розуміти і визначати пріоритети вразливостей, виявляти загрози і швидко реагувати на атаки, а також застосовувати правильні заходи безпеки в потрібний час для зниження ризиків. У цьому контексті застосування методології Red Teaming та методів тестування на проникнення залишається одним із перспективних напрямів. Методологія Red Teaming — це практика енергійного тестування політик, планів, систем та припущень безпеки за допомогою змагального підходу. Моделювання атак хакерів робить методологію Red Teaming більш надійною, оскільки вона виявляє вразливості системи і реалізує її можливу експлуатацію як хакера. Комбінуючи такі процеси там, де є в цьому потреба, методологія Red Teaming зламує цифрову безпеку компанії, щоб з'ясувати її найгірші сторони. Водночас тестування на проникнення є вибором для організації, безпека якої перебуває на початковому етапі. Однак, якщо компанія шукає більш кращих політик безпеки та заходи щодо посилення безпеки, то тут треба застосовувати інші методи Red Teaming.

Список використаної літератури

*1. **Даник Ю. Г., Катков Ю. І., Пічугін М. Ф.** Національна безпека: запобігання критичним ситуаціям: монографія. Житомир: Рута, 2006. 386 с.*

*2. **Катков Ю. І.** Методи, моделі та технології оцінки критичних ситуацій в інтелектуальній інформаційній інфраструктурі на основі когнітивних методів: дис. на здобуття наук. ступеня доктора техн. наук: [спец.] 05.13.06 «Інформаційні технології». Київ, 2021. 400 с.*

*3. **Катков Ю. І.** Аналіз причин критичних ситуацій в інформаційно-інтелектуальних системах // Зв'язок. 2018. №3(133). С. 12–19.*

*4. **Діогенес Ю.** Кібербезпека. Стратегії атак та оборони. Київ: Вид-во ДМК Пресс, 2016. 327 с.*

*5. **OWASP.** Application Threat Modeling [Електронний ресурс]. URL: https://owasp.org/wwwcommunity/Application_Threat_Modeling (дата звернення: 04.12.2022).*

*6. **Agile Modeling.** Security Threat Models: An Agile Introduction [Електронний ресурс]. URL: http://www.agilemodeling.com/artifacts/securityThreatModel.htm (дата звернення: 4.12.2022).*

*7. **Guzman A., Gupta A.** IoT Penetration Testing Cookbook: Identify Vulnerabilities and Secure your Smart Devices. Packt Publishing, 2017. P. 34–35.*

*8. **Shostack A.** Threat Modeling: Designing for Security. Adam Shostack. Wiley, 2014. 624 p.*

*9. **Versprite.** Application Threat Modeling Helping Clients Learn & Build Risk-Based Threat Models [Електронний ресурс]. URL: https://versprite.com/security-offerings/appsec/application-threat-modeling/ (дата звернення: 10.12.2022).*

*10. **Threatmodeler.** Getting Started with Threat Modeling: How to Identify Your Mitigation Strategy [Електронний ресурс]. URL: https://threatmodeler.com/getting-started-with-threat-modeling-how-to-identify-your-mitigation-strategy/ (дата звернення: 10.12.2022).*

*11. **Locking** Down PowerShell to Foil Attackers: 3 Essentials [Електронний ресурс]. URL: https://www.databreachtoday.com/locking-down-powershell-to-foil-attackers-3-essentials-a-10662 / (дата звернення: 10.12.2022).*

*12. **What** is PowerShell? [Електронний ресурс]. URL: https://learn.microsoft.com/ru-ru/powershell/scripting/overview?view=powershell-7.3 (дата звернення: 10.12.2022).*

*13. **PowerShell** is a great attack vector for fileless threats More details? [Електронний ресурс]. URL: https://www.securitylab.ru/blog/company/PandaSecurityRus/345805.php (дата звернення 10.12.2022).*

*14. **Barwise I.** The Red Team Guide. Київ: Вид-во PEER-LYST, 2016. 241 с.*

*15. **What** is Red Teaming? Benefits & Methodology. Updated on: March 9, 2022 // [Електронний ресурс]. URL: https://www.getastra.com/blog/security-audit/red-team-methodology/#:~:text=Red%20Team%20Methodology%20gives%20a,system%20against%20a%20real%20cyberattack*

*16. **Things** Every Red Team Needs to Optimize Operations //[Електронний ресурс]. URL: https://www.netspi.com/resources/tip-sheets/5-things-every-red-team-needs-to-optimize-operations/*

*17. **ISO 19011:2011.** Настанови щодо проведення аудитів систем менеджменту. Чинний з 24-02-12. Київ: Міжнародна організація зі стандартизації, 2013. 45 с.*

Yu. I. Katkov, O. V. Zinchenko, S. S. Tsibulnik, Yu. O. Vitenko

MEASURES TO COUNTER THREAT TO INTELLIGENT SYSTEMS FROM FILELESS MALWARE

The article is devoted to the topical issue of finding means of countering threats to intelligent systems from fileless malware. The task: in order to timely localize and minimize possible damages from the impact of threats (vulnerabilities, attacks) from fileless malware on critical IT objects — the infrastructure of the enterprise's intelligent systems, it is necessary to determine the best direction for the creation of methods for their timely detection. The article shows that the enterprise's intellectual system has vulnerability and criticality from the weak influence of threats (vulnerabilities, attacks) of the attacker on critical objects of the IT infrastructure of the enterprise. As a result of this, the contradiction between the complexity of the methods of protecting objects of the critical IT infrastructure of the enterprise from malicious software, including fileless malware, and their effectiveness in terms of timely localization and minimizing possible damage from the influence of threats is exacerbated. But the resolution of this contradiction is possible due to the creation of flexible organizational structures for monitoring synergistic representations of the process of adaptation of the enterprise's intelligent systems to threats from fileless malware. It is shown that, first of all, there is a need for constant improvement of the methods of detecting the impact of threats in the direction of their prediction. To do this, based on the analysis of the algorithm of the fileless malware through exploit sets, malicious Microsoft Word macros and compromised network equipment, a mechanism for influencing PowerShell of Windows, Unix. operating systems was determined. Actions to protect against fileless malware are proposed based on this mechanism. To implement these actions, it is shown how it is possible to search for macros, detect and confirm the presence of fileless threats. It is proposed to apply the verification of information security of the enterprise using audit methods. The application of the Red Teaming hacker attack simulation methodology and penetration testing methods is proposed as the main audit method. Consider ways to use them.

Keywords: threat model; open source; fileless software; PowerShell.