https://doi.org/10.31891/2219-9365-2024-80-46
УДК 004.056:004.852:004.75

КАШТАЛЬЯН Антоніна
Хмельницький національний університет
https://orcid.org/0000-0002-4925-9713
e-mail: antonina.kashtalian@gmail.com

ОСОБЛИВОСТІ ПРАВИЛ ФОРМУВАННЯ ВАРІАНТІВ ЦЕНТРАЛІЗАЦІЇ ДЛЯ МУЛЬТИКОМП'ЮТЕРНИХ СИСТЕМ В КОРПОРАТИВНИХ МЕРЕЖАХ

Для забезпечення виявлення зловмисного програмного забезпечення та комп'ютерних атак архітектура мультикомп'ютерних систем антивірусних комбінованих приманок і пасток повинна мати механізми для перебудови її окремих частин, центру системи і в цілому всієї архітектури самостійно без залучення адміністратора. В роботі проаналізовано методи і системи, які дають змогу здійснити перебудову архітектури систем. Особлива увага приділена перебудові центру системи. Цей напрям досліджень визначено як перспективний.

В роботі деталізовано потенційно можливі варіанти централізації в архітектурі мультикомп'ютерних систем, яку задано моделлю множин її елементів та графа зв'язків між компонентами системи. Особливістю такого подання є врахування поділу компонентів на компоненти з центром системи та компоненти без центру системи. Також, задано можливі топології для центру системи і визначено кількість варіантів централізації для кожної топології.

Напрямами подальших досліджень буде розроблення на основі топологій центру мультикомп'ютерних систем.правил та методу визначення наступного варіанту централізації в системах, які б забезпечували функціонування систем при перебудові їх архітектури без залучення адміністратора.

Ключові слова: централізація, мультикомп'ютерні системи, deception-системи, зловмисне програмне забезпечення, комп'ютерні атаки, приманки, пастки.

KASHTALIAN Antonina
Khmelnytskyi National University

FEATURES OF THE RULES OF FORMATION OF CENTRALIZATION OPTIONS FOR MULTICOMPUTARY SYSTEMS IN CORPORATE NETWORKS

In order to identify malicious care and computer attacks, multi -computer systems of antivirus combined baits and traps must have mechanisms for restructuring its individual parts, the center of the system, and in general, the entire architecture alone without involving the administrator. The work analyzes the methods and systems that make it possible to restructure the architecture of systems. Particular attention is paid to the restructuring of the system center. This area of research is defined as promising.

The work details potentially possible variants of centralization in the architecture of multi -computer systems, which is given a model of the set of its elements and a column of communication between system components. The peculiarity of such a submission is to take into account the division of components into components with the center of the system and components without the center of the system. Also, possible topologies for the center center and determined the number of centralization options for each topology.

Further research will be the development on the basis of the topologies of the Center of Multicomputary Systems. Rights and the method of determining the next variant of centralization in systems that would ensure the functioning of systems in the restructuring of their architecture without involving the administrator.

Keywords: centralization, multicomputary systems, deceps, malicious software, computer attacks, baits, traps

ПОСТАНОВКА ПРОБЛЕМИ У ЗАГАЛЬНОМУ ВИГЛЯДІ ТА ЇЇ ЗВ'ЯЗОК ІЗ ВАЖЛИВИМИ НАУКОВИМИ ЧИ ПРАКТИЧНИМИ ЗАВДАННЯМИ

Архітектура мультикомп'ютерних систем антивірусних комбінованих приманок і пасток, яку синтезовано [1-3] згідно їх визначальних характеристичних властивостей, потребує деталізації в частині організації внутрішніх складових частин таких систем та загальної організації для забезпечення ефективного функціонування таких систем. Кожна з включених визначальних характеристичних властивостей в архітектуру [4-6] мультикомп'ютерних систем потребує розроблення функційного подання для забезпечення реалізації так і для поєднання з рештою визначальних властивостей. З цих визначальних властивостей особливе місце займають контролер та централізація. Вони першочергово впливають на роботу таких систем через прийняття ними рішень. Тому, в контексті синтезу мультикомп'ютерних систем антивірусних комбінованих приманок і пасток розглянемо забезпечення централізації.

АНАЛІЗ ДОСЛІДЖЕНЬ ТА ПУБЛІКАЦІЙ

Системи приманок розгортаються різними організаціями для захисту своїх систем від зовнішніх та внутрішніх загроз. Приманки є одним з найуспішніших методів збору патернів атак з метою їх аналізу та ідентифікації [7]. Системи приманок є ефективними для захисту різних типів цільових систем від атак різних типів. З розвитком сучасних технологій розширюється спектр вразливостей сучасних систем, зокрема корпоративних мереж, баз даних, пристроїв IoT, хмарних сервісів, веб-сервісів, промислових мереж тощо.

Системи приманок здатні перешкоджати розвідці мережі, затримуючи методи сканування зловмисниками, мінімізуючи вплив на продуктивність цільового мережевого трафіку, така система затримує пошук вразливих хостів зловмисниками [8]. Певні типи атак вимагають проактивного підходу до безпеки, [9] система проактивного використання приманок, керовану центральним вузлом, що дозволяє додавати, видаляти, підтримувати та оновлювати приманки, а також аналізувати загрози

Використання мереж приманок робить їх використання більш безпечним, в роботі [10] досліджено використання різних типів приманок для спрощення їх розгортання та керування. Ефективність використання приманок залежить від їх розгортання, в тому числі як засіб захисту поєднує обманні вузли приманок з робочими вузлами, щоб примусити зловмисників виявляти свої атаки, одночасно мінімізуючи втрати для системи захисту [11]. Цільові та обманні об'єкти мають суттєві ознаки, за якими зловмисник може їх відрізнити, тому важливою задачею є розпоробка приманок, які важко відрізнити від реальних вузлів, це досягається завдяки зміні характеристик цільових та обманних об'єктів. В роботі [12] запропоновано метод кіберобману на основі теоретико-ігрової моделі двостороннього обману, яка дозволяє отримувати стратегії ефективного приховування приманок. Для покращення захисту та приховування приманок також пропонується використовувати двосторонній обман, ідея якого полягає в маніпуляції приманок та цільових вузлів таким чином, що цільова система отримує ознаки схожості із приманками, а приманки виконуються схожими на цільові вузли. Схожість досягається модифікацією цільових систем та примано [13].

Статична схема конфігурування приманок немає можливості адаптації до умов середовища, яке постійно змінюється, цей недолік долають віртуальні мережі з віртуальними приманками, які підтримують гнучку схему керування та мають можливість модифікації після розгортання [14]. Для розгортання приманок використовують методи контейнеризації[15], що дозволяють динамічно створювати мережі приманок для хмарних сервісів та забезпечувати оманливе середовище для зловмисників, що дозволяє захистити сервіси від зловмисників та моніторити їх активність. В роботі [16] запропоновано активний спільний захист між приманкою та хмарною платформою для виявлення розподілених DoS атак та захисту від них, зокрема для атак IoT пристроїв. В роботі [17] запропоновано приманку на основі програмовано визначеної мережі HoneyProxy, яка запобігає цільовим атакам на приманку, внутрішньому розповсюдженню шкідливих програм в мережах приманок та відсутності переходу на приманку, завдяки режимам багато адресної передачі та ретрансляції

Інтеграція систем виявлення вторгнень та мереж приманок зменшує частоту хибно позитивних спрацювань. Система виявлення вторгнень, що включає трьохетапну обробку трафіку, фільтрацію, виявлення вторгнень та аналіз приманкою, забезпечує виявлення атак внутрішні та зовнішніх зловмисників та підтримує продуктивність системи [18]. Для ефективного захисту системи виявлення вторгнень повинні мати здатність масштабуватися до потреб великих мереж та можливих загроз масованих атак. В роботі [19] пропонується колаборативна система виявлення вторгнень, в складі якої містяться централізовані та розподілені компоненти для збору та обміну даними між вузлами системи.

ВИКЛАД ОСНОВНОГО МАТЕРІАЛУ

Задамо модель $M_{A^{\mathfrak{S}}}^{\mathfrak{S}}$ системи $A^{\mathfrak{S}}$ класу $\mathfrak{S}$ [1, 6] так:

$$M_{A^{\mathfrak{S}}}^{\mathfrak{S}} = \left(A_1^{\mathfrak{S}}, A_2^{\mathfrak{S}}, \mathrm{G}_{A^{\mathfrak{S}}}^{\mathfrak{S}}\right), \tag{1}$$

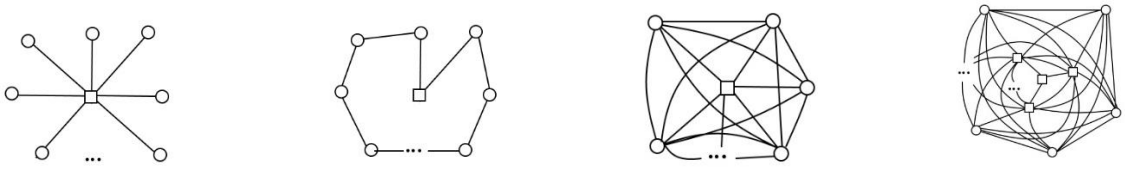
де $\mathrm{G}_{A^{\mathfrak{S}}}^{\mathfrak{S}}$ – граф, в якому вершинами є компоненти системи $A^{\mathfrak{S}}$.

В моделі $M_{A^{\mathfrak{S}}}^{\mathfrak{S}}$, яку задано згідно [1, 6] розділено елементи підмножин $A_1^{\mathfrak{S}}$ та $A_2^{\mathfrak{S}}$, що відповідає поділу компонент центру системи і компонент, які не відносяться до компонент з центром системи в поточний момент функціонування системи. Підмножини $A_1^{\mathfrak{S}}$ та $A_2^{\mathfrak{S}}$ в певні поточні моменти часу можуть бути порожніми обидві одночасно або одна з них. Крім того, компоненти системи $A^{\mathfrak{S}}$, які були задані для її формування і які в поточний момент часу перебувають у вимкнених комп'ютерних станціях, не приймають участі в функціонуванні системи, тому їх не розглядатимемо в контексті завдання формування варіанту централізації.

Задамо граф $\mathrm{G}_{A^{\mathfrak{S}}}^{\mathfrak{S}}$ так, щоб ним відображалась топологія системи та централізація в ній.

Розглянемо варіант формування централізації в системі за наявності елементу $\mathrm{m}_{\mathfrak{B}_2,\mathrm{centr},\mathrm{v}_1,1}$, який відображає централізовану архітектуру систем класу $\mathfrak{S}$. Тоді, граф $\mathrm{G}_{A^{\mathfrak{S}}}^{\mathfrak{S}}$ зобразимо на рис. 1 з урахуванням ієрархії та без ієрархії серед компонент центру, розподілення центру між компонентами та його цілісність, компонент центру, різні відношення компонентів між собою при встановлення зв'язку між ними і центру зокрема тощо. На рисунку вершини графа для компонент з центром системи позначимо квадратами, вершини для компонент без центру системи – колами.

Зображені на рис. 1 а)-г) чотири варіанти саме централізованої архітектури системи не є усіма можливими варіантами, а лише зображають частину основних з них і є необхідними для забезпечення розмежування цієї централізованої архітектури від решти варіантів централізації з можливою архітектурою в системах, які задані елементами $m_{\mathfrak{V}_2,centr,v_1,2}$, $m_{\mathfrak{V}_2,centr,v_1,3}$, $m_{\mathfrak{V}_2,centr,v_1,4}$ з множини $M_{\mathfrak{V}_2,centr,v_1}$.



а) **б)** **в)** **г)**

Рис. 1. Варіанти централізованої архітектури в системі A^{Θ}:
а) центр в одній компоненті, зв'язок компонентів системи тільки винятково з одним центром, топологія «зірка»; б) центр в одній компоненті, зв'язок компонентів системи тільки винятково з сусідніми компонентами, топологія «кільце»; в) центр в одній компоненті, зв'язок компонентів системи з одним центром та між собою; г) центр розподілений в декількох компонентах, компоненти з центром містяться на двох рівнях ієрархії, зв'язок компонентів системи тільки з компонентами з центром другого рівня ієрархії.

Але такі варіанти централізації не збіжні з варіантом часткової централізації, бо при ній центр системи не тільки перебуває в декількох компонентах, але й навіть при його поділі між компонентами напрацьовує єдине рішення або певну кількість варіантів рішень. У варіанті ж часткової централізації в системі центр системи буде функціонувати в декількох компонентах і не в одній компоненті. Компоненти системи з центром будуть напрацьовувати рішення в кожній з активних компонент центру системи, а далі сумісно визначатимуть один або більше варіантів рішень згідно принципу децентралізації. Тобто при прийнятті остаточного рішення всі компоненти з центром системи будуть функціонувати як децентралізована система. Напрацьовані рішення центром системи будуть обов'язковими для виконання призначеним компонентам. Часткова централізація в системі A^{Θ} може бути організована в ієрархічно сформованих компонентах центру системи. Це може бути при поділі функцій центру компоненти між декількома компонентами, але при цьому таких компонент центру, для яких функції поділені між декількома компонентами, є не менше двох. Таким чином, навіть при поділі функцій компоненти центру рішень, які напрацьовані в цих компонентах, буде не менше двох. На рис. 2 зображено варіанти частково централізованих розподілених систем.

а) **б)**

Рис. 2. Варіанти частково централізованих розподілених систем
а) в кожній компоненті знаходяться і виконуються всі функції центру системи;
б) функції центру системи компоненти розподілені між декількома компонентами.

Для варіанту з елементом $m_{\mathfrak{V}_2,centr,v_1,3}$ множини $M_{\mathfrak{V}_2,centr,v_1}$ отримуємо частково децентралізовану архітектуру системи A^{Θ}. Її особливість полягає в тому, що частина компонентів системи відносяться до системи, в якій децентралізована архітектура, і рішення в них напрацьовуються за принципом децентралізації. Ці компоненти вважатимемо основними компонентами системи. Решта компонент системи, які не відносяться до основних компонент системи, пов'язані з окремими основними компонентами та для них рішення та команди надсилаються із з'єднаними з ними основних компонент. Зв'язки таких неосновних компонент з рештою компонент системи, крім однієї основної компоненти, відсутні. На рис. 3 зображено варіант системи з частково децентралізованою архітектурою.

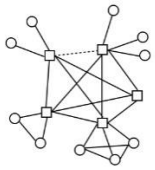
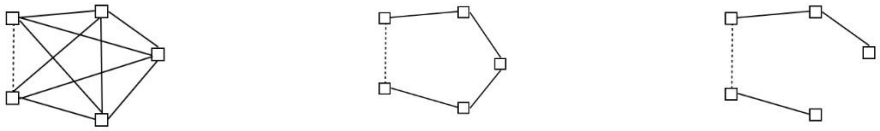


Рис. 3. Варіант частково децентралізованої розподіленої системи

Варіант системи $A^{\mathfrak{S}}$ з елементом $m_{\mathfrak{V}_2,centr,v_1,4}$ відображає децентралізовану архітектуру. Характерними особливостями систем з такою архітектурою є рівнозначність кожної компоненти як центру системи. Особливими випадками для децентралізованої архітектури зв'язки між всіма компонентами або частиною з них, встановлення ієрархії при поділі функцій центру системи між декількома компонентами тощо. На рис. 3.4 зображено варіанти системи $A^{\mathfrak{S}}$ з децентралізованою архітектурою.



а) б) с)

Рис. 4. Варіанти частково централізованих розподілених систем
а) всі компоненти є однаковими і між ними встановлено відношення «всі до всіх»;
б) всі компоненти є однаковими і зв'язки між ними встановлені згідно топології «кільце»; с) всі компоненти є однаковими і зв'язки між ними встановлені згідно топології «дерева».

Аналогічно до рис. 1 граф $G_{A^{\mathfrak{S}}}^{\mathfrak{S}}$ задано на рис. 2 – 4. Тому, модель задана формулою (1) конкретизована на рівні множин та графа зв'язків між елементами множин, які позначено двома типами вершин. Базуючись на моделі поданій так, дослідимо ефективність системи для чотирьох випадків, які формуються елементами множини $M_{\mathfrak{V}_2,centr,v_1}$.

Таким чином, зображені на рис. 1-4 приклади можливих варіантів централізації в системах класу $\mathfrak{S}$ враховують особливості елементів множини $M_{\mathfrak{V}_2,centr,v_1}$ і частково особливості певних елементів множин. Кількість можливих варіантів централізації в системах класу $\mathfrak{S}$ з їх врахуванням є суттєво більшою. Це створює проблеми зловмисникам при спробах проникнення в корпоративну мережу. Для моделі системи, яку задано за формулою (1), графи з рис. 1-4 належать множині графів $G_{A^{\mathfrak{S}}}^{\mathfrak{S}}$. Також, різні можливі варіанти графів, які задані на підмножинах $A_1^{\mathfrak{S}}$ та $A_2^{\mathfrak{S}}$, належать множині графів $G_{A^{\mathfrak{S}}}^{\mathfrak{S}}$, тому модель $M_{A^{\mathfrak{S}}}^{\mathfrak{S}}$ системи $A^{\mathfrak{S}}$ однозначно визначається підмножинами $A_1^{\mathfrak{S}}$ та $A_2^{\mathfrak{S}}$ та множиною графів $G_{A^{\mathfrak{S}}}^{\mathfrak{S}}$.

Не всі компоненти системи можуть мати між собою зв'язки. Така організація функціонування компонент в системі і комунікація передбачена можливими варіантами централізації. Тому, оскільки не всі варіанти системи передбачають наявність зв'язків між усіма компонентами, тоді ефективність організації системи в різних варіантах централізації буде відрізнятись між ними. Крім того, різні варіанти централізації в системі розподіляються за чотирма класами архітектури систем: централізовані; частково централізовані; частково децентралізовані; децентралізовані. Внутрішня організація централізації у відповідних компонентах впливатиме також на кількість зв'язків між компонентами, бо зв'язки будуть і між компонентами центру системи. Система може змінювати варіант централізації довільним чином, тобто може переходити від використання варіанту з одного класу до будь-якого з трьох класів або залишатись в тому ж класі, але з іншим варіантом централізації. Все це впливатиме на ефективність функціонування класу $\mathfrak{S}$. Тому, необхідно дослідити вплив кількості зв'язків між компонентами системи в залежності від варіантів централізації і враховувати отриманий результат для визначення наступного варіанту централізації в системі в процесі її функціонування.

Дослідимо систему за параметром кількості зв'язків між її компонентами з врахуванням різної архітектури центру системи та типу його розміщення в компонентах. Ефективність типу архітектури в системі будемо розглядати за критеріями кількості можливих варіантів центру системи та кількості зв'язків між компонентами. Наявність більшої кількості варіантів центру системи дає перевагу системі при спробі зловмисника виявити вузли комп'ютерної мережі з компонентами, в яких міститься центр системи. Тоді, системи з таким типом централізації будуть ефективніші за критерієм кількості варіантів центру системи порівняно з архітектурою, в якій кількість варіантів є меншою. Другим важливим критерієм є кількість зв'язків між компонентами. При цьому окремо потрібно розглядати компоненти з центром системи та компоненти без центру системи в поточний момент часу. Оскільки система є розподіленою, то кількість зв'язків буде впливати на оперативність при надсиланні повідомлень, команд тощо. А також буде впливати на зворотній зв'язкок між компонентами. Розподіл центру між частиною компонент може бути здійснено так: центр розподілений на окремі функції, які розміщено в різних компонентах; центр розміщено в різних компонентах і всі ці компоненти організовані згідно децентралізованої архітектури та містять однаковий функціонал центру системи. У випадку розподілу центру системи на окремі функції, які розміщено в різних компонентах, підтримка стабільного зв'язку суттєво впливатиме на функціонування системи. Тому, така організація централізації порівняно з варіантом децентралізованого функціонування компонент з центром є більш залежною від стабільності з'єднань між компонентами системи.

Встановлення кращого з чотирьох основних типів централізації в архітектурі системи дасть змогу при визначенні наступного варіанту архітектури за допомогою відповідної функції оцінювання підготувати

кращий варіант архітектури, середній та гірший. Далі контролер для вибору остаточного варіанту здійснить вибір в залежності від минулих рішень щодо вибору типу централізації в архітектурі системи. Таким чином, поділ варіантів централізації в архітектурі системи на чотири розглядувані класи дасть змогу здійснити їх цілісне оцінювання в контексті їх ефективності порівняно між собою.

Якщо в системі в поточний момент часу централізована архітектура, то може бути щонайменше два основних випадки: центр системи перебуває в одній компоненті; центр системи розподілений між декількома компонентами. У випадку розподілення центру системи між декількома компонентами, між якими розподіляються функції з центру системи, тобто окремі функції системи будуть в окремих компонентах, а разом вони формуватимуть центр системи, кількість варіантів центру системи буде залежати від кількості залучених компонент, в які буде розподілений центр системи.

Кількість варіантів централізації $\mathrm{N}_{\mathrm{centr},1}^{A^{\mathfrak{S}}}$ в системі з централізованою архітектурою обчислюємо для випадку розміщення центру системи в одній компоненті так:

$$\mathrm{N}_{\mathrm{centr},1}^{A^{\mathfrak{S}}} = \sum_{\mathrm{i}=1}^{N_{A^{\mathfrak{S}}}} \mathrm{C}_{\mathrm{i}}^{1} = \frac{N_{A^{\mathfrak{S}}}+1}{2} \cdot N_{A^{\mathfrak{S}}}, \tag{2}$$

де $N_{A^{\mathfrak{S}}}$ – загальна кількість компонент в системі $A^{\mathfrak{S}}$.

Кількість варіантів централізації $\mathrm{N}_{\mathrm{centr},2}^{A^{\mathfrak{S}}}$ в системі з централізованою архітектурою обчислюємо для випадку розміщення центру системи в декількох компонентах так:

$$\mathrm{N}_{\mathrm{centr},2}^{A^{\mathfrak{S}}} = \sum_{\mathrm{i}=2}^{N_{A^{\mathfrak{S}}}} \sum_{\mathrm{j}=2}^{\mathrm{i}} \mathrm{C}_{\mathrm{i}}^{\mathrm{j}} \cdot \mathrm{C}_{\mathrm{j}}^{1}, \tag{3}$$

де $N_{A^{\mathfrak{S}}}$ – загальна кількість компонент в системі; $N_{A^{\mathfrak{S}}} > 1$; i – кількість компонент в системі в поточний момент часу; j – кількість компонент з центром системи в поточний момент часу.

В розглядуваному варіанті централізованої архітектури кількість компонент з центром є меншою на одну компоненту від кількості всіх компонент системи, які активні в поточний момент часу. Кількість варіантів централізації $\mathrm{N}_{\mathrm{centr},12}^{A^{\mathfrak{S}}}$ в системі з централізованою архітектурою обчислюємо так:

$$\mathrm{N}_{\mathrm{centr},12}^{A^{\mathfrak{S}}} = \sum_{\mathrm{i}=1}^{N_{A^{\mathfrak{S}}}} \mathrm{C}_{\mathrm{i}}^{1} + \sum_{\mathrm{i}=2}^{N_{A^{\mathfrak{S}}}} \sum_{\mathrm{j}=2}^{\mathrm{i}} \mathrm{C}_{\mathrm{i}}^{\mathrm{j}} \cdot \mathrm{C}_{\mathrm{j}}^{1}. \tag{4}$$

Для частково централізованої архітектури системи $A^{\mathfrak{S}}$ характерним, на відміну від централізованої архітектури з поділом центру між компонентами і розподіленням між ними функцій центру, є те, що компоненти з центром системи містять однакові функціонали і прийняття рішення такими центрами здійснюється згідно принципу децентралізації за певними критеріями. Такими критеріями може буде варіант з пропонованих рішень, який отримано з більшості компонент з центром системи. Або варіант, що отримано з призначених для формування рішення компонент системи з центром. Але, незалежно від критерію формування варіанту прийнятого рішення, визначальним для такої архітектури системи є обов'язкова наявність мінімум в двох компонентах центру системи з однаковим функціоналом і їх функціонування згідно принципу децентралізації. Тоді, кількість варіантів часткової централізації в системі буде обчислюватись так:

$$\mathrm{N}_{\mathrm{centr},3}^{A^{\mathfrak{S}}} = \sum_{k=2}^{N_{A^{\mathfrak{S}}}-1} \sum_{i=2}^{N_{A^{\mathfrak{S}}}} \sum_{j=2}^{i} C_i^j \cdot C_j^k, \tag{5}$$

де $N_{A^{\mathfrak{S}}}$ – загальна кількість компонент в системі; i – кількість компонент в системі в поточний момент часу; j – кількість компонент з центром системи в поточний момент часу; k – кількість компонент, в яких може бути центр системи, тобто компоненти, в які при встановлені системи встановлено функціонал центру системи.

Для наявності саме часткової централізації в архітектурі системи і, відповідно, уникнення переходу до децентралізованої архітектури кількість компонент з центром системи в поточний момент часу $\mathrm{j} < N_{A^{\mathfrak{S}}}$. Аналогічно, $\mathrm{j} > 1$, інакше архітектура системи стане централізованою.

Частково децентралізована архітектура системи $A^{\mathfrak{S}}$ визначається компонентами з центром системи, які організовані згідно принципу децентралізації, кількість яких менше загальної кількості компонент, бо решта компонент мають з'єднання з компонентами з центром системи і отримують від них команди та повідомлення. Таким чином, кількість варіантів в системах з частково децентралізованою архітектурою обчислюємо так:

$$\mathrm{N}_{\mathrm{centr},4}^{A^{\mathfrak{S}}} = \sum_{j=2}^{N_{A^{\mathfrak{S}}}-1} \sum_{i=2}^{N_{A^{\mathfrak{S}}}} C_i^j, \tag{6}$$

де $N_{A^{\mathfrak{S}}}$ – загальна кількість компонент в системі; i – кількість компонент в системі в поточний момент часу; j – кількість компонент з центром системи в поточний момент часу.

Для уникнення переходу до повністю децентралізованої архітектури кількість компонент з центром системи в поточний момент часу $j < N_{A^{\mathfrak{S}}}$.

Можуть бути інші варіанти формування рішень компонентами з центром системи, але їх кількість за наявності різних варіантів буде фактично однаковою, тому формула (6) дає змогу обчислити кількість варіантів.

Децентралізована архітектура в системі $A^{\mathfrak{S}}$ визначається всіма компонентами з центром системи. Всі компоненти системи в такій архітектурі є однаковими і рішення приймається ними на основі певних заданих для цього критеріїв. Таким чином, кількість варіантів в системах з децентралізованою архітектурою обчислюємо так:

$$N_{centr,5}^{A^{\mathfrak{S}}} = \sum_{i=N_{A^{\mathfrak{S}}}}^{N_{A^{\mathfrak{S}}}} C_i^{N_{A^{\mathfrak{S}}}} = 1, \quad (7)$$

де $N_{A^{\mathfrak{S}}}$ – загальна кількість компонент в системі; i – кількість компонент з центром системи в поточний момент часу.

Розглянемо приклади з різною кількістю різних варіантів централізації в архітектурі системи. В табл. 2 наведено приклади та результати щодо кількості варіантів, які обчислено за формулами (2), (3), (5)-(7), для $N_{A^{\mathfrak{S}}} = 50$.

Таблиця 1

Кількість варіантів централізації в архітектурі систем

Тип централізації в архітектурі систем	Наявність поділу центру між компонентами системи	Формула	Значення кількості варіантів
Централізована	ні	(2)	1275
Централізована	так	(3)	$\sum_{i=2}^{50}\sum_{j=2}^{i} C_i^j \cdot C_j^1$
Частково централізована	так	(5)	$\sum_{k=2}^{49}\sum_{i=2}^{50}\sum_{j=2}^{i} C_i^j \cdot C_j^k$
Частково децентралізована	так	(6)	$\sum_{j=2}^{49}\sum_{i=2}^{50} C_i^j$
Децентралізована	так	(7)	1

Таким чином, загальна кількість варіантів формування центру системи, обчислення яких задано формулами (4)-(7), є прийнятною для використання в системах класу $\mathfrak{S}$, оскільки така стратегія в контексті формування центру системи в її компонентах знімає обмеження, дозволяє центру системи підібрати наступний варіант свого розміщення та варіанту архітектури. Це дає змогу створювати проблеми зловмисникам у виявленні компонент з центром системи та встановленні типу архітектури централізації з метою розуміння ним принципів її функціонування. Загальна кількість варіантів центру системи є більшою, бо залежить від поділу центру між компонентами для частково централізованої та частково децентралізованої архітектури. Розподілення центру між компонентами системи з використанням різної топології між ними збільшить кількість варіантів централізації. Крім того, на кількість варіантів впливатиме також і кількість компонент в системі, в яких не буде центру системи, оскільки всі вони разом будуть об'єктами дослідження зловмисників. Враховуючи таку різноманітність варіантів, яка відображена кількісно формулами (2), (3), (5)-(7), потребують розроблення правила, визначення наступного типу централізації архітектури в компонентах системи та визначення їх безпосереднього розміщення в конкретних компонентах.

ВИСНОВКИ З ДАНОГО ДОСЛІДЖЕННЯ І ПЕРСПЕКТИВИ ПОДАЛЬШИХ РОЗВІДОК У ДАНОМУ НАПРЯМІ

В результаті деталізовано потенційно можливі варіанти централізації в архітектурі мультикомп'ютерних систем, яку задано моделлю множин її елементів та графа зв'язків між компонентами системи. Особливістю такого подання є врахування поділу компонентів на компоненти з центром системи та компоненти без центру системи. Також, задано можливі топології для центру системи і визначено кількість варіантів централізації для кожної топології.

Напрямами подальших досліджень буде розроблення на основі топологій центру мультикомп'ютерних систем.правил та методу визначення наступного варінату централізації в системах, які б забезпечували функціонування систем при перебудові їх архітектури без залучення адміністратора.

Література

1. Kashtalian, A., Lysenko, S., Savenko, O., Nicheporuk, A., Sochor, T., & Avsiyevych, V. (2024). Multi-computer malware detection systems with metamorphic functionality. Radioelectronic and Computer Systems, 2024(1), 152-175.

2. Savenko, O., Sachenko, A., Lysenko, S., Markowsky, G., & Vasylkiv, N. (2020). Botnet detection approach based on the distributed systems. International Journal of Computing, 19(2), 190-198.

3. B. Savenko, A. Kashtalian, S. Lysenko and O. Savenko. (2023). Malware Detection By Distributed Systems with Partial Centralization," 2023 IEEE 12th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Dortmund, Germany, 265-270.

4. D. Denysiuk, O. Savenko, S. Lysenko, B. Savenko and A. Kashtalian. (2023). Method for Detecting Steganographic Changes in Images Using Machine Learning. 2023 13th International Conference on Dependable Systems, Services and Technologies (DESSERT), Athens, Greece, 1-6.

5. Lysenko S, Bobrovnikova K, Kharchenko V, Savenko O. (2022). IoT Multi-Vector Cyberattack Detection Based on Machine Learning Algorithms: Traffic Features Analysis, Experiments, and Efficiency. Algorithms, 15(7):239.

6. Kashtalian, A., Lysenko, S., Savenko, B., Sochor, T., & Kysil, T. (2023). Principle and method of deception systems synthesizing for malware and computer attacks detection. Radioelectronic and Computer Systems, 0(4), 112-151.

7. Selvaraj R., Madhava V., Marwala T. (2016). Honey Pot: A Major Technique for Intrusion Detection. 380, 73-82. 10.1007/978-81-322-2523-2_7.

8. Achleitner S., Porta T., McDaniel P., Sugrim S., Krishnamurthy S., Chadha R. (2016). Cyber Deception: Virtual Networks to Defend Insider Reconnaissance. 57-68. 10.1145/2995959.2995962.

9. Yang X., Yuan J., Yang H., Kong Y., Zhang H., Zhao J. (2023). A Highly Interactive Honeypot-Based Approach to Network Threat Management. Future Internet, MDPI, March, 15(4), 1-31.

10. Wafi H., Fiade A., Hakiem N., Bahaweres R. (2017). Implementation of a modern security systems honeypot Honey Network on wireless networks. 91-96. 10.1109/YEF-ECE.2017.7935647.

11. Zaman M., Tao L., Maldonado M., Liu C., Sunny A., Xu S., Chen L. (2023). Optimally Blending Honeypots into Production Networks: Hardness and Algorithms. Science of Cyber Security : 5th International Conference, SciSec Melbourne, VIC, Australia, July 11–14, Proceedings. Springer-Verlag, Berlin, Heidelberg, 285–304.

12. Miah, M.S., Gutierrez, M., Veliz, O., Thakoor, O., & Kiekintveld, C. (2020). Concealing Cyber-Decoys using Two-Sided Feature Deception Games. Hawaii International Conference on System Sciences. 10.24251/HICSS.2020.235.

13. Aggarwal, P., Du, Y., Singh, K., & González, C. (2021). Decoys in Cybersecurity: An Exploratory Study to Test the Effectiveness of 2-sided Deception. ArXiv, abs/2108.11037.

14. Fan W., Fernández D., Du Z. (2015). Adaptive and Flexible Virtual Honeynet. 10.1007/978-3-319-25744-0_1.

15. Ravi A., Sharma B., Mukherjee A. (2023). A Cloud-Native Honeynet Automation and Orchestration Framework. 10.31219/osf.io/xkqzr.

16. Huang, J.X., Zhou, S., Savage, N., & Zhang, W. (2021). A Distributed Cloud Honeypot Architecture. 2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC), 1176-1181.

17. Kyung, S., Han, W., Tiwari, N.K., Dixit, V.H., Srinivas, L., Zhao, Z., Doupé, A., & Ahn, G. (2017). HoneyProxy: Design and implementation of next-generation honeynet via SDN. 2017 IEEE Conference on Communications and Network Security (CNS), 1-9.

18. Agrawal N., Tapaswi S. (2017). The Performance Analysis of Honeypot Based Intrusion Detection System for Wireless Network. International Journal of Wireless Information Networks. 24. 10.1007/s10776-016-0330-3.

19. Wang K, Tong M, Yang D, Liu Y. A Web-Based Honeypot in IPv6 to Enhance Security. Information. 2020; 11(9):440.

References

1. Kashtalian, A., Lysenko, S., Savenko, O., Nicheporuk, A., Sochor, T., & Avsiyevych, V. (2024). Multi-computer malware detection systems with metamorphic functionality. Radioelectronic and Computer Systems, 2024(1), 152-175.

2. Savenko, O., Sachenko, A., Lysenko, S., Markowsky, G., & Vasylkiv, N. (2020). Botnet detection approach based on the distributed systems. International Journal of Computing, 19(2), 190-198.

3. B. Savenko, A. Kashtalian, S. Lysenko and O. Savenko. (2023). Malware Detection By Distributed Systems with Partial Centralization," 2023 IEEE 12th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Dortmund, Germany, 265-270.

4. D. Denysiuk, O. Savenko, S. Lysenko, B. Savenko and A. Kashtalian. (2023). Method for Detecting Steganographic Changes in Images Using Machine Learning. 2023 13th International Conference on Dependable Systems, Services and Technologies (DESSERT), Athens, Greece, 1-6.

5. Lysenko S, Bobrovnikova K, Kharchenko V, Savenko O. (2022). IoT Multi-Vector Cyberattack Detection Based on Machine Learning Algorithms: Traffic Features Analysis, Experiments, and Efficiency. Algorithms, 15(7):239.

6. Kashtalian, A., Lysenko, S., Savenko, B., Sochor, T., & Kysil, T. (2023). Principle and method of deception systems synthesizing for malware and computer attacks detection. Radioelectronic and Computer Systems, 0(4), 112-151.

7. Selvaraj R., Madhava V., Marwala T. (2016). Honey Pot: A Major Technique for Intrusion Detection. 380, 73-82. 10.1007/978-81-322-2523-2_7.

8. Achleitner S., Porta T., McDaniel P., Sugrim S., Krishnamurthy S., Chadha R. (2016). Cyber Deception: Virtual Networks to Defend Insider Reconnaissance. 57-68. 10.1145/2995959.2995962.

9. Yang X., Yuan J., Yang H., Kong Y., Zhang H., Zhao J. (2023). A Highly Interactive Honeypot-Based Approach to Network Threat Management. Future Internet, MDPI, March, 15(4), 1-31.

10. Wafi H., Fiade A., Hakiem N., Bahaweres R. (2017). Implementation of a modern security systems honeypot Honey Network on wireless networks. 91-96. 10.1109/YEF-ECE.2017.7935647.

11. Zaman M., Tao L., Maldonado M., Liu C., Sunny A., Xu S., Chen L. (2023). Optimally Blending Honeypots into Production Networks: Hardness and Algorithms. Science of Cyber Security : 5th International Conference, SciSec Melbourne, VIC, Australia, July 11–14, Proceedings. Springer-Verlag, Berlin, Heidelberg, 285–304.

12. Miah, M.S., Gutierrez, M., Veliz, O., Thakoor, O., & Kiekintveld, C. (2020). Concealing Cyber-Decoys using Two-Sided Feature Deception Games. Hawaii International Conference on System Sciences. 10.24251/HICSS.2020.235.

13. Aggarwal, P., Du, Y., Singh, K., & González, C. (2021). Decoys in Cybersecurity: An Exploratory Study to Test the Effectiveness of 2-sided Deception. ArXiv, abs/2108.11037.

14. Fan W., Fernández D., Du Z. (2015). Adaptive and Flexible Virtual Honeynet. 10.1007/978-3-319-25744-0_1.

15. Ravi A., Sharma B., Mukherjee A. (2023). A Cloud-Native Honeynet Automation and Orchestration Framework. 10.31219/osf.io/xkqzr.

16. Huang, J.X., Zhou, S., Savage, N., & Zhang, W. (2021). A Distributed Cloud Honeypot Architecture. 2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC), 1176-1181.

17. Kyung, S., Han, W., Tiwari, N.K., Dixit, V.H., Srinivas, L., Zhao, Z., Doupé, A., & Ahn, G. (2017). HoneyProxy: Design and implementation of next-generation honeynet via SDN. 2017 IEEE Conference on Communications and Network Security (CNS), 1-9.

18. Agrawal N., Tapaswi S. (2017). The Performance Analysis of Honeypot Based Intrusion Detection System for Wireless Network. International Journal of Wireless Information Networks. 24. 10.1007/s10776-016-0330-3.

19. Wang K, Tong M, Yang D, Liu Y. A Web-Based Honeypot in IPv6 to Enhance Security. Information. 2020; 11(9):440.