УДК 004.49
DOI https://doi.org/10.32782/2663-5941/2024.5.1/19

Денисюк Д.О.
Хмельницький національний університет

Савенко Б.О.
Хмельницький національний університет

Каштальян А.С.
Хмельницький національний університет

Іванченко О.В.
Національний технічний університет «Дніпровська політехніка»

МЕТОД ВИЯВЛЕННЯ ЗЛОВМИСНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ НА ОСНОВІ АНАЛІЗУ ПОВЕДІНКОВИХ ПАТЕРНІВ СИСТЕМИ

У статті досліджено розробку та оцінювання методу виявлення зловмисного програмного забезпечення, яке використовує стеганографічні зображення для прихованого управління системою із застосуванням технології приманок. Запропонований підхід базується на використанні методів для виявлення стеганографічних змін та методу із застосуванням приманок для виявлення зловмисного програмного забезпечення. Методи LSB і DCT були обрані через їхню здатність приховувати дані в зображеннях, що дозволило використовувати їх у стеганографічних атаках. Під час дослідження проведено серію експериментів для порівняння методів виявлення стеганографічних змін, що дозволило оцінити їх точність, чутливість та швидкість реагування на різні типи зображень, які містили приховані дані. Результати експериментів продемонстрували, що зображення, які містять приховані команди, створені за допомогою методу DCT, виявляються з більшою точністю порівняно з методом LSB.

Хоча метод виявився ефективним, одним із його обмежень є потреба у використанні двох окремих моделей для різних етапів аналізу, що може ускладнити його інтеграцію в системи з обмеженими ресурсами. Використання окремих моделей для виявлення стеганографічних змін та аналізу поведінки системи збільшує навантаження на обчислювальні ресурси, що є критичним для деяких середовищ з обмеженими можливостями. Подальші дослідження будуть спрямовані на оптимізацію підходу шляхом об'єднання цих моделей в єдину систему для зниження обчислювальних витрат і підвищення ефективності. Також, важливим напрямком майбутніх досліджень є інтеграція цього методу з існуючими системами захисту інформації. Це дозволить забезпечити комплексний підхід до захисту інформаційних систем від сучасних загроз, які використовують стеганографічні техніки для приховування своєї активності. Інтеграція з існуючими рішеннями у сфері захисту інформації також дозволить автоматизувати процес виявлення загроз та підвищити загальний рівень захисту системи від стеганографічних атак.

Ключові слова: *зловмисне програмне забезпечення, захист інформації, моніторинг аномалій, стеганографічне зображення, приманки, програмні пастки.*

Постановка проблеми. З розвитком веб-сервісів та веб-ресурсів питання їхнього захисту від несанкціонованого доступу стає все більш актуальним. Однією з ключових загроз є програмні імпланти та бот-мережі, які можуть порушити безпеку систем. Хоча ці атаки мають певні спільні риси, але програмні імпланти [1, с. 82] виділяються своєю здатністю використовувати специфіку програмного забезпечення для реалізації шкідливих дій. Зокрема, імпланти можуть експлуатувати бібліотеки, які розробники інтегрують у свій код, або ж використовувати спеціально залишені фрагменти коду подвійного призначення. Наприклад, код, призначений для обробки зображень з метою їх оптимізації, може бути таємно налаштований для виконання шкідливих команд, вбудованих у саме зображення з використанням технології стеганографії. Цей підхід стає дедалі популярнішим серед зловмисників через його приховану природу і складність виявлення.

Оскільки зображення, які містять стеганографію, не можуть бути виявлені звичайним аналізом коду.

Починаючи з 2021 року, загроза програмних імплантів для систем суттєво зросла, особливо у зв'язку з уразливістю програмного забезпечення в ланцюгах постачання [2, с. 275]. Програмні імпланти дозволяють зловмисникам отримувати віддалений контроль над скомпрометованими системами, обходити захист, викрадати дані та запускати додаткові шкідливі програми. Наприклад, під час атак SolarWinds [3, с. 2356] і DNS-угонів, зловмисники використовували програмні імпланти для довготривалого кібер-шпигунства, зокрема для викрадення облікових даних та віддаленого завантаження шкідливих файлів. Інший аспект – це зловмисні програми, які встановлюються через компрометовані ланцюги постачання [4, с. 180] або під час передачі систем між виробником та кінцевим користувачем. Це дозволяє зловмисникам імплантувати зловмисне ПЗ ще до того, як система потрапить до жертви, що значно ускладнює виявлення таких загроз. Таким чином, програмні імпланти несуть велику загрозу для безпеки даних користувача і потребують сучасних методів виявлення таких загроз.

Аналіз останніх досліджень і публікацій. Програмні імпланти є специфічним типом кібератак, для ефективного виявлення яких доцільно використовувати honeypot-системи. Вони класифікуються за рівнем взаємодії на низький, середній та високий [1, с. 6]. Низькорівневі системи емулюють окремі сервіси з мінімальними витратами ресурсів, але збирають обмежені дані [2, с. 30]. Високорівневі системи імітують повноцінні середовища, що дозволяє глибокий аналіз зловмисних дій [3, с. 15228; 4, с. 92]. Середній рівень пропонує баланс між ефективністю та ресурсозатратами [5, с. 475]. Honeypot-системи поділяються на виробничі та дослідницькі. Виробничі використовуються для захисту реальних мереж та виявляють атаки в режимі реального часу [6, с. 522]. Наприклад, Cowrie імітує SSH та Telnet-сервери для збору даних про зловмисну активність [7, с. 6]. Дослідницькі системи, такі як Dionaea, фокусуються на вивченні технік атак [8, с. 87]. Крім того, honeypot-системи застосовуються для виявлення ботнетів за допомогою методів DNS-тунелювання та машинного навчання, що підвищує рівень кібербезпеки [9, с. 25; 10, с. 89]. Розгортання honeypot-систем можливе в різних середовищах, таких як хмарні платформи, демілітаризовані зони (DMZ) або мережі IoT, кожне з яких має свої переваги та виклики [11, с. 3]. Наприклад, у мережах IoT honeypot-системи допомагають ідентифікувати загрози, але потребують точного налаштування [12, с. 340]. Вибір архітектури honeypot-систем, зокрема вдосконалені сенсори другого та третього поколінь, підвищує стійкість до виявлення зловмисниками [13, с. 520]. Зростає кількість атак, що використовують стеганографію для приховування зловмисного коду в зображеннях, що потребує нових підходів для ефективного виявлення та протидії [14, с. 3; 15, с. 67].

Постановка завдання. Метою роботи є створення підходу для виявлення зловмисного програмного забезпечення (ЗПЗ), яке використовує стеганографічні зображення для передачі команд, з акцентом на збільшення точності його ідентифікації на етапі завантаження. Використання методів приманок є ключовим компонентом цього підходу, оскільки вони допомагають імітувати інфраструктуру, націлену на зловмисне ПЗ, що дозволяє перехопити та виявити активність, пов’язану зі стеганографічною передачею команд. Такий підхід має на меті підвищити ефективність виявлення завдяки імітації сценаріїв, у яких зловмисне ПЗ активується при завантаженні зображення, тим самим дозволяючи виявляти приховані загрози ще на ранніх етапах їх діяльності.

Виклад основного матеріалу. Для виявлення ЗПЗ, яке керується через зображення, необхідний комплексний підхід. Перш за все, важливо застосувати методи виявлення стеганографії в зображеннях, щоб встановити, чи може дане зображення містити команди для ЗПЗ. Це дозволить ефективно використовувати ресурси сервера та уникнути зайвого навантаження на систему перевірками. Архітектуру системи виявлення ЗПЗ в зображеннях, засновану на роботі спостерігача, представлено на рис. 1.

Процес роботи методу починається після того, як користувач завантажує зображення на вебсервер. Після завантаження зображення формату PNG, JPG або JPEG воно перетворюються на матрицю каналів R – червоний, G – зелений, B – синій та A – альфа. Також зберігаються розміри зображення – ширина та висота в пікселях. Після завантаження і декодування кожен піксель представляється 8-бітними значеннями інтенсивності каналів. Математично зображення можна представити наступним чином:

$$I = \left\{ P_{ij} \mid 1 \le i \le H, 1 \le j \le W \right\}, \qquad (1)$$

де I – вхідне зображення, W – його ширина, H – його висота. P_{ij} – піксель з координатами (i, j),

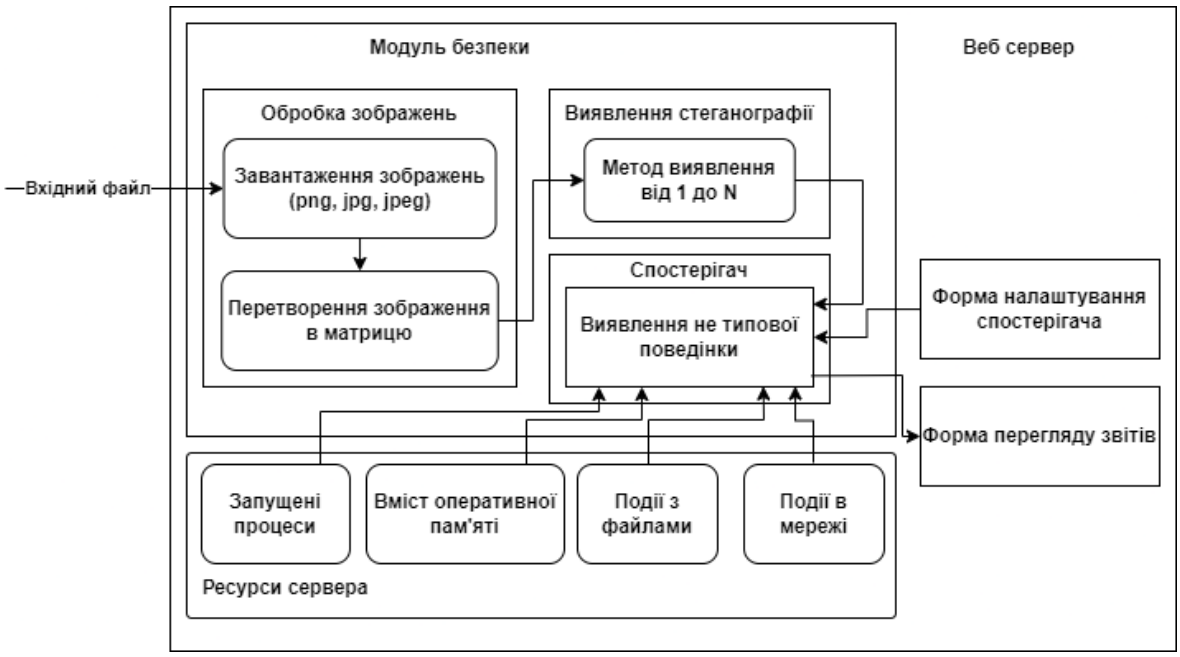

Рис. 1. Архітектура системи виявлення ЗПЗ

який складається з чотирьох компонент $P_{ij} = (R_{ij}, G_{ij}, B_{ij}, A_{ij})$, де $R_{i,j}, G_{i,j}, B_{i,j}, A_{i,j}$ – значення інтенсивностей червоного, зеленого, синього каналів та альфа-каналу відповідно.

Отримана матриця є основою для подальших обчислень, зокрема для застосування алгоритмів виявлення стеганографії. У даному дослідженні було використано два широко відомих підходи для виявлення стеганографічних змін в зображеннях. Перший підхід базується на методі опорних векторів (SVM, Support Vector Machine) [16, с. 1659], який забезпечує класифікацію зображень шляхом побудови оптимальної гіперплощини, що розділяє стеганографічні та нестеганографічні зображення. Цей метод є особливо ефективним для задач, де важлива висока точність класифікації при наявності невеликої кількості навчальних даних. Другий підхід застосовує алгоритм Random Forest [17, с. 560], який використовує ансамбль дерев рішень для аналізу численних параметрів зображення. Завдяки багатократному розподілу і усередненню результатів різних дерев, цей метод дозволяє підвищити стійкість і точність виявлення стеганографічних змін, навіть у випадках складної структури даних. Поєднання цих двох підходів дозволяє значно покращити ефективність системи виявлення, забезпечуючи надійну і точну ідентифікацію прихованих даних в цифрових зображеннях. Метод виявлення стеганографії застосовує алгоритми для аналізу матриці, перевіряючи наявність аномалій або специфічних патернів, які можуть свідчити про присутність стеганографічного контенту.

$$Y = \{M_n, | \, 1 \leq n \leq K\}, \qquad (2)$$

де Y – вихідний набір звітів, що містять інформацію про наявність або відсутність стеганографії в зображенні.

Кожен звіт містить ймовірність виявлення стеганографії, отриману за допомогою методу M_n де n – номер методу, а K – загальна кількість методів, що брали участь у перевірці. Збільшення кількості методів може призвести до зростання часу обробки. Після виявлення стеганографії метод створює спостерігача для подальшого моніторингу. Спостерігач відстежує процеси, які звертаються до зображення, збирає інформацію про виконувані операції та аналізує використання пам'яті, проводить моніторинг процесів, що звертаються до зображення, а також моніторить вихідні з'єднання. Перед початком роботи спостерігач налаштовується через конфігуратор, що дозволяє виключити безпечні процеси й уникнути хибних спрацьовувань. Конфігуратор визначає параметри для розпізнавання підозрілих дій, встановлюючи допустимі операції з зображеннями та шаблони доступу. Він враховує можливі аномалії, що можуть вказувати на стеганографію, допомагаючи спостерігачу ефективніше виявляти загрози та зменшувати хибні спрацьовування.

Нехай X – вхідне зображення з високою ймовірністю стеганографії. Спостерігач аналізує

набір процесів P , які взаємодіють із зображенням X :

$$P = \{p_1, p_2, \dots, p_n\}. \quad (3)$$

де p_i – окремий процес, що здійснює операції з зображенням. Спостерігач збирає дані про дії A , виконувані кожним процесом p_i :

$$A = \{a_1, a_2, \dots, a_j\}. \quad (4)$$

де a_j – конкретна дія, здійснена процесом p_i .

Конфігуратор C визначає множину параметрів Q , яка характеризує нормальну поведінку процесів щодо зображень:

$$Q = \{q_1, q_2, \dots, p_k\}. \quad (5)$$

де p_k – параметр, що описує допустиму дію або поведінку.

Нехай D множина підозрілих дій. Спостерігач порівнює зібрані дані A з параметрами конфігуратора Q . Якщо дії A не відповідають параметрам Q , спостерігач ідентифікує їх як підозрілі:

$$D = A - Q. \quad (6)$$

Цей підхід дозволяє провести первинний аналіз дій над зображенням для виявлення аномалій. Для прискорення виявлення нетипової поведінки можна застосувати метод співставлення шаблонів. Якщо спостерігач виявляє хоча б одну підозрілу дію, для детального аналізу використовується нейронна мережа. Подальший етап включає аналіз поведінки з використанням нейронної мережі, яка аналізує всі операції, виконувані над зображенням. Операції, що не відповідають визначеним нормам, класифікуються як підозрілі, після чого нейронна мережа виконує точніший аналіз для виявлення прихованих загроз. Метою нейронної мережі є точна ідентифікація прихованих загроз на основі вхідних даних та попередньо виявлених підозрілих дій. Застосування нейронної мережі на другому етапі аналізу значно підвищує точність виявлення стеганографічних активностей, оскільки вона здатна враховувати складні взаємозв'язки та патерни в даних, які можуть залишатися непомітними для традиційних методів. Таким чином, інтеграція первинного аналізу зі спостерігачем і глибокого аналізу за допомогою нейронної мережі забезпечує комплексний підхід до виявлення стеганографії та підвищує рівень кібербезпеки. Для обробки зображень можуть бути використані згорткові нейронні мережі CNN [18, с. 203] та рекурентні нейронні мережі RNN [19, с. 372].

Використання цього методу дозволяє виявляти та аналізувати стеганографічні активності для ідентифікації програмних імплантів в інформаційних системах. Такий підхід підвищує рівень безпеки, забезпечуючи своєчасне виявлення прихованих загроз і запобігання їхньому впливу.

Експерименти. Для проведення експериментів було підготовлено два набори даних: один містив зображення зі стеганографічними змінами, а інший – зразки ЗПЗ, яке використовувало стеганографію для передачі команд. Для створення стеганографічних зображень застосовувалися методи, такі як метод найменш значущих бітів (LSB) [20, с. 3] та дискретне косинусне перетворення (DCT) [21, с. 36458]. Ці методи були обрані через їх відносну простоту, оскільки вони не вимагають складних обчислювальних операцій з боку ЗПЗ для вилучення прихованих даних. Зокрема, метод LSB забезпечує базовий рівень приховування, тоді як метод DCT є більш стійким до виявлення та атак.

Для моделювання складних поведінкових патернів було розроблено програмне забезпечення, яке виконувало дві основні функції: змінювало розмір зображень та очікувало на отримання нових зображень із прихованими командами, які потім виконувалися. Це програмне забезпечення дозволило вивчити інтерактивну природу взаємодії між стеганографією та зловмисним кодом

Таблиця 1

Результати експериментів з виявлення стеганографічних змін

Метод	Тип	TP	TN	FP	FN	TPR (%)	FPR (%)
LSB	Без команд	120	200	30	50	70.59	13.04
		115	205	35	45	71.88	14.58
		120	200	25	40	75.00	11.11
	З командами	110	190	25	40	73.33	11.63
		105	185	20	35	75.00	9.76
DCT	Без команд	130	210	15	30	81.25	6.67
		125	195	40	50	71.43	125
	З командами	145	225	5	15	90.63	2.17
		135	215	15	25	84.38	6.52
		140	220	10	20	87.50	4.35

у реальних сценаріях. Крім того, таке середовище експериментів забезпечувало можливість оцінити ефективність методів виявлення стеганографічних змін та їх стійкість до різних атак.

Результати експериментів підтверджують ефективність методу виявлення нетипової поведінки системи у відповідь на зображення, що містять стеганографічні зміни, створені методами LSB та DCT. Зображення, що містили стеганографію, вбудовану за допомогою методу DCT, продемонстрували вищі показники TPR (True Positive Rate) та нижчі показники FPR (False Positive Rate) у порівнянні з методом LSB. Це свідчить про більшу ефективність виявлення прихованих даних, що були створені методом DCT, а також про його стійкість до хибних спрацьовувань.

Висновки. Дослідження привело до розроблення методу для виявлення ЗПЗ, яке використовує стеганографічні зображення для керування. Проведені експерименти підтвердили ефективність методу у виявленні різних типів ЗПЗ та стеганографічних змін у зображеннях. Метод швидко реагує на появу зображень зі стеганографією, що дозволяє мінімізувати потенційні ризики.

Серед обмежень методу є використання двох окремих механізмів для аналізу, що може впливатиме на системи з обмеженими ресурсами, ускладнюючи їх інтеграцію. У подальших дослідженнях планується оптимізувати метод шляхом створення єдиного рішення, що знизить вимоги до обчислювальних ресурсів і підвищить його ефективність у ресурсно обмежених середовищах. Важливим напрямком майбутніх досліджень є також інтеграція цього методу з існуючими системами захисту інформації для покращення захисту від стеганографічних атак.

Список літератури:

1. 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, 15(4), 127.

2. Bhargavi Movva, R., & Sandeep Chaitanya, N. (2021). Identification of Security Threats Using Honeypots. In Proceedings of International Conference on Advances in Computer Engineering and Communication Systems: ICACECS 2020 (pp. 273-282). Springer Singapore.

3. Franco, J., Aris, A., Canberk, B., & Uluagac, A. S. (2021). A survey of honeypots and honeynets for internet of things, industrial internet of things, and cyber-physical systems. IEEE Communications Surveys & Tutorials, 23(4), 2351-2383.

4. Fediushyn, O., Ruzhentsev, V., Fedorov, I., & Moskvin, K. (2021, October). Honeypot Data Storage and Analysis Software to Prevent Intrusions. In 2021 IEEE 8th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T) (pp. 169-173). IEEE.

5. Bartwal, U., Mukhopadhyay, S., Negi, R., & Shukla, S. (2022, June). Security orchestration, automation, and response engine for deployment of behavioural honeypots. In 2022 IEEE Conference on Dependable and Secure Computing (DSC) (pp. 1-8). IEEE.

6. Negi, P. S., Garg, A., & Lal, R. (2020, January). Intrusion detection and prevention using honeypot network for cloud security. In 2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence) (pp. 129-132). IEEE.

7. AbuOdeh, M., Adkins, C., Setayeshfar, O., Doshi, P., & Lee, K. H. (2021). A Novel AI-based Methodology for Identifying Cyber Attacks in Honey Pots. Proceedings of the AAAI Conference on Artificial Intelligence, 35(17), 15224-15231. https://doi.org/10.1609/aaai.v35i17.17786

8. 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. https://doi.org/10.47839/ijc.19.2.1761

9. Shivaprasad, T., Moulya, A. S., & Guruprasad, N. (2024, February). Enhancing Network Security through a Multi-layered Honeypot Architecture with Integrated Network Monitoring Tools. In 2024 11th International Conference on Computing for Sustainable Global Development (INDIACom) (pp. 473-477). IEEE.

10. Moneva, A., Leukfeldt, E. R., & Romagna, M. (2023). Fieldwork experiences researching cybercriminals. Fieldwork Experiences in Criminology and Security Studies: Methods, Ethics, and Emotions, 511-533.

11. Bartwal, U., Mukhopadhyay, S., Negi, R., & Shukla, S. (2022, June). Security orchestration, automation, and response engine for deployment of behavioural honeypots. In 2022 IEEE Conference on Dependable and Secure Computing (DSC) (pp. 1-8). IEEE.

12. Javadpour, A., Ja'fari, F., Taleb, T., Shojafar, M., & Benzaïd, C. (2024). A comprehensive survey on cyber deception techniques to improve honeypot performance. Computers & Security, 103792.

13. B. Savenko, S. Lysenko, K. Bobrovnikova, O. Savenko, G. Markowsky. Detection DNS Tunneling Botnets // Proceedings of the 2021 IEEE 11th International Conference on Intelligent Data Acquisition and

Advanced Computing Systems: Technology and Applications (IDAACS), IDAACS'2021, Cracow, Poland, September 22-25, 2021.

14. Priya, V. D., & Chakkaravarthy, S. S. (2023). Containerized cloud-based honeypot deception for tracking attackers. Scientific Reports, 13(1), 1437.

15. Lysenko S, Bobrovnikova K, Kharchenko V, Savenko O. IoT multi-vector cyberattack detection based on machine learning algorithms: traffic features analysis, experiments, and efficiency. Algorithms. 2022;15(7):239

16. Dong, J., Wang, Z., Wu, J., Cui, X., & Pei, R. (2024). A novel runoff prediction model based on support vector machine and gate recurrent unit with secondary mode decomposition. Water Resources Management, 38(5), 1655-1674.

17. Sidharth, V., & Kavitha, C. R. (2021, September). Network Intrusion Detection System Using Stacking and Boosting Ensemble Methods. In 2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA) (pp. 357-363). IEEE.

18. A. Nicheporuk, O. Savenko, A. Nicheporuk, Y. Nicheporuk, An android malware detection method based on CNN mixed-data model, CEUR Workshop Proceedings, Vol. 2732, 2020, pp. 198–213

19. Dhruv, P., & Naskar, S. (2020). Image classification using convolutional neural network (CNN) and recurrent neural network (RNN): A review. Machine learning and information processing: proceedings of ICMLIP 2019, 367-381.

20. Priyadharshini, A., Umamaheswari, R., Jayapandian, N., & Priyananci, S. (2021, February). Securing medical images using encryption and LSB steganography. In 2021 international conference on advances in electrical, computing, communication and sustainable technologies (ICAECT) (pp. 1-5). IEEE.

21. Song, X., Yang, C., Han, K., & Ding, S. (2022). Robust JPEG steganography based on DCT and SVD in nonsubsampled shearlet transform domain. Multimedia Tools and Applications, 81(25), 36453-36472.

Denysiuk D.O., Savenko B.O., Kashtalian A.S., Ivanchenko O.V. METHOD FOR DETECTING MALICIOUS SOFTWARE BASED ON SYSTEM BEHAVIORAL PATTERN ANALYSIS

The paper investigates the development and evaluation of a method for detecting malware that uses steganographic images to covertly control a system using decoy technology. The proposed approach is based on the use of methods for detecting steganographic changes and a decoy method for detecting malware. The LSB and DCT methods were chosen because of their ability to hide data in images, which allowed them to be used in steganographic attacks. During the study, a series of experiments were conducted to compare the methods of detecting steganographic changes, which allowed us to evaluate their accuracy, sensitivity and speed of response to different types of images containing hidden data. The experimental results demonstrated that images containing hidden commands created using the DCT method were detected with greater accuracy compared to the LSB method.

Although the method has proven to be effective, one of its limitations is the need to use two separate models for different stages of analysis, which can make it difficult to integrate into systems with limited resources. Using separate models for detecting steganographic changes and analyzing system behavior increases the load on computing resources, which is critical in some resource-constrained environments. Further research will be aimed at optimizing the approach by combining these models into a single system to reduce computational costs and increase efficiency. Another important area of future research is the integration of this method with existing information security systems. This will provide a comprehensive approach to protecting information systems from modern threats that use steganographic techniques to hide their activity. Integration with existing information security solutions will also automate the threat detection process and increase the overall level of system protection against steganographic attacks.

Key words: *malicious software, information security, anomaly monitoring, steganographic image, decoys, software traps.*