КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 1 (33), 2026
ISSN 2663 – 4023
DOI 10.28925/2663-4023.2026.33.1130
УДК 004.77

Гаджиев Матін Магсуд-огли
доктор технічних наук, професор
кафедра інженерії програмного забезпечення
Державний університет інтелектуальних технологій і зв'язку, м. Одеса, Україна,
ORCID: 0000-0001-7280-3863
gadjievmm@ukr.net

Бабіч Юрій Олегович
кандидат технічних наук
кафедра інженерії програмного забезпечення
Державний університет інтелектуальних технологій і зв'язку, м. Одеса, Україна,
ORCID: 0000-0002-7888-7591
y.o_babich@suitt.edu.ua

Перекрестов Ігор Сергійович
кандидат технічних наук
кафедра інженерії програмного забезпечення
Державний університет інтелектуальних технологій і зв'язку, м. Одеса, Україна,
ORCID: 0009-0007-3805-8143
perekrestov.igor@gmail.com

Подпригорщук Ігор Миколайович
Студент
Одеський технічний фаховий коледж
Одеського національного технологічного університету, Одеса, Україна
ORCID: 0009-0005-2962-6407
podprigorshyk.igor262@gmail.com

РОЗРОБКА ПРОГРАМНОГО ЗАСОБУ ДЛЯ ДИСТАНЦІЙНОГО АДМІНІСТРУВАННЯ ВУЗЛІВ ЛОКАЛЬНОЇ МЕРЕЖІ

Анотація. У статті описано технічну реалізацію програмного комплексу для дистанційного адміністрування вузлів локальної мережі. Розробка базується на мові C# та платформі .NET. Взаємодія з вузлами здійснюється через UDP-сокети шляхом передачі датаграм. Архітектура системи містить модулі виконання команд у середовищах CMD та PowerShell через клас Process. Реалізовано функції масового завершення процесів прикладного ПЗ, очищення системних об'єктів, дистанційного перезавантаження та вимкнення живлення вузлів. Технологія Wake-on-LAN реалізована через трансляцію Magic Packet на порти 7 та 9 за списком MAC-адрес. Структура Magic Packet містить префікс із 6 байт 0xFF та 16-кратне повторення MAC-адреси цільового пристрою. Пакет керування містить IP-адресу цільового вузла, ідентифікатор користувача та текстове тіло команди. Ідентифікатор користувача зарезервовано для подальшої реалізації багаторівневої системи авторизації. Захист керуючого трафіку базується на шифруванні AES у режимі CBC із заповненням за стандартом ISO10126. Програмна реалізація використовує статичний ключ та вектор ініціалізації (IV). Графічний інтерфейс адміністратора побудовано на базі WinForms. Вибір цільових вузлів здійснюється через компонент CheckedListBox. Результати відправки пакетів та статус виконання операцій візуалізуються у RichTextBox. Програма підтримує роботу з широкомовними адресами для одночасного керування групою пристроїв у межах підмережі. Система призначена для автоматизації технічного обслуговування мережевої інфраструктури та управління робочими станціями. Описаний інструментарій виконує пряму взаємодію з операційною системою без використання сторонніх агентів. Результати тестування підтверджують працездатність обраної моделі взаємодії та стабільність доставки датаграм у локальному сегменті мережі. Програмний комплекс відповідає вимогам технічного завдання щодо швидкодії та функціонального наповнення.

Ключові слова: дистанційне адміністрування, локальна мережа, UDP-сокети, Wake-on-LAN, CMD, PowerShell, AES, C#.

ВСТУП

Імплементацію бізнес-процесів сучасної компанії неможливо уявити без автоматизації, зокрема впровадження інструментів штучного інтелекту на робочому місці. Це призводить до стрімкого збільшення парку комп'ютерів у компанії, а це, у свою чергу, актуалізує задачу дослідження і вдосконалення методів адміністрування комп'ютерних систем і мереж підприємства [1, 2]. Адміністрування парку комп'ютерної техніки в локальних мережах передбачає виконання масових операцій керування живленням та системними процесами. Використання протоколів із встановленням з'єднання створює значні накладні витрати на мережеву інфраструктуру при одночасному зверненні до великої кількості вузлів. Ситуацію ускладнює те, що такі витрати збільшуються із зростанням числа вузлів. Застосування безз'єднувального протоколу UDP забезпечує швидку трансляцію керуючих сигналів без попереднього узгодження зв'язку [3]. Програмна реалізація на базі мережевих сокетів забезпечує пряму взаємодію з компонентами операційної системи [4].

Отже, вдосконалення методу адміністрування парку комп'ютерної техніки в локальних мережах дозволить зменшити навантаження на мережну інфраструктуру компанії, підвищити безпеку процесу адміністрування, скоротити час необхідний для розповсюдження керуючих сигналів в межах мережної інфраструктури компанії, що, в кінцевому підсумку, зменшить витрати на адміністрування мережної інфраструктури. Це сприятиме підвищенню ефективності бізнес-процесів та прибутковості компанії.

Вказана проблематика активно досліджується науковою спільнотою. Так у роботі [5] представлено систему керування апаратними ресурсами, яка використовує стек протоколів TCP/IP та web-технології. Дана публікація актуалізує тематику керування інфраструктурою інформаційної мережі і пропонує рішення даної задачі. Однак, запропоноване рішення спирається на транспортний протокол TCP, а загальне безпека процесу адміністрування може бути підвищена шляхом впровадження шифрування [6], що не реалізовано.

В роботі [7] представлено рішення для аналізу трафіка та адміністрування мережної інфраструктури. Наведене рішення більше пристосовано для аналізу трафіку, а для вирішення певних завдань адміністрування використовується протокол транспортного рівня Transmission Control Protocol (TCP). Аналізуючи дану роботу, можна стверджувати, що діапазон вирішуваних задач по адмініструванню мережної інфраструктури може бути розширений, потенційно можна підвищити безпеку шляхом шифрування, а швидкість роботи може бути збільшена шляхом застосування транспортного протоколу User Datagram Protocol (UDP) [3].

Остання миля мережі, яка заадмініструється, може бути реалізована із застосуванням радіо-інтерфейсу, а це може призводити до нових викликів. Наприклад, у роботі [8] вказується на те, що застосування транспортного протоколу TCP призводить до TCP performance collapse problem (TPCP) у міліметровому діапазоні. Очевидно, що ця обставина має бути врахована при проектуванні системи адміністрування мережі, але вона не враховується у роботах [5, 7].

Відомі рішення для керування мережною інфраструктурою і у випадку радіо-інтерфейсу. Зокрема, рішення представлене у роботі [9]. Запропоноване рішення залишає простір для вдосконалення у частині безпеки та швидкості роботи.

Аналіз джерел за досліджуваною тематикою вказує на науковий інтерес до неї і наявність реалізованих інженерних рішень, наприклад [5, 7, 9]. Однак, наявні рішення залишають простір для вдосконалення у частині підвищення швидкодії, підвищення безпеки шляхом впровадження шифрування, наприклад представленого у [10], та розширення номенклатури вирішуваних адміністративних завдань.

Мета статті. Метою даної роботи є синтез системи адміністрування мережної інфраструктури компанії. Розроблювана система має підтримувати централізоване виконання команд у середовищі Command Prompt (CMD), включати інтеграцію механізму Wake-on-LAN для дистанційного ввімкнення вузлів та захист трафіку алгоритмом Advanced Encryption Standard (AES), що дозволить впровадити шифрування та значно підвищити безпеку комунікації в процесі адміністрування.

Для досягнення мети вона може бути декомпозована на наступні задачі:

- реалізувати підтримку інтерфейсу CMD;
- інтегрувати до системи механізм Wake-on-LAN;
- забезпечити шифрування алгоритмом AES;
- система має ґрунтуватись на транспортному протоколі UDP;
- провести оцінку ефективності розробленої системи.

Проблема полягає у забезпеченні стабільної доставки датаграм та їх коректній обробці клієнтськими модулями в умовах широкомовного сегмента мережі. Необхідна розробка структури пакетів із цільовими параметрами вузла та зашифрованим тілом команди.

Досягнення мети передбачає реалізацію механізмів та технічний опис передачі датаграм за протоколом UDP (що підвищить швидкість роботи порівняно із протоколом TCP), інтеграцію технології Wake-on-LAN для управління живленням та впровадження криптографічного захисту керуючих команд на основі алгоритму AES у статичному режимі.

МЕТОДИКА ДОСЛІДЖЕННЯ

Методика дослідження базується на побудові клієнт-серверної моделі взаємодії вузлів у локальній мережі з використанням безз'єднувального протоколу передачі даних, описаного у [1, 11]. Програмна реалізація виконана в середовищі Visual Studio мовою C# на базі платформи .NET [12]. Основний акцент зроблено на розробці алгоритмів формування низькорівневих пакетів керування та їх криптографічного захисту.

Процес реалізації дистанційного ввімкнення вузлів (Wake-on-LAN) передбачає трансформацію рядкового представлення MAC-адреси у байтовий масив. Алгоритм містить етап валідації вхідних даних та формування структури Magic Packet шляхом конкатенації шести байтів 0xFF та шістнадцятикратного повторення отриманої фізичної адреси, згідно структури наданої у [13]. Надсилання сформованого масиву здійснюється через об'єкт класу Socket із використанням широкомовної адреси підмережі (наприклад, 192.255.255.255) та портів 7 або 9 [4].

Методика віддаленого виконання команд CMD базується на ітераційному аналізі стану елементів інтерфейсу CheckedListBox. Програма здійснює перевірку вибору конкретних вузлів через метод GetTruePC. Для кожного активного вузла формується пакет даних, що містить цільову IP-адресу, ідентифікатор адміністратора та текст команди. Передача інформації реалізована через метод SendTo класу Socket, що забезпечує доставку датаграм на порт 4341 цільових машин.

Криптографічний захист керуючого трафіку реалізовано через інтеграцію модуля AES у режимі CBC [10] . Методика шифрування передбачає використання статичного ключа та вектора ініціалізації [6]. Перед відправкою у мережу текстові дані проходять етап трансформації у байтовий масив із застосуванням кодування UTF-8 та подальшим накладанням шифру. На боці отримувача передбачено зворотну дешифрацію та передачу результату системному інтерпретатору командного рядка.

Експериментальна частина дослідження проведена в умовах локальної мережі комп'ютерного класу, що налічує 15 робочих станцій. Методика тестування включала перевірку стабільності пробудження вузлів із вимкненого стану, а також оцінку надійності виконання групових команд taskkill та shutdown. Оцінка результатів здійснювалася на основі аналізу логів відправки пакетів, зафіксованих у компоненті RichTextBox.

СИНТЕЗ СИСТЕМИ АДМІНІСТРУВАННЯ ВУЗЛІВ ЛОКАЛЬНОЇ МЕРЕЖІ

В основі функціонування системи лежить використання транспортного протоколу UDP [3]. Безз'єднувальний характер протоколу мінімізує накладні витрати на встановлення сесії, забезпечуючи високу швидкість трансляції команд у локальних мережах, що відомо з [11]. Взаємодія здійснюється через мережеві сокети, де адміністративний вузол виступає ініціатором широкомовних (broadcast) або адресних запитів [4].

Загальна структурна схема розробленої системи адміністрування представлена на рис. 1.

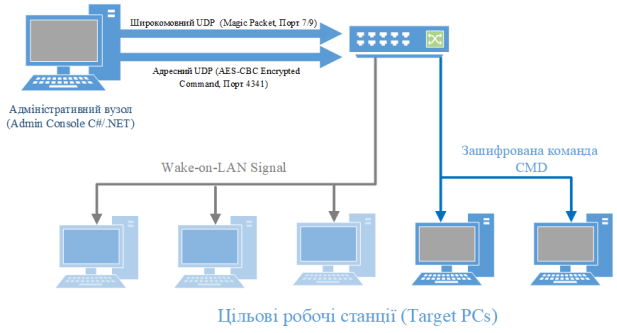

Рис. 1. Структурна схема системи адміністрування вузлів локальної мережі

Згідно з рис. 1, взаємодія вузлів базується на UDP-broadcast. Для дистанційного ввімкнення (Wake-on-LAN) використовуються порти 7 та 9. Керування активними станціями реалізовано через порт 4341.

Селекція цільового вузла здійснюється програмним агентом на прикладному рівні: після дешифрування проводиться верифікація вкладеної IP-адреси (UTF-8) шляхом порівняння з адресою локального хоста.

Криптографічний захист керуючих інструкцій реалізовано за допомогою стандарту AES [10]. Для забезпечення унікальності шифротексту при повторенні однакових команд використано режим зчеплення блоків шифру (Cipher Block Chaining, CBC). У даному режимі кожен блок відкритого тексту перед шифруванням піддається операції XOR з результатом шифрування попереднього блоку [6].

Математично процес шифрування в режимі CBC описується формулою:

$$C_i = E_k(P_i \oplus C_{i-1}) C_0 = IV \quad (1)$$

де C_i – блок шифротексту, E_k – функція шифрування з ключем K, P_i – блок відкритого тексту, IV – вектор ініціалізації [6].

Для доведення довжини вхідних даних до розміру, кратного 16 байтам, застосовується метод заповнення ISO10126. Даний стандарт передбачає заповнення останнього блоку випадковими байтами, де останній байт вказує на кількість доданих символів. Це підвищує стійкість шифротексту до статистичного аналізу. Використання статичного ключа та вектора ініціалізації в межах локального сегмента мережі забезпечує базову конфіденційність адміністративних команд [6].

Експериментальна перевірка розробленого програмного комплексу проводилася в локальній мережі комп'ютерного класу, що налічує 15 робочих станцій. Аналіз отриманих результатів дозволяє виділити кілька ключових аспектів функціонування системи.

Ефективність модуля Wake-on-LAN базується на коректному формуванні Magic Packet. Математично структура сформованого пакета P описується виразом:

$$P = \{FF_6, (MAC)_{16}\} \quad (2)$$

де FF_6 – блок із шести байтів значенням 0xFF, $(MAC)_{16}$ – шістнадцятикратне повторення фізичної адреси цільового мережевого адаптера [13].

Нижче наведено програмну реалізацію методу формування та відправки пакета мовою C# з використанням бібліотеки сокетів [4]:

```
public void WakeOnLan(string mac)
{
    Socket socket = new Socket(SocketType.Dgram, ProtocolType.Udp);
    string[] macByteStr = mac.Split(':', '-');
    byte[] macByte = new byte[6];
    for (int i = 0; i < 6; ++i)
        macByte[i] = Convert.ToByte(macByteStr[i], 16);

    byte[] sixByteFF = { 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF };
    List<byte> bytes = new List<byte>();
    bytes.AddRange(sixByteFF);
    for (int i = 0; i <= 15; ++i)
        bytes.AddRange(macByte);

    byte[] data = bytes.ToArray();
    socket.SendTo(data, data.Length, SocketFlags.DontRoute, endPointPort9);
}
```

Тестування показало 100% успішність пробудження вузлів при їх знаходженні в одному сегменті мережі. Час розсилки пакетів на всі вузли не перевищує 10 мс.

Управління процесами здійснюється через передачу текстових інструкцій інтерпретатору CMD. Програмна логіка методу «SendSocketCMD» забезпечує формування датаграм, що містять метадані вузла та тіло команди [4]:

```
public void SendSocketCMD(string massege)
{   Socket socket = new Socket(SocketType.Dgram, ProtocolType.Udp);
    bool[] isTruePc = GetTruePC(); // Повертає список цільових комп'ютерів
    for (int i = 0; i < macStr.Length; i++)
    {
        if (!isTruePc[i]) continue;
        string suf = (i < 9) ? "0" + (i + 1) : (i + 1).ToString();
        string currentIP = "192.168.0.1" + suf + "|" + "Admin" + "|" + GetInvisible();
        byte[] buffer = Encoding.UTF8.GetBytes(currentIP + massege);
        int len = socket.SendTo(buffer, buffer.Length, SocketFlags.None, endPointCMD);
        richTextBox1.Text += ($"Відправлено({len}) комп'ютеру {i + 1}\n");}
}
```

ОБГОВОРЕННЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕНЬ

Показники швидкодії та надійності виконання типових операцій наведено у Таблиці 1. Середня затримка виконання команди (від моменту відправки до запуску процесу на цільовій ОС) становить 45-120 мс.

Таблиця 1

Показники ефективності виконання команд у локальній мережі

Тип операції	Метод передачі	Сер. затримка (мс)	Успішність (%)
Wake-on-LAN	UDP (Port 9)	1,5	100
Taskkill (закриття ПЗ)	UDP (Port 4341)	65	98
Shutdown / Restart	UDP (Port 4341)	80	100

Використання алгоритму AES у режимі CBC забезпечує базову конфіденційність трафіку [10]. Незважаючи на статичний характер ключів, структура команди повністю маскується, що унеможливлює її перехоплення стандартними засобами аналізу мережі [6]. Візуальний контроль за станом мережі та результатами логування здійснюється через графічний інтерфейс (рис. 2).

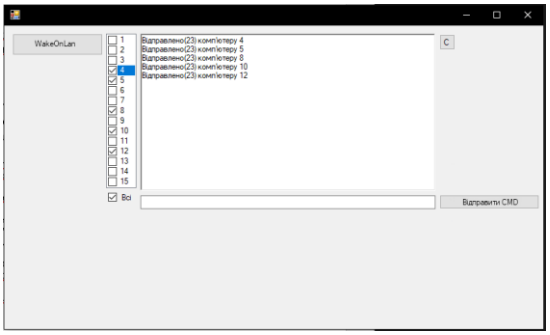

Рис. 2. Графічний інтерфейс програми з результатами логування відправлених команд

Логування у RichTextBox дозволяє адміністратору ідентифікувати вузли, які не отримали пакет через технічні несправності або відсутність живлення. Обгрунтуванням вибору UDP є можливість одночасного керування всіма вузлами без встановлення індивідуальних сесій, що критично для оперативного адміністрування комп'ютерного класу [3].

На рис. 3 та 4 наведена графічна ілюстрація отриманих результатів дослідження розроблюваної системи мережного адміністрування, яка ґрунтується на протоколі транспортного рівня UDP.

Для дослідження були обрані команди із різною алгоритмічною складністю для більш різносторонньої оцінки розробленої системи адміністрування.

З рис. 3 видно, що впровадження транспортного протоколу UDP, як базису системи адміністрування дало часовий виграш у порівнянні з системами, які реалізовані на протоколі TCP, а саме рішення представлені у [5, 7].

Однак, порівняння було б однобічним без оцінки відсотка успішних команд, адже протокол UDP працює без підтвердження доставки датаграм [11].

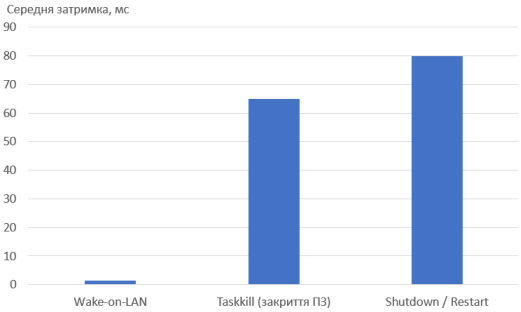

Рис. 3. Середня часова затримка для різних команд системи адміністрування

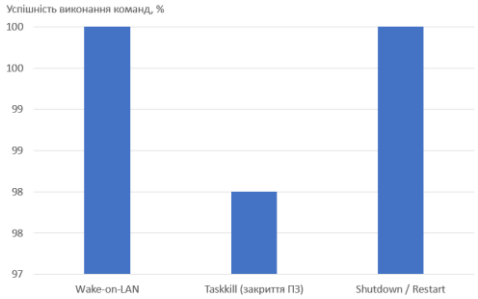

Рис. 4. Відсоток успішно виконаних команд системи адміністрування

З рис. 4 видно, що найгірший отриманий результат складає 98% для команд закриття додатків, які працюють на віддалених робочих станціях.

Таким чином, аналізуючи спільно рисунки 3 та 4, можна стверджувати, що синтезована система адміністрування комп'ютерної інфраструктури підприємства забезпечує більшу швидкість виконання команд порівняно із конкурентами, які ґрунтуються на транспортному протоколі TCP, наприклад рішення [5, 7], при прийнятному рівні успішно виконаних команд.

Разом з тим, рис. 4 вказує на виявлене обмеження розроблюваної системи адміністрування – наявність невиконаних команд через безз'єднувальний принцип роботи протоколу UDP. Але для об'єкту дослідження виявлений рівень успішності виконання команд є прийнятним. Слід відмітити, що усі команди Wake-on-LAN та Shutdown/Restart були виконані без втрат.

Отже, у подальшому дослідженні цікавість представляє питання залежності втрат від обсягу робочих станцій у мережі.

Інша перспектива подальших досліджень полягає у розширенні функціональних можливостей системи. Пріоритетним напрямком є інтеграція оболонки PowerShell для виконання складних сценаріїв автоматизації та перехід від статичних ключів шифрування до протоколів динамічного обміну ключами (наприклад, на основі протоколу Діффі-Геллмана [6]). Також планується впровадження багаторівневої системи авторизації на основі зарезервованих у структурі пакета ідентифікаторів користувачів для розмежування прав доступу адміністраторів.

ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

У ході дослідження розроблено та апробовано програмний комплекс для дистанційного керування вузлами локальної мережі. Реалізована архітектура на базі протоколу UDP підтвердила свою ефективність для виконання групових адміністративних завдань у межах комп'ютерної мережі підприємства. Використання безз'єднувального протоколу забезпечило високу швидкість трансляції керуючих сигналів із мінімальними накладними витратами на мережеву інфраструктуру.

Результати тестування підтверджують повне виконання технічного завдання. Модуль Wake-on-LAN забезпечує стабільне дистанційне ввімкнення пристроїв через формування та відправку Magic Packet. Механізм взаємодії з інтерфейсом CMD дозволяє здійснювати повний контроль над системними процесами, включаючи примусове завершення програм та управління станом живлення робочих станцій. Впровадження криптографічного модуля AES у режимі CBC забезпечує базову конфіденційність трафіку, захищаючи системні інструкції від прямого перехоплення та аналізу в межах мережевого сегмента.

У процесі дослідження виявлено невиконання 2% команд типу «Taskkill», яке пояснюється безз'єднувальною роботою транспортного протоколу UDP. З огляду на отримане пришвидшення виконання адміністративних команд, такий рівень втрат є прийнятним. Характер залежності цього показника від обсягу робочих станцій у мережі є предметом подальшого дослідження.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Marcham, A. (2021). Introduction to network technology. In *Understanding infrastructure edge computing: Concepts, technologies, and considerations* (pp. 21-52). Wiley. https://doi.org/10.1002/9781119763260.ch3
2. Limoncelli, T., Hogan, C., & Chalup, S. (2016). *The practice of system and network administration: DevOps and other best practices for enterprise IT* (3rd ed., Vol. 1). Addison-Wesley Professional
3. Postel, J. (1980). *User datagram protocol (RFC 768)*. Internet Engineering Task Force. https://datatracker.ietf.org/doc/html/rfc768

4. Microsoft Learn. (2024). *Socket class (System.Net.Sockets)*. https://learn.microsoft.com/en-us/dotnet/api/system.net.sockets.socket
5. Damian, C., Lunca, E., & Ilinca, M. (2014). Remote administration of hardware resources using TCP/IP protocol and web technologies. In *2014 International Conference and Exposition on Electrical and Power Engineering (EPE)* (pp. 123-126). IEEE. https://doi.org/10.1109/ICEPE.2014.6969881
6. Stallings, W. (2023). *Cryptography and network security: Principles and practice* (8th ed.). Pearson.
7. Wan, M. H., & Horng, M. F. (2008). An intelligent monitoring system for local-area network traffic. In *2008 Eighth International Conference on Intelligent Systems Design and Applications* (pp. 657-660). IEEE. https://doi.org/10.1109/ISDA.2008.366
8. Kim, M., Ko, S. W., Kim, H., Kim, S., & Kim, S. L. (2018). Exploiting caching for millimeter-wave TCP networks: Gain analysis and practical design. *IEEE Access*, 6, 69769-69781. https://doi.org/10.1109/ACCESS.2018.2880774
9. Rohila, D., & Jain, N. (2014). RFID network administration and control. In *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)* (pp. 2009-2014). IEEE. https://doi.org/10.1109/ICACCI.2014.6968637
10. Evans, D. L., Bond, P. J., & Brown, K. H. (2023). *Advanced encryption standard (AES) (FIPS 197-upd1)*. National Institute of Standards and Technology. https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197-upd1.pdf
11. Tanenbaum, A. S., & Wetherall, D. J. (2011). *Computer networks* (5th ed.). Pearson Education.
12. Burns, S. (2019). *Hands-on network programming with C# and .NET Core*. Packt Publishing.
13. Advanced Micro Devices. (1995). *Magic packet technology* (White paper No. 20213). https://www.amd.com/content/dam/amd/en/documents/archived-tech-docs/white-papers/20213.pdf

КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 1 (33), 2026
ISSN 2663 – 4023

Matin Hadzhyiev
Doctor of Technical Sciences, Professor
Department of Software Engineering
State University of Intelligent Technologies and Telecommunications, Odesa, Ukraine
ORCID: 0000-0001-7280-3863
gadjievmm@ukr.net

Yurii Babich
PhD in Telecommunication Systems and Networks
Department of Software Engineering
State University of Intelligent Technologies and Telecommunications, Odesa, Ukraine
ORCID: 0000-0002-7888-7591
y.o_babich@suitt.edu.ua

Ihor Perekrestov
PhD in Radio Engineering and Television Systems,
Department of Software Engineering
State University of Intelligent Technologies and Telecommunications, Odesa, Ukraine
ORCID: 0009-0007-3805-8143
perekrestov.igor@gmail.com

Ihor Podpryhorshchuk
Student
Odesa Technical Professional College
of Odesa National University of Technology, Odesa, Ukraine
ORCID: 0009-0005-2962-6407
podprigorshyk.igor262@gmail.com

DEVELOPMENT OF A SOFTWARE TOOL FOR REMOTE ADMINISTRATION OF LOCAL NETWORK NODES

Abstract. The article describes the technical implementation of a software complex for remote administration of local network nodes. The development is based on the C# language and the .NET framework. Interaction with nodes is carried out through UDP sockets by transmitting datagrams. The system architecture contains modules for executing commands in CMD and PowerShell environments via the Process class. Functions for bulk termination of application processes, clearing system objects, remote reboot, and powering off nodes are implemented. Wake-on-LAN technology is implemented via Magic Packet translation to ports 7 and 9 according to a list of MAC addresses. The Magic Packet structure contains a prefix of 6 bytes of 0xFF and a 16-fold repetition of the target device's MAC address. The control packet contains the target node's IP address, a user identifier, and a text command body. The user identifier is reserved for further implementation of a multi-level authorization system. Protection of control traffic is based on AES encryption in CBC mode with padding according to the ISO10126 Standard. The software implementation uses a static key and an initialization vector (IV). The administrator's graphical interface is built on the basis of WinForms. The selection of target nodes is carried out through the CheckedListBox component. The results of sending packets and the status of operations are visualized in the RichTextBox. The program supports working with broadcast addresses for simultaneous management of a group of devices within a subnet. The system is designed for automating technical maintenance of network infrastructure and managing workstations. The described toolkit performs direct interaction with the operating system without the use of third-party agents. The testing results confirm the efficiency of the selected interaction model and the stability of datagram delivery in the local network segment. The software complex meets the requirements of the technical task regarding performance and functional content.

Keywords: remote administration, local network, UDP sockets, Wake-on-LAN, CMD, PowerShell, AES, C#.

REFERENCES (TRANSLATED AND TRANSLITERATED)

1. Marcham, A. (2021). Introduction to network technology. In *Understanding infrastructure edge computing: Concepts, technologies, and considerations* (pp. 21-52). Wiley. https://doi.org/10.1002/9781119763260.ch3
2. Limoncelli, T., Hogan, C., & Chalup, S. (2016). *The practice of system and network administration: DevOps and other best practices for enterprise IT* (3rd ed., Vol. 1). Addison-Wesley Professional.
3. Postel, J. (1980). *User datagram protocol (RFC 768)*. Internet Engineering Task Force. https://datatracker.ietf.org/doc/html/rfc768
4. Microsoft Learn. (2024). *Socket class (System.Net.Sockets)*. https://learn.microsoft.com/en-us/dotnet/api/system.net.sockets.socket
5. Damian, C., Lunca, E., & Ilinca, M. (2014). Remote administration of hardware resources using TCP/IP protocol and web technologies. In *2014 International Conference and Exposition on Electrical and Power Engineering (EPE)* (pp. 123-126). IEEE. https://doi.org/10.1109/ICEPE.2014.6969881
6. Stallings, W. (2023). *Cryptography and network security: Principles and practice* (8th ed.). Pearson.
7. Wan, M. H., & Horng, M. F. (2008). An intelligent monitoring system for local-area network traffic. In *2008 Eighth International Conference on Intelligent Systems Design and Applications* (pp. 657-660). IEEE. https://doi.org/10.1109/ISDA.2008.366
8. Kim, M., Ko, S. W., Kim, H., Kim, S., & Kim, S. L. (2018). Exploiting caching for millimeter-wave TCP networks: Gain analysis and practical design. *IEEE Access*, 6, 69769-69781. https://doi.org/10.1109/ACCESS.2018.2880774
9. Rohila, D., & Jain, N. (2014). RFID network administration and control. In *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)* (pp. 2009-2014). IEEE. https://doi.org/10.1109/ICACCI.2014.6968637
10. Evans, D. L., Bond, P. J., & Brown, K. H. (2023). *Advanced encryption standard (AES) (FIPS 197-upd1)*. National Institute of Standards and Technology. https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197-upd1.pdf
11. Tanenbaum, A. S., & Wetherall, D. J. (2011). *Computer networks* (5th ed.). Pearson Education.
12. Burns, S. (2019). *Hands-on network programming with C# and .NET Core*. Packt Publishing.
13. Advanced Micro Devices. (1995). *Magic packet technology* (White paper No. 20213). https://www.amd.com/content/dam/amd/en/documents/archived-tech-docs/white-papers/20213.pdf

Отримано редакцією журналу / Received: 16.02.26
Прорецензовано / Revised: 28.02.26
Схвалено до друку / Accepted: 25.06.26