УДК 004.4
DOI https://doi.org/10.32838/2663-5941/2022.3/07

Кандиба І.О.
Чорноморський національний університет імені Петра Могили

Горбань Г.В.
Чорноморський національний університет імені Петра Могили

Фісун М.Т.
Чорноморський національний університет імені Петра Могили

Ткаченко М.П.
Чорноморський національний університет імені Петра Могили

ДОСЛІДЖЕННЯ АПАРАТНО-ТЕХНІЧНОГО СТАНУ ЛОКАЛЬНОЇ МЕРЕЖІ ЗАКЛАДУ ВИЩОЇ ОСВІТИ ЗАСОБАМИ МОВИ PYTHON

У статті представлено дослідження інструментарію аналізу апаратного забезпечення, що доступний для інтеграції з мовою загального призначення Python. Проведено аналіз сучасних досліджень в галузі моніторингу апаратно-технічного стану локальної мережі. Визначені бібліотеки, що розв'язують задачі моніторингу навантаження апаратних складових локальної мережі. Виявлено основні особливості роботи локальної мережі закладу вищої освіти. Наведено опис засобів Windows Management Instrumentation у якості інструменту визначення характеристик центрального процесора, оперативної пам'яті, накопичувачів даних тощо. Описано особливості впливу шкідливого програмного забезпечення на апаратне забезпечення. Представлено можливість поточного визначення поточного навантаження оперативної пам'яті psutil. Розглянуто інструменти моніторингу навантаження центрального процесора. Досліджені засоби моніторингу роботи накопичувачів даних. Визначено доцільну модель для зберігання даних моніторингу. Розроблено структуру реляційної бази, що містить характеристики та дані навантаження апаратного забезпечення. Визначено найбільш доцільну СКБД, що зберігатиме зазначені дані. Наведено переваги застосування SQLite для реалізації програмного забезпечення моніторингу апаратного забезпечення. Представлено архітектуру розподіленого програмного забезпечення для моніторингу апаратного забезпечення. Запропоновано застосування засобів візуалізація застосованих даних шляхом підключення бібліотеки Matplotlib. Запропоновано застосування бібліотеки Statistics для використання методів визначення середнього показника навантаження. Реалізовано методи статистичної обробки даних споживання апаратних ресурсів. Описано можливість використання методу розрахунку середньоквадратичного відхилення для виявлення зміни навантаження та пошуку шкідливого програмного забезпечення. Визначено подальші шляхи розвитку запропонованої системи апаратно-технічного стану локальної мережі закладу вищої освіти.

Ключові слова: *Python, WMI, psutil, СКБД, Matplotlib, моніторинг апаратного забезпечення.*

Постановка задачі. Локальні мережі закладів вищої освіти (ЗВО) складаються з обладнання різного типу. До складу цих мереж входять комп'ютери з різними операційними системи та з різними апаратними архітекторами. Відмінності версій операційних систем (ОС) та архітектури комп'ютерів потребує тонкого налаштування мережевих серверів та служб [1, с. 117]: Active directory, DHCP, серверів оновлень тощо. Розв'язувати цю задачу не можливо за допомогою вбудованих засобів ОС.

Окрім того, моніторинг апаратно-технічного є необхідною складовою для забезпечення безпеки інформаційних ресурсів. Не зважаючи на різноманітність шкідливого програмного забезпечення (ПЗ), всі представники цього класу ПЗ мають спільну рису – інтенсивне використання апаратних ресурсів [2, с. 5]. В залежності від класу шкідливого ПЗ відбувається використання різних апаратних ресурсів: жорсткого диска, центрального процесору (ЦП), оперативної пам'яті тощо.

Необхідним є створення засобів централізованого моніторингу апаратно-технічного стану локальної мережі в ЗВО, що дасть змогу системним адміністраторам виявлення шкідливого програмного забезпечення та дозволить спростити процес налаштування мережевих ресурсів та сервісів.

Аналіз останніх досліджень і публікацій. Однією з головних цілей моніторингу апаратно-технічного стану локальної мережі є убезпечення інформаційних ресурсів від несанкціонованого доступу. У роботі [3, с. 14] досліджено особливості використання спеціалізованого апаратного забезпечення для перехоплення даних та порушення інформаційної безпеки локальної мережі. Автор описує можливість застосування пристроїв на основі мікроконтролерів для перехоплення натиснутих клавіш та пересилання перехоплених даних локальною мережею. В роботі недостатньо повне дослідження можливості перехоплення клавіш, а точніше використання програмних рішень для реалізації перехоплення.

Робота [4, с. 4] присвячена дослідженню ознак шкідливого програмного забезпечення. Висвітлено взаємозв'язок навантаження на апаратну складову комп'ютера та наявність шкідливого програмного забезпечення. Розглянуто ряд програмного забезпечення для моніторингу апаратних складових. Однак, у роботі не запропоновано централізованих методів моніторингу локальної мережі.

Визначення навантаження на апаратне забезпечення досліджено у [5, с. 66]. У цій роботі досліджено моніторинг апаратних ресурсів з метою поліпшення продуктивності комп'ютерів в локальній мережі. Запропоноване застосування мови програмування загального призначення Python для відстежування запущених процесів та спожитих ними ресурсів. Наведене дослідження містить аналіз реалізації моніторингу апаратних ресурсів, але недостатньою увагу приділено можливості визначення шкідливого програмного забезпечення.

Моніторинг мережевої активності досліджено у роботі [6, с. 105]. Це дослідження присвячено аналізу можливості використання об'єктних баз даних для зберігання даних мережевого трафіку. Запропоновано метод створення OLAP-куба мережевого трафіку на основі багатомірної моделі. В статті недостатньою увагу приділено можливості моніторингу та зберігання даних використання апаратних ресурсів комп'ютерної мережі.

Постановка завдання. Метою роботи є вдосконалення процесу дослідження апаратно-технічного стану локальної мережі закладу вищої освіти шляхом розробки спеціалізованого програмного забезпечення.

Для досягнення мети розв'язуються такі наукові завдання:

− дослідження засобів моніторингу апаратного забезпечення основі використання мови загального призначення Python;

− формування моделі даних для зберігання даних моніторингу апаратного забезпечення;

− реалізація засобів статистично обробки результатів зазначеного моніторингу;

− розробка засобів візуалізації отриманих даних моніторингу.

Виклад основного матеріалу дослідження. Локальна комп'ютерна мережа ЗВО має містити апаратне забезпечення різного типу для забезпечення навчально процесу різних спеціальностей інформаційних технологій. Налаштування цієї мережі можливе лише за наявності актуальних даних моніторингу комп'ютерів, що входять до її складу.

Мережеві сервіси для забезпечення коректної роботи студентів ЗВО, наприклад Active directory, вимагають створення окремих налаштувань для різних архітектур (AMD64, ARM тощо), версій операційних систем та ін. У ЗВО, як правило, наявна велика кількість персональних комп'ютерів (ПК), наприклад мережа ЧНУ імені Петра Могили містить більше ніж 500 робочих місць, що розміщенні у різних наукових лабораторіях та відділах. Адміністратори відповідальні за апаратне та програне забезпечення цих комп'ютерів, фізично не мають змоги аналізувати роботу цього обладнання та вести моніторинг його поточного стану.

Моніторинг апаратних ресурсів ПК є комплексною задачею, що потребує аналізу великої кількості складових. Розв'язати цю задачу можливо шляхом розробки програмного забезпечення мовою загального призначення.

Специфіка комп'ютерної мережі ЗВО вимагає створення універсального програмного забезпечення з високим ступенем модифікованості і переносності. Мова програмування загального призначення Python підтримує велику кількість модулів для наукових досліджень (Statistics, NumPy тощо) і підтримує можливість роботи з більшістю сучасних операційних систем [7, с. 3]. Наведений факт робить цю мову програмування доцільною для використання у якості засобу моніторингу апаратно-технічного стану локальної мережі ЗВО.

Моніторинг апаратного забезпечення реалізується кількома бібліотеками мови Python. За умови використання операційної системи сімейства Windows функціональною є бібліотека wmi, що є реалізацією Windows Management Instrumentation (WMI) [8, с. 2]. Цей інструмент є засобом керування ОС Windows та містить набір аналізу апаратного забезпечення.

WMI встановлюється за допомогою вбудованого менеджера керування пакунками python.

Моніторинг апаратного забезпечення відбувається в режимі реального часу, що дозволяє визначення характеристик та стану апаратних компонентів.

Оперативна пам'ять впливає на продуктивність комп'ютеру і є однією з головних апаратних складових. Визначити характеристики цього компоненту можливо за рахунок використання функції Win32_PhysicalMemory(). Цей метод повертає набір характеристик: модель, серійний номер, швидкість нити тощо. Використання цього методу дозволяє визначити поточну інформацію про фізичний оперативної пам'яті.

Виявлення шкідливого програмного забезпечення за рахунок моніторингу ресурсів потребує відстежування поточного навантаження на оперативну пам'ять. Реалізувати цей процес дозволяє бібліотека psutil, що також підключається до мови Python. Аналіз поточного навантаження оперативної пам'яті відбувається методом virtual_memory(), що відображає загальний об'єм, доступна та кількість оперативної пам'яті, яка використовується.

Характеристики ЦП можна визначити також за допомогою WMI, а саме методу Win32_Processor. Цей метод повертає назву моделі, тактову частоту, розмір кешу, набори команд ЦП та ін.. Наведена інформація є важливою для організації навчального ЦП, а саме для визначення можливостей проведення занять певних дисципліну у певних навчальних лабораторіях.

Значне навантаження на ЦП є одним з визначників показників наявності шкідливого програмного забезпечення. Моніторинг поточного застосування ЦП можливий також засобами psutil, що включає метод cpu_percent(). Цей метод повертає поточне навантаження кожного ядра ЦП.

Дізнатись характеристики накопичувачів даних, графічних процесорів, мережевих адапторів та інших компонентів можливо іншими методами WMI, наприклад Win32_DiskDrive(), Win32_VideoController(), Win32_NetworkAdapterConfiguration(). Відстежити поточний стан навантаження на апаратні частини можливо отримати засобами psutil, наприклад disk_partitions(), psutil.disk_usage().

Шкідливе програмне забезпечення завжди має бути представлене процесом. У більшості випадків процес маскується під інше програмне забезпечення, але його можна виявити за споживанням ресурсів. Аналіз запущених процесів можливо провести засобами psutil шляхом застосування методу Win32_Process(), що відображає запущені процеси. Окрім того, шкідливе програмне забезпечення може маскуватись серед запущених сервісів. Сервіси можливо моніторити також за допомогою psutil методом Win32_Service().

Важливим фактором є розподіленість локальної мережі ЗВО, де кожна лабораторія може бути зоною відповідальності окремого адміністратору, викладача або кафедри. Наприклад в ЧНУ ім.

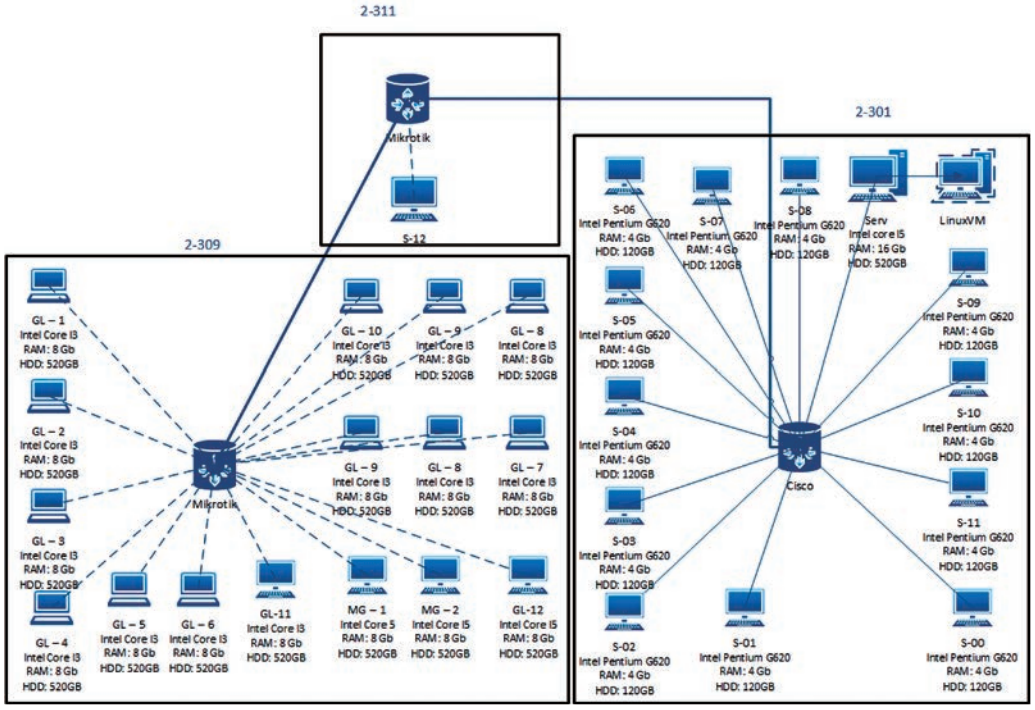

Рис. 1. Схематично відображення частини мережі ЗВО

Петра Могили на кафедрі інженерії програного забезпечення наявні лабораторії системного програмного забезпечення та інженерії програмного забезпечення, що призначені для викладання абсолютно різних дисциплін і відповідно мають різне апаратне забезпечення (рис. 1). Централізована система апаратно-технічного стану дасть змогу визначати обчислювальні ресурси для можливості встановлення найновіших версій ПЗ.

Зібрані дані необхідно зберігати у спеціалізованому сховищі. Таким сховищем є база даних (БД), що базується на реляційній моделі даних. У БД цього типу окремими множинами можливо представити: процеси, характеристики апаратного забезпечення, зібрані дані навантаження тощо. Математично множини можливо представити у вигляді множини: $D_1=\{d_{11}, d_{12}, d_{13},\ldots,d_{1n}\}$, $D_2=\{d_{21}, d_{22}, d_{23},\ldots,d_{2m}\}$, $D_3=\{d_{31}, d_{32}, d_{33},\ldots,d_{3p}\}$,…, $D_k=\{d_{k1}, d_{k2}, d_{k3},\ldots,d_{kS}\}$.

Кожен комп'ютер є підмножиною декартового добутку згаданих вище множин:

$$D_1\times D_2\times D_3\times D_k=\{(d_{11},d_{21}, d_{31},\ldots d_{k1}),\ldots, (d_{1n},d_{2m}, d_{3p},\ldots d_{kS})\} \quad (1)$$

Зберігання зібраних даних у реляційній БД потребує використання спеціалізованого програмного забезпечення системи керування баз даних (СКБД). Прикладом такого програмного забезпечення є SQLite. Це СКБД підтримується багатьма мовами програмування загального призначення, а також не вимагає наявності додаткового програмного забезпечення для обробки запитів [9, с. 241]. Ця СКБД представлена у вигляді окремих бібліотек та не має окремого серверу. Перевагою її використання є можливість перенесення створеної бази даних на локальний комп'ютер шляхом простого копіювання та значна швидкодія.

При створенні БД у СКБД SQLite для зберігання даних апаратно-технічного стану локальної мережі закладу вищої освіти необхідне врахувати характеристики апаратного забезпечення, статистичні дані навантаження, запущені процеси та сервіси (рис. 2).

Структура системи моніторингу даних апаратно-технічного стану локальної мережі бути побудована на основі архітектури клієнт-сервер. Де сервер представлений програмним забезпеченням для роботи з СКБД, а клієнт виконує збір даних поточних характеристик апаратного забезпечення та передачу зібраних даних до сервера.

Мова загального призначення Python підтримує можливість роботу з вебфремворками, наприклад Django. Цей фреймворк дозволяє отримати

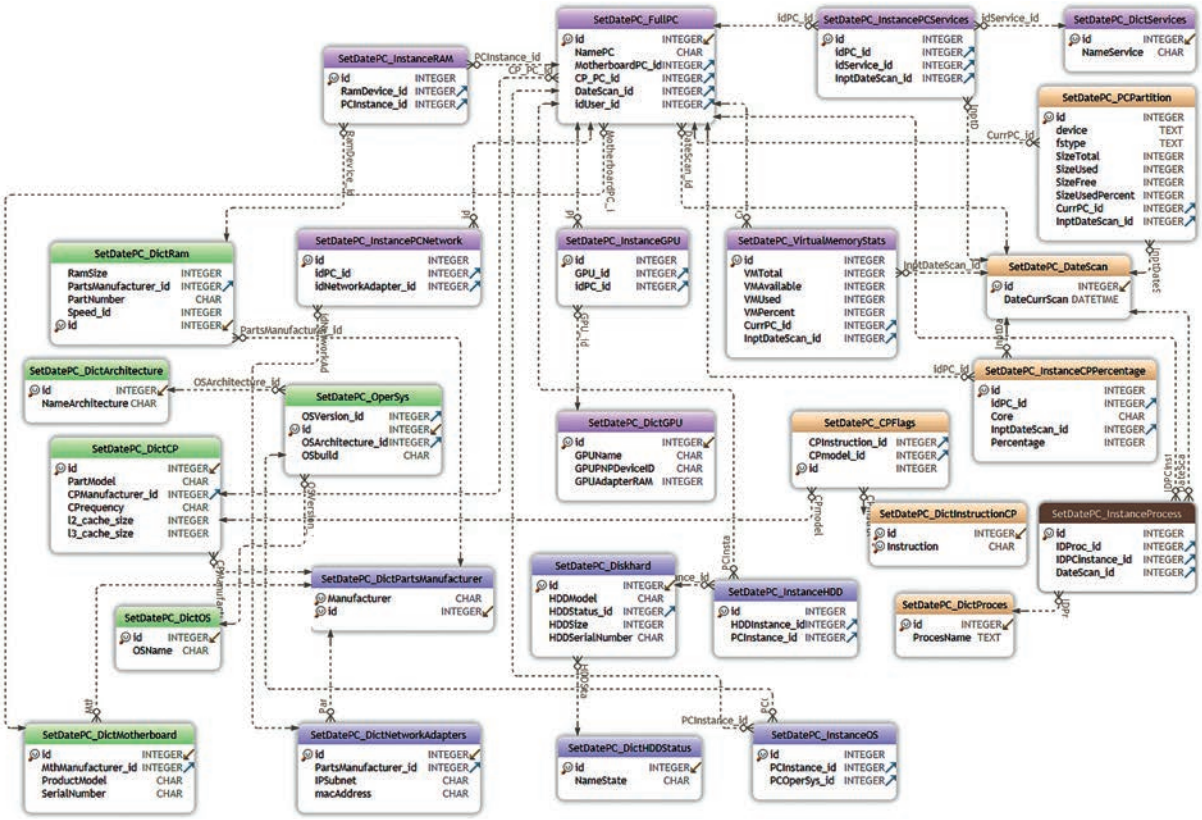

Рис. 2. Даталогічна модель БД для зберігання даних апаратно-технічного стану локальної мережі

дані від застосунку клієнту засобами вебзапиту і в майбутньому надати доступ для перегляду отриманих даних.

За замовчуванням Django підтримує можливість роботи з СКБД SQLite та реалізує об'єктно-реляційну модель (ОРМ) для спрощення роботи. Описані особливості дозволяють уникнути процесу написання модулю виконання запитів мовою SQL [10, с. 153].

Розподілена архітектура дозволяє використання одного серверу для підключення великої кількості клієнтів. Цей тип архітектури, за необхідності, дозволяє також використовувати окремий сервер з СКБД та знизити навантаження на сервер (рис. 3).

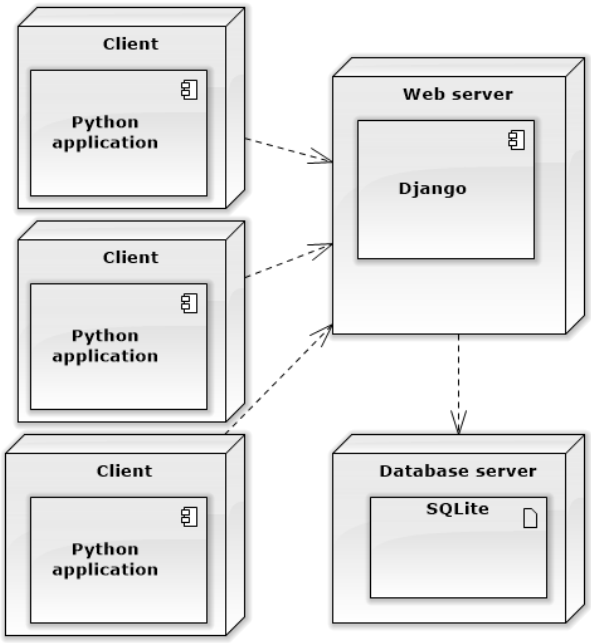

Рис. 3. Діаграма розгортання застосунку моніторингу даних апаратно-технічного стану локальної мережі ЗВО

Аналіз зібраних даних з метою виявлення аномальних показників навантаження на апаратне забезпечення потребує використання додаткових бібліотек, наприклад Matplotlib для графічного відображення отриманих даних.

Matplotlib є бібліотекою для роботи з двовимірною графікою на мові програмування Python [11, с. 3], за допомогою якої можна створювати високоякісні рисунки технічного характеру: двомірні, тримірні графіки тощо. При реалізації проєкту моніторингу апаратно-технічного стану локальної мережі ЗВО Matplotlib може бути застосований для відображення навантаження задіяного об'єму оперативної пам'яті у різні дні (рис. 4).

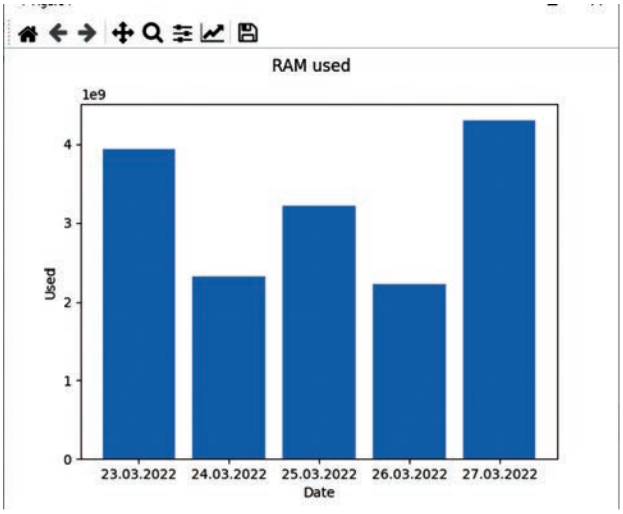

Рис. 4. Використання Matplotlib для візуалізації навантаження на апаратно складову

Візуалізація навантаження дає змогу визначити деякі значні відхилення, але більш дієвим є застосування засобів статистичної обробки, а саме бібліотеки Statistics [12, с. 43]. Ця бібліотека дозволяє визначити середній рівень навантаження апаратних складових різними методами:

– **мода** значення, що зустрічається найчастіше;

– **медіана** середній елемент сортованого ряду значень;

$$\widetilde{X}_i = \frac{X_j + X_{j+1}}{2}, \qquad (2)$$

де j=1/2 N, N кількість показників, X – ряду показників певної характеристики.

– **квартілі** – це певний показник, що ділить сукупність на чотири рівні частини. Найчастіше застосовують верхній (3) або нижній (4) квартіль:

$$Q_1 = (N+1) * 1/4, \qquad (3)$$

де N кількість значень характеристики.

Отже, значення показників за методом нижнього квартілю можна представити наступним чином:

$$\tilde{X} = X_{Q_1}$$

Верхій квартіль (Q_3) можна розрахувати за формулою

$$Q_3 = (N+1) * 3/4 \qquad (4)$$

На основі отриманих середніх значень можливо визначити середнє квадратичне (5), що може свідчити про появу шкідливого програмного забезпечення.

$$\text{stDev}(X) = \sqrt{\frac{\sum_{i=1}^{N}(X_i - X`)^2}{N-1}}, \qquad (5)$$

де $X`$ – середнє певної характеристики.

Висновки

В статті наведено дослідження засобів моніторингу апаратного забезпечення основі використання мови загального призначення Python. Розроблено БД для зберігання результатів моніторингу характеристик апаратної складової локальної мережі ЗВО. Запропоновано реалізацію візуального відображення зібраних даних для визначення збільшення споживання апаратних ресурсів шкідливим програмним забезпеченням. Розроблено програмні засоби статистичної обробки навантаження на апаратного забезпечення.

В подальшому планується інтеграція методів штучного інтелекту для визначення взаємозв'язків апаратного забезпечення та активності шкідливого програмного забезпечення.

Список літератури:

1. Aleksieva V., Slavov S. Managed Active directory in directory-as-a-service. Fundamental sciences and applications. Vol. 24, 2018. P. 117–122.
2. Войтович О. П., Вітюк В. О., Каплун В. А. Особливості дослідження ознак шкідливого програмного забезпечення без наявності вихідних кодів. Інформаційні технології та комп'ютерна інженерія. Випуск 3. С. 4–9.
3. Гульчак Ю. П., Теренчук А. Т. Моніторинг використання пристроїв введення даних ПК локальних мереж. Склад організаційно-програмного комітету чотирнадцятої МНТК ВОТТП 14-15. 2015.С. 205.
4. Войтович О. П., Вітюк В. О., Каплун В. А. Особливості дослідження ознак шкідливого програмного забезпечення без наявності вихідних кодів. Інформаційні технології та комп'ютерна інженерія. Випуск 3. С. 2013. 4–9.
5. Velasco-Montero D., Fernández-Berni J., Rodr'iguez-Vázquez A. Relevant Hardware Metrics for Performance Evaluation. Visual Inference for IoT Systems: A Practical Approach. Springer, 2022. P. 61–88.
6. Фісун М. Т., Журавська І. М., Горбань Г. В. Інтеграція даних мережевого трафіку мультісервісної корпоративної мережі з класами постреляційної СКБД Caché. Наукові праці. Комп'ютерні технології. Том 161, 2011. С. 105–110.
7. Ziogas A. N., Schneider T., Ben-Nun T. Productivity, portability, performance: data-centric Python. Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis(2021). P. 1–13.
8. Graeber M. Abusing Windows Management Instrumentation (WMI) to Build a Persistent, Asyncronous, and Fileless Backdoor. Black Hat. Las Vegas, NV, USA. 2015. URL: https://app.oarklibrary.com/file/2/8101c69f-f4c0-4812-adca-1051f065b155/b36aa1b0-e925-4352-9cc4-48594c709efa.pdf
9. Fisun M., Horban H., Kandyba I. Processing of Relational Algebra Expressions by the Shunting Yard Algorithm. 2019 IEEE 14th International Conference on Computer Sciences and Information Technologies (CSIT)(2019). P. 240–243.
10. Mele A. Django 3 By Example: Build powerful and reliable Python web applications from scratch. Packt Publishing Ltd, 2020. 533 p.
11. Ari N., Ustazhanov M. Matplotlib in python. 2014 11th International Conference on Electronics, Computer and Computation (ICECCO)(2014). P. 1–6.
12. Kandyba I. O., Fisun M. T. Information technology for expert evaluation processing in scenario analysis of subject domain. SWorldJournal. Issue 9. P. 43–48.

Kandyba I.O., Horban H.V., Fisun M.T., Tkachenko M.P. INVESTIGATING THE HARDWARE STATUS OF THE UNIVERSITY'S LOCAL NETWORK USING PYTHON

This paper presents a study of the hardware analysis tools available for integration with the general-purpose Python language. An analysis of current research on monitoring the hardware and technical condition of the local network was carried out. The libraries that solve the task of monitoring the loading of the hardware components of the local network have been identified. The main features of the university's local network were revealed. The description of Windows Management Instrumentation as a tool for determining the characteristics of the central processor, RAM, data storages, etc. has been given. The peculiarities of malware's impact on hardware are described. The current RAM load detection capability of psutil is presented. Consideration is given to CPU load monitoring tools. The tools to monitor data storage performance are examined. Determined an appropriate model for storing monitoring data. Developed a relational database structure that contains hardware characteristics and load data. Determined the most appropriate DBMS to store the specified data. The advantages of using SQLite to implement hardware monitoring software are presented. The architecture of distributed software for hardware monitoring is presented. It is proposed to use tools for visualization of applied data by connecting Matplotlib library. The use of Statistics library to use methods for determining the average load indicator is proposed. Methods for statistical processing of hardware resource consumption data have been implemented. The possibility of using the method of calculating the standard deviation for detecting load changes and searching for malicious software is described. Further ways of development of the proposed system of hardware state of the local network of higher education are defined.

Key words: *Python, WMI, psutil, DBMS, Matplotlib, Hardware monitoring.*