ISSN 2786-5371 print
ISSN 2786-538X online
Технології та інжиніринг, № 3(14), 2023

https://doi.org/10.30857/2786-5371.2023.3.1

УДК 004.384

Андрощук А. В.
Київський національний університет технологій та дизайну, Україна

ЗАХИСТ ВІД DDOS-АТАК НА МОВАХ ПРОГРАМУВАННЯ JAVA ТА C#

Мета. *Проаналізувати і порівняти можливості кіберзахисту у Java та C# та визначити переваги та недоліки кожної мови з точки зору безпеки програмного забезпечення.*

Методика. *В основу розробки системи ефективних засобів кіберзахисту було покладено основні механізми кіберзахисту, доступні у мовах програмування Java та C#, такі як виключення, контроль доступу, шифрування даних та перевірка вводу.*

Результати. *В ході дослідження було проведено аналіз загроз та вразливостей, що стосуються програм, написаних на мовах програмування Java та C#. Було виявлено, що такі загрози, як ін'єкція SQL-запитів, вразливості XSS (міжсайтовий скриптінг), вразливості переповнення буфера та інші, можуть становити серйозну загрозу для безпеки програмного забезпечення.*

Для захисту програм від цих загроз було розглянуто основні механізми кіберзахисту, доступні у мовах програмування Java та C#. Серед них були виокремлені виключення, контроль доступу, шифрування даних та перевірка вводу. Ці механізми можуть бути використані для запобігання різним видам атак та злому програмного забезпечення.

Порівняно можливості кіберзахисту у Java та C# та визначено переваги та недоліки кожної мови з точки зору безпеки програмного забезпечення.

Зроблено висновки про важливість кіберзахисту на мовах програмування Java та C# та надано рекомендації для розробників програмного забезпечення стосовно свідомого підходу до кіберзахисту та використання відповідних методів та технік для забезпечення безпеки своїх програм.

Наукова новизна*. Аналітично визначено переваги захисту від DDOS-атак шляхом використання обмеження кількості запитів за певний проміжок часу. А також реалізації авторизації та ролей користувачів.*

Практична цінність*. Запропоновано використання захисту від DDOS-атак шляхом обмеження кількості запитів, авторизацію користувачів та доступ по ролям, що може бути корисним для розробників програмного забезпечення, які працюють з мовами програмування Java та C#.*

Ключові слова*: кіберзахист; мови програмування Java та C#; атаки та вразливості; атаки на мережу; захист від атак; практики програмування; шифрування даних.*

Вступ. У сучасному цифровому світі, де все більше і більше даних зберігається та передається за допомогою мереж, забезпечення кібербезпеки є надзвичайно важливим завданням. Кібербезпека охоплює широкий спектр заходів та технологій, які використовуються для захисту від кібератак, шахрайства та шпигунства.

Java та C# є двома популярними мовами програмування, які використовуються для розробки веб-додатків, мобільних додатків, десктопних програм та багатьох інших проектів. Однак, якщо не вживати належних заходів безпеки, програми, написані на цих мовах програмування, можуть бути вразливі до кібератак та інших загроз [1].

Проблема полягає у тому, що кібератаки стають все більш спеціалізованими та складними, тому необхідно застосовувати належні заходи безпеки, щоб уникнути вразливості. Недостатність захисту може призвести до витоку конфіденційної інформації, втрати даних, порушення законодавства та інших наслідків, які можуть значно завдати шкоди бізнесу та привести до серйозних фінансових втрат.

Тому важливо коректно використовувати методи та технології забезпечення кіберстійкості на Java та C#, щоб забезпечити надійний захист від можливих кібератак та інших загроз. Нижче розглянуті основні методи та технології, які можна використовувати для забезпечення кіберстійкості на цих мовах програмування.

Постановка задачі. Метою дослідження є проведення детального аналізу та дослідження методів кіберзахисту на мовах програмування Java та C#. Головним завданням є визначення актуальних загроз та вразливостей, які можуть впливати на безпеку програм, написаних на цих мовах, а також вивчення ефективних методів та практик для їх запобігання. Крім того, будуть представлені конкретні методи та техніки кіберзахисту, які можуть бути застосовані у розробці програм на Java та C#. Буде досліджено використання захищених класів, валідацію даних, контроль доступу та шифрування даних як основні методи для запобігання атакам та забезпечення безпеки програм.

Результати дослідження. Для забезпечення кібербезпеки в програмах на C# існують різноманітні методи. Один з таких методів – це застосування шифрування даних. Шифрування можна використовувати для захисту конфіденційних даних, таких як паролі, ключі API або особисті дані користувачів.

Іншим методом забезпечення кібербезпеки на C# є використання мережевих механізмів безпеки, таких як SSL або TLS. Ці механізми забезпечують безпеку при передачі даних між клієнтом і сервером, під час шифрування і перевірки автентичності.

Також на C# можна використовувати механізми контролю доступу, такі як ролі користувачів і авторизація на основі ролей. Ці механізми дозволяють обмежити доступ до функцій і ресурсів програми тільки тим користувачам, які мають на це право.

Для забезпечення безпеки в програмах на C# використовуються такі техніки, як валідація даних, обмеження розміру вхідних даних, перевірка цілісності даних і контроль доступу до баз даних. Також використовуються різноманітні бібліотеки і фреймворки, що забезпечують безпеку в програмах на C#.

Недоліком методів, що базуються на захисті прикладних програм на рівні коду, є те, що вони можуть залишатися вразливими до нових видів атак, які можуть бути відкриті в майбутньому [2]. Наприклад, хакери можуть використовувати вразливості в сторонніх бібліотеках, які використовуються у програмі, щоб отримати доступ до даних.

Крім того, використання методів, які забезпечують кібербезпеку на рівні програмного коду, може бути витратним у плані часу та зусиль. При зміні коду часто потрібно проводити повторні тестування та перевірку безпеки, що може затримувати процес розробки та випуску продукту на ринок [3].

Забезпечення кібербезпеки на рівні програмного коду не є універсальним рішенням, оскільки воно не враховує відмінностей у конфігурації серверів та мереж, на яких запускається програмне забезпечення. Тому, щоб забезпечити повну кібербезпеку, потрібно використовувати комплексні методи, що враховують різні аспекти безпеки, такі як фізичний доступ до серверів, захист мережі, захист даних тощо.

Якщо використання наявних методів кіберзахисту на C# не дозволяє досягнути необхідного рівня безпеки, можна розглянути інші рішення [4].

Одним з таких рішень є використання сукупності методів для забезпечення кіберзахисту.

Також можна використовувати аудит безпеки, щоб знайти потенційні вразливості в системі та виправити їх до того, як зловмисники зможуть ними скористатися.

Нарешті, необхідною складовою забезпечення кібербезпеки є культура безпеки в компанії. Кожен співробітник повинен бути свідомим можливих загроз та знати, як поводитися в разі підозри на кібератаку [5].

На Java існує велика кількість інструментів та бібліотек, що допомагають забезпечити кіберзахист додатків. Найпоширеніші методи для забезпечення кіберзахисту на Java включають:

1. Використання SSL / TLS для шифрування даних, переданих між сервером та клієнтом. Це дозволяє забезпечити конфіденційність та цілісність даних.

2. Використання криптографічних бібліотек, таких як Bouncy Castle або Java Cryptography Extension (JCE), для забезпечення безпеки при збереженні та передачі конфіденційної інформації, такої як паролі та ключі шифрування.

3. Використання функцій хешування, таких як SHA-256 або SHA-512, для збереження паролів та інших конфіденційних даних у безпечному форматі. Хешування дозволяє забезпечити захист від атак на перехоплення та злом паролів.

4. Використання механізмів контролю доступу, таких як Java Security Manager, для забезпечення безпеки на рівні коду. Це дозволяє забезпечити захист від виконання небезпечного коду та вразливостей, які можуть призвести до витоку даних або інших проблем.

5. Використання захисту від вразливостей веб-додатків, таких як OWASP Top 10, для забезпечення безпеки веб-додатків та захисту від атак, таких як SQL Injection або Cross-Site Scripting (XSS).

6. Використання механізмів моніторингу та логування, таких як Log4j або Apache Commons Logging, для виявлення та аналізу небезпечних дій або помилок в додатках.

Хоча існують різноманітні методи для забезпечення кіберзахисту на Java, вони також мають свої недоліки. Один з найбільш відчутних недоліків полягає в тому, що вони можуть бути обійдені зловмисниками. Наприклад, шифрування може бути розшифроване, якщо зловмисник знає ключ [6,7].

Іншою проблемою є те, що застосування деяких методів кіберзахисту може збільшити навантаження на систему. Наприклад, перевірка цифрових підписів може займати значну кількість часу, що може знизити продуктивність системи.

Крім того, методи кіберзахисту можуть бути складні для реалізації і підтримки, особливо в комплексних системах. Це може створювати проблеми для розробників і адміністраторів систем.

Однак, є різноманітні способи вирішення цих недоліків [8, 9]. Наприклад, можна використовувати більш складні алгоритми шифрування та автентифікації, що є більш надійними, та які важче обійти. Також можна зменшити навантаження на систему, шляхом оптимізації в роботі з даними, або використати спеціальні пристрої для шифрування та дешифрування даних.

Для розв'язання проблеми складності реалізації та підтримки методів кіберзахисту можна використовувати готові бібліотеки та фреймворки, які містять реалізації різних методів кіберзахисту та забезпечують їх просту інтеграцію в програми. Також, важливо проводити регулярну перевірку наявних методів кіберзахисту [9, 10].

На рис. 1 зображено результат успішного виконання програми, яка використовує авторизацію, захист від DDOS атак по даних користувача та IP, захист по ролях.

На рис. 2 зображено захист від DDOS атак шляхом встановлення кількості доступних запитів за хвилину.

На рис. 3 зображено захист по ролях користувачів, таким чином користувач 2, не має доступу до даних користувача 1.

На рисунку під підписом 1, зображено ввід url, по якій отримується доступ до даних.

На рисунку під підписом 2, зображено використання автентифікації, за допомогою якої передається ім’я користувача і пароль (зображено на рисунку під підписом 3) у вигляді тексту через HTTP-запит.

На рисунку під підписом 4 зображено відповідь від серверу з успішною авторизацією та доступом до даних.

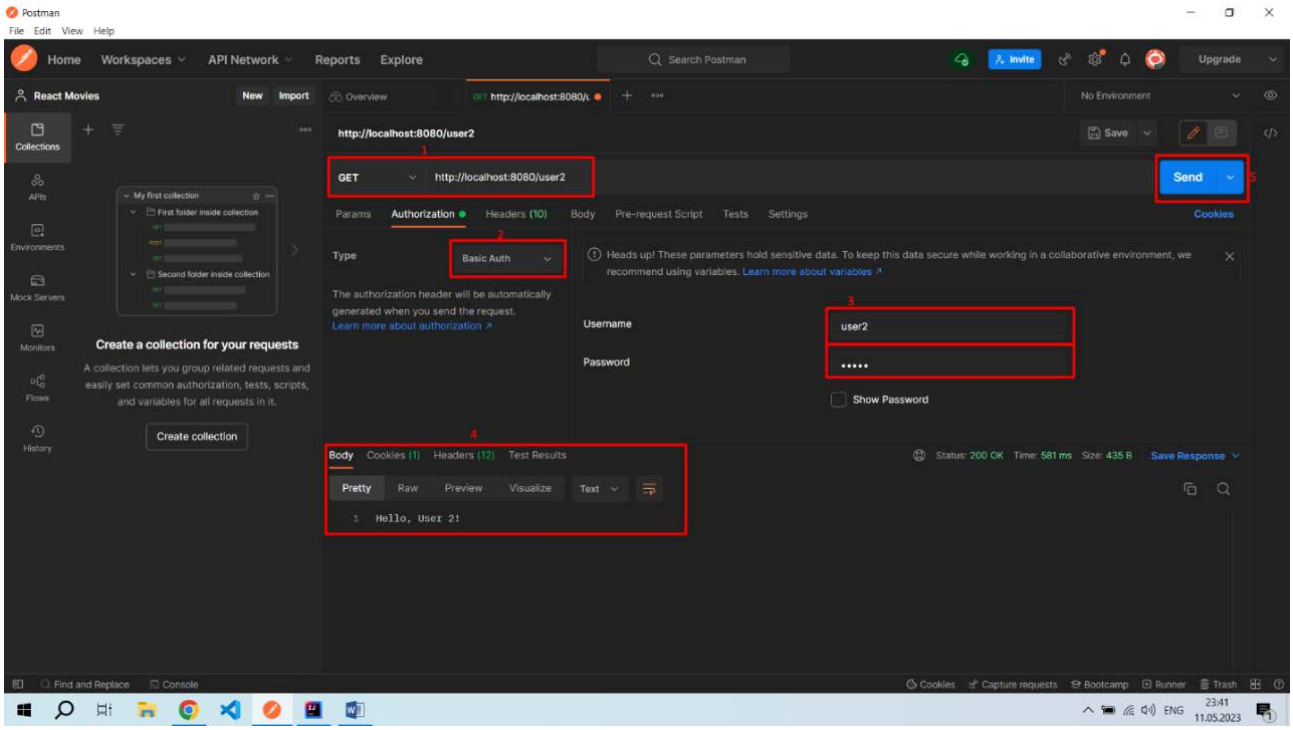

Рис. 1. **Результат успішного виконання програми**

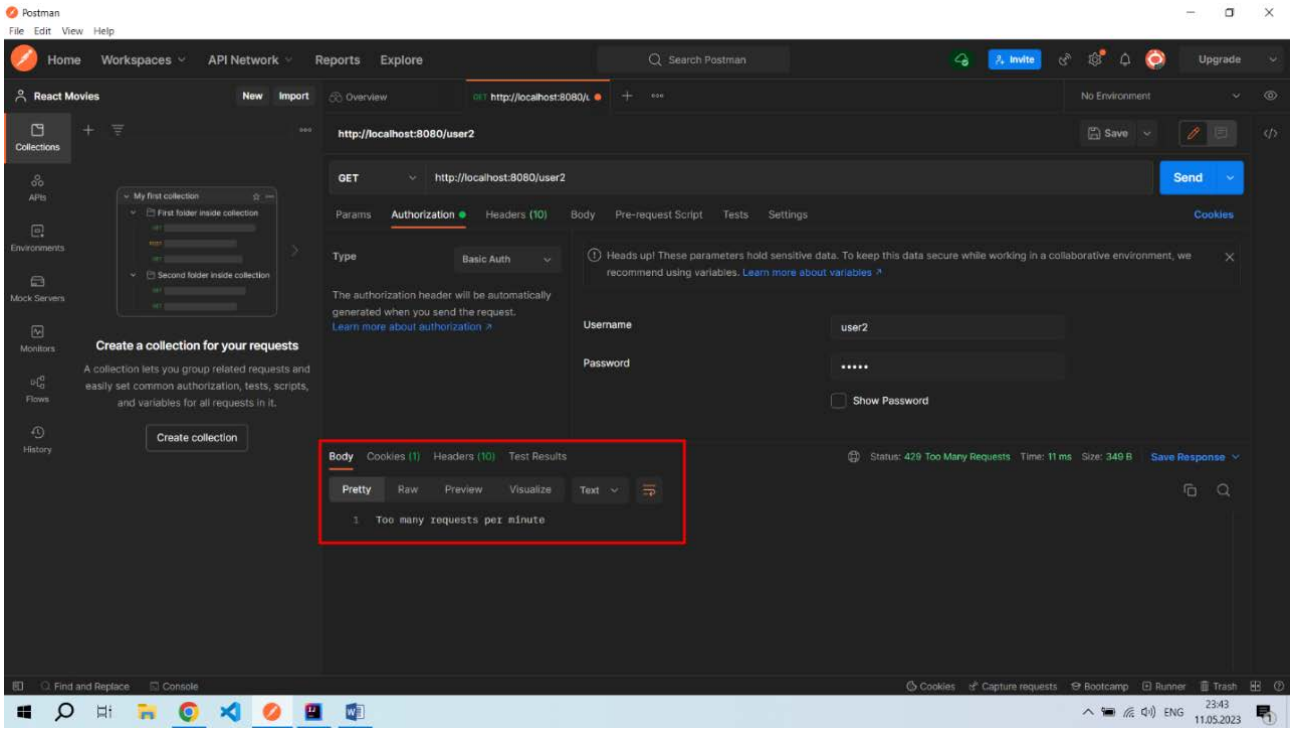

Рис. 2. **Результат виконання програми зі статусом «Перевищено ліміт запитів за хвилину»**

Висновки:

В ході аналізу можливостей кіберзахисту у мовах програмування Java та C#, було виявлено, що обидві мови надають різноманітні механізми та інструменти для досягнення безпеки програмного забезпечення. Основні механізми, такі як виключення, контроль

доступу, шифрування даних та перевірка вводу, доступні в обох мовах і можуть бути використані для захисту від різних загроз.

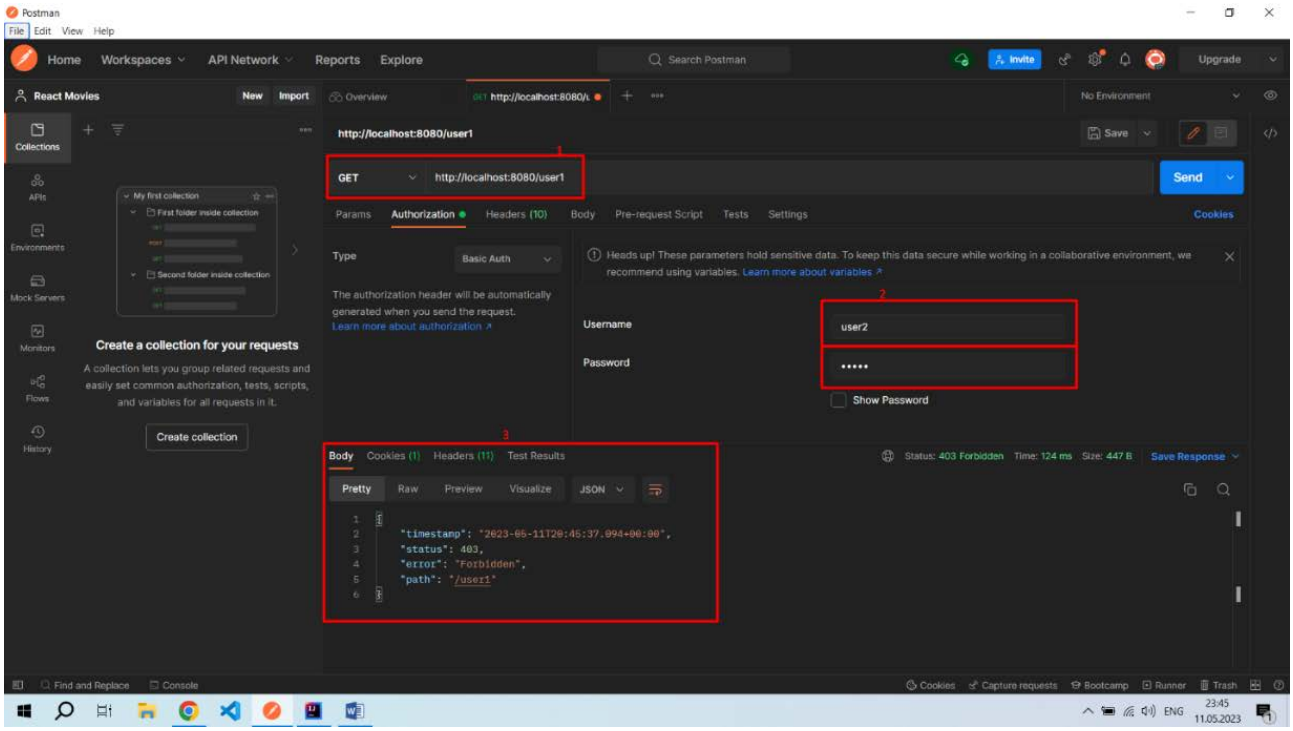

Рис. 3. **Результат виконання програми зі статусом «Заборонено»**

Однак, дослідження показало, що існують загрози та вразливості, які можуть впливати на безпеку програм, написаних на обох мовах. Зокрема, ін'єкція SQL-запитів, вразливості XSS та переповнення буфера виявлені як загрози, які можуть бути успішно використані зловмисниками для атак на програмне забезпечення.

Отже, хоча мови програмування Java та C# надають інструменти для кіберзахисту, є необхідність уважно враховувати потенційні загрози та вразливості при розробці програм. Тому розробникам важливо приділяти достатню увагу безпеці програмного забезпечення та застосовувати належні практики кіберзахисту, незалежно від обраної мови програмування.

References

1. Walther, S. (2010). ASP.NET 4 Unleashed. Indianapolis: Sams Publishing. 1680 p.
2. Howard, M., LeBlanc, D., Viega, J. (2009). 24 Deadly Sins of Software Security: Programming Flaws and How to Fix Them. New York: McGraw-Hill. 432 p.
3. Michaelis, M. (2017). Essential C# 7.0. Boston: Addison-Wesley Professional. 1008 p.
4. MacDonald, M. (2005). Pro.NET 2.0 Code and Design Standards in C#. New York: Apress. 968 p.
5. Manico, J., Detlefsen, A. (2014). Iron-Clad Java: Building Secure Web Applications. New York: McGraw-Hill Education. 456 p.
6. Hooker, D. (2014). Professional Java for Web Applications. Indianapolis: Wrox. 936 p.

Література

1. Walther S. ASP.NET 4 Unleashed. Indianapolis: Sams Publishing, 2010. 1680 p.
2. Howard M., LeBlanc D., Viega J. 24 Deadly Sins of Software Security: Programming Flaws and How to Fix Them. New York: McGraw-Hill, 2009. 432 p.
3. Michaelis, M. (2017). Essential C# 7.0. Boston: Addison-Wesley Professional. 1008 p.
4. MacDonald M. Pro.NET 2.0 Code and Design Standards in C#. New York: Apress, 2005. 968 p.
5. Manico J., Detlefsen A. Iron-Clad Java: Building Secure Web Applications. New York: McGraw-Hill Education, 2014. 456 p.
6. Hooker, D. Professional Java for Web Applications. Indianapolis: Wrox, 2014. 936 p.

7. Schneier, B. (1996). Applied Cryptography: Protocols, Algorithms, and Source Code in C. New York: John Wiley & Sons. 758 p.
8. McGraw, G. (2006). Software Security: Building Security In. Boston: Addison-Wesley Professional. 396 p.
9. Tanenbaum, A. S., van Steen, M. (2002). Distributed Systems: Principles and Paradigms. Pearson Education, Upper Saddle River. 803 p.
10. Mirkovic, J., Dietrich, S., Dittrich, D. (2016). DDoS Attacks: Evolution, Detection, Prevention, Reaction, and Tolerance. 763 p.

7. Schneier B. Applied Cryptography: Protocols, Algorithms, and Source Code in C. New York: John Wiley & Sons, 1996. 758 p.
8. McGraw G. Software Security: Building Security In. Boston: Addison-Wesley Professional, 2006. 396 p.
9. Tanenbaum A. S., van Steen M. Distributed Systems: Principles and Paradigms. Pearson Education, Upper Saddle River, 2002. 803 p.
10. Mirkovic J., Dietrich S., Dittrich D. DDoS Attacks: Evolution, Detection, Prevention, Reaction, and Tolerance, 2016. 763 p.

ANDROSHCHUK ANDRII
Kyiv National University of Technologies and Design, Ukraine
https://orcid.org/0000-0002-8452-4797
Scopus Author ID: 6506601603
E-mail: andreykoandroshchuk@gmail.com

ANDROSHCHUK A. V.
Kyiv National University of Technologies and Design, Ukraine

CYBER PROTECTION IN PROGRAMMING LANGUAGES JAVA AND C#

Goal. *Analyze and compare the cyber security capabilities of Java and C# and identify the advantages and disadvantages of each language from a software security perspective.*

Method. *The basis of the development of a system of effective cyber protection tools was the basic cyber protection mechanisms available in the Java and C# programming languages, such as exclusions, access control, data encryption and input validation.*

The results. *In the course of the study, an analysis of threats and vulnerabilities related to programs written in the Java and C# programming languages was carried out. It has been found that threats such as SQL injection, XSS (cross-site scripting) vulnerabilities, buffer overflow vulnerabilities and others can pose a serious threat to software security.*

To protect applications from these threats, the main cyber defense mechanisms available in the Java and C# programming languages have been reviewed. Among them, exclusions, access control, data encryption and input verification were singled out. These mechanisms can be used to prevent various types of attacks and software hacking.

It compares the cyber defense capabilities of Java and C# and identifies the advantages and disadvantages of each language from a software security perspective.

Conclusions are drawn on the importance of cyber security in the Java and C# programming languages, and software developers are encouraged to take a conscious approach to cyber security and use appropriate methods and techniques to secure their applications.

Scientific novelty. *Updated analysis and research in the field of cyber security in the Java and C# programming languages is offered. A comparison of cyber security capabilities in Java and C#. An analysis of their advantages and disadvantages from the point of view of software security is made, and recommendations are made for choosing a language with cyber security in mind.*

Practical value. *This article provides practical value to software developers working with the Java and C# programming languages. It offers an analysis of threats and vulnerabilities that can affect the security of programs written in these languages. This analysis helps developers to be more aware of potential risks and take appropriate measures to protect their applications.*

Keywords: *Cyber protection; Java and C# programming languages; Attacks and Vulnerabilities; Attacks on the network; Protection against attacks; Programming practices; Data encryption.*