ҚАЗАҚСТАН РЕСПУБЛИКАСЫ ҒЫЛЫМ ЖӘНЕ ЖОҒАРЫ БІЛІМ МИНИСТРЛІГІ

«Л.Н. ГУМИЛЕВ АТЫНДАҒЫ ЕУРАЗИЯ ҰЛТТЫҚ УНИВЕРСИТЕТІ» КЕАҚ

Студенттер мен жас ғалымдардың
«ǴYLYM JÁNE BILIM - 2025»
XIX Халықаралық ғылыми конференциясының
БАЯНДАМАЛАР ЖИНАҒЫ

СБОРНИК МАТЕРИАЛОВ
XX Международной научной конференции
студентов и молодых ученых
«ǴYLYM JÁNE BILIM - 2025»

PROCEEDINGS
of the XX International Scientific Conference
for students and young scholars
«ǴYLYM JÁNE BILIM - 2025»

2025
Астана

УДК 001(06)
ББК 72я631
F96

«ǴYLYM JÁNE BILIM – 2025» студенттер мен жас ғалымдардың XX Халықаралық ғылыми конференциясы = XX Международная научная конференция студентов и молодых ученых «ǴYLYM JÁNE BILIM – 2025» = The XX International Scientific Conference for students and young scholars «ǴYLYM JÁNE BILIM – 2025». – Астана: – 3813 б. - қазақша, орысша, ағылшынша.

ISBN 978-601-08-5373-7

Жинаққа студенттердің, магистранттардың, докоранттардың және жас ғалымдардың жаратылыстану-техникалық және гуманитарлық ғылымдардың өзекті мәселелері бойынша баяндамалары енгізілген.

The proceedings are the papers of students, undergraduates, doctoral students and young researchers on topical issues of natural and technical sciences and humanities.В сборник вошли доклады студентов, магистрантов, докторантов имолодыхученых по актуальным вопросам естественно-технических и гуманитарных наук.

УДК 001(06)
ББК 72я431
F96

ISBN 978-601-08-5373-7

Л.Н. Гумилев атындағы Еуразия ұлттық университеті, 2025

СЕКЦИЯ 2

СОВРЕМЕННЫЕ ИНФОРМАЦИОННЫЕ И КОММ УНИКАЦИОННЫЕ ТЕХНОЛОГИИ

Подсекция 2.2

Интеллектуальные информационные системы

255.	Сафонова Софья Александровна «Современные аспекты информационной безопасности в облачных вычислениях: модели, угрозы и методы защиты»	1034
256.	Смаилова Назгүл Батырбекқызы «Терең оқыту арқылы кітап ұсыныстарын әзірлеу: collaborative filtering, content-based және nlp әдістерінің комбинациясы»	1041
257.	Тажібай Аружан Айдосқызы, Кудубаева Сауле Альжановна «Көру қабілеті әлсіз адамдарға арналған аі дауыстық көмекші: нақты уақытта объектілерді анықтау және қашықтықты бағалау»	1046
258.	Тайжанов Азамат Жанкелдіұлы «Python тілінде фильмдердің интеллектуалды ұсыныс жүйесін әзірлеу»	1051
259.	Умирзахов Сундетали Кабылбекович «Сұраныстарды интеллектуалды талдау негізінде ұйымның сайты үшін чат-бот құру»	1055
260.	Шайхстан Марғұлан «IOT Сенсорлары негізінде ауа ластану деңгейін болжау»	1060

Подсекция 2.3

Современные тенденции в программной инженерии и управлении в условиях цифровой индустрии

261.	Bekenova A.B. «Development of a registration panel for users and doctors with integration into the database»	1077
262.	Bolat A.Zh. «Data analysis methods and decision making using big data and machine learning tools»	1081
263.	Алтайұлы А. «Visual studio интегралды ортасында «қойма қызметкерлеріне арналған» мәліметтер қорын жобалау»	1086
264.	Арап А.Қ. «Ақылды сурет салушы роботты әзірлеу»	1088
265.	Артыкбекқызы А. «Ақылды үйлердегі заттар интернеті(iot) мен робототехниканың өзара әрекеттесуі»	1091
266.	Ахметова А.Д. «Тоңазытқыштағы өнімдерді бақылауға және тағам әзірлеу ұсынысын беруге арналған программалық қосымша»	1096
267.	Дәрібай Д.Д. «Робототехниканы қолдану арқылы қойма логистикасындағы қолданыстағы басқару жүйелерін талдау»	1100
268.	Жамбулов С.Ж. «Білім алушыларды информатика және программалау олимпиадаларына дайындауда жасанды интеллекттің қолданысы»	1102
269.	Каиржан Р.С. «Development of system for recognition of emotional states of employees based on computer vision methods on Raspberry Pi»	1108
270.	Кайрекенова Н.Р. «Өнеркәсіптік роботты көру үшін машиналық оқытудың заманауи тәсілдері: әдістер, деректер жиынтығы және оптимизациялау»	1111

СЕКЦИЯ 3
ЕСТЕСТВЕННЫЕ НАУКИ

21. S. S. Shah and S. P. Bendale, “An Intuitive Study: Intrusion Detection Systems and Anomalies, How AI can be used as a tool to enable the majority, in 5G era,” *ICCUBEA Int. Conf.*, Sep. 2019, doi: 10.1109/iccubea47591.2019.9128786.

22. A. Singh, K. Chatterjee, and S. C. Satapathy, “An edge based hybrid intrusion detection framework for mobile edge computing,” *Complex & Intelligent Systems*, Aug. 2021, doi: 10.1007/s40747-021-00498-4.

23. S. Sullivan, A. Brighente, S. A. P. Kumar, and M. Conti, “5G Security Challenges and Solutions: A Review by OSI Layers,” *IEEE Access*, vol. 9, pp. 116294–116314, 2021, doi: 10.1109/access.2021.3105396.

24. B. A. Tama and S. Lim, “Ensemble learning for intrusion detection systems: A systematic mapping study and cross-benchmark evaluation,” *Computer Science Review*, vol. 39, p. 100357, Feb. 2021, doi: 10.1016/j.cosrev.2020.100357.

25. R. Ullah, M. A. U. Rehman, M. A. Naeem, B.-S. Kim, and S. Mastorakis, “ICN with edge for 5G: Exploiting in-network caching in ICN-based edge computing for 5G networks,” *Future Generation Computer Systems*, vol. 111, pp. 159–174, Oct. 2020, doi: 10.1016/j.future.2020.04.033.

26. N. Yadav, S. Pande, A. Khamparia, and D. Gupta, “Intrusion Detection System on IoT with 5G Network Using Deep Learning,” Wireless Communications and Mobile Computing, vol. 2022, pp. 1–13, Mar. 2022, doi: 10.1155/2022/9304689.

27. X. Yan, Y. Xu, X. Xing, B. Cui, Z. Guo, and T. Guo, “Trustworthy Network Anomaly Detection Based on an Adaptive Learning Rate and Momentum in IIoT,” IEEE Transactions on Industrial Informatics, vol. 16, no. 9, pp. 6182–6192, Sep. 2020, doi: 10.1109/tii.2020.2975227.

28. H. Yu, H. Lee, and H. Jeon, “What is 5G? Emerging 5G Mobile Services and Network Requirements,” Sustainability, vol. 9, no. 10, p. 1848, Oct. 2017, doi: 10.3390/su9101848.

UDC 004.056

KEY ATTACKS IN WEB FORENSICS: XSS, SQL INJECTION AND RCE

Yerzhanova Yerke Yerzhankyzy
yyerzhanova@mail.ru
Master’s student at the Faculty of Information Technology
L.N. Gumilyov Eurasian National University, Astana, Kazakhstan
Scientific supervisor – S.A. Santeyeva

Information security is one of the critical subjects that requires in-depth analysis by every state. Vulnerabilities in web applications are continuously being identified, necessitating their timely detection and mitigation. Organizations must consistently conduct comprehensive testing of their applications to ensure resilience against potential threats. Web forensics is a specialized domain within web forensics that focuses on investigating incidents related to web applications, websites, and online activities. Its primary objective is identifying, analyzing, and reconstructing digital evidence following cyberattacks and security breaches.

The key areas of focus in web forensics include:

- Analysis of web server logs;
- Investigation of compromised web applications;
- Forensic analysis of browsers and user data;
- De-anonymization of attackers;
- Forensic examination of cloud-based web services [1].

This article examines the compromised web application by analyzing the aforementioned web server logs. According to BI.Zone as of November 2024, one-quarter of newly emerging vulnerabilities each month pose a significant threat to an organization's information security. In other words, if over a thousand vulnerabilities related to web applications are discovered monthly, 25% of them are classified as highly critical based on the CVSS scale. Moreover, detailed exploit examples for 4% of these vulnerabilities are publicly available on the internet. Cybercriminals continuously develop two to three new attack methods targeting websites each week. Consequently, organizations that seek to ensure their security and operational continuity must promptly release updates and notify users about potential threats [2].

By the end of 2024, XSS attacks and SQL injections accounted for 26% and 22% of detected web vulnerabilities, respectively. Additionally, attackers frequently attempt to exploit RCE (Remote Code Execution) vulnerabilities. Notably, 60% of malicious code found online was associated with attempts to exploit these vulnerabilities [2].

It is well known that websites are developed in various programming languages. However, one of the most widely used is PHP. According to the fact, 73% of identified vulnerabilities were found in websites written in PHP. Reports indicate that seven out of ten PHP-based websites, when developed without additional security tools, can be compromised. Meanwhile, vulnerabilities in JavaScript and Java accounted for 13% and 12%, respectively. C# represented only 2%, which may be attributed to its lower usage in modern web development. Nevertheless, as an older programming language, it remains susceptible to security flaws [2]. The vulnerability distribution across different programming languages is illustrated in Figure 1.

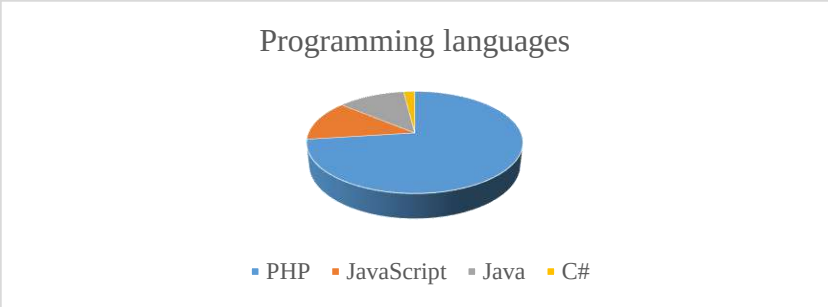

Figure 1 – Vulnerabilities in Programming Languages.

To demonstrate the exploitation of the aforementioned vulnerabilities (XSS attacks, SQL injections, and RCE), the following tools and environments were utilized: a Kali Linux machine, the Wazuh SIEM system, the OWASP Juice Shop web application with known vulnerabilities for testing, and a website developed in PHP.

Now, focusing on one of the most prevalent vulnerabilities—SQL injection—various techniques are commonly used to bypass security mechanisms. These methods include:

- Using Unicode encoding – representing special characters in encoded form, such as %27 instead of '.

- Cascading parentheses – inserting unnecessary additional parentheses and comments to disrupt security mechanisms, e.g., --, /* */.

- Hiding through comments – obfuscating queries using inline comments, for example: SELECT/**/password/**/FROM/**/users.

- Mixing uppercase and lowercase letters – altering letter cases to bypass filters, e.g., sEleCT * frOM users.

- Encoding mixing – writing parts of a query in ASCII while encoding others in hexadecimal or UTF-8 formats [3].

An example of an SQL injection detected using rules in the Wazuh SIEM system is presented in Figure 2, In this case, comment-based obfuscation was used to bypass security mechanisms [4].

Figure 2 – SQL injection.

The next widely prevalent vulnerability is the XSS attack. To bypass security mechanisms, the following techniques are considered:

- Using HTML encoding: writing `&#x3C;script&#x3E;` instead of `<script>`
- Injecting through attributes such as onerror, onclick, onmouseover: `<img src="x" onerror="alert(1)">`
- Using the javascript: prefix for links: `<a href="JaVaScRiPt:alert(1)">Click</a>`
- Encoding the payload with Base64 or URL encoding
- Exploiting non-standard or deprecated HTML tags (e.g., `<marquee>`, `<xml>`) and others [5]

To demonstrate this vulnerability, OWASP Juice Shop provides a guide. By accessing the feedback section, a malicious script can be injected into the comment field. This allows SIEM systems to create rules based on keywords such as `<script>`, `onerror=`, and `alert`. Figure 3 below illustrates an example of injecting an XSS payload into a web application [6].

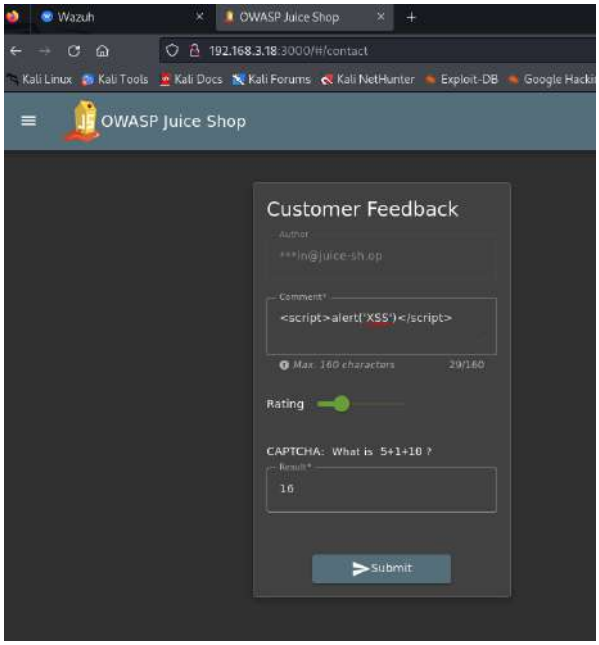

Figure 3 – XSS injecting malicious code.

The final vulnerability to be demonstrated is RCE (Remote Code Execution), a security flaw that allows an attacker to execute code or commands remotely on a server or web application. If an RCE vulnerability exists within a system, a hacker can potentially gain full control over the server. RCE vulnerabilities typically arise in the following scenarios:

- When user-supplied input is passed to system functions without proper validation. For example, insecure use of functions such as system(), exec(), eval(), and passthru() without adequate filtering.

- Lack of filtering during file uploads, which can allow an attacker to upload a malicious script or shell.

- Vulnerabilities in third-party libraries, which can introduce remote control capabilities through external modules [7].

Figures 4 and 5 illustrate an example of an RCE vulnerability in a PHP-based test website created by the author. The vulnerability allows execution of arbitrary commands, such as 'uname', which retrieves information about the operating system on which the site is running. Figure 4 shows the homepage of the author's PHP website.

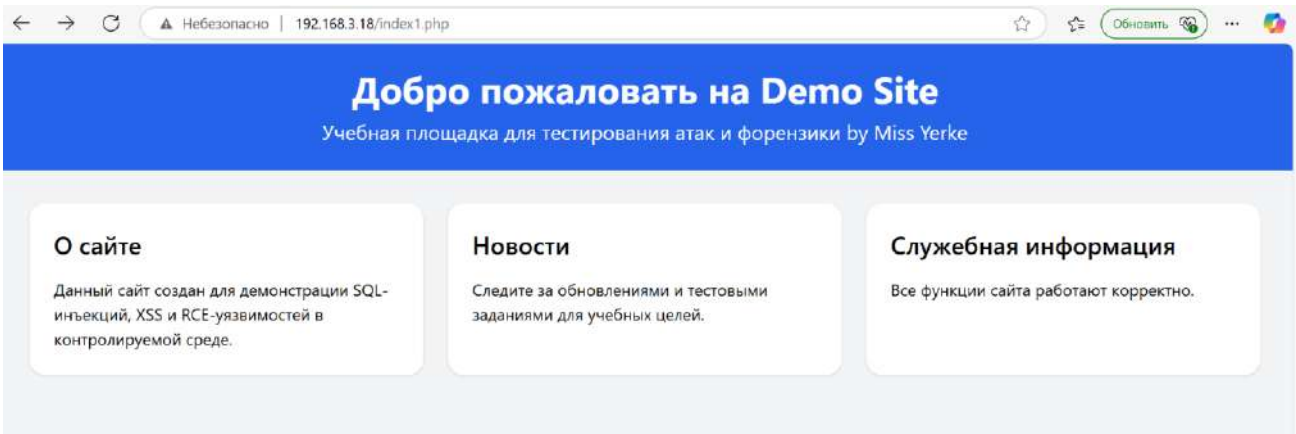

Figure 4 – Appearance of the PHP website.

Figure 5 demonstrates the RCE vulnerability by executing the `uname` command to retrieve information about the operating system. As mentioned earlier, this vulnerability allows obtaining critical data by inputting standard commands. Additionally, other commands such as `whoami`, `ls -la`, and many others can also be executed [8].

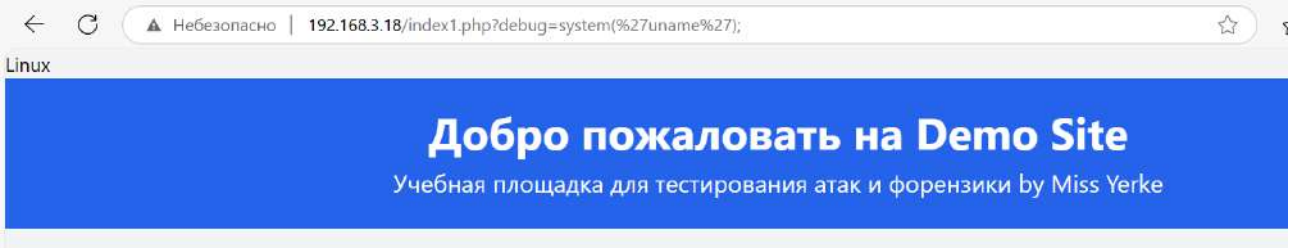

Figure 5 – RCE vulnerability.

In conclusion, XSS, RCE, and SQL injection are among the most critical attack vectors targeting web applications. XSS allows an attacker to execute malicious scripts within a user's browser, leading to session hijacking or phishing attacks. SQL injection enables unauthorized database queries, potentially exposing sensitive information or corrupting data. RCE (Remote Code Execution) allows an attacker to remotely execute arbitrary code on a server, potentially gaining full control over the system. These attacks primarily stem from inadequate input validation and insufficient filtering, posing severe security threats.

To enhance web application security, all input data must be rigorously validated and sanitized. Preventing SQL injection requires using parameterized queries and ORM instead of raw SQL queries. Defending against XSS attacks involves applying context-aware output encoding and implementing a Content Security Policy (CSP). To mitigate RCE risks, developers should avoid dangerous functions such as `eval`, `system`, and `exec`, ensuring that external input is never directly executed. Additionally, timely updates of web servers and applications, enforcing the principle of least privilege, and continuous log monitoring are essential security practices.

Literature

1. Что такое компьютерная криминалистика (форензика)? // internet-resource “Habr” – 21 august 2024 y. – URL: https://habr.com/ru/articles/837460/

2. Безопасность веб приложений // internet-resource “Tadviser” – 14 november 2024 y. – URL: https://www.tadviser.ru/index.php/%D0%A1%D1%82%D0%B0%D1%82%D1%8C%D1%8F:%D0%91%D0%B5%D0%B7%D0%BE%D0%BF%D0%B0%D1%81%D0%BD%D0%BE%D1%81%D1%82%D1%8C_%D0%B2%D0%B5%D0%B1-%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B9

3. SQL Injection: Bypassing Common Filters // internet-resource “PortSwigger” – URL: https://portswigger.net/support/sql-injection-bypassing-common-filters

4. Mohammad Asif Matin, Letsdefend Alert: SOC165 — Possible SQL Injection Payload Detected / Mohammad Asif Matin // internet-resource “Medium” – 18 november 2024 y. – URL: https://medium.com/@mmatinau/letsdefend-alert-soc165-possible-sql-injection-payload-detected-056044964ba1

5. XSS Filter Evasion: How Attackers Bypass XSS Filters – And Why Filtering Alone Isn’t Enough // internet-resource “Acunetix” – 12 february 2025 y. – URL: https://www.acunetix.com/blog/articles/xss-filter-evasion-bypass-techniques/

6. Detecting XSS (Cross-Site Scripting) Attacks in SOC Environment // internet-resource “LetsDefend” – 26 august 2024 y. – URL: https://letsdefend.io/blog/detecting-xss-cross-site-scripting-attacks-in-soc-environment

7. 5 способов получить RCE на практике // internet-resource “Habr” – 8 january 2025 y. – URL: https://habr.com/ru/articles/872110/

8. Remote code execution (RCE) // internet-resource “Invicti” – URL: https://www.invicti.com/learn/remote-code-execution-rce/

FUNDAMENTALS OF MODERN CRYPTOGRAPHY: FROM ENCRYPTION TO DIGITAL SIGNATURES.

Zhakay Aikerim
aikerim.zhakai@mail.ru
Status (Master’s student) L.N. Gumilyov Eurasian National University, Astana
Supervisor – Tashenova Zh. M.

ABSTRACT

The issue of changing and protecting information so that it cannot be read by other outsiders has been thinking about humanity since ancient times. The history of cryptography is developing in parallel with the history of human language. Even the writing itself was originally a cryptographic system because, in ancient times, only a select few knew how to write. When writing became widespread, cryptography developed as a separate science. Cryptographic systems developed well during the years of the First and Second World Wars. Since the post-war time, the emergence of computing equipment up to this point accelerated the creation and improvement of cryptographic methods. In the modern world, cryptography has entered into everything, from mobile phones to e-mail programs, bank cards, smart homes and even into medical implants. This is quite different from the past, where cryptography had been traditionally confined to very specific applications, especially government communications and banking systems. As a consequence of the pervasiveness of crypto algorithms, people should understand how they work and how they can be applied in practice. This article addresses this issue by providing a comprehensive introduction to modern applied cryptography and provides the reader with a deep understanding of how modern cryptographic schemes work.

1.INTRODUCTION

The swift advancement of internet technology has rendered information security, driven by network communication encryption and cryptography, an essential aspect of contemporary society. Cryptography, the science of securing information, has evolved from ancient techniques of secret communication to highly advanced systems underpinning modern digital security. With the advent of the internet, cryptographic methods play a crucial role in safeguarding sensitive data, enabling private communication, and building trust in digital interactions. At its core, cryptography involves transforming data into a secure format that can only be read or decrypted by those who have the appropriate key or knowledge. From encrypting emails and safeguarding financial transactions to securing national defense systems, cryptography ensures its basic principles such as the confidentiality, integrity, authenticity of information and non-repudiation. This article explores the core principles of modern cryptography, including encryption, symmetric and asymmetric cryptography, and digital signatures.

1.2 LITERATURE REVIEW